

Elia Esther Cano Acosta

Teoría de Grafos Aplicada al Análisis
de Redes S4PR y Subclases: El
Problema de Cálculo de los Cerrojos
Mínimos

Departamento
Informática e Ingeniería de Sistemas

Director/es
Colom Piazuelo, José Manuel

<http://zaguan.unizar.es/collection/Tesis>

Tesis Doctoral

TEORÍA DE GRAFOS APLICADA AL ANÁLISIS DE REDES
S4PR Y SUBCLASES: EL PROBLEMA DE CÁLCULO DE LOS
CERROJOS MÍNIMOS

Autor

Elia Esther Cano Acosta

Director/es

Colom Piazuelo, José Manuel

UNIVERSIDAD DE ZARAGOZA

Informática e Ingeniería de Sistemas

2011

**Teoría de Grafos Aplicada al Análisis de Redes S^4PR y
Subclases. El Problema de Cálculo de los Cerrojos
Mínimos.**

Elia Esther Cano Acosta

Departamento de Informática e Ingeniería de Sistemas

TESIS DOCTORAL

Director: José Manuel Colom Piazuelo

Universidad de Zaragoza

Zaragoza, España

Septiembre 2011

*Porque el Señor da la sabidura,
de su boca proceden la ciencia y la inteligencia.*

Proverbios 2,6.

*Dedicado a:
mis padres Tomasa y Agustin,
mi esposo Carlos y mi hijo Juan Carlos.*

Agradecimientos

Doy gracias a Dios por permitirme culminar mis estudios doctorales y a la Santísima Virgen María por todas sus “apariciones”.

Deseo expresar mi agradecimiento al Dr. José Manuel Colom Piazuolo, director de esta tesis doctoral, por toda la ayuda brindada en la realización de la misma. De igual forma, deseo expresar mi agradecimiento a todos los miembros del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza por todo su apoyo, en especial a: Dr. Joaquin Ezpeleta, Dr. Unai Arronategui, Dr. Santiago Velilla, Dr. José Angel Castellanos, Dra. Sandra Baldassarri, Dr. José Jesús Guerrero, Dr. Carlos Sagüés, Dr. Javier Minguez y Dr. Eduardo Mena. También, agradezco a los miembros del Grupo de Ingeniería de Sistemas de Eventos Discretos (GISED), en especial a: Dr. Manuel Silva, Dr. Javier Campos, Dr. José Merseguer, Dr. Fernando Tricas y a la Dra. Simona Bernardi. Extiendo mi agradecimiento a: Angel Vargas, José Luis Gutierrez, Andrés Ruiz y a todo el personal de secretaría del Departamento en especial a: Alberto Ferrer y José Antonio Azorin.

Deseo agradecer a: Dr. Victor Sánchez, Ing. Ulises Gurruchaga, Dr. Hector Montes y Dra. Delva Batista por toda la confianza depositada en mí.

De igual forma, expreso mi agradecimiento a mis padres y hermanos. A mis tíos Carmen y José Miranda e hijos. A Bolivar Rovetto, gracias por todos sus sabios consejos. A Natalia Rovetto quien siempre estuvo anuente a realizar todo los trámites necesarios durante estos años en los cuales he estado fuera de mi país y por su apoyo a mis padres. A mi esposo Carlos quien me animó a iniciar este doctorado, gracias por todo el apoyo brindado. A mi hijo Juan Carlos por darme cada día un motivo más para continuar. A: Erika Diaz, Gloriela Zapata e hijas, Olga Yolanda Beitia, Nivia Gutierrez, Zuly, Leonor, Eneida y Edith, a los esposos Sr. Arturo López y Sra. Conchi Vaquero de la Comunidad Familia, Evangelio y Vida, gracias por todas sus oraciones.

Quiero agradecer a todas las autoridades, personal docente y administrativo de la Universidad Tecnológica de Panamá, en especial a todos mis compañeros de trabajo y ex-estudiantes del Centro Regional de Bocas del Toro. Deseo expresar un agradecimiento muy personal al Ing. Salvador Rodriguez, Rector de la Universidad en el período 2003-2008 y a la Ing. Marcela Paredes, actual Rectora de la Universidad. De igual forma, agradezco a: Ing. Lionel Pimentel, Lic. Alex Matus, Ing. José Mendoza, Lic. Juana Aparicio, Dr. Clifton Clunie, Ing. Raúl Barahona, Dr. Alexis Tejedor, Dr. Ignacio Chang, Dr. Domingo Vega, Lic. Dariana Atencio, Lic. Abdiel Saavedra, Lic. Zuleyka Blandford, Lic. Damaris Najera,

Tec. Maria de Araúz, Lic. Lisenia Sanjur, Lic. Elena Mendoza, Ing. Francisco Flores, Lic. Jorge Avila y Sra. Betty Tejedor por toda la ayuda brindada.

Extiendo mi agradecimiento a todos mis compañeros del Laboratorio de Becarios, en especial a: Pilar Albert, Javier Celaya, Javier Mauricio Antelis, Dorian Gálvez, Juan Pablo Lopez-Grao y Diego Pérez. A mis ex-compañeros del Grupo de Sistemas de Información Distribuidos (SID), en especial a: Raquel Trillo, Carlos Bobed y al Dr. Jorge Gracia por su ayuda desinteresada.

A todas las personas que han contribuido de una forma u otra a que el sueño de unos años atrás sea hoy una realidad, mil gracias y que Dios los bendiga.

Esta Tesis Doctoral ha sido financiada parcialmente por la Universidad Tecnológica de Panamá, Fundación Carolina, Universidad de Zaragoza, Banco Santander Central Hispano, y realizada en el marco del European Community's Seventh Framework Programme bajo el proyecto DISC (Grant Agreement n. INFISO-ICT-224498) y el proyecto CICYT-FEDER DPI2006-15390.

Índice general

Introducción	XVII
1. Modelos para el Diseño de Sistemas de Asignación de Recursos	1
1.1. Introducción	2
1.2. Modelos de Asignación de Recursos utilizando las Redes de Petri	5
1.3. Análisis y Síntesis de Sistemas de Asignación de Recursos modelados con Redes de Petri	8
1.4. Metodología Modular para construir Redes de Petri para un Sistema de Fabricación Flexible visto como un Sistema de Asignación de Recursos	14
1.5. Un Ejemplo Introdutorio	17
1.6. Semántica de la Clase S^4PR en Sistemas de Fabricación Flexible	20
1.7. Conclusión y Justificación	22
2. El Conjunto de Cerrojos Mínimos en Redes S^4PR	25
2.1. Introducción	26
2.2. Caracterización de la Vivacidad en Redes S^4PR	27
2.3. Propiedades de los Cerrojos Mínimos en redes S^4PR	30
2.4. Número de Cerrojos Mínimos en Redes S^4PR y Subclases	37
2.5. Conclusión	43
3. La Relación de Poda entre Cerrojos en Redes S^4PR	45
3.1. Introducción	46
3.2. La Relación de Poda entre Cerrojos	47
3.3. El Grafo de Poda	56
3.4. Propiedades del Grafo de Poda	62
3.5. La Relación Extendida de Poda en redes S^4PR	67
3.6. El Grafo de Poda Extendido	71
3.7. Conclusión	72

4. Cerrojos en S^4PR y Grafo de Poda	75
4.1. Introducción	75
4.2. Cálculo de Cerrojos a partir del Grafo de Poda	85
4.3. Un Nuevo Algoritmo para calcular los cerrojos mínimos de una red S^4PR	91
4.4. Análisis del Coste Computacional del Cálculo de Cerrojos Mínimos mediante el Grafo de Poda	97
4.5. Teorema de Vivacidad y Grafo de Poda en Redes S^4PR	98
4.6. Conclusión	108
5. Especialización para Subclases de S^4PR	109
5.1. Introducción	110
5.2. Redes $L - S^3PR$	110
5.3. Redes S^3PR	118
5.4. Redes $SOAR^2$	123
5.5. Conclusión	129
6. Conclusiones	133
Bibliografía	135
A. Definiciones Básicas y Terminología sobre las Redes de Petri	143
A.1. Conceptos Básicos y Terminología	143
A.1.1. Funcionamiento de una Red de Petri	145
A.1.2. Componentes Estructurales de Redes de Petri	145
A.2. Propiedades de las Redes de Petri	146
A.2.1. Propiedades Estructurales	146
A.2.2. Propiedades Dinámicas	146
B. Definiciones Básicas sobre los Grafos	147

Índice de figuras

1.1.	Grafo de Asignación de Recursos [SGG05] entre los procesos p_1 y p_3 . . .	3
1.2.	Abstracción de la Secuencia de operaciones de un sistema con dos tipos diferentes de procesos P_1 y P_2	6
1.3.	Red de Petri modelando la secuencia de operaciones de los procesos P_1 y P_2 de la figura 1.2.	7
1.4.	Cerrojo “malo” $D = \{r_1, r_2, r_3, op_{12}, op_{121}, op_{13}, op_{131}, op_{23}\}$ de la red de Petri modelada en la figura 1.3 que se resalta mediante los correspondientes lugares punteados.	10
1.5.	Un red de Petri con problemas de bloqueo.	11
1.6.	Grafo de Alcanzabilidad de la red mostrada en la figura 1.5. El marcado muerto de la red se indica mediante líneas punteadas.	12
1.7.	Cerrojo “malo” $D = \{R, S, B, D\}$ de la Red de Petri modelada en la figura 1.5.	12
1.8.	Red resultante después de agregar el lugar de control CP a la red de la figura 1.5, esta red es viva.	13
1.9.	Grafo de Alcanzabilidad de la red de la figura 1.8 en donde se puede observar que al agregar el lugar CP se ha eliminado el marcado muerto. . .	13
1.10.	Esquema de una celda de producción de un SMF.	18
1.11.	Planes de Producción (a) Parte P1 (b) Parte P2.	19
1.12.	La RdP modelando el procesamiento de las partes de la celda de producción.	20
2.1.	Un ejemplo de una red S^4PR	31
2.2.	Una red S^4PR que no contiene cerrojos mínimos en la clase \mathcal{D}^4	33
2.3.	El cerrojo mínimo $\{c, r\}$ está estrictamente contenido en el soporte del p-semiflujo mínimo \mathbf{y}_r	35
2.4.	Una red S^4PR con un número de cerrojos mínimos igual a la cota inferior del lema 7.	38
2.5.	Una red S^4PR con un número de cerrojos mínimos igual a la cota superior del lema 7.	39

2.6. Red de Petri perteneciente a la clase $L - S^3PR$	42
3.1. El lugar p es esencial para el cerrojo mínimo $D_r = \{p, u, r\}$ y el lugar q es esencial para el cerrojo mínimo $D_s = \{q, v, s\}$	48
3.2. El lugar p es esencial para el cerrojo mínimo $D_r = \{p, r\}$, y sigue siendo esencial en el cerrojo $D_{rs} = D_r \cup D_s = \{p, q, r, s\}$ dado que es el único lugar de entrada a la transición $t_1 \in \bullet r$	49
3.3. El lugar b es esencial para los cerrojos D_x y D_y . Al realizar la operación de unión entre ambos cerrojos no existe ningún lugar que se convierta en no-esencial.	50
3.4. Diagrama que ilustra como lugares proceso de los cerrojos D_R y D_S llegan a ser no-esenciales en el cerrojo $D_R \cup D_S$	51
3.5. Una red S^4PR	52
3.6. El cerrojo $D_r = \{r, p_1, p_2, p_3\}$ está relacionado consigo mismo por la relación de poda pero es un cerrojo mínimo.	55
3.7. El cerrojo no-mínimo $D_r = \{a, b, r, c\}$ no está relacionado consigo mismo por la relación de poda.	56
3.8. Una red S^4PR que no tiene cerrojos mínimos que contengan más de un recurso.	59
3.9. El Grafo de Poda, G , de la red de la figura 3.8 y las funciones que etiquetan sus vértices y arcos.	59
3.10. Diagrama en donde un conjunto de lugares pertenecientes a un cerrojo se convierten en no-esenciales.	60
3.11. El lugar $A1$ perteneciente al cerrojo mínimo D_r contenido en el conjunto $P_{parcial}$ no puede ser podado.	61
3.12. Una red S^4PR en donde el lugar $p1$ es esencial para el cerrojo D_R y el mismo sólo puede ser eliminado del cerrojo D_R cuando se realiza la operación de unión entre los cerrojos D_R y D_S	62
3.13. Grafo de Poda, G , de la figura 3.12 en donde se puede observar que la información del bucle en el vértice R es necesaria para poder calcular el cerrojo mínimo $D_{RS} = \{R, S, p2, p3, p4, p6\}$ resultante de la unión de los cerrojos mínimos de un recurso D_R y D_S , $D_{RS} = D_R \cup D_S$	63
3.14. El Grafo de Poda, G , de la red de la figura 2.5.	64
3.15. El Grafo de Poda G y los Subgrafos $G1, G2, G3$ and $G4$ de la figura 2.2.	65
3.16. Una red de la clase S^3PR	68
3.17. Una red S^4PR	72
3.18. El Grafo de Poda PG y Grafo de Poda Extendido EPG de la red de la figura 3.17 utilizando todos los cerrojos mínimos de la red.	73

4.1. Red de Petri S^4PR	79
4.2. Evolución del número de filas durante la ejecución del algoritmo de cálculo de cerrojos mínimos de la red de la figura 4.1 utilizando el método [ECS93].	84
4.3. Una red S^4PR en donde el cerrojo S_G que contiene a los recursos U y V es no-mínimo porque contiene el cerrojo mínimo D_U	86
4.4. Un Grafo de Poda Extendido G de la red de la figura 4.3 en donde se puede observar que el cerrojo D_U se poda a sí mismo el conjunto de lugares $K_G(U) = \{a, b\}$	88
4.5. Una red S^4PR en donde el cerrojo S_G que contiene a los recursos $R1$ y $R2$ es no-mínimo porque contiene el cerrojo mínimo D_{R1}	89
4.6. Un Grafo de Poda Extendido G de la red de la figura 4.5 en donde se puede observar que el cerrojo D_{R1} se poda a sí mismo el lugar $K_G(R1) = \{a\}$	89
4.7. Un grafo de poda G de la red de la figura 4.5 en la que se han utilizado dos cerrojos mínimos, cada uno asociado a un recurso, $R1$ y $R2$, respectivamente.	90
4.8. Una red S^4PR con dos cerrojos mínimos conteniendo más de un recurso.	93
4.9. El Grafo de Poda G de la red de la figura 4.8 y subgrafo fuertemente conexo $G1$ caracterizando D_{ST}	96
4.10. Una red S^4PR utilizada para mostrar en su grafo de poda la espera circular que existe entre los recursos A, B y C	101
4.11. Grafo de poda G de la red de la figura 4.10.	102
4.12. Red utilizada para mostrar los marcados que inevitablemente conducen a un bloqueo.	106
4.13. Grafo de poda G de la red de la figura 4.12.	107
5.1. Red de Petri de la Clase LS^3PR	113
5.2. Grafo de poda de la red de la figura 5.1.	116
5.3. Red de Petri de la Clase S^3PR	121
5.4. Grafo de poda de la red de la figura 5.3.	124
5.5. Red de Petri de la Clase $SOAR^2$	127
5.6. Grafo de poda de la red de la figura 5.5.	130

Índice de cuadros

3.1. Valores de la función $K_G(r)$ para el Grafo de Poda, G , de la figura 3.9.	63
4.1. Ubicación de los lugares procesos marcados en los K_G de los vértices del grafo de poda 4.13.	105
4.2. Tabla que muestra la solicitud de marcas para el vértice S en el grafo G .	108
5.1. Elementos de la red $L - S^3PR$ de la figura 5.1.	112
5.2. Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red $L - S^3PR$ de la figura 5.1.	113
5.3. Elementos de la red S^3PR de la figura 5.3.	120
5.4. Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red S^3PR de la figura 5.3.	121
5.5. Elementos de la red $SOAR^2$ de la figura 5.5.	128
5.6. Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red $SOAR^2$ de la figura 5.5.	129

Introducción

El análisis de aspectos relacionados a la asignación de recursos es una precondition para el diseño y control de muchos Sistemas Dinámicos de Eventos Discretos. Ejemplos de esta clase de sistemas, en los que una visin como Sistemas de Asignación de Recursos es una fase del diseño, puede ser encontrada en el contexto de los Sistemas de Fabricación Flexible, aplicaciones de Ingeniería de Software, Sistemas de Flujo de Trabajo, Sistemas Logísticos, etc.

El análisis de Sistemas de Asignación de Recursos de un sistema requiere un proceso de abstracción con el propósito de retener solamente aquellos aspectos relevantes de la representación de la asignación de recursos. La formulación de esta abstracción de estos problemas conducidos por la aplicación en términos de Redes de Petri han sido un exitoso enfoque en muchos campos [Rev05, ECM95, LZ09, LGC10, WLR⁺09, RCC10a, RCC10b], conduciendo a varias subclases de modelos de Redes de Petri, con una caracterización basada en una estructura específica. La subclase de redes S^3PR es probablemente la más ampliamente estudiada en la literatura aún en nuestros días. En estas subclases, la propiedad de vivacidad ha sido caracterizada en términos de algún cerrojo de la red [Rev05, ECM95]. Una de la clase más general de estas subclases de redes con una sólida teoría de análisis desarrollada es la clase S^4PR [Tri03, Col03].

En la subclase de redes S^4PR , la propiedad de vivacidad ha sido caracterizada en términos de algunos cerrojos de la red: [Rev05, Tri03]. Además, la síntesis de modelos vivos en S^4PR (denominados *forzando la vivacidad*¹ por algunos autores) está basada en técnicas que, en muchos casos, requiere conocer los cerrojos malos de la red [Rev03, TGVCE05]. Por lo tanto, la disponibilidad de algoritmos eficientes para el cálculo de los cerrojos mínimos u otros conjuntos de cerrojos es indispensables en los diferentes dominios que trabajan con esta clase de modelos de Redes de Petri.

Muchos algoritmos, para este propósito, han sido propuestos en los últimos años: [LZ08, LZJ08, WLJZ09, Cha09, XZW⁺11], todos ellos diseñados para las redes S^3PR . Estos algoritmos presentan una buena ejecución para esta clase de redes debido a las restricciones estrictas impuestas por las mismas. Sin embargo, los mismos no pueden ser utilizados para el cálculo de los cerrojos mínimos en las redes S^4PR , dado que las redes S^4PR son una clase más general que contienen estrictamente a las redes S^3PR . Por ejemplo, un proceso activo en una red S^3PR sólo puede utilizar un recurso en cada estado

¹*Liveness enforcing* en terminología inglesa.

alcanzable. Esta restricción no aparece en una red S^4PR .

Los algoritmos que pueden ser aplicados para el cálculo de los cerrojos mínimos en las redes S^4PR son aquellos aplicados a redes de Petri generales tales como: [BL89, CX97, ECS93, Lau87]. Todos estos algoritmos trabajan operando con los lugares y las transiciones de la red y están relacionados al problema del cálculo de las direcciones extremales de un cono convexo positivo y ellos están basados en invariantes del método de eliminación Fourier–Motzkin para resolver sistemas de inecuaciones lineales [Wil86]. En estos métodos la idea básica es eliminar variables del sistema agregando a ellos todas las inecuaciones resultantes de las combinaciones lineales positivas de pares de inecuaciones. El método procede eliminando todas las variables y memorizando, para cada inecuación final, los coeficientes de la combinación lineal positiva de las filas que la generaron. Estos coeficientes dicen que lugares pertenecen al cerrojo. La dificultad de cálculo del método consiste en el gran número de inecuaciones agregadas en el proceso de eliminación de variables. Muchas de ellas son redundantes y pueden ser excluidas. Las diferentes variantes publicadas [CS91] buscan el objetivo común de romper el crecimiento de las inecuaciones eliminando las redundantes por medio de heurísticas sofisticadas. En todo caso, los algoritmos resultantes reportados son prohibitivos desde el punto de vista de la ocupación de memoria incluso para ejemplos relativamente pequeños.

En esta memoria se propone un nuevo algoritmo que trabaja con objetos de más alto nivel que los lugares y transiciones permitiendo una mayor eficiencia, al menos, en la ocupación de memoria en los pasos intermedios. Estos objetos de más alto nivel son los cerrojos mínimos de la red que contienen solamente un recurso. La existencia de uno de estos cerrojos mínimos para cada recurso de la red está garantizado por la definición de la clase (ésta requiere un p -semiflujo mínimo por recurso). Se probará adicionalmente que para cada subconjunto de recursos existe, al menos, un cerrojo mínimo conteniendo estos recursos. Por lo tanto, el algoritmo propone la construcción de cada cerrojo mínimo por medio de la unión de los cerrojos mínimos con un recurso correspondiente a los recursos incluidos en el cerrojo a ser construidos. El resultado de esta unión es un cerrojo, pero no mínimo. Por esta razón se introduce una relación de poda en el conjunto de los cerrojos mínimos con un recurso, para remover los lugares no–esenciales de cada cerrojo mínimo con un recurso.

La representación de la relación de poda por medio de un grafo nos permite obtener los cerrojos mínimos calculando y manipulando los subgrafos fuertemente conexos máximos del grafo de la relación (sus componentes fuertemente conexos). Por lo tanto, la memoria requerida es muy pequeña y del orden del tamaño del grafo de poda.

Este trabajo contribuye a las Redes de Petri, específicamente a la clase S^4PR en:

1. Nuevas propiedades estructurales para las redes S^4PR .

2. Cotas inferior y superior de los cerrojos mínimos para las redes S^4PR , $L - S^3PR$, S^3PR y $SOAR^2$.
3. Caracterización de los cerrojos mínimos por el conjunto de recursos que contienen.
4. Nuevas herramientas: Relación de Poda y Grafo de Poda, que nos permiten obtener los cerrojos mínimos del conjunto de cerrojos que una red S^4PR posee.
5. Algoritmo para calcular los cerrojos de una red S^4PR que contengan determinados recursos.
6. Algoritmo para calcular los cerrojos mínimos para redes S^4PR .
7. Especialización de los algoritmos de cálculo de cerrojos mínimos para subclases de S^4PR utilizando el Grafo de Poda.

La presente memoria se ha organizado de la siguiente manera. En el capítulo 1 se presenta una introducción a los modelos para el diseño de Sistemas de Asignación de Recursos y se definen los elementos que lo componen. De igual forma, se presentan las principales restricciones sobre las cuales se derivan diversos enfoques en la fase del diseño que se pueden encontrar en la literatura. También, se presenta una introducción al modelado de Sistemas de Asignación de Recursos utilizando las Redes de Petri enfocándonos en la clase de redes que se utilizará en el desarrollo de esta memoria.

El capítulo 2 está dedicado a explicar las propiedades estructurales que poseen los cerrojos en las redes S^4PR . Estas propiedades las podemos derivar de la propia definición de la clase. Además, veremos que el conjunto de cerrojos mínimos se puede particionar en clases, en donde, cada una de estas clases está caracterizada por el número de recursos pertenecientes a cada cerrojo dentro de la misma. También, se caracterizan los cerrojos mínimos por el conjunto de recursos que contienen y se indica que el lugar reposo no puede pertenecer a un cerrojo mínimo con recursos. Finalmente, se indican las cotas inferior y superior de los números de cerrojos mínimos de una red S^4PR y de las subclases $L - S^3PR$, S^3PR y $SOAR^2$.

En el capítulo 3 se presenta la *Relación de Poda* entre los cerrojos mínimos de un solo recurso, la cual nos permite determinar los elementos candidatos a ser podados del conjunto de cerrojos mínimos de un solo recurso en la operación de unión. Para representar la relación de poda sobre un conjunto de cerrojos conteniendo cada uno de ellos solo un recurso se define el *Grafo de Poda* y sus tres funciones de etiquetado. Este grafo será utilizado en diversas aplicaciones como se explicará en el siguiente capítulo. Además, se presenta las propiedades que posee el grafo de poda. A partir de la relación de poda se define la *Relación de Poda Extendida* la cual constituye una generalización de la anterior.

El objetivo de la relación de poda extendida es considerar otros conjuntos de cerrojos que no contengan el mismo conjunto de recursos que los antes considerados en la relación de poda. Entre los nuevos cerrojos a considerar se incluyen los cerrojos no-mínimos y cerrojos conteniendo más de un recurso. La principal diferencia entre ambas relaciones de poda la podemos ubicar en la no-minimalidad de los cerrojos.

El capítulo 4 muestra diversas aplicaciones en la que se utiliza el grafo de poda de una red S^4PR . Una de las aplicaciones más generales en la que podemos utilizar este grafo es para calcular los cerrojos de una red S^4PR que contengan determinados recursos. Este resultado se deriva de la operación de unión de los cerrojos mínimos de un solo recurso explicado de manera informal en el capítulo anterior.

El cálculo de los cerrojos mínimos de una red S^4PR es otra aplicación del grafo de poda. En esta aplicación es necesario utilizar las herramientas de poda explicadas en el capítulo anterior para cribar entre el conjunto de cerrojos aquellos que son mínimos. Para hacer este trabajo de manera más eficiente se ha desarrollado un nuevo algoritmo que evita tener que evaluar muchas uniones no-productivas de cerrojos mínimos de la clase \mathcal{D}^1 . De manera similar, utilizando el grafo de poda se presenta la interpretación del Teorema de vivacidad en redes S^4PR .

En el capítulo 5 se analizan las subclases $L - S^3PR$, S^3PR y $SOAR^2$, basados en la definición de cada una de éstas redes, se presentan las propiedades que nos permiten determinar los cerrojos mínimos que pertenecen a la clase de cerrojos mínimos de un recurso. Se caracteriza la relación de poda para éstas redes y la representación de las mismas en el grafo de poda. Finalmente, se presenta la especialización de sus algoritmos de cálculo de cerrojos mínimos.

Capítulo 1

Modelos para el Diseño de Sistemas de Asignación de Recursos

Resumen

Un Sistema de Asignación de Recursos está compuesto por un conjunto de procesos los cuales comparten en forma competitiva un conjunto finito de recursos. Esta relación de competencia por los recursos entre los procesos puede causar problemas de bloqueos.

Los problemas de bloqueo que pueden surgir en los sistemas diseñados con el enfoque de los sistemas de asignación de recursos pueden ser analizados y solucionados de una manera más eficiente mediante la utilización de modelos matemáticos tales como las Redes de Petri. Utilizando esta perspectiva, diversos sistemas pueden ser modelados, entre ellos, los Sistemas de Fabricación Flexible. La formulación de estos sistemas bajo este enfoque han conducido a varias subclases de modelos de Redes de Petri, con una caracterización basada en una estructura específica. Una de las clase más general de estas subclases de redes con una sólida teoría de análisis desarrollada es la clase S^4PR .

En este capítulo se introducen los modelos para el diseño de Sistemas de Asignación de Recursos y se definen los elementos que los componen. De igual forma, se presentan las principales restricciones que estos sistemas poseen. También, se introduce el modelado de Sistemas de Asignación de Recursos utilizando las Redes de Petri enfocándonos en la clase de redes que se utilizará en el desarrollo de esta memoria y se explican aspectos relacionados al análisis y síntesis de estos modelos. Finalmente, se presentan los modelos de Asignación de Recursos para los Sistemas de Fabricación Flexible, se describe la metodología modular usada para su construcción y su interpretación semántica.

1.1. Introducción

Un Sistema de Asignación de Recursos (*RAS*¹) [PS89, SGG05, GS08] está compuesto por un conjunto de procesos los cuales comparten en forma competitiva un conjunto finito de recursos. Enfocándonos en un dominio de aplicación dado, un *proceso* es un programa en ejecución, una parte que entra al sistema para ser procesada, etc. Los *recursos* se dividen en varios tipos, constando cada uno de ellos de un cierto número de instancias o copias. Son tipos de recursos: impresoras, robots, máquinas, espacio de memoria, herramientas, ciclos de CPU, etc. Una cantidad igual de cada tipo de recurso constituye una instancia del mismo, esto es, si un sistema tiene tres impresoras, entonces el tipo de recurso impresora tiene tres instancias.

Un proceso debe solicitar un recurso antes de utilizarlo y liberarlo luego de usarlo. Adicionalmente, un proceso puede solicitar la cantidad de recursos que necesite para llevar a cabo las tareas o actividades que tenga asignadas. Obviamente, el número de recursos solicitados no puede exceder el total de recursos disponibles en el sistema.

En la literatura, diferentes clases de Sistemas de Asignación de Recursos han sido considerados. Las diferencias entre ellos se basan principalmente en las restricciones impuestas en el uso de los recursos por cada proceso y en las restricciones en la estructura del conjunto de secuencias de procesamiento.

Enfocándonos, en primer lugar, en las restricciones impuestas en el uso de los recursos por cada proceso podemos decir que en un sistema pueden existir recursos de varios tipos, estos pueden ser consumibles o conservativos (reutilizados en serie). En el caso de recursos conservativos un amplio conjunto de clases de sistemas pueden ser considerados empezando por aquellos en los cuales solo una copia del recurso puede ser utilizada en cada paso del procesamiento [ECM95, EGVC98], hasta aquellos en los cuales se puede usar un multiconjunto de recursos [TGVCE98, Rev98, TCE99].

Con respecto a las restricciones impuestas en la estructura del conjunto de secuencias de procesamiento, un sistema de asignación de recursos puede clasificarse en:

1. *Secuencial (S-RAS)*: en donde los procesos para terminar exitosamente la tarea asignada siguen una secuencia de pasos o estados en relación con los recursos necesarios para realizar las tareas de procesamiento que se pueden resumir en el siguiente protocolo:
 - a) Solicitud de los recursos del sistema a ser utilizados por el proceso.
 - b) Adquisición del recurso: la cual depende de su disponibilidad. Si el recurso se

¹“Resource Allocation Systems” en terminología inglesa.

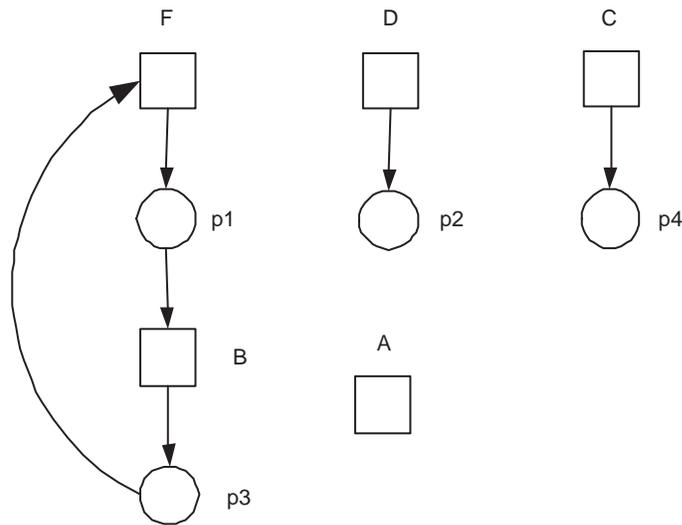


Figura 1.1: Grafo de Asignación de Recursos [SGG05] entre los procesos $p1$ y $p3$.

encuentra disponible le será otorgado al proceso, en caso contrario, el proceso deberá esperar hasta que el mismo se encuentre disponible y le sea otorgado.

- c) Liberación del recurso: esta condición se da cuando el proceso ha terminado de utilizar el recurso del sistema.

2. *No Secuencial (NS-RAS)*: en donde al menos un proceso tiene una naturaleza no secuencial.

La relación de competencia por los recursos entre los procesos en los sistemas de asignación de recursos puede causar problemas de *bloqueos*.

Un conjunto de procesos estará bloqueado cuando todos los procesos del conjunto se encuentren esperando por al menos un recurso que otro proceso del conjunto esté utilizando. Por ejemplo, consideremos un sistema en el cual cuatro procesos $p1, p2, p3, p4$ se ejecutan concurrentemente. El sistema cuenta con los siguientes recursos A, B, C, D, E, F , en donde, cada uno de ellos posee una instancia del recurso. En un instante dado, los procesos solicitan y adquieren sus recursos de la siguiente forma: $p1$ adquiere el recurso F , $p2$ adquiere el recurso D , $p3$ adquiere el recurso B y $p4$ adquiere el recurso C . A continuación dos de los procesos hacen la siguiente solicitud: $p1$ solicita el recurso B y $p3$ solicita el recurso F , dado que ambos recursos no se encuentran disponibles, cada uno de los procesos $p1$ y $p3$ debe esperar hasta que los recursos sean liberados. Esta situación de espera entre estos dos procesos no terminará debido a que ninguno de ellos liberará el

recurso que tiene asignado hasta adquirir el recurso solicitado y terminar la tarea en la cual es necesario el recurso que tiene en uso. Por lo tanto, los procesos p_1 y p_3 se encuentran bloqueados. La figura 1.1 muestra un *Grafo de Asignación de Recursos* entre los procesos del sistema del ejemplo anterior en un instante determinado de su ejecución. En el grafo los recursos son representados mediante cuadrados, los procesos mediante círculos y un arco dirigido desde el recurso al proceso muestra la asignación del recurso al proceso mientras que un arco dirigido del proceso al recurso muestra la solicitud del recurso por el proceso.

En un sistema que realiza sus actividades de forma concurrente utilizando un conjunto compartido de recursos, si se da una situación de bloqueo entonces se cumplen simultáneamente las siguientes cuatro condiciones [CES71], (*condiciones necesarias para la aparición de bloqueos*):

1. *Exclusión mutua de recursos*: cada instancia de cada recurso del sistema no puede ser utilizado por más de un proceso simultáneamente.
2. *Retención y espera*: si un proceso usa alguna instancia de algún recurso y necesita un nuevo recurso para cambiar su estado, este cambio de estado no puede llevarse a cabo si el nuevo recurso solicitado no se encuentra disponible por lo que el proceso quedará a la espera del recurso solicitado reteniendo los recursos que ya tenía asignados.
3. *Sin desalojo*: un proceso liberará los recursos que esté utilizando voluntariamente una vez dicho proceso haya completado la tarea para la cual necesita el recurso en cuestión.
4. *Espera circular*: se ha alcanzado un estado en el cual existe una cadena cerrada de procesos tal que cada proceso está utilizando algún recurso que es requerido por el siguiente proceso en la cadena para cambiar su estado y dado que esta cadena es cerrada, ninguno puede continuar.

Los problemas de bloqueo que pueden surgir en los sistemas diseñados con el enfoque de los sistemas de asignación de recursos pueden ser analizados y solucionados de una manera más eficiente mediante la utilización de modelos matemáticos tales como las Redes de Petri [Sil85, Mur89]. Utilizando esta perspectiva, diversas clases de sistemas pueden ser modeladas tales como: Sistema de Fabricación Flexible, Sistemas de Transporte, Redes de Interconexión, Sistemas de Transacciones basados en Internet, Sistemas de Administración de Flujo de Trabajo, etc. El modelado de estos sistemas bajo este enfoque han conducido a varias subclases de modelos de Redes de Petri, con una caracterización basada en una estructura específica. Una de las clase más general de estas subclases de redes con una sólida teoría de análisis desarrollada es la clase S^4PR [Tri03, Col03], otras

clases más generales aunque con un menor número de resultados analíticos son las redes *SPQR* [LGC10].

En esta memoria se considera el enfoque de los sistemas de asignación de recursos secuenciales en la fase de diseño para los Sistemas de Fabricación Flexible modelados usando la subclase *S⁴PR* de las Redes de Petri como herramienta matemática para el estudio de los problemas de bloqueo que pueden surgir en estos tipos de sistemas. Estos modelos constan de las siguientes características:

1. Sistemas secuenciales de procesamiento (S-RAS).
2. Uso conservativo de los recursos del sistema: esta condición impone que los recursos no se pueden crear ni destruir.
3. La flexibilidad en el encaminamiento es permitida: un proceso puede seguir diferentes rutas en el sistema y la decisión puede ser hecha durante el procesamiento.

El resto del capítulo se organiza como sigue. En la sección 1.2 se hace una introducción a los modelos de asignación de recursos utilizando las Redes de Petri. En la sección 1.3 se explican aspectos relacionados al análisis y síntesis de modelos de Sistemas de Asignación de Recursos utilizando las Redes de Petri. En la sección 1.4 se explica con más detalle la metodología para la construcción de modelos de asignación de recursos para los Sistemas de Fabricación Flexible. En la sección 1.5 se presenta un pequeño ejemplo que sirve en la sección 1.6 para dar una interpretación semántica de la red en término de elementos del sistema a modelar. Finalmente, la sección 1.7 presenta algunas conclusiones del capítulo.

1.2. Modelos de Asignación de Recursos utilizando las Redes de Petri

Las Redes de Petri han sido utilizadas para modelar un amplio conjunto de sistemas de asignación de recursos. En este contexto dos elementos son usualmente utilizados al clasificar los sistemas: la estructura de las Redes de Petri modelando las secuencias de procesamiento y la forma en la cual se permite el uso de los recursos. Por ejemplo, la figura 1.2 muestra la abstracción modelada mediante un grafo acíclico de la secuencia de operaciones de un sistema en el cual existen dos tipos diferentes de procesos *P1* y *P2*. La información en este grafo está compuesta por dos conjuntos, en donde, el primer conjunto contiene el recurso en uso y el segundo conjunto la operación que se está realizando.

El proceso *P1* solicita el recurso *r1* para realizar la operación *op11*, a continuación puede solicitar el recurso *r2* para llevar a cabo la operación *op12* o el recurso *r3* para

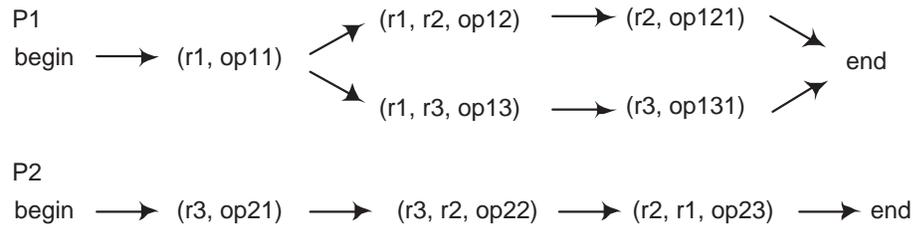


Figura 1.2: Abstracción de la Secuencia de operaciones de un sistema con dos tipos diferentes de procesos $P1$ y $P2$.

realizar la operación $op13$. Una vez concluída alguna de las dos operaciones $op12$ ó $op13$ el recurso $r1$ es devuelto al sistema. El recurso $r2$ es utilizado por la operación $op121$ y devuelto al sistema una vez se ha terminado dicha operación y el proceso $P1$ termina. De manera similar, el recurso $r3$ es utilizado por la operación $op131$ hasta finalizar la misma, entonces el recurso $r3$ será devuelto al sistema y el proceso $P1$ termina. Por otro lado, el proceso $P2$ solicita el recurso $r3$ para efectuar la operación $op21$, entonces solicita el recurso $r2$ para realizar la operación $op22$. Una vez terminada esta operación se solicita el recurso $r1$, se devuelve el recurso $r3$ y se procede a realizar la operación $op23$. Cuando se ha finalizado esta última operación ambos recursos $r2$ y $r1$ son retornados al sistema y el proceso termina. La figura 1.3 muestra la red de Petri que modela el sistema bosquejado en la figura 1.2.

Las restricciones impuestas en la estructura de las secuencias de procesamiento y el uso de recursos da lugar a una jerarquía de clases de sistemas a partir de las cuales si nos enfocamos al dominio de aplicación y a las restricciones utilizadas en esta memoria podemos resumir en $L - S^3PR$, S^3PR y S^4PR . Estas clases se han desarrollado para aquellos sistemas en donde se ejecutan de forma concurrente distintos tipos de procesos y en los cuales se comparte un conjunto finito de recursos. A continuación definimos sus principales características.

$L - S^3PR$: posee una estructura secuencial de los procesos en la cual no se permiten las decisiones. Además, cada proceso puede utilizar un solo tipo de recurso a la vez.

S^3PR : posee una estructura secuencial de los procesos en donde se permiten las decisiones. Al igual que la clase anterior, cada proceso puede utilizar un solo tipo de recurso a la vez.

S^4PR : es una generalización de la clase anterior cuya principal característica radica en que los procesos pueden utilizar varios tipos de recursos a la vez.

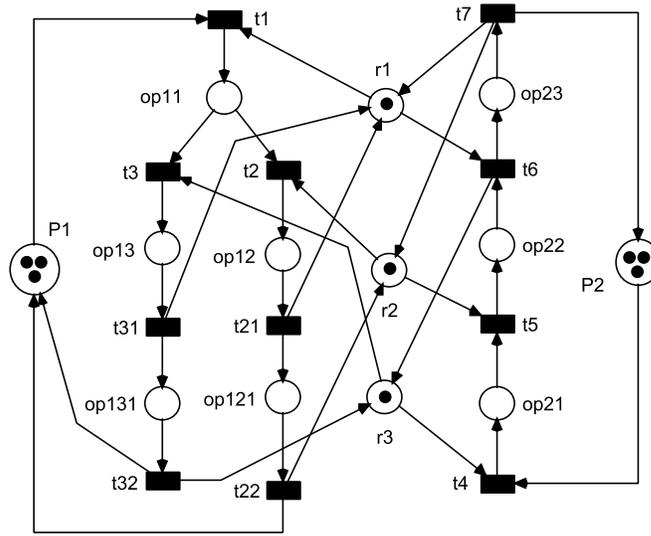


Figura 1.3: Red de Petri modelando la secuencia de operaciones de los procesos $P1$ y $P2$ de la figura 1.2.

Como se ha indicado previamente la clase S^4PR introducida en [TCE99] es una generalización de las clases introducidas previamente denominadas S^3PR en [ECM95] y $L-S^3PR$ en [EGVC98]. Además las relaciones entre estas clases son: $L-S^3PR \subset S^3PR \subset S^4PR$.

Esta clase es similar a otras clases que han sido presentadas en trabajos previos, WS^3PR en [TM95], S^4R en [BCZ97] y S^3PGR^2 en [PR01a]. A continuación se recuerda la definición de la clase S^4PR .

Definición 1. La clase de red S^4PR [TCE99].

Sea $I_{\mathcal{N}} = \{1, 2, \dots, m\}$ un conjunto finito de índices. Una red S^4PR es una Red de Petri pura, generalizada y conexa, $\mathcal{N} = \langle P, T, C \rangle$, donde:

1. $P = P_0 \cup P_S \cup P_R$ es una partición de P tal que:

- a) $P_S = \bigcup_{i \in I_{\mathcal{N}}} P_{S_i}$, $P_{S_i} \neq \emptyset$, $P_{S_i} \cap P_{S_j} = \emptyset$, $\forall i \neq j$.
- b) $P_0 = \bigcup_{i \in I_{\mathcal{N}}} \{p_{0_i}\}$.
- c) $P_R = \{r_1, \dots, r_n\}$, $n > 0$.

2. $T = \bigcup_{i \in I_{\mathcal{N}}} T_i$, $T_i \neq \emptyset$, $T_i \cap T_j = \emptyset$, para todo $i, j \in I_{\mathcal{N}}$ tal que $i \neq j$.

3. Para todo $i \in I_{\mathcal{N}}$, la subred \mathcal{N}_i generada por $P_{S_i} \cup \{p_{0_i}\} \cup T_i$ es una máquina de estados fuertemente conexa, en la que cada ciclo contiene p_{0_i} .
4. Para cada $r \in P_R$ existe un P -Semiflujo mínimo, $\mathbf{y}_r \in \mathbb{N}^{|P|}$, tal que $\{r\} = \|\mathbf{y}_r\| \cap P_R$, $\mathbf{y}_r[r] = 1$, $P_0 \cap \|\mathbf{y}_r\| = \emptyset$, y $P_S \cap \|\mathbf{y}_r\| \neq \emptyset$.
5. $P_S = \bigcup_{r \in P_R} (\|\mathbf{y}_r\| \setminus \{r\})$.

Los lugares pertenecientes al conjunto P_S se les denomina *lugares proceso*.

El lugar p_{0_i} perteneciente al conjunto P_0 representa el estado en la máquina i en el cual los procesos se encuentran inactivos o en reposo. A este lugar se le denomina *lugar reposo*.

Los recursos son representados mediante los *lugares recurso* pertenecientes al conjunto P_R .

Cada plan de proceso se modela mediante una máquina de estados fuertemente conexa, \mathcal{N}_i , (definición 1.3).

A continuación se definirá el *marcado inicial aceptable* para completar la definición anterior.

Definición 2. Sea $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ una red S^4PR . Un *marcado inicial* \mathbf{m}_0 es *aceptable* para \mathcal{N} si, y solo si:

1. $\forall i \in I_{\mathcal{N}}, \mathbf{m}_0[p_{0_i}] > 0$.
2. $\forall p \in P_S, \mathbf{m}_0[p] = 0$.
3. $\forall r \in P_R, \mathbf{m}_0[r] \geq \max\{\mathbf{y}_r[p] \mid p \in \|\mathbf{y}_r\| \setminus \{r\}\}$.

En el apéndice A se puede encontrar mayor información sobre la notación y definiciones relacionadas con las Redes de Petri.

1.3. Análisis y Síntesis de Sistemas de Asignación de Recursos modelados con Redes de Petri

Uno de los problemas fundamentales en el diseño de sistemas de asignación de recursos está relacionado con la aparición de bloqueos. Es por ello que el esfuerzo de abstraer estos sistemas y modelarlos con redes de Petri, persigue detectar en etapas tempranas del diseño estos problemas y proceder a corregir los problemas modificando la red de Petri de forma que pueda ser corregido el diseño de partida. La detección de los

problemas se basa en técnicas de análisis de la vivacidad de redes de Petri y la corrección en técnicas de síntesis que fuercen la propiedad de vivacidad.

Para las clases antes mencionadas existe una caracterización de los bloqueos basada en la estructura de la red. El objetivo de esta caracterización es sintetizar controladores para prevenir/evitar los problemas de bloqueos en los sistemas de asignación de recursos.

Las causas estructurales de no-vivacidad en esta clase de redes están relacionadas con la existencia de ciertos cerrojos, denominados en este contexto “*cerrojos malos*”. Esta característica de la vivacidad ha sido publicada de diferentes formas [TCE99][Rev03]. Estos cerrojos son importantes para caracterizar la propiedad de *vivacidad* en estos modelos.

Los cerrojos “malos” están relacionados con cierto marcado en donde los procesos activos se encuentran bloqueados debido a que sus transiciones de salida están muertas porque los recursos que necesitan no se encuentran disponibles y además tampoco se encontrarán disponibles en el futuro. Esto sugiere que existe una espera circular entre el conjunto de procesos bloqueados. Por ejemplo en la red de la figura 1.3 existe el cerrojo “malo” $D = \{r1, r2, r3, op12, op121, op13, op131, op23\}$. En la figura 1.4 mediante líneas punteadas se han representado los lugares del cerrojo malo D que indica que tendremos problemas de bloqueos. Efectivamente, al marcarse los lugares $op11$ y $op22$ se desmarcan los recursos $r1, r2$ y $r3$, respectivamente, obteniéndose de esta forma el cerrojo “malo” $D = \{r1, r2, r3, op12, op121, op13, op131, op23\}$ en donde todas las transiciones de salida para estos lugares están muertas.

Los cerrojos son importantes para caracterizar y forzar la vivacidad en esta clase de redes. En la figura 1.6 se muestra el grafo de alcanzabilidad de la red de la figura 1.5, en esta figura se indica mediante líneas punteadas el marcado en el que se bloquea completamente la red. De manera similar, en la figura 1.7 se muestra mediante líneas punteadas los lugares del cerrojo “malo” $D = \{R, S, B, D\}$ para la red de la figura 1.5. Podemos eliminar este marcado muerto controlando el cerrojo malo mediante la adición de un *lugar de control*² a la red que restringirá el disparo de las transiciones del cerrojo impidiendo que se quede sin marcas.

Iterando un número finito de veces este procedimiento hasta controlar todos los cerrojos malos obtendremos una nueva red la cual será viva. Por ejemplo, en la red de la figura 1.5 se ha eliminado el marcado muerto al añadir el lugar de control CP como se muestra en la figura 1.8. De esta manera hemos eliminado el marcado muerto de la red como se puede observar en el grafo de alcanzabilidad de la figura 1.9 y la red resultante es viva.

Estas técnicas de corrección del modelo basadas en la eliminación de estados de

²Denominado *monitores* por algunos autores.

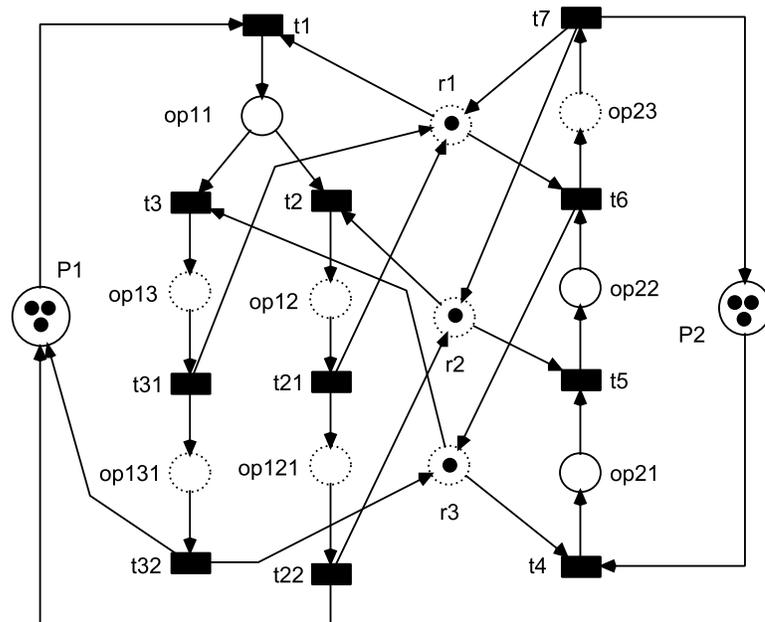


Figura 1.4: Cerrojo “malo” $D = \{r1, r2, r3, op12, op121, op13, op131, op23\}$ de la red de Petri modelada en la figura 1.3 que se resalta mediante los correspondientes lugares punteados.

