

## Trabajo Fin de Máster

Desarrollo de un algoritmo basado en la pre-asignación de buffers que permita la vivacidad de sistemas de red DSSP inicialmente no vivos.

Autor/es

Daniel Clavel Villagrassa

Director/es

Cristian Mahulea  
Manuel Silva Suarez

Universidad de Zaragoza / EINA  
2016

# TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Daniel Clavel Villagrasa,

con nº de DNI 17765413-Y en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster, (Título del Trabajo)

Desarrollo de un algoritmo basado en la preasignación de buffers que permite la vivacidad de sistemas de red DSSP inicialmente no vivos.

---

---

---

---

, es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 23/06/2016

Fdo: Daniel Clavel Villagrasa

---

# Agradecimientos

---

Agradecer en primer lugar a todos los profesores, investigadores y colaboradores del Departamento de Informática e Ingeniería de Sistemas con quienes comparto mi tiempo y de una forma u otra me ayudan desinteresadamente.

También quiero agradecer al director, C. Mahulea, responsable del proyecto que me animó a dar los primeros pasos en el mundo de la investigación. Por todas las horas que ha dedicado a resolver los problemas que han ido surgiendo.

Finalmente mis agradecimientos al co-director, M. Silva, por su tiempo invertido en mejorar este proyecto.



---

# **Desarrollo de un algoritmo basado en la preasignación de buffers que permita la vivacidad de sistemas de red DSSP inicialmente no vivos.**

---

## **Resumen**

Deterministically Synchronized Sequential Process (DSSP) definen una subclase de sistemas descritos con Redes de Petri (RdP). Están compuestos por un conjunto de máquinas de estado (también llamadas agentes) que cooperan a través del paso asíncrono de mensajes. La estructura modular de las redes DSSP permite obtener fuertes resultados analíticos (por ejemplo, el teorema de rango proporciona condiciones necesarias y suficientes para asegurar vivacidad estructural). Este trabajo considera el problema de forzar estructuras vivas a partir de otras de tipo DSSP estructuralmente no vivas. Para algunas estructuras particulares de tipo DSSP en las cuales el teorema de rango demuestra que son estructuralmente no vivas, se provee un algoritmo basado en la preasignación de los buffers que la asegura. El algoritmo ha sido aplicado a DSSP que modelan sistemas reales de producción y salud. Se ha observado que en los sistemas de producción (lógica de consumo/producción de ítems) el algoritmo además de convertir la red viva, mantiene el significado físico del modelo. Por contra, en los sistemas de salud (lógica de producción/consumo de mensajes) al aplicar el algoritmo la red resultante es viva pero puede perderse el propósito del modelo.



---

# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Redes de Petri: notación, conceptos y tipos</b>	<b>5</b>
2.1.	Conceptos básicos . . . . .	5
2.2.	Subclases de Redes de Petri . . . . .	7
2.3.	Conceptos estructurales y comportamentales . . . . .	8
<b>3</b>	<b>Redes DSSP</b>	<b>13</b>
3.1.	Definición y teorema de rango . . . . .	13
3.2.	Aplicación a sistemas reales . . . . .	14
3.2.1.	Sistemas productivos . . . . .	14
3.2.2.	Sistemas de salud . . . . .	17
3.3.	Motivación del trabajo: bloqueos en redes DSSP . . . . .	17
3.4.	Vivacidad en DSSP . . . . .	20
3.4.1.	Método de los sifones malos . . . . .	20
3.4.2.	Método de preasignación de buffers . . . . .	22
<b>4</b>	<b>Desarrollo del algoritmo</b>	<b>27</b>
4.1.	Aplicación del algoritmo a $TSF_1$ . . . . .	29
4.2.	Aplicación del algoritmo a $TSF_2$ . . . . .	36
4.3.	Aplicación del algoritmo a $TSF_3$ . . . . .	40
<b>5</b>	<b>Conclusiones</b>	<b>45</b>
	<b>Bibliografía</b>	<b>47</b>

---

# Índice de figuras

---

2.1. Ejemplo RdP (izq) y su grafo de alcanzabilidad (dcha) . . . . .	7
2.2. Ejemplo de sifón (izq) y de trampa (drch) . . . . .	11
3.1. Layout de una línea de producción . . . . .	15
3.2. Modelo DSSP vivo de la línea de producción de la Fig. 3.1 . . .	16
3.3. Ejemplo DSSP sistema de salud . . . . .	18
3.4. Modelo DSSP de sistema productivo de mesas A y B . . . . .	20
3.5. Lugar de monitorización ( $p_m$ ) que impide el marcado $\mathbf{m}'$ . . . . .	22
3.6. Preasignación de buffers en la RdP de la Fig. 3.4 . . . . .	23
3.7. Preasignación de los buffers en la DSSP de la Fig. 3.3 . . . . .	24
4.1. Comprobación de vivacidad en DSSP. . . . .	28
4.2. Preasignación desde una perspectiva local frente a una global. . . . .	32
4.3. DSSP con $TSF_2$ . . . . .	38
4.4. Combinación al añadir arcos de lectura . . . . .	39
4.5. DSSP con $TSF_3$ disjuntos . . . . .	41
4.6. DSSP con $TSF_3$ no disjuntos y SSP estructuralmente no viva . . . . .	42
4.7. DSSP con $TSF_3$ no disjuntos y SSP estructuralmente viva . . . . .	43

---

# Índice de cuadros

---

4.1.	T-semiflujos locales de la red de la Fig. 3.4 . . . . .	29
4.2.	T-semiflujo locales en la red de la Fig 4.2 . . . . .	31
4.3.	T-semiflujo locales en la red de la Fig 4.5 . . . . .	40



# Capítulo 1

---

## Introducción

---

Las RdP constituyen una familia de formalismos apropiado para el modelado de sistemas de eventos discretos (sistemas en los que las variables de estado toman valores discretos de un conjunto que se puede contar, no necesariamente finito). Su estado distribuido es un vector de números enteros no negativos. Esto es una ventaja importante con respecto a otros formalismos de modelado de sistemas discretos, como los autómatas, donde el espacio de estados es un conjunto de símbolos no estructurados de estados globales. Esta propiedad ha sido explotada para desarrollar muy diversas técnicas de análisis que no requieren la enumeración del espacio de estados (análisis estructural). Desde una perspectiva de modelado, una característica clave de las RdP es su capacidad de representación gráfica y visualización del modelo.

Las DSSP [10] son comúnmente usadas para modelar sistemas distribuidos formados por diversos agentes cooperativos. Esta cooperación es realizada mediante un conjunto de buffers desde/hacia los cuales los agentes consumen/producen items. Existen dos restricciones en la asignación de los buffers: (1) *las marcas de un buffer dado pueden ser consumidas únicamente por un agente particular (los buffers son de salida privada)* (2) *un buffer nunca puede condicionar decisiones internas del agente de destino, solo un progreso secuencial*. En sistemas productivos, los agentes pueden representar diferentes estaciones de trabajo donde se procesan los productos, mientras que las marcas de los buffers, modelan la disponibilidad o requerimiento de materia prima o productos intermedios durante el proceso productivo.

En cuanto a los sistemas de salud, en [1] se propone un sistema de ayuda a la decisión para la gestión de hospitales a través del modelado de

guiás clínicas usadas para el tratamiento de diferentes patologías médicas. El modelo se basa en Unified Modeling Language (UML) que es trasladado a RdP coloreadas para el análisis. Ya que los sistemas de salud son complejos debido a su tamaño y estructura, dependiendo de diferentes visiones y expectativas, el sistema puede ser tratado desde diferentes perspectivas que se llaman *facetas*. Dos posibles facetas son identificadas: (1) *faceta de gestión de recursos* y (2) *faceta de comunicación entre guías clínicas*. Los módulos se obtienen en ambos casos mediante el decolorado del modelo inicial, quitando la información irrelevante. En la primera faceta, se obtienen modelos característicos de los sistemas de asignación de recursos (máquinas de estado interconectadas mediante los recursos compartidos modelan el flujo de pacientes) mientras en el segundo caso redes DSSP son obtenidas (máquinas de estado interconectadas mediante buffers que modelan los canales de comunicación). Nos centraremos en la vivacidad de las redes DSSP resultantes de la faceta de comunicación entre guías clínicas. Esto es, evitar que algún paciente quede atrapado en un protocolo médico sin posibilidad de continuar.

La vivacidad estructural de sistemas DSSP ha sido estudiada en la literatura [10], pero no existen trabajos enfocados a conseguir vivacidad estructural en sistemas DSSP inicialmente no vivos. La primera idea para conseguir vivacidad es usar el método basado en sifones empleado en sistemas de asignación de recursos [3, 7, 5, 2]. Este método consiste en calcular el conjunto de sifones malos (sifones que pueden vaciarse) y prevenir su vaciado, por ejemplo, usando lugares de monitorización. Sin embargo, nosotros ilustramos en este trabajo, que aplicando este método a sistemas DSSP, los nuevos buffers (correspondientes con los lugares de monitorización), además de condicionar decisiones internas en los agentes, pueden no ser privados. De este modo se están incumpliendo las dos restricciones que definen las redes DSSP. Este trabajo provee un método diferente que además de no añadir nuevos lugares, mantiene la restricción de los sistemas DSSP de privacidad de los buffers (cada buffer tiene un único agente como destino), aunque no se mantiene la restricción que impide a los buffers influir en decisiones internas de los agentes. La idea básica del método propuesto consiste en que cada vez que una transición en conflicto de un agente es disparada, al menos existe un T-semifluido local (un componente local repetitivo) que puede ser disparado hasta el final. Esto significa que todos los buffers de entrada a dicho T-semifluido contienen marcas suficientes para disparar sus transiciones. Obviamente, no es suficiente considerar los T-semiflujos locales, sino que es necesaria una perspectiva global del sistema y los T-semiflujos globales

deben de ser considerados. Se da un algoritmo que para algunas estructuras de T-semiflujos locales/globales asegura la vivacidad estructural del sistema. En ciertas estructuras particulares el algoritmo elimina los bloqueos, mientras que para estructuras generales, algunos pasos adicionales son requeridos.

El trabajo está organizado del siguiente modo: el Capítulo 2 introduce resultados previos, notación y tipos de Redes de Petri. En el Capítulo 3 se muestra la definición de los sistemas DSSP, la propiedad del teorema de rango, aplicaciones reales, situaciones de bloqueo y posibles métodos para su prevención. Finalmente se da la idea principal del algoritmo propuesto. En el Capítulo 4 se desarrolla formalmente el algoritmo y se muestran las estructuras en las que es aplicable. El Capítulo 5 presenta las conclusiones obtenidas, así como futuros trabajos.



## Capítulo 2

---

# Redes de Petri: notación, conceptos y tipos

---

Las RdP son un formalismo bien conocido para modelar, analizar y diseñar sistemas de eventos discretos. Los sistemas de eventos discretos que se caracterizan por ser concurrentes, asíncronos, distribuidos, paralelos, no deterministas y/o estocásticos se pueden modelar y analizar por medio de las RdP. Similar a grafos, diagramas de bloques y redes de flujo, como una herramienta gráfica, las RdP pueden ser útiles en la ayuda a la comunicación visual. Además, en este tipo de redes, se utilizan marcas para simular las actividades dinámicas y simultáneas de sistemas. Como herramienta matemática, es posible establecer ecuaciones de estado, ecuaciones algebraicas, y otros modelos matemáticos que rigen el comportamiento de los sistemas.

Se asume que el lector está familiarizado con las RdP (ver libros [8, 6, 4]). El principal objetivo de esta sección es fijar la notación y recordar el material requerido para este trabajo.

### 2.1. Conceptos básicos

En este subapartado se muestra al lector la definición formal de las RdP y los sistemas de RdP, así como la ecuación de estado que permite modelar matemáticamente la evolución de la RdP.

**Redes y Sistemas.** Nosotros indicamos una red P/T (lugares/transiciones) como  $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ , donde  $P$  y  $T$  son dos conjuntos no vacíos y disjuntos de *lugares* y *transiciones*, y  $\mathbf{Pre}, \mathbf{Post} \in \mathbb{N}_{\geq 0}^{|P| \times |T|}$  son pre y post

*matrices de incidencia* que representan las conexiones entre transiciones y lugares o entre lugares y transiciones. Por ejemplo,  $\mathbf{Post}[p, t] = w$  significa que existe un *arco* desde  $t$  hacia  $p$  con *peso* (o multiplicidad)  $w$ . Por otro lado,  $\mathbf{Pre}[p, t] = w$  significa que existe un arco de peso  $w$  desde el lugar  $p$  a la transición  $t$ . Cuando todos los arcos tienen peso igual a uno, la red se llama *ordinaria*. Para los conjuntos de nodos pre y post, se usa la notación convencional de puntos, e.g.,  $\bullet t = \{p \in P \mid \mathbf{Pre}[p, t] \neq 0\}$ . Si  $\mathcal{N}'$  es una subred de  $\mathcal{N}$  definida mediante  $P' \subset P$  y  $T' \subset T$ , entonces  $\mathbf{Pre}' = \mathbf{Pre}[P', T']$  y  $\mathbf{Post}' = \mathbf{Post}[P', T']$ .

Un *marcado* es un vector de valores naturales de tamaño  $|P|$  asignando a cada lugar un numero natural de marcas. Un sistema P/T es una pareja  $\varphi = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ , donde  $\mathbf{m}_0$  es el *marcado inicial*. Una transición  $t$  está *habilitada* en  $\mathbf{m}$  si  $\mathbf{m} \geq \mathbf{Pre}[P, t]$  es decir, todos los lugares de entrada a  $t$  tienen un numero de marcas mayor o igual que el peso del arco que los une con la transición; su disparo lleva a un nuevo marcado  $\mathbf{m}' = \mathbf{m} + \mathbf{C}[P, t]$ , donde  $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$  es la matriz de flujo de marcas de la red. Este hecho es denotado mediante  $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ . Una *secuencia de ocurrencias* desde  $\mathbf{m}$  es una secuencia de transiciones  $\sigma = t_1 \dots t_k \dots$  tal que  $\mathbf{m} \xrightarrow{t_1} \mathbf{m}_1 \dots \mathbf{m}_{k-1} \xrightarrow{t_k} \dots$ . El conjunto de todos los marcados alcanzables, o *conjunto de alcanzabilidad*, desde  $\mathbf{m}$ , se denota como  $RS(\mathcal{N}, \mathbf{m})$ . La relación de alcanzabilidad es convencionalmente denotada mediante *grafo de alcanzabilidad*  $RG(\mathcal{N}, \mathbf{m})$  donde los nodos son los marcados alcanzables y hay un arco etiquetado  $t$  desde el nodo  $\mathbf{m}'$  hasta  $\mathbf{m}''$  si  $\mathbf{m}' \xrightarrow{t} \mathbf{m}''$ .

**Ecuación de transición.** Representando mediante  $\boldsymbol{\sigma} \in \mathbb{N}_{\geq 0}^{|P| \times |T|}$  el *vector contador de disparos* de  $\sigma$ , donde  $|T|$  es el tamaño de  $\boldsymbol{\sigma}$ ,  $\boldsymbol{\sigma}[t_i]$  es el número de veces que  $t_i$  aparece en  $\sigma$ . Dado  $\sigma$  tal que  $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$ , entonces

$$\mathbf{m}' = \mathbf{m} + \mathbf{C} \cdot \boldsymbol{\sigma}.$$

Esto se conoce como *ecuación de transición* de  $\varphi$  porque cada estado (marcado) alcanzable lo satisface. Sin embargo, no necesariamente un vector que satisfaga la ecuación de estado es realmente un marcado alcanzable, porque la ecuación de estado no comprueba la disparabilidad de una secuencia con el vector contador de disparos  $\boldsymbol{\sigma}$ . Tales vectores se denominan *marcados espurios* [11].

La parte izquierda de la Fig. 2.1 es una RdP  $\mathcal{N}$  donde dos procesos ( $p_2$  y  $p_4$ ) compiten por un recurso compartido ( $p_5$ ). El conjunto de lugares es  $P = \{p_1, p_2, p_3, p_4, p_5\}$ , el conjunto de transiciones es  $T = \{t_1, t_2, t_3, t_4\}$ , y

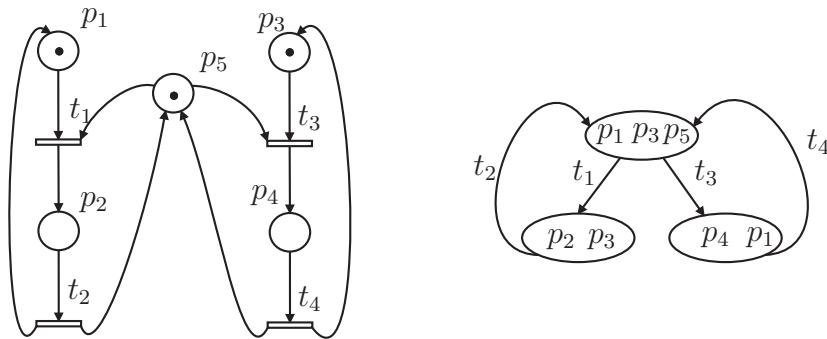


Figura 2.1: Ejemplo RdP (izq) y su grafo de alcanzabilidad (dcha)

las matrices de incidencia:

$$\mathbf{Pre} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \mathbf{Post} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

El marcado inicial es  $\mathbf{m}_0 = (1 \ 0 \ 1 \ 0 \ 1)^T$  y las transiciones  $t_1$  y  $t_3$  están habilitadas en  $\mathbf{m}_0$  ya que  $\mathbf{Pre}[P, t_1] > \mathbf{m}_0$  y  $\mathbf{Pre}[P, t_3] > \mathbf{m}_0$ . El disparo de  $t_1$  conduce al marcado  $\mathbf{m}' = (0 \ 1 \ 0 \ 0 \ 1)^T$ , es decir,  $\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}'$ . El grafo de alcanzabilidad  $RG(\mathcal{N}, \mathbf{m}_0)$  esta representado en la parte derecha de la Fig. 2.1\*.

## 2.2. Subclases de Redes de Petri

Existen multitud de subclases de RdP definidas a través de restricciones en su estructura. Estas restricciones permiten un análisis más sencillo de algunas propiedades, pero por el contrario, limitan la capacidad de modelado. En este apartado se van a ver aquellas estructuras básicas, poniendo especial interés en las que son más relevantes para el proyecto:

- $\mathcal{N}$  es *ordinaria* si todos los pesos de los arcos son unitarios, es decir,  $\forall p_i \in P, \forall t_j \in T, \mathbf{Pre}[p_i, t_j] \leq 1$  y  $\mathbf{Post}[p_i, t_j] \leq 1$ ;

---

\*En este trabajo se usa la notación multiconjunto para vectores. Por ejemplo,  $\mathbf{m}_0 = p_1 + p_3 + p_5$  es un vector donde los únicos elementos distintos de cero son  $p_1, p_3$  y  $p_5$  y su valor es el número que tienen delante.

- $\mathcal{N}$  es una *Maquina de estados* (SM) si es ordinaria y  $\forall t \in T : |\bullet t| = |\bullet^t| = 1$ . Esto es, cualquier transición tiene un único lugar de entrada y un único lugar de salida. Las máquinas de estado permiten el modelado de decisiones (conflictos) y atribuciones (un lugar con varias transiciones de entrada). Son siempre conservativas y el único P-semiflujo mínimo es 1; vivacidad es equivalente a una RdP fuertemente conexa y un marcado inicial no vacío. El concepto de maquina de estado, considerado como una subclase de RdP, es más general que los clásicos diagramas de estado o grafos de estado, ya que puede tener más de una marca.
- $\mathcal{N}$  es un *Grafo Marcado* (MG) si es ordinaria y  $\forall p \in P : |p^\bullet| = |\bullet p| = 1$ .
- $\mathcal{N}$  es *libre de sincronizaciones* (JF) si  $\forall t \in T, |\bullet t| \leq 1$ ;
- $\mathcal{N}$  es un *MG con peso* si  $\forall p \in P : |p^\bullet| = |\bullet p| = 1$ ;
- $\mathcal{N}$  es una red *libre de elecciones* (CF) si  $\forall p \in P, |p^\bullet| \leq 1$ ;
- $\mathcal{N}$  es una red *con elecciones libres* (FC) si es ordinaria y  $\forall t, t' \in T$ , si  $\bullet t \cap \bullet t' \neq \emptyset$  entonces  $\bullet t = \bullet t'$ ;
- $\mathcal{N}$  es una red de *igual conflicto* (EQ) si y solo si  $\forall t_1, t_2 \in T$  tal que  $\bullet t_1 \cap \bullet t_2 \neq \emptyset, \textbf{Pre}[P, t_1] = \textbf{Pre}[P, t_2]$ .

Las redes DSSP que se estudiaran a fondo en el Capítulo 4 son una subclase de RdP compuesta por diferentes módulos SM también llamados agentes que cooperan a través del paso asíncrono de mensajes mediante buffers. Los buffers únicamente pueden tener un agente como salida y no pueden influenciar la resolución de conflictos dentro de los agentes, es decir, las transiciones en conflicto de las SM deben pertenecer a una misma clase equivalente en EQ de la red DSSP.

### 2.3. Conceptos estructurales y comportamentales

Una vez definida la estructura de las RdP, así como la ecuación de transición, se proveen propiedades y conceptos comportamentales como la vivacidad y los bloqueos de los sistemas de RdP, así como también estructurales:

vivacidad estructural, leyes invariantes, sifones y trampas y finalmente conflictos.

**Vivacidad, Vivacidad Estructural y Bloqueos.** La vivacidad es una propiedad relacionada con la disparabilidad potencial de las transiciones en todos los marcados alcanzables. Una transición es viva si es potencialmente disparable en todos los marcados alcanzables. En otras palabras, una transición es viva si nunca pierde la posibilidad de disparo (es decir, de realizar una actividad). Una transición  $t$  es potencialmente disparable en  $\mathbf{m}$  si existe una secuencia de disparo  $\sigma$  que lleva a un marcado  $\mathbf{m}'$  en el cual  $t$  está habilitada, es decir,  $\mathbf{m}[\sigma]\mathbf{m}' \geq \mathbf{Pre}[P, t]$ . Una RdP es viva si todas sus transiciones son vivas. La no vivacidad para marcados iniciales arbitrarios refleja una patología de la estructura de la red: no vivacidad estructural. Una RdP es estructuralmente viva si existe al menos un marcado inicial vivo. En los marcados de bloqueo todas las transiciones están muertas, así que ninguna puede ser disparada. Un sistema se dice que está libre de bloqueos (es decir, para cualquier marcado alcanzable) si al menos siempre se puede disparar una transición. La vivacidad es una condición más fuerte que la libertad de bloqueos.