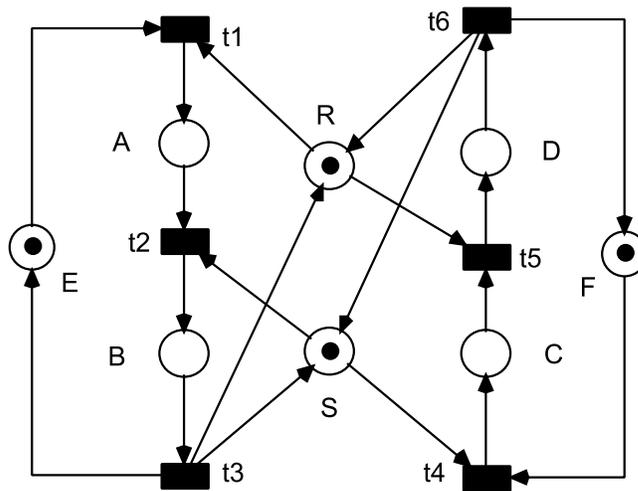


Figura 1.5: Un red de Petri con problemas de bloqueo.

bloqueo o que inevitablemente conducen a bloqueo se han desarrollado en multitud de trabajos publicados como [ECM95],[TCE99][Rev03]. Todas ellas se basan en intervenir en los cerrojos malos considerados como causa directa de la aparición de los bloqueos y que hay que conocer.

A partir de lo antes explicado, podemos decir que es muy importante tener algoritmos eficientes que nos permitan calcular esta clase de cerrojos “malos” para estas redes. Existen muchos algoritmos [Lau87, ECS93, BM94, CX97, LZ04, CFP05, Cha06, LL07, LZJ08, Cha08, WLJZ09, Cha09, XZW⁺11] para calcular los cerrojos, pero todos ellos son muy costosos en la cantidad de memoria ocupada en los pasos intermedios del cálculo. Esto se debe a que estos algoritmos trabajan con la matriz de incidencia y durante su ejecución pueden requerir una cantidad de memoria de orden exponencial con respecto al número de lugares de la red.

En esta memoria se ha desarrollado un nuevo algoritmo, como se explicará en el capítulo 4, para calcular los cerrojos que resulta más eficiente en la ocupación de memoria. Este algoritmo toma ventaja sobre los anteriores porque el mismo trabaja sobre objetos de más alto nivel que sus predecesores. Mientras que los anteriores trabajan sobre los lugares y transiciones (*algunos de ellos realizando transformaciones en la red de Petri previamente a los cálculos*), el algoritmo que se presentará en el capítulo 4 trabaja con objetos de más alto nivel, estos son los cerrojos mínimos de un solo recurso de la red. El uso de estos objetos de más alto nivel para el cálculo de los cerrojos mínimos de la red mejora significativamente el tiempo y el espacio necesitado para este propósito

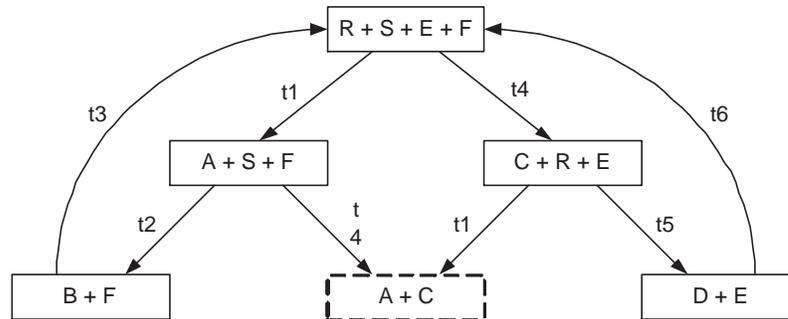


Figura 1.6: Grafo de Alcanzabilidad de la red mostrada en la figura 1.5. El marcado muerto de la red se indica mediante líneas punteadas.

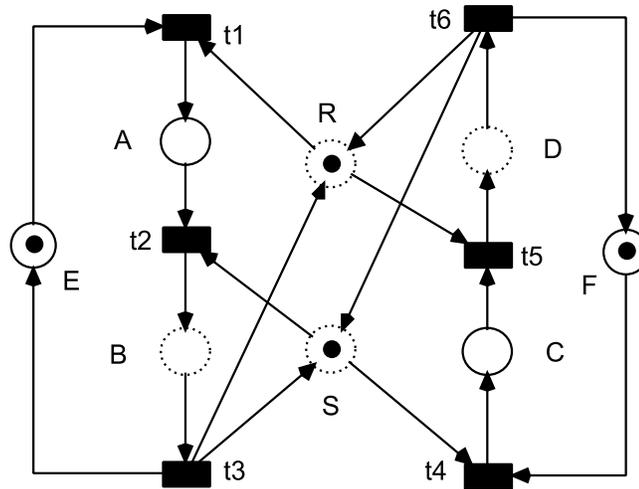


Figura 1.7: Cerrojo "malo" $D = \{R, S, B, D\}$ de la Red de Petri modelada en la figura 1.5.

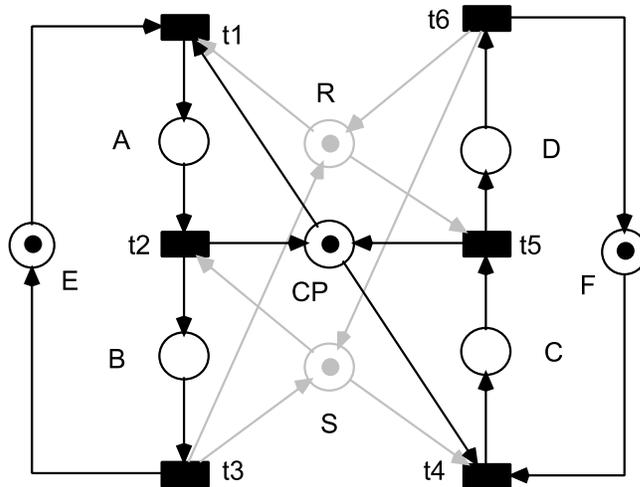


Figura 1.8: Red resultante después de agregar el lugar de control CP a la red de la figura 1.5, esta red es viva.

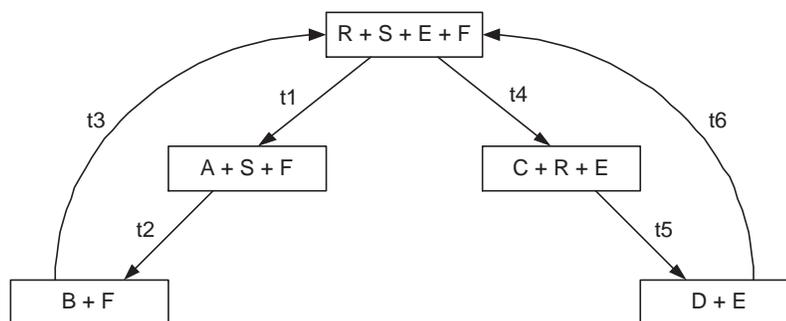


Figura 1.9: Grafo de Alcanzabilidad de la red de la figura 1.8 en donde se puede observar que al agregar el lugar CP se ha eliminado el marcado muerto.

con respecto a los algoritmos conocidos. De manera similar podemos comentar que el algoritmo propuesto resulta más económico que los algoritmos que utilizan métodos basados en formulación lógica [BAEE97, CFP02] o incluso en aquellos predecesores que utilizan la teoría de grafos [BL89, YTW98, JPH99], esto se debe a que los cerrojos mínimos de un solo recurso se obtienen a partir de las propiedades estructurales de las redes S^4PR y de la propia definición de la clase como se explicará en el capítulo 2.

1.4. Metodología Modular para construir Redes de Petri para un Sistema de Fabricación Flexible visto como un Sistema de Asignación de Recursos

Bajo la perspectiva de los Sistema de Asignación de Recursos, un Sistema de Fabricación Flexible, es percibido como un conjunto de procesos (partes, trabajos, etc) siguiendo unos planes de producción predefinidos solicitando en una forma competitiva diferentes cantidades de un número finito de recursos compartidos (máquinas, herramientas, etc). La perspectiva adoptada corresponde al nivel de coordinación en la típica arquitectura jerárquica del control de un Sistema de Fabricación Flexible.

Los Sistemas de Fabricación Flexible son muy importantes en la manufactura debido a la flexibilidad que proporcionan, la cual permite cambiar rápidamente los ambientes de producción a un bajo coste, adaptando así la producción a las nuevas demandas del consumidor. De esta forma, los sistemas de producción pueden mejorar su productividad, confiabilidad y disponibilidad en términos de poco esfuerzo y tiempo requerido para realizar los cambios. La flexibilidad que ofrecen los Sistemas de Fabricación Flexible a este entorno se puede interpretar en varios sentidos: flexibilidad a la hora de introducir nuevas gamas de productos en el proceso de producción (flexibilidad a largo plazo) y flexibilidad a la hora de producir simultáneamente varias gamas de productos (flexibilidad a corto plazo).

En esta sección definiremos los Sistemas de Fabricación Flexible, luego se explicará la metodología modular que existe para construir Redes de Petri para un Sistema de Fabricación Flexible visto como un sistema de asignación de recursos. A partir de estos modelos se analizarán los problemas de bloqueos que pueden surgir y las diversas soluciones que se pueden brindar.

Los Sistemas de Fabricación Flexible son una clase de sistemas de producción concebida para la manufactura de un conjunto de diferentes tipos de productos. Cada tipo de producto sigue una secuencia predefinida de operaciones llamada *planes de producción* solicitando en forma competitiva un conjunto de recursos compartidos del sistema como

son robots, máquinas, herramientas, etc. Esta relación de competencia entre diferentes planes de producción ejecutándose concurrentemente en un Sistema de Fabricación Flexible puede causar situaciones de bloqueo. Rigurosamente hablando, un bloqueo es un estado del sistema en el cual algunos planes de producción no pueden terminar.

El análisis de los aspectos de la asignación de recursos es una precondition para diseñar y controlar Sistemas de Fabricación Flexible. Este análisis persigue una eficiente caracterización de los bloqueos originados a partir de la perspectiva de la asignación de los recursos que puedan ser utilizados para sintetizar controladores con el objetivo de prevenir/evitar/detectar bloqueos en el sistema. Las Redes de Petri han sido utilizadas en el modelado de Sistemas de Asignación de Recursos para este propósito [ECM95, BA96, EGVC98, TCE99, PR00, Cha06, Cha08, LZJ08, WLJZ09, XZW⁺11].

Las metodologías usadas para construir modelos basados en el enfoque de los Sistemas de Asignación de Recursos en la fase del diseño usualmente tiene una naturaleza modular, donde los módulos son los planes de producción que se componen via la fusión de un conjunto de recursos compartidos.

La mayoría de los enfoques para construir el modelo en Redes de Petri de un Sistema de Fabricación Flexible visto como un Sistema de Asignación de Recursos asumen una metodología modular de tres pasos [Col03]:

1. *Caracterización de los Planes de Producción:* Cada plan de producción modela las diferentes actividades llevadas a cabo en el procesamiento de un tipo de parte. En lo que sigue se considerarán planes de producción que solo transformarán o transferirán una parte. En ningún caso se considerarán operaciones de ensamblado o desensamblado. Es decir, una parte no perderá su identidad e integridad a lo largo del proceso. También, establece la forma en la cual los recursos son utilizados en dicho procesamiento.

Los planes de producción, pueden ser modelados en Redes de Petri a través de una máquina de estados fuertemente conexas, \mathcal{N} . Los lugares de \mathcal{N} , están relacionados con las diferentes operaciones de transformación o de transporte llevadas a cabo sobre las partes. Dichas operaciones pueden ser: transformaciones, manejo, etc. A estos lugares se les denomina lugares proceso.

Cada parte que entra al sistema es un proceso. Su avance se modela como marcas que se mueven a través de \mathcal{N} . En un plan de producción se pueden distinguir los puntos de entrada de la materia prima y los puntos de salida de los productos terminados. Una parte que entra en un plan de producción avanza hacia su punto de salida a través de las transiciones contenidas en \mathcal{N} .

La ejecución de un plan de producción se realiza por la ejecución de una ruta de producción. En un mismo plan de producción pueden existir varias rutas de

producción. Una ruta de producción es una secuencia de transiciones disparables en \mathcal{N} cuya ocurrencia representa la producción de un producto terminado.

2. *La incorporación de los Recursos a cada Plan de Producción:* Los recursos son los elementos físicos que componen un Sistema de Fabricación Flexible, por ejemplo: máquinas, robots, herramientas, buffers, etc. El conjunto de recursos del sistema puede ser clasificado en:

- a) aquellos recursos que hacen alguna transformación a las partes, llamados *procesadores*, entre ellos podemos encontrar tornos, fresadoras, sierras, trituradoras, etc.
- b) aquellos recursos que no transforman las partes, llamados *manipuladores*, entre ellos podemos encontrar robots, almacenes, cinta transportadora, etc.

Cada uno de estos recursos posee una capacidad la cual es finita. Esta capacidad es modelada por medio de un marcado inicial mientras que el tipo recurso es representado mediante un lugar.

Los arcos que unen los lugares recursos con las transiciones modelan el cambio de estado de los recursos a medida que la parte evoluciona en el sistema. Por ejemplo, los arcos de salida desde un lugar recurso hacia una transición modelan la adquisición de los recursos mientras que los arcos que van de las transiciones a los lugares recursos modelan su liberación.

Cada estado en un plan de producción tiene asociado el conjunto de recursos requeridos en cada paso del procesamiento. Esto impone que cada paso del procesamiento use al menos un recurso. De igual forma, se establece que cada recurso debe ser reutilizado en serie, esto es, los recursos no pueden ser creados ni destruidos.

3. *La construcción del Modelo Global por la composición de los Planes de Producción con Recursos:* En un Sistema de Fabricación Flexible existe un conjunto de planes de producción, un tipo por cada tipo de producto. Para obtener el modelo global del Sistema de Asignación de Recursos se debe componer los planes de producción mediante los recursos requeridos. Esta composición está basada en la fusión de los lugares recurso que representan el mismo tipo de recurso en los diferentes planes de producción. El marcado inicial de los recursos después de la fusión es normalmente calculado como el máximo de los marcados iniciales de las instancias que han sido fusionadas. La estrategia usada para construir el modelo global ha sido aplicada en trabajos anteriores tales como: [LT79, BK90, HC92, ECM95, TGVCE98, XHC96, PR00].

1.5. Un Ejemplo Introductorio

En esta sección se introduce un ejemplo [CRC10a] con el objetivo de presentar de una manera intuitiva algunos de los conceptos que se van a utilizar a la siguiente sección.

La figura 1.10 muestra el esquema de una celda de producción de un Sistema de Fabricación Flexible la cual procesa dos tipos diferentes de partes $P1$ y $P2$. Las partes sin manufacturar (denominadas también materias primas de las partes) llegan a la celda a través de una cinta transportadora. De manera similar, las partes cuyo procesamiento en la celda ha finalizado (producto terminado) son evacuadas de la misma a través de otra cinta transportadora.

La celda está compuesta por dos máquinas $M1$ y $M2$ y por dos robots $R1$ y $R2$. Cada una de las máquinas posee una capacidad de procesamiento igual a uno. Por otro lado, los robots transportan las partes dentro de la celda. Cada uno de estos robots puede sujetar solo una pieza a la vez, por lo cual su capacidad es de uno. El robot $R1$ puede cargar las máquinas $M1$ y $M2$ desde el punto de entrada $I1$ y puede descargar la máquina $M2$ hacia el punto de salida $O2$. Por otro lado, el robot $R2$ carga las partes a la máquina $M2$ desde el punto de entrada $I2$ y descarga las máquinas $M1$ y $M2$ hacia el punto de salida $O1$.

Los dos tipos de partes procesadas en esta celda siguen su propio plan de producción. Los planes de producción de cada tipo de parte establecen la correcta secuencia de operaciones que deben llevarse a cabo en cada parte para culminar su procesamiento. Estas secuencias son establecidas en términos de transformaciones a realizar sobre la parte. Para la celda, estas secuencias de operaciones son transformadas en pares (recurso, operación) lo cual establece, para cada operación, los recursos donde las operaciones tienen que ser realizadas. Cada secuencia de operaciones de los tipos de partes puede ser modelada por un grafo aciclico, la figura 1.11 representa el *grafo de operaciones* correspondiente a cada uno de los dos diferentes planes de producción de cada tipo de parte. La información contenida en el grafo de operaciones está compuesta por dos conjuntos. El primer conjunto contiene el tipo de parte que el sistema está procesando. El segundo conjunto indica el recurso que se está utilizando.

Cada plan de producción tiene un nodo inicial (denominados $I1$ e $I2$ en la figura 1.11) que modela la entrada de las partes a ser manufacturadas. Los otros nodos corresponden a las etiquetas de las transformaciones que los recursos realizan sobre la parte o el transporte de la parte durante su procesamiento. Finalmente, el último nodo corresponde a la salida de la celda de la parte terminada. Esto es, la materia prima de la parte del tipo $P1$ son tomadas desde el punto de entrada $I1$, luego son procesadas en la máquina $M1$ ó $M2$ para finalmente ser descargadas como producto terminado en el punto de salida $O1$. De manera similar, la materia prima correspondiente a la parte del tipo $P2$ son descargadas en el sistema desde el punto de entrada $I2$, entonces son procesadas en la máquina $M2$ y

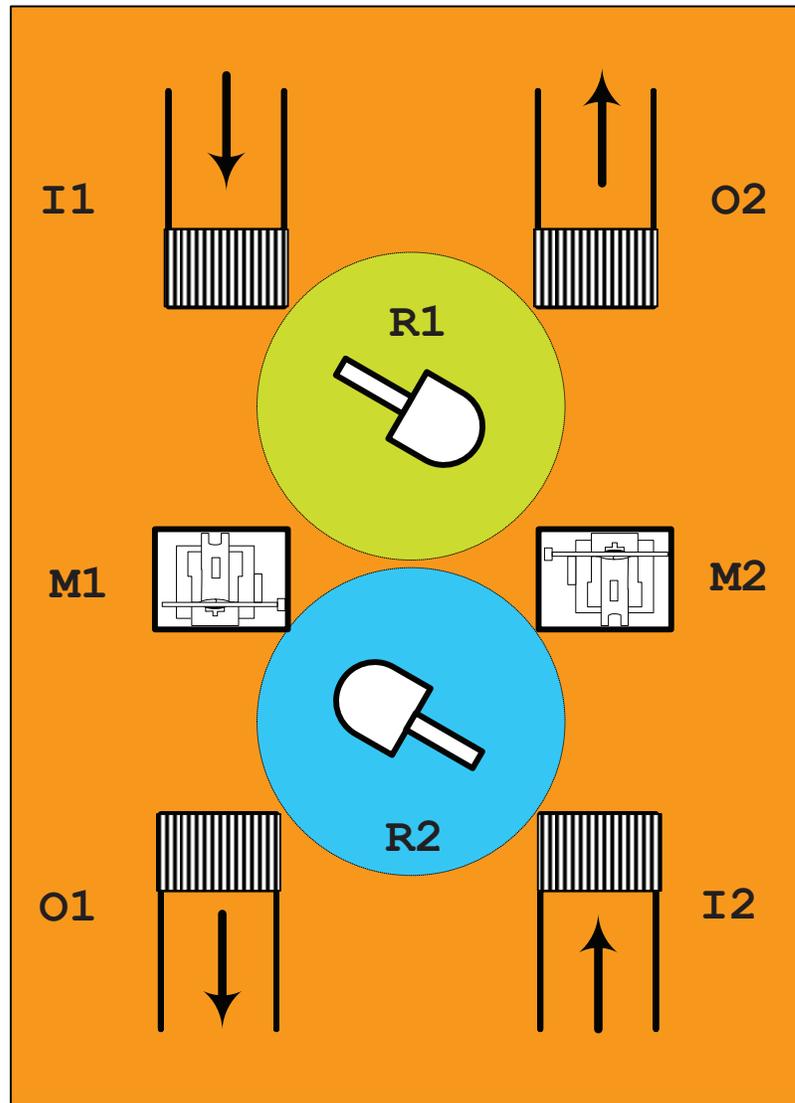


Figura 1.10: Esquema de una celda de producción de un SMF.

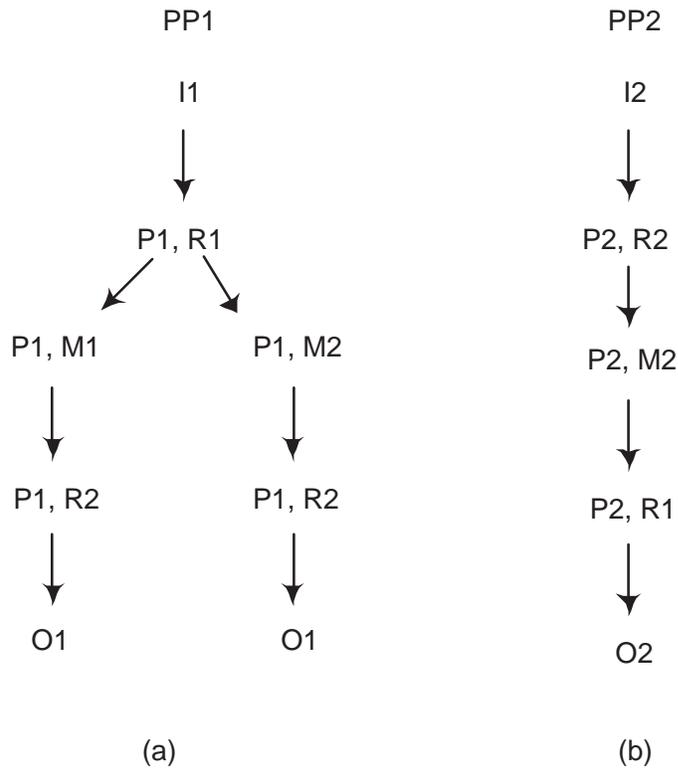


Figura 1.11: Planes de Producción (a) Parte P1 (b) Parte P2.

finalmente son descargadas al terminar su procesamiento en el punto de salida $O2$.

La Red de Petri de la figura 1.12 modela los planes de producción de cada uno de los tipos de partes $P1$ y $P2$ de la celda de la figura 1.10 desde la entrada a la celda de la materia prima hasta que la parte terminada es evacuada de la misma. Cada plan de producción es un máquina de estados fuertemente conexas, \mathcal{N} , en la figura 1.12. Estas máquinas de estados \mathcal{N} comparten los recursos $M2$, $R1$ y $R2$ modelados mediante lugares con estos nombres en la Red de Petri.

En este modelo los lugares alcanzados por una parte están indicados por el nombre de la parte ($P1$ ó $P2$) seguida por el nombre del recurso utilizado. Por ejemplo, $P2M2$ significa que la parte $P2$ está siendo procesada en la máquina $M2$.

Cada parte en uno de estos lugares necesitan algún número de recursos del sistema ($M1$, $M2$, $R1$, $R2$), la disponibilidad de estos recursos es representada por el marcado de cada uno de ellos. El marcado inicial en los recursos $M1$ y $M2$ es igual al número de partes que cada máquina puede procesar concurrentemente. Por otro lado, el marcado inicial en

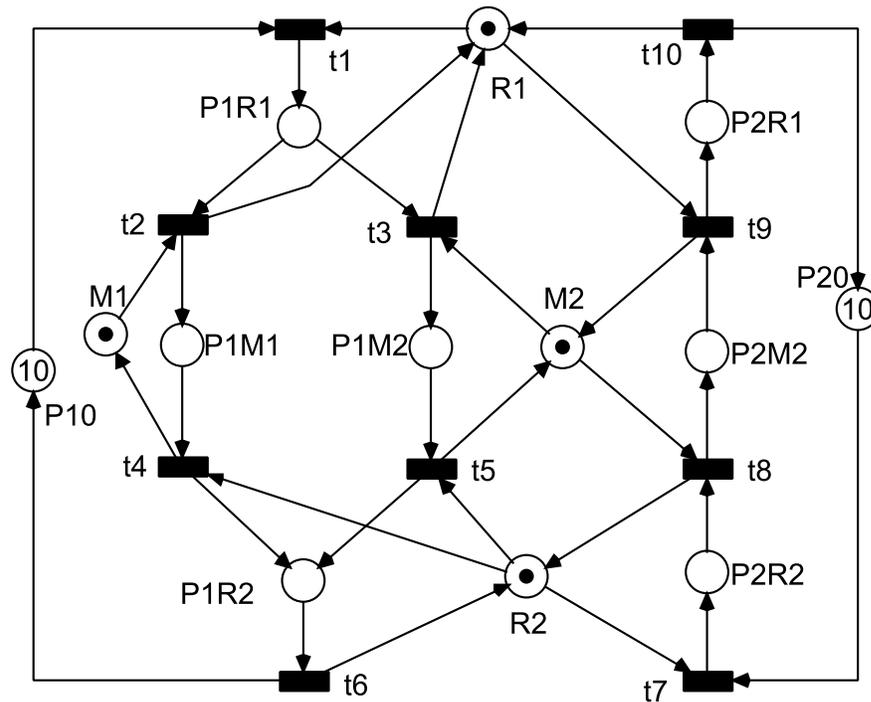


Figura 1.12: La RdP modelando el procesamiento de las partes de la celda de producción.

los recursos $R1$ y $R2$ indica la cantidad de partes que cada robot puede sujetar a la vez.

El número máximo de partes del tipo $P1$ y $P2$ a ser simultáneamente producidas en el sistema está modelado por el marcado inicial de los lugares $P10$ y $P20$, respectivamente. Estos lugares han sido agregados para modelar la naturaleza repetitiva del procesamiento de diferentes partes del mismo tipo.

El modelo de la figura 1.12 pertenece a la clase de redes llamada S^4PR . A continuación se presenta la semántica para cada uno de los elementos de esta clase de red.

1.6. Semántica de la Clase S^4PR en Sistemas de Fabricación Flexible

A continuación se presentará la semántica que se utilizará a lo largo de este trabajo para cada uno de los elementos y propiedades de una red perteneciente a la clase S^4PR [Col03, Tri03].

1. *Un plan de producción está compuesto por rutas de producción.* En la clase S^4PR cada ruta de producción corresponde a un T-semiflujo mínimo. Este pertenece a una posible secuencia de procesamiento de una parte desde su inicio hasta su terminación. Desde el punto de vista de la aplicación, esto equivale al movimiento de una marca a partir del lugar reposo siguiendo una ruta en un plan de producción hasta llegar nuevamente al mismo, en donde se completa el procesamiento de dicha parte.

Por otro lado, un plan de producción es una máquina de estados fuertemente conexa que está cubierto por los T-semiflujos correspondientes al conjunto de rutas de producción que contiene.

2. *Las rutas de producción de los planes de producción son reproducibles.* Cada ciclo en una máquina de estados representa la secuencia de disparos con la cual se completa el procesamiento de una parte a partir del lugar reposo hasta el final (punto anterior). La naturaleza reproducible de esta secuencia es capturada en Red de Petri por el concepto de T-semiflujo.
3. *Cada ruta de producción debe poder ser ejecutada aisladamente.* Esta propiedad se mantiene en la clase S^4PR debido a que cada plan de producción tiene una estructura de máquina de estados fuertemente conexa (definición 1.3) y el marcado inicial del lugar reposo es no-vacío (definición 2.1).
4. *Dado un estado de ejecución intermedio de un plan de producción aislado, siempre es posible para el mismo terminar recuperando su estado inicial.* Esta propiedad se mantiene en la clase S^4PR porque una máquina de estados fuertemente conexa puede mover todos los tokens a cualquier lugar de la red.
5. *Todas las transiciones entre las operaciones de producción definidas en un plan de producción son vivas.* Esta propiedad se mantiene por las mismas razones del punto anterior.
6. *Los planes de producción usan los recursos en forma conservativa.* Esta propiedad está caracterizada en la clase S^4PR por la existencia de un P-Semiflujo mínimo para cada recurso, \mathbf{y}_r tal que $\mathbf{y}_r[r] = 1$, impuesto por la definición de la clase (definición 1.4). Además, estos P-semiflujos mínimos \mathbf{y}_r inducen una relación invariante de marcado, $(\forall \mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0), \mathbf{y}_r \cdot \mathbf{m} = \mathbf{y}_r \cdot \mathbf{m}_0 = \mathbf{m}_0[r])$, que implica la reusabilidad de los recursos. Esto es, los procesos no pueden cambiar los recursos que utilizan. Por lo tanto, los recursos no pueden ser creados ni destruidos.

7. *Todos los estados de un plan de producción usan algún recurso para llevar a cabo sus operaciones.* Esta propiedad es la última de la definición 1. Por lo tanto, los modelos de redes diseñados utilizando el enfoque de los Sistemas de Asignación de Recursos son conservativos.
8. *Un plan de producción con recursos está bien definido.* Esto significa que cuando no hay actividad en el sistema, cada posible ruta de producción tiene los recursos suficientes para ejecutarse aisladamente (definición 2) [TCE99].

Los sistemas diseñados utilizando este enfoque con redes S^4PR permiten modelar decisiones en línea o rutas alternativas y el uso de múltiples recursos en cada paso del procesamiento. Por otro lado, dichos modelos son limitados debido a que operaciones tales como ensamblado/desensamblado desarrollándose en un solo proceso dentro del sistema no pueden ser representados. Esta última condición corresponde a la idea de *evolución del procesamiento en el sistema* (definición 1.2).

1.7. Conclusión y Justificación

En este capítulo se han presentado los modelos para el diseño de Sistemas de Asignación de Recursos. Se han definido los elementos que componen estos sistemas y se han presentado las principales restricciones sobre las cuales se derivan diversos enfoques en la fase del diseño de los Sistemas de Asignación de Recursos que se pueden encontrar en la literatura.

De manera similar, se ha presentado una introducción al modelado de Sistemas de Asignación de Recursos utilizando las redes de Petri enfocándonos en las redes S^4PR las cuales serán objeto de estudio en el desarrollo de esta memoria. Aspectos relacionados al análisis y síntesis para estas redes se han explicado basados en los cerrojos causa estructural de no-vivacidad en las mismas. También, se expone la importancia de tener algoritmos de cálculo de cerrojos más eficientes para las redes S^4PR . Con esta finalidad se han estudiado y desarrollado en esta memoria propiedades, herramientas y algoritmos relacionados con los cerrojos de redes S^4PR por dar soporte a las tareas de análisis y síntesis. Los resultados se han organizado de la siguiente forma: en el capítulo 2 se han estudiado nuevas propiedades estructurales que los cerrojos de las redes S^4PR poseen. De igual forma, se presentan cotas inferior y superior, ambas alcanzables, del número cerrojos mínimos para estas redes.

En el capítulo 3 se han desarrollado nuevas relaciones sobre el conjunto de cerrojos tales como la Relación de Poda, con la cual podemos determinar si existen elementos candidatos (*lugares*) a ser podados en un cerrojo mínimo de un recurso por otro cerrojo

mínimo de un recurso en la operación de unión de ambos cerrojos. Esta relación de poda es representada mediante el grafo de poda y tres funciones de etiquetado de arcos y vértices asociadas al mismo. Estas dos herramientas nos permitirán obtener en el capítulo 4 los cerrojos mínimos del conjunto de cerrojos que una red S^4PR posee.

En el capítulo 4 se calculan los cerrojos de una red S^4PR que contenga un conjunto determinado de recursos, se calculan los cerrojos mínimos para las redes S^4PR , se realiza en análisis del coste computacional del cálculo de cerrojos mínimos utilizando el grafo de poda y se explica el Teorema de vivacidad utilizando el grafo de poda.

Finalmente, en el capítulo 5 se explicarán nuevos algoritmos que nos permiten realizar de manera más eficiente los cálculos de los cerrojos mínimos para subclases de las S^4PR .

Capítulo 2

El Conjunto de Cerrojos Mínimos en Redes S^4PR

Resumen

El conjunto de cerrojos en redes S^4PR constituyen una herramienta muy útil para multitud de tareas de análisis y síntesis. Mediante los cerrojos podemos caracterizar los bloqueos en términos de elementos estructurales sin tener que recurrir a la enumeración de marcados. Sin embargo, el cálculo de estos componentes estructurales puede consumir mucho tiempo o incluso, puede ser imposible en la práctica. Los cerrojos son usados para caracterizar la vivacidad en las redes S^4PR .

Para obtener el conjunto de cerrojos de una red S^4PR , de manera más eficiente, es necesario investigar las propiedades que éstos poseen en esta clase particular de redes. A partir de estas nuevas propiedades construiremos diversas herramientas las cuales nos facilitan el cálculo del conjunto de todos los cerrojos de la red, tanto los mínimos como algunos no-mínimos.

En este capítulo se presentan nuevos resultados que no se han investigado antes en la literatura tales como las propiedades sobre la estructura que poseen los cerrojos en las redes S^4PR , se caracterizan los cerrojos mínimos por el conjunto de recursos que contienen, se obtienen las cotas superior e inferior al número de cerrojos mínimos de una red S^4PR .

2.1. Introducción

Los cerrojos [Com72] son muy importantes porque están asociados a los bloqueos en las Redes de Petri. Fácilmente se puede demostrar que en cualquier marcado muerto, en donde no existen transiciones habilitadas, hay al menos un cerrojo con todas sus transiciones de salida deshabilitadas por los propios lugares del cerrojo. Esta propiedad es fuerte en el caso en los Sistemas de Asignación de Recursos y para ciertas subclases de Redes de Petri la misma se convierte en una herramienta muy potente para la caracterización de la vivacidad. Su estudio lo podemos ubicar dentro de las técnicas de análisis estructural. Estas técnicas nos permiten demostrar una serie de propiedades de las Redes de Petri basadas en la estructura de una red y de una manera casi independiente del marcado inicial que la misma posea. En el apéndice A el lector puede encontrar mayor información relacionada con las notaciones, definiciones y propiedades de las Redes de Petri que se utilizarán en esta tesis.

Obtenemos una mayor ventaja de los cerrojos al utilizarlos en subclases de redes, tal como lo estamos haciendo en este caso para la subclase de red S^4PR y en sus subclases anteriores S^3PR [ECM95] y $L - S^3PR$ [EGVC98] en las cuales se utilizan extensivamente los cerrojos. Otras subclases de redes en las cuales se han utilizado de forma exitosa los cerrojos son: redes de libre elección (“free-choice”) [ES92], redes extendidas de libre elección [BM92], redes de libre elección asimétricas [BPP96], Redes de Petri con recursos (PNR) [JXH00], sistemas de procesos secuenciales simples con requerimientos de recursos generales (S^3PGR^2) [PR01b], entre otros.

Los cerrojos son un objeto estructural definido como un conjunto de lugares no-vacío. Estos lugares están conectados a través de transiciones de tal manera que si no hay marcas dentro de los mismos no es posible obtener marcas procedentes del resto de la red. Esta última condición constituye la propiedad que hace relevante el uso de los cerrojos [Sil85], la cual se puede resumir de la siguiente forma: “un cerrojo inicialmente desmarcado o que se desmarca debido a una secuencia de disparos, permanecerá desmarcado para cualquier evolución posterior posible de la red. Es decir, al menos todas sus transiciones de salida, por lo tanto, las de entrada no son vivas, y, por consiguiente, la red no es viva”. Debido a esta propiedad, los cerrojos son importantes para resolver los problemas de vivacidad en las Redes de Petri. Además, en subclases de redes como las S^4PR los cerrojos nos permiten estudiar la vivacidad de una red sin tener que recurrir a la enumeración de los marcados que la misma posee. A continuación se define formalmente el concepto de cerrojo.

Definición 3. Sea $\mathcal{N} = \langle P, T, C \rangle$ una red. Un subconjunto de lugares $D \subseteq P$ es un cerrojo si, y solo si, $\bullet D \subseteq D^\bullet$.

Los cerrojos mínimos desempeñan un papel muy importante en la teoría de análisis de Redes de Petri [Hac72, CHEP71, Com72, Bes86, DE95]. En un cerrojo mínimo con

al menos dos lugares, el proceso de eliminar un lugar del cerrojo hace que el resto de los lugares no puedan formar un nuevo cerrojo. Esto es, si D (donde $|D| \geq 2$) es un cerrojo mínimo, para cada $p \in D$, existe una transición $t \in \bullet D \setminus \{p\}$, tal que $\bullet t \cap D = \{p\}$. A continuación se definen los cerrojos mínimos.

Definición 4. *Un cerrojo D se dice que es mínimo si, y solo si, este no contiene a otro cerrojo no-vacío como subconjunto propio.*

En el caso de las redes S^4PR la definición de la clase impone estrictas restricciones en la estructura y en particular en los lugares recursos, y por consiguiente los cerrojos en las S^4PR tienen una estructura con propiedades particulares útiles. Estas propiedades estructurales son investigadas por primera vez en la literatura para las redes S^4PR en [CRC10a, CRC10b], en los trabajos previos [TCE99, Tri03] se asumía que los cerrojos mínimos en estas redes no tenían propiedades diferentes a las que poseían los cerrojos de las redes generales.

El resto del capítulo se organiza como sigue. En la sección 2.2 se introduce la caracterización de la vivacidad en redes S^4PR utilizando la definición de la S^4PR y del marcado inicial aceptable para estas redes dadas en el capítulo 1. En la sección 2.3 se estudian nuevas propiedades sobre la estructura y composición de los cerrojos para las redes S^4PR . Por medio de estas propiedades se logra caracterizar la forma que tienen los cerrojos en estas redes. Esto es muy importante para desarrollar en el capítulo 3 nuevas herramientas las cuales nos permitirán calcular el conjunto de cerrojos mínimos que una red posee al igual que obtener el cerrojo para un conjunto de recursos dados, como se verá en el capítulo 4. Merced a las propiedades de la sección 2.3, en la sección 2.4 se presentan las cotas alcanzables del número de cerrojos mínimos en esta clase de redes. Estos valores se presentan por primera vez en la literatura y hacen que el número de cerrojos mínimos en estas redes sea un número bastante más reducido que lo que se presentaba con anteriormente al considerar que los cerrojos en S^4PR no presentaban propiedades especiales con respecto a las de las redes generales. Finalmente, la sección 2.5 presenta algunas conclusiones del capítulo.

2.2. Caracterización de la Vivacidad en Redes S^4PR

En sistemas concurrentes una propiedad importante a garantizar es la *vivacidad*. A través de la vivacidad se garantiza que desde cualquier estado alcanzable todas las actividades pueden llegar a realizarse. Esta propiedad abstracta es de gran importancia y significación en el contexto de los Sistemas de Fabricación Flexible. Para observar esto, veremos a continuación algunas consecuencias derivadas de la ausencia de vivacidad en el

Sistema de Fabricación Flexible y por ende en el modelo S^4PR que lo abstrae. Si un sistema es no vivo, esto significa que a partir de un estado determinado existe al menos una acción o actividad que ya nunca podrá volver a realizarse. Esto puede tener varias consecuencias. La primera es que existe un proceso pendiente de finalizar su procesamiento y no puede acabar debido a que la acción a emprender no puede iniciarse debido a que no existe el número de recursos necesarios para ello, y además en el futuro no aparecerán más recursos para poder realizar esa actividad. Esta situación descrita es lo que denominaremos *bloqueo*: un proceso queda parado indefinidamente sin terminar por falta de recursos. Una segunda consecuencia puede ser que una acción nunca puede ser realizada pero no se llega a bloquear ningún proceso porque éstos tienen rutas alternativas para terminar. En este caso, dicha actividad puede ser eliminada del sistema ya que no se ejecuta a partir de un estado determinado. Veremos que esto no es posible en sistemas modelados con S^4PR .

El siguiente teorema presenta la caracterización de la vivacidad para la clase de redes S^4PR la cual es completamente comportamental. La siguiente notación es necesaria para entender el enunciado del teorema. Dado un marcado \mathbf{m} en una red S^4PR , una transición t está habilitada por proceso en \mathbf{m} (está deshabilitada por proceso en \mathbf{m}) si, y solo si, está (no-está) habilitada por sus lugares proceso de entrada, y está habilitada por recurso en \mathbf{m} (está deshabilitada por recurso en \mathbf{m}) si, y solo si, todos (alguno de) sus lugares recurso de entrada tienen (no-tiene) suficientes marcas para su disparo.

Adicionalmente, utilizaremos las siguientes definiciones. Dado un recurso r y basado en los p -semiflujos mínimos \mathbf{Y}_r el *conjunto de usuarios*¹ del recurso r se definen como el conjunto de los lugares proceso que pueden utilizar dicho recurso.

Definición 5. Sea $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red S^4PR . Sea $r \in P_R$. El conjunto de usuarios de r es el soporte del p -semiflujo mínimo \mathbf{Y}_r sin el lugar r : $\mathcal{H}_r = ||\mathbf{Y}_r|| \setminus \{r\}$. Esta definición puede extenderse de manera natural a los conjuntos de recursos $A \subseteq P_R$: $\mathcal{H}_A = \bigcup_{r \in A} \mathcal{H}_r$.

Teorema 1 ([Tri03]).

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red S^4PR marcada. El sistema es no-vivo si, y solo si, existe un marcado $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ tal que el conjunto de transiciones habilitadas por proceso en \mathbf{m} es no-vacío y cada una de esas transiciones está deshabilitada por recurso en \mathbf{m} .

El Teorema 1 relaciona la no-vivacidad con la existencia de un marcado en donde los procesos activos están bloqueados. Las transiciones de salida de estos procesos necesitan unos recursos que no se encuentran disponibles y que además en el futuro tampoco se

¹Holders, \mathcal{H}_r , en terminología inglesa.

encontrarán disponibles. Esto sugiere que existe una espera circular entre un conjunto de procesos bloqueados.

Este concepto de espera circular es capturado en el modelo por la existencia de un cerrojo² (causa estructural para la no vivacidad en términos de Redes de Petri) cuyos lugares recurso son los lugares que impiden el disparo de las transiciones de salida de los lugares proceso que están marcadas. El siguiente teorema caracteriza la no-vivacidad en términos de los cerrojos estableciendo un puente entre la estructura del modelo en Redes de Petri y su comportamiento.

Teorema 2 ([Tri03]).

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red S^4PR marcada. La red es no-viva si, y solo si, existe un marcado $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ y un cerrojo D tal que $\mathbf{m}[P_S] > 0$ y el disparo de cada transición habilitada por proceso en \mathbf{m} está impedido por un conjunto de lugares recurso pertenecientes a D . Además, el cerrojo D es tal que:

1. $D_R = D \cap P_R = \{r \in P_R \mid \exists t \in r^\bullet \text{ tal que } \mathbf{m}[r] < \text{Pre}[r, t] \text{ y } \mathbf{m}[\bullet t \cap P_S] > 0\} \neq \emptyset$;
2. $D_S = D \cap P_S = \{p \in \mathcal{H}_{D_R} \mid \mathbf{m}[p] = 0\} \neq \emptyset$;

Un cerrojo D y un marcado $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ que verifican las propiedades del Teorema 2 se dirá que son un *cerrojo malo* y un *marcado D -bloqueado*, respectivamente.

Definición 6 ([Tri03]).

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red S^4PR marcada. Sea D un cerrojo de \mathcal{N} . Entonces, $\mathcal{T}h_D = \mathcal{H}_{D_R} \setminus D_S$ es el conjunto de ladrones de D ³.

La anterior definición representa lugares de la red que usan recursos de los cerrojos y que no pertenecen a ellos.

La siguiente caracterización de la vivacidad establece que cuando una S^4PR no es viva, existe un *marcado D -bloqueado* tal que todos sus procesos activos están *robando* marcas del conjunto de recursos de un cerrojo asociado D llamado *cerrojo malo*.

Teorema 3 ([Tri03]).

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$, una red S^4PR marcada. La red es no-viva si, y solo si, existe un cerrojo D y un marcado $\mathbf{m}_D \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ tal que:

1. $\mathbf{m}_D[P_S] > 0$.

²Siphon en terminología inglesa.

³Algunas veces se usa $\mathcal{T}h_{D_R}$ para mostrar la relación entre estos dos conjuntos.

2. $\mathbf{m}_D[P_S \setminus \mathcal{T}h_D] = 0$.
3. $\forall p \in \mathcal{T}h_{D_R}$ tal que $\mathbf{m}_D[p] > 0$, el disparo de cada $t \in p^\bullet$ es impedido por un conjunto de lugares recurso pertenecientes a D .

La vivacidad en esta clase de redes está relacionada con la existencia de cerrojos malos. En esta situación las transiciones de salida de los lugares en el cerrojo están muertas. Por ejemplo, la red de la figura 1.12 tiene dos cerrojos: $D = \{R1, R2, M1, M2, P1R2, P2R1\}$ y $D' = \{R2, M2, P2M2, P1R2\}$ los cuales se desmarcan con el marcado $m = (P1R1, P1M1, P1M2, P2R2, 7 \cdot P10, 9 \cdot P20)$ y $m' = (P1M2, P2R2, 9 \cdot P20, 9 \cdot P10, M1, R1)$, respectivamente.

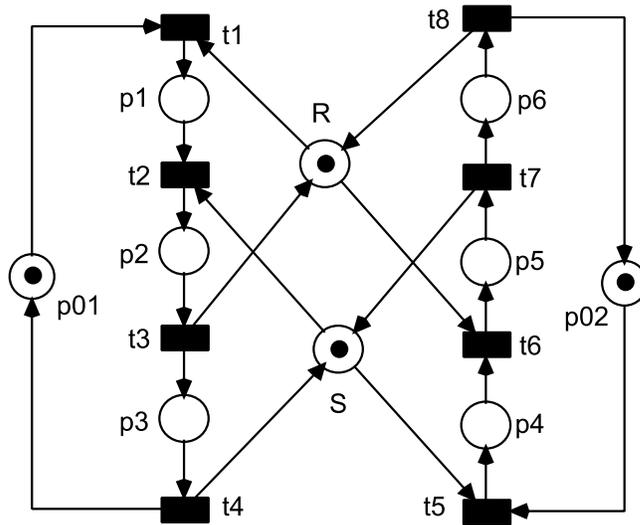
2.3. Propiedades de los Cerrojos Mínimos en redes S^4PR

En esta sección se estudiarán las propiedades que poseen los cerrojos en la clase de redes S^4PR . Estas propiedades las podemos derivar de la definición de dicha clase y las mismas establecen ciertas restricciones que nos facilitarán el cálculo de la familia de sus cerrojos mínimos, tema del cual nos ocuparemos en el capítulo 4.

La primera propiedad está relacionada con los lugares que componen un cerrojo clasificadas según la partición realizada de los lugares en la definición de la clase S^4PR (definición 1.1). El primer lema indica que un cerrojo conteniendo lugares recurso debe contener necesariamente lugares proceso.

Lema 1. Sea \mathcal{N} una red S^4PR y $D \subseteq P$ un cerrojo de \mathcal{N} . $D \cap P_R \neq \emptyset \implies D \cap P_S \neq \emptyset$.

Demostración. Probaremos el resultado por contradicción. Supongamos que $D \neq \emptyset$ es un cerrojo de \mathcal{N} tal que $D \cap P_S = \emptyset$. Sea \mathcal{C} un circuito dirigido en un máquina de estados k de \mathcal{N} tal que $\mathcal{C} = t_0 t_1 \cdots t_u$, $t_0 \in (p_{0_k})^\bullet$, $k \in I_{\mathcal{N}}$, $t_j \in (t_{j-1}^\bullet)^\bullet$ para todo $j \in \{1, \dots, u-1\}$, $t_0 = t_u$, y $\tau^{\mathcal{C}} = \{t_0, t_1, \dots, t_u\}$. Sea $\tau_D^{\mathcal{C}}$ el conjunto de transiciones de entrada de los lugares en D pertenecientes al circuito dirigido \mathcal{C} de \mathcal{N} , esto es, $\tau_D^{\mathcal{C}} = \bullet D \cap \tau^{\mathcal{C}}$. Al menos uno de los conjuntos $\tau_D^{\mathcal{C}}$, para algún circuito dirigido de alguna máquina de estados no está vacío porque el cerrojo no está vacío. Usando el ordenamiento impuesto en las transiciones por el circuito dirigido \mathcal{C} podemos ordenar las transiciones del conjunto $\tau_D^{\mathcal{C}}$. Sea $t \in \tau_D^{\mathcal{C}}$ la última transición de acuerdo a este ordenamiento y $r \in D$ el recurso tal que $t \in \bullet r$. Puesto que D es un cerrojo conteniendo solamente lugares recurso, existe otro lugar recurso $r' \in D$ tal que $t \in (r')^\bullet$. Además, no hay transición, t' , después de t en el circuito \mathcal{C} , tal que $t' \in \bullet r'$ porque t es la última transición del conjunto $\tau_D^{\mathcal{C}}$ en el circuito. Por lo tanto, si construimos el p-semiflujo mínimo correspondiente al lugar recurso r' , para anular las columnas en la matriz de incidencia correspondientes a las transiciones en el circuito \mathcal{C} , desde t hasta la


 Figura 2.1: Un ejemplo de una red S^4PR .

transición t_u , necesitamos los lugares proceso de salida de estas transiciones, incluyendo el lugar reposo. Pero esto contradice la hipótesis que \mathcal{N} es una red S^4PR , donde cada recurso pertenece a un p-semiflujo mínimo que no incluye al lugar reposo. \square

En la figura 2.1 podemos observar que los dos cerrojos $D_R = \{R, p1, p2, p5, p6\}$ y $D_S = \{S, p2, p3, p4, p5\}$ contienen a los lugares recurso R y S al igual que un conjunto no-vacío de lugares proceso como indica el lema anterior.

El recíproco de este lema no es cierto, porque cada subred \mathcal{N}_i (definición de S^4PR) generada por $P_{S_i} \cup \{p_{0_i}\} \cup T_i$ es una máquina de estados fuertemente conexa y por lo tanto es un cerrojo mínimo que no contiene lugares recurso. Por ejemplo la red de la figura 2.1 posee dos cerrojos mínimos $D = \{p01, p1, p2, p3\}$ y $D' = \{p02, p4, p5, p6\}$ que corresponden a cada una de las máquinas de estado fuertemente conexas.

La siguiente propiedad de los cerrojos en la clase S^4PR la obtenemos a partir del lema anterior y de la propia definición de cerrojo mínimo (definición 4) y establece que no existen dos cerrojos mínimos distintos con el mismo conjunto de lugares recurso.

Lema 2. Sea \mathcal{N} una red S^4PR y $D \subseteq P$ un cerrojo mínimo no-vacío de \mathcal{N} conteniendo al menos un lugar recurso. D es el único cerrojo mínimo de \mathcal{N} conteniendo exactamente el conjunto de recursos $D_R = D \cap P_R$.