**T-semiflujos, P-semiflujos e Invariantes.** Los T-semiflujos son anuladores derechos no negativos de  $\mathbf{C}$ . Así que un vector  $\mathbf{x} \geq 0$  es un T-semifluido si  $\mathbf{C} \cdot \mathbf{x} = 0$ . Nosotros definimos mediante  $\|\mathbf{x}\|$  el soporte del vector  $\mathbf{x}$  que contiene todos los elementos no nulos:  $\|\mathbf{x}\| = \{i | \mathbf{x}[i] \neq 0\}$ . Un T-semifluido  $\mathbf{x}$  se dice que es mínimo cuando no existe un T-semifluido  $\mathbf{x}'$  tal que  $\|\mathbf{x}'\| \subset \|\mathbf{x}\|$ . Del mismo modo, definimos P-semiflujos como anuladores izquierdos no negativos de  $\mathbf{C}$ . Un vector  $\mathbf{y} \geq 0$  es un P-semifluido si  $\mathbf{y} \cdot \mathbf{C} = 0$ .

Premultiplicando la ecuación de estado por un P-semifluido  $\mathbf{y}$  obtenemos una ley invariante (una relación lineal entre las variables del marcado que permanece cierta en todos los estados alcanzables)

$$\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m}_0 + \mathbf{y} \cdot \mathbf{C} \cdot \boldsymbol{\sigma} = \mathbf{y} \cdot \mathbf{m}_0 = \text{constante}.$$

Sustituyendo  $\boldsymbol{\sigma}$  por un T-semifluido  $\mathbf{x}$  en la ecuación de estados, si había una secuencia disparable con un vector contador de disparos, vemos que este corresponde a una secuencia repetitiva.

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \mathbf{x} = \mathbf{m}_0$$

La existencia de P/T-semiflujos proporciona información interesante acerca del comportamiento del sistema. Si existe un P-semifluido  $\mathbf{y} > \mathbf{0}$

entonces  $\mathcal{N}$  es *conservativa*. El estado conservativo  $\mathcal{N}$  asegura la acotación estructural. Además,  $\mathcal{N}$  es *consistente* si existe un T-semiflujo  $\mathbf{x} > \mathbf{0}$ . Un sistema que es vivo y acotado debe de ser consistente, ya que un marcado obtenido mediante el disparo de una secuencia repetitiva que contiene todas las transiciones corresponde a un T-semiflujo positivo.

**Trampas.** Una trampa es un subconjunto de lugares tal que el conjunto de sus transiciones de salida está contenido en el conjunto de sus transiciones de entrada:  $\theta \subseteq P$  es una trampa si  $\theta^\bullet \subseteq {}^*\theta$ . La siguiente propiedad se satisface en RdP ordinarias: *si una trampa está (o se convierte) marcada, permanecerá marcada para cualquier evolución posible del sistema*.

**Sifones.** Un sifón es un subconjunto de lugares tal que el conjunto de sus transiciones de entrada está contenido en el conjunto de sus transiciones de salida:  $\Sigma \subseteq P$  es un sifón si  ${}^*\Sigma \subseteq \Sigma^\bullet$ . Un sifón es mínimo si cualquier subconjunto contenido en él no es un sifón. La siguiente propiedad se satisface en RdP ordinarias:

1. Si  $\mathbf{m}$  es un marcado de bloqueo, i.e., un marcado muerto, entonces  $\Sigma = \{p | \mathbf{m}(p) = 0\}$  es un sifón no marcado (vacío).
2. *Si un sifón es (o se convierte) en no marcado, este permanecerá sin marcas para cualquier posible evolución futura.* Además todas sus transiciones de entrada y salidas estarán muertas (deshabilitadas). Así que el sistema será no vivo, aunque pueda estar libre de bloqueos.

En la Fig. 2.2 aparece una RdP que contiene un sifón y una trampa: en la parte izquierda aparece la RdP con los lugares que pertenecen al sifón de color gris, mientras que en la parte de derecha se resaltan los lugares que forman la trampa:

$$(Sifón) \quad \left| \begin{array}{l} \Sigma = \{p_1, p_3, p_6, p_7\} \\ {}^*\Sigma = \{t_6, t_1, t_3, t_4\} \\ \Sigma^\bullet = \{t_1, t_3, t_4, t_5, t_6\} \\ \Rightarrow {}^*\Sigma \subseteq \Sigma^\bullet \end{array} \right. \quad (Trampa) \quad \left| \begin{array}{l} \theta = \{p_1, p_3, p_6, p_8\} \\ {}^*\theta = \{t_6, t_1, t_3, t_7, t_5\} \\ \theta^\bullet = \{t_1, t_3, t_5, t_6\} \\ \Rightarrow {}^*\theta \subseteq \theta^\bullet \end{array} \right.$$

Un **sifón malo** es un sifón que no contiene ninguna trampa:  $\Sigma$  es un sifón malo si no existe un  $\theta$  tal que  $\theta \subseteq \Sigma$ . Estos sifones se pueden vaciar durante la evolución de la RdP produciendo bloqueos totales o parciales.

**Conflictos y Conflictos Estructurales.** Un *conflicto* es la situación donde no todas las transiciones habilitadas pueden ocurrir a la vez. Formalmente,  $t, t' \in T$  están en relación de conflicto en un marcado  $\mathbf{m}$  si

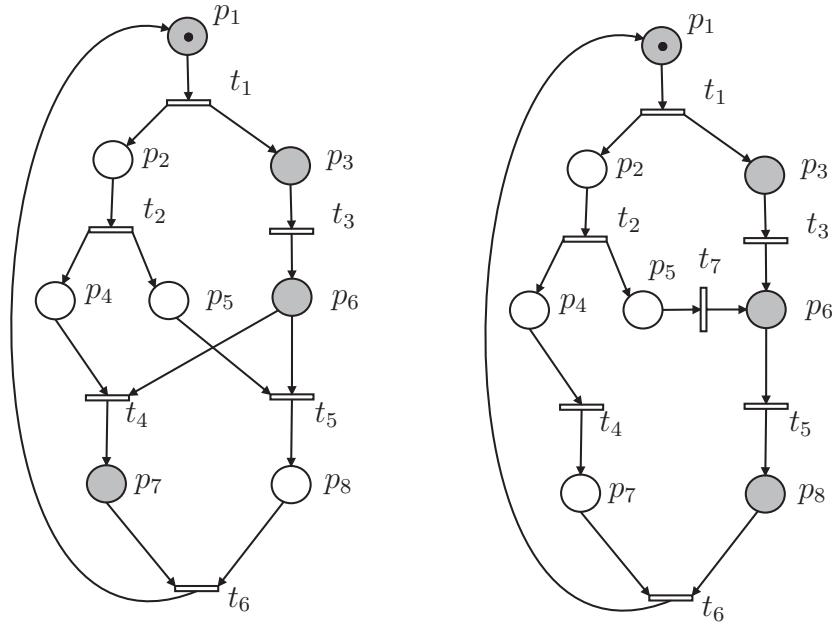


Figura 2.2: Ejemplo de sifón (izq) y de trampa (drch)

existe  $k, k' \in \mathbb{N}$  tal que  $\mathbf{m} \geq k \cdot \mathbf{Pre}[P, t]$  y  $\mathbf{m} \geq k' \cdot \mathbf{Pre}[P, t']$ , pero  $\mathbf{m} \not\geq k \cdot \mathbf{Pre}[P, t] + k' \cdot \mathbf{Pre}[P, t']$ . Para satisfacer la condición anterior es necesario que  $\bullet t \cap \bullet t' \neq \emptyset$ . En la RdP de la Fig. 2.1 en el marcado  $\mathbf{m}_0$  las transiciones  $t_1$  y  $t_3$  están en conflicto. En general, cuando  $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t'] \neq 0$ ,  $t$  y  $t'$  están en *relación de igual conflicto (EQ)*. Esto significa que ambas transiciones están habilitadas cuando una lo está. Por definición, una transición está siempre en relación de EQ con ella misma, esto es una relación de *equivalencia* en el conjunto de las transiciones y cada clase equivalente es un conjunto en igual conflicto denotado, para una transición dada  $t$ ,  $EQS(t)$ .  $SEQS$  es el conjunto de todos los conjuntos en conflicto de igualdad de una red dada.



## **Capítulo 3**

---

# **Redes DSSP (Deterministically Sincronized Secuential Process)**

---

En este capítulo se presentan formalmente las DSSP y la propiedad del teorema de rango. También se va a ver su utilidad para el modelado de sistemas reales, así como los estados de bloqueo que pueden alcanzar. Finalmente, se mostrarán y compararán dos métodos diferentes para prevenir marcados de bloqueo: (1) sifones malos y (2) preasignación de buffers.

### **3.1. Definición y teorema de rango**

Las DSSP son subclases de RdP formadas por agentes secuenciales cooperando a través del paso asíncrono de mensajes mediante buffers. Los agentes secuenciales son modelados por máquinas de estado vivas y seguras (redes P/T ordinarias y fuertemente conectadas donde cada transición tiene exactamente un lugar de entrada y un lugar de salida, y las cuales contienen únicamente una marca; los lugares representan los posibles estados, y el estado actual está indicado mediante una única marca). El esquema de producción/consumo está representado mediante lugares que reciben marcas de los productores y las dan a los consumidores. Para distinguir estos lugares de los que modelan los estados de los agentes, se llamarán buffers. En los dibujos que representan los modelos, una doble circunferencia indica que un lugar es un buffer. Debido a que las DSSP quieren capturar solo aspectos de cooperación en el sistema, introducen dos restricciones sintácticas en la conexión de los buffers que previenen competición. La primera es que las

marcas de un buffer dado solo pueden ser consumidas por un agente particular, llamado el destino del buffer. La segunda es que un buffer nunca puede condicionar elecciones internas en su agente destino. Estas restricciones nos conducen a la siguiente definición formal:

**Definition 3.1.** [10] Un sistema  $P/T \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m}_0 \rangle$  es un DSSP, donde  $P$  es la unión de los conjuntos disjuntos de lugares  $P_1, \dots, P_n$ , y  $B$ , y  $T$  es la unión de los conjuntos disjuntos de transiciones  $T_1, \dots, T_n$ , y se cumple lo siguiente:

1. Sea  $\mathcal{N}_i = \langle P_i, T_i, \mathbf{Pre}[P_i, T_i], \mathbf{Post}[P_i, T_i] \rangle$  una subred llamada agente  $i$ . Entonces,  $\langle \mathcal{N}_i, \mathbf{m}_0[P_i] \rangle$  es una máquina de estados viva 1-marcada. Dado un nodo  $x \in \bigcup_{i=1}^n (P_i \cup T_i)$ ,  $I(x)$  indica el índice de la subred a la que pertenece, es decir,  $x \in P_{I(x)} \cup T_{I(x)}$ .
2. Para cada  $b \in B$ , existe un  $\text{dest}(b) \in \{1, \dots, n\}$  tal que  $b^\bullet \subseteq T_{\text{dest}(b)}$ .
3. Si  $t, t' \in p^\bullet$ , donde  $p \in P_{\text{dest}(b)}$ , entonces  $\mathbf{Pre}[b, t] = \mathbf{Pre}[b, t']$ .

Como se ha dicho anteriormente, este tipo de redes (DSSP) no permiten modelar sistemas en los que exista competencia entre los agentes por un recurso o que los buffers condicionen decisiones internas de los agentes. Estas limitaciones en el modelado permiten obtener fuertes resultados analíticos que sirven para comprobar algunas propiedades del sistema. Una buena prueba de ello es el teorema de rango, que provee condiciones necesarias y suficientes para determinar la vivacidad estructural en DSSP [12, 9]:

**Theorem 3.2** (Teorema de Rango). [10] Sea  $\mathcal{N}$  una DSSP conservativa. Existe un marcado  $\mathbf{m}_0$  tal que  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  es una DSSP viva si y solo si  $\mathcal{N}$  es consistente y  $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$ .

## 3.2. Aplicación a sistemas reales

En este apartado se van a ilustrar dos modelados de sistemas reales pertenecientes a la subclase DSSP. El primer sistema es una línea de producción de mesas (sistema productivo SP1), mientras que el segundo corresponde con dos guías clínicas que necesitan sincronizarse (sistema de salud SS1).

### 3.2.1. Sistemas productivos

El SP1 está formado por tres estaciones de trabajo que: fabrican patas (ET1), fabrican tableros (ET2) o ensamblan la mesa (ET3). El layout, junto

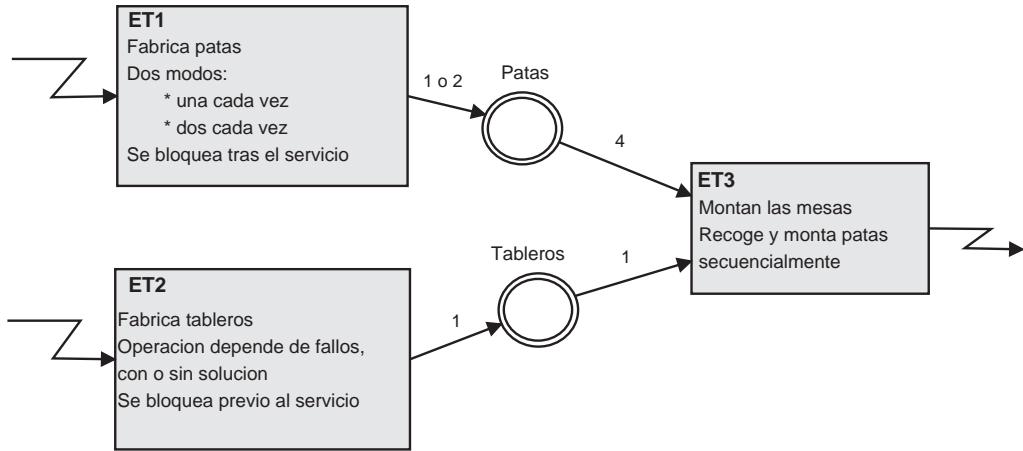


Figura 3.1: Layout de una línea de producción [10]

con algunas características de las estaciones de trabajo está representado en la Fig. 3.1.

La ET1 y ET2 depositan sus respectivos productos (patas y tableros) en dos buffers intermedios para que sean utilizados por la ET3. Cuando un buffer se vacía su consumidor está esperando y cuando se llena su productor está bloqueado. La Fig. 3.2 muestra el modelo del sistema descrito mediante una DSSP. Cada una de las ET están modeladas con máquinas de estado vivas, seguras y 1-marcadas. Tanto los productos intermedios como la materia prima para las patas y los tableros se modelan mediante marcas en los buffers.  $b_1$  y  $b_3$  representan la disponibilidad de materia prima para la fabricación de patas y tableros respectivamente, mientras que  $b_2$  y  $b_4$  modelan los productos acabados y disponibles para ser ensamblados de patas y tableros.

La ET1 puede producir una o dos patas a la vez dependiendo de razones económicas o de mantenimiento. Abstrayendo la razón precisa, esto puede ser modelado por una elección interna que decide entre las dos posibilidades ( $t_1^1$  y  $t_4^1$ ).

La ET2 fabrica los tableros de las mesas, y estos pueden tener diferentes tipos de fallos (fallos reparables, fallos irreparables o sin fallos) que también son modelados mediante elecciones internas (entre  $t_4^2$ ,  $t_6^2$  y  $t_2^2$ ).

Finalmente la ET3 ensambla la mesa de forma secuencial, es decir, monta cada ítem por separado, cogiendo en cada momento los componentes que

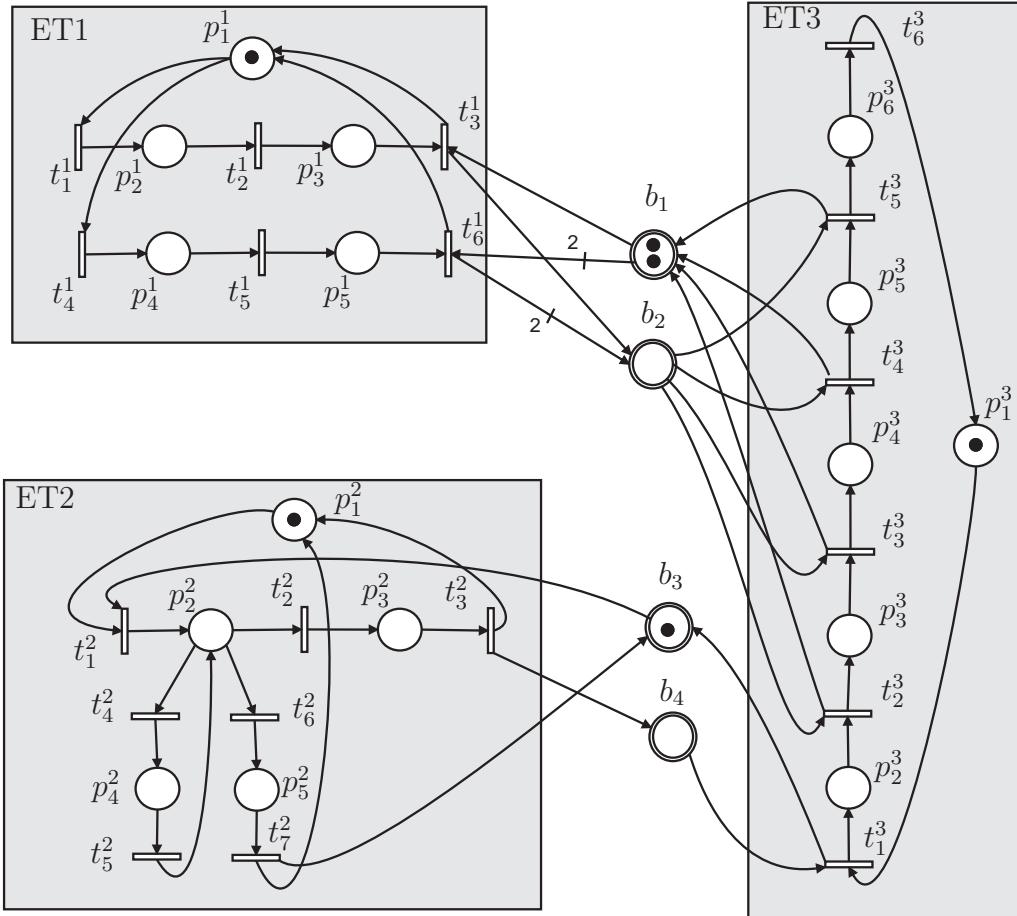


Figura 3.2: Modelo DSSP vivo de la línea de producción de la Fig. 3.1 [10]

necesite.

La DSSP de la Fig. 3.2 es una red consistente y conservativa compuesta por 3 agentes (ET1, ET2, ET3) y cuatro buffers ( $b_1, b_2, b_3$  y  $b_4$ ). El conjunto de todos los conjuntos en igual conflicto es:  $SEQS = \{\{t_1^1, t_4^1\}, \{t_1^1\}, \{t_3^1\}, \{t_5^1\}, \{t_6^1\}, \{t_1^2\}, \{t_2^2, t_4^2, t_6^2\}, \{t_3^2\}, \{t_5^2\}, \{t_7^2\}, \{t_1^3\}, \{t_2^3\}, \{t_3^3\}, \{t_4^3\}, \{t_5^3\}, \{t_6^3\}\}$ . La red satisface el teorema de rango (Teorema 3.2):  $|SEQS| - 1 = 15 = rank(\mathbf{C})$ . De acuerdo con este teorema la red es estructuralmente viva, es decir, existe al menos un marcado inicial que hace vivo el sistema. El marcado actual de la Fig. 3.2 es  $\mathbf{m}_0 = p_1^1 + p_2^1 + p_3^1 + 2b_1 + b_3$  y hace vivo el sistema.

### 3.2.2. Sistemas de salud

Se considera el sistema de salud de la Fig. 3.3. Existen dos guías clínicas (GC1 y GC2) modeladas mediante SM. Cada una de ellas puede tratar a pacientes quemados ( $\{t_1, t_2, t_3, t_4\}$  para la GC1 y  $\{t_9, t_{10}, t_{11}, t_{12}\}$  para la GC2) o a pacientes que han sufrido un accidente de tráfico ( $\{t_5, t_6, t_7, t_8\}$  para la GC1 y  $\{t_{13}, t_{14}, t_{15}, t_{16}\}$  para la GC2). Ambos tratamientos requieren la utilización de dos quirófanos. Con objeto de mejorar el rendimiento del hospital se pretende que ambas guías clínicas traten a pacientes con la misma patología simultáneamente para no tener que estar acondicionando los quirófanos con tanta frecuencia. Para ello, la GC1 y GC2 intercambian información por medio de los buffers (o canales de comunicación)  $b_1, b_2, b_3$  y  $b_4$  del siguiente modo: la GC1 cuando selecciona un paciente quemado para su tratamiento lo comunica a la GC2 por medio del  $b_1$ . De igual modo, la GC2 lo comunica a la GC1 por medio del  $b_2$ . Ambas guías clínicas necesitan recibir un mensaje de que la otra guía clínica también tiene listo un paciente quemado para poder iniciar los tratamientos. De forma similar sucede con los pacientes que han sufrido un accidente de tráfico: GC1 envía el mensaje por medio del  $b_3$  y la GC2 mediante  $b_4$ .

La sistema resultante que aparece en la Fig. 3.3 es una DSSP consistente y conservativa compuesta por dos agentes (GC1 y GC2) que cooperan a través del paso asíncrono de mensajes mediante los buffers  $b_1, b_2, b_3$  y  $b_4$ . El conjunto de todos los conjuntos en igual conflicto es:  $SEQS = \{\{t_1, t_5\}, \{t_3\}, \{t_4\}, \{t_6\}, \{t_7\}, \{t_8\}, \{t_9, t_{13}\}, \{t_{10}\}, \{t_{11}\}, \{t_{12}\}, \{t_{14}\}, \{t_{15}\}, \{t_{16}\}\}$ . La red no satisface el teorema de rango (Teorema 3.2):  $|SEQS| - 1 = 13 \neq rank(\mathbf{C}) = 14$ . De acuerdo con este teorema, la red es estructuralmente no-viva, es decir, no existe ningún marcado inicial que hace vivo el sistema. El marcado actual de la Fig. 3.3 es  $\mathbf{m}_0 = p_1 + p_8$  que mediante el disparo de  $\sigma = t_1 t_{13}$  lleva al marcado de bloqueo  $\mathbf{m}' = p_2 + p_{12} + b_1 + b_4$ .