Demostración. Probaremos este resultado por contradicción. Supongamos que existen dos cerrojos mínimos D y D' tal que $D \neq D'$ y $D_R = D \cap P_R = D' \cap P_R \neq \emptyset$. Sea τ_0 el conjunto

de las transiciones de entrada a los lugares recurso pertenecientes a D_R tal que ellas no tienen lugares de entrada pertenecientes a D_R , i.e. $\tau_0 = \{t \in T \mid t \in \bullet D_R, \bullet t \cap D_R = \emptyset\}$. El conjunto τ_0 debe ser no-vacío, puesto que $\tau_0 = \emptyset$ implica que D_R debe ser un cerrojo y esto no es posible de acuerdo con el Lema 1. Teniendo en cuenta que D y D' son cerrojos mínimos de \mathcal{N} , para todo $t \in \tau_0$ existe $p \in D \cap P_S$ y $p' \in D' \cap P_S$ tal que $p \in \bullet t$ y $p' \in \bullet t$. Pero p y p' son lugares proceso pertenecientes a la misma máquina de estados conteniendo la transición t , por lo tanto $p = p'$. Sea D_{S_0} el conjunto de lugares proceso pertenecientes a ambos cerrojos que son lugares de entrada a las transiciones del conjunto τ_0 . Además, si $p \in D_{S_0}$, existe un lugar recurso $r \in D_R$ tal que $p \in \|\mathbf{y}_r\|$, i.e. p pertenece al conjunto de usuarios de algún $r \in D_R$. Sea τ_1 el conjunto de transiciones de entrada a los lugares pertenecientes a $D_1 = D_R \cup D_{S_0}$ tal que ellos no tienen lugares de entrada pertenecientes a D_1 , i.e. $\tau_1 = \{t \in T \mid t \in \bullet D_1, \bullet t \cap D_1 = \emptyset\}$. Entonces, se distinguen dos casos

$\tau_1 = \emptyset$. En este caso hemos probado que para todo $t \in \bullet D_1 \implies t \in D_1^\bullet$, es decir, D_1 es un cerrojo. Además hemos probado que $D_1 \subseteq D$ y $D_1 \subseteq D'$, lo cual contradice la minimalidad de D y D' . Por lo tanto, $D = D' = D_1$.

$\tau_1 \neq \emptyset$. En este caso $t \in \tau_1 \implies t \in \bullet D_{S_0}$, y por las mismas razones antes indicadas para las transiciones en τ_0 , existe un lugar proceso q perteneciente a ambos cerrojos D y D' tal que $q \in \bullet t$. Además, si el lugar $p \in \bullet t \cap D_{S_0}$ pertenece al soporte del p-semiflujo \mathbf{y}_r , para algún $r \in D_R$, entonces $q \in \|\mathbf{y}_r\|$. Por lo tanto, obtendríamos los conjuntos D_{S_1} y $D_2 = D_R \cup D_{S_0} \cup D_{S_1}$ como en el caso del conjunto τ_0 . Podemos iterar este procedimiento un número finito de veces, k , alcanzando un conjunto $\tau_k = \emptyset$. En efecto, desde la iteración j hasta la iteración $j+1$ haciendo una propagación hacia atrás a partir de cada lugar $p \in D_{S_j}$ hasta un lugar $q \in D_{S_{j+1}}$ si, y solo si, $\tau_j \cap \bullet p \neq \emptyset$. Ambos lugares p y q pertenecen al soporte de un p-semiflujo \mathbf{y}_r de un recurso $r \in D_R$. Por lo tanto, esta propagación hacia atrás desde cada lugar $p \in D_{S_0}$ siempre termina en un lugar $q \in (r^\bullet)^\bullet$ en la mayoría de las k iteraciones, donde k es la longitud máxima de una secuencia de lugares proceso de \mathbf{y}_r , $r \in D_R$, en la misma máquina de estados. En este caso hemos probado que para todo $t \in \bullet D_k \implies t \in D_k^\bullet$, es decir, D_k es un cerrojo. Además, hemos probado que $D_k \subseteq D$ y $D_k \subseteq D'$, contradiciendo la minimalidad de D y D' . Por lo tanto, $D = D' = D_k$.

□

A partir del lema anterior, dada una red S^4PR , \mathcal{N} , sus conjuntos de cerrojos mínimos, pueden ser particionados en $|P_R| + 1$ clases, cada una caracterizada por el número de recursos pertenecientes a cada cerrojo en la clase. Indicaremos la partición del conjunto \mathcal{D}

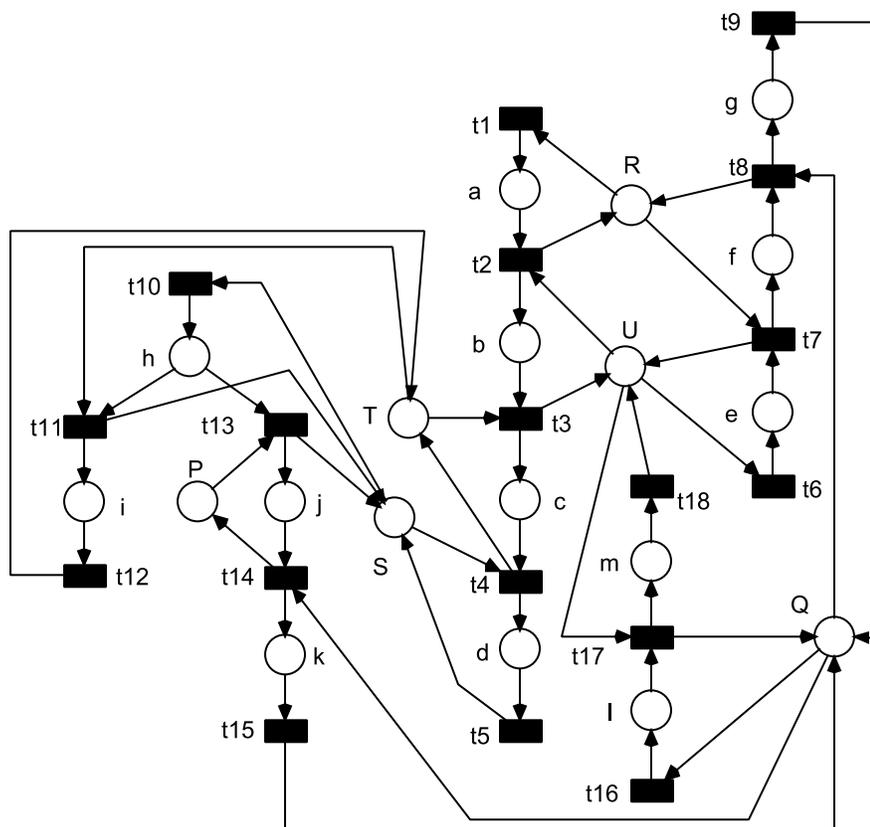


Figura 2.2: Una red S^4PR que no contiene cerrojos mínimos en la clase \mathcal{D}^4 .

de los cerrojos mínimos de \mathcal{N} en las clases apuntadas como,

$$\mathcal{D} = \bigcup_{i=0}^{|P_R|} \mathcal{D}^i$$

donde \mathcal{D}^i es el subconjunto de cerrojos mínimos tal que cada uno contiene exactamente i lugares recurso; y para todo $i, j \in \{0, 1, \dots, |P_R|\}$ con $i \neq j$, $\mathcal{D}^i \cap \mathcal{D}^j = \emptyset$.

La partición de los cerrojos mínimos en el conjunto de clases \mathcal{D}^i no significa que debamos tener por lo menos un cerrojo en cada clase. Por ejemplo, la red de la figura 2.2 tiene seis cerrojos mínimos de un recurso: $D_P = \{P, j\}$, $D_Q = \{Q, k, g, l\}$, $D_R = \{R, a, f\}$, $D_S = \{S, d, h\}$, $D_T = \{T, c, i\}$ y $D_U = \{U, b, e, m\}$ pero tenemos solamente cuatro cerrojos mínimos conteniendo más de un recurso. Además, en la clase \mathcal{D}^1 tenemos los seis cerrojos mínimos de un recurso antes apuntados y en las clases \mathcal{D}^2 , \mathcal{D}^3 , \mathcal{D}^5 y \mathcal{D}^6 tenemos solamente un cerrojo por clase. Estos son: $\mathcal{D}^2 = \{\{R, U, b, f, m\}\}$, $\mathcal{D}^3 = \{\{Q, R, U, b, g, k, m\}\}$, $\mathcal{D}^5 = \{\{P, Q, S, T, U, d, e, g, i, k, m\}\}$ y $\mathcal{D}^6 = \{\{P, Q, R, S, T, U, d, g, i, k, m\}\}$. Esta red será utilizada en el capítulo 3 de esta memoria en donde se mostrará su grafo de poda y posteriormente en el capítulo 4 se explica cómo se obtienen los cerrojos mínimos de la clase \mathcal{D}^i utilizando el grafo de poda y el algoritmo desarrollado en dicho capítulo para tal fin. El ejemplo anterior nos sirve para ilustrar que cualquiera de las posibles clases de cerrojos mínimos que contienen el mismo número de recursos pueden estar vacías, como en este caso sucede con \mathcal{D}^4 . Esta es la mayor dificultad para construir algoritmos eficientes para el cálculo de los cerrojos mínimos que proceden de una manera incremental a partir de la clase \mathcal{D}^i hasta la clase \mathcal{D}^{i+1} utilizando solamente los cerrojos mínimos de la clase \mathcal{D}^1 y los de la clase \mathcal{D}^i .

A partir de la definición de las redes S^4PR la clase de los cerrojos mínimos \mathcal{D}^0 está completamente caracterizada y no es necesario su cálculo. En efecto, los únicos cerrojos mínimos conteniendo cero recursos son cada uno de los conjuntos de lugares de cada una de las máquinas de estados fuertemente conexas definidas en la red. Por lo tanto, de acuerdo a la definición de las S^4PR el conjunto \mathcal{D}^0 de \mathcal{N} es,

$$\mathcal{D}^0 = \{P_{S_i} \cup \{p_{0_i}\} \mid i \in I_{\mathcal{N}} = \{1, 2, \dots, m\}\}$$

Para la figura 2.1 la clase \mathcal{D}^0 está formada por los cerrojos mínimos $D = \{p_{01}, p_1, p_2, p_3\}$ y $D' = \{p_{02}, p_4, p_5, p_6\}$.

Por otro lado, la clase \mathcal{D}^1 de los cerrojos mínimos conteniendo solamente un recurso puede ser directamente obtenida a partir del conjunto de los p-semiflujos mínimos asociados a los recursos. En el caso de la figura 2.1 los dos cerrojos $D_R = \{R, p_1, p_2, p_5, p_6\}$ y $D_S = \{S, p_2, p_3, p_4, p_5\}$ corresponden a los cerrojos mínimos conteniendo los recursos R y S , respectivamente, y por tanto la clase \mathcal{D}^1 está formada

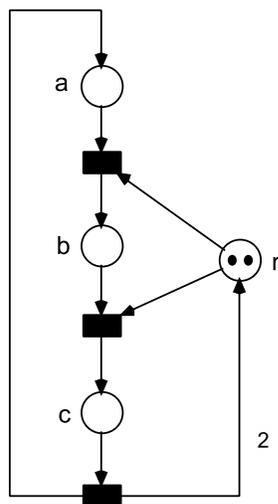


Figura 2.3: El cerrojo mínimo $\{c, r\}$ está estrictamente contenido en el soporte del p-semiflujo mínimo \mathbf{y}_r .

por ambos cerrojos. En otras palabras, a partir del lema 2 solo pueden existir como mucho $|P_R|$ cerrojos mínimos que contengan un único recurso. Dado que en la definición de la clase S^4PR existe un p-semiflujo mínimo asociado a un recurso, y el soporte del p-semiflujo es un cerrojo, a partir de los p-semiflujos podremos definir los cerrojos de esta clase.

El siguiente lema formaliza esta propiedad de los cerrojos mínimos conteniendo un recurso, i.e pertenecientes a la clase \mathcal{D}^1 .

Lema 3. *Sea \mathcal{N} una red S^4PR . Para cada $r \in P_R$ existe un cerrojo mínimo, $D_r \in \mathcal{D}^1$, tal que $D_r \subseteq \|\mathbf{y}_r\|$.*

Demostración. Sea $r \in P_R$ un lugar recurso, trivialmente, el soporte del p-semiflujo mínimo asociado a r , $\mathbf{y}_r \in \mathbb{N}^{|P|}$, es un cerrojo de la red. Esto es porque \mathbf{y}_r es un p-semiflujo y con el propósito de anular cada columna de la matriz de incidencia correspondiente a una transición, t , en el conjunto $\bullet(\|\mathbf{y}_r\|) \cup (\|\mathbf{y}_r\|)^\bullet$ nosotros necesitamos al menos un lugar de entrada y un lugar de salida de t dado que la red es fuertemente conexa. Esto significa que $\bullet(\|\mathbf{y}_r\|) = (\|\mathbf{y}_r\|)^\bullet$, i.e. $\|\mathbf{y}_r\|$ es un cerrojo. No obstante, en general, $\|\mathbf{y}_r\|$ no es un cerrojo mínimo, pero contiene un cerrojo mínimo D_r . Probaremos por contradicción que $r \in D_r$. Supongamos que $r \notin D_r$, puesto que r es el único recurso en $\|\mathbf{y}_r\|$ y $\|\mathbf{y}_r\| \cap P_0 = \emptyset$, la subred generada por D_r es un conjunto de rutas dirigidas en las máquinas de estados fuertemente conexas de \mathcal{N} , tal que para cada ruta existe al menos una transición sin lugares de entrada

y entonces D_r no puede ser un cerrojo, en contradicción con la hipótesis. \square

En la figura 2.3 se puede observar que el soporte del p-semiflujo mínimo de r , $\|\mathbf{y}_r\| = \{r, b, c\}$, es un cerrojo pero este no es mínimo debido a que contiene al cerrojo mínimo $\{r, c\}$. Cada cerrojo mínimo de \mathcal{D}^1 está contenido en el soporte del correspondiente p-semiflujo mínimo y dicho cerrojo mínimo es único (lemma 2). Es decir, la cardinalidad de \mathcal{D}^1 es igual a $|P_R|$ y cada cerrojo mínimo es obtenido a partir del p-semiflujo, posiblemente, eliminando algunos lugares para hacerlo mínimo.

El siguiente lema presenta un resultado técnico el cual dice que un cerrojo mínimo conteniendo un conjunto de recursos $D_R \neq \emptyset$, está siempre contenido en la unión de los cerrojos mínimos $D_r \in \mathcal{D}^1, r \in D_R$.

Lema 4. Sea \mathcal{N} una red S^4PR , $D \subseteq P$ un cerrojo mínimo de \mathcal{N} y $D_R = D \cap P_R \neq \emptyset$. $D \subseteq \bigcup_{r \in D_R} D_r, D_r \in \mathcal{D}^1$

Demostración. $D' = \bigcup_{r \in D_R} D_r, D_r \in \mathcal{D}^1$, es un cerrojo porque cada D_r es también un cerrojo. Supongamos que $D \not\subseteq D'$. Esto significa que existe, al menos, un lugar proceso $p \in D \cap P_s$ tal que $p \notin D'$. Además, $p \notin D_r, \forall r \in D_R$, porque $D' = \bigcup_{r \in D_R} D_r$. D es un cerrojo mínimo, por lo tanto el lugar p es esencial para el cerrojo D , i.e. existe $t \in p^\bullet$ tal que ${}^\bullet t \cap D = \{p\}$ y $t^\bullet \cap D \neq \emptyset$. Sea q un lugar tal que $q \in t^\bullet \cap D$. Si q es un lugar recurso, entonces el lugar p es esencial para el cerrojo mínimo D_q , pero esto contradice los resultados anteriores diciendo que $p \notin D_r, \forall r \in D_R$. Si q es un lugar proceso, por la minimalidad de D , q es esencial para D y podemos aplicar nuevamente el razonamiento anterior en p pero para q . Iterando el esquema hemos construido una ruta $ptq\dots$ hasta alcanzar un lugar recurso $x \in D$, y en una forma similar al caso donde q es un lugar recurso podemos concluir que p es esencial para el cerrojo D_x , contradiciendo los resultados previous diciendo que $p \notin D_r, \forall r \in D_R$. Por lo tanto, podemos concluir que $D \subseteq \bigcup_{r \in D_R} D_r, D_r \in \mathcal{D}^1$. \square

A partir de los dos resultados anteriores, el siguiente lema puede ser trivialmente obtenido.

Lema 5. Sea \mathcal{N} una red S^4PR , $D \subseteq P$ un cerrojo mínimo de \mathcal{N} y $D_R = D \cap P_R \neq \emptyset$. $D \subseteq \bigcup_{r \in D_R} \|\mathbf{y}_r\|$

El siguiente Lema dice que no es posible encontrar lugares reposo pertenecientes a un cerrojo mínimo con recursos.

Lema 6. Sea \mathcal{N} una red S^4PR y $D \subseteq P$ un cerrojo mínimo no-vacio de \mathcal{N} conteniendo al menos un lugar recurso. $D \cap P_0 = \emptyset$.

Demostración. De acuerdo al Lema 5, $D \subseteq \bigcup_{r \in D_r} \|\mathbf{y}_r\|$, donde $D_R = D \cap P_R$. Por la definición de las redes S^4PR , $P_0 \cap \|\mathbf{y}_r\| = \emptyset$, para todo $r \in P_R$. Por lo tanto, $D \cap P_0 \subseteq (\bigcup_{r \in D_r} \|\mathbf{y}_r\|) \cap P_0 = \emptyset$. \square

Este último resultado nos permite eliminar los lugares reposo de la red antes de iniciar el cálculo de los cerrojos mínimos, reduciendo el tamaño de la entrada.

2.4. Número de Cerrojos Mínimos en Redes S^4PR y Subclases

En esta sección se investigan el número de cerrojos mínimos que puede tener una red S^4PR , estas cotas son presentadas por primera vez en la literatura en [CRC10a, CRC10b]. En trabajos previos realizados para las redes S^4PR [TCE99, Tri03] no se ha presentado el número de cerrojos mínimos que éstas poseen, de manera similar ocurre para las redes S^3PR [ECM95, LZ04, Cha06, LZ08, XZW⁺11]. En todos estos trabajos previos se asume que las redes pueden tener un número de cerrojos mínimos igual a $2^{|P|}$, donde P es el conjunto de todos los lugares proceso de la red. En [WLJZ09] se presentan para las redes $L - S^3PR$ la cota superior para los cerrojos mínimos estrictos y para los cerrojos elementales sin llegar a acotar el número de cerrojos mínimos para estas redes. En esta sección se presentan las cotas superior e inferior para los cerrojos mínimos de las redes $L - S^3PR$ y se demuestra que las mismas no pueden ser mejoradas.

En la sección anterior, gracias al lema 2, hemos visto que no pueden existir dos cerrojos mínimos distintos que contengan el mismo subconjunto de recursos. Esto nos ha llevado a una clasificación de los cerrojos mínimos según el número de recursos que contienen. Para la clase \mathcal{D}^0 y \mathcal{D}^1 con ayuda del lema 2, hemos establecido que el número de cerrojos de cada clase es igual a $|I_{\mathcal{N}}|$ y $|P_R|$ respectivamente por la definición de la clase S^4PR .

Para las clases restantes, \mathcal{D}^i e $i > 1$, lo único que podemos decir a priori es que $|\mathcal{D}^i| \leq \binom{|P_R|}{i}$, esto es, el número de cerrojos mínimos conteniendo i recursos está definido por el número de subconjuntos de i lugares del conjunto P_R .

A partir de los argumentos discutidos en la sección anterior se obtiene el siguiente lema el cual es trivial y especifica las cotas superior e inferior del número de cerrojos mínimos de una red S^4PR .

Lema 7. *Sea \mathcal{N} una red S^4PR . El número de cerrojos mínimos de \mathcal{N} , n_D , satisface, $n_I + n_R \leq n_D \leq n_I + 2^{n_R} - 1$, donde n_I es el número de máquinas de estados fuertemente conexas de \mathcal{N} , y $n_R = |P_R|$. Además, estas cotas no se pueden mejorar.*

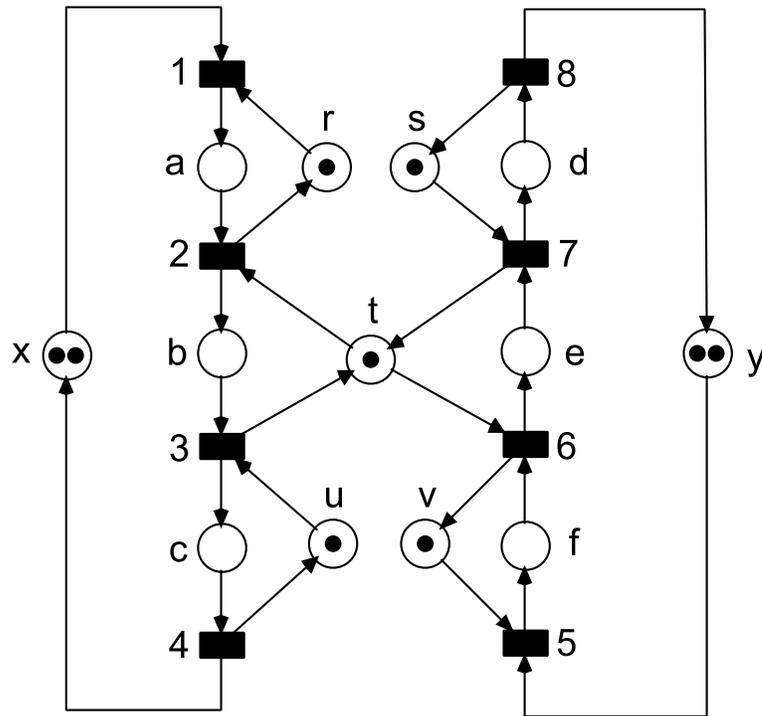


Figura 2.4: Una red S^4PR con un número de cerrojos mínimos igual a la cota inferior del lema 7.

Ambas cotas son alcanzables, por ejemplo, en la red de la figura 2.3 el número de cerrojos mínimos es igual a la cota inferior, esto es, $S_1 = \{r, c\}$ y el cerrojo formado por los lugares de la máquina de estados. Otro ejemplo lo encontramos en la red de la figura 2.4 en donde podemos verificar que el número de cerrojos mínimos que la misma posee es igual a la cota inferior del número de cerrojos mínimos de la red, estos son $S_1 = \{x, a, b, c\}$, $S_2 = \{y, d, e, f\}$, $S_3 = \{r, a\}$, $S_4 = \{s, d\}$, $S_5 = \{t, b, e\}$, $S_6 = \{u, c\}$ y $S_7 = \{v, f\}$.

Por otro lado, la red de la figura 2.5 posee un número de cerrojos mínimos igual a la cota superior del número de cerrojos mínimos especificada en el lema, los cuales son $S_1 = \{x, a, b, c\}$, $S_2 = \{y, d, e, f, \}$, $S_3 = \{r, a, b, d\}$, $S_4 = \{s, b, e\}$, $S_5 = \{t, c, e, f\}$, $S_6 = \{r, s, b, d\}$, $S_7 = \{r, t, a, c, d, f\}$, $S_8 = \{s, t, c, e\}$ y $S_9 = \{r, s, t, c, d\}$.

Los resultados presentados hasta el momento referidos a los cerrojos se refieren a la clase completa S^4PR . En el resto de la sección revisamos estas propiedades mirando algunas subclases de la clase S^4PR con son $L-S^3PR$ y S^3PR .

En primer lugar se aprecia que cuando a la clase de cerrojos \mathcal{D}^0 \mathcal{D}^1 , el número de

elementos que pertenece a cada uno de ellos no se ve alterado porque la red sea S^4PR o una de sus subclases. Ello es debido a que todas las subclases referidas en la literatura de la clase S^4PR comparten en su definición las máquinas de estado fuertemente conexas que dan origen a los cerrojos mínimos sin recurso de la clase \mathcal{D}^0 , y la existencia de un único p-semiflujo mínimo asociado a cada recurso de la red. Por lo tanto, de esta última característica y dado que el soporte de un p-semiflujo siempre define un cerrojo, por el lema 2 y el lema 3, concluimos que para todas las subclases conocidas en la literatura de la clase S^4PR se cumple que el cardinal de la clase \mathcal{D}^1 es igual al número de recursos de la red.

No obstante, para la subclases $L - S^3PR$, S^3PR ó $SOAR^2$ [Rov11] se puede formular una revisión más estricta del lema 3. Ello es debido a la particular forma que tienen los p-semiflujos mínimos de los recursos y las subredes generadas por $\|\mathbf{y}_r\| \cup (\|\mathbf{y}_r\|)^\bullet$, para todo $r \in P_R$.

Esta versión más estricta del lema 3 establece que los cerrojos que contienen un único recurso son precisamente los soportes de los p-semiflujos de recursos. Para demostrar esta propiedad, primero demostraremos el siguiente lema técnico referido a la clase más general.

Lema 8. *Sea \mathcal{N} una red S^4PR y $r \in P_R$ un recurso de \mathcal{N} tal que el p-semiflujo mínimo de r , \mathbf{y}_r , verifica que $\mathbf{y}_r \in \{0, 1\}^{|P|}$. El cerrojo mínimo que contiene el recurso r , $D_r \in \mathcal{D}^1$, es $D_r = \|\mathbf{y}_r\|$.*

Demostración. El soporte de \mathbf{y}_r es un cerrojo por construcción ya que $(\|\mathbf{y}_r\|)^\bullet \subseteq \|\mathbf{y}_r\|^\bullet$ y contiene exactamente un recurso r . Demostramos a continuación que $\|\mathbf{y}_r\|$ es un cerrojo mínimo por contradicción. Supongamos que $\|\mathbf{y}_r\|$ es no mínimo. Esto quiere decir que existen al menos un lugar $p \in \|\mathbf{y}_r\|$ que es no esencial para el cerrojo, es decir, que si se elimina de $\|\mathbf{y}_r\|$ lo que queda es un cerrojo.

Ahora bien, p no puede ser el recurso r ya que en este caso el cerrojo $\|\mathbf{y}_r\| \setminus \{r\}$ estaría formado solo por lugares proceso. Sea q uno de estos lugares proceso, por la propiedad de cerrojo y teniendo en cuenta que todos los circuitos de la máquina de estados pasan por el lugar reposo, llegaríamos a que el lugar reposo pertenece a $\|\mathbf{y}_r\| \setminus \{r\}$ pero esto no es posible por la definición de redes S^4PR que dice que el lugar reposo nunca puede pertenecer a un p-semiflujo mínimo con recurso.

El lugar p tampoco puede ser un lugar proceso porque para que llegue a ser no esencial se requiere que para toda transición $t \in p^\bullet$ tal que $t^\bullet \cap \|\mathbf{y}_r\| \setminus \{p\} \neq \emptyset$ exista otro lugar $q \in t^\bullet$ que también pertenezca a $\|\mathbf{y}_r\| \setminus \{p\}$. La única posibilidad es que el lugar q anterior sea el lugar recurso r ya que $\|\mathbf{y}_r\| \setminus \{r\}$ son lugares proceso que pertenecen a una máquina de estados fuertemente conexas. Ahora bien, esto es imposible dado que tendríamos que la transición t verificaría:

$$1. \bullet t \cap \|\mathbf{y}_r\| = \{p, r\}.$$

$$2. t^\bullet \cap \|\mathbf{y}_r\| = \{s\}, s \in P_S.$$

Observe que s solo puede ser un lugar proceso ya que r es el único recurso que pertenece a $\|\mathbf{y}_r\|$. Por lo tanto; si tenemos que anular la columna t , es decir, $\mathbf{y}_r^t \cdot C[t] = 0$ no podemos tener que $\mathbf{y}_r \in \{0, 1\}^{|P|}$ ya que $-\mathbf{y}_r[p] - \mathbf{y}_r[r] + \mathbf{y}_r[s] = 0$ no tiene ninguna solución binaria con las tres variables distintas de cero.

Una vez demostrado que $\|\mathbf{y}_r\|$ es un cerrojo mínimo, por el lema 2 anterior se trata de único cerrojo mínimo que contiene al conjunto de recursos r . \square

Por lo tanto, como consecuencia del resultado anterior podemos escribir el resultado anunciado para las subclases $L - S^3PR$, S^3PR y $SOAR^2$. La clase de cerrojos conteniendo un único recurso está formada

Lema 9. *Sea \mathcal{N} una red perteneciente a la clase $L - S^3PR$ ó S^3PR ó $SOAR^2$. La clase de cerrojos mínimos conteniendo un único recurso está formada por todos y cada uno de los soportes de los p -semiflujos mínimos de recursos. Es decir, $\mathcal{D}^1 = \{\|\mathbf{y}_r\|, | \mathbf{y}_r \text{ es un } p\text{-semiflujo mínimo que contiene } r \in P_R\}$.*

Demostración. Los p -semiflujos mínimos de los recursos en esta clase de redes son todos p -semiflujos binarios, i.e., $\forall r \in P_R, \mathbf{y}_r \in \{0, 1\}^{|P|}$. Por lo tanto, de acuerdo con el lema 8 todo cerrojo mínimo que contenga solo un recurso es igual al soporte del correspondiente p -semiflujo mínimo. \square

De acuerdo con lo anterior, se concluye que la cota inferior del número de cerrojos mínimos en las subclases $L - S^3PR$, S^3PR y $SOAR^2$ no pueden ser mejorados con respecto a la previamente establecida por la clase más general de las redes S^4PR .

La cota superior establecida para el número de cerrojos mínimos en la clase S^4PR también es alcanzable en la subclase más pequeña de las consideradas como pone de manifiesto la red de Petri de la figura 2.6. En efecto, los cerrojos mínimos para esta red que pertenece a la clase $L - S^3PR$ son:

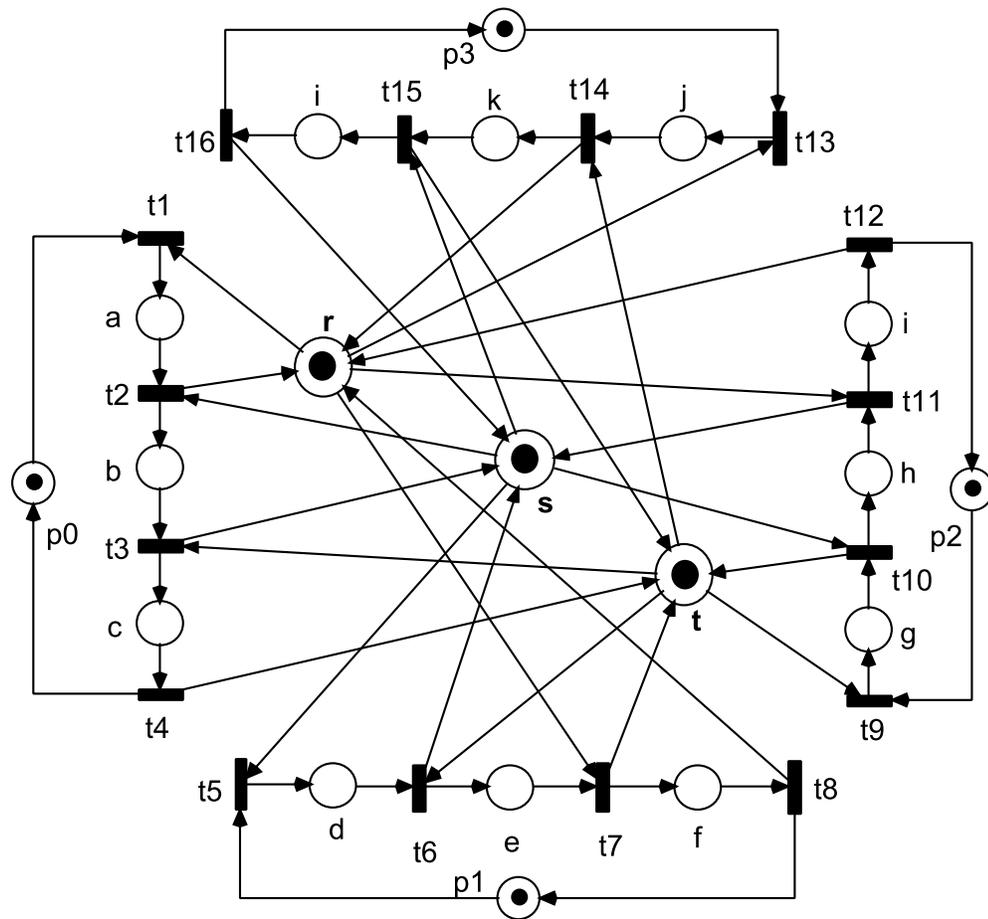


Figura 2.6: Red de Petri perteneciente a la clase $L - S^3PR$.

$$\begin{aligned}
D_r &= \{r, a, f, i, j\} \\
D_s &= \{s, b, d, h, l\} \\
D_t &= \{t, c, e, g, k\} \\
D_{rs} &= \{r, s, f, i, j, b, d, l\} \\
D_{st} &= \{s, t, h, l, c, e\} \\
D_{rt} &= \{r, t, a, f, i, c, g, k\} \\
D_{rst} &= \{r, s, t, f, i, l, c\}
\end{aligned}$$

Junto con los cuatro cerrojos mínimos correspondientes a las cuatro máquinas de estado que aparecen en la red:

$$\begin{aligned}
D_1 &= \{p_0, a, b, c\} \\
D_2 &= \{p_1, d, e, f\} \\
D_3 &= \{p_2, g, h, i\} \\
D_4 &= \{p_3, j, k, l\}
\end{aligned}$$

Por lo tanto el número de cerrojos mínimos es igual a once cerrojos que es precisamente el valor de la cota superior establecida para la clase de redes S^4PR .

$$2^{|P_R|} + |I_{\mathcal{N}}| - 1 = 2^3 + 4 - 1 = 11$$

Con lo cual el resultado no puede ser mejorado.

2.5. Conclusión

En este capítulo se han presentado las propiedades de estructura y composición que poseen los cerrojos de las redes S^4PR . Estas propiedades no han sido antes investigadas en la literatura de la clase S^4PR . En trabajos previos se asumía que los cerrojos mínimos en las redes S^4PR no tenían propiedades más específicas que las de las redes generales y por tanto los algoritmos de cálculo de los cerrojos mínimos no aprovechaban las propiedades estructurales que estas redes poseen.

Partiendo de la definición de la clase podemos establecer propiedades básicas tales como: todo cerrojo debe poseer lugares proceso, así como también cada cerrojo con al menos un lugar recurso debe tener lugares proceso. Además, cada cerrojo mínimo es único para un determinado conjunto de recursos. En este punto podemos destacar que es la primera vez que se caracterizan los cerrojos mínimos por el conjunto de recursos que contienen.

Además, se particiona en clases, \mathcal{D}^i , al conjuntos de cerrojos mínimos de una red S^4PR . En donde cada una de ellas está caracterizada por el número de recursos pertenecientes a cada cerrojo dentro de la misma. Adicionalmente, se demostró que esta partición de los cerrojos mínimos en el conjunto de clases, \mathcal{D}^i , no significa que se debe tener por lo menos un cerrojo en cada clase. También se demostró que cada cerrojo mínimo es obtenido a partir del p-semiflujo, posiblemente, eliminando algunos lugares para hacerlo mínimo.

Otros resultados importantes establecen: que un cerrojo mínimo conteniendo un conjunto de recursos $D_R \neq \emptyset$, está siempre contenido en la unión de los cerrojos mínimos $D_r \in \mathcal{D}^1, r \in D_R$. También, se indican las cotas inferior y superior de los números de cerrojos mínimos de una red S^4PR siendo este resultado un aporte nuevo en la literatura. Mediante ejemplos se ha demostrado que estas cotas son alcanzables. La última propiedad de los cerrojos indica que el lugar reposo no puede pertenecer a un cerrojo mínimo con recursos.

Finalmente, es importante destacar que estas propiedades nos permitirán desarrollar en el siguiente capítulo las herramientas necesarias para calcular en el capítulo 4 el conjunto de cerrojos para una red S^4PR que contenga determinados recursos al igual que el conjunto de cerrojos mínimos para la misma de una manera más eficiente.

Capítulo 3

La Relación de Poda entre Cerrojos en Redes S^4PR

Resumen

En el capítulo anterior se demostró que como mucho existe un cerrojo mínimo que contenga un subconjunto determinado de recursos. Aprovechando esta propiedad se puede concebir un algoritmo para calcular el conjunto de cerrojos mínimos realizando un cálculo para cada subconjunto de recursos consistente en realizar la unión de los cerrojos mínimos que contienen cada uno un recurso del subconjunto de recursos dado. Los cerrojos así obtenidos son no-mínimos.

Para obtener del conjunto de cerrojos resultante aquellos que son mínimos debemos desarrollar nuevas herramientas cuyo objetivo será cribar los cerrojos mínimos del conjunto de cerrojos resultantes de la operación de unión. La primera herramienta a definir en este capítulo será la *Relación de Poda* la cual nos permite determinar los lugares candidatos a ser podados de un cerrojo mínimo de un solo recurso en la operación de unión por parte de otro cerrojo de las mismas características con él relacionado.

Para representar la relación de poda entre cerrojos se define el *Grafo de Poda* y tres funciones de etiquetado las cuales nos permiten determinar si existe algún lugar perteneciente a un cerrojo mínimo de un recurso que pueda ser podado por otro cerrojo mínimo de un recurso. A partir de esta información podemos calcular correctamente los lugares a eliminar de dicho cerrojo. Se investigarán las propiedades que poseen estos grafos de tal forma que nos conduzcan a una correcta caracterización de los cerrojos resultantes de la operación de unión de los mismos.

De igual forma, se define la *Relación de Poda Extendida* que nos permitirá considerar otros conjuntos de cerrojos que no contengan el mismo conjunto de recursos que los

considerados en la relación de poda previa, entre estos se pueden incluir los cerrojos no-mínimos y cerrojos conteniendo más de un recurso. La principal diferencia entre ambas relaciones de poda la podemos ubicar en la no-minimalidad de los cerrojos. Se representará la relación de poda extendida mediante el *Grafo de Poda Extendido* y tres funciones de etiquetado.

3.1. Introducción

En el capítulo anterior se explicó que un cerrojo mínimo conteniendo un conjunto de recursos $D_R \neq \emptyset$, está siempre contenido en la unión de los cerrojos mínimos de un solo recurso $D_r \in \mathcal{D}^1, r \in D_R$. Dado que cada uno de los operandos en la operación de unión es un cerrojo, obviamente, el resultado de esta operación será un cerrojo pero no es mínimo.

Por lo previamente explicado, si deseamos obtener sólo aquellos cerrojos que son mínimos como resultado de la operación de unión de cerrojos mínimos conteniendo un solo recurso debemos identificar las diferentes *fuentes de no-minimalidad* que pueden presentar cada uno de los cerrojos involucrados en dicha operación. A partir de esta información podemos definir formalmente las herramientas que nos permitirán conservar del conjunto de cerrojos aquellos que nos conduzcan a obtener un cerrojo mínimo como resultado de la operación de unión de los mismos.

En este capítulo nos centraremos en explicar cada uno de los conceptos antes mencionados y en desarrollar las herramientas mediante las cuales obtener los cerrojos mínimos como resultado de la operación de unión de un conjunto de cerrojos. En donde, la primera herramienta a desarrollar será la *relación de poda* por medio de la cual podamos determinar los elementos candidatos a ser podados del conjunto de cerrojos mínimos de un solo recurso en la operación de unión.

El resto del capítulo se organiza como sigue. En la sección 3.2 se explica la relación de poda. Para representar la relación de poda entre éstos cerrojos se define el *Grafo de Poda* en la sección 3.3 y tres funciones de etiquetado asociadas, las cuales nos permiten determinar si existe algún lugar perteneciente a un cerrojo mínimo de un recurso que pueda ser podado por otro cerrojo mínimo de un recurso. A partir de esta información podemos calcular correctamente los lugares a eliminar de dicho cerrojo. Adicionalmente, en la sección 3.4 se investigan las propiedades que poseen estos grafos de tal forma que nos conduzcan a una correcta caracterización de los cerrojos resultantes de la operación de unión de los mismos. En la sección 3.5 se define la *Relación de Poda Extendida* que nos permitirá considerar otros conjuntos de cerrojos que no contengan el mismo conjunto de recursos que los considerados en la relación de poda previa, entre estos se pueden

incluir los cerrojos no-mínimos y cerrojos conteniendo más de un recurso. Esta relación será representada mediante el *Grafo de Poda Extendido* en la sección 3.6 y tres funciones de etiquetado asociadas. En la sección ?? se investigarán las propiedades para este nuevo grafo de tal forma que caractericen a cada uno de los conjuntos de cerrojos que este posee.

En lo que sigue en este capítulo nos concentraremos solo en los cerrojos que contienen algún recurso. Dado que estos cerrojos nunca pueden contener el lugar reposo, en lo que sigue las redes S^4PR que aparecen en las figuras, en la mayor parte de los casos, no contienen los lugares reposo para simplificar la representación.

3.2. La Relación de Poda entre Cerrojos

El algoritmo, que se explicará en el siguiente capítulo, para calcular los cerrojos mínimos en las redes S^4PR está principalmente basado en el Lema 2: esto es, dado un conjunto de recursos $D_R \subseteq P_R$ existe como mucho un cerrojo mínimo conteniendo este conjunto de recursos. Por lo tanto, para calcular todos los cerrojos mínimos necesitamos, al menos, considerar $2^{|P_R|} - 1$ subconjuntos candidatos, cada uno conteniendo un subconjunto no-vacio de recursos diferentes. Una forma fácil para construir cada uno de estos $2^{|P_R|} - 1$ cerrojos candidatos es por la unión de los cerrojos mínimos de \mathcal{D}^1 correspondientes a los recursos incluidos en el cerrojo a ser construido.

Obviamente, el resultado de la operación de unión es un cerrojo porque cada uno de los operandos es un cerrojo, pero *no es mínimo*. Esta *no-minimalidad* puede surgir de los lugares proceso, p , que se convierten en *no-esenciales*, i.e. hay otros lugares de entrada a las transiciones p^\bullet que permiten eliminar a p . La única posibilidad, en este caso, es que estos nuevos lugares sean lugares recurso diferentes a esos que previamente hacen a p *esencial*. Estos lugares recurso aparecen en la operación de unión. En la figura 3.1 encontramos dos cerrojos mínimos D_r y D_s para los que los lugares p y q son esenciales, respectivamente. Efectivamente, por ejemplo, en el cerrojo $D_r = \{p, u, r\}, t_2 \in \bullet r$ y por lo tanto, $t_2 \in \bullet D_r$. Para que t_2 pertenezca a D_r^\bullet obligatoriamente se debe cumplir que p pertenezca al cerrojo (ya que la condición para que un conjunto de lugares, D , sea un cerrojo es que $\bullet D \subseteq D^\bullet$). Lo mismo ocurre con el lugar q en el cerrojo $D_s = \{q, v, s\}$ razonando con respecto a la transición t_5 .

No obstante, cuando consideramos el cerrojo resultante de la unión de ambos $D_{rs} = D_r \cup D_s$, ambos lugares, p y q , dejan de ser esenciales en el cerrojo D_{rs} y por lo tanto éste es no mínimo. Efectivamente, en D_{rs} para la transición $t_2 \in \bullet r$ tenemos ahora dos lugares candidatos que permiten que $t_2 \in D_{rs}^\bullet$. Estos lugares son el lugar p y el recién incorporado lugar s , que es el lugar recurso que aporta el cerrojo D_s en la unión y que hace que el lugar p deje de ser esencial. Por lo tanto, si queremos obtener un cerrojo mínimo a partir de D_{rs}

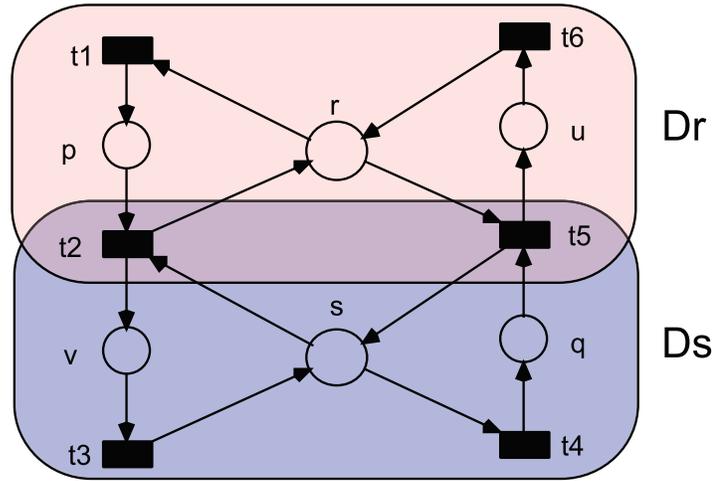


Figura 3.1: El lugar p es esencial para el cerrojo mínimo $D_r = \{p, u, r\}$ y el lugar q es esencial para el cerrojo mínimo $D_s = \{q, v, s\}$.

el lugar p deberá ser eliminado de D_{rs} . De la misma forma se puede razonar para el caso del lugar q , que deja de ser esencial en el cerrojo D_{rs} a causa del lugar recurso r aportado por el cerrojo D_r .

La figura 3.2 muestra un ejemplo que ilustra el hecho que un lugar para llegar a ser no esencial en la unión de varios cerrojos mínimos debe ser no esencial para todas sus transiciones de salida.

Efectivamente, en la figura 3.2 el lugar p es no esencial en el cerrojo $D_{rs} = D_r \cup D_s = \{p, q, r, s\}$ si se atiende exclusivamente a la transición t_2 , dado que ahora será el lugar s el que haga que $t_2 \in D_{rs}^\bullet$ dado que $t_2 \in s^\bullet$. Sin embargo, no se puede eliminar el lugar p del cerrojo D_{rs} dado que p es el único lugar de entrada a la transición t_1 y por lo tanto p debe pertenecer a D_{rs} si queremos que r se mantenga en el cerrojo.

La otra fuente de *no-minimalidad* surge cuando ninguno de los lugares de un cerrojo $D_r \in \mathcal{D}^1$ se convierte en *no-esencial*. En este caso, el cerrojo mínimo completo D_r está contenido en el cerrojo resultante de la operación de unión y por lo tanto este *no será mínimo*. Además, en este caso, no hay un cerrojo mínimo conteniendo el pretendido conjunto de recursos puesto que, al menos, uno de ellos: r , no puede pertenecer porque D_r está contenido. La red de la figura 3.2 bosqueja esta fuente de *no-minimalidad*. Ya hemos visto que p no puede eliminarse del cerrojo D_{rs} dado que es esencial debido a r y la transición t_1 . Por lo tanto, $\{r, p\} = D_r \subseteq D_{rs}$, es decir, D_{rs} contiene el cerrojo mínimo D_r y por tanto, el mismo D_{rs} no puede ser mínimo.

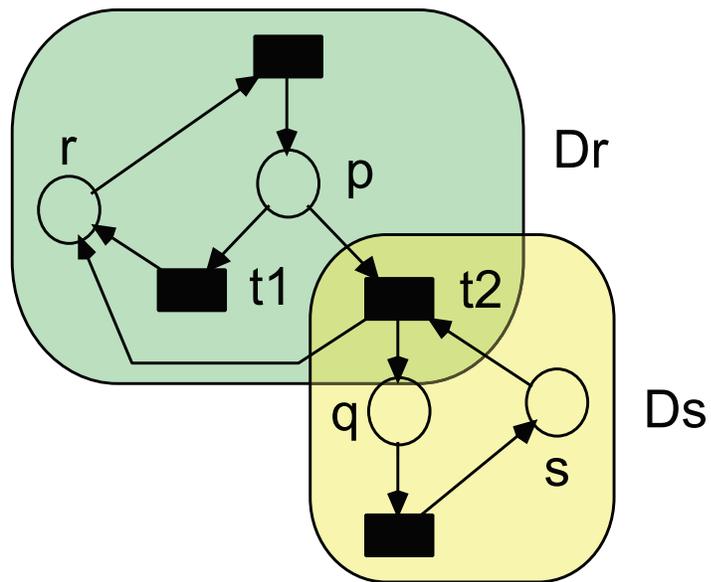


Figura 3.2: El lugar p es esencial para el cerrojo mínimo $D_r = \{p, r\}$, y sigue siendo esencial en el cerrojo $D_{rs} = D_r \cup D_s = \{p, q, r, s\}$ dado que es el único lugar de entrada a la transición $t_1 \in \bullet r$.

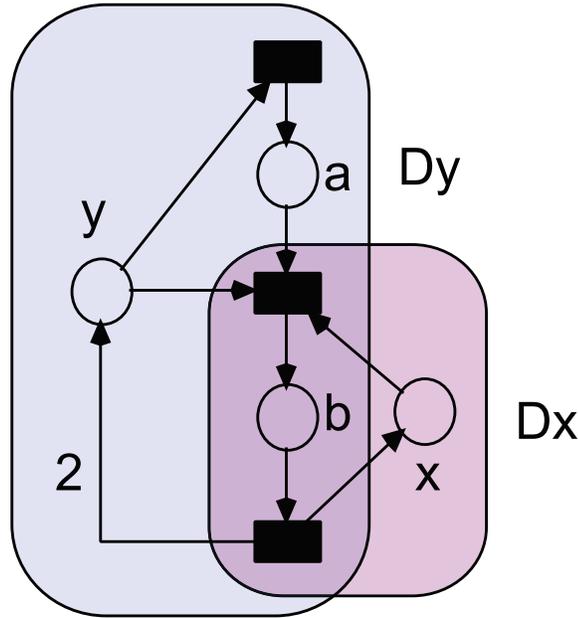


Figura 3.3: El lugar b es esencial para los cerrojos D_x y D_y . Al realizar la operación de unión entre ambos cerrojos no existe ningún lugar que se convierta en no-esencial.

La red de Petri de la figura 3.3 muestra dos cerrojos mínimos $D_x, D_y \in \mathcal{D}^1$. En la misma podemos observar que el lugar b es esencial para los cerrojos D_x y D_y y no existe ningún lugar que se convierta en no-esencial al realizar la operación de unión entre ambos cerrojos. Por lo tanto el cerrojo resultante de la operación de unión D_{xy} es no-mínimo.