## 3.3. Motivación del trabajo: bloqueos en redes DSSP

En la Sección 3.2 se han visto dos sistemas (producción de mesas y salud) modelados mediante DSSP en los cuales se ha considerado su posible vivacidad mediante el teorema de rango. El sistema de producción es un sistema estructuralmente vivo, mientras que el de salud no. Como se ha observado el teorema de rango caracteriza estos comportamientos, pero existe una au-

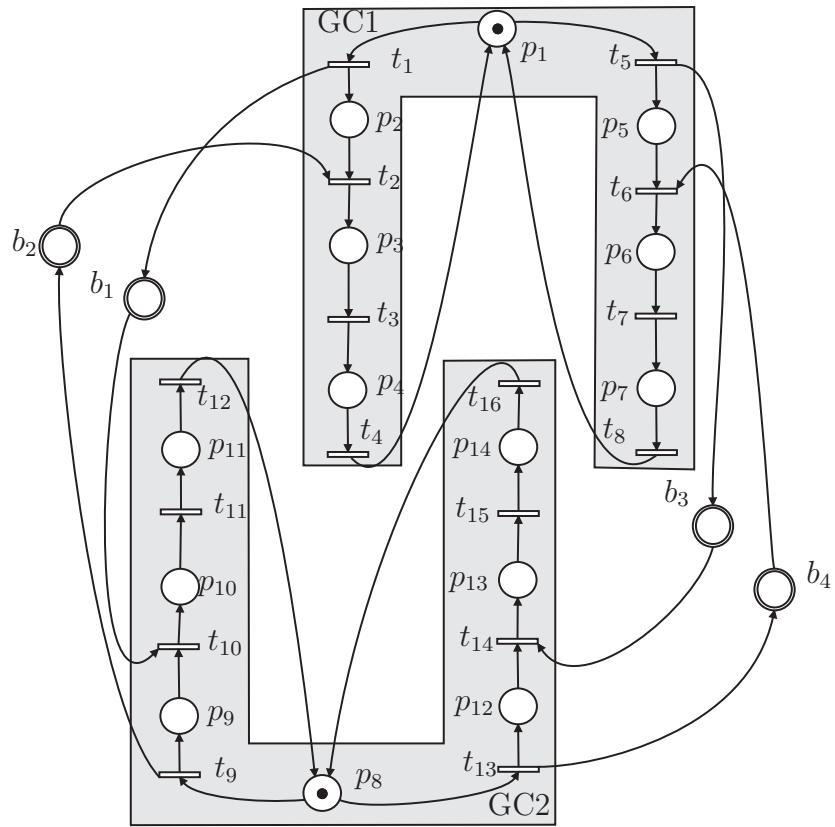


Figura 3.3: Ejemplo DSSP sistema de salud

sencia de métodos o algoritmos que permitan transformar los sistemas no vivos en vivos. En este apartado se van han a realizar algunos cambios en el sistema de producción (SP1) que van a provocar que la DSSP resultante no sea viva. Finalmente se explicaran las situaciones de bloqueo en el sistema de salud y en el nuevo sistema productivo resultante de la modificaciones.

Al sistema productivo de la Fig. 3.1 se añade un nuevo modelo de mesa (mesa B) que es capaz de soportar el doble de peso que el anterior. Para ello, manteniendo el mismo tablero que se utilizaba en la mesa anterior (mesa A), se han diseñado unas nuevas patas más altas y robustas de un material más resistente que permiten soportar este nuevo requisito estructural. Las nuevas patas son fabricadas por la misma estación de trabajo que las anteriores (ET1). Además para simplificar la red, tanto las patas A como las patas B son fabricadas en lotes de 4 representados por una única marca. Las mesas B son ensambladas en la misma estación de trabajo que las mesas A (ET3). Como resultado de incorporar al sistema productivo la fabricación de las mesas B, se obtiene la estructura DSSP representada en la Fig. 3.4. En este

caso la ET1 puede fabricar patas A o B, esta situación está representada mediante la elección interna entre ( $t_1^1$  y  $t_4^1$ ). De igual modo se ha añadido una nueva línea de montaje en la ET3 que permite ensamblar las mesas B. Ambas posibilidades, ensamblar mesas A o B, están representadas mediante la elección interna entre ( $t_1^3$  y  $t_4^3$ ). Además dos nuevos buffer  $b_5$  y  $b_6$  se añaden a la red. Estos buffers representan la disponibilidad de material para fabricar patas B ( $b_5$ ) y los lotes de patas B disponibles para ser ensamblados ( $b_6$ ).

**Example 3.3.** La RdP de la Fig. 3.4 es una DSSP consistente y conservativa compuesta por 3 agentes (ET1, ET2, ET3) y seis buffers ( $b_1$ ,  $b_2$ ,  $b_3$  y  $b_4$   $b_5$   $b_6$ ). El conjunto de todos los conjunto en igual conflicto es:  $SEQS = \{\{t_1^1, t_4^1\}, \{t_1^1\}, \{t_2^1\}, \{t_3^1\}, \{t_5^1\}, \{t_6^1\}, \{t_1^2\}, \{t_2^2, t_4^2, t_6^2\}, \{t_3^2\}, \{t_5^2\}, \{t_7^2\}, \{t_1^3, t_4^3\}, \{t_2^3\}, \{t_3^3\}, \{t_5^3\}, \{t_6^3\}\}$ . La estructura DSSP no satisface el teorema de rango:  $|SEQS| - 1 = 14 \neq \text{rank}(\mathbf{C}) = 15$ . De acuerdo con el Teorema 3.2 la RdP es estructuralmente no viva, es decir, no existe ningún marcado inicial que haga vivo el sistema. Para el marcado de la Fig. 3.4  $\mathbf{m}_0 = p_1^1 + p_1^2 + p_1^3 + b_1 + b_4 + b_6$ , la secuencia de disparo  $\sigma = t_4^1 t_5^1 t_1^3 t_1^2 t_2^2 t_3^2$  lleva al marcado de bloqueo  $\mathbf{m}' = p_5^1 + p_1^2 + p_1^3 + b_1 + b_4 + b_6$ .

Este bloqueo es debido a que la ET1 decide fabricar patas B ( $t_4^1$ ) sin disponer de su materia prima ( $b_5$ ), mientras la ET3 decide montar mesas A ( $t_1^3$ ) sin disponer de patas tipo A ( $b_2$ ). Además, para poder disponer de materia prima para fabricar patas B es necesario montar mesas B y para poder disponer de patas A es necesario fabricar patas A. Como ambas estaciones de trabajo están bloqueadas, y requieren de la ejecución de la otra para salir del bloqueo, el sistema esta bloqueado.

En el caso del sistema de salud el bloqueo es debido a que la GC1 debe sincronizarse con la GC2, es decir la elección de tratar a pacientes quemados o que han sufrido un accidente de tráfico de la GC1 ( $t_1$  o  $t_5$ ) debe estar sincronizada con esta misma elección en la GC2 ( $t_9$  o  $t_{13}$ ). La causa estructural del bloqueo es que ambas GC toman sus respectivas decisiones de forma libre e independiente antes de ser sincronizadas. Es decir, cada guía puede mandar el mensaje de que tiene preparado a un tipo de paciente diferente, y esperar indefinidamente el mensaje de la otra guía.

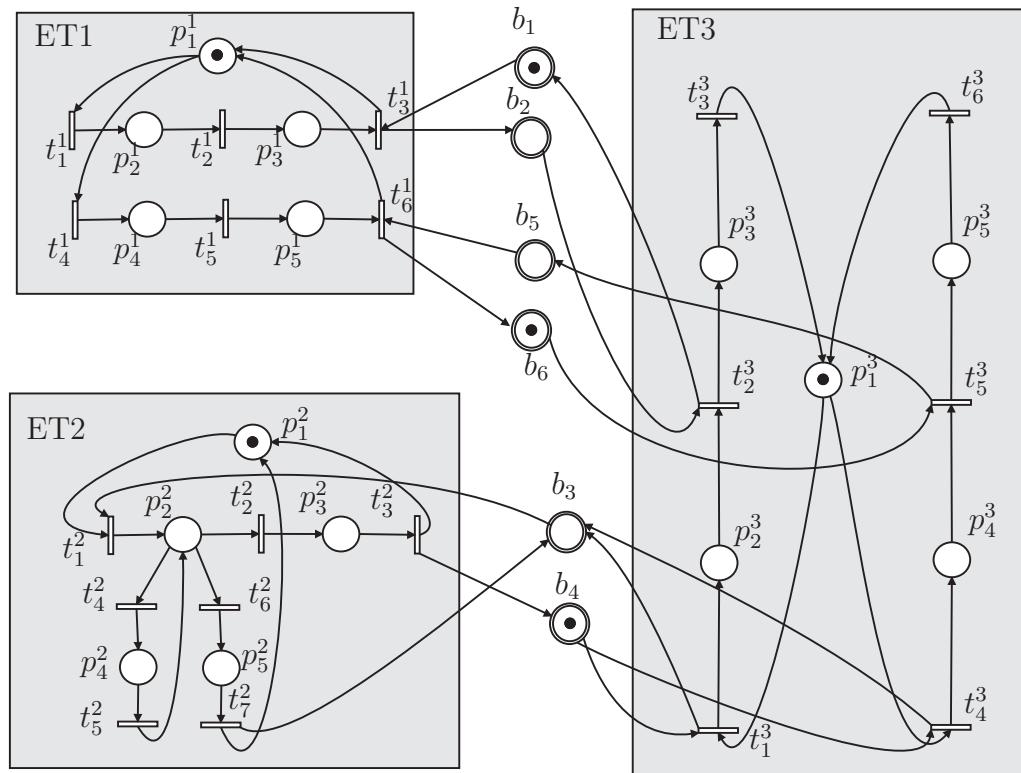


Figura 3.4: Modelo DSSP de sistema productivo de mesas A y B

### 3.4. Diferentes enfoques para garantizar vivacidad estructural en DSSP

En este apartado se van a proponer dos métodos diferentes para intentar prevenir las situaciones de bloqueo en DSSP. Ambos se van a comparar, estudiando sus limitaciones y ventajas. El primero de estos métodos se basa en calcular los sifones malos de la red y prevenir su vaciado, aproximación que ha sido utilizada en otro tipo de redes ordinarias [5]. El segundo método, basado en la preasignación de los buffers, intenta que siempre que una transición de un agente dado sea disparada, al menos un T-semiflujo del agente, pueda ser disparado hasta el final.

#### 3.4.1. Método basado en controlar los sifones malos mediante lugares de monitorización

Un conocido método para conseguir vivacidad en RdP ordinarias consiste en controlar los sifones malos. Para ello, primero se calcula el conjunto

de sifones malos y después, por cada uno, se introduce un lugar de control para prevenir su vaciado. El proceso debe ser iterativo, ya que los lugares de control introducidos, pueden añadir nuevos sifones malos al sistema. Sin embargo, además de que esta estrategia solo podría ser utilizada en DSSP ordinarias, como resultado de su aplicación nuevos buffers son añadidos (los lugares de control pueden ser vistos como nuevos buffers). Estos nuevos buffers pueden incumplir ambas restricciones de las DSSP: condicionan decisiones internas dentro de un agente y tienen transiciones de salida en más de un agente.

**Definition 3.4.** *Sea  $\Sigma_k = \{p_1, \dots, p_n\}$  un sifón malo y sea  $p_k$  el lugar de monitorización que impide el vaciado de  $\Sigma_k$ . Se cumple lo siguiente:*

1.  $\mathbf{C}[p_k, t_j] = \sum_{l=1}^n \mathbf{C}[p_l, t_j]; p_l \in \Sigma_k;$
2.  $\mathbf{m}_0[p_k] = \sum_{l=1}^n \mathbf{m}_0[p_l] - 1; p_l \in \Sigma_k.$

En el Ej. 3.3 se ha mostrado que el sistema de RdP de la Fig. 3.4 tiene un marcado de bloqueo  $\mathbf{m}' = p_5^1 + p_1^2 + p_2^3 + b_1 + b_4 + b_6$ . El sifón vacío en este marcado es  $\Sigma_1 = \{p_1^1, p_2^1, p_3^1, p_1^2, p_3^2, p_4^2, p_5^2, b_2, b_5\}$ . El lugar de monitorización que impide su vaciado (Def. 3.4) es  $p_m$ , donde  $\mathbf{C}[p_m, T] = [t_3^1, -t_4^1, -t_1^3, t_5^3]$  y  $\mathbf{m}_0[p_m] = 1$ . Este lugar está ilustrado en azul en la Fig. 3.5.