A partir de la discusión informal previa con respecto a la *no-minimalidad* del resultado de la unión de un subconjunto de cerrojos pertenecientes a \mathcal{D}^1 , en esta sección definimos un conjunto de herramientas y resultados formales. Éstos nos permiten cribar cada uno de los $2^{|P_R|} - 1$ cerrojos candidatos, para conservar solamente aquellos cerrojos que dan lugar a los cerrojos mínimos.

La primera herramienta es la así llamada *relación de poda*¹ definida sobre el conjunto \mathcal{D}^1 . Diremos que el cerrojo $D_r \in \mathcal{D}^1$ poda al cerrojo $D_x \in \mathcal{D}^1, r \neq x$, si, y solo si, $T_{rx} = D_r \bullet \cap D_x \bullet \neq \emptyset$ (los dos cerrojos comparten alguna transición) y $U_r = r \bullet \cap T_{rx} \cap (\bullet T_{rx} \cap D_x \cap P_s) \bullet \neq \emptyset$ (existen transiciones comunes con un lugar proceso de entrada perteneciente a D_x y también el recurso r entra a esas transiciones). Por ejemplo, en la

¹En [CRC10a] y [CRC10b] esta relación se denominó *pruning relation*, y es por ello que aquí se ha traducido literalmente a español como relación de poda.

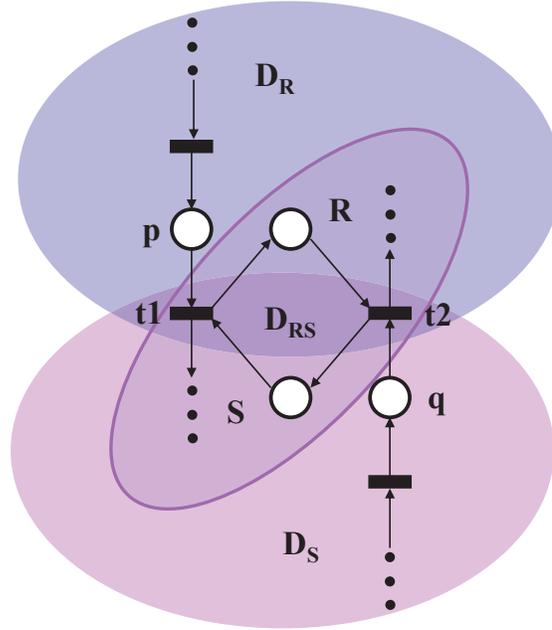
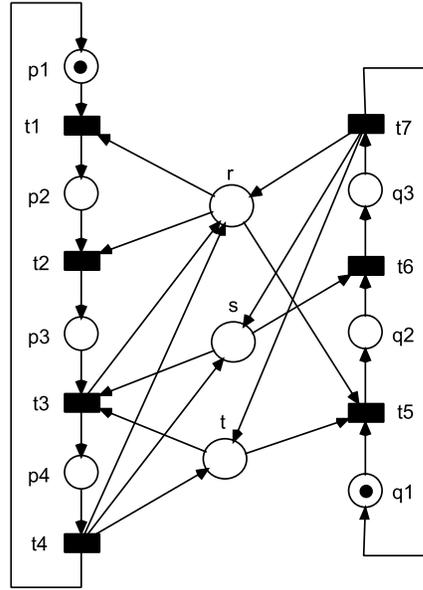


Figura 3.4: Diagrama que ilustra como lugares proceso de los cerrojos D_R y D_S llegan a ser *no-esenciales* en el cerrojo $D_R \cup D_S$.

figura 3.4 se ilustra de una manera gráfica la relación de poda definida en los cerrojos mínimos $D_R, D_S \in \mathcal{D}^1$. Estos cerrojos comparten las transiciones $T_{rx} = \{t_1, t_2\}$, en donde, $t_1 \in S^\bullet$ y $t_2 \in R^\bullet$. Además, ${}^\bullet T_{rx} \cap D_R \cap P_S = \{p\}$, ${}^\bullet T_{rx} \cap D_S \cap P_S = \{q\}$, $({}^\bullet T_{rx} \cap D_R \cap P_S)^\bullet = \{t_1\}$ y $({}^\bullet T_{rx} \cap D_S \cap P_S)^\bullet = \{t_2\}$ verificando que $U_r \neq \emptyset$.

Los elementos candidatos a ser podados del cerrojo D_x por el cerrojo D_r están en cada uno de los pares $(t, {}^\bullet t \cap P_S)$, donde $t \in U_r$. El lugar candidato a ser eliminado por uno de esos pares es ${}^\bullet t \cap P_S$. Este lugar puede ser eliminado porque se convierte en *no-esencial* en $D_r \cup D_x$, sólo si $({}^\bullet t \cap P_S)^\bullet \subseteq U_r$, i.e. todas sus transiciones de salida pertenecen al conjunto previamente definido U_r . Estos corresponden a los lugares p y q en la figura 3.4 como hemos explicado anteriormente.

A partir de la discusión informal anterior, a continuación se formaliza la relación de poda entre cerrojos de una red S^4PR que contengan sólo un lugar recurso. La razón de ello ya fue apuntada anteriormente, y se debe a que el conjunto de cerrojos \mathcal{D}^1 de una red S^4PR no es necesario calcularlo ya que viene definido a través de los p-semiflujos mínimos de cada uno de los recursos de la red. Además, nuestro objetivo finalmente será el cálculo de todos los cerrojos mínimos partiendo de este conjunto. No obstante, esta relación será extendida posteriormente para considerar clases más amplias de cerrojos de

Figura 3.5: Una red S^4PR .

una red S^4PR .

Definición 7. Relación de poda entre cerrojos que contienen un sólo recurso en redes S^4PR .

Sea $\mathcal{N} = (P_0 \cup P_S \cup P_R, T, C)$ una red S^4PR y sean D_r, D_s dos cerrojos de \mathcal{N} que contienen solo un recurso: $D_r \cap P_R = \{r\}$ y $D_s \cap P_R = \{s\}$. El cerrojo D_r está en relación de poda con el cerrojo D_s o simplemente el cerrojo D_r poda al cerrojo D_s , si y solo si,

$$U_r = r^\bullet \cap T_{rs} \cap (\bullet T_{rs} \cap D_s \cap P_s)^\bullet \neq \emptyset$$

con

$$T_{rs} = D_r^\bullet \cap D_s^\bullet$$

□

La relación de poda definida sobre conjuntos de cerrojos, cada uno de los cuales contiene exactamente un recurso, en general no es reflexiva, ni simétrica, ni transitiva. Analizamos este punto con un poco más de detalle.

Lema 10. Sea $\mathcal{N} = (P_0 \cup P_S \cup P_R, T, C)$ una red S^4PR y sea D_r un cerrojo de \mathcal{N} que contiene un sólo recurso: $D_r \cap P_R = \{r\}$. Si existe una transición t tal que tiene un lugar

proceso, p , del cerrojo y el lugar recurso r como entradas: $p \in \bullet t$ y $r \in \bullet t$ con $p, r \in D_r$; entonces el cerrojo D_r está relacionado consigo mismo mediante la relación de poda. \square

Demostración. De acuerdo con la definición de la relación de poda D_r estará relacionado con si mismo si

$$U_r = r^\bullet \cap T_{rr} \cap (\bullet T_{rr} \cap D_r \cap P_s)^\bullet \neq \emptyset$$

Teniendo en cuenta que $T_{rr} = D_r^\bullet \cap D_r^\bullet = D_r^\bullet$ la expresión de U_r queda:

$$U_r = r^\bullet \cap D_r^\bullet \cap (\bullet (D_r^\bullet) \cap D_r \cap P_s)^\bullet = r^\bullet \cap (D_r \cap P_s)^\bullet = r^\bullet \cap (D_r \setminus \{r\})^\bullet$$

Dado que $p \in D_r$ y $p \in \bullet t$ a la par que $r \in \bullet t$, concluimos que $t \in U_r$ y por tanto $U_r \neq \emptyset$. \square

En el ejemplo de la figura 3.5 se puede observar las siguientes relaciones para los cerrojos

$$\begin{aligned} D_r &= \{r, p_2, p_3, p_4, q_2, q_3\} \\ D_s &= \{s, p_4, q_3\} \\ D_t &= \{t, p_4, q_2, q_3\} \end{aligned}$$

a) D_r está relacionado consigo mismo dado que

$$\begin{aligned} U_r &= r^\bullet \cap T_{rr} \cap (\bullet T_{rr} \cap D_r \cap P_s)^\bullet \\ &= \{t_1, t_2, t_5\} \cap \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\} \cap (\{p_2, p_3, p_4, q_2, q_3\})^\bullet \\ &= \{t_1, t_2, t_5\} \cap (\{p_2, p_3, p_4, q_2, q_3\})^\bullet \\ &= \{t_1, t_2, t_5\} \cap \{t_2, t_3, t_4, t_6, t_7\} \\ &= \{t_2\} \neq \emptyset \end{aligned}$$

Es decir, se trata de un cerrojo con una transición como la t_2 que cumple las condiciones de la propiedad anterior. Óbserve, que en este caso p_2 es no-esencial y puede ser eliminado de D_r resultando entonces un cerrojo mínimo.

b) D_s no está relacionado consigo mismo por la relación de poda. Efectivamente,

$$\begin{aligned} U_s &= s^\bullet \cap T_{ss} \cap (\bullet T_{ss} \cap D_s \cap P_s)^\bullet \\ &= s^\bullet \cap (D_s \setminus \{s\})^\bullet \\ &= \{t_3, t_6\} \cap \{t_4, t_7\} \\ &= \emptyset \end{aligned}$$

- c) Los cerrojos D_r y D_s no están relacionados simétricamente, es decir, D_r no está relacionado con D_s y D_s está relacionado con D_r . Efectivamente,

$$\begin{aligned}
 U_r &= r^\bullet \cap T_{rs} \cap (\bullet T_{rs} \cap D_s \cap P_s)^\bullet \\
 &= \{t_1, t_2, t_5\} \cap \{t_3, t_4, t_6, t_7\} \cap (\{p_3, p_4, q_2, q_3\} \cap \{p_4, q_3\})^\bullet \\
 &= \{t_1, t_2, t_5\} \cap \{t_3, t_4, t_6, t_7\} \cap \{t_4, t_7\} \\
 &= \emptyset
 \end{aligned}$$

$$\begin{aligned}
 U_s &= s^\bullet \cap T_{sr} \cap (\bullet T_{sr} \cap D_r \cap P_s)^\bullet \\
 &= \{t_3, t_6\} \cap \{t_3, t_4, t_6, t_7\} \cap (\bullet \{t_3, t_4, t_6, t_7\} \cap \{r, p_2, p_3, p_4, q_2, q_3\} \cap P_s)^\bullet \\
 &= \{t_3, t_6\} \cap (\{p_3, p_4, q_2, q_3\})^\bullet \\
 &= \{t_3, t_6\} \neq \emptyset
 \end{aligned}$$

De manera similar, los cerrojos D_t y D_s no están relacionados simétricamente, es decir, D_t no está relacionado con D_s y D_s está relacionado con D_t . Efectivamente,

$$\begin{aligned}
 U_t &= t^\bullet \cap T_{ts} \cap (\bullet T_{ts} \cap D_s \cap P_s)^\bullet \\
 &= \{t_3, t_5\} \cap \{t_3, t_4, t_6, t_7\} \cap (\bullet \{t_3, t_4, t_6, t_7\} \cap \{s, p_4, q_3\} \cap P_s)^\bullet \\
 &= \{t_3\} \cap (\{p_3, p_4, q_2, q_3\} \cap \{p_4, q_3\})^\bullet \\
 &= \{t_3\} \cap \{t_4, t_7\} \\
 &= \emptyset
 \end{aligned}$$

$$\begin{aligned}
 U_s &= s^\bullet \cap T_{st} \cap (\bullet T_{st} \cap D_t \cap P_s)^\bullet \\
 &= \{t_3, t_6\} \cap \{t_3, t_4, t_6, t_7\} \cap (\bullet \{t_3, t_4, t_6, t_7\} \cap \{t, p_4, q_2, q_3\} \cap P_s)^\bullet \\
 &= \{t_3, t_6\} \cap (\{p_3, p_4, q_2, q_3\} \cap \{p_4, q_2, q_3\})^\bullet \\
 &= \{t_3, t_6\} \cap \{t_4, t_6, t_7\} \\
 &= \{t_6\} \neq \emptyset
 \end{aligned}$$

- d) Los cerrojos D_r y D_t no están relacionados transitivamente. En efecto, como hemos demostrado antes los cerrojos D_r y D_s no están en relación de poda. Por lo tanto, podemos concluir que no existe una relación transitiva entre los cerrojos D_r y D_t . Efectivamente,

$$\begin{aligned}
 U_r &= r^\bullet \cap T_{rt} \cap (\bullet T_{rt} \cap D_t \cap P_s)^\bullet \\
 &= \{t_1, t_2, t_5\} \cap \{t_3, t_4, t_5, t_6, t_7\} \cap (\{p_3, p_4, q_2, q_3\} \cap \{p_4, q_2, q_3\})^\bullet \\
 &= \{t_5\} \cap \{t_4, t_6, t_7\} \\
 &= \emptyset
 \end{aligned}$$

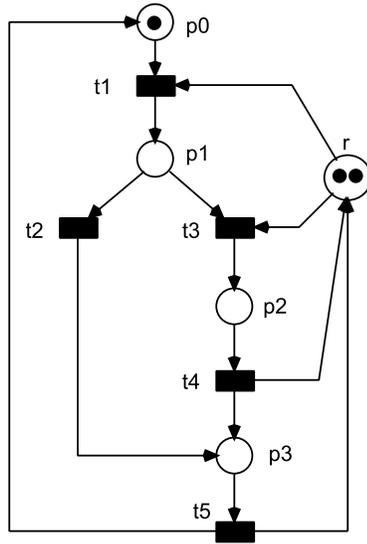


Figura 3.6: El cerrojo $D_r = \{r, p_1, p_2, p_3\}$ está relacionado consigo mismo por la relación de poda pero es un cerrojo mínimo.

En los ejemplos posteriores se verán más ejemplos de relaciones existentes entre los elementos de un conjunto de cerrojos cada uno de los cuales contienen sólo un recurso. No obstante, en las secciones siguientes nos centraremos sobre todo en la relación de poda definida sobre el conjunto de cerrojos \mathcal{D}^1 , es decir, cerrojos mínimos que contienen exactamente un recurso.

Conviene resaltar que el hecho que un cerrojo esté relacionado consigo mismo por la relación de poda no significa que el cerrojo sea no mínimo como queda puesto de manifiesto en el cerrojo $D_r = \{r, p_1, p_2, p_3\}$ de la red S^4PR de la figura 3.6. En efecto,

$$\begin{aligned} U_r &= r^\bullet \cap T_{rr} \cap (\bullet T_{rr} \cap D_r \cap P_s)^\bullet = r^\bullet \cap (D_r \setminus \{r\})^\bullet \\ &= \{t_1, t_3\} \cap \{t_2, t_3, t_4, t_5\} = \{t_3\} \neq \emptyset \end{aligned}$$

Por lo tanto, al ser $U_r \neq \emptyset$, concluimos que D_r está relacionado consigo mismo por la relación de poda. Sin embargo, se puede apreciar que el cerrojo es mínimo (si se trata de quitar cualquiera de los lugares del cerrojo, el resultado no es un cerrojo).

En el ejemplo de la figura 3.7 se puede observar que el cerrojo no-mínimo $D_r = \{a, b, r, c\}$ no está relacionado consigo mismo por la relación de poda. Efectivamente,

$$U_r = r^\bullet \cap T_{rr} \cap (\bullet T_{rr} \cap D_r \cap P_s)^\bullet$$

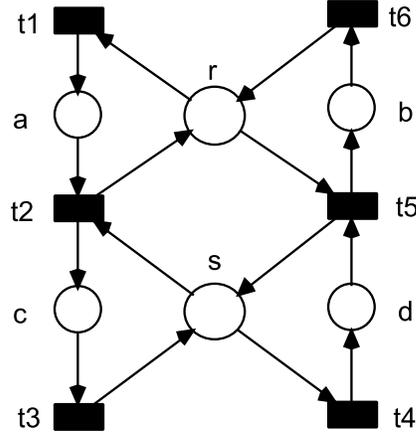


Figura 3.7: El cerrojo no-mínimo $D_r = \{a, b, r, c\}$ no está relacionado consigo mismo por la relación de poda.

$$\begin{aligned}
 &= \{t_1, t_5\} \cap \{t_1, t_2, t_5, t_6, \} \cap (\bullet\{t_1, t_2, t_5, t_6\} \cap \{a, b, r, c\} \cap P_s)^\bullet \\
 &= \{t_1, t_5\} \cap (\{a, b\})^\bullet = \{t_1, t_5\} \cap \{t_2, t_6\} = \emptyset
 \end{aligned}$$

3.3. El Grafo de Poda

Para realizar una presentación más intuitiva y razonar de una manera gráfica en los desarrollos posteriores en este capítulo representaremos la relación de poda sobre un conjunto de cerrojos cada uno de los cuales contiene exactamente un recurso mediante un grafo que denominaremos *Grafo de Poda*² (PG). En el apéndice B el lector puede encontrar la notación y definiciones utilizadas en relación con la Teoría de Grafos.

Definición 8. Sea \mathcal{N} una red S^4PR y P_R el conjunto de lugares recurso. El Grafo de Poda (PG) de \mathcal{N} es un grafo $G = (V, E)$ donde,

1. $V = P_R$; y
2. $E \subseteq V \times V$ y para todo $r, x \in P_R$, $(r, x) \in E$ si, y solo si, el cerrojo $D_r \in \mathcal{D}^1$ poda al cerrojo $D_x \in \mathcal{D}^1$, es decir,

$$U_r = r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap D_x \cap P_s)^\bullet \neq \emptyset$$

²Pruning Graph en terminología inglesa.

con

$$T_{rx} = D_r^\bullet \cap D_x^\bullet.$$

Óbserve que según la definición anterior un vértice puede tener un bucle si el cerrojo que contiene está en relación de poda consigo mismo como se indica en el lema 10.

Dado un grafo de poda $G = (V, E)$, definimos el subgrafo de poda G' de G inducido por el conjunto de vértices $V' \subseteq V$, como el grafo $G' = (V', E \cap (V' \times V'))$. Asociado al grafo o subgrafo de poda definimos las siguientes tres funciones de etiquetado de los conjuntos V y E .

Definición 9. Sea \mathcal{N} una red S^4PR y $G = (V, E)$ un grafo o subgrafo de poda de \mathcal{N} .

- a) La función de etiquetado de los Recursos. $S : P_R \rightarrow \mathcal{D}^1$, donde para todo $r \in P_R$, $S(r) = D_r \in \mathcal{D}^1$.
- b) La función de etiquetado de los Arcos. $L : E \rightarrow 2^{T \times P_s}$, donde $\forall (r, x) \in E$,

$$L((r, x)) = \{(t, p) \mid t \in r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap S(x) \cap P_s)^\bullet; T_{rx} = S(r)^\bullet \cap S(x)^\bullet; \{p\} = \bullet t \cap P_s\}.$$
- c) La función de etiquetado de Poda. $K_G : V \rightarrow 2^{P_s}$, donde para todo $r \in V$, $K_G(r) \subseteq P_s$ y se calcula mediante el algoritmo 1.

La función S asigna a cada vértice del grafo, i.e. cada recurso de la red, el cerrojo mínimo perteneciente a \mathcal{D}^1 conteniendo el recurso representado en el vértice del grafo. Obsérvese, que en las redes S^4PR existe un p-semiflujo mínimo para cada recurso y por lo tanto existe un cerrojo mínimo para cada recurso y que contiene exactamente un recurso. Por esta razón la función S está bien definida sobre el conjunto P_R . Por ejemplo, para construir el Grafo de Poda, G , de la red de la figura 3.8, la función S asigna a cada vértice del grafo los cerrojos mínimos $S(R) = D_R \in \mathcal{D}^1$, $S(S) = D_S \in \mathcal{D}^1$ y $S(T) = D_T \in \mathcal{D}^1$, donde los valores de D_R, D_S y D_T se describen en la figura 3.9.

La función L etiqueta cada arco $(r, x) \in E$ con el conjunto de pares (t, p) representando un lugar candidato p a ser podado por el cerrojo mínimo D_r en el cerrojo mínimo D_x cuando construimos $D_r \cup D_x$. Además, para aquellos cerrojos que están relacionados consigo mismo teniendo en cuenta el lema 10 la función L etiqueta cada arco $(r, r) \in E$ con el conjunto de pares (t, p) representando el lugar candidato p a ser podado en el cerrojo D_r . Ambas funciones, S y L , están definidas para el grafo $G = (V, E)$ y para cada subgrafo $G' = (V', E')$, $V' \subseteq V$ y $E' = E \cap (V' \times V')$ las correspondientes funciones S y L son las funciones para G pero restringidas al conjunto de vértices V' .

La función K_G es una función que se tiene que calcular para cada grafo o subgrafo. Para cada nodo r , $K_G(r)$ es el conjunto de lugares que deberán ser podados del cerrojo mínimo

Algoritmo 1 Función de etiquetado de Poda de G.

Entrada: $\mathcal{N} = \langle P_0 \cup P_s \cup P_R, T, C \rangle$ – La red S^4PR .

$r \in P_R$ – El recurso de un cerrojo $D_r \in \mathcal{D}^1$ para el cual calcularemos el conjunto de poda en el grafo G.

$G = (V, E)$ – El grafo o subgrafo de poda de \mathcal{N} y las funciones asociadas L y S .

Salida: $K_G(r) \subseteq P_s$

1. **Inicio**

2. $T_F = \{t | (t, p) \in L((x, r)) \text{ y } (x, r) \in E\}$

3. $P_{parcial} = \{p | (t, p) \in L((x, r)); (x, r) \in E; \text{ y } p^\bullet \cap T_F \neq p^\bullet\}$

4. $K_G(r) = \{p | (t, p) \in L((x, r)); (x, r) \in E; \text{ y } p^\bullet \subseteq T_F\}$

5. $T_{nuevo} = \bullet K_G(r) \setminus r^\bullet$

6. **Mientras que** $T_{nuevo} \neq \emptyset$ **Hacer**

7. $T_F = T_F \cup T_{nuevo}$

8. $P_{nuevo} = \bullet T_{nuevo} \cap P_s \cap S(r)$

9. $A = \{p | p \in P_{nuevo}; p^\bullet \subseteq T_F\}$

10. $B = \{p | p \in P_{parcial}; p^\bullet \subseteq T_F\}$

11. $C = \{p | p \in P_{nuevo}; p^\bullet \cap T_F \neq p^\bullet\}$

12. $K_G(r) = K_G(r) \cup A \cup B$

13. $P_{parcial} = (P_{parcial} \setminus B) \cup C$

14. $T_{nuevo} = \bullet (A \cup B) \setminus r^\bullet$

15. **Fin Mientras que**

16. **Fin**

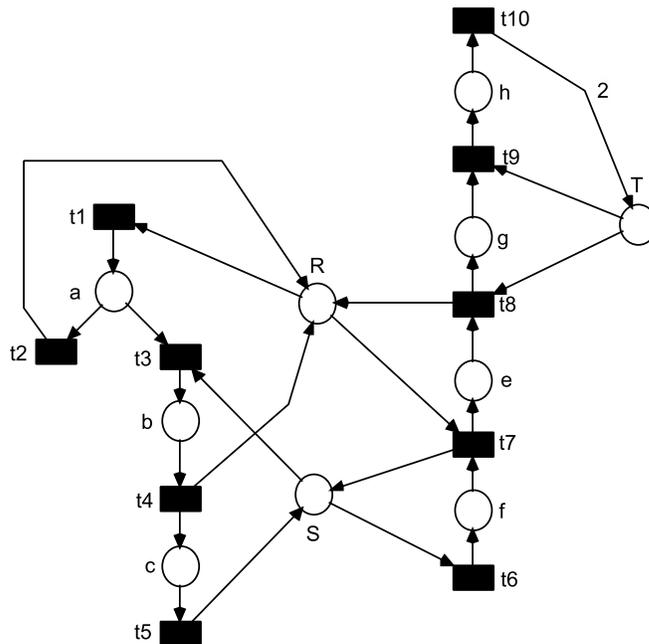


Figura 3.8: Una red S^4PR que no tiene cerrojos mínimos que contengan más de un recurso.

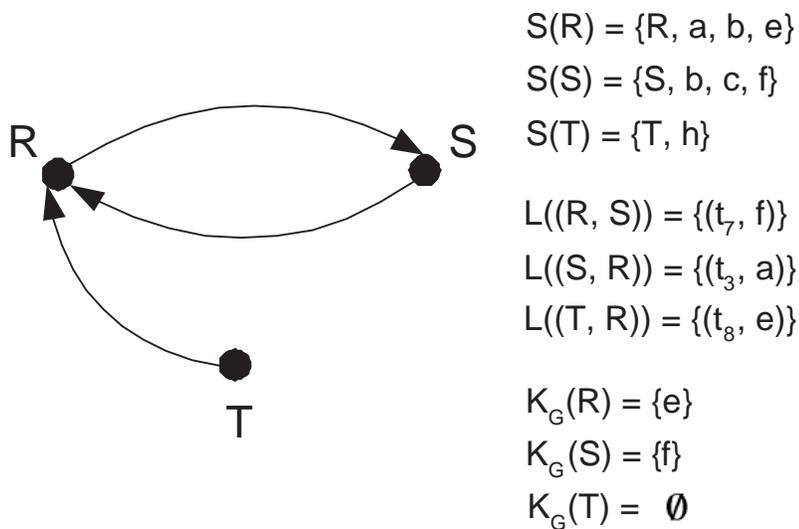


Figura 3.9: El Grafo de Poda, G , de la red de la figura 3.8 y las funciones que etiquetan sus vértices y arcos.

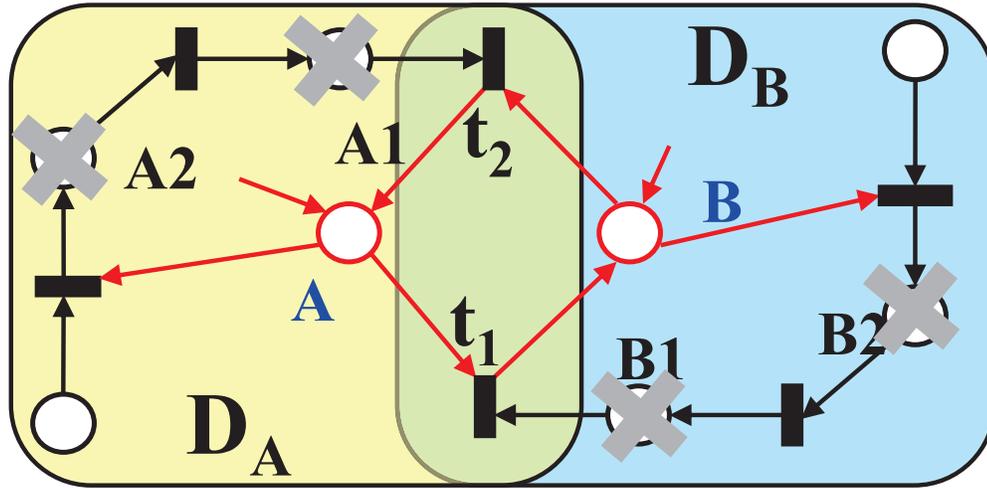


Figura 3.10: Diagrama en donde un conjunto de lugares pertenecientes a un cerrojo se convierten en no-esenciales.

D_r por todos los cerrojos mínimos D_x , tal que, existe un arco (x, r) en el grafo G en el cual estamos calculando $K_G(r)$. De igual forma, $K_G(r)$ contiene aquellos lugares que el propio cerrojo D_r se debe podar a sí mismo teniendo en cuenta el lema 10. En otras palabras, $K_G(r)$ consiste en los lugares que deben ser eliminados de D_r cuando construimos el cerrojo resultante, al menos, a partir de la unión de D_r y todos los cerrojos D_x . Obsérvese que el algoritmo usa como semilla los pares que etiquetan los arcos, (x, r) , porque en estos pares podemos encontrar los lugares proceso candidatos que pueden ser eliminados por un lugar recurso. Después de la detección del primer lugar a ser podado es posible que un nuevo conjunto de lugares se conviertan en no-esenciales y entonces deban ser podados. Por ejemplo, si hemos eliminado el lugar p de D_r , un nuevo candidato a ser podado es un lugar $q \in \bullet(\bullet p) \cap D_r \cap P_s$ si p es el único lugar de salida en D_r de las transiciones $\bullet p$. Esto lo podemos observar de una manera gráfica en la figura 3.10. En esta figura tenemos dos cerrojos mínimos $D_A, D_B \in \mathcal{D}^1$. Éstos dos cerrojos comparten las transiciones t_1 y t_2 y para calcular $K_G(A)$ observamos que el cerrojo D_B poda del cerrojo D_A el lugar $A1$ (usado como semilla por la transición t_2) el cual se convierte en no-esencial para el cerrojo resultante de la unión de los cerrojos D_A y D_B . Observe que la acción de eliminar el lugar $A1$ hace no-esencial al lugar $A2$ el cual es también incorporado al conjunto $K_G(A)$, obteniendo $K_G(A) = \{A1, A2\}$. De manera similar se calcula $K_G(B)$ obteniendo que este es igual a $\{B1, B2\}$.

Adicionalmente, en el algoritmo, a través del conjunto $P_{parcial}$ se consideran los lugares

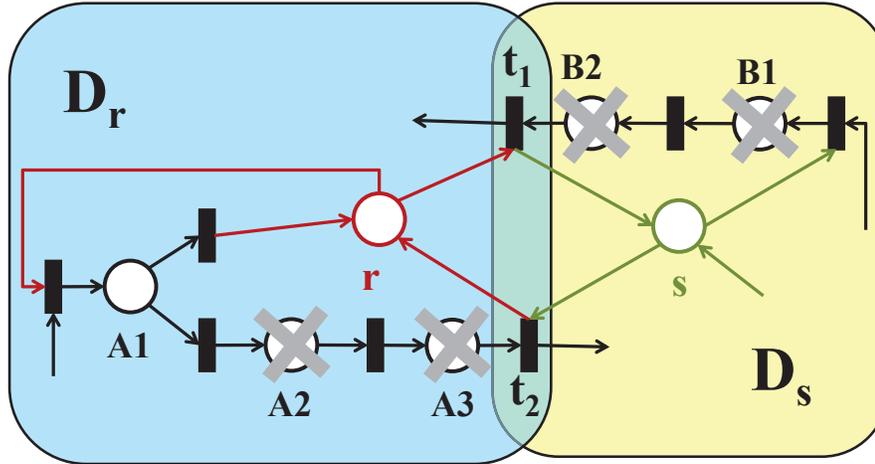


Figura 3.11: El lugar $A1$ perteneciente al cerrojo mínimo D_r contenido en el conjunto $P_{parcial}$ no puede ser podado.

que son candidatos a ser eliminados pero que no pueden ser eliminados porque tienen una transición de salida que es de entrada de un lugar que no puede ser podado del cerrojo D_r y en esa transición el lugar es el único de entrada perteneciente al cerrojo.

En la figura 3.11 ilustramos de una manera gráfica el trabajo del Algoritmo 1 y el significado de la función K_G . En la figura tenemos dos cerrojos mínimos $D_r, D_s \in \mathcal{D}^1$. Éstos dos cerrojos comparten las transiciones t_1 y t_2 y para calcular $K_G(r)$ observamos que el cerrojo D_s poda del cerrojo D_r el lugar $A3$ (usado como semilla por la transición t_2) el cual se convierte en no-esencial para el cerrojo resultante de la unión de los cerrojos D_r y D_s . Observe que la acción de eliminar el lugar $A3$ hace no-esencial al lugar $A2$ el cual es también incorporado al conjunto $K_G(r)$. El lugar $A1$ es agregado al conjunto $P_{parcial}$ en el algoritmo 1 pero finalmente este no es agregado a $K_G(r)$ porque éste es esencial en el cerrojo $D_r \cup D_s$ debido a su transición de salida que lo conecta con el lugar r . De una manera similar puede ser calculado $K_G(s)$ obteniendo que este es igual a $\{B1, B2\}$. Se puede observar otro ejemplo similar al antes explicado en la red de la figura 3.8 en donde los valores para la función $K_G(r)$ obtenidas para el grafo de la figura 3.9 son resumidos en el cuadro 3.1. En este cuadro se puede observar que el lugar a perteneciente al conjunto $P_{parcial}$ no puede ser eliminado del cerrojo mínimo D_R .

La figura 3.12 muestra una red la cual posee dos cerrojos mínimos conteniendo cada uno un único recurso: $D_R = \{R, p1, p2, p3, p4, p6\}$ y $D_S = \{S, p4, p5\}$. En esta red el lugar $p1$ es esencial para el cerrojo D_R . Sin embargo, al realizar la operación de unión entre los cerrojos D_R y D_S , $D_{RS} = D_R \cup D_S$, el lugar $p1$ se convierte en no-esencial para el cerrojo

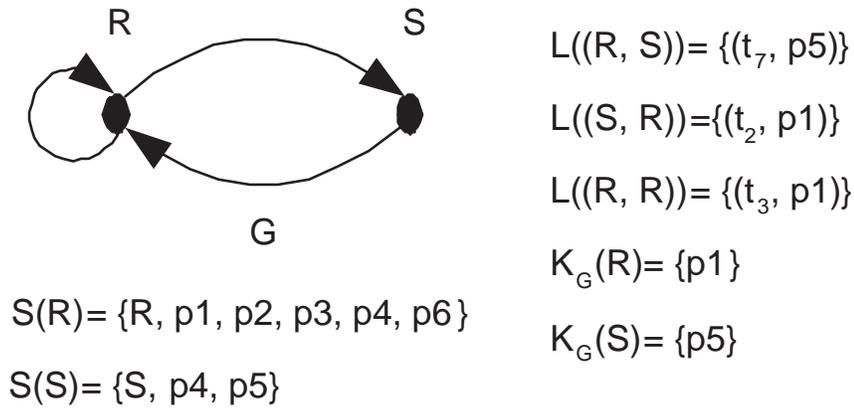


Figura 3.13: Grafo de Poda, G , de la figura 3.12 en donde se puede observar que la información del bucle en el vértice R es necesaria para poder calcular el cerrojo mínimo $D_{RS} = \{R, S, p2, p3, p4, p6\}$ resultante de la unión de los cerrojos mínimos de un recurso D_R y D_S , $D_{RS} = D_R \cup D_S$.

r	T_F	$P_{parcial}$	$K_G(r)$	T_{nuevo}
R	$\{t_3, t_8\}$	$\{a\}$	$\{e\}$	\emptyset
S	$\{t_7\}$	\emptyset	$\{f\}$	\emptyset
T	\emptyset	\emptyset	\emptyset	\emptyset

Cuadro 3.1: Valores de la función $K_G(r)$ para el Grafo de Poda, G , de la figura 3.9.

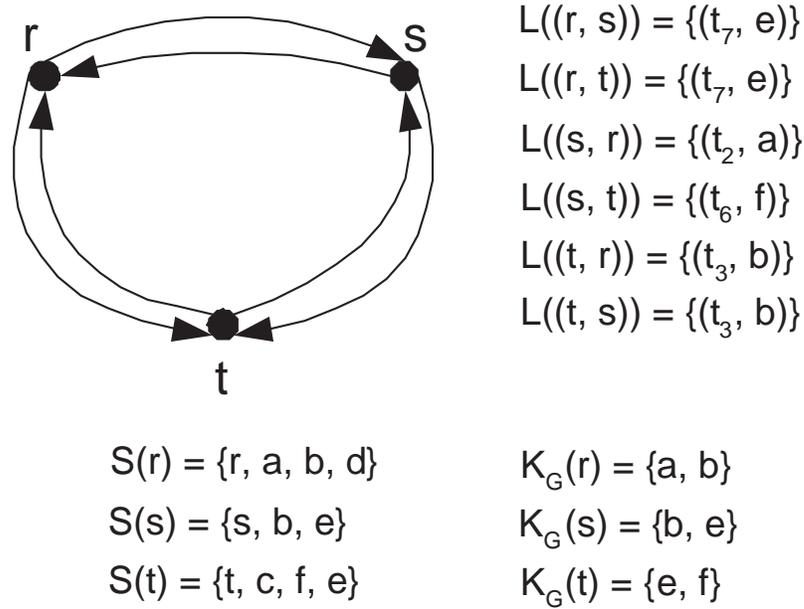


Figura 3.14: El Grafo de Poda, G , de la red de la figura 2.5.

uniones de conjuntos de cerrojos que contengan un solo recurso. Esta propiedad viene a decir que recursos que en el grafo de poda no tengan arcos de entrada no pueden formar parte de un cerrojo mínimo que contenga todos los recursos del grafo de poda.

Lema 11. Sea $G = (V, E)$ el grafo de poda de \mathcal{N} . Si G es acíclico entonces no existe un cerrojo mínimo que contenga los recursos $V \subseteq P_R$.

Demostración. Si G es acíclico entonces existe un vértice r tal que no tiene ningún arco de entrada. Si aplicamos el algoritmo 1 a r en G , resulta que $K_G(r) = \emptyset$ y por lo tanto el resto de recursos de V no consiguen que ningún lugar de $S(r)$ llegue a ser no-esencial y por lo tanto el cerrojo $S(r)$ completo estará contenido en cualquier cerrojo que contenga todos los recursos de V y por tanto no podrá ser mínimo. \square

La siguiente propiedad generaliza la anterior pero para grafos de poda no fuertemente conexos.

Lema 12. Sea $G = (V, E)$ el grafo de poda de \mathcal{N} . Si G no es fuertemente conexo entonces no existe un cerrojo mínimo que contenga los recursos $V \subseteq P_R$.

Demostración. Si el grafo no es fuertemente conexo, se pueden calcular las componentes fuertemente conexas de G y a continuación construir el grafo de condensación de G , que

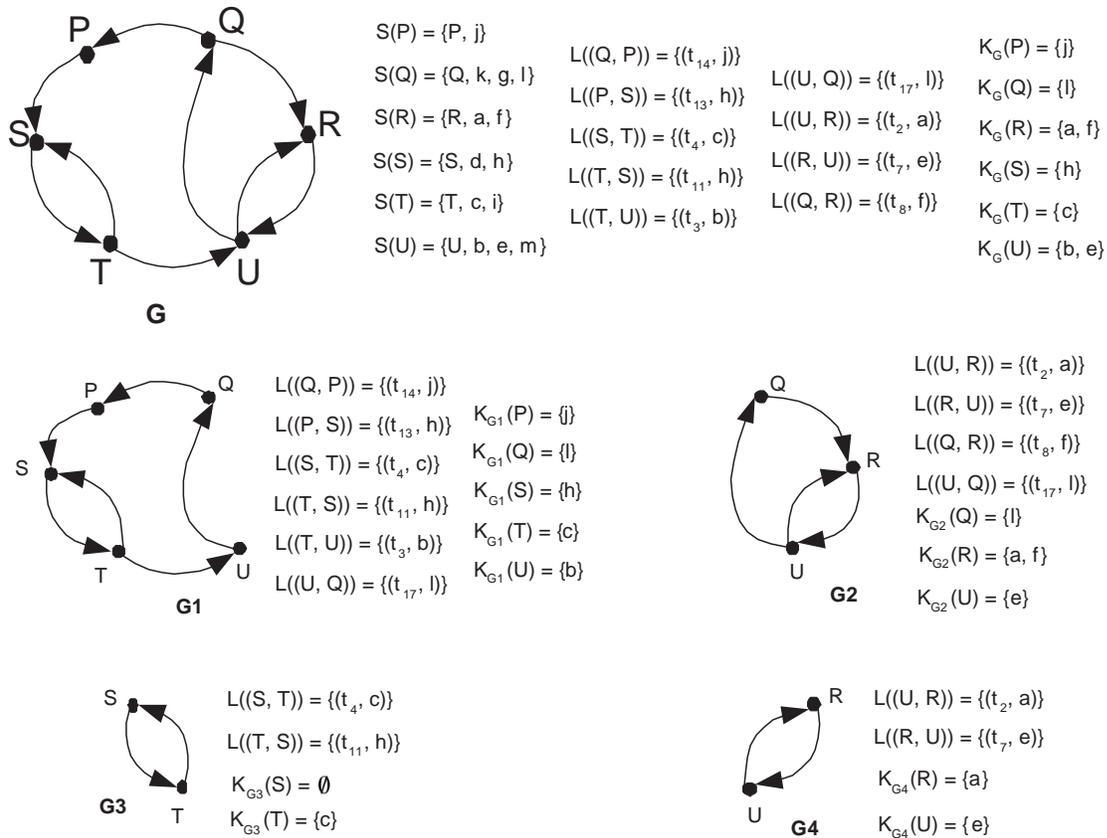


Figura 3.15: El Grafo de Poda G y los Subgrafos $G1, G2, G3$ and $G4$ de la figura 2.2.

llamaremos G' , y en el que cada componente fuertemente conexa es sustituida por un vértice manteniendo los arcos de entrada y salida de la componente fuertemente conexa que conectan con otras componentes fuertemente conexas. Los nuevos vértices tienen asociados los recursos correspondientes a los nodos de la componente fuertemente conexa que generó el vértice y la función S del nuevo vértice es la unión de los cerrojos de los recursos de la componente fuertemente conexa. El grafo G' es un grafo acíclico, por tanto y de acuerdo con el lema 11 no existe un cerrojo mínimo que contenga todos los recursos $V \subseteq P_R$. \square

A partir de los Lemas previos podemos caracterizar la clase de cerrojo resultante de la unión de los cerrojos mínimos contenidos en los vértices de un grafo, en donde, al menos uno de los vértices no posee arcos que incidan positivamente en él. De igual forma, se establece el valor de la función de etiquetado de poda para este vértice en el grafo.

Lema 13. *Sea $G = (V, E)$ el grafo de poda de \mathcal{N} y $K_G(r)$ la función de etiquetado de poda para un recurso $r \in V$. Si en G , $K_G(r) = \emptyset$ para el recurso r entonces el cerrojo resultante de la unión del cerrojo D_r con todos los demás cerrojos del grafo será no-mínimo.*

Demostración. Por la definición de la función de etiquetado de poda, $K_G(r)$ contiene el conjunto de lugares proceso que serán eliminados del cerrojo D_r cuando unimos éste cerrojo con algún otro cerrojo del grafo de poda porque llegan a ser no-esenciales en el cerrojo unión. Por tanto si $K_G(r) = \emptyset$ todos los lugares del cerrojo $S(r)$ son esenciales y el cerrojo completo está contenido con lo cual la unión no será mínima. \square

Basados en la definición de subgrafos podemos establecer la siguiente propiedad relacionada con el valor de la función de etiquetado de poda para cada vértice en los subgrafos.

Lema 14. *Monotonía de $K_G(r)$ en subgrafos que contienen un recurso r . Sea $G = (V, E)$ el grafo de poda de \mathcal{N} y $G' = (V', E')$ un subgrafo de poda G' de G inducido por el conjunto de vértices $V' \subseteq V$. En G' la función de etiquetado de poda para todo recurso $r \in V'$, $K_{G'}(r)$, verifica la siguiente propiedad de monotonía:*

$$K_G(r) \supseteq K_{G'}(r)$$

Demostración. De acuerdo a la definición del grafo de poda la función de etiquetado de poda $K_G(r) \subseteq P_S$ es obtenida por el algoritmo 1 utilizando la información contenida en las etiquetas de los arcos $(x, r) \in E$. Por lo tanto, si G' es un subgrafo el número de arcos será igual o menor con lo que el algoritmo 1 generará un conjunto menor o igual que $K_G(r)$. \square

3.5. La Relación Extendida de Poda en redes S^4PR

La relación de poda definida en apartados anteriores ha sido concebida para conjuntos de cerrojos que son mínimos. La razón de ello es, que como se verá en el capítulo siguiente, la utilización de cerrojos mínimos en el grafo de poda suministra la máxima información para desestimar muchos subconjuntos de recursos que en ningún caso podrán pertenecer simultáneamente a un cerrojo mínimo. Esta detección precoz se realiza sobre el valor de la función K_G de cada nodo. Si el K_G de un nodo, cuyo cerrojo asociado es mínimo, es igual a vacío veremos que un cerrojo obtenido por unión de los cerrojos de este grafo no puede ser mínimo porque contendrá el cerrojo mínimo completo cuyo K_G es vacío. Por el contrario, si tenemos un cerrojo no mínimo asociado a un vértice su K_G correspondiente deberá ser forzosamente no vacío por lo que perdemos la capacidad de detección de uniones improductivas asociada a los valores de K_G .

No obstante, la relación de poda definida previamente, puede ser extendida para ser aplicada a cualquier conjunto de cerrojos. Aunque por el cálculo de, por ejemplo, el conjunto de cerrojos mínimos no es muy útil, si que tiene utilidad para comprender la estructura y relaciones del conjunto de cerrojos mínimos, por ejemplo.

Definición 10. La Relación de Poda Extendida

Sea $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red S^4PR y sean D_r, D_s dos cerrojos de \mathcal{N} tal que $v_r = D_r \cap P_R$ y $v_s = D_s \cap P_R$. El cerrojo D_r está en relación de poda con el cerrojo D_s , o simplemente el cerrojo D_r poda al cerrojo D_s , si y solo si,

- caso $v_r \neq v_s$) $U_r = v_r^\bullet \cap T_{rs} \cap (\bullet T_{rs} \cap D_s \cap P_S)^\bullet \neq \emptyset$ con $T_{rs} = D_r^\bullet \cap D_s^\bullet$.
- caso $v_r = v_s$) $U_r = v_r^\bullet \cap (D_r \cap P_S)^\bullet \neq \emptyset$ ó $\exists p \in D_r \cap P_S$ tal que $p \notin \mathcal{H}_{v_r}$.

A continuación se analizan unos cuantos ejemplos en los que un par de cerrojos de la red de la figura 3.16 están en relación de poda.

Ejemplo 1.

Considérese el cerrojo no mínimo $D_r = \{b, h, r, c\}$. Este cerrojo está en relación de poda consigo mismo. Efectivamente, de acuerdo con la definición 10.b, $v_r = D_r \cap P_R = \{r\}$ y $c \in D_r \cap P_S$, cumpliéndose que $c \notin \mathcal{H}_{v_r}$. En otras palabras, el lugar c es un lugar no esencial para un cerrojo mínimo que contenga los recursos v_r ya que para que un lugar proceso pertenezca a un cerrojo, éste debe ser un lugar usuario (holder) de alguno de los recursos contenidos en el cerrojo. Esta última condición necesaria no es más que una forma alternativa de interpretar el lema 5 presentado en el capítulo anterior.

Ejemplo 2.

Considérense los dos cerrojos mínimos $D_1 = \{r, s, c, h\}$ y $D_2 = \{s, t, g, d\}$. Estos dos

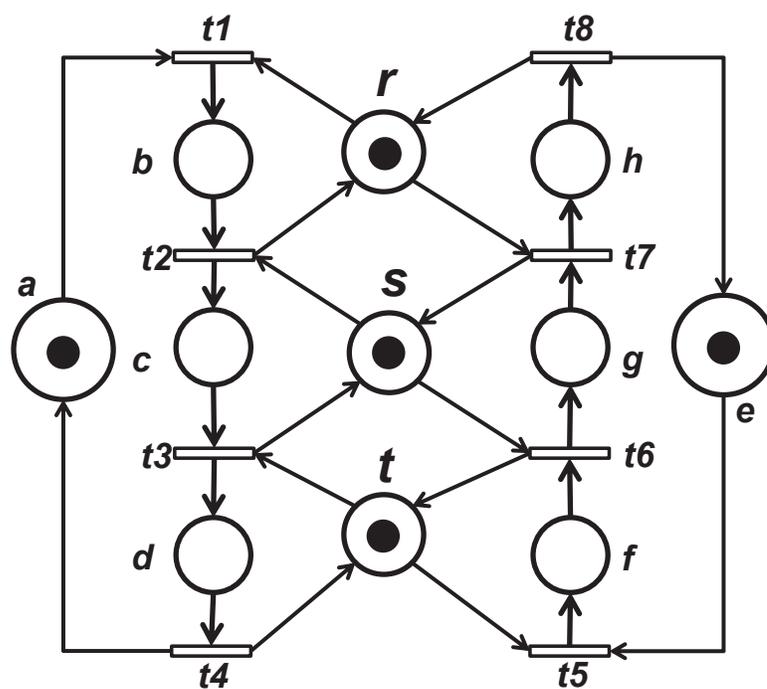


Figura 3.16: Una red de la clase S^3PR .

cerrojos están en una relación de poda. Efectivamente, $T_{12} = D_1^\bullet \cap D_2^\bullet = \{t2, t3, t6, t7\}$, $v_1 = \{r, s\}$ y $v_2 = \{s, t\}$. Además,

$$\begin{aligned}
 U_1 &= v_1^\bullet \cap T_{12} \cap (\bullet T_{12} \cap D_2 \cap P_S)^\bullet \\
 &= \{t1, t2, t6, t7\} \cap \{t2, t3, t6, t7\} \cap (\{b, s, c, t, f, s, g, r\} \cap \{s, t, g, d\} \cap P_S)^\bullet \\
 &= \{t2, t6, t7\} \cap \{g\}^\bullet \\
 &= \{t2, t6, t7\} \cap \{t7\} \\
 &= \{t7\} \neq \emptyset
 \end{aligned}$$

Es decir, el cerrojo D_1 poda al cerrojo D_2 alrededor de la transición $t7 \in U_1$ y el lugar proceso que poda es precisamente el lugar $g \in \bullet t7$. También el cerrojo D_2 poda al cerrojo D_1 . Efectivamente,

$$\begin{aligned}
 U_2 &= v_2^\bullet \cap T_{12} \cap (\bullet T_{12} \cap D_1 \cap P_S)^\bullet \\
 &= \{t2, t3, t5, t6\} \cap \{t2, t3, t6, t7\} \cap (\{b, s, c, t, f, s, g, r\} \cap \{r, s, c, h\} \cap P_S)^\bullet \\
 &= \{t2, t3, t6\} \cap \{c\}^\bullet \\
 &= \{t2, t3, t6\} \cap \{t3\} \\
 &= \{t3\} \neq \emptyset
 \end{aligned}$$

y al igual que en el caso anterior D_2 poda a D_1 en una transición, $t3 \in U_2$, y existe un lugar podado, $c \in \bullet t3$.