La RdP resultante no es DSSP porque las propiedades 2) y 3) en Def. 3.1 no se cumplen. La segunda propiedad dice que las marcas de un buffer dado solo pueden ser consumidas por un agente particular (consumidor). Este no es el caso porque el nuevo buffer  $p_m$  proporciona marcas a los agentes  $ET1$  and  $ET3$ . Tampoco se satisface la tercera propiedad, ya que el buffer  $p_m$  está condicionando las elecciones internas en los agentes  $ET1$  y  $ET3$ .

Si una DSSP es consistente y conservativa pero no cumple el teorema de rango, es imposible impedir los marcados de bloqueo sin salirse del tipo de red. Esto es debido a que los bloqueos se producen a causa de que diferentes agentes disparan libremente transiciones en conflicto que pertenecen a T-semiflujos locales que posteriormente van a requerir de marcas en algún buffer sin tener en cuenta el marcado de estos. Si queremos prevenir los marcados de bloqueo, va a ser necesario que los buffers puedan condicionar la toma de estas decisiones. De este modo, se está incumpliendo la condición 3 de las DSSP (Def. 3.1) y consecuentemente se perderá la clase de red.

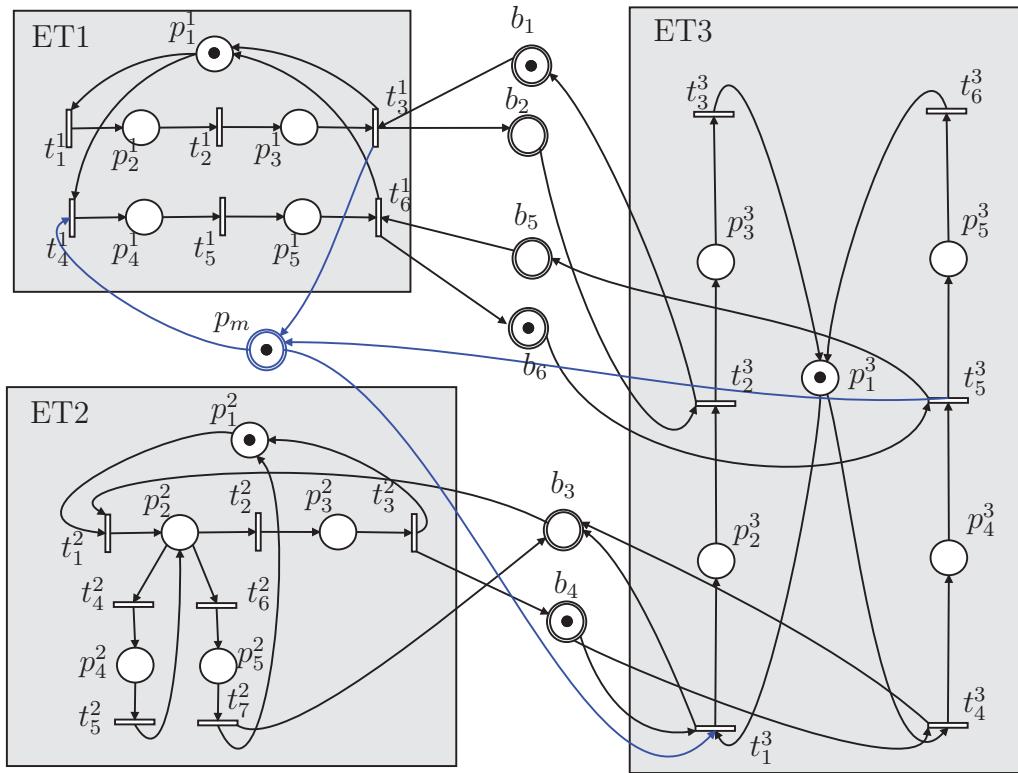


Figura 3.5: Lugar de monitorización ( $p_m$ ) que previene el marcado de bloqueo  $\mathbf{m}'$  de la DSSP de la Fig. 3.4

### 3.4.2. Método de preasignación de buffers

Como se ha explicado en la Sec. 3.4.1 resulta imposible prevenir situaciones de bloqueo en DSSP consistentes y conservativas que no cumplen el teorema de rango sin salir de la clase de sistemas/modelos.

En esta sección se propone un método basado en la preasignación de buffers que pueden condicionar las decisiones internas de los agentes, pero mantienen su privacidad (un agente de destino por buffer). Este método no añade nuevos lugares a la red, sino que modifica los arcos de salida de algunos buffers y añade nuevos arcos. La preasignación consiste en adelantar la asignación de todos los buffers de un T-semifluido local (perteneciente a un solo agente) desde sus transiciones originales hasta otras transiciones previas que estén en EQ. De este modo se está asegurando que al disparar una transición en conflicto, el agente pueda completar el disparo del T-semifluido local y regresar a su lugar de espera. En la Sec. 4, donde se desarrolla el algoritmo, se explica por qué no solo hay que tener en cuenta los T-semiflujos locales, sino que es necesaria una perspectiva global del sistema

para prevenir los bloqueos.

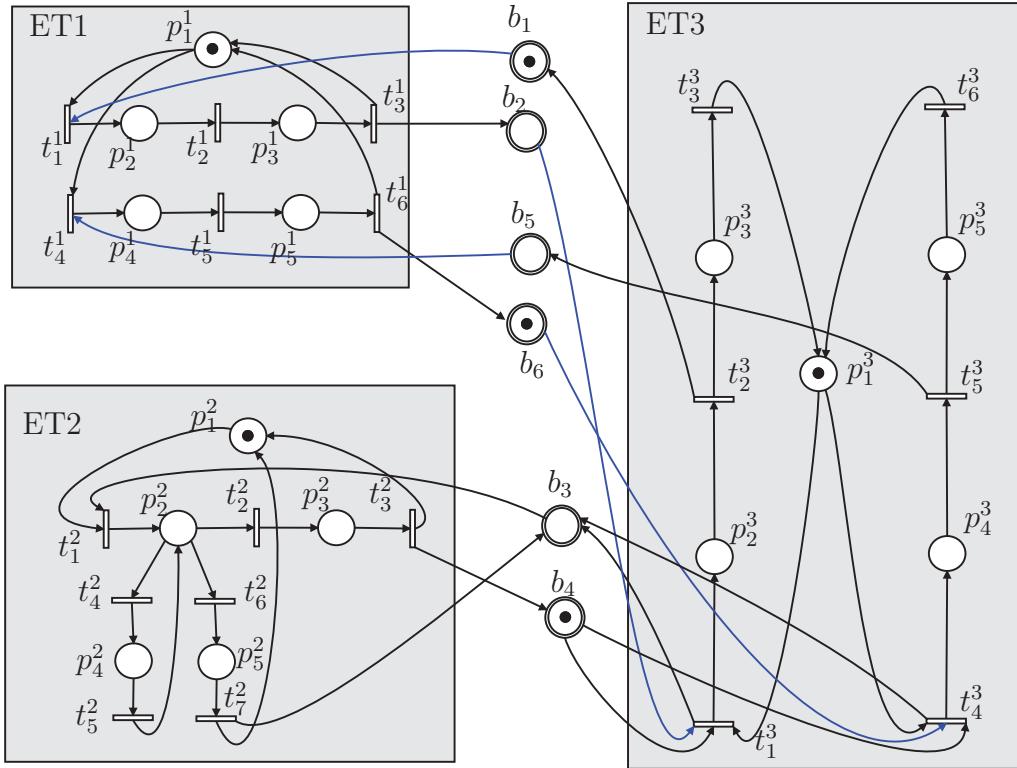


Figura 3.6: Preasignación de buffers en la RdP de la Fig. 3.4

Se va a realizar la preasignación de buffers de una forma intuitiva en la red de la Fig. 3.4. La ET1 (fabrica patas) es una máquina de estados formada por dos T-semiflujos locales:  $\mathbf{x}_1^1 = t_1^1 + t_2^1 + t_3^1$  y  $\mathbf{x}_2^1 = t_4^1 + t_5^1 + t_6^1$  que representan la fabricación de patas A y B respectivamente. El proceso de fabricación de patas A/B ( $\mathbf{x}_1^1/\mathbf{x}_2^1$ ) requiere consumir una marca de  $b_1/b_5$  (materia prima de tipo A/B). El consumo de esta marca se produce en la  $t_3^1/t_6^1$ , mientras que la decisión se toma libremente en la  $t_1^1/t_4^1$  sin saber si hay materia prima disponible en  $b_1/b_5$ . Para evitar la situación de haber tomado la decisión de fabricar un tipo de pata y que no se disponga de materia prima, se realiza la preasignación de los buffers  $b_1/b_5$  a las transiciones  $t_1^1/t_4^1$ . De este modo siempre que la ET1 decida fabricar patas A o B dispondrá de materia prima.

De igual modo sucede con la ET3, donde la asignación del buffer  $b_2/b_6$  se adelanta a la transición  $t_1^3/t_4^3$  para prevenir situaciones de bloqueo. En la ET2 no es necesario realizar ninguna preasignación, como se explicara más adelante, esto se debe a que solo contiene un T-semifljo local que perte-

nezca a uno global. Estas modificaciones pueden observarse en el sistema de la Fig. 3.6 donde los arcos modificados aparecen en color azul. Como resultado de estas modificaciones se obtiene una red libre de bloqueos donde los buffers pueden condicionar los conflictos. Ya no es una DSSP, nosotros llamamos a la nueva clase resultante *Synchronized Sequential Process* (SSP). La definición formal es igual que la Def. 3.1 sin la condición 3.