Ejemplo 3.

Considérese el cerrojo $D = \{r, s, t, h, c, d\}$. Este cerrojo está en relación de poda consigo mismo, aunque satisfaciendo una condición diferente a la esgrimida en el ejemplo 1. En efecto, ahora todos los lugares proceso del cerrojo D , los lugares h, c y d , son lugares usuario (holders) de alguno de los lugares recurso que pertenecen al cerrojo. Por lo tanto, no es de aplicación la condición usada en el ejemplo 1. Sin embargo, atendiendo a la otra condición expresada en la segunda parte de la definición 10, tenemos lo siguiente: $v = D \cap P_R = \{r, s, t\}$ y por lo tanto

$$\begin{aligned}
U &= v^\bullet \cap (D \cap P_S)^\bullet \\
&= \{r, s, t\}^\bullet \cap (\{r, s, t, h, c, d\} \cap P_S)^\bullet \\
&= \{t1, t2, t3, t5, t6, t7\} \cap \{h, c, d\}^\bullet \\
&= \{t1, t2, t3, t5, t6, t7\} \cap \{t3, t4, t8\} \\
&= \{t3\} \neq \emptyset
\end{aligned}$$

Es decir, D se poda a sí mismo debido a la transición $t3 \in U$ y el lugar podado es precisamente el lugar $c \in {}^\bullet t3$.

Ejemplo 4.

Otro ejemplo interesante lo proporciona la red de Petri de la figura ??, cuando se considera el cerrojo $D = \{R, S, T, a, h, d, e\}$ que es un cerrojo no mínimo. En este caso, se observa también que el cerrojo D está en relación de poda consigo mismo ya que como en el ejemplo anterior se verifica: $v = D \cap P_R = \{R, S, T\}$

$$\begin{aligned}
U &= v^\bullet \cap (D \cap P_S)^\bullet \\
&= \{R, S, T\}^\bullet \cap (\{R, S, T, a, h, d, e\} \cap P_S)^\bullet \\
&= \{t1, t3, t4, t5, t8, t9, t10\} \cap \{a, h, d, e\}^\bullet \\
&= \{t1, t3, t4, t5, t8, t9, t10\} \cap \{t2, t3, t11, t6, t7\} \\
&= \{t3\} \neq \emptyset
\end{aligned}$$

D está en relación de poda consigo mismo debido a la transición $t3 \in U$, pero al intentar eliminar al lugar proceso $a \in {}^\bullet t3$ vemos que esto no es posible ya que el lugar a es un lugar esencial para el cerrojo D ya que si intentásemos eliminarlo D dejará de ser un cerrojo ya que R , debido a su transición de entrada $t2$, necesita el lugar a para mantener la propiedad de cerrojo (a es esencial para R en la transición $t2$).

Lo interesante de este ejemplo está en que el cerrojo D es no mínimo pero a través de la relación de poda consigo mismo no es posible descubrir la no minimalidad de D y por tanto caracterizar los lugares que hay que eliminar para hacerlo mínimo. De hecho, para obtener un cerrojo mínimo a partir de D no basta con eliminar lugares proceso. En efecto, en este caso hay que eliminar también lugares recursos dado que en realidad el cerrojo D es la unión de dos cerrojos mínimos $D_R = \{R, a, h\}$ y $D_{ST} = \{S, T, g, d, e\}$ después de haber eliminado el lugar g que es podado por el cerrojo D_R ó si se quiere por el recurso R .

3.6. El Grafo de Poda Extendido

De forma similar a como se ha representado la relación de poda utilizando un grafo en la sección anterior, la relación de poda extendida será representada mediante un grafo al cual denominaremos *Grafo de Poda Extendido*. A continuación se define este grafo.

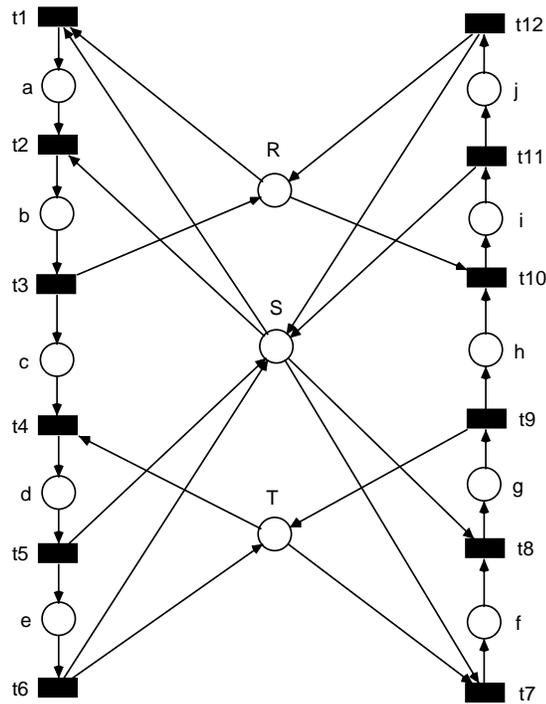
Definición 11. Sea \mathcal{N} una red S^4PR y P_R el conjunto de los lugares recurso. Sea Σ un conjunto de cerrojos de \mathcal{N} tal que no hay dos cerrojos diferentes con el mismo conjunto de recursos. El Grafo de Poda Extendido (EPG) de \mathcal{N} es un grafo $G = (V, E)$ donde,

1. $V = \{D \cap P_R \mid D \in \Sigma\} \subseteq 2^{P_R}$;
2. $E \subseteq V \times V$ y
 - a) Para todo $v_r, v_x \in V$, $v_r \neq v_x$, $(v_r, v_x) \in E$ si, y solo si, $U_r = v_r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap D_x \cap P_s)^\bullet \neq \emptyset$ con $T_{rx} = D_r^\bullet \cap D_x^\bullet$, $v_r = D_r \cap P_R$, $v_x = D_x \cap P_R$, y $D_x, D_r \in \Sigma$.
 - b) Para todo $v_r \in V$, $(v_r, v_r) \in E$ si, y solo si, $U_r = v_r^\bullet \cap (D_r \cap P_s)^\bullet \neq \emptyset$ o $\exists p \in D_r \cap P_s$ tal que $p \notin \mathcal{H}_{v_r}$, con $v_r = D_r \cap P_R$ y $D_r \in \Sigma$.

Dado un grafo de poda extendido $G = (V, E)$, definimos el subgrafo de poda G' de G inducido en el conjunto de vértices $V' \subseteq V$, como el grafo $G' = (V', E \cap (V' \times V'))$. Asociado al grafo de poda o subgrafo definimos las siguientes tres funciones de etiquetado.

Definición 12. Sea \mathcal{N} una red S^4PR y $G = (V, E)$ un grafo de poda extendido o subgrafo de \mathcal{N} .

- a) La función de etiquetado de los Recursos. $S: V \rightarrow \Sigma$, donde para todo $v_r \in V$, $S(v_r) = D_r \in \Sigma$ tal que $D_r \cap P_R = v_r$.
- b) La función de etiquetado de los Arcos. $L: E \rightarrow 2^{T \times P_s}$, donde
 - a) Para todo $(r, x) \in E$, $r \neq x$, $L((r, x)) = \{(t, p) \mid t \in r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap S(x) \cap P_s)^\bullet; T_{rx} = S(r)^\bullet \cap S(x)^\bullet; \{p\} = \bullet t \cap P_s\}$.
 - b) Para todo $(r, r) \in E$, $L((r, r)) = \{(t, p) \mid t \in r^\bullet \cap (S(r) \cap P_s)^\bullet; \{p\} = \bullet t \cap P_s\} \cup \{(t, p) \mid p \in (S(r) \cap P_s) \setminus \mathcal{H}_r; t \in p^\bullet\}$
- c) La función de etiquetado de Poda. $K_G: V \rightarrow 2^{P_s}$, donde para todo $r \in V$, $K_G(r) \subseteq P_s$ calculado por un algoritmo como el algoritmo 1 donde ahora los nodos del grafo son subconjuntos de lugares en vez de un único recurso y las funciones L y S allí utilizadas son las definidas aquí.

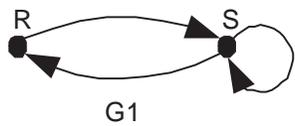
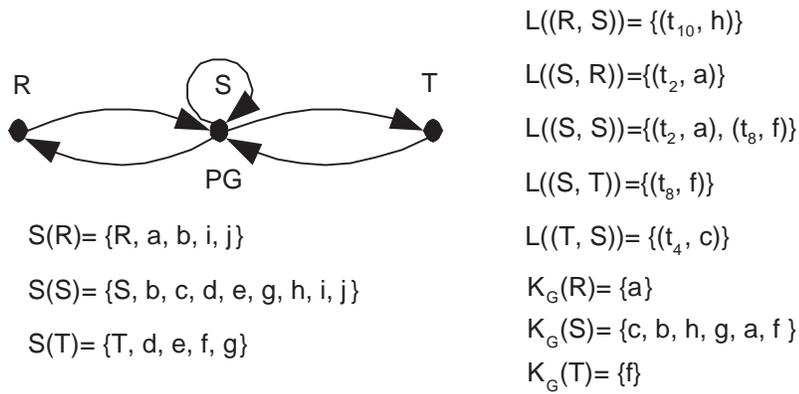
Figura 3.17: Una red S^4PR .

Como un primer ejemplo del grafo de poda extendido consideramos el cerrojo $v = \{r, b, c\}$ de la red en la figura 2.3. Observe que este cerrojo es el soporte del p-semiflujo del recurso r , y éste no es un cerrojo mínimo. El EPG, cuando consideramos el conjunto de cerrojos $\Sigma = \{v\}$, tiene solamente un nodo, denominado $v_r = \{r\}$, y sólo un arco del nodo a sí mismo. Este arco aparece porque el conjunto U_r en la definición 11.2.b es no-vacío y es igual a la transición b^\bullet . El conjunto de lugares a ser eliminados para hacer al cerrojo mínimo es igual al conjunto $K_G(v_r) = \{b\}$.

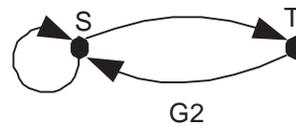
Como un segundo ejemplo, en la figura 3.18 hemos bosquejado el Grafo de Poda Extendido para el conjunto de todos los cerrojos mínimos de la red de la figura 3.17 que contienen por lo menos un recurso.

3.7. Conclusión

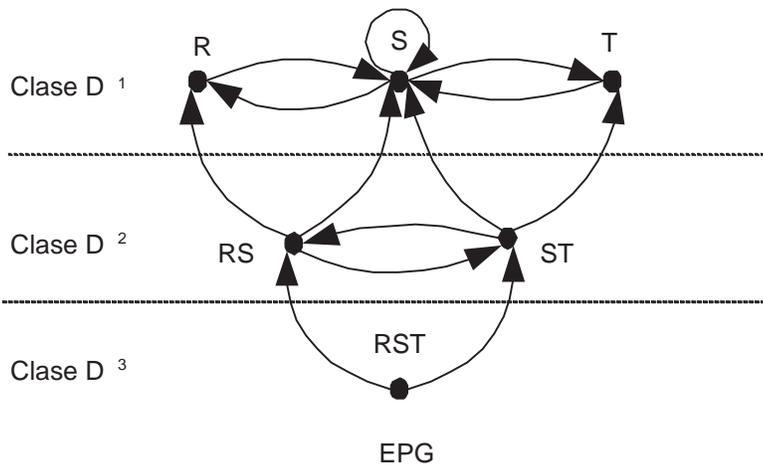
En este capítulo hemos presentado la *Relación de Poda* entre los cerrojos mínimos de un solo recurso. Mediante esta relación podrán cribarse los conjuntos de cerrojos cuya



$L((R, S)) = \{(t_{10}, h)\}$
 $L((S, R)) = \{(t_2, a)\}$
 $L((S, S)) = \{(t_2, a), (t_8, f)\}$
 $K_{G1}(R) = \{a\}$
 $K_{G1}(S) = \{a, f, h, g\}$



$L((S, T)) = \{(t_8, f)\}$
 $L((T, S)) = \{(t_4, c)\}$
 $L((S, S)) = \{(t_2, a), (t_8, f)\}$
 $K_{G2}(T) = \{f\}$
 $K_{G2}(S) = \{a, f, c, b\}$



$L((RS, S)) = \{(t_{10}, h)\}$
 $L((RS, R)) = \{(t_2, a)\}$
 $L((ST, T)) = \{(t_8, f)\}$
 $L((ST, S)) = \{(t_4, c)\}$
 $L((RST, ST)) = \{(t_{10}, h)\}$
 $L((RST, RS)) = \{(t_4, c)\}$
 $L((RS, ST)) = \{(t_{10}, h)\}$
 $L((ST, RS)) = \{(t_4, c)\}$
 $K_G(R) = \{a\}$
 $K_G(S) = \{a, f, c, b, h, g\}$
 $K_G(T) = \{f\}$
 $K_G(RS) = \{c\}$
 $K_G(ST) = \{h\}$
 $K_G(RST) =$

Figura 3.18: El Grafo de Poda PG y Grafo de Poda Extendido EPG de la red de la figura 3.17 utilizando todos los cerrojos mínimos de la red.

unión no puede dar lugar a un cerrojo mínimo como se verá en el capítulo siguiente. Como se ha explicado, el resultado de la operación de unión de los cerrojos mínimos de un solo recurso es un cerrojo debido a que cada uno de los operandos es un cerrojo, pero es no-mínimo.

Se ha presentado la relación de poda sobre un conjunto de cerrojos cada uno de los cuales contiene un recurso mediante el *Grafo de Poda* y sus tres funciones de etiquetado asociadas. Mediante esta herramienta podemos determinar si existen lugares que puedan ser podados de un cerrojo mínimo de un recurso por otro cerrojo mínimo de un recurso. De igual forma nos permite calcular correctamente estos lugares. Adicionalmente, se han desarrollado nuevas propiedades del grafo de poda.

A partir de la relación de poda se define la *Relación de Poda Extendida* la cual constituye una generalización de la anterior. El objetivo de la relación de poda extendida es considerar otros conjuntos de cerrojos que no contengan el mismo conjunto de recursos que los antes considerados en la relación de poda. Entre los nuevos cerrojos a considerar se incluyen los cerrojos no-mínimos y cerrojos conteniendo más de un recurso. La principal diferencia entre ambas relaciones de poda la podemos ubicar en la no-minimalidad de los cerrojos. Se ha representado la relación de poda extendida mediante el *Grafo de Poda Extendido* con sus tres funciones de etiquetado asociados.

Todos estos conceptos y herramientas nos permitirán en el próximo capítulo realizar los cálculos de los cerrojos que contengan determinados recursos y el cálculo de los cerrojos mínimos de una manera más sencilla.

Capítulo 4

Cerrojos en S^4PR y Grafo de Poda

Resumen

El grafo de poda de una red S^4PR puede ser utilizado en diversas aplicaciones como se mostrará en este capítulo. Una de las aplicaciones más generales en la que podemos utilizar el grafo de poda es el cálculo de cerrojos de una red S^4PR que contengan determinados recursos. Este resultado se deriva de la operación de unión de los cerrojos mínimos de un solo recurso explicado de manera informal en el capítulo anterior.

Otra aplicación del grafo de poda la encontramos en el cálculo de los cerrojos mínimos de una red S^4PR . En esta aplicación es necesario utilizar las herramientas de poda explicadas en el capítulo anterior para eliminar entre el conjunto de cerrojos aquellos que son no-mínimos. Para hacer este trabajo de manera más eficiente se ha desarrollado un nuevo algoritmo que evita tener que evaluar muchas uniones no-productivas de cerrojos mínimos de la clase \mathcal{D}^1 . Además, se realiza el análisis del coste computacional para este algoritmo.

Finalmente se presenta cómo se puede determinar si una red S^4PR es viva a partir de un marcado dado utilizando el grafo de poda tal y como se realiza con el Teorema de vivacidad. También se expone un método que identifica los marcados que bloquean estas redes.

4.1. Introducción

En el capítulo anterior se ha planteado la posibilidad de calcular los cerrojos de una red mediante la operación de unión de dos o más cerrojos mínimos conteniendo al menos un

recurso y que el resultado de dicha operación es, por lo general, un cerrojo no-mínimo. De igual forma se han definido la relación de poda y el grafo de poda como una herramienta para calcular todos los cerrojos mínimos de una red.

En este capítulo mostraremos diversas aplicaciones del grafo de poda en una red S^4PR . Iniciaremos la explicación con el cálculo del conjunto de los cerrojos no-mínimos dado a que éste constituye un resultado más general que se deriva de la operación de unión de los cerrojos mínimos de un recurso. Después de este resultado se presentará otro resultado que nos permite cribar entre el conjunto de cerrojos aquellos que son mínimos contribuyendo en este punto con un nuevo algoritmo para el cálculo de los cerrojos mínimos de una red S^4PR basado en el grafo de poda.

En la literatura podemos encontrar diversos trabajos realizados para obtener las familias generatrices de los cerrojos y trampas (o componentes, denominación genérica usada tanto para los cerrojos como para las trampas). El cálculo de las familias generatrices fue presentado por primera vez en [Sif79]. Las componentes son caracterizadas mediante un conjunto de predicados lógicos en las que las variables lógicas utilizadas representan la pertenencia o no de un lugar a la componente. Un método para determinar asignaciones a estas variables lógicas que satisfagan estos predicados lógicos, y por lo tanto caractericen una componente, consiste en la derivación de los predicados lógicos de sistemas de inequaciones (desigualdades) lineales en enteros [Sil85]. Finalmente, mediante la introducción de variables de holgura se logra transformar el sistema de inequaciones en un sistema homogéneo de ecuaciones pudiendo utilizar para el cálculo de las soluciones técnicas estándar del álgebra lineal. Otros métodos algebraicos que calculan estas componentes a través de soluciones a sistemas de ecuaciones o inequaciones y mediante transformaciones de la red los podemos encontrar en [Tou81, AT85, Lau87, ES92, EC91, ECS93].

Por otro lado, existen otros métodos que calculan los cerrojos mínimos pero no mediante el cálculo de las familias generatrices. Entre ellos podemos destacar el trabajo realizado en [BL89] el cual presenta una caracterización de los cerrojos mínimos basada en la teoría de grafos. Otras aproximaciones que utilizan los grafos para el cálculo de los cerrojos mínimos son [YTW98, JPH99, WLJZ09], en donde el último es utilizado para calcular los cerrojos mínimos para redes $L - S^3PR$. En [CX97] se desarrolla un método basado en programación matemática y programación entera mixta para redes de libre elección asimétricas y grafos marcados aumentados, otros trabajos que se basan en este método para obtener los cerrojos mínimos son: [HJXC01, LL07, Cha09].

La dificultad inherente de todos estos algoritmos reside en el número de componentes que tienen que calcular que puede resultar ser un número combinatorio con respecto al número de lugares de la red. Esto hace que el cálculo, incluso de un función generatriz, consuma un tiempo prohibitivo ya que uno de los factores es el número de objetos. En el

caso de las S^4PR , como se ha mostrado en los capítulos precedentes, aunque el número, por ejemplo, de cerrojos mínimos puede ser menor que otras clases de redes, no debemos olvidar que la cota superior del cardinal de este conjunto es del orden de $2^{|P_R|}$ donde P_R es el número de lugares recurso de la red.

Un segundo aspecto importante de la dificultad de muchos de estos algoritmos reside en sus necesidades de memoria. Los métodos generales, que pueden ser aplicados a cualquier clase de red, más extendidos en los paquetes de software de análisis de redes de Petri son métodos como los de [Tou81, Lau87, ES92, EC91, ECS93]. Estos métodos son los que deberían utilizarse para el caso de las redes S^4PR ya que no existen algoritmos especializados para esta clase de redes, ya que los que existen solo son aplicables a subclases de las S^4PR . Todos ellos están basados en mayor o menor medida en los algoritmos de Fourier-Motzkin de eliminación Gaussiana sobre la matriz de incidencia de la red o una obtenida por transformación a partir de ella. Estos algoritmos, que proceden anulando columnas seleccionadas según diversas heurísticas, se basan en la adición de nuevas filas obtenidas por combinación lineal de filas pre-existentes y que resultan de la anulación de la columna en proceso. Es este proceso de adición de nuevas filas por combinación lineal de dos existentes, el que puede hacer que en pasos intermedios del algoritmo las necesidades de memoria crezcan de una forma dramática incluso llegando a la situación de no poder finalizar el algoritmo (bien por falta de memoria – ausencia de recursos – bien por fenómenos de trashing con disco que hagan temporalmente inviable acabar el algoritmo).

A modo de ejemplo para ilustrar este fenómeno presentamos una traza de ejecución del algoritmo de [ECS93]. Esencialmente, este algoritmo calcula las direcciones extremas y del cono convexo $y^t \cdot C_s \leq 0$, donde C_s es esencialmente la matriz de incidencia de la red de Petri en la que se han reequilibrado los pesos de algunos arcos de entrada a las transiciones de la red. La red de Petri de la figura 4.1 tiene una matriz C_s de la forma,

$$C_s = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ R \\ S \end{matrix} & \begin{pmatrix} -1 & 0 & +1 & 0 & 0 & 0 \\ +1 & -2 & 0 & 0 & 0 & 0 \\ 0 & +1 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & +1 \\ 0 & 0 & 0 & +1 & -2 & 0 \\ 0 & 0 & 0 & 0 & +1 & -2 \\ -1 & +1 & 0 & 0 & -2 & +1 \\ 0 & -2 & +1 & -1 & +1 & 0 \end{pmatrix} \end{matrix}$$

Para calcular las direcciones extremas del cono $y^t \cdot C_s \leq 0$ con $y \geq 0$, se pueden introducir variables de holgura y entonces aplicar los algoritmos presentados en [CS91].

La ejecución de este algoritmo se ilustra a continuación sobre la red de Petri de la figura 4.1 transformada su matriz de incidencia como se ha indicado antes para poder aplicar el método presentado en [ECS93].

La presentación se realiza en cuadros, cada uno dedicado a la anulación de una columna de la matriz inicial. En cada cuadro se aprecian dos zonas, la de la izquierda corresponde a la matriz en el estado actual de procesamiento (incluyendo las filas originales y las que se han añadido en pasos anteriores de anulación de columnas), y donde se distinguen dos partes: la superior corresponde al estado antes de empezar la anulación de la columna y debajo de la línea continúa las filas añadidas que corresponden a la anulación de la columna. La zona derecha de cada cuadro corresponde a la combinación lineal de filas originales que ha dado lugar a la fila de la matriz que aparece a su izquierda. Para identificar las filas utilizadas se ha usado el nombre del lugar al que corresponde la fila ($A, B, etc., \dots$) ó el nombre de la variable de holgura introducida ($h1, h2, etc., \dots$). Al pasar de un cuadro a otro, las nuevas filas añadidas en el cuadro origen pasan a ser filas de la matriz del cuadro destino y se eliminan de la matriz las filas utilizadas para obtener las nuevas filas y todas aquellas que no sean de soporte mínimo de la combinación utilizada.

1	2	3	4	5	6	Iteración columna 1
-1	0	+1	0	0	0	A
+1	-2	0	0	0	0	B
0	+1	-2	0	0	0	C
0	0	0	-1	0	+1	D
0	0	0	+1	-2	0	E
0	0	0	0	+1	-2	F
-1	+1	0	0	-2	+1	R
0	-2	+1	-1	+1	0	S
+1	0	0	0	0	0	$h1$
0	+1	0	0	0	0	$h2$
0	0	+1	0	0	0	$h3$
0	0	0	+1	0	0	$h4$
0	0	0	0	+1	0	$h5$
0	0	0	0	0	+1	$h6$
0	-2	+1	0	0	0	$A + B$
0	-1	0	0	-2	+1	$B + R$
0	0	+1	0	0	0	$A + h1$
0	+1	0	0	-2	+1	$R + h1$

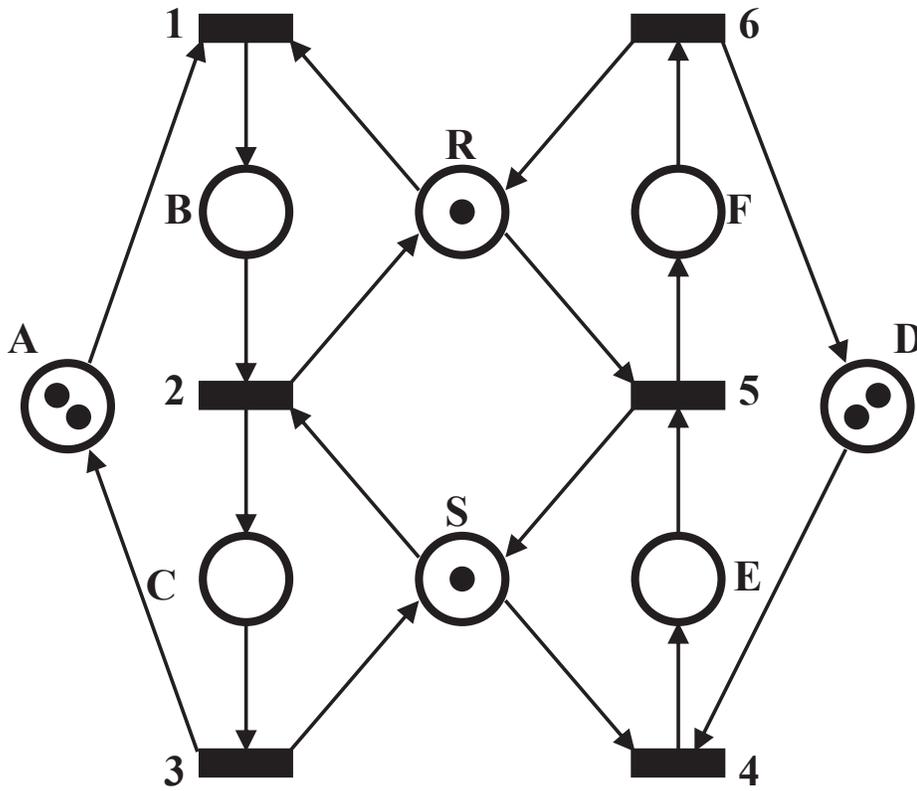


Figura 4.1: Red de Petri S^4PR .

1	2	3	4	5	6	Iteración columna 3
0	+1	-2	0	0	0	C
0	0	0	-1	0	+1	D
0	0	0	+1	-2	0	E
0	0	0	0	+1	-2	F
0	-2	+1	-1	+1	0	S
0	+1	0	0	0	0	$h2$
0	0	+1	0	0	0	$h3$
0	0	0	+1	0	0	$h4$
0	0	0	0	+1	0	$h5$
0	0	0	0	0	+1	$h6$
0	-2	+1	0	0	0	$A+B$
0	-1	0	0	-2	+1	$B+R$
0	0	+1	0	0	0	$A+h1$
0	+1	0	0	-2	+1	$R+h1$
0	+1	0	0	0	0	$2A+C+2h1$
0	-3	0	-2	+2	0	$C+2S$
0	-3	0	0	0	0	$2A+2B+C$
0	+1	0	0	0	0	$C+2h3$

1	2	3	4	5	6	Iteración columna 6
0	0	0	-1	0	+1	D
0	0	0	+1	-2	0	E
0	0	0	0	+1	-2	F
0	+1	0	0	0	0	$h2$
0	0	0	+1	0	0	$h4$
0	0	0	0	+1	0	$h5$
0	0	0	0	0	+1	$h6$
0	-1	0	0	-2	+1	$B+R$
0	+1	0	0	-2	+1	$R+h1$
0	+1	0	0	0	0	$2A+C+2h1$
0	-3	0	-2	+2	0	$C+2S$
0	-3	0	0	0	0	$2A+2B+C$
0	+1	0	0	0	0	$C+2h3$
0	0	0	-2	+1	0	$2D+F$
0	+2	0	0	-3	0	$F+2R+2h1$
0	-2	0	0	-3	0	$2B+F+2R$
0	0	0	0	+1	0	$F+2h6$

1	2	3	4	5	6	Iteración columna 4
0	0	0	+1	-2	0	E
0	+1	0	0	0	0	$h2$
0	0	0	+1	0	0	$h4$
0	0	0	0	+1	0	$h5$
0	+1	0	0	0	0	$2A + C + 2h1$
0	-3	0	-2	+2	0	$C + 2S$
0	-3	0	0	0	0	$2A + 2B + C$
0	+1	0	0	0	0	$C + 2h3$
0	0	0	-2	+1	0	$2D + F$
0	+2	0	0	-3	0	$F + 2R + 2h1$
0	-2	0	0	-3	0	$2B + F + 2R$
0	0	0	0	+1	0	$F + 2h6$
0	-3	0	0	-2	0	$2E + C + 2S$
0	0	0	0	-3	0	$2E + 2D + F$
0	-3	0	0	+2	0	$C + 2S + 2h4$
0	0	0	0	+1	0	$2D + F + 2h4$

1	2	3	4	5	6	Iteración columna 2
0	+1	0	0	0	0	$h2$
0	0	0	0	+1	0	$h5$
0	+1	0	0	0	0	$2A + C + 2h1$
0	-3	0	0	0	0	$2A + 2B + C$
0	+1	0	0	0	0	$C + 2h3$
0	+2	0	0	-3	0	$F + 2R + 2h1$
0	-2	0	0	-3	0	$2B + F + 2R$
0	0	0	0	+1	0	$F + 2h6$
0	-3	0	0	-2	0	$2E + C + 2S$
0	0	0	0	-3	0	$2E + 2D + F$
0	-3	0	0	+2	0	$C + 2S + 2h4$
0	0	0	0	+1	0	$2D + F + 2h4$
0	0	0	0	0	0	$2A + 2B + C + 3h2$
0	0	0	0	-3	0	$2B + F + 2R + 2h2$
0	0	0	0	-2	0	$2E + C + 2S + 3h2$
0	0	0	0	+2	0	$C + 2S + 3h2 + 2h4$
0	0	0	0	0	0	$8A + 2B + 4C + 6h1$
0	0	0	0	-9	0	$4A + 4B + 2C + 3F + 6R + 6h1$
0	0	0	0	-2	0	$6A + 4C + 2E + 2S + 6h1$
0	0	0	0	+2	0	$6A + 4C + 2S + 6h1 + 2h4$

1	2	3	4	5	6	Iteración columna 2 - Continuación
0	0	0	0	0	0	$2A + 2B + 4C + 6h3$
0	0	0	0	-3	0	$2B + 2C + F + 2R + 4h3$
0	0	0	0	-2	0	$4C + 2E + 2S + 6h3$
0	0	0	0	+2	0	$4C + 2S + 6h3 + 2h4$
0	0	0	0	-3	0	$4A + 2B + 2C + F + 2R + 4h1$
0	0	0	0	-6	0	$2B + 2F + 4R + 2h1$
0	0	0	0	-13	0	$2C + 4E + 3F + 6R + 4S + 6h1$
0	0	0	0	-5	0	$2C + 3F + 6R + 4S + 6h1 + 4h4$

1	2	3	4	5	6	Iteración columna 5
0	0	0	0	+1	0	$h5$
0	0	0	0	+1	0	$F + 2h6$
0	0	0	0	-3	0	$2E + 2D + F$
0	0	0	0	+1	0	$2D + F + 2h4$
0	0	0	0	0	0	$2A + 2B + C + 3h2$
0	0	0	0	-3	0	$2B + F + 2R + 2h2$
0	0	0	0	-2	0	$2E + C + 2S + 3h2$
0	0	0	0	+2	0	$C + 2S + 3h2 + 2h4$
0	0	0	0	0	0	$8A + 2B + 4C + 6h1$
0	0	0	0	-9	0	$4A + 4B + 2C + 3F + 6R + 6h1$
0	0	0	0	-2	0	$6A + 4C + 2E + 2S + 6h1$
0	0	0	0	+2	0	$6A + 4C + 2S + 6h1 + 2h4$
0	0	0	0	0	0	$2A + 2B + 4C + 6h3$
0	0	0	0	-3	0	$2B + 2C + F + 2R + 4h3$
0	0	0	0	-2	0	$4C + 2E + 2S + 6h3$
0	0	0	0	+2	0	$4C + 2S + 6h3 + 2h4$
0	0	0	0	-3	0	$4A + 2B + 2C + F + 2R + 4h1$
0	0	0	0	-6	0	$2B + 2F + 4R + 2h1$
0	0	0	0	-13	0	$2C + 4E + 3F + 6R + 4S + 6h1$
0	0	0	0	-5	0	$2C + 3F + 6R + 4S + 6h1 + 4h4$
0	0	0	0	0	0	$2A + 2B + C + 3h2$
0	0	0	0	0	0	$2A + 2B + 4C + 6h3$
0	0	0	0	0	0	$8A + 2B + 4C + 6h1$
0	0	0	0	0	0	$2E + 2D + F + 3h5$
0	0	0	0	0	0	$2E + 2D + 4F + 6h6$
0	0	0	0	0	0	$2E + 8D + 4F + 6h4$

1	2	3	4	5	6	Iteración columna 5 - Continuación
0	0	0	0	0	0	$2B + F + 2R + 2h2 + 3h5$
0	0	0	0	0	0	$2B + 4F + 2R + 2h2 + 6h6$
0	0	0	0	0	0	$2B + 6D + 4F + 2R + 2h2 + 6h4$
0	0	0	0	0	0	$4B + 3C + 2F + 4R + 6S + 13h2 + 6h4$
0	0	0	0	0	0	$2E + C + 2S + 3h2 + 2h5$
0	0	0	0	0	0	$2E + C + 2S + 2F + 3h2 + 4h6$
0	0	0	0	0	0	$2E + 2C + 4S + 6h2 + 2h4$
0	0	0	0	0	0	$2B + 2C + F + 2R + 4h3 + 3h5$
0	0	0	0	0	0	$2B + 2C + 4F + 2R + 4h3 + 6h6$
0	0	0	0	0	0	$2B + 2C + 6D + 4F + 2R + 4h3 + 6h4$
0	0	0	0	0	0	$4B + 16C + 2F + 4R + 6S + 26h3 + 6h4$
0	0	0	0	0	0	$2B + 2F + 2h1 + 6h5$
0	0	0	0	0	0	$2B + 8F + 4R + 2h1 + 12h6$
0	0	0	0	0	0	$2B + 12D + 8F + 4R + 2h1 + 12h4$
0	0	0	0	0	0	$4B + 24C + 4F + 8R + 12S + 4h1 + 36h3 + 12h4$
0	0	0	0	0	0	$2E + 4C + 2S + 6h3 + 2h5$
0	0	0	0	0	0	$2E + 4C + 2S + 2F + 6h3 + 4h6$
0	0	0	0	0	0	$2E + 8C + 4S + 12h3 + 2h4$
0	0	0	0	0	0	$6A + 4C + 2E + 2S + 6h1 + 2h5$
0	0	0	0	0	0	$6A + 4C + 2E + 2F + 2S + 6h1 + 4h6$
0	0	0	0	0	0	$12A + 8C + 2E + 4S + 12h1 + 2h4$
0	0	0	0	0	0	$4E + 2C + 4S + 3F + 6R + 6h1 + 13h5$
0	0	0	0	0	0	$2C + 4E + 16F + 6R + 4S + 6h1 + 26h6$
0	0	0	0	0	0	$56C + 8E + 6F + 12R + 34S + 88h1 + 26h4$
0	0	0	0	0	0	$2C + 3F + 6R + 4S + 6h1 + 4h4 + 5h5$
0	0	0	0	0	0	$2C + 8F + 6R + 4S + 6h1 + 4h4 + 10h6$
0	0	0	0	0	0	$2C + 10D + 8F + 6R + 4S + 6h1 + 14h4$
0	0	0	0	0	0	$9C + 6F + 12R + 18S + 12h1 + 15h3 + 18h4$
0	0	0	0	0	0	$30A + 24C + 6F + 12R + 18S + 42h1 + 18h4$

Conjunto de Cerrojos

- $\{D, E, F\}$
- $\{B, F, R\}$
- $\{C, E, S\}$
- $\{A, B, C\}$
- $\{C, F, R, S\}$

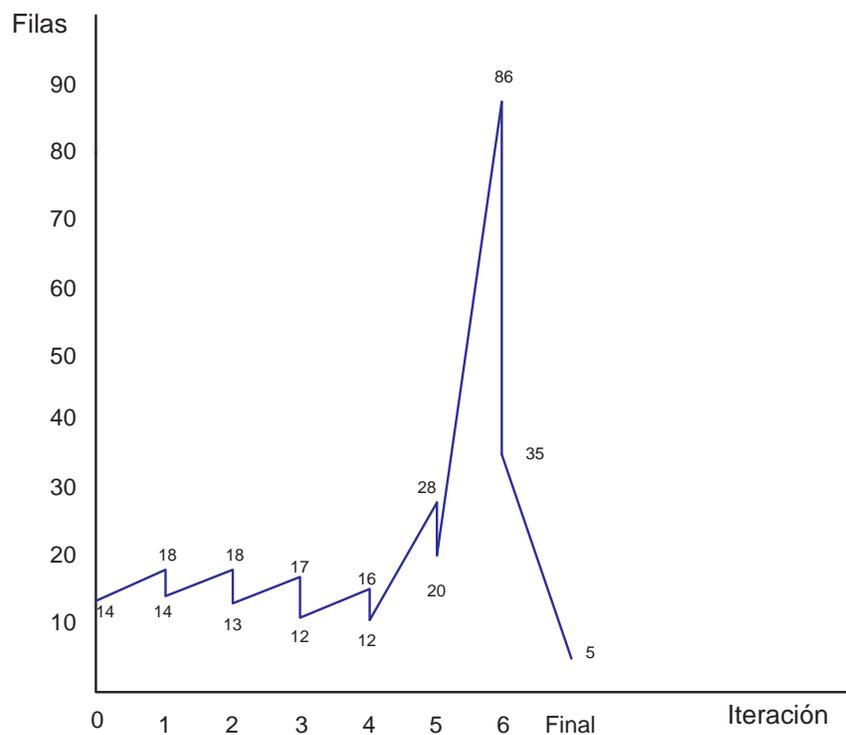


Figura 4.2: Evolución del número de filas durante la ejecución del algoritmo de cálculo de cerrojos mínimos de la red de la figura 4.1 utilizando el método [ECS93].

Como se puede apreciar en el ejemplo anterior, la heurística de selección de columnas permite mantener el número de filas de la matriz dentro de unos valores alrededor de las dimensiones iniciales. No obstante, hacia el final del algoritmo la explosión del número de filas es combinatorio para el pequeño tamaño de la red. Además, como puede apreciarse este enorme número de filas contrasta con el reducido número final de cerrojos que el algoritmo debe devolver.

Este comportamiento de los algoritmos de cálculo de cerrojos que se encuentran implantados en muchas herramientas software de análisis de redes de Petri (GreatSPN [CGH⁺], INA [Sta03], etc.), es muy común. Además, las redes de tipo S^4PR están especialmente mal condicionadas para este tipo de algoritmos, habiendo comprobado para S^4PR relativamente modestas en tamaño que el algoritmo de cálculo no termina, desbordando la cantidad de memoria disponible en las máquinas utilizadas. Ejemplos de estos casos de imposibilidad se han extraído de [RCC10a, RCC10b].

Como veremos en este capítulo, el método de cálculo de los cerrojos mínimos de una red de Petri que presentamos basado en el grafo de poda salva estos problemas de necesidades desmesuradas de memoria al presentar un algoritmo que precisa una cantidad de memoria durante su ejecución del orden del tamaño de la red de entrada.

El resto del capítulo se organiza como sigue. En la sección 4.2 se presenta como se obtienen cerrojos de una red S^4PR utilizando el grafo de poda. En la sección 4.3 se presenta el resultado que caracteriza un cerrojo mínimo en un subgrafo de poda de la red inducido por el conjunto de recursos del cerrojo mínimo y se presenta el nuevo algoritmo para calcular los cerrojos mínimos de una red S^4PR . En la sección 4.4 se presenta el análisis del coste computacional que posee el algoritmo de cálculo de cerrojos mínimos para redes S^4PR . En la sección 4.5 se presenta la relación entre el Teorema de vivacidad y el grafo de poda. Finalmente, la sección 4.6 presenta algunas conclusiones del capítulo.

4.2. Cálculo de Cerrojos a partir del Grafo de Poda

En esta sección usaremos el grafo de poda para calcular cerrojos de una red S^4PR . Primero se presentará un resultado técnico sobre la operación de cálculo de un cerrojo y a continuación se explican los tipos de cerrojos que se obtienen.

El siguiente lema presenta cómo obtener un cerrojo a partir del grafo o subgrafo de poda G de una red. Designaremos este cerrojo asociado como S_G .

Lema 15. *Sea \mathcal{N} una red S^4PR y G un subgrafo inducido por el conjunto de recursos $D_R \neq \emptyset$. El conjunto $S_G = \bigcup_{r \in D_R} S(r) \setminus K_G(r)$ es un cerrojo y $S_G \cap P_R = D_R$.*

Demostración. Es obvio que $\bigcup_{r \in D_R} S(r)$ es un cerrojo porque cada $S(r) \in \mathcal{D}^1$ es también un cerrojo. Además, cada lugar proceso $p \in K_G(r)$ se elimina de $S(r)$ porque p se convierte

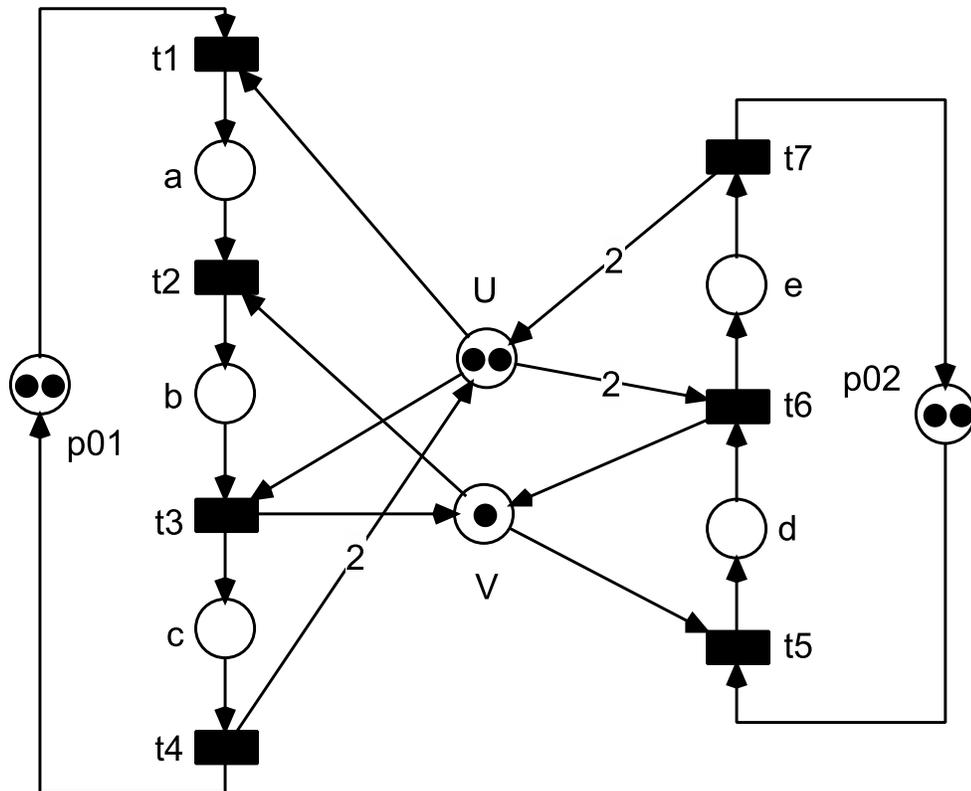


Figura 4.3: Una red S^4PR en donde el cerrojo S_G que contiene a los recursos U y V es no-mínimo porque contiene el cerrojo mínimo D_U .

en no–esencial en S_G . Esto es, la aplicación del algoritmo 1 del capítulo 3 para calcular $K_G(r)$, hace a p no–esencial dado que para todo $t \in p^\bullet : t^\bullet \cap S(r) = \emptyset$; ó $t^\bullet \cap S(r) = \{q\}$ y $q \in K_G(r)$; o existe, al menos, un recurso $x \in D_R$ tal que $x \in \bullet t$. Además, $D_R \subseteq S_G$ porque $K_G(r) \subseteq P_s$ y $r \in S(r)$ para todo $r \in D_R$. \square

Utilizando el resultado anterior en la red mostrada en la figura 4.3 se calcula un cerrojo que contenga los lugares recurso U y V utilizando un grafo de poda extendido G como el que se muestra en la figura 4.4. Este grafo se construye con dos cerrojos conteniendo un solo recurso, $D_U = \{U, a, b, c, e\}$ y $D_V = \{V, b, d\}$, en donde el cerrojo D_V es mínimo mientras que el cerrojo D_U es no–mínimo. Es decir, la función S aplicada a cada uno de los vértices tendrá los siguientes valores: $S(U) = \{U, a, b, c, e\}$ y $S(V) = \{V, b, d\}$. Para obtener el cerrojo mínimo D_U es necesario utilizar la información contenida en el $K_G(U)$ calculada utilizando el bucle ubicado en el vértice U el cual nos indica que el propio cerrojo se poda a sí mismo el conjunto de lugares $K_G(U) = \{a, b\}$. El cerrojo S_G que contiene los recursos U y V se calcula aplicando el Lema anterior de la siguiente forma

$$\begin{aligned} S_G &= S(U) \setminus K_G(U) \cup S(V) \setminus K_G(V) \\ S_G &= \{U, a, b, c, e\} \setminus \{a, b\} \cup \{V, b, d\} \setminus \{b, d\} \\ S_G &= \{U, V, c, e\} \end{aligned}$$

Resulta obvio a partir del valor obtenido para S_G que se trata de un cerrojo no–mínimo ya que contiene el cerrojo mínimo D_U . No obstante, como el lema establece, se trata de un cerrojo aunque no–mínimo. De lo anterior resulta que a través del grafo de poda mediante una operación de unión de los cerrojos que contienen un solo recurso disminuidos por los conjuntos K_G podemos obtener cerrojos que contengan cualquier subconjunto dado de recursos. No obstante, como el ejemplo anterior pone de manifiesto, puede ser que el cerrojo obtenido no sea mínimo al contener, como es el caso, un cerrojo más pequeño. En este caso se trata del cerrojo asociado al nodo U del grafo de poda.

De manera similar, se calcula el cerrojo S_G para la red de la figura 4.5 que contenga a los recursos $R1$ y $R2$. Para ello, construiremos el grafo de poda extendido partiendo de dos cerrojos, que son precisamente los soportes de los p–semiflujos mínimos asociados a los recursos $R1$ y $R2$: $D_{R1} = \{R1, a, b, c, e\}$ y $D_{R2} = \{R2, b, c, d\}$. Notése que el cerrojo D_{R1} es no–mínimo, mientras que el cerrojo D_{R2} es mínimo. La figura 4.6 muestra el grafo de poda extendido G para esta red. Se puede verificar en el grafo de poda G como utilizando la información del bucle ubicado en el vértice $R1$ se elimina el lugar a del cerrojo D_{R1} para obtener el cerrojo mínimo $D_{R1} = \{R1, b, c, e\}$.

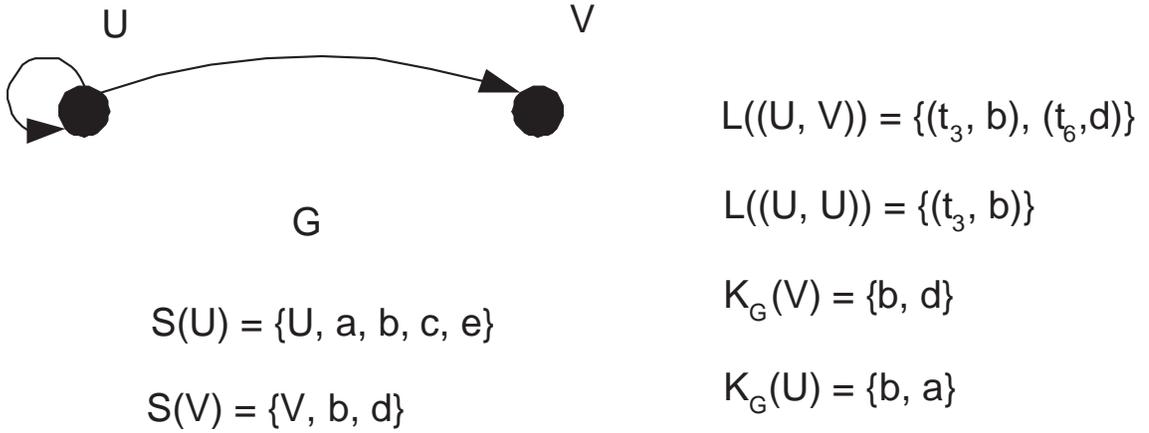


Figura 4.4: Un Grafo de Poda Extendido G de la red de la figura 4.3 en donde se puede observar que el cerrojo D_U se poda a sí mismo el conjunto de lugares $K_G(U) = \{a, b\}$.

$$S_G = S(R1) \setminus K_G(R1) \cup S(R2) \setminus K_G(R2)$$

$$S_G = \{R1, a, b, c, e\} \setminus \{a\} \cup \{R2, b, c, d\} \setminus \{d\}$$

$$S_G = \{R1, R2, b, c, e\}$$

Resultando una vez más un cerrojo que contiene los recursos $R1$ y $R2$ pero es no-mínimo ya que contiene el cerrojo mínimo de un recurso D_{R1} .

De los ejemplos anteriores deducimos que para calcular los cerrojos mínimos de una red S^4PR utilizando un grafo de poda necesitamos hacer como mucho $2^{|P_R|} - 1$ composiciones de cerrojos conteniendo un único recurso y cada una de estas composiciones es la unión de los cerrojos disminuidos en los K_G correspondientes que aparecen en el grafo de poda inducido por el conjunto de recursos considerado. Como hemos visto en los ejemplos anteriores, si los cerrojos que contienen un único recurso que se utilizan para construir el grafo de poda son mínimos entonces no serán necesarios los autobucles que hemos visto en los ejemplos. Aunque como se vio en el ejemplo de la figura 3.6 no siempre es posible eliminar todos los autobucles aunque el cerrojo sea mínimo.