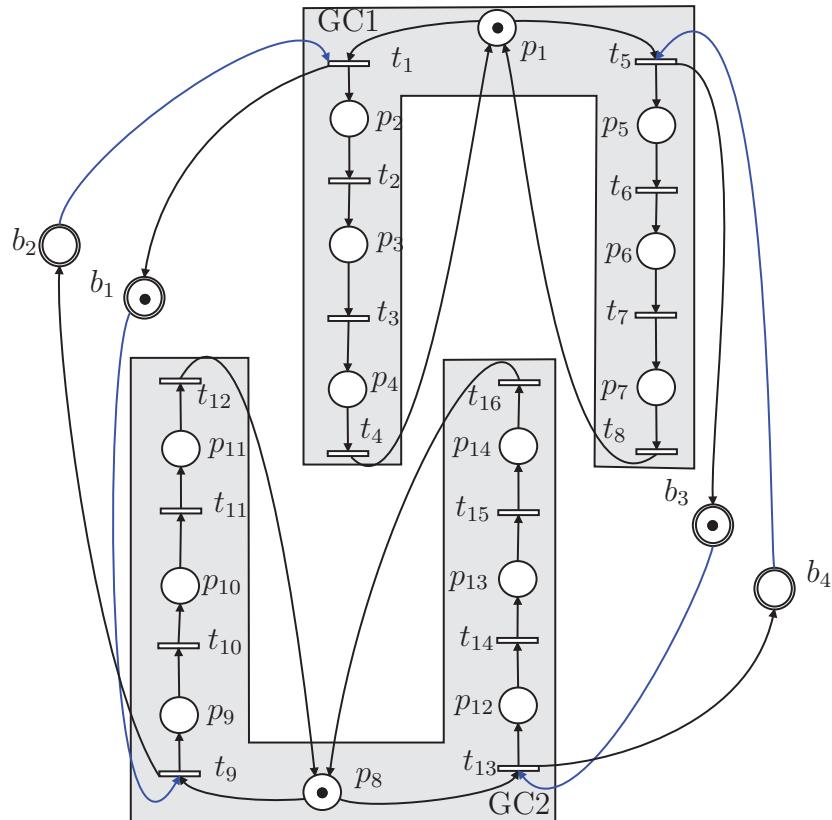


Figura 3.7: Preasignación de los buffers en la DSSP de la Fig. 3.3

De una forma similar se puede realizar la preasignación de los buffers en la DSSP de la Fig. 3.3 que modela el sistema de salud. La SSP resultante aparece en la Fig. 3.7 donde los nuevos arcos son de color azul. Esta red es estructuralmente viva, pero el marcado inicial que tenia en la DSSP de la Fig. 3.3 no es un marcado vivo, ya que no es posible disparar ninguna transición. Para conseguir un marcado vivo es necesario colocar una marca en los buffers  $b_1$  y  $b_3$ . Con este nuevo marcado inicial  $\mathbf{m}_0 = p_1 + p_8 + b_1 + b_3$  la red es viva, pero no se esta cumpliendo el objetivo del sistema de sincronizar ambas guías clínicas, ya que la GC2 puede tratar a un paciente quemando y a otro que ha sufrido un accidente de tráfico, sin que la GC1 haya tratado a

ningún paciente. Esto es debido a que las marcas que añadimos en los buffers representan mensajes de confirmación enviados por la GC1 simulando la selección de un paciente quemado y otro que ha sufrido un accidente de tráfico. En realidad si que se produce una sincronización del sistema, pero con una distancia distinta de 0. Se ha introducido una relación maestro-esclavo entre la GC2 y la GC1 respectivamente. Este problema no sucede en los sistemas de red que representan sistemas productivos, ya que las marcas de los buffers modelan almacenes de productos intermedios y al aumentar su marcado, lo único que se hace es aumentar el tamaño de estos almacenes. Por esto, en aquellas DSSP en las que se intercambien mensajes, si es necesario aumentar el marcado de los buffers para conseguir vivacidad, podemos estar cambiando el comportamiento deseado del sistema (en este caso sincronización con distancia 0).

Tanto la RdP de la Fig 3.4 como la de la Fig. 3.3 son sencillas y las modificaciones de los arcos inmediatas. En el capítulo 4 se verán estructuras más complejas en las que las máquinas de estado tienen varios conflictos consecutivos. Para estas, las modificaciones no son inmediatas y es necesario seguir una metodología que no añada “demasiada” restricciones a la red que conduzcan a un bloqueo.



## Capítulo 4

---

# Algoritmo basado en preasignación de buffers para conseguir vivacidad estructural.

---

En este capítulo se desarrolla un algoritmo que para algunas estructuras de redes DSSP inicialmente no vivas fuerza su vivacidad estructural. El algoritmo se basa en la idea de prevenir el bloqueo de las máquinas de estado haciendo que cada vez que una transición en conflicto sea disparada, al menos un componente repetitivo ( $T$ -semiflujo) de esta máquina de estados pueda ser disparado hasta el final. Para conseguir esto, es necesario preasignar los buffers de entrada.

El diagrama de flujo de la Fig. 4.1 muestra los pasos a seguir para conseguir vivacidad a partir de una red DSSP. Primero la estructura de la red  $\mathcal{N}$  debe de ser consistente y conservativa. Si  $\mathcal{N}$  satisface el teorema de rango, existe un marcado inicial vivo  $\mathbf{m}_0$ . En caso contrario, sera necesario aplicar el Alg. 4.9 para obtener un sistema SSP alternativo pero vivo.

Antes de explicar como realizar la preasignación de los buffers, es necesario fijar la notación de algunos conceptos y definiciones:

**Definition 4.1.** *Sea  $\mathcal{N}_i$  un agente de una DSSP  $\mathcal{N}$ . Los  $T$ -semiflujos de  $\mathcal{N}_i$  se llaman  $T$ -semiflujos locales del agente  $i$ . Los  $T$ -semiflujos de  $\mathcal{N}$  se llaman  $T$ -semiflujos globales de  $\mathcal{N}$ .*

**Property 4.2.** *Si  $\mathbf{X}_g$  es un  $T$ -semiflujo global, entonces se puede escribir como una combinación lineal de  $T$ -semiflujos locales.  $\mathbf{x}_g = \alpha \cdot \mathbf{x}_1 + \beta \cdot \mathbf{x}_2 +$*

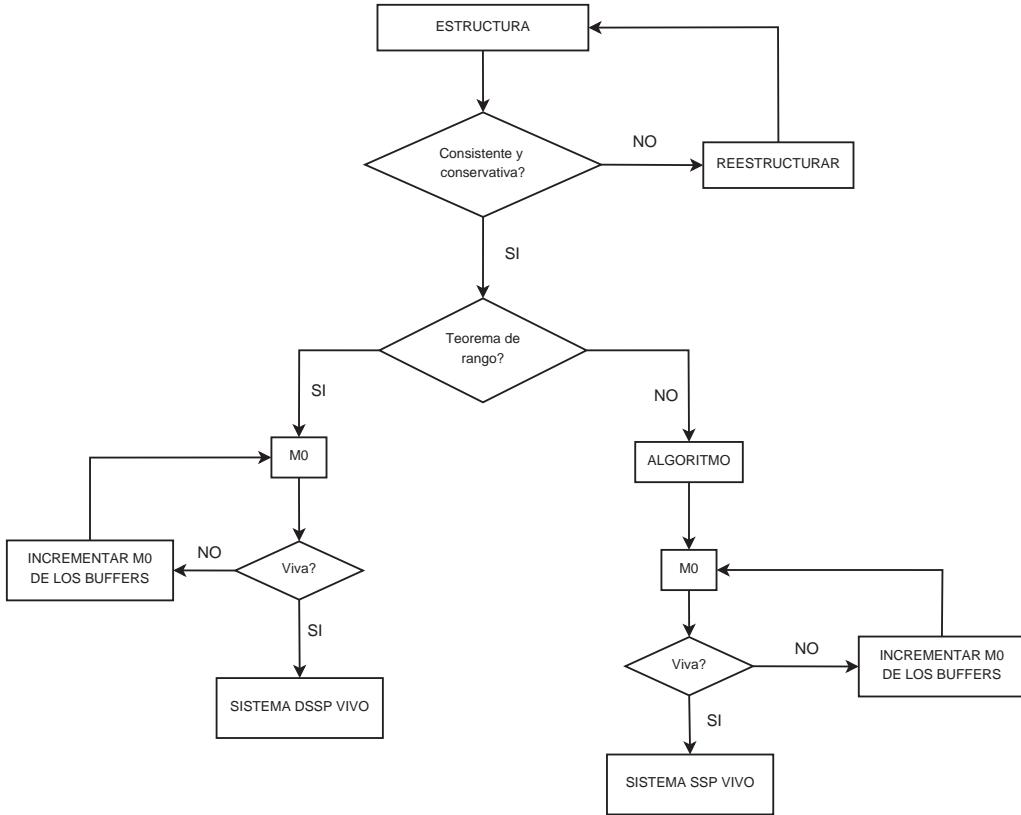


Figura 4.1: Comprobación de vivacidad en DSSP.

$\dots + \gamma \cdot \mathbf{x}_l$  donde  $\alpha, \beta, \dots, \gamma \in \mathbb{R}_{\geq 0}$  y  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l$  son  $T$ -semiflujos locales.

**Definition 4.3.** Un  $T$ -semiflujo global  $\mathbf{X}_g$  se llama  $T$ -semiflujo global completo si  $\mathbf{X}_g$  no es un  $T$ -semiflujo local, i.e., está compuesto por dos o más  $T$ -semiflujos locales de agentes distintos.

**Definition 4.4.** Sea  $\mathcal{N}_i$  un agente de una DSSP  $\mathcal{N}$  tal que tiene dos o más  $T$ -semiflujos locales que forman  $T$ -semiflujos globales completos. Sean  $\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_k^i$  los  $T$ -semiflujos locales de  $\mathcal{N}_i$  que pertenecen a  $T$ -semiflujos globales y sea  $\bar{P}_i = \left( \bigcap_{n=1}^k \bullet \|\mathbf{x}_n^i\| \right)$ . Un lugar  $p_e \in \bar{P}_i$  se llama de espera si no hay en  $\mathcal{N}_i$  ningún camino (utilizando los lugares pertenecientes a  $\bar{P}_i$ ) de  $p_e$  a otro lugar de  $\bar{P}_i$ .

Vamos a ver los  $T$ -semiflujo locales, globales y los lugares de espera de la red DSSP de la Fig. 3.4. La estructura está compuesta por tres estaciones

de trabajo: ET1, ET2 y ET3 que son modeladas mediante las subredes (SM)  $\mathcal{N}_1$ ,  $\mathcal{N}_2$  y  $\mathcal{N}_3$  respectivamente. En la Tab. 4.1 se muestran los T-semiflujos locales: la primera columna identifica al T-semiflujo, en la segunda aparece a la red que pertenece, la tercera muestra las transiciones que lo componen y finalmente en la cuarta aparecen los T-semiflujos globales a los que pertenece. Esta DSSP tiene 2 T-semiflujos globales totales:  $\mathbf{x}_g^1 = \mathbf{x}_1^1 + \mathbf{x}_1^2 + \mathbf{x}_1^3$  y  $\mathbf{x}_g^2 = \mathbf{x}_2^1 + \mathbf{x}_2^2 + \mathbf{x}_2^3$ .

Cuadro 4.1: T-semiflujos locales de la red de la Fig. 3.4

Id.	Red	Transiciones	$\mathbf{X}_g$ a los que pertenece
$\mathbf{x}_1^1$	$\mathcal{N}_1$	$t_1^1, t_2^1, t_3^1$	$x_g^1$
$\mathbf{x}_2^1$	$\mathcal{N}_1$	$t_4^1, t_5^1, t_6^1$	$x_g^2$
$\mathbf{x}_1^2$	$\mathcal{N}_2$	$t_2^2, t_3^2, t_1^2$	$x_g^1, x_g^2$
$\mathbf{x}_2^2$	$\mathcal{N}_2$	$t_4^2, t_5^2$	-
$\mathbf{x}_3^2$	$\mathcal{N}_2$	$t_5^2, t_6^2, t_1^2$	-
$\mathbf{x}_1^3$	$\mathcal{N}_3$	$t_1^3, t_2^3, t_3^3$	$x_g^1$
$\mathbf{x}_2^3$	$\mathcal{N}_3$	$t_4^3, t_5^3, t_6^3$	$x_g^2$

Una vez identificados los T-semiflujos locales de cada máquina es posible determinar el lugar de espera. Para  $\mathcal{N}_1$  y  $\mathcal{N}_3$  los lugares de espera son  $p_1^1$  y  $p_1^3$  respectivamente, mientras que  $\mathcal{N}_2$  no tiene lugar de espera, ya que solo contiene un T-semiflujo local ( $\mathbf{x}_1^2$ ) que pertenezca a alguno global. Por ello no es necesario realizar preasignaciones en la máquina de estados de la ET2

## 4.1. Aplicación del algoritmo a $TSF_1$

Para desarrollar el algoritmo, se ha empezado aplicando a redes DSSP en las que los T-semiflujos globales tienen algunas restricciones. Posteriormente estas restricciones se relajan y se añaden pasos en el algoritmo para conseguir vivacidad en estructuras más generales. Con este objetivo se definen los T-semiflujos de tipo 1 ( $TSF_1$ ):

**Definition 4.5.** Sea  $\mathcal{N}$  una DSSP compuesta por  $n$  agentes  $\langle \mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_n \rangle$ . Un T-semiflujo global  $\mathbf{x}_g$  de  $\mathcal{N}$  es un  $TSF_1$  si cumple las siguientes condiciones.