El uso de cerrojos mínimos que contengan sólo un recurso, en la construcción de un grafo de poda a través del cual queremos calcular un cerrojo conteniendo unos recursos determinados, veremos posteriormente que es una estrategia útil para reducir el número de cálculos innecesarios porque nos conducen a cerrojos no mínimos.

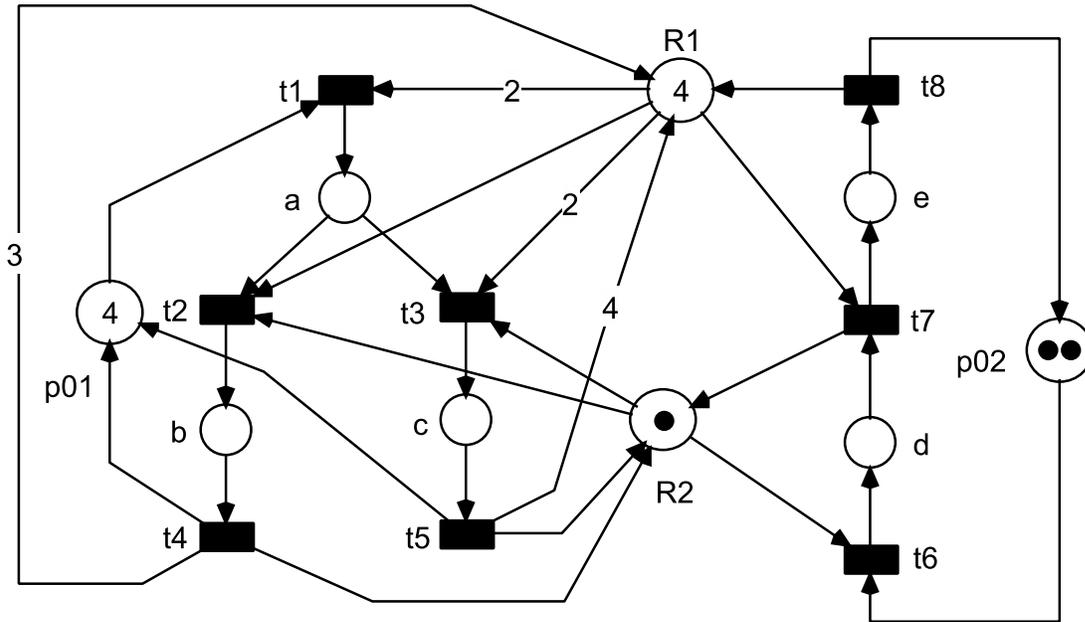


Figura 4.5: Una red S^4PR en donde el cerrojo S_G que contiene a los recursos $R1$ y $R2$ es no-mínimo porque contiene el cerrojo mínimo D_{R1} .

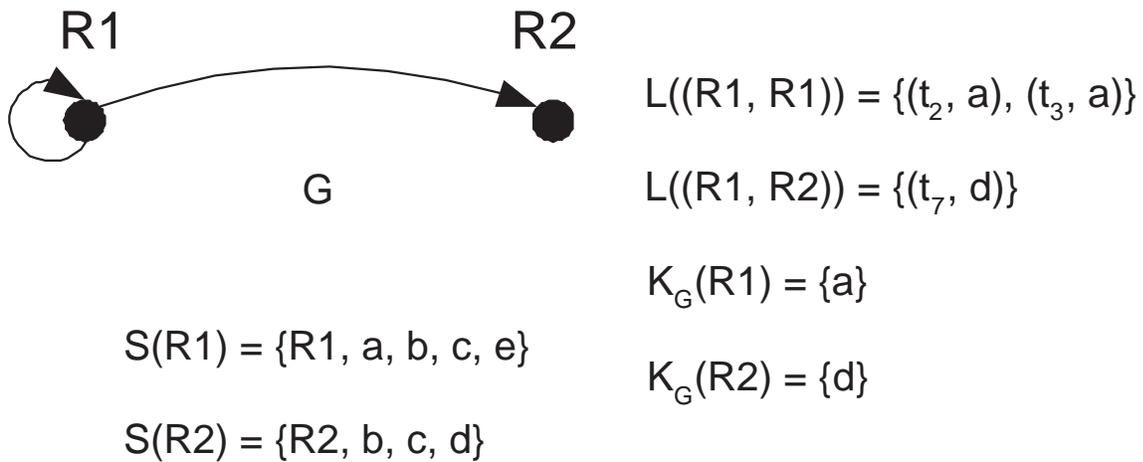


Figura 4.6: Un Grafo de Poda Extendido G de la red de la figura 4.5 en donde se puede observar que el cerrojo D_{R1} se poda a sí mismo el lugar $K_G(R1) = \{a\}$.

Para ver el tipo de ventaja que se obtiene utilizando cerrojos mínimos en lugar de cualquier otro tipo de cerrojo, retomamos el mismo ejemplo anterior pero ahora los cerrojos que vamos a utilizar son: $D_{R1} = \{R1, b, c, e\}$ y $D_{R2} = \{R2, b, c, d\}$; que en este caso ambos son cerrojos mínimos de la red de la figura 4.5. Si procedemos a calcular un grafo de poda con estos vértices se obtiene el grafo y las funciones de etiquetado de la figura 4.7

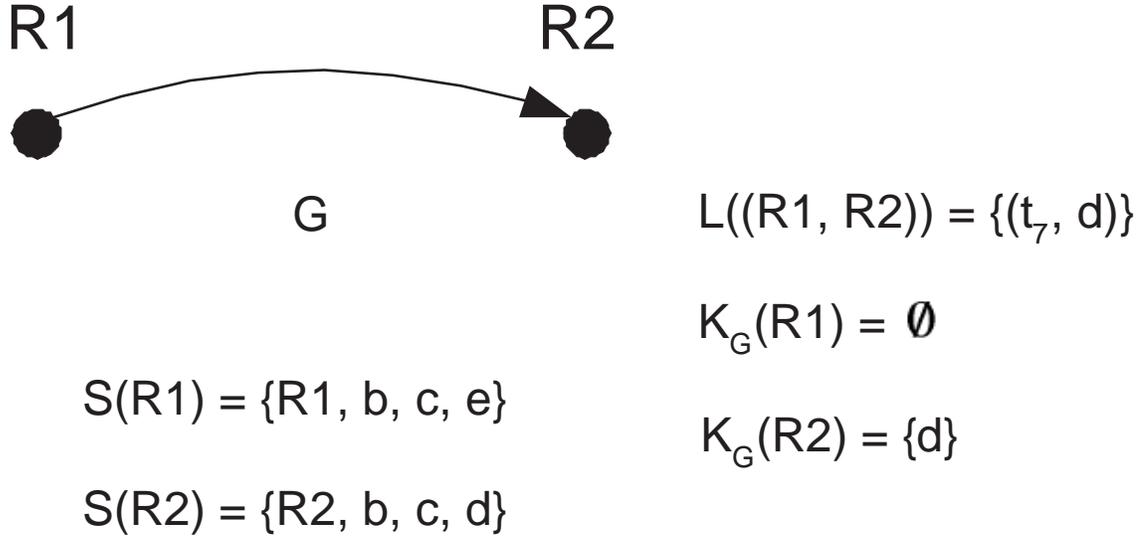


Figura 4.7: Un grafo de poda G de la red de la figura 4.5 en la que se han utilizado dos cerrojos mínimos, cada uno asociado a un recurso, $R1$ y $R2$, respectivamente.

Los cálculos anteriores para calcular el cerrojo S_G conteniendo los recursos $R1$ y $R2$ no dan un resultado diferente al obtenido antes.

$$\begin{aligned}
 S_G &= S(R1) \setminus K_G(R1) \cup S(R2) \setminus K_G(R2) \\
 S_G &= \{R1, b, c, e\} \setminus \emptyset \cup \{R2, b, c, d\} \setminus \{d\} \\
 S_G &= \{R1, R2, b, c, e\}
 \end{aligned}$$

Pero aquí queda patente que el cálculo no hubiese sido necesario al ver que $K_G(R1) = \emptyset$, ya que esto significa que ninguno de los lugares de $S(R1)$ llega a ser no-esencial y por lo tanto, como se aprecia en la expresión anterior el cerrojo completo $S(R1)$ entra en la operación de unión con lo cual el resultado no puede ser mínimo.

Obviamente, la estrategia que se deduce de los párrafos anteriores para calcular el conjunto de cerrojos mínimos es bastante económica en memoria en cuanto que sólo necesitamos almacenar el grafo de poda y los cerrojos que vayamos obteniendo tratando de eliminar cuanto antes aquellos que podamos concluir que son no-mínimos porque contienen alguno ya calculado. En este sentido, la economía de memoria reside en las estructuras preliminares a la obtención de un cerrojo necesarias para construir cerrojos como hemos visto en los algoritmos basados en la anulación de columnas de la matriz de incidencia por combinación de pares de filas. Sin embargo, este número de cerrojos memorizados es innecesariamente grande dado que hay algunas composiciones de cerrojos que directamente tenemos la garantía de que su resultado va a ser no-mínimo. Esto es lo que nos ha ilustrado el último ejemplo, al hacer la unión de los dos cerrojos disminuidos, uno de ellos ha sido disminuido por un conjunto K_G que era igual al conjunto vacío lo cual quiere decir que dicho cerrojo de un solo recurso no ha sido disminuido en ningún lugar y por lo tanto el cerrojo completo ha entrado en la unión de cerrojos y consecuentemente, independientemente de lo que se agregue adicionalmente a la composición, el resultado contendrá estrictamente al cerrojo no disminuido o disminuido por un K_G vacío. Es decir, el resultado no será mínimo.

De la anterior discusión, se concluye que no memorizaremos aquellos resultados de composiciones en las que el K_G de alguno de las componentes sea \emptyset , lo cual se puede detectar a través de un simple test sobre los K_G 's previamente calculados.

4.3. Un Nuevo Algoritmo para calcular los cerrojos mínimos de una red S^4PR

En el capítulo anterior se ha definido el Grafo de Poda PG de una red S^4PR que se utilizará como herramienta central para calcular los cerrojos mínimos de la red. La idea básica del método es construir un cerrojo mínimo D de la red, \mathcal{N} , por la unión de los cerrojos mínimos de \mathcal{D}^1 (cerrojos mínimos de \mathcal{N} conteniendo sólo un recurso) correspondientes a los recursos contenidos en D , y cada uno es disminuído en los lugares de poda especificados por la función K_G .

Como hemos visto en la red de la figura 2.5 el número de cerrojos mínimos de una red S^4PR , conteniendo recursos, puede ser igual a $2^{|R|} - 1$ (el número de subconjuntos no-vacíos de recursos). No obstante, el número de cerrojos mínimos, en la práctica, es mucho menor que este número. Esto significa que deberíamos evaluar muchas uniones no-productivas de cerrojos mínimos de \mathcal{D}^1 .

El siguiente resultado caracteriza un cerrojo mínimo en términos de un subgrafo de poda de la red inducido en el conjunto de recursos del cerrojo mínimo.

Teorema 4. Sea \mathcal{N} una red S^4PR y G el correspondiente grafo de poda. $D \subseteq P$ es un cerrojo mínimo de \mathcal{N} conteniendo el conjunto de recursos $D_R \neq \emptyset$, si, y solo si,

- a) El subgrafo G' de G , inducido por el conjunto D_R es fuertemente conexo.
- b) Para todo $r \in D_R, K_{G'}(r) \neq \emptyset$.
- c) No existe un subgrafo fuertemente conexo G'' de G' , inducido por $D'_R \neq \emptyset, D'_R \subset D_R$ tal que los cerrojos asociados a G'' y G' verifican $S_{G''} \subset S_{G'} = D$

Demostración. \Rightarrow) Si D es un cerrojo mínimo de \mathcal{N} conteniendo el conjunto de recursos $D_R \neq \emptyset$, entonces el grafo de poda PG inducido por D_R, G' , es fuertemente conexo. Probaremos esta afirmación por contradicción. Supongamos que G' no es fuertemente conexo. Las componentes fuertemente conexas (*cfc*) de G' son sus subgrafos fuertemente conexos máximos y tendríamos más de uno. Si cada *cfc* es contraída a un único vértice, el grafo resultante es un grafo acíclico dirigido llamado el *grafo de condensación* de G' . Sea D'_R el conjunto de vértices de una *cfc* de G' tal que esta no tiene *cfc* predecesora en el grafo de condensación (existe al menos una porque el grafo de condensación es acíclico). El subgrafo de poda inducido por D'_R, G'' , es fuertemente conexo y $K_{G''}(r) = K_{G'}(r)$, para todo $r \in D'_R$, porque no hay arcos desde los vértices en $D_R \setminus D'_R$ a los vértices en D'_R , y por lo tanto el algoritmo 1 da los mismos resultados, para todo $r \in D'_R$, en los dos grafos G' y G'' . Por el lema 15, $S_{G''} = \bigcup_{r \in D'_R} S(r) \setminus K_{G''}(r)$ es un cerrojo, y también es $S_{G'} = \bigcup_{r \in D_R} S(r) \setminus K_{G'}(r) = (\bigcup_{r \in D'_R} S(r) \setminus K_{G'}(r)) \cup (\bigcup_{r \in D_R \setminus D'_R} S(r) \setminus K_{G'}(r)) = S_{G''} \cup (\bigcup_{r \in D_R \setminus D'_R} S(r) \setminus K_{G'}(r))$. Esto es, $S_{G''} \subset S_{G'}$ contradiciendo que D es un cerrojo mínimo. Por lo tanto, si D es un cerrojo mínimo G' es fuertemente conexo. Además, si D es mínimo $K_{G'}(r) \neq \emptyset$, para todo $r \in D_R$, puesto que $K_{G'}(r) = \emptyset$, para algún $r \in D_R$ implica que $S(r) \subseteq D$ contradiciendo la minimalidad de D . Finalmente, si existe un subgrafo fuertemente conexo G'' de G' inducido en el conjunto $D'_R \subset D_R$ tal que $D'_R \neq \emptyset$ y $S_{G''} \subset S_{G'} = D$ no puede ser mínimo, contradiciendo que D es mínimo. Por lo tanto, la condición c) del enunciado del teorema es una condición necesaria.

\Leftarrow) Por el lema 15, $D = \bigcup_{r \in D_R} S(r) \setminus K_{G'}(r)$ es un cerrojo y $D_R \subseteq D$. Probaremos por contradicción que D es un cerrojo mínimo. Supongamos que las tres condiciones del teorema son satisfechas y D es no-mínimo. Si D es no-mínimo, existe un cerrojo mínimo $D' \subset D$, y de acuerdo a la parte necesaria de este teorema previamente probada, existe un subgrafo G'' de G inducido por $D'_R = D' \cap P_R \subseteq D_R \neq \emptyset$ que satisface las tres condiciones previas (i.e. G'' es fuertemente conexo y $K_{G''}(r) \neq \emptyset$, para todo $r \in D'_R$). G'' será también un subgrafo de G' . Ahora probaremos que $D' = S_{G''}$, y por lo tanto $S_{G''} \subset D = S_{G'}$, contradiciendo que la condición

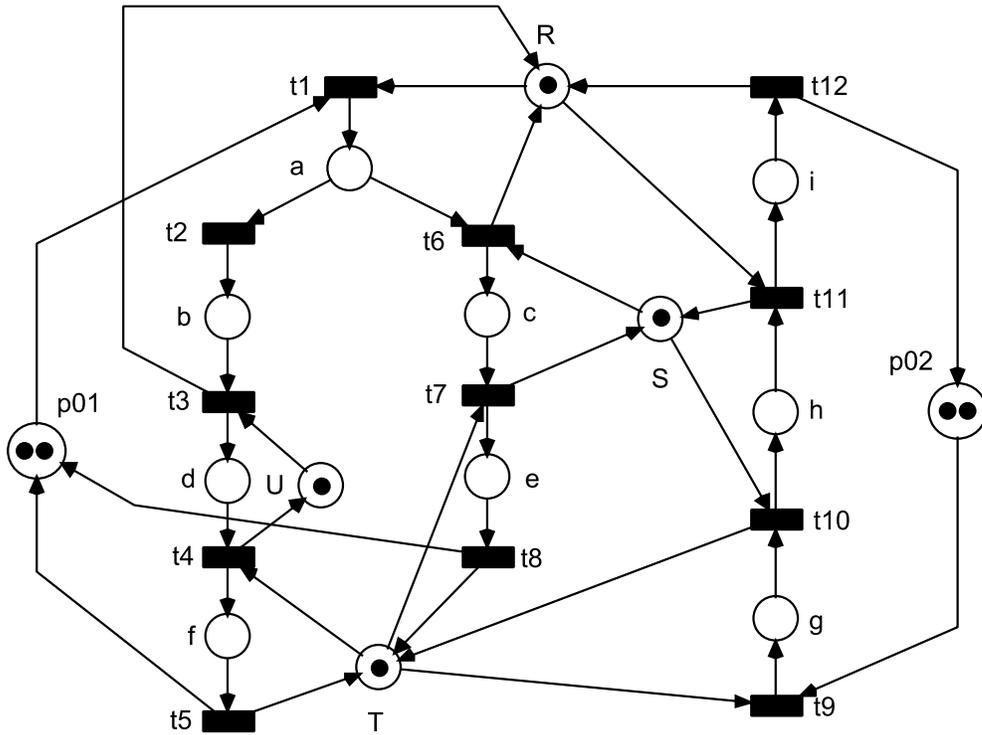


Figura 4.8: Una red S^4PR con dos cerrojos mínimos conteniendo más de un recurso.

tercera del teorema sea cierta para D . En efecto, por el lema 4 $D' \subseteq \bigcup_{r \in D'_R} S(r)$. Además, por la prueba del lema 15, $K_{G'}(r)$ representa todos los lugares que se convierten en no-esenciales en $S(r)$ por el resto de lugares de D' , por consiguiente $D' \subseteq \bigcup_{r \in D'_R} S(r) \setminus K_{G'}(r) = S_{G''}$. Por el lema 15, $S_{G''}$ es un cerrojo conteniendo D'_R recursos, y tomando en cuenta que el número de lugares a podar por el algoritmo 1 es máximo, por lo tanto $D' = S_{G''}$.

□

El siguiente es un algoritmo para calcular los cerrojos mínimos de un red S^4PR basado en la caracterización de los cerrojos mínimos en términos de los grafos de poda presentados en el teorema 4.

Obsérvese que el algoritmo procede en dos fases. La primera es una estrategia para calcular todos los subgrafos fuertemente conexos máximos del grafo de poda de la red, donde todos los vértices tienen una función de etiquetado de poda, K_G , diferente al conjunto vacío (condiciones a) y b) en el teorema 4). La segunda fase remueve todos

Algoritmo 2 Cálculo de los Cerrojos Mínimos de una red S^4PR \mathcal{N} .

Entrada: $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ – La red S^4PR .

\mathcal{D}^1 – Los cerrojos mínimos con solo un recurso.

Salida: \mathcal{D} – Los cerrojos mínimos de \mathcal{N} conteniendo al menos un recurso.

1. **Inicio**
 2. $\mathcal{D} := \mathcal{D}^1$
 3. Calcular el grafo de poda $G = (V, E)$ de \mathcal{N} (definición 8), y las funciones S, L y K_G (definición 9).
 4. Calcular todos los subgrafos fuertemente conexos máximos (componentes fuertemente conexas, *cfcc*) de G , y para cada *cfcc*, G' , calcular las funciones S, L y $K_{G'}$ para formar el grafo de poda G' . Agregar todos los grafos de poda G' con más de un vértice al conjunto g_{nuevo} .
 5. $g_{bueno} := \emptyset$
 6. **Mientras que** $g_{nuevo} \neq \emptyset$ **Hacer**
 7. Extraer $G' = (V', E')$ de g_{nuevo}
 8. **Si** existe $r \in V'$ tal que $K_{G'}(r) = \emptyset$ **Entonces**
 9. Calcular $A := \{r | r \in V' \text{ y } K_{G'}(r) = \emptyset\}$
 10. Calcular el subgrafo de poda de G' inducido por $V' \setminus A$, denominado G'' .
 11. Calcular todos los *cfcc* de G'' , y para cada *cfcc*, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el grafo de poda G''' . Agregar todos los grafos de poda G''' con más de un vértice al conjunto g_{nuevo} .
 12. **Sino**
 13. Agregar G' al conjunto g_{bueno}
 14. **Para** cada $r \in V'$ **Hacer**
 15. Calcular el subgrafo de poda de G' inducido por $V' \setminus \{r\}$, G''
 16. Calcular todos los *cfcc* de G'' , y para cada *cfcc*, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el grafo de poda G''' . Agregar todos los grafos de poda G''' con más de un vértice al conjunto g_{nuevo}
 17. **Fin Para**
 18. **Fin Si**
 19. **Fin Mientras que**
 20. Remover todos los grafos de g_{bueno} , $G = (V, E)$, para los cuales exista otro $G' = (V', E')$ en g_{bueno} que satisfice: (1) $V' \subseteq V$; y (2) $S_{G'} \subset S_G$
 21. **Para** cada grafo $G' = (V', E') \in g_{bueno}$ **Hacer**
 22. Agregar a \mathcal{D} el cerrojo $D = \bigcup_{r \in V'} S(r) \setminus K_{G'}(r) = S_{G'}$
 23. **Fin Para**
 24. **Fin**
-

los subgrafos conteniendo un subgrafo que satisface la condición $c)$ del teorema 4.

Ilustraremos el trabajo del algoritmo 2 utilizando la red de la figura 4.8. Esta red posee cuatro cerrojos mínimos de un recurso $D_R = \{R, a, b, i\}$, $D_S = \{S, c, h\}$, $D_T = \{T, e, f, g\}$ y $D_U = \{U, d\}$ y sólo dos cerrojos mínimos con más de un recurso. El grafo de poda para la red de la figura 4.8 se muestra en la figura 4.9. En la primera fase el algoritmo calcula el grafo fuertemente conexo máximo para la red G y el subgrafo fuertemente conexo $G1$. Observe que en esta fase, el subgrafo formado por los vértices R y S es eliminado debido a que $K_G(R) = \emptyset$. Finalmente, en la segunda fase del algoritmo se verifica que el subgrafo $G1$ no esté contenido en el grafo G . Entonces, aplicando el algoritmo 2 al Grafo de Poda de la figura 4.9 obtenemos los dos cerrojos mínimos para la red de la figura 4.8 a partir del grafo G y del subgrafo $G1$, respectivamente, de la siguiente forma:

$$\begin{aligned} D_{RSTU} &= S(R) \setminus K_G(R) \cup S(S) \setminus K_G(S) \cup S(T) \setminus K_G(T) \cup S(U) \setminus K_G(U) \\ D_{RSTU} &= \{R, a, b, i\} \setminus \{a, b\} \cup \{S, c, h\} \setminus \{c, h\} \cup \{T, e, f, g\} \setminus \{g\} \cup \{U, d\} \setminus \{d\} \\ D_{RSTU} &= \{R, i\} \cup \{S\} \cup \{T, e, f\} \cup \{U\} \\ D_{RSTU} &= \{R, S, T, U, e, f, i\} \end{aligned}$$

$$\begin{aligned} D_{ST} &= S(S) \setminus K_{G1}(S) \cup S(T) \setminus K_{G1}(T) \\ D_{ST} &= \{S, c, h\} \setminus \{c\} \cup \{T, e, f, g\} \setminus \{g\} \\ D_{ST} &= \{S, h\} \cup \{T, e, f\} \\ D_{ST} &= \{S, T, e, f, h\} \end{aligned}$$

Debemos decir que es posible derivar muchas variantes de este algoritmo. Un primer grupo es originado dependiendo del lugar en donde ubiquemos la comparación entre los grafos para eliminar aquellos que son no-mínimos. En el algoritmo hemos presentado esto al final de la fase uno, pero esto puede ser también ser hecho en el paso 13 cuando un nuevo grafo, el cual satisface la condición $a)$ y $b)$ del teorema 4, es almacenado en g_{bueno} . En este momento, podemos comparar el nuevo grafo con los que se han agregado previamente y remover aquellos que contienen al nuevo. Otras variantes dependen de las estructuras de datos usadas para implementar algunos objetos del algoritmo. Por ejemplo, podemos utilizar para implementar g_{bueno} un árbol en lugar de un conjunto. Esto hace más fácil la comparación evitando un gran número de pruebas.

Aplicando el algoritmo 2 al *Grafo de Poda* mostrado en la figura 3.15 los cuatro cerrojos mínimos de la red de la figura 2.2 son obtenidos como sigue: $D = \{R, U, b, f, m\}$ del subgrafo $G4$, $D = \{Q, R, U, b, g, k, m\}$ del subgrafo $G2$, $D = \{P, Q, S, T, U, d, e, g, i, k, m\}$

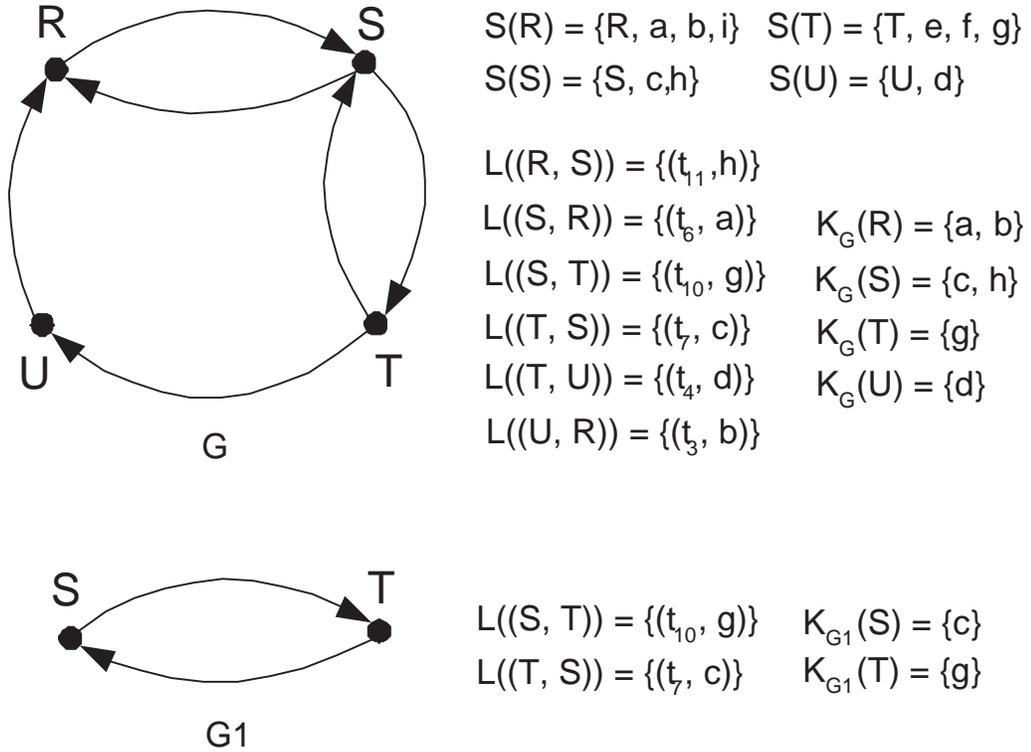


Figura 4.9: El Grafo de Poda G de la red de la figura 4.8 y subgrafo fuertemente conexo G_1 caracterizando D_{ST} .

del subgrafo $G1$ y finalmente, $D = \{P, Q, R, S, T, U, d, g, i, k, m\}$ del grafo de poda PG . Obsérvese, que el grafo $G3$ no pasa la fase 1 porque $K_{G3}(S) = \emptyset$. En la misma forma, los tres cerrojos mínimos para la red de la figura 3.17 son calculados por medio del grafo de poda mostrado en la figura 3.18 como sigue: $D = \{R, S, b, c, d, e, i, j\}$ de $G1$, $D = \{S, T, d, e, g, h, i, j\}$ de $G2$ y $D = \{R, S, T, d, e, i, j\}$ de G .

4.4. Análisis del Coste Computacional del Cálculo de Cerrojos Mínimos mediante el Grafo de Poda

En esta sección se muestran los resultados correspondientes a la terminación del algoritmo 2, desarrollado en la sección anterior, y al análisis del coste computacional del mismo. Este algoritmo utiliza el algoritmo 1 para calcular la función de etiquetado de poda K_G que se explicó en la sección 3.3 del capítulo 3. El siguiente lema establece la terminación del algoritmo 2.

Lema 16. *El algoritmo 2 aplicado a las redes S^4PR , \mathcal{N} , termina y calcula todos los cerrojos mínimos de \mathcal{N} conteniendo al menos un lugar recurso.*

Demostración. El algoritmo termina porque en cada iteración del primer ciclo que inicia en la línea 6, se agrega al conjunto g_{nuevo} los subgrafos fuertemente conexos máximos que contienen al menos un vértice menos que el grafo utilizado en la iteración para calcular estos subgrafos fuertemente conexos máximos. Además, todos los nuevos subgrafos agregados tienen un conjunto de vértices (recursos) que es diferente a aquellos grafos previamente agregados y con al menos dos recursos. Esto significa que el ciclo se ejecutará al menos $2^{|P_R|} - |P_R| - 1$ veces, donde $|P_R|$ es el número de lugares recurso de \mathcal{N} .

El algoritmo calcula todos los cerrojos mínimos conteniendo al menos un recurso porque \mathcal{D}^1 es agregado en el inicio del algoritmo al conjunto final. El primer ciclo calcula todos los subgrafos fuertemente conexos máximos donde todos los vértices tienen una función K_G no-vacía. La tarea final en el paso 20 extrae de este conjunto todos los grafos que contienen al menos uno de esos grafos con las mismas propiedades. Por lo tanto, el conjunto g_{bueno} contendrá sólo aquellos grafos de poda caracterizando los cerrojos mínimos con al menos dos recursos como lo indica el Teorema 4. Esto es, el conjunto final \mathcal{D} contendrá todos los cerrojos mínimos de \mathcal{N} . \square

El siguiente resultado formaliza la complejidad del algoritmo 2 la cual corresponde al tiempo de ejecución del mismo en el peor de los casos. La ejecución del algoritmo 2 está dominada por exactamente el número de cerrojos mínimos conteniendo al menos dos recursos.

Lema 17. *El algoritmo 2 puede encontrar todos los cerrojos mínimos de una red S^4PR con una complejidad en el peor de los casos de $O(B^2(\frac{|P_R|+1}{2}) + B(|P_R|^2 + |P_R| \cdot |P_S| - \frac{1}{2}))$ donde $B = 2^{|P_R|} - |P_R| - 1$.*

Demostración. La complejidad del algoritmo 2 en el peor de los casos corresponde al caso en el cual se tiene un número de cerrojos mínimos igual a la cota superior indicada en el Lema 7, i.e. $|\mathcal{D}| = 2^{|P_R|} - 1$ al final del algoritmo 2. Teniendo en cuenta que todos los cerrojos mínimos que contienen exactamente un recurso son directamente añadidos al conjunto \mathcal{D} en el inicio del algoritmo (debido a que ellos son datos de entrada), el cálculo sólo debe encontrar $B = 2^{|P_R|} - |P_R| - 1$ cerrojos mínimos: aquellos que contienen más de un recurso. El ciclo principal del algoritmo que inicia en la línea 6 será ejecutado exactamente B veces, porque el conjunto $g_{nuevo} \cup g_{bueno}$ no puede contener dos grafos con el mismo conjunto de recursos y todos ellos deben contener más de un recurso. Para garantizar esta última propiedad, en cada iteración es necesario hacer un número de comparaciones entre los conjuntos de vértices de los nuevos grafos con los conjuntos de vértices de los grafos almacenados en $g_{nuevo} \cup g_{bueno}$ lo cual está limitado por $O(|P_R| \cdot \frac{B}{2})$. Adicionalmente, para cada uno de los B cerrojos mínimos se deben calcular los subgrafos de poda a partir del grafo de poda dado de \mathcal{N} con una complejidad de $O(|P_R| \cdot |P_S|)$; y calcular el conjunto de componentes fuertemente conexas el cual para cada grafo con V vértices puede ser hecho en un tiempo de $O(V^2)$. En nuestro caso puede ser limitado por $O(|P_R|^2)$ y por lo tanto para los B cerrojos mínimos se requiere un tiempo en el número de operaciones de estas clases de $O(B(|P_R| \cdot |P_S| + |P_R|^2))$. Finalmente, la última tarea del algoritmo requiere detectar los cerrojos no-mínimos derivados de la comparación de pares de cerrojos obtenidos después del ciclo principal. Estas comparaciones deben ser hechas en el conjunto g_{bueno} que contiene B cerrojos mínimos, por lo tanto, esta tarea debe ser hecha en un tiempo de $O(\frac{B \cdot (B-1)}{2})$. Combinando las tres expresiones previas obtenemos la expresión del enunciado. \square

4.5. Teorema de Vivacidad y Grafo de Poda en Redes S^4PR

La presente sección tiene como objetivo relacionar el grafo de poda con el Teorema de vivacidad presentado en la sección 2.2 del capítulo 2. La idea consiste en, a partir de un marcado dado, determinar el estado de una red S^4PR utilizando el grafo como lo hace el Teorema 3. Obviamente, el estado que deseamos conocer de la red analizada, en cuestión, es si la misma se encuentra en estado de bloqueo o no.

Antes de proceder a desarrollar esta nueva aplicación para el grafo de poda debemos

considerar ciertos conceptos que nos serán útiles para obtener con exactitud el resultado. Los conceptos que se considerarán a continuación los podemos ubicar alrededor de los dos puntos importantes en un Sistema de Asignación de Recursos estos son los procesos y los recursos, como se indicó en el capítulo 1, dado que vamos a trabajar con un grafo en el cual se va a verificar la información de un marcado existente tanto para los lugares proceso como para los recursos.

El primer concepto relaciona los recursos y el marcado que estos poseen y los podemos obtener a partir de trabajos previos [PS89, TCE99, Tri03, Col03, SGG05, GS08]. Para que exista un estado de bloqueo en un grafo debemos tener: una espera circular entre un conjunto de lugares procesos por un conjunto de lugares recursos tal que dichos recursos no puedan satisfacer la solicitud realizada por el proceso solicitante y además existe un conjunto de lugares proceso con un marcado igual a cero pertenecientes a un cerrojo malo como lo indica el Teorema 3 del capítulo 2. Esta espera circular puede ser representada como se realiza en [PS89, SGG05, GS08] mediante un grafo que sea fuertemente conexo, en el cual la condición antes apuntada sobre el marcado insuficiente en los recursos en donde al menos un recurso tiene un marcado inicial mayor a la unidad se convierte en condición suficiente y necesaria para que exista un bloqueo.

El segundo concepto se refiere a los procesos que conducen a bloquear la red, los cuales teniendo en uso al menos un recurso han solicitado un recurso diferente al que están utilizando o más cantidad del recurso que tienen en uso y cuya solicitud no puede ser satisfecha por el recurso en cuestión. Estos lugares han sido definidos para redes S^4PR en [TCE99, Tri03, Col03] como se indica en la definición 6 del capítulo 2 y los podemos ubicar en el grafo de poda utilizando la definición 9.c del capítulo 3 en el conjunto de lugares pertenecientes al K_G del cerrojo mínimo ubicado en un vértice del grafo.

A partir de los dos conceptos antes mencionados podemos indicar que tendremos un bloqueo en la red si tenemos en el grafo de poda $G = (V, E)$ un subgrafo fuertemente conexo $G' = (V', E')$ en donde el $K_{G'}$ del cerrojo mínimo ubicado en cada vértice, $K_{G'}(x); x \in V'$, tenga marcado al menos un lugar p , i.e. $p \in K_{G'}(x) | \mathbf{m}[p] > 0$ y cuyo marcado para los recursos contenidos en los vértices es insuficiente, i.e. $\mathbf{m}[x] < \mathbf{Pre}[x, t]; \forall x \in V'$.

La siguiente notación es necesaria para poder ubicar la transición t que se encuentra deshabilitada por el recurso x en el grafo o subgrafo G . Diremos que el recurso x posee un marcado insuficiente tal que impide el disparo de la transición t si el $K_G(x)$ posee al menos un lugar q con un marcado $\mathbf{m}[q] > 0$ y $\mathbf{m}[x] < \mathbf{Pre}[x, t]$ tal que existe un arco $(x, r) \in E$ que sale del vértice conteniendo al recurso x etiquetado con el par $(t, p) \in L((x, r))$.

Efectivamente, el lugar q o algunos de sus lugares precedentes, $q' \in \bullet(\bullet q) \cap D_x \cap P_s$, pertenecientes al $K_G(x)$ tiene las marcas del recurso x que son solicitadas por el proceso $p \in K_G(r)$ a través de la transición t para avanzar. Para ilustrar lo antes expuesto

utilizaremos como ejemplo la red de la figura 4.10. Esta red posee tres cerrojos mínimos de un recurso $D_A = \{A, p1, p2, p8, p9\}$, $D_B = \{B, p3, p5, p6\}$ y $D_C = \{C, p4, p7\}$ y para un marcado dado igual a $\mathbf{m} = p2 + p3 + p7 + p02$ la red se encuentra en estado de bloqueo, esto se debe a que entre estos tres recursos A, B y C existe una espera circular la cual es representada utilizando el grafo de poda de la figura 4.11. Si observamos en el grafo de poda podemos verificar que los tres lugares $p2, p3$ y $p7$ pertenecen cada uno a un K_G de un cerrojo mínimo de un recurso ubicado en un vértice del grafo. En efecto, $p2 \in K_G(A)$, $p3 \in K_G(B)$ y $p7 \in K_G(C)$. Estos lugares tienen las marcas de los recursos A, B y C que impiden el disparo de las transiciones que se encuentran en las etiquetas de los arcos que salen de dichos vértices del grafo estas son: $p3$ tiene las marcas del recurso B que el lugar $p2$ necesita para tener habilitada por recurso a su transición de salida $t3$; $p2$ tiene las marcas del recurso A que el lugar $p7$ necesita para tener habilitada por recurso a su transición de salida $t9$, y finalmente, $p7$ tiene las marcas del recurso C que el lugar $p3$ necesita para tener habilitada por recurso a su transición de salida $t4$.

Como se puede observar sólo son verificados los marcados de los lugares proceso mientras que el marcado de los lugares reposo no se verifican, esto se debe a que como se ha indicado en el Teorema 3 del capítulo 2 la espera circular se presenta sólo entre un conjunto de lugares proceso y un conjunto de lugares recurso. Por esta razón, de ahora en adelante y dado que las redes S^4PR consideradas tienen un marcado inicial admisible, en las pruebas que se realizan al marcado no se considera el marcado del lugar reposo.

Lo antes explicado caracteriza aquellos estados bloqueados que estando en una espera circular son representados en el grafo de poda mediante subgrafos fuertemente conexos, sin embargo es necesario caracterizar aquellos estados bloqueados en los que no se puede formar un subgrafo fuertemente conexo y en los que se cumple las condiciones que indica el Teorema 3 del capítulo 2: se tiene un conjunto de lugares proceso cuyo marcado es mayor que cero y cuyas transiciones de salida están deshabilitadas por recurso y un conjunto de lugares desmarcados pertenecientes a un cerrojo malo.

Para ilustrar este nuevo caso observemos la red de la figura 4.3, en esta red existe un lugar que pertenece a dos cerrojos de un solo recurso, en efecto, el lugar b pertenece al cerrojo D_U y D_V . Este lugar a su vez pertenece a ambos K_G s en el grafo de poda como lo podemos observar en la figura 4.4 lo cual indica que al marcarse este lugar utiliza a ambos recursos U y V . Además en el grafo de la figura 4.4 podemos observar que existe un vértice conteniendo un recurso r cuyo $\bar{K}_G(r) = S(r) \setminus K_G(r)$ es igual a vacío indicando que todos los lugares proceso del cerrojo pertenecen al K_G del recurso y si éstos se encuentran marcados, o al menos uno de ellos, y además están marcados otros lugares de los restantes K_G s de los otros recursos ubicados en los vértices del grafo tendremos un estado de bloqueo.

Para ilustrar el caso antes descrito tomemos como ejemplo el siguiente marcado $\mathbf{m} =$

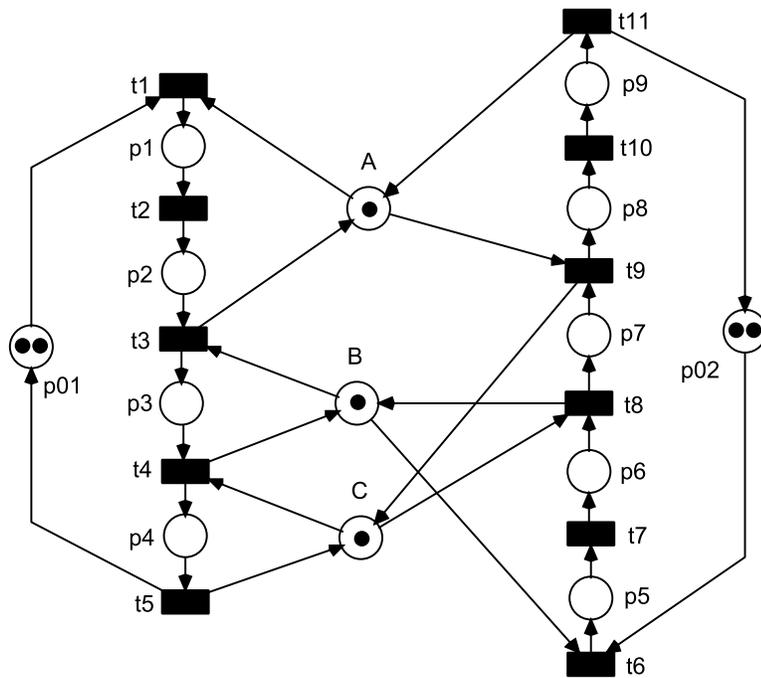
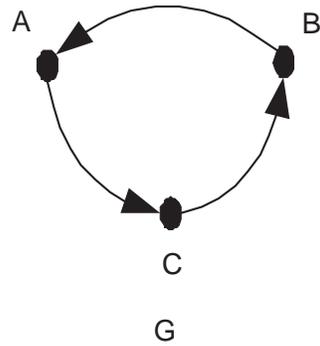


Figura 4.10: Una red S^4PR utilizada para mostrar en su grafo de poda la espera circular que existe entre los recursos A, B y C .



$$L((A,C)) = \{(t_9, p7)\}$$

$$L((C, B)) = \{(t_4, p3), (t_8, p6)\}$$

$$L((B, A)) = \{(t_3, p2)\}$$

$$K_G(A) = \{p1, p2\}$$

$$K_G(B) = \{p3, p5, p6\}$$

$$K_G(C) = \{p7\}$$

$$S(A) = \{A, p1, p2, p8, p9\}$$

$$S(B) = \{B, p3, p5, p6\}$$

$$S(C) = \{C, p4, p7\}$$

Figura 4.11: Grafo de poda G de la red de la figura 4.10.

$a + b + 2p02$, para este marcado la red de la figura 4.3 está bloqueada. Además el $K_G(U)$ tiene marcado a todos los lugares que el mismo contiene, estos son los lugares a y b , donde el lugar a tiene una marca del recurso U y el lugar b el cual también pertenece al $K_G(V)$ tiene en uso una marca del recurso U y una marca del recurso V dejando de esta forma desmarcados ambos recursos lo cual lleva a que la transición de salida del lugar b , $t3$, esté deshabilitada por el recurso U teniendo de esta manera una red bloqueada.

El siguiente ejemplo ilustra el caso en el cual tenemos marcados un lugar de cada K_G del cerrojo mínimo de un recurso ubicado en los vértices del grafo de poda y en donde las transiciones de salida de los mismos se encuentran deshabilitadas por recurso. Utilizando la red de la figura 4.3 y su grafo de poda mostrado en la figura 4.4, para el marcado $\mathbf{m} = a + d + U + p01 + p02$ se observa que el lugar a perteneciente al $K_G(U)$ se encuentra marcado al igual que el lugar d perteneciente al $K_G(V)$ y que el recurso U solicitado por la transición de salida del lugar d , $t6$, está marcado pero no tiene suficientes marcas para habilitar ésta transición, por otro lado el lugar a tampoco puede avanzar debido a que el disparo de su transición de salida, $t2$, es impedido por el recurso V que tiene en uso el proceso d , por lo tanto la red se encuentra bloqueada. Mediante este ejemplo se observa como existe una espera circular entre los lugares proceso a y d aunque la misma no se refleje gráficamente en el grafo de poda.

Un caso similar al antes explicado se presenta en la red de la figura 4.5, en donde, para el marcado $\mathbf{m} = 4a + d + R1 + p02$ la red se encuentra bloqueada. Para este marcado ambos K_G s del grafo de poda de la red mostrado en la figura 4.6 tienen un lugar proceso

marcado. En efecto, el lugar a perteneciente al $K_G(R1)$ se encuentra marcado y el lugar d perteneciente al $K_G(R2)$ también se encuentra marcado, sin embargo, ninguno de los dos lugares pueden avanzar debido a que el disparo de sus transiciones de salida $a^\bullet = \{t2, t3\}$ y $d^\bullet = \{t7\}$ es impedido por los recursos que tienen un marcado insuficiente que les imposibilita el disparo de estas transiciones. De esta forma, se tiene una espera circular entre los procesos a y d que bloquea la red y que no es reflejado gráficamente en el grafo de poda.

Después de la discusión anterior podemos decir que para identificar, como lo hace el Teorema 3, un bloqueo en una red S^4PR a partir de un marcado dado utilizando el grafo de poda debemos distinguir de forma general dos casos:

- a) cuando existe al menos un subgrafo fuertemente conexo en donde cada cerrojo mínimo de un recurso contenido en el vértice tenga marcado al menos un lugar de su K_G y en donde los recursos tienen un marcado insuficiente para habilitar las transiciones de salida de los lugares proceso marcados pertenecientes a dichos K_G s.
- b) cuando todos los cerrojos mínimos de un recurso contenidos en los vértices del grafo de poda tienen al menos un lugar de su K_G marcado y existe al menos un recurso con un marcado insuficiente y además se puede verificar alguna de las dos siguientes condiciones:
 - b.1) si el marcado para todo lugar perteneciente a todos los K_G s negados, $\bar{K}_G(r)$, del grafo de poda es igual a cero.
 - b.2) si existe al menos un cerrojo mínimo de un recurso contenido en el vértice del grafo de poda cuyo K_G negado, $\bar{K}_G(r)$, es igual a vacío.

A partir de lo antes explicado podemos presentar formalmente el algoritmo que realiza la prueba del marcado para determinar si a partir de un marcado dado una red S^4PR se encuentra en estado de bloqueo.

El algoritmo anterior sólo nos permite detectar para un marcado dado el bloqueo en una red pero no nos permite determinar el marcado que inevitablemente nos conducirá a bloquear la red. En este apartado iremos un paso más allá de lo que hace el Teorema 3, con el objetivo de identificar aquellos marcados que inevitablemente conducen a bloquear la red utilizando el grafo de poda como lo hemos realizado antes.

Para ello se utilizará el Teorema 4 el cual caracteriza los cerrojos mínimos de una red \mathcal{N} conteniendo un conjunto de recursos. El marcado en el cual estamos interesados está relacionado con los subgrafos fuertemente conexos $G' = (V', E')$ del grafo de poda en donde el número de vértices sea superior a dos, $|V'| > 2$. Esto se debe a que la detección de un marcado bloqueado para aquellos subgrafos en donde $|V'| = 2$ ya lo realiza el algoritmo 3.

Algoritmo 3 Prueba del marcado para determinar si a partir de un marcado dado una red S^4PR se encuentra en estado de bloqueo.

Entrada: $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ — La red S^4PR .

$G = (V, E)$ — El grafo de poda de \mathcal{N} y las tres funciones S, L y K .

\mathbf{m} — Un marcado dado para la red \mathcal{N} .

Salida: *bloqueo* — El estado de la red.

1. **Inicio**
 2. bloqueo = falso
 3. **Si** G contiene al menos un subgrafo fuertemente conexo **Entonces**
 4. **Para** cada subgrafo fuertemente conexo, G' , **Hacer**
 5. **Si** $\forall x \in V', \exists q | \mathbf{m}[q] > 0$ y $q \in K_{G'}(x)$ y $\mathbf{m}[x] < \mathbf{Pre}[x, t]; t | (t, p) \in L((x, r))$ y $(x, r) \in E'$ **Entonces**
 6. bloqueo = cierto
 7. **Fin Si**
 8. **Fin Para**
 9. **Sino**
 10. **Si** $\forall x \in V, \exists q | \mathbf{m}[q] > 0$ y $q \in K_G(x)$ y existe al menos un $\mathbf{m}[x] < \mathbf{Pre}[x, t]; t | (t, p) \in L((x, r))$ y $(x, r) \in E$ **Entonces**
 11. **Si** $\forall q' \in \bar{K}_G(x)$ se verifica que $\mathbf{m}[q'] = 0$ **Entonces**
 12. bloqueo = cierto
 13. **Sino Si** existe al menos un $\bar{K}_G(x) = \emptyset$ **Entonces**
 14. bloqueo = cierto
 15. **Fin Si**
 16. **Fin Si**
 17. **Fin Si**
 18. **Fin**
-

$\mathbf{m}[p] > 0$	$p \in K_G(r)$	Vértice Marcado
a	$a \in K_G(R)$	R
d	$d \in K_G(T)$	T

Cuadro 4.1: Ubicación de los lugares procesos marcados en los K_G de los vértices del grafo de poda 4.13.

La idea general del método se puede resumir en tres pasos:

1. identificar en los vértices del grafo, los K_G s que tengan lugares proceso marcados y que pertenezcan a al menos un subgrafo fuertemente conexo con más de dos vértices.
2. establecer el recorrido que tienen estos subgrafos.
3. verificar si existen vértices en común para cada recorrido de los vértices con lugares marcados en su K_G . Si esta condición se cumple, verificar para cada vértice en común si
 - la solicitud de marcas del recurso r por otros recursos x en el subgrafo es superior al marcado existente del mismo, esto es, $\mathbf{m}[r] < \sum \mathbf{Pre}[r,t]; \forall t | t \in (t,p) \in L((r,x)); \forall (r,x) \in E'$. Si esta condición se cumple el marcado de los lugares conduce inevitablemente a un bloqueo.

Para ilustrar el método antes presentado utilizaremos la red de la figura 4.12 y su grafo de poda mostrado en la figura 4.13. Esta red posee tres cerrojos mínimos de un recurso $D_R = \{R, a, f\}$, $D_S = \{S, b, e\}$ y $D_T = \{T, c, d\}$, de igual forma posee tres cerrojos mínimos de más de un recurso $D_{RS} = \{R, S, b, f\}$, $D_{ST} = \{S, T, c, e\}$ y $D_{RST} = \{R, S, T, c, f\}$ de éstos sólo se utiliza para este ejemplo el último de ellos debido a que es el único en el cual puede presentarse el caso de un marcado que lleve a bloquear la red. En efecto, el marcado $\mathbf{m} = a + d + S + 9p10 + 9p20$ es un marcado para el cual la red no se encuentra en estado de bloqueo sin embargo es un marcado que inevitablemente conduce a bloquear la red como se verá al aplicar el nuevo método.

Aplicando el paso 1 del método antes presentado se identifican dos vértices en el grafo de poda que tienen marcados sus lugares proceso, los vértices R y T como se resume en la tabla 4.1.

Aplicando el paso 2 se establece sólo un recorrido en un subgrafo fuertemente conexo con más de dos vértices, el subgrafo maximal para el grafo de poda G . En este recorrido ambos vértices marcados comparten el vértice S , paso 3. Obsérvese que este vértice tiene dos arcos de salida (S, R) y (S, T) con solicitud de marcas por los lugares marcados a y

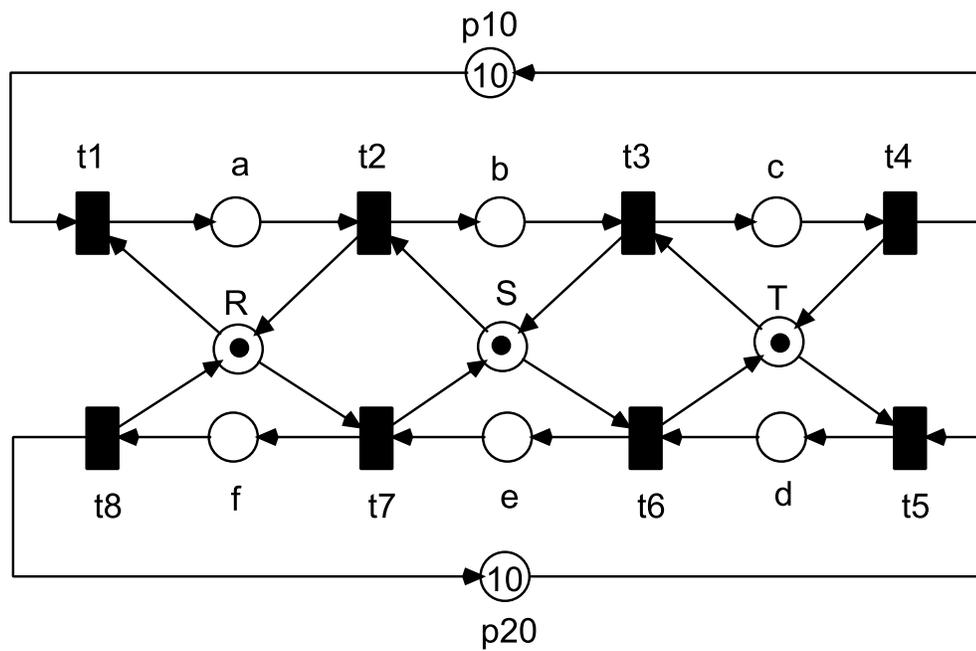
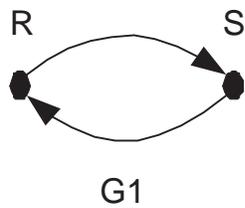
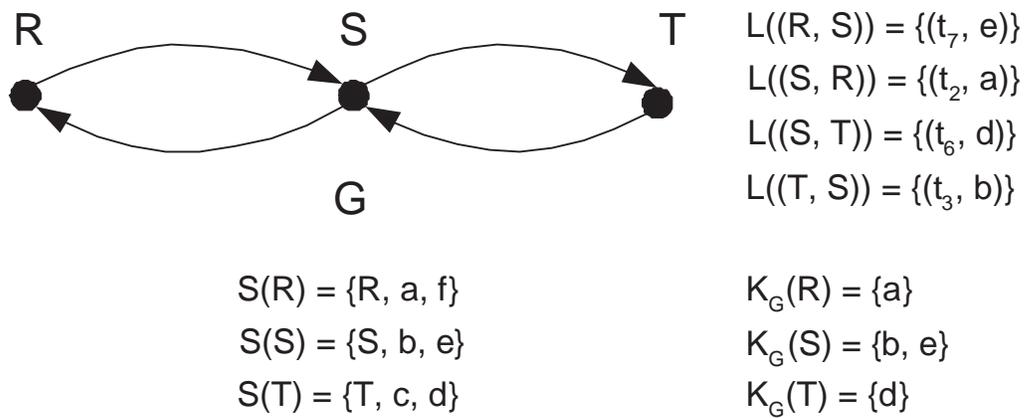
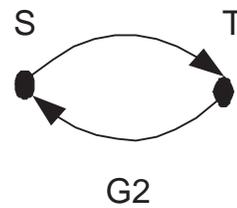


Figura 4.12: Red utilizada para mostrar los marcados que inevitablemente conducen a un bloqueo.



$$L((R, S)) = \{(t_7, e)\} K_{G_1}(R) = \{a\}$$

$$L((S, R)) = \{(t_2, a)\} K_{G_1}(S) = \{e\}$$



$$L((S, T)) = \{(t_6, d)\} K_{G_2}(S) = \{b\}$$

$$L((T, S)) = \{(t_3, b)\} K_{G_2}(T) = \{d\}$$

Figura 4.13: Grafo de poda G de la red de la figura 4.12.

Vértice Común	Arcos que salen del vértice	Etiqueta del arco	Marcas solicitadas	$\sum \text{Pre}[r,t]$	Marcado del recurso
S	$(S,R), (S,T)$	$(t2,a), (t6,d)$	1, 1	2	1

Cuadro 4.2: Tabla que muestra la solicitud de marcas para el vértice S en el grafo G .

d superior al marcado que el recurso S posee como se muestra en la tabla 4.2. De esta manera, aplicando el método podemos determinar que este marcado conduce a bloquear la red.

4.6. Conclusión

En este capítulo se han mostrado diversas aplicaciones del grafo de poda. La primera ha sido una aplicación más general, el cálculo de los cerrojos no-mínimos, el cual lo derivamos de la operación de unión de los cerrojos mínimos de un recurso de una red S^4PR .

Otra aplicación del grafo de poda que se ha explicado es el cálculo de los cerrojos mínimos de una red S^4PR . Para realizar esta aplicación se han utilizado las herramientas de poda explicadas en el capítulo anterior con el objetivo de cribar entre el conjunto de cerrojos aquellos que son mínimos. De igual forma, se ha desarrollado un nuevo algoritmo que evita tener que evaluar muchas uniones no-productivas de cerrojos mínimos de la clase \mathcal{D}^1 . Diversas variantes se han explicado para este algoritmo con el objetivo de hacer más eficiente su implementación. También, se ha mostrado el coste computacional que el algoritmo posee.

Finalmente se ha presentado la relación entre el Teorema de vivacidad y el grafo de poda, se presenta además un nuevo método que nos permite identificar los marcados que inevitablemente conducen a bloquear la red.

Capítulo 5

Especialización para Subclases de S^4PR

Resumen

Las redes $L - S^3PR$, S^3PR y $SOAR^2$ son clases de redes que han sido desarrolladas para modelar diversas clases de sistemas. Por ejemplo, las redes $L - S^3PR$ y S^3PR han sido muy utilizadas en la literatura en el modelamiento de Sistemas de Fabricación Flexible, mientras que la nueva clase de redes $SOAR^2$ se utiliza para modelar el encaminamiento en Redes de Interconexión de Computadores ó en Sistemas de Vehículos Guiados.

Cada una de estas redes posee características estructurales que establecen diferencias con respecto a las redes S^4PR . Basados en la definición de cada una de estas redes, en este capítulo, se realiza una especialización para los algoritmos de cálculo de cerrojos mínimos para las mismas. Para conseguir este objetivo, se analiza la definición de cada clase y se buscan propiedades que nos permitan simplificar los cálculos intermedios del algoritmo, enfocándonos en aspectos tales como la relación de poda entre los cerrojos mínimos de un recurso, el cálculo del grafo de poda y sus funciones de etiquetado y de permitirlo la clase, realizar alguna mejora al algoritmo de cálculo de cerrojos mínimos.

Finalmente, como resultado del análisis de cada una de las clases, se desarrollarán nuevos algoritmos que nos permitirán calcular, de manera más económica, el conjunto de cerrojos mínimos que estas redes poseen. De igual manera, se presenta el análisis del coste computacional de éstos algoritmos.

5.1. Introducción

Uno de los temas más investigados en la actualidad, para ciertas subclases de redes de Petri con una caracterización basada en su estructura, lo constituyen el cálculo de los cerrojos mínimos conteniendo más de un recurso. Esto se debe a que para estas subclases, la propiedad de vivacidad ha sido caracterizada en términos de estos cerrojos y por lo tanto contar con algoritmos eficientes para el cálculo de los mismos resulta indispensable.

Tres de estas subclases, las redes $L - S^3PR$, S^3PR y $SOAR^2$ son las que se van a analizar en este capítulo, con el objetivo de obtener propiedades que nos permitan desarrollar algoritmos de cálculo de cerrojos mínimos con más de un recurso que sean más eficientes que el presentado en el capítulo anterior. Basados en la definición de cada subclase, se estudian las propiedades para cada una de ellas y se analiza en primer lugar la relación de poda entre los cerrojos mínimos de un recurso y su representación mediante el grafo de poda y sus funciones de etiquetado, investigando si es posible obtener nuevos algoritmos para el grafo y sus funciones asociadas que logren caracterizar dicha relación de una manera más simplificada. De manera similar, se investigan para cada clase la posibilidad de realizar alguna mejora al algoritmo de cálculo de cerrojos mínimos.

Las redes $L - S^3PR$ [EGVC98, GV99], S^3PR [ECM95, Ezp93] y $SOAR^2$ [Rov11] son clases de redes que han sido desarrolladas para modelar, analizar y sintetizar diversas clases de sistemas. En este capítulo se analizarán cada una de ellas con la finalidad de, en primer lugar, caracterizar los cerrojos mínimos de un solo recurso para después hacer el análisis de la relación de poda entre estos cerrojos. A partir de esta información, se presentará el análisis para la obtención del grafo de poda de la red, definiendo si ese el caso, las diferencias y los nuevos algoritmos que mejoren al algoritmo antes desarrollado. De igual forma se realizará el análisis del coste computacional para los nuevos algoritmos que se introduzcan.

El resto del capítulo se organiza como sigue. En la sección 5.2 se realiza el análisis para las redes $L - S^3PR$. El estudio de las redes S^3PR se lleva a cabo en la sección 5.3. La clase de redes $SOAR^2$ son investigadas en la sección 5.4. Finalmente, la sección 5.5 presenta algunas conclusiones del capítulo.

5.2. Redes $L - S^3PR$

En esta sección se realiza una especialización del algoritmo de cálculo de cerrojos mínimos conteniendo más de un recurso para las redes $L - S^3R$ [EGVC98, GV99]. Estas redes son una subclase de las S^3PR presentadas en [ECM95]. La principal característica de esta subclase es que no existe selección en los procesos. A continuación se define formalmente la clase $L - S^3PR$.

Definición 13. La clase de red $L - S^3PR$ [EGVC98, GV99].

Una Lineal S^3PR ($L - S^3PR$) es una red ordinaria $\mathcal{N} = \langle P, T, F \rangle$, tal que:

1. $P = P_R \cup P_S \cup P_0$ es una partición tal que:
 - a) $P_0 = \{p_0^1, \dots, p_0^k\}, k > 0$ (lugar reposo).
 - b) $P_S = \bigcup_{i=1}^k P_S^i$, donde $P_S^i \cap P_S^j = \emptyset$, para todo $i \neq j$.
 - c) $P_R = \{r_1, \dots, r_n\}, n > 0$.
2. $T = \bigcup_{i=1}^k T^i$, donde $T^i \cap T^j = \emptyset$, para todo $i \neq j$.
3. $\forall i \in \{1, \dots, k\}$ la subred \mathcal{N}_i generada por $\{p_0^i\} \cup P_S^i \cup T^i$ es una máquina de estados fuertemente conexa, tal que cada ciclo contiene $\{p_0^i\}$ y $\forall p \in P_S^i, |\bullet p| = |p \bullet| = 1$.
4. $\forall i \in \{1, \dots, k\}, \forall p \in P_S^i, \bullet\bullet p \cap P_R = p \bullet\bullet \cap P_R$ y $|\bullet\bullet p \cap P_R| = 1$.
5. \mathcal{N} es fuertemente conexa.

A partir de la definición anterior podemos indicar lo siguiente, los *lugares proceso* de la red son los lugares pertenecientes al conjunto P_S . Cada subred \mathcal{N}^i define un *Proceso Lineal Secuencial Simple* $L - S^2P$. Los lugares pertenecientes al conjunto P_R se les denomina *lugares recurso* de la red. En cada estado un proceso usa un único recurso aparte del lugar reposo, por lo tanto, se define un *Proceso Lineal Secuencial Simple con Recursos* $L - S^2PR$. La interacción de un proceso con el resto de los procesos en todo el sistema ($L - S^3PR$) se realiza compartiendo un conjunto de recursos.

El lugar p_0^i representa el estado de reposo del proceso i y se le llama *lugar de reposo*. En este estado no existe interacción entre los procesos, por lo tanto, no se utiliza ningún recurso.

El nombre de *lineal* y *simple* que tienen los procesos en una $L - S^3PR$ se deriva de la restricción especial impuesta en las máquinas de estados y en la forma en la ellas utilizan el conjunto de recursos. La figura 5.1 muestra una red $L - S^3PR$.

Se define la siguiente terminología para el conjunto de recursos P_R [EGVC98, GV99]:

- para un $p \in P_S$ dado, $\bullet\bullet p \cap P_R = \{r_p\}$. El recurso r_p es llamado el *recurso usado por* p .
- para un $r \in P_R$ dado, $H(r) = \bullet\bullet r \cap P_S (= r \bullet\bullet \cap P_S)$ es llamado el conjunto de usuarios de r (estados que usan r).

Elementos
$P_0 = \{p01, p02, p03\}$
$P_S^1 = \{p1, p2, p3\}$
$P_S^2 = \{p4, p5, p6\}$
$P_S^3 = \{p7, p8, p9\}$
$P_R = \{X, Y, Z\}$
$T^1 = \{t1, t2, t3, t4\}$
$T^2 = \{t5, t6, t7, t8\}$
$T^3 = \{t9, t10, t11, t12\}$
$H(X) = \{p1, p5, p8\}$
$H(Y) = \{p2, p4, p9\}$
$H(Z) = \{p3, p6, p7\}$

Cuadro 5.1: Elementos de la red $L - S^3PR$ de la figura 5.1.

Para la red de la figura 5.1 los elementos antes definidos son mostrados en la tabla 5.1.

A continuación, para completar la definición anterior, se introduce una familia de marcados iniciales para las redes $L - S^3PR$ [EGVC98, GV99].

Definición 14. Sea $\mathcal{N} = \langle P_S \cup P_0 \cup P_R, T, F \rangle$ una red $L - S^3PR$. Un marcado inicial \mathbf{m}_0 es llamado un marcado inicial admisible para \mathcal{N} si, y solo si:

1. $\mathbf{m}_0[p_0] \geq 1, \forall p_0 \in P_0$
2. $\mathbf{m}_0[p] = 0, \forall p \in P_S$
3. $\mathbf{m}_0[r] \geq 1, \forall r \in P_R$

Utilizando las definiciones anteriores se puede apreciar que para esta clase de red los cerrojos mínimos que contienen un único recurso son precisamente los soportes de los p-semiflujos de recurso tal y como se indica en el lema 9 del capítulo 2. Por lo tanto, la clase de los cerrojos mínimos \mathcal{D}^1 está completamente caracterizada y no es necesario su cálculo. Por ejemplo, los cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 para la red de la figura 5.1 son mostrados en la tabla 5.2 y como se puede observar corresponden a los soportes de los p-semiflujos de sus recursos.

A partir de lo antes presentado podemos establecer la relación de poda en el conjunto \mathcal{D}^1 como se ha hecho en el capítulo 3: el cerrojo $D_r \in \mathcal{D}^1$ poda al cerrojo $D_x \in \mathcal{D}^1, r \neq x$, si, y solo si, ambos cerrojos comparten alguna transición $T_{rx} = D_r^\bullet \cap D_x^\bullet \neq \emptyset$ y en esa transición común entran: un lugar proceso perteneciente a D_x y el recurso r , i. e., $U_r =$

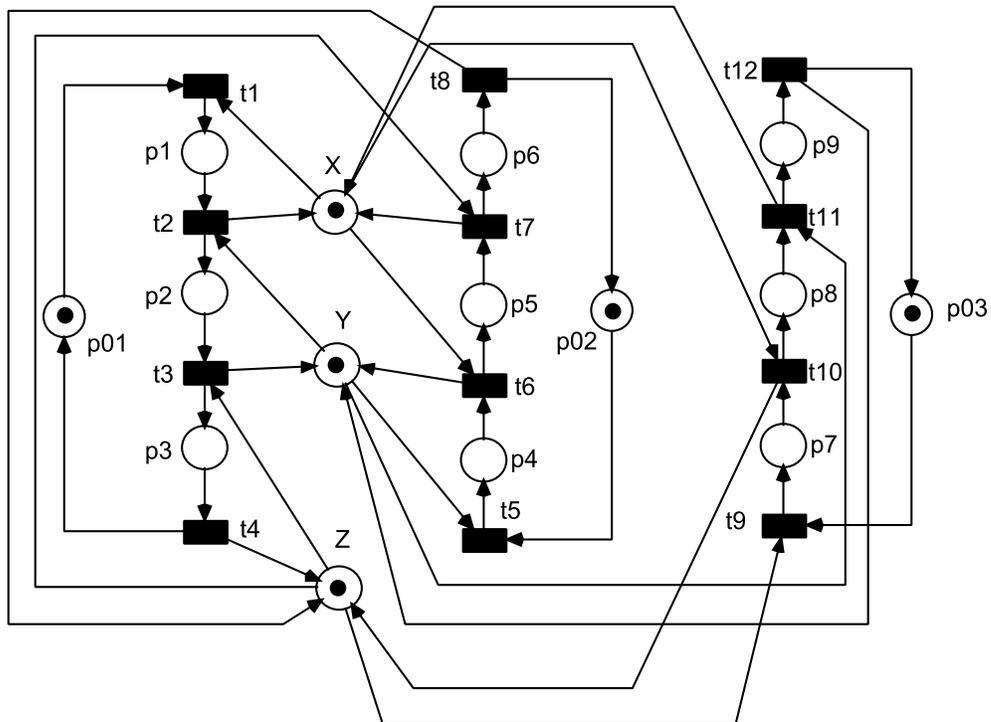


Figura 5.1: Red de Petri de la Clase LS^3PR .

Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1
$D_X = \{X, p1, p5, p8\}$
$D_Y = \{Y, p2, p4, p9\}$
$D_Z = \{Y, p3, p6, p7\}$

Cuadro 5.2: Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red $L - S^3PR$ de la figura 5.1.

$r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap D_x \cap P_s)^\bullet \neq \emptyset$. Por lo tanto, la definición de la relación de poda entre cerrojos que contienen un sólo recurso en redes S^4PR antes presentada en el capítulo 3 (definición 7) puede ser utilizada para las redes de la clase $L - S^3PR$. Sin embargo, dado que para la clase $L - S^3PR$ los cerrojos mínimos de un recurso contienen los soportes de los p-semiflujos del recurso debemos indicar que un cerrojo no puede podarse a sí mismo como se realiza en una red S^4PR y que lo formaliza el lema 10 del capítulo 3. De esta manera, podemos indicar que la relación de poda entre cerrojos que contienen un sólo recurso puede ser representada mediante el grafo de poda como se ha hecho en el capítulo 3 considerando en la definición del mismo lo antes apuntado: un cerrojo mínimo de un único recurso en una red $L - S^3PR$ no puede podarse a sí mismo. Lo cual lleva a establecer una nueva restricción a la definición del grafo de poda dada en el capítulo 3 (definición 8) para poder utilizarla en las redes $L - S^3PR$. A continuación se formaliza la nueva definición del grafo de poda para redes $L - S^3PR$.

Definición 15. Grafo de Poda para redes $L - S^3PR$.

Sea \mathcal{N} una red $L - S^3PR$ y P_R el conjunto de lugares recurso. El Grafo de Poda (PG) de \mathcal{N} es un grafo $G = (V, E)$ donde,

1. $V = P_R$; y
2. $E \subseteq V \times V$ y para todo $r, x \in P_R$, $r \neq x$, $(r, x) \in E$ si, y solo si, el cerrojo $D_r \in \mathcal{D}^1$ poda al cerrojo $D_x \in \mathcal{D}^1$, es decir,

$$U_r = r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap D_x \cap P_s)^\bullet \neq \emptyset$$

con

$$T_{rx} = D_r^\bullet \cap D_x^\bullet.$$

Obsérvese que en la definición anterior un vértice no puede tener un bucle debido a la restricción antes discutida sobre los cerrojos mínimos de un recurso para las redes $L - S^3PR$ y que impide a un cerrojo mínimo de un recurso estar en relación de poda consigo mismo como lo pueden hacer las redes S^4PR y que fue formalizado en el lema 10 del capítulo 3.

Las funciones asociadas al grafo de poda pueden definirse como se ha realizado en el capítulo 3, sin embargo y tomándo ventaja de la estructura de las redes $L - S^3PR$ podemos obtener un algoritmo más económico para calcular la función de etiquetado de poda K_G del grafo. Efectivamente, a partir de la definición de la clase (definición 13.4) podemos simplificar el algoritmo 1 debido a que para las redes $L - S^3PR$ no es necesario realizar una búsqueda de posibles lugares candidatos a ser podados como se tiene que realizar en las redes S^4PR , por tal motivo, el algoritmo se reduce. A continuación se definen las

funciones de etiquetado y el nuevo algoritmo calcular la función de etiquetado de poda K_G .

Definición 16. Sea \mathcal{N} una red $L - S^3PR$ y $G = (V, E)$ un grafo o subgrafo de poda de \mathcal{N} .

a) La función de etiquetado de los Recursos. $S : P_R \rightarrow \mathcal{D}^1$, donde para todo $r \in P_R$, $S(r) = D_r \in \mathcal{D}^1$.

b) La función de etiquetado de los Arcos. $L : E \rightarrow 2^{T \times P_S}$, donde $\forall (r, x) \in E$,

$$L((r, x)) = \{(t, p) | t \in r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap S(x) \cap P_S)^\bullet; T_{rx} = S(r)^\bullet \cap S(x)^\bullet; \{p\} = \bullet t \cap P_S\}.$$

c) La función de etiquetado de Poda. $K_G : V \rightarrow 2^{P_S}$, donde para todo $r \in V$, $K_G(r) \subseteq P_S$, $K_G(r) = \{p | (t, p) \in L((x, r)); (x, r) \in E\}$.

A partir de la definición del grafo del poda y sus funciones de etiquetado se obtiene para la red de la figura 5.1 el grafo de poda mostrado en la figura 5.2. Como se puede observar en el grafo de poda de la figura 5.2 cada transición en común entre los cerrojos mínimos de un recurso poda sólo un lugar del cerrojo.

Para construir el conjunto de cerrojos mínimos de una red $L - S^3PR$ se utiliza el resultado mostrado en el Teorema 4 del capítulo 4 que caracteriza un cerrojo mínimo en términos de un subgrafo de poda de la red inducido en el conjunto de recursos del cerrojo mínimo. El cálculo de los cerrojos mínimos para las redes $L - S^3PR$ se puede realizar como se ha hecho para las redes S^4PR utilizando el algoritmo 2 presentado en el capítulo 4. Sin embargo, dado que en las redes $L - S^3PR$ no existe selección en los procesos, el algoritmo 2 puede ser simplificado con el objetivo de obtener un algoritmo más económico. Basados en la estructura de las redes $L - S^3PR$ y en el apartado 4 de la definición 13 podemos mejorar el algoritmo 2 del capítulo 4 eliminando los pasos 7 al 11 debido a que en un grafo de poda obtenido para las redes $L - S^3PR$ todos los K_G s de los recursos contenidos en los vértices del grafo G o de sus subgrafos G' son diferentes de vacío, i.e., $\forall i \in \{1, \dots, k\}, \forall p \in P_S^i, \bullet\bullet p \cap P_R = p^\bullet\bullet \cap P_R$ y $|\bullet\bullet p \cap P_R| = 1$ y por la definición 15.2 podemos concluir que $K_G(r) \neq \emptyset; \forall r \in V$.

Calcular el grafo y subgrafos de poda para esta clase de redes tiene una complejidad de $O(|P_R| \cdot |P_S|)$. Por otro lado, calcular los subgrafos fuertemente conexos para cada grafo tiene una complejidad de $O(|V|^2)$, lo cual está limitado en nuestro caso por $O(|P_R|^2)$. Para los B cerrojos mínimos se requiere para estas clases de un tiempo en el número de operaciones de $O(B(|P_R|^2 + |P_R| \cdot |P_S|))$. Por lo tanto podemos concluir que el costo computacional para este algoritmo es igual al apuntado en la sección 4.4 del capítulo 4. A continuación se muestra el nuevo algoritmo para el cálculo de los cerrojos mínimos para redes $L - S^3PR$.

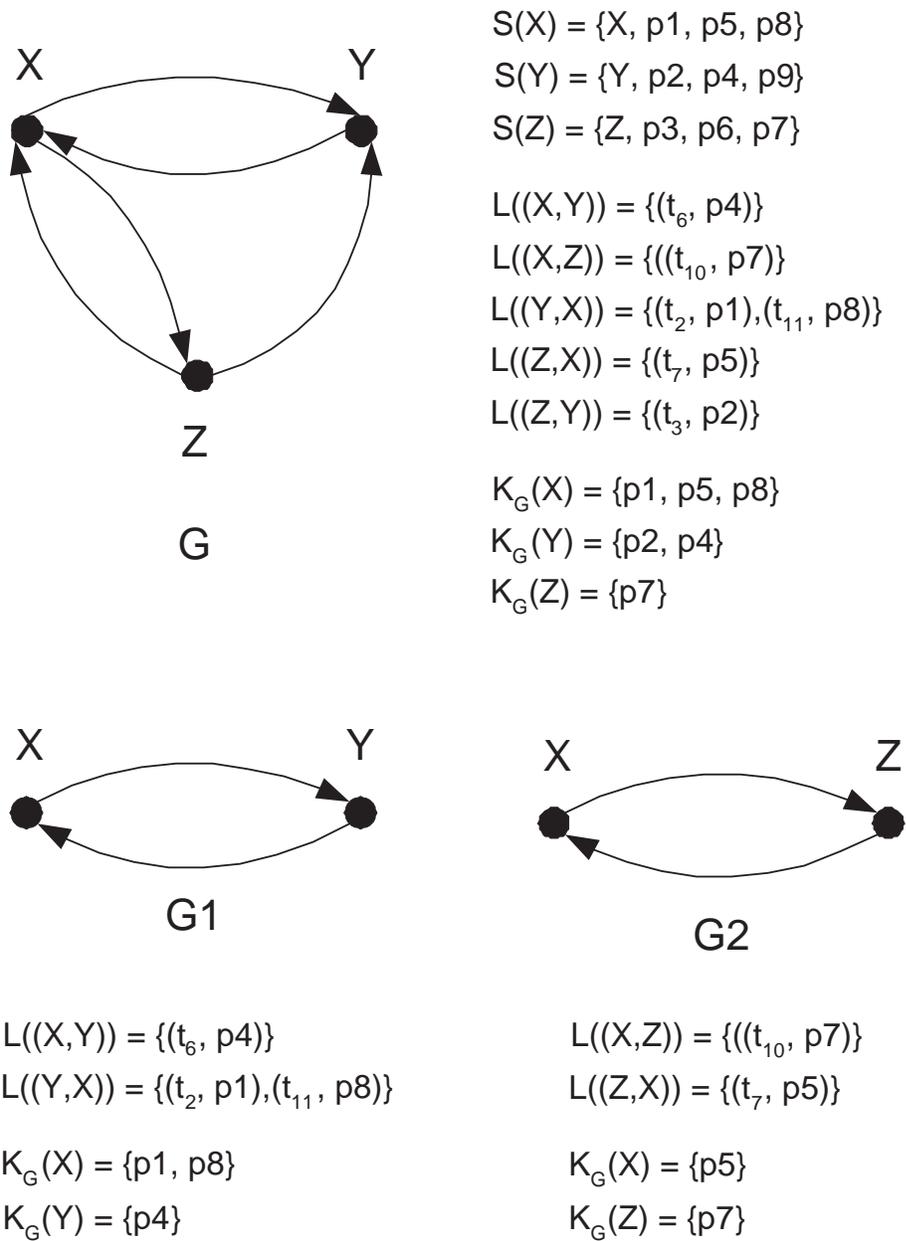


Figura 5.2: Grafo de poda de la red de la figura 5.1.

Algoritmo 4 Cálculo de los Cerrojos Mínimos de \mathcal{N} para redes $L - S^3PR$.

Entrada: $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ – La red $L - S^3PR$.

\mathcal{D}^1 – Los cerrojos mínimos con solo un recurso.

Salida: \mathcal{D} – Los cerrojos mínimos de \mathcal{N} conteniendo al menos un recurso.

1. **Inicio**
 2. $\mathcal{D} := \mathcal{D}^1$
 3. Calcular el grafo de poda $G = (V, E)$ de \mathcal{N} (definición 15), y las funciones S, L y K_G (definición 16).
 4. Calcular todos los subgrafos fuertemente conexos máximos (componentes fuertemente conexas, scc) de G , y para cada scc, G' , calcular las funciones S, L y $K_{G'}$ para formar el $PG G'$. Agregar todos los $PGs G'$ con más de un vértice al conjunto g_{nuevo} .
 5. $g_{bueno} := \emptyset$
 6. **Mientras que** $g_{nuevo} \neq \emptyset$ **Hacer**
 7. Extraer $G' = (V', E')$ de g_{nuevo}
 8. **Si** no existe otro $G = (V, E)$ en g_{bueno} que satisfice: (1) $V' \subseteq V$; y (2) $S_{G'} \subset S_G$
 Entonces
 9. Agregar G' al conjunto g_{bueno}
 10. **Fin Si**
 11. **Para** cada $r \in V'$ **Hacer**
 12. Calcular el subgrafo de poda de G' inducido en $V' \setminus \{r\}, G''$
 13. Calcular todos los scc de G'' , y para cada scc, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el $PG G'''$. Agregar todos los $PGs G'''$ con más de un vértice al conjunto g_{nuevo}
 14. **Fin Para**
 15. **Fin Mientras que**
 16. **Para** cada grafo $G' = (V', E') \in g_{bueno}$ **Hacer**
 17. Agregar a \mathcal{D} el cerrojo $D = \bigcup_{r \in V'} S(r) \setminus K_{G'}(r) = S_{G'}$
 18. **Fin Para**
 19. **Fin**
-

5.3. Redes S^3PR

Los Sistemas de Procesos Secuenciales Simples con Recursos S^3PR presentada en [ECM95, Ezp93] es una clase de redes que está formada por un conjunto de procesos secuenciales (modelados mediante máquinas de estados fuertemente conexas) que utilizan un recurso en cada estado, siendo esta última la principal característica de estas redes.

La especialización del algoritmo de cálculo de cerrojos mínimos conteniendo más de un recurso para las redes S^3PR [ECM95, Ezp93] se realiza en esta sección. En [ECM95, Ezp93] se define esta clase de redes de manera constructiva a partir de la definición de la clase de los *Procesos Secuenciales Simples (PSS)*. De manera similar en esta memoria se procede a definir primero la clase de los Procesos Secuenciales Simples (PSS ó S^2P), la cual será extendida a la clase de Procesos Secuenciales Simples con Recursos (PSSR ó S^2PR), y finalmente se definirá la clase de los Sistemas de Procesos Secuenciales Simples con Recursos (SPSSR ó S^3PR) por la composición de redes S^2PR mediante la fusión de un conjunto de lugares comunes. Las definiciones corresponden a [Ezp93].

Definición 17. *Un Proceso Secuencial Simple (PSS) es una red de Petri ordinaria $\mathcal{N} = \langle P \cup \{p^0\}, T, F \rangle$ tal que:*

1. \mathcal{N} es una máquina de estados fuertemente conexas.
2. $p^0 \notin P$.
3. cada circuito de \mathcal{N} contiene al lugar p^0 .

Como se ha indicado en la sección anterior el lugar p^0 corresponde al lugar de reposo, en donde, el mismo representa la inactividad del sistema.

Definición 18. *Un Proceso Secuencial Simple con Recursos (PSSR) es una red de Petri ordinaria $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R, T, F \rangle$ tal que:*

1. La subred generada por $X = P \cup \{p^0\} \cup T$ es un PSS.
2. $P_R \neq \emptyset$ y $(P \cup \{p^0\}) \cap P_R = \emptyset$.
3. $\forall p \in P, \forall t \in \bullet p, \forall t' \in p^\bullet : \bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$.
4. Las dos siguientes condiciones son verificadas:
 - $\forall r \in P_R, \bullet r \cap r^\bullet = \emptyset$
 - $\forall r \in P_R, \bullet\bullet r \cap P = r^\bullet\bullet \cap P \neq \emptyset$

$$5. \bullet\bullet\{p^0\} \cap P_R = \{p^0\} \bullet\bullet \cap P_R = \emptyset$$

Los lugares r modelan los recursos del sistema, mientras que P_R contiene el conjunto de recursos del sistema. La condición 3 de la definición modela la disponibilidad del recurso r usado por el lugar p , de esta forma, se establece los lugares usuarios de un recurso r_p y restringe el conjunto de recursos para cada estado del proceso a un único elemento. Para un recurso $r \in P_R$, el conjunto de lugares $H(r) = (\bullet\bullet r \cup r \bullet\bullet) \cap P$ se denomina *conjunto de usuarios de r* .

A continuación se introduce los marcados iniciales para los PSSR.

Definición 19. Sea $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R, T, F \rangle$ un PSSR. Un marcado inicial \mathbf{m}_0 se dice que es admisible para \mathcal{N} si, y solo si:

1. $\mathbf{m}_0[p^0] \geq 1$
2. $\mathbf{m}_0[p] = 0, \forall p \in P$
3. $\mathbf{m}_0[r] \geq 1, \forall r \in P_R$

Notación En lo sucesivo, dado un PSSR $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R, T, F \rangle$ denotaremos $P^0 = \{p^0\}$.

Antes de introducir la definición de las S^3PR se define la composición de redes vía un conjunto de lugares comunes.

Definición 20. Sean $\mathcal{N}_i = \langle P_i, T_i, F_i \rangle, i \in \{1, 2\}$ dos redes de Petri ordinarias. Entonces:

1. \mathcal{N}_1 y \mathcal{N}_2 son componibles por fusión de lugares si, y sólo si, $T_1 \cap T_2 = \emptyset$ y $P_C = P_1 \cap P_2 \neq \emptyset$.
2. la red $\mathcal{N} = \langle P, T, F \rangle$ tal que $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, $F = F_1 \cup F_2$ se denomina red compuesta de \mathcal{N}_1 y \mathcal{N}_2 , y se denotará como $\mathcal{N} = \mathcal{N}_1 \circ \mathcal{N}_2$.

A continuación se define recursivamente las redes S^3PR .

Definición 21. Un SPSSR se define recursivamente como sigue:

1. Todo PSSR es un SPSSR.
2. Si $\mathcal{N}_i = \langle P_i \cup P_i^0 \cup P_{R_i}, T_i, F_i \rangle, i \in \{1, 2\}$ son dos SPSSR componibles por fusión de un conjunto de recursos comunes $P_C \subseteq P_{R_1} \cap P_{R_2}$, entonces $\mathcal{N} = \mathcal{N}_1 \circ \mathcal{N}_2$ es también un SPSSR.

Elementos
$P_0 = \{p01, p02\}$
$P_1 = \{a, b, c, d, e\}$ $P_2 = \{f, g, h\}$ $P_1 = \{a, b, c, d, e, f, g, h\}$
$P_R = \{R, S, T\}$
$T^1 = \{t1, t2, t3, t4, t5, t6, t7\}$ $T^2 = \{t8, t9, t10, t11\}$ $T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11\}$
$H(R) = \{a, h\}$ $H(S) = \{c, g\}$ $H(T) = \{d, e, f\}$

Cuadro 5.3: Elementos de la red S^3PR de la figura 5.3.

Una red S^3PR es mostrada en la figura 5.3, los elementos que la componen son resumidos en la tabla 5.3.

Después de la definición de la clase S^3PR se define un conjunto de marcados admisibles para completar la definición de la misma.

Definición 22. Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SPSSR marcado. Diremos que está admisiblemente marcado cuando se verifica una de las siguientes condiciones:

1. $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ es un PSSR admisiblemente marcado
2. $\mathcal{N} = \mathcal{N}_1 \circ \mathcal{N}_2$, tal que $\langle \mathcal{N}_i, \mathbf{m}_{0i} \rangle$ es un PSSR admisiblemente marcado y los \mathbf{m}_{0i} verifican que
 - $\forall i \in \{1, 2\}, \forall p \in P_i \cup P_i^0, \mathbf{m}_0[p] = \mathbf{m}_{0i}[p]$
 - $\forall i \in \{1, 2\}, \forall r \in P_{R_i} \setminus P_C, \mathbf{m}_0[r] = \mathbf{m}_{0i}[r]$
 - $\forall r \in P_C, \mathbf{m}_0[r] = \max\{\mathbf{m}_{01}[r], \mathbf{m}_{02}[r]\}$

A partir de la definición anterior de la clase S^3PR podemos indicar que en esta clase de red al igual que en las redes $L - S^3PR$ los soportes de los p-semiflujos de recurso forman el cerrojo mínimo del recurso. Esto se obtiene de la definición de la clase y es formalizado en el lema 9 del capítulo 2. De esta manera y al igual que en las redes $L - S^3PR$, la clase \mathcal{D}^1 que contiene el conjunto de cerrojos mínimos de un recurso se obtiene directamente

Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1
$D_R = \{R, a, h\}$
$D_S = \{S, c, g\}$
$D_T = \{T, d, e, f\}$

Cuadro 5.4: Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red S^3PR de la figura 5.3.

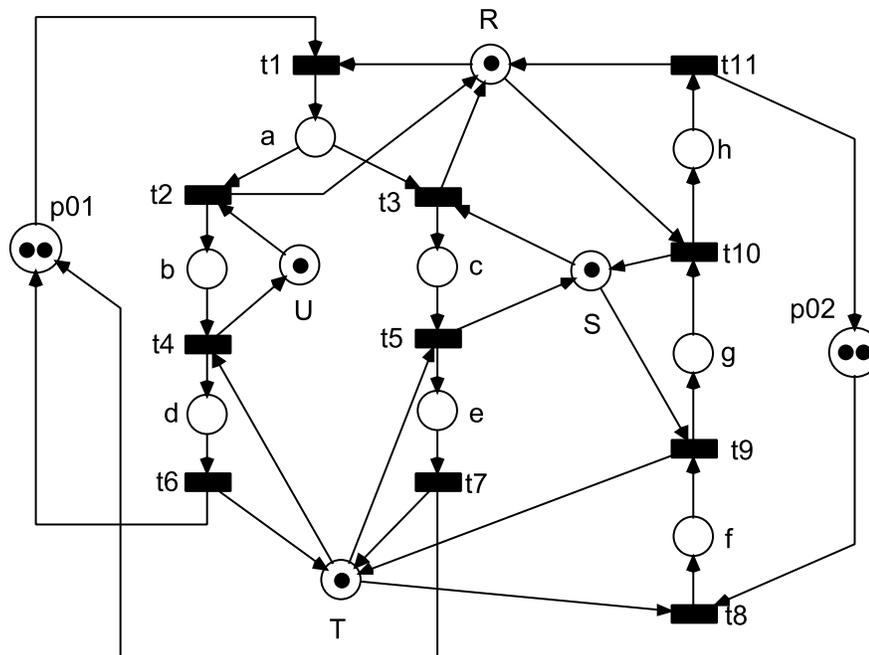


Figura 5.3: Red de Petri de la Clase S^3PR .

de la definición de las S^3PR . Los cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 para la red de la figura 5.3 son mostrados en la tabla 5.4.

Para esta clase de red la relación de poda definida entre el conjunto de cerrojos mínimos de un solo recurso en el capítulo 3 (definición 7) puede ser utilizada, sin embargo y al igual que en las redes $L - S^3PR$, los cerrojos mínimos de un recurso no puede podarse a sí mismo debido a la definición 18.3. Entonces, de manera similar a como se realizó con las redes $L - S^3PR$, podemos representar la relación de poda entre cerrojos de un sólo recurso mediante el grafo de poda considerando lo antes explicado, que un cerrojo mínimo de un único recurso en una red S^3PR no puede podarse a sí mismo y por lo tanto el grafo de poda para las redes S^3PR no pueden tener bucles. Lo cual sugiere introducir la misma restricción que se estableció para las redes $L - S^3PR$ en la definición del grafo de poda. Por lo tanto, para definir el grafo de poda para las redes S^3PR se utilizará la definición 15 dada en la sección anterior para las redes $L - S^3PR$.

Otro aspecto importante que debemos tener en cuenta está relacionado con la estructura de las redes S^3PR y las funciones de etiquetado asociadas al grafo de poda definidas en el capítulo 3, específicamente con la función K_G para la cual su algoritmo de cálculo puede ser optimizado igual a como se realizó en la sección anterior para las redes $L - S^3PR$. En efecto, utilizando la definición de la clase podemos calcular los lugares a ser eliminados de un cerrojo de un recurso de una forma más económica. Basándonos en el apartado 3 de la definición 18, i.e. $\forall p \in P, \forall t \in \bullet p, \forall t' \in p^\bullet : \bullet t \cap P_R = t' \bullet \cap P_R = \{r_p\}$, sabemos que sólo existe un lugar proceso que pueda ser eliminado de un cerrojo de un recurso D_r por otro cerrojo de un recurso D_x y de esta manera no es necesario hacer la búsqueda de un conjunto de posibles lugares a eliminar como se realiza para las redes S^4PR . A continuación se introduce la definición de las funciones de etiquetado asociadas al grafo de poda en la cual se indica que la función de etiquetado de poda K_G se realiza utilizando el algoritmo de poda para las redes S^3PR . De igual forma, se presenta el nuevo algoritmo para calcular la función de etiquetado de poda K_G .

Definición 23. Sea \mathcal{N} una red S^3PR y $G = (V, E)$ un grafo o subgrafo de poda de \mathcal{N} .

- a) La función de etiquetado de los Recursos. $S : P_R \rightarrow \mathcal{D}^1$, donde para todo $r \in P_R$, $S(r) = D_r \in \mathcal{D}^1$.
- b) La función de etiquetado de los Arcos. $L : E \rightarrow 2^{T \times P_s}$, donde $\forall (r, x) \in E$,

$$L((r, x)) = \{(t, p) | t \in r^\bullet \cap T_{rx} \cap (\bullet T_{rx} \cap S(x) \cap P_s)^\bullet; T_{rx} = S(r)^\bullet \cap S(x)^\bullet; \{p\} = \bullet t \cap P_s\}.$$
- c) La función de etiquetado de Poda. $K_G : V \rightarrow 2^{P_s}$, donde para todo $r \in V$, $K_G(r) \subseteq P_s$, y se calcula mediante el algoritmo 5.

Algoritmo 5 Función de etiquetado de Poda de G para redes S^3PR .

Entrada: $\mathcal{N} = \langle P_0 \cup P_s \cup P_R, T, C \rangle$ – La red S^3PR .

$r \in P_R$ – Los recursos de un cerrojo $D_r \in \mathcal{D}^1$ para el cual calcularemos el conjunto de poda en el grafo G .

$G = (V, E)$ – El grafo o subgrafo de poda de \mathcal{N} .

Salida: $K_G(r) \subseteq P_s$

1. $T_F = \{t \mid (t, p) \in L((x, r)) \text{ y } (x, r) \in E\}$
 2. $K_G(r) = \{p \mid (t, p) \in L((x, r)); (x, r) \in E; \text{ y } p^\bullet \subseteq T_F\}$
-

El algoritmo 5 verifica que todas las transiciones de salida del lugar p se encuentren en las etiquetas de los arcos de entrada al vértice r para que el lugar pueda ser eliminado. Esto lo podemos observar en el grafo de poda PG de la figura 5.4 correspondiente a la red de la figura 5.3, en donde todas las transiciones de salida del lugar a se encuentran contenidas en los arcos de entrada al vértice R .

El conjunto de cerrojos mínimos para las redes S^3PR se obtienen al igual que para las redes $L - S^3PR$ utilizando el Teorema 4 del capítulo 4 y puede utilizarse para este propósito el algoritmo 2. Sin embargo, si observamos la definición de las redes S^3PR y la forma en la cual los recursos son utilizados por los procesos podemos simplificar los pasos 7 al 11 del algoritmo 2 del capítulo 4 debido que todos los posibles vértices del grafo con un $K'_G(r) = \emptyset$ corresponden a los subgrafos fuertemente conexos máximos. Por lo tanto, la verificación que antes se realizaba en esos pasos puede ser simplificada en un sólo paso consiguiéndose de esta forma un algoritmo más eficiente. Dado que los cambios principales que se han realizado debido a las restricciones impuestas por esta clase de red corresponden al cálculo del grafo o subgrafos de poda y que estos tienen una complejidad de $O(|P_R| \cdot |P_s|)$ y que la complejidad del cálculo de los subgrafos fuertemente conexos es igual al antes apuntado $O(|P_R|^2)$, podemos indicar que el costo computacional para este algoritmo es igual al apuntado en la sección 4.4 del capítulo 4. A continuación se introduce el nuevo algoritmo para el cálculo de los cerrojos mínimos para redes S^3PR .

5.4. Redes $SOAR^2$

Las redes $SOAR^2$ [Rov11] son una nueva clase de redes desarrolladas para modelar el encaminamiento en Redes de Interconexión de Computadores ó en Sistemas de Vehículos Guiados. La principal característica de estas redes radica en que el marcado inicial en cada recurso es igual a la unidad. En esta sección se realiza la especialización del algoritmo de cálculo de cerrojos mínimos para las redes $SOAR^2$. A continuación se define formalmente

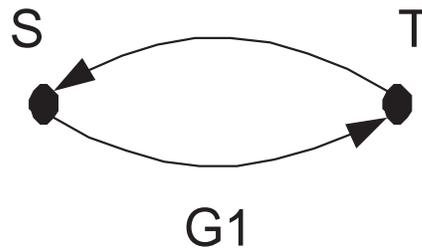
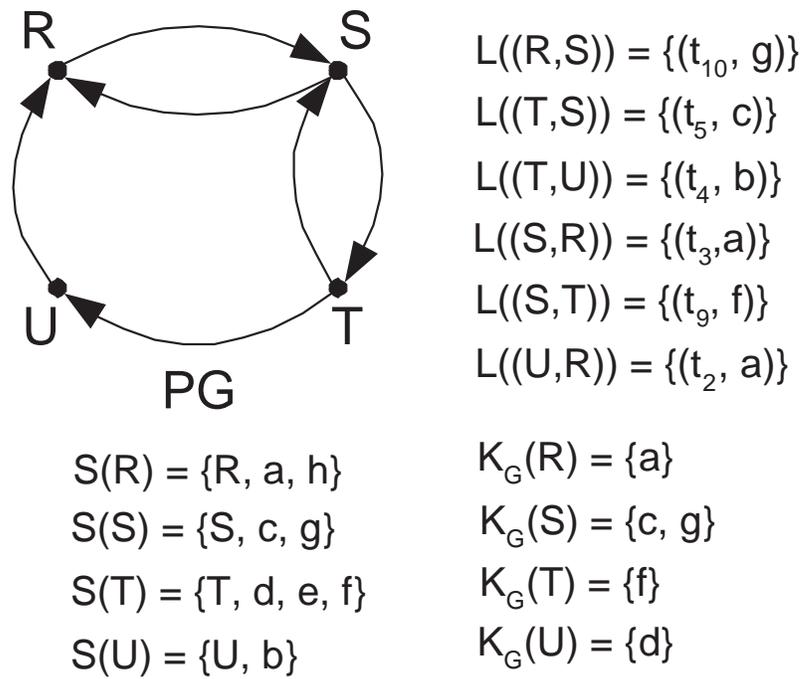


Figura 5.4: Grafo de poda de la red de la figura 5.3.

Algoritmo 6 Cálculo de los Cerrojos Mínimos de \mathcal{N} para redes S^3PR .

Entrada: $\mathcal{N} = \langle P_0 \cup P_s \cup P_R, T, C \rangle$ – La red S^3PR .

\mathcal{D}^1 – Los cerrojos mínimos con solo un recurso.

Salida: \mathcal{D} – Los cerrojos mínimos de \mathcal{N} conteniendo al menos un recurso.

1. **Inicio**

2. $\mathcal{D} := \mathcal{D}^1$

3. Calcular el grafo de poda $G = (V, E)$ de \mathcal{N} (definición 15), y las funciones S, L y K_G (definición 23).

4. Calcular todos los subgrafos fuertemente conectados máximos (componentes fuertemente conectadas, scc) de G , y para cada scc, G' , calcular las funciones S, L y $K_{G'}$ para formar el $PG G'$. Agregar todos los $PGs G'$ con más de un vértice al conjunto g_{nuevo} .

5. $g_{bueno} := \emptyset$

6. **Mientras que** $g_{nuevo} \neq \emptyset$ **Hacer**

7. Extraer $G' = (V', E')$ de g_{nuevo}

8. **Si** $\forall r \in V', K_{G'}(r) \neq \emptyset$ y no existe otro $G = (V, E)$ en g_{bueno} que satisfice: (1) $V' \subseteq V$; y (2) $S_{G'} \subset S_G$ **Entonces**

9. Agregar G' al conjunto g_{bueno}

10. **Fin Si**

11. **Para** cada $r \in V'$ **Hacer**

12. Calcular el subgrafo de poda de G' inducido en $V' \setminus \{r\}, G''$

13. Calcular todos los scc de G'' , y para cada scc, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el $PG G'''$. Agregar todos los $PGs G'''$ con más de un vértice al conjunto g_{nuevo}

14. **Fin Para**

15. **Fin Mientras que**

16. **Para** cada grafo $G' = (V', E') \in g_{bueno}$ **Hacer**

17. Agregar a \mathcal{D} el cerrojo $D = \bigcup_{r \in V'} S(r) \setminus K_{G'}(r) = S_{G'}$

18. **Fin Para**

19. **Fin**

las redes $SOAR^2$.

Definición 24. [La clase de redes $SOAR^2$ [Rov11]].

Sea $I_{\mathcal{N}} = \{1, 2, \dots, m\}$ un conjunto finito de índices. Una Red de Petri de la clase $SOAR^2$, es una red lugar/transición ordinaria, pura y conexa $\mathcal{N} = \langle P, T, \mathbf{C} \rangle$, tal que:

1. $P = P_0 \cup P_S \cup P_R$ es una partición tal que:
 - a) $P_S = \bigcup_{i \in I_{\mathcal{N}}} P_{S_i}$, $P_{S_i} \neq \emptyset$, $P_{S_i} \cap P_{S_j} = \emptyset$, $\forall i \neq j$.
 - b) $P_0 = \bigcup_{i \in I_{\mathcal{N}}} \{p_{0i}\}$.
 - c) $P_R = \{r_1, \dots, r_n\}$, $n > 0$.
2. $T = T_a \cup T_r$ es una partición tal que:
 - a) $T_a = \bigcup_{i \in I_{\mathcal{N}}} T_{ai}$, $T_{ai} \neq \emptyset$, $T_a = P_R^\bullet$, para todo $i, j \in I_{\mathcal{N}}$, $i \neq j$, $T_{ai} \cap T_{aj} = \emptyset$; y $\forall t \in T_a$, $|t^\bullet \cap P_R| = 1$, $|t^\bullet \cap P_S| = 0$.
 - b) $T_r = \bigcup_{i \in I_{\mathcal{N}}} T_{ri}$, $T_{ri} \neq \emptyset$, $T_r = {}^\bullet P_R$, para todo $i, j \in I_{\mathcal{N}}$, $i \neq j$, $T_{ri} \cap T_{rj} = \emptyset$, y $\forall t \in T_r$, $|t^\bullet \cap P_R| = 1$, $|t^\bullet \cap P_S| = 0$.
3. Para todo $i \in I_{\mathcal{N}}$, la subred \mathcal{N}_i generada por $P_{S_i} \cup \{p_{0i}\} \cup T_{ai} \cup T_{ri}$ es una máquina de estados fuertemente conexa, en la que cada ciclo contiene a p_{0i} .
4. Para cada $r \in P_R$ existe un único p -semiflujo mínimo, $\mathbf{y}_r \in \{0, 1\}^{|P|}$, tal que $\{r\} = \|\mathbf{y}_r\| \cap P_R$, $P_0 \cap \|\mathbf{y}_r\| = \emptyset$, y $P_S \cap \|\mathbf{y}_r\| \neq \emptyset$.
5. $P_S = \bigcup_{r \in P_R} (\|\mathbf{y}_r\| \setminus \{r\})$.
6. Para cada conjunto máximo de zonas solapadas la relación de precedencia, $<$, es una relación de orden total estricta.

En la definición anterior, los *lugares proceso* son los lugares pertenecientes al conjunto P_S . Los *lugares recurso* representan los recursos de la red los cuales pertenecen al conjunto P_R . Los estados inactivos de los procesos son representados mediante los lugares p_{0i} del conjunto P_0 y se les denominan *lugar de reposo* de la máquina de estados i -ésima. En la figura 5.5 se puede observar una red $SOAR^2$, los elementos de esta red son resumidos en la tabla 5.5.

Los lugares proceso que usan un recurso son denominados *usuarios del recurso*. A continuación se formaliza la definición del conjunto de usuarios de un recurso.

Definición 25. Sea $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red $SOAR^2$. Sea $r \in P_R$. El conjunto de usuarios de r es el soporte del p -semiflujo mínimo \mathbf{y}_r sin el lugar r : $\mathcal{H}_r = \|\mathbf{y}_r\| \setminus \{r\}$.

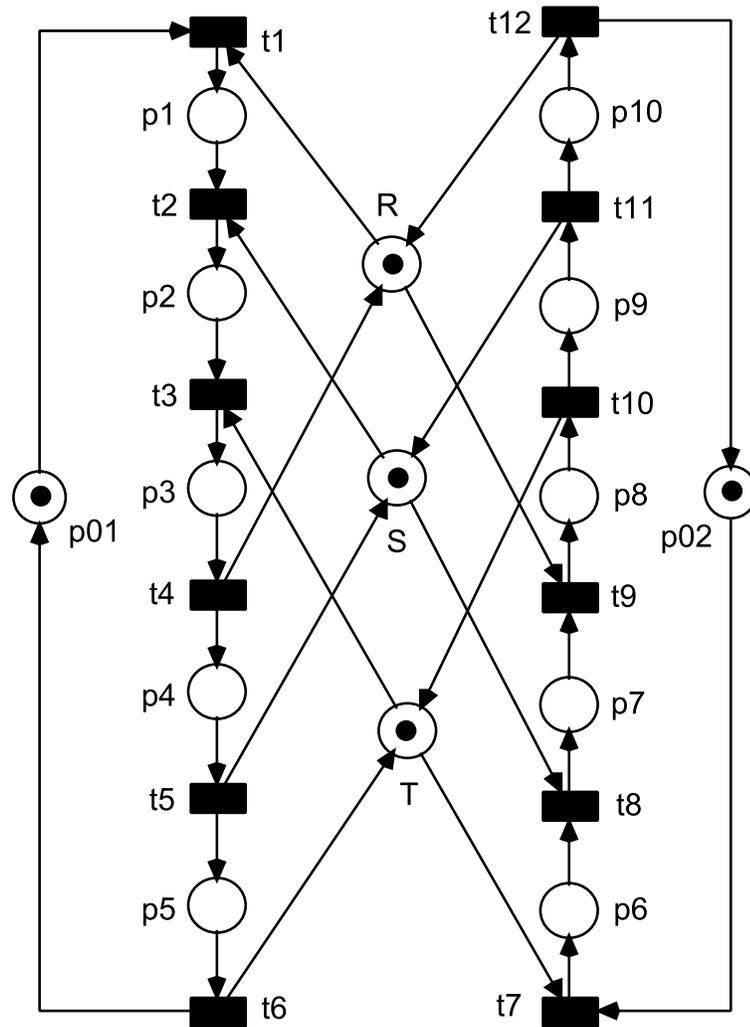


Figura 5.5: Red de Petri de la Clase $SOAR^2$.

Elementos
$P_0 = \{p01, p02\}$
$P_S = \{p1, p2, p3, p4, p5, p6, p7, p8, p9, p10\}$
$P_R = \{R, S, T\}$
$T_a = \{t1, t2, t3, t7, t8, t9\}$
$T_r = \{t4, t5, t6, t10, t11, t12\}$
$T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12\}$
$H(R) = \{p1, p2, p3, p8, p9, p10\}$
$H(S) = \{p2, p3, p4, p7, p8, p9\}$
$H(T) = \{p3, p4, p5, p6, p7, p8\}$

Cuadro 5.5: Elementos de la red $SOAR^2$ de la figura 5.5.

Esta definición puede extenderse de manera natural a los conjuntos de recursos $A \subseteq P_R : \mathcal{H}_A = \bigcup_{r \in A} \mathcal{H}_r$.

A continuación se define el *mercado inicial aceptable*[Rov11] para completar la definición de las redes $SOAR^2$ antes presentada.

Definición 26. Sea $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$ una red $SOAR^2$. Un mercado inicial \mathbf{m}_0 es aceptable para \mathcal{N} si, y solo si:

1. $\forall i \in I_{\mathcal{N}}, \mathbf{m}_0[p_{0i}] > 0$.
2. $\forall p \in P_S, \mathbf{m}_0[p] = 0$.
3. $\forall r \in P_R, \mathbf{m}_0[r] = 1$.

Basados en la definición de las redes $SOAR^2$ podemos indicar que los cerrojos mínimos de un recurso están formados por los soportes de sus p-semiflujos, esto se formalizó en el lema 9 del capítulo 2. A partir de lo antes explicado podemos decir que la clase \mathcal{D}^1 que contiene el conjunto de cerrojos mínimos de un recurso también se encuentra caracterizada y que la obtenemos directamente de la definición de las redes $SOAR^2$. Para ilustrar lo antes expuesto en la tabla 5.6 se resumen los cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 para la red de la figura 5.5.

Para estas redes la relación de poda entre el conjunto de cerrojos mínimos de un solo recurso presentada en el capítulo 3 (definición 7) puede ser utilizada, sin embargo teniendo en cuenta el apartado 4 de la definición de las redes $SOAR^2$ (definición 24.4) los cerrojos mínimos de un recurso no pueden podarse a sí mismo como se puede hacer en las redes S^4PR y que es formalizado en el lema 10 del capítulo 3. Esto impone una nueva restricción

Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1
$H(R) = \{p1, p2, p3, p8, p9, p10\}$
$H(S) = \{p2, p3, p4, p7, p8, p9\}$
$H(T) = \{p3, p4, p5, p6, p7, p8\}$

Cuadro 5.6: Cerrojos mínimos pertenecientes a la clase \mathcal{D}^1 de la red $SOAR^2$ de la figura 5.5.

a ser considerada en la representación de la relación de poda entre cerrojos de un sólo recurso. En efecto, esta relación de poda puede ser también representada utilizando el grafo de poda como se ha realizado con las clases anteriores, sin embargo al igual que en éstas se debe introducir la misma restricción que se estableció para las redes $L - S^3PR$ y S^3PR en la definición del grafo de poda. De esta forma, utilizaremos la definición 15 para definir el grafo de poda para las redes $SOAR^2$. Como resultado de la utilización de esta definición los grafos de poda para las redes $SOAR^2$ no tendrán bucles en sus vértices. Para calcular las funciones de etiquetado de asociadas al grafo de poda podemos utilizar la definición dada en el capítulo 3 (definición 9) y el algoritmo 1 debido a que en estas redes se debe realizar la búsqueda del conjunto de posibles lugares a eliminar como se realiza para las redes S^4PR . La figura 5.6 muestra el grafo de poda para la red $SOAR^2$ de la figura 5.5.

El conjunto de cerrojos mínimos para esta clase de redes se obtiene utilizando el algoritmo 2 presentado en el capítulo anterior. Al utilizar el algoritmo 2 en una red $SOAR^2$ no debemos olvidar que se debe utilizar la definición 15 para calcular el grafo de poda. Con el objetivo de mostrar de forma más clara la única modificación que sufre el algoritmo 2 para realizar el cálculo de los cerrojos mínimos para esta clase de red se escribe el algoritmo 7.

El cambio realizado debido a las restricciones impuestas por la clase de red corresponden al cálculo del grafo de poda (línea 3 del algoritmo), en donde, la complejidad para este cálculo se mantiene igual al apuntado en la sección 4.4 del capítulo 4. Por lo tanto, el costo computacional para el algoritmo 7 es igual al indicado en dicha sección.

5.5. Conclusión

En este capítulo se ha realizado la especialización para los algoritmos de cálculo de cerrojos mínimos para las clases de redes $L - S^3PR$, S^3PR y $SOAR^2$. Basados en la definición de cada una de estas redes, se han caracterizado los cerrojos mínimos de un

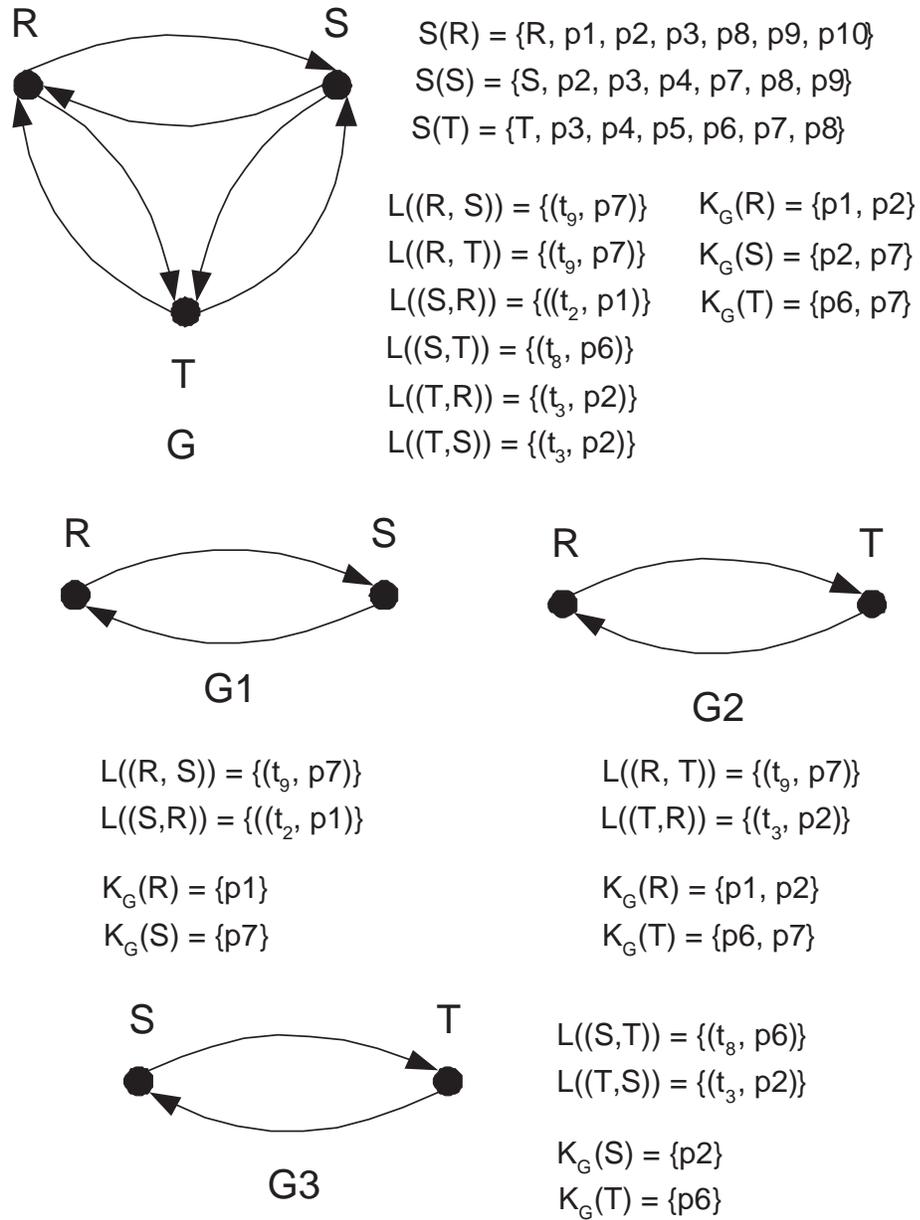


Figura 5.6: Grafo de poda de la red de la figura 5.5.

Algoritmo 7 Cálculo de los Cerrojos Mínimos de una red $SOAR^2$ \mathcal{N} .

Entrada: $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ – La red $SOAR^2$.

\mathcal{D}^1 – Los cerrojos mínimos con solo un recurso.

Salida: \mathcal{D} – Los cerrojos mínimos de \mathcal{N} conteniendo al menos un recurso.

1. **Inicio**
 2. $\mathcal{D} := \mathcal{D}^1$
 3. Calcular el grafo de poda $G = (V, E)$ de \mathcal{N} (definición 15), y las funciones S, L y K_G (definición 9).
 4. Calcular todos los subgrafos fuertemente conexos máximos (componentes fuertemente conexas, *cfcc*) de G , y para cada *cfcc*, G' , calcular las funciones S, L y $K_{G'}$ para formar el grafo de poda G' . Agregar todos los grafos de poda G' con más de un vértice al conjunto g_{nuevo} .
 5. $g_{bueno} := \emptyset$
 6. **Mientras que** $g_{nuevo} \neq \emptyset$ **Hacer**
 7. Extraer $G' = (V', E')$ de g_{nuevo}
 8. **Si** existe $r \in V'$ tal que $K_{G'}(r) = \emptyset$ **Entonces**
 9. Calcular $A := \{r | r \in V' \text{ y } K_{G'}(r) = \emptyset\}$
 10. Calcular el subgrafo de poda de G' inducido por $V' \setminus A$, denominado G'' .
 11. Calcular todos los *cfcc* de G'' , y para cada *cfcc*, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el grafo de poda G''' . Agregar todos los grafos de poda G''' con más de un vértice al conjunto g_{nuevo} .
 12. **Sino**
 13. Agregar G' al conjunto g_{bueno}
 14. **Para** cada $r \in V'$ **Hacer**
 15. Calcular el subgrafo de poda de G' inducido por $V' \setminus \{r\}$, G''
 16. Calcular todos los *cfcc* de G'' , y para cada *cfcc*, G''' , calcular las funciones S, L y $K_{G'''}$ para formar el grafo de poda G''' . Agregar todos los grafos de poda G''' con más de un vértice al conjunto g_{nuevo}
 17. **Fin Para**
 18. **Fin Si**
 19. **Fin Mientras que**
 20. Remover todos los grafos de g_{bueno} , $G = (V, E)$, para los cuales exista otro $G' = (V', E')$ en g_{bueno} que satisfice: (1) $V' \subseteq V$; y (2) $S_{G'} \subset S_G$
 21. **Para** cada grafo $G' = (V', E') \in g_{bueno}$ **Hacer**
 22. Agregar a \mathcal{D} el cerrojo $D = \bigcup_{r \in V'} S(r) \setminus K_{G'}(r) = S_{G'}$
 23. **Fin Para**
 24. **Fin**
-

solo recurso y la clase que contiene el conjunto de estos cerrojos.

De manera similar, se ha presentado el análisis de la relación de poda entre cerrojos mínimos de un recurso, en este apartado, basados en la definición de las redes $L - S^3PR$, S^3PR y $SOAR^2$ se ha introducido una nueva restricción a la relación de poda que consiste en que para estas redes un cerrojo no puede podarse a sí mismo como se realiza en las redes S^4PR . Esta restricción se transfiere a la definición del grafo de poda para las mismas y como resultado se obtiene que el grafo de poda para estas redes no puede tener bucles.

Además, a partir de la definición de las redes $L - S^3PR$ y S^3PR se obtiene una simplificación para el algoritmo de cálculo de la función de etiquetado de poda asociadas al grafo o subgrafos de poda. Utilizando los cambios antes apuntados, se indican los mismos en los respectivos algoritmos de cálculo de los cerrojos mínimos para estas redes. Por otro lado, para las redes $SOAR^2$ se define el cambio a realizar en el algoritmo de cálculo de los cerrojos mínimos. Finalmente, se presenta la complejidad para todos los algoritmos desarrollados.

Capítulo 6

Conclusiones

La presente memoria utiliza la teoría de grafos en el análisis de redes S^4PR . La necesidad de disponer de algoritmos eficientes para calcular los cerrojos mínimos u otros conjuntos de cerrojos para esta clase de redes es el motivo que ha conducido esta investigación. Para alcanzar el objetivo trazado se ha desarrollado como punto de partida las propiedades estructurales que poseen las redes S^4PR , esto se ha hecho debido a que en todos los trabajos anteriores se utilizaban las mismas propiedades que poseen las redes de Petri generales y no se tomaba ventaja de la estructura que este tipo de redes posee.

Otros aportes adicionales al antes mencionado en este apartado lo constituyen la caracterización de los cerrojos mínimos en función del conjunto de recursos que estos poseen y las cotas inferior y superior de cerrojos mínimos que una red S^4PR puede tener. Además, se presentan las cotas para las redes $L - S^3PR$, S^3PR y $SOAR^2$.

Basado en los resultados anteriores se ha desarrollado la herramienta denominada *Relación de Poda entre los cerrojos mínimos de un recurso* la cual nos permite determinar el conjunto de lugares que un cerrojo mínimo de un recurso puede podar a otro cerrojo mínimo de un recurso.

Para representar la Relación de Poda entre los cerrojos mínimos de un recurso se utiliza un grafo al cual se le ha llamado *Grafo de Poda* y tres funciones de etiquetado asociadas al mismo. Este grafo nos permite realizar el cálculo de cerrojos para un conjunto de recursos dado o el cálculo de cerrojos mínimos de una manera más eficiente. Esto lo consiguen los algoritmos al trabajar calculando y manipulando los subgrafos fuertemente conexos y por lo tanto la memoria requerida para dichos cálculos es del orden del tamaño del grafo de poda.

Finalmente, se presenta la especialización de los algoritmos de cálculo de cerrojos mínimos con más de un recurso para las redes $L - S^3PR$, S^3PR y $SOAR^2$ cuyo objetivo es obtener algoritmos que sean más eficientes que el desarrollado para las redes S^4PR .

Bibliografía

- [AT85] H. Alaiwan and J.M. Toudic. Recherche des semi-flots, des verroux et des trappes dans les réseaux de Petri. In *Technique et Science Informatiques*, pages 103–112, 1985.
- [BA96] K. Barkaoui and I.B. Abdallah. Analysis of a resource allocation problem in FMS using structure theory of Petri nets. In *M. Silva, R. Valette, and K. Takahashi, editors, Procs of the workshop: Manufacturing and Petri Nets*, pages 62–76, Osaka, Japan, June 1996.
- [BAEE97] I. Ben Abdallah, H.A. ElMaraghy, and T. ElMekkawy. A logic programming approach for finding minimal siphons in s3 pr nets applied to manufacturing systems. In *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on*, volume 2, pages 1710 –1715 vol.2, oct 1997.
- [BCZ97] K. Barkaoui, A. Chaoui, and B. Zouari. Supervisory control of discrete event systems based on structure of Petri nets. In *Procs of the 1997 Int. Conf. on Systems, Man and Cybernetics*, pages 3750–3755, Orlando (USA), October 1997.
- [Bes86] E. Best. Structure theory of petri nets: the free choice hiatus. In Springer Berlin / Heidelberg, editor, *Petri Nets: Central Models and Thier Properties*, volume 254 of *Lecture Notes in Computer Science*, pages 168–205, Berlin Heidelberg, 1986. Springer–Verlag.
- [BK90] Z. A. Banaszak and B. H. Krogh. Deadlock avoidance in flexible manufacturing systems with concurrency competing process flows. *IEEE Trans. on Robotics and Automation*, 6(6):724–734, December 1990.
- [BL89] K. Barkaoui and B. Lemaire. An effective characterization of minimal deadlocks and traps in petri nets based on graph theory. In *Proceeding of the 10th. International Conference on Theory and Applications of Petri Nets*, pages 1–22, Bonn, 1989.
- [BM92] Kamel Barkaoui and Michel Minoux. A polynomial-time graph algorithm to decide liveness of some basic classes of bounded petri nets. In K. Jensen, editor, *Application*

- and Theory of Petri Nets 1992*, volume 616 of *Lecture Notes in Computer Science*, pages 62–75. Springer Berlin / Heidelberg, 1992.
- [BM94] Erwin R. Boer and Tadao Murata. Generating basis siphons and traps of petri nets using the sign incidence matrix. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(4):266–271, April 1994.
- [BM08] J.A. Bondy and U.S.R. Murty. *Graph Theory*. Springer, 2008.
- [BPP96] Kamel Barkaoui and Jean-François Pradat-Peyre. On liveness and controlled siphons in petri nets. In Jonathan Billington and Wolfgang Reisig, editors, *Application and Theory of Petri Nets 1996*, volume 1091 of *Lecture Notes in Computer Science*, pages 57–72. Springer Berlin / Heidelberg, 1996.
- [CES71] E. G. Coffman, M. J. Elphick, and A. Shoshani. System deadlocks. *ACM Computer Surveys*, 3(2):67–78, 1971.
- [CFP02] R. Cordone, L. Ferrarini, and L. Piroddi. Characterization of minimal and basis siphons with predicate logic and binary programming. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 193 – 198, 2002.
- [CFP05] R. Cordone, L. Ferrarini, and L. Piroddi. Enumeration algorithms for minimal siphons in petri nets based on place constraints. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(6):844 – 854, nov. 2005.
- [CGH⁺] G. Chiola, R. Gaeta, A. Horvath, M. De Pierro, M. Beccuti, S. Baarir, and S. Cerotti. Graphical editor and analyzer for timed and stochastic petri nets.
- [Cha06] D. Y. Chao. Computation of elementary siphons in petri nets for deadlock control. *The Computer Journal*, 49(4):470–479, July 2006.
- [Cha08] D. Y. Chao. Incremental approach to computation of elementary siphons for arbitrary simple sequential processes with resources. *IET Control Theory and Applications*, 2(2):168–179, February 2008.
- [Cha09] Daniel Chao. Direct minimal empty siphon computation using mip. *The International Journal of Advanced Manufacturing Technology*, 45:397–405, 2009.
- [CHEP71] F. Commoner, A. W. Holt, S. Even, and A. Pnuelli. Marked directed graphs. *Journal of Computer and Systems Science*, 5(5):511–523, October 1971.
- [Col03] José-Manuel Colom. The resource allocation problem in flexible manufacturing systems. In W.M.P. van der Aalst and E. Best, editors, *Proc. of the 24th Int. Conf. on Applications and Theory of Petri Nets*, volume 2679 of *Lecture Notes in Computer Science*, pages 23–35, Eindhoven, Netherlands, June 2003. Springer-Verlag.

- [Com72] F.G. Commoner. *Deadlocks in Petri Nets*. Massachusetts Computer Assoc., Inc., 1972.
- [CRC10a] E.E. Cano, C.A. Rovetto, and J. Colom. On the computation of the minimal siphons of s4pr nets from a generating family of siphons. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8, sept. 2010.
- [CRC10b] Elia E. Cano, Carlos A. Rovetto, and José-Manuel Colom. An algorithm to compute the minimal siphons in S^4PR nets. *10th. Int. Workshop on Discrete Event Systems (WODES 2010)*, pages 18–23, August 30 - September 1 2010.
- [CS91] J.M. Colom and M. Silva. Convex geometry and semiflows in p/t nets. a comparative study of algorithms for computation of minimal p-semiflows. In Grzegorz Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 79–112. Springer Berlin / Heidelberg, 1991.
- [CX97] F. Chu and X. L. Xie. Deadlock analysis of petri nets using siphons and mathematical programming. *IEEE Trans. on Robotics and Automation*, 13(6):793–804, December 1997.
- [DE95] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.
- [Die06] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2006.
- [EC91] J. Ezpeleta and J. M. Couvreur. A new technique for finding a generating family of siphons, traps and st-components. application to colored petri nets. In *Proceeding of the 12th. International Conference on Application and Theory of Petri Nets*, pages 145–164, Aarhus, Denmark, June 1991.
- [ECM95] Joaquin Ezpeleta, José-Manuel Colom, and Javier Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 11(2):173–184, April 1995.
- [ECS93] J. Ezpeleta, J. M. Couvreur, and M. Silva. A new technique for finding a generating family of siphons, traps and st-components. application to colored petri nets. In *Advances in Petri Nets*, Lecture Notes in Computer Science, pages 126–147, Aarhus, Denmark, 1993. Springer–Verlag.
- [EGVC98] J. Ezpeleta, F. Garca-Vallés, and J. Colom. A class of well structured petri nets for flexible manufacturing systems. In Jörg Desel and Manuel Silva, editors, *Application and Theory of Petri Nets 1998*, volume 1420 of *Lecture Notes in Computer Science*, pages 64–83. Springer Berlin / Heidelberg, 1998.
- [ES92] Javier Esparza and Manuel Silva. A polinomial-time algorithm to decide liveness of bounded free choice nets. *Theoretical Computer Sciences*, 102(1):185–205, 1992.

- [Ezp93] Joaquin Ezpeleta. *Análisis y Síntesis de Modelos Libres de Bloqueos para Sistemas Concurrentes*. PhD thesis, Zaragoza. España, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Junio 1993.
- [GS08] J. Garrido and R. Schlesinger. *Principles of modern operating system*. Jones and Bartlett Publisher, Inc., 2008.
- [GV99] Fernando García Vallés. *Contributions to the Structural and Symbolic Analysis of Place/Transition Nets with Applications to Flexible Manufacturing Systems and Asynchronous Circuits*. PhD thesis, Zaragoza. España, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, 1999.
- [Hac72] M. Hack. Analysis of production schemata by petri nets. Technical Report TR-94, Massachusetts Institute of Technology, Cambridge, MA, USA, February 1972.
- [HC92] F. Hsieh and S. Chang. Deadlock avoidance controller synthesis for flexible manufacturing systems. In *Proc. of Rensselaer's 3rd Int. Conference on Computer Integrated Manufacturing*, pages 252–261, Troy, New York, May 1992.
- [HJXC01] Y. S. Huang, M. Jeng, X. L. Xie, and S. L. Chung. Deadlock prevention policy based on petri nets and siphons. *Int. J. Prod. Res.*, 39(3):283–305, March 2001.
- [JPH99] Muder Jeng, Maoyu Peng, and Yisheng Huang. An algorithm for calculating minimal siphons and traps of Petri nets. *Int. Journal of Intelligent Control and Systems*, 3(3):263–275, 1999.
- [JXH00] MuDer Jeng, Xiaolan Xie, and YiSheng Huang. Manufacturing modeling using process nets with resources. In *Procs IEEE International Conference on Robotics and Automation*, pages 2185–2190, San Francisco, USA, April 2000.
- [Lau87] Kurt Lautenbach. Linear algebraic calculation of deadlocks and traps. In Voss, Genrich, and Rozenberg, editors, *Concurrency and Nets Advances in Petri Nets*, pages 315–336, New York, 1987. Springer-Verlag.
- [LGC10] J.P. Lopez-Grao and J.M. Colom. The resource allocation problem in software applications: A Petri Net Perspective. *Proceedings of the International Workshop on Petri Nets and Software Engineering PNSE'10*, pages 7–21, June 2010.
- [LL07] Zhiwu Li and Ding Liu. A correct minimal siphons extraction algorithm from a maximal unmarked siphon of a Petri net. *The International Journal of Production Research*, 45:2161–2165, 2007.
- [LT79] K. Lautenbach and P. S. Thiagarajan. Analysis of a resource allocation problem using Petri nets. In J.C. Syre, editor, *1st. European Conf. on Parallel and Distributed Systems*, pages 1–17, 1979.

- [LZ04] Z. W. Li and M. C. Zhou. Elementary siphons of petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Trans. Conf. on Systems, Man and Cybernetics*, 34(part A):38–56, January 2004.
- [LZ08] Z. W. Li and M. C. Zhou. On siphon computation for deadlock control in a class of petri nets. *IEEE Trans. Conf. on Systems, Man and Cybernetics*, 38(part A):667–679, May 2008.
- [LZ09] Z. Li and M.C. Zhou. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. Advances in Industrial Control. Springer, 2009.
- [LZJ08] Zhiwu Li, MengChu Zhou, and MuDer Jeng. A maximally permissive deadlock prevention policy for fms based on petri net siphon control and the theory of regions. *Automation Science and Engineering, IEEE Transactions on*, 5(1):182 –188, jan. 2008.
- [Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
- [Pet81] J. L. Peterson. *Petri Nets Theory and The Modeling of Systems*. Prentice-Hall, 1981.
- [PR00] Jonghun Park and Spyros Reveliotis. Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems. *IEEE Trans. on Robotics and Automation*, 16(2):190–195, April 2000.
- [PR01a] Jonghun Park and S.A. Reveliotis. Algebraic deadlock avoidance policies for conjunctive/disjunctive resource allocation systems. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 70 – 76 vol.1, 2001.
- [PR01b] Jonghun Park and Spyros Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 46(10):1572–1583, October 2001.
- [PS89] J. L. Peterson and A. Silberschatz. *Sistemas Operativos. Conceptos fundamentales*. Editorial Reverté, 1989.
- [RCC10a] C. A. Rovetto, E. E. Cano, and J. M. Colom. Deadlock analysis in minimal adaptive routing algorithms using petri nets. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2619–2626, Istanbul, Turkey., October 2010.
- [RCC10b] C. A. Rovetto, E. E. Cano, and J. M. Colom. Deadlock control software for two automated guided vehicles using Petri nets. *Proceedings of the International Workshop on Petri Nets and Software Engineering PNSE'10*, pages 1–15, June 2010.

- [Rei85] W. Reisig. *Petri Nets. An Introduction*. EATCS Monographs on Theoretical Computer Science, W. Brauer, G. Rozendberg, A. Saloma, Eds., Springer-Verlag, 1985.
- [Rev98] Spyros A. Reveliotis. Accommodating fms operational contingencies through routing flexibility. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 573–579, Leuven, Belgium, May 1998.
- [Rev03] S. A. Reveliotis. On the Siphon-Based characterization of liveness in sequential resource allocation systems. In Springer Berlin / Heidelberg, editor, *Applications and Theory of Petri Nets*, volume 2679 of *LNCS*, pages 241–255. Springer Berlin / Heidelberg, 2003.
- [Rev05] S. A. Reveliotis. *Real-Time management of resource allocation Systems. A Discrete Event Systems Approach*. Springer Verlag, New York, USA, 2005.
- [Rov11] Carlos A. Rovetto. *Métodos basados en Redes de Petri para el diseño de algoritmos de encaminamiento adaptativos mínimos libres de bloqueos*. PhD thesis, Zaragoza. España, Departamento de Informática e Ingeniería de Sistemas Computacionales, Universidad de Zaragoza, Septiembre 2011.
- [SGG05] A. Silberschatz, P. Galvin, and G. Gagne. *Operating system concepts*. Mc Graw Hill, 2005.
- [Sif79] J. Sifakis. *Contrôle des systèmes asynchrones: concepts, propriétés, analyse statique*. PhD thesis, Thèse de l'Université Scientifique et Médical de Grenoble, 1979.
- [Sil85] M. Silva, editor. *Las Redes de Petri en la Automática y la Informática*. Editorial AC, Madrid, 1985.
- [Sta03] Peter H. Starke. Integrated net analyzer ina, July 2003.
- [TCE99] Fernando Tricas, José-Manuel Colom, and Joaquin Ezpeleta. A solution to the problem of deadlocks in concurrent systems using Petri nets and integer linear programming. In G. Horton, D. Moller, and U. Rude, editors, *Proc. of the 11th European Simulation Symposium*, pages 542–546, Erlangen, Germany, October 1999. The society for Computer Simulation Int.
- [TGVCE98] Fernando Tricas, Fernando García-Vallés, José-Manuel Colom, and Joaquin Ezpeleta. A structural approach to the problem of deadlock prevention in processes with resources. In *Proc. of the Int. Workshop on Discrete Event Systems (WODES'98)*, pages 273–278, August 1998.
- [TGVCE05] Fernando Tricas, Fernando Garca-Valls, José-Manuel Colom, and Joaquin Ezpeleta. A Petri net structure-based deadlock prevention solution for sequential resource allocation systems. In *In Proc. of the 2005 IEEE Int. Conf. on Robotics and Automatization*, pages 272–278, Barcelona, Spain, April 2005.

- [TM95] Fernando Tricas and Javier Martnez. An extension of the liveness theory for concurrent sequential processes competing for shared resources. In *Procs of the 1995 IEEE Int. Conf. on Systems, Man and Cybernetics.*, pages 3035–3040, Vancouver, BC, Canada, Oct 22-25 1995.
- [Tou81] J. M. Toudic. *Algorithmes d'analyse structurelle de réseaux de Petri*. PhD thesis, Université Pierre et Marie Curie (Paris VI), 1981.
- [Tri03] Fernando Tricas. *Analysis, Prevention and Avoidance of Deadlocks in Sequential Resource Allocation Systems*. PhD thesis, Zaragoza. España, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, May 2003.
- [Tut01] W. T. Tutte. *Graph Theory*. Cambridge University Press, 2001.
- [Wil86] H.P. Williams. Fourier's method of linear programming and its dual. *American Mathematical Monthly*, pages 681–695, November 1986.
- [WLJZ09] A.R. Wang, Z.W. Li, J.Y. Jia, and M.C. Zhou. An effective algorithm to find elementary siphons in a class of petri nets. In *2009 IEEE International Transactions on Systems, Man and Cybernetics*, pages 912–923, July 2009.
- [WLR⁺09] Yin Wang, Hongwei Liao, S. Reveliotis, T. Kelly, S. Mahlke, and S. Lafortune. Gadara nets: Modeling and analyzing lock allocation for deadlock avoidance in multithreaded software. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 497–4976, dec. 2009.
- [XHC96] K. Xing, B. Hu, and H. Chen. Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources. *IEEE Trans. on Automatic Control*, 41(2):289–295, February 1996.
- [XZW⁺11] Keyi Xing, MengChu Zhou, Feng Wang, Huixia Liu, and Feng Tian. Resource-transition circuits and siphons for deadlock control of automated manufacturing systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(1):74–84, jan. 2011.
- [YTW98] M. Yamauchi, S. Tanimoto, and T. Watanabe. Extracting siphons containing a specified set of places in a petri net. In *1998 IEEE International Conference on Systems, Man and Cybernetics*, pages 142–147, San Diego, USA, October 1998.

Apéndice A

Definiciones Básicas y Terminología sobre las Redes de Petri

En este apéndice se presentan las principales definiciones de los elementos relacionados con las Redes de Petri utilizados en esta memoria. El lector interesado en profundizar en este tema puede encontrar mayor información en: [Pet81, Sil85, Rei85, Mur89].

A.1. Conceptos Básicos y Terminología

Una *Red de Petri generalizada (RdPG)*, \mathcal{N} es una cuadrupla $\mathcal{N} = \langle P, T, Post, Pre \rangle$, donde,

1. P es un conjunto finito y no vacío de elementos llamados *lugares*.
2. T es un conjunto finito y no vacío de elementos llamados *transiciones*, tal que, $P \cap T = \emptyset$.
3. Pre ($Post$) es la función de incidencia previa (posterior):
 - $Pre: (P \times T) \rightarrow \mathbb{N}$.
 - $Post: (P \times T) \rightarrow \mathbb{N}$.

Una Red de Petri se representa gráficamente mediante un grafo orientado con dos tipos de nodos: lugares y transiciones. Los lugares se representan mediante círculos y las transiciones mediante rectángulos. Existe un arco orientado de un lugar p a una transición t si $Pre(p, t) \neq 0$. Análogamente, existe un arco de una transición t a un lugar p si

$Post(p, t) \neq 0$. Cada arco orientado se etiqueta por un entero natural, $Pre(p, t)$ ó $Post(p, t)$ que se denomina *peso del arco*. Por convenio, un arco no etiquetado posee un peso unitario.

Sea $\mathcal{N} = \langle P, T, Post, Pre \rangle$ una RdPG. Entonces, se puede definir alternativamente como $\mathcal{N} = \langle P, T, \Gamma, V \rangle$ donde:

1. $\Gamma \subseteq (P \times T) \cup (T \times P)$ es el grafo subyacente, de manera que existe un arco $(p, t) \in \Gamma$ sii $Pre(p, t) \neq 0$. Análogamente, existe un arco $(t, p) \in \Gamma$ sii $Post(p, t) \neq 0$.
2. $V : \Gamma \rightarrow \mathbb{N}$ se denomina una valuación de arcos, y se define como $V(p, t) = Pre(p, t)$ y $V(t, p) = Post(p, t)$.

Sea $\mathcal{N} = \langle P, T, Post, Pre \rangle$ una RdPG, y sean $t \in T$ y $p \in P$. Se definen los siguientes conjuntos:

1. Conjunto de lugares de entrada de $t : \bullet t = \{p \in P | Pre(p, t) \neq 0\}$.
2. Conjunto de lugares de salida de $t : t \bullet = \{p \in P | Post(p, t) \neq 0\}$.
3. Conjunto de transiciones de entrada de $p : \bullet p = \{t \in T | Post(p, t) \neq 0\}$.
4. Conjunto de transiciones de salida de $p : p \bullet = \{t \in T | Pre(p, t) \neq 0\}$.
5. Si $E \subseteq P \cup T$, $\bullet E = \bigcup_{x \in E} \bullet x$, y $E \bullet = \bigcup_{x \in E} x \bullet$.

Una RdP $\mathcal{N} = \langle P, T, Post, Pre \rangle$ se dice ordinaria sii $\forall p \in P, t \in T, Post(p, t) \in \{0, 1\}$ y $Pre(p, t) \in \{0, 1\}$.

Una *máquina de estados* es una red ordinaria tal que para cada transición $t \in T$, $|\bullet t| = |t \bullet| = 1$.

Sea $\mathcal{N} = \langle P, T, Post, Pre \rangle$ una RdPG. La matriz $C = Post - Pre$ se denomina *matriz de incidencia* de la red. También se suele denominar *matriz de flujo* de la red.

1. Un lugar se dice puro sii $\bullet p \cap p \bullet = \emptyset$.
2. Una transición se dice pura sii $\bullet t \cap t \bullet = \emptyset$.
3. Una RdP se dice pura sii todos sus lugares (o transiciones) son puros.

Un *camino* π de \mathcal{N} es una secuencia de nodos $\pi = x_1 x_2 \dots x_n$ pertenecientes a $P \cup T$ tal que $x_{i+1} \in x_i \bullet$, para todo $i \in \{1, \dots, n-1\}$. Un camino es *simple* si todos los nodos son diferentes. Un *circuito* es un camino tal que $x_1 = x_n$. Un *circuito simple* es un circuito tal que todos los nodos, excepto el primero y el último son diferentes.

A.1.1. Funcionamiento de una Red de Petri

Un *marcado* m de una RdP \mathcal{N} es una función: $m : P \rightarrow \mathbb{N}$.

Las marcas son representadas por puntos negros dentro de los lugares.

El *soporte* de un marcado, $\|m\|$, es el conjunto de lugares los cuales están marcados en m , i.e. $\|m\| = \{p \in P \mid m[p] \neq 0\}$.

Una *red de Petri marcada* es el par $\langle \mathcal{N}, m_0 \rangle$, donde \mathcal{N} es una RdP y m_0 es un *marcado inicial* de la red.

Se dice que \mathcal{N} es la estructura del sistema mientras que m_0 representa el estado del sistema.

Sea $\langle \mathcal{N}, m_0 \rangle$ una RdP marcada. Una transición $t \in T$ está *sensibilizada* (también *habilitada*) sii $\forall p \in \bullet t . m_0[p] \geq \text{Pre}(p, t)$, lo cual es denotado por $m_0[t]$.

El *disparo* de una transición sensibilizada t cambia el estado del sistema a $\langle \mathcal{N}, m_1 \rangle$, donde $\forall p \in P . m_1[p] = m_0[p] + C[p, t]$, esto es denotado por $m_0[t]m_1$.

Una *secuencia de disparos* σ de $\langle \mathcal{N}, m_0 \rangle$ es una secuencia de transiciones no-vacia $\sigma = t_1 t_2 \dots t_k$ tal que $m_0[t_1]m_1[t_2] \dots m_{k-1}[t_k]$. El disparo de σ es denotado por $m_0[\sigma]t_k$.

Se denomina vector *característico* asociado $\bar{\sigma}$ de una secuencia de disparo σ a la aplicación $\bar{\sigma} : T \rightarrow \mathbb{N}^{|T|}$ tal que $\bar{\sigma}(t)$ es el número de veces que la transición t aparece en la secuencia. $\bar{\sigma}$ se denomina también *vector contador de disparos*. El soporte de $\bar{\sigma}$ es denotado por $\|\bar{\sigma}\|$.

Un marcado m es *alcanzable* de $\langle \mathcal{N}, m_0 \rangle$ sii existe una *secuencia de disparos* σ tal que $m_0[\sigma]m$. El *conjunto de alcanzabilidad* $RS(\mathcal{N}, m_0)$ es el conjunto de marcados alcanzables, i.e. $RS(\mathcal{N}, m_0) = \{m \mid \exists \sigma . m_0[\sigma]m\}$.

A.1.2. Componentes Estructurales de Redes de Petri

Un *p-semiflujo* (*t-semiflujo*) es un vector $Y \in \mathbb{N}^{|P|}$, $Y \neq \mathbf{0}$ ($X \in \mathbb{N}^{|T|}$, $X \neq \mathbf{0}$), el cual es un anulador izquierdo (derecho) de la matriz de incidencia, $Y \cdot C = \mathbf{0}$ ($C \cdot X = \mathbf{0}$).

El soporte de un p-semiflujo (t-semiflujo) es denotado $\|Y\|$ ($\|X\|$), y sus lugares (transiciones) están cubiertas por Y (X).

Un *p-semiflujo mínimo* (*t-semiflow mínimo*) es un p-semiflujo (t-semiflujo) tal que el máximo común divisor de sus componentes no-nulas es uno y su soporte $\|Y\|$ ($\|X\|$) no contiene el soporte de ningún otro p-semiflujo (t-semiflujo).

Sea $\mathcal{N} = \langle P, T, C \rangle$ una RdP. Un subconjunto de lugares $D \subseteq P$ es un *cerrojo* si, y solo si, $\bullet D \subseteq D^\bullet$.

Sea $\mathcal{N} = \langle P, T, C \rangle$ una RdP. Un subconjunto de lugares $D \subseteq P$ es una *trampa* si, y solo si, $D^\bullet \subseteq \bullet D$.

A.2. Propiedades de las Redes de Petri

A.2.1. Propiedades Estructurales

Una RdP \mathcal{N} es *conservativa* (*consistente*) sii cada lugar (transición) está cubierto por un p-semiflujo (t-semiflujo).

Un p-semiflujo, y define la siguiente propiedad de invarianza:

$$\forall \mathbf{m}_0 \cdot \forall \mathbf{m} \in RS(\mathcal{N}, \mathbf{m}_0) \cdot \mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m}_0$$

(ley de comportamiento cíclico).

A.2.2. Propiedades Dinámicas

Una transición $t \in T$ es *viva* sii para cada marcado alcanzable $m \in RS(\mathcal{N}, m_0)$, $\exists m' \in RS(\mathcal{N}, m)$ tal que $m'[t]$.

El sistema $\langle \mathcal{N}, m_0 \rangle$ es *vivo* sii cada transición es viva. De otra manera, $\langle \mathcal{N}, m_0 \rangle$ es *no-viva*.

Una transición $t \in T$ es *muerta* sii no existe un marcado alcanzable $m \in RS(\mathcal{N}, m_0)$ tal que $m[t]$.

El sistema $\langle \mathcal{N}, m_0 \rangle$ es un *bloqueo total* sii cada transición está muerta, i.e. ninguna transición está sensibilizada.

El sistema $\langle \mathcal{N}, m_0 \rangle$ está *libre de bloqueo* sii $\forall m \in RS(\mathcal{N}, m_0)$ existe al menos una transición disparable.

Apéndice B

Definiciones Básicas sobre los Grafos

En este apéndice se presentan las principales definiciones de los elementos relacionados con los grafos utilizados en esta memoria. El lector interesado en profundizar en este tema puede encontrar mayor información en: [Tut01, Die06, BM08].

Un *Digrafo o Grafo Dirigido* es una terna $G = (V, E, \varphi)$ con $V \neq \emptyset$ donde:

- $V = \{v_1, v_2, \dots, v_n\}$: conjunto de vértices o nodos.
- $E = \{e_1, e_2, \dots, e_n\}$: conjunto de aristas o arcos.
- $\varphi : A \rightarrow V \times V$ función de incidencia. En donde, si $\varphi(e) = (v, w)$ se dice que
 - los vértices v y w son adyacentes.
 - e incide positivamente en w y negativamente en v .
 - v es extremo inicial de la arista e , w es extremo final de e .

Un *multidigrafo o multigrafo dirigido* es un grafo $G = (V, E)$ formado por un conjunto finito de vértices V y una familia finita E de aristas orientadas

$$E = \{e_i\}_{i \in I}$$

donde I es un conjunto finito y $\forall i \in I$ se verifica que $e_i \in V \times V$.

Un *subgrafo* de un grafo $G = (V, E)$ es un grafo $H = (V', E')$ tal que $V' \subseteq V$ y $E' \subseteq E$.

Obsérvese que esta definición se extiende de forma natural a multidigrafos, siendo E' una subfamilia de E .

Sea $G = (V, E)$ un digrafo, y $\emptyset \neq U \subseteq V$. Un *subgrafo inducido* por U es aquel subgrafo de G cuyos vértices forman U y tal que contiene a todos los lados tomados de G de la forma $(x, y), \forall x, y \in U$.

Un digrafo posee *aristas paralelas* sii φ no es inyectiva. Es decir, dado $e_1 \in E$ y $e_2 \in E$, e_1 y e_2 son aristas paralelas sii $\varphi(e_1) = \varphi(e_2)$.

Un digrafo posee un *lazo o bucle* si existe $e \in E$ tal que $\varphi(e) = (v, v)$.

Un *camino de longitud n* entre los vértices a y b de un multigrafo dirigido es una sucesión finita (e_0, \dots, e_{n-1}) de aristas del multigrafo dirigido:

$$e_0 = (v_0, v_1), e_1 = (v_1, v_2), \dots, e_{n-1} = (v_{n-1}, v_n)$$

de manera que $v_0 = a$, $v_n = b$ y cada arista sucesiva empieza donde terminó la anterior. Cuando se trata de un digrafo, el camino (e_0, \dots, e_{n-1}) queda perfectamente determinado por la sucesión de vértices

$$(a, v_1, v_2, \dots, v_{n-1}, b).$$

Diremos que el camino anterior *pasa por* (o *atraviesa*) los vértices $a, v_1, v_2, \dots, v_{n-1}, b$.

Se dice que un camino es un *circuito* si es cerrado, esto es, empieza y termina en el mismo vértice, es decir, si $a = b$.

Se dice que un camino es *simple* si no contiene a la misma arista más de una vez.

Se dice que un *grafo es conexo* si para cada par de vértices distintos existe un camino que los une. En caso contrario se dice que es un grafo no conexo o disjunto. La conectividad define un conjunto finito de *componentes conexas* (que son los subgrafos conexos más grandes que hay en el grafo).

Se dice que un *digrafo es conexo* si lo es el grafo no dirigido correspondiente. En caso contrario se dice que es un digrafo no conexo o disjunto.

Se dice que un *digrafo es fuertemente conexo* si para cada par de vértices distintos existe un camino orientado que los une.

Componentes fuertemente Conexos de un Digrafo: son sus subgrafos máximos fuertemente conexos.