1. Es una combinación de dos o más T-semifujos locales:  $\mathbf{x}_g = \alpha \cdot \mathbf{x}_1 + \beta \cdot \mathbf{x}_2 + \dots + \gamma \cdot \mathbf{x}_l$  donde  $\alpha, \beta, \dots, \gamma \in \mathbb{R}_{\geq 0}$  y  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l$  son T-semifujos locales.
2.  $\mathbf{x}_g$  está compuesto como máximo por un T-semiflujo local por agente;
3. Existen buffers de entrada/salida privados para cada  $\mathbf{x}_g$ , es decir un bufer solo puede tener transiciones de entrada/salida que pertenezcan al mismo  $\mathbf{x}_g$ :  $\forall b \in (\bullet || \mathbf{x}_g || \cup || \mathbf{x}_g || \bullet) \Rightarrow (\bullet b \cup b \bullet) \in || \mathbf{x}_g ||$ .

Con este tipo de estructuras ( $TSF_1$ ) estamos asegurando que las marcas de los buffers que relacionan T-semifujos locales de un  $TSF_1$ , solo pueden ser producidas/consumidas por sus transiciones. Además al tratarse de una red consistente y conservativa, existe una secuencia de disparos de los T-semifujos locales que permiten disparar el  $TSF_1$  completamente. Si realizamos una preasignación de los buffers de entrada de todos los T-semifujos locales que forman un  $TSF_1$  a las primeras transiciones de su secuencia vista desde sus lugares de espera, estaremos asegurando que:

1. Existe un marcado inicial  $\mathbf{m}_0$  que para cualquier evolución del sistema, al menos siempre un T-semiflujo local perteneciente al  $TSF_1$  va a estar habilitado.
2. Todos T-semiflujo local de  $TSF_1$  podrán estar habilitados en alguna evolución del sistema
3. Solo aquellos T-semifujos locales pertenecientes al  $TSF_1$  cuyos buffers de entrada tengan suficientes marcas para completar el disparo, podrán iniciar la secuencia.

En necesario que la preasignación se realice desde una perspectiva global, es decir: de forma secuencial a los T-semifujos globales (en este caso a  $TSF_1$ ). De este modo si existe una transición  $(t_1)$  perteneciente a dos T-semiflujo locales ( $\mathbf{x}_1^1$  y  $\mathbf{x}_2^1$ ) en una máquina ( $\mathcal{N}_1$ ) y otra transición  $(t_2)$  perteneciente a otros dos T-semiflujo locales ( $\mathbf{x}_1^2$  y  $\mathbf{x}_2^2$ ) en otra máquina ( $\mathcal{N}_2$ ), y siendo  $\mathbf{x}_g^1 = \mathbf{x}_1^1 + \mathbf{x}_2^1$  y  $\mathbf{x}_g^2 = \mathbf{x}_1^2 + \mathbf{x}_2^2$  dos T-semifujos globales de tipo  $TSF_1$ , se asegura que los buffers que se preasignen a  $(t_1)$  y  $(t_2)$  sean buffers de entrada de un mismo  $TSF_1$ .

Además, sera necesario añadir arcos desde algunas transiciones a buffers. El objetivo de estos arcos es reconstruir el marcado de un buffer que ha sido preasignado a una transición de un T-semiflujo concreto y finalmente este no se ha disparado.

Vamos a considerar la red DSSP de la Fig. 4.2 formada por los arcos de color negro y rojo. En la preasignación de los buffers, los arcos rojos son eliminados mientras que los azules son añadidos.

La estructura original es una DSSP compuesta por dos agentes  $\mathcal{N}_1$  y  $\mathcal{N}_2$ , ambos tienen 4 T-semifujos locales que son dados en Cuadro 4.2. Además, la red global tiene también 4 T-semiflujos ( $TSF_1$  según Def. 4.5). Hay que tener en cuenta que el T-semiflujo global  $\mathbf{x}_g^1$  está formado por  $\mathbf{x}_1^1$  (T-semiflujo local de  $\mathcal{N}_1$ ) y  $\mathbf{x}_1^2$  (T-semiflujo local de  $\mathcal{N}_2$ ), es decir,  $\mathbf{x}_g^1 = \mathbf{x}_1^1 + \mathbf{x}_1^2$ . Además,  $\mathbf{x}_g^2 = \mathbf{x}_2^1 + \mathbf{x}_2^2$ ,  $\mathbf{x}_g^3 = \mathbf{x}_3^1 + \mathbf{x}_3^2$  y  $\mathbf{x}_g^4 = \mathbf{x}_4^1 + \mathbf{x}_4^2$ .

Cuadro 4.2: T-semiflujo locales en la red de la Fig 4.2

Id.	Red	Transiciones	$\mathbf{X}_g$ a los que pertenece
$\mathbf{x}_1^1$	$\mathcal{N}_1$	$t_1-t_3$	$\mathbf{x}_g^1$
$\mathbf{x}_2^1$	$\mathcal{N}_1$	$t_1, t_9, t_{10}$	$\mathbf{x}_g^2$
$\mathbf{x}_3^1$	$\mathcal{N}_1$	$t_{11}-t_{13}$	$\mathbf{x}_g^3$
$\mathbf{x}_4^1$	$\mathcal{N}_1$	$t_{11}, t_{19}, t_{20}$	$\mathbf{x}_g^4$
$\mathbf{x}_1^2$	$\mathcal{N}_2$	$t_4-t_6$	$\mathbf{x}_g^1$
$\mathbf{x}_2^2$	$\mathcal{N}_2$	$t_6-t_7$	$\mathbf{x}_g^2$
$\mathbf{x}_3^2$	$\mathcal{N}_2$	$t_{14}-t_{16}$	$\mathbf{x}_g^3$
$\mathbf{x}_4^2$	$\mathcal{N}_2$	$t_{16}-t_{18}$	$\mathbf{x}_g^4$

En este ejemplo se muestra que si los buffers son preasignados mirando solo los T-semiflujos locales, el sistema puede llegar a bloqueo. La parte de la izquierda, compuesta por  $\mathbf{x}_g^1$  y  $\mathbf{x}_g^2$  ha sido preasignada sin tener en cuenta los T-semiflujos globales de la siguiente forma: en primer lugar, para controlar  $\mathbf{x}_1^1$ , el  $b_1$  se preasigna a  $t_1$  asegurando que  $\mathbf{x}_1^1$  comienza a dispararse solo si  $b_1$  tiene marca. Para hacer esto, el arco  $(b_1, t_3)$  es reemplazado por  $(b_1, t_1)$ . Se añade también el arco  $(t_{10}, b_1)$  para devolver la marca al buffer si el conflicto en  $p_2$  se resuelve a través de  $t_{10}$ . Como  $\mathbf{x}_1^1$  forma parte del T-semiflujo global de tipo 1 ( $TSF_1$ )  $\mathbf{x}_g^1 = \mathbf{x}_1^1 + \mathbf{x}_1^2$  ahora habría que realizar la preasignación en  $\mathbf{x}_1^2$ . Sin embargo, se realiza la preasignación de  $\mathbf{x}_2^1$  eliminando el arco  $(b_3, t_9)$  y añadiendo  $(b_3, t_{10})$ . Del mismo modo se realizan las preasignaciones de  $\mathbf{x}_2^2$  (los arcos  $(b_4, t_6)$  y  $(t_5, b_4)$  son añadidos y  $(b_4, t_8)$  se elimina) seguida de  $\mathbf{x}_1^2$  ( $(b_2, t_5)$  se añade y  $(b_2, t_4)$  se elimina). Ahora la parte izquierda de la red puede bloquearse. Esto ocurre si la secuencia  $\sigma = t_1 t_2 t_3$  es disparada en el marcado de la figura. Mediante el disparo de  $\sigma$ , se elimina una marca de  $b_1$  y crea una marca en  $b_2$ . Ahora, las transiciones  $t_1$  y  $t_6$  nunca pueden

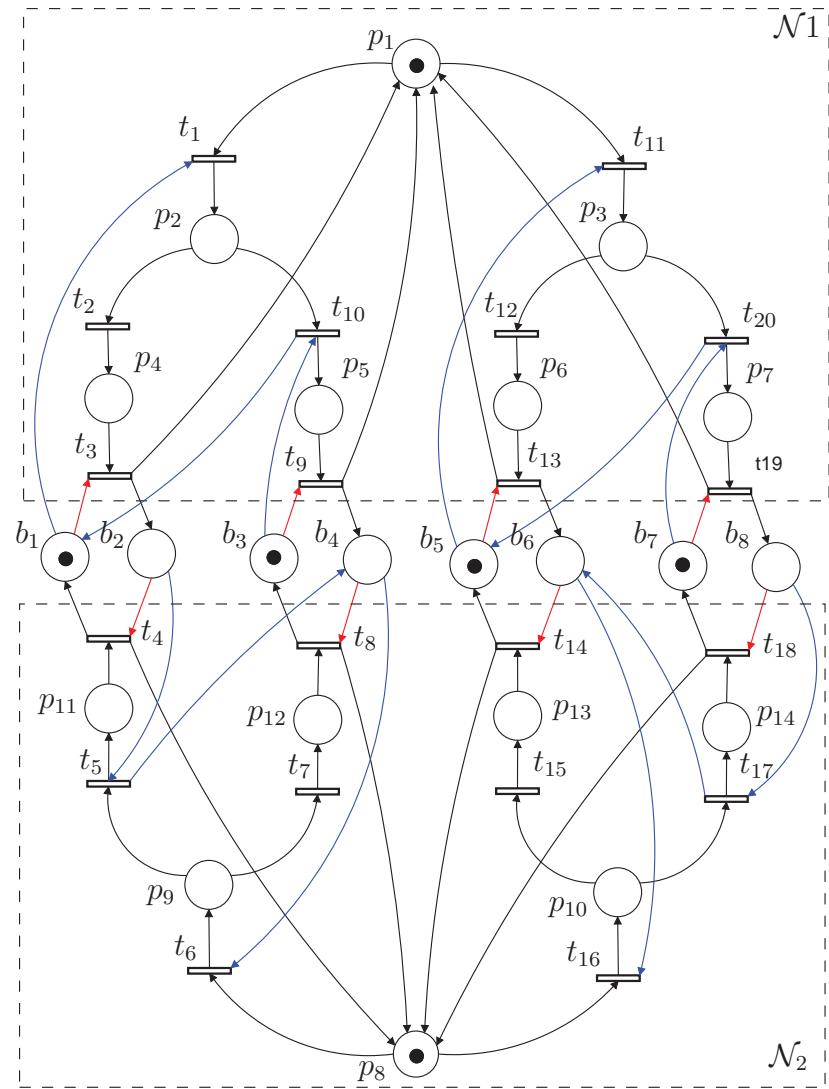


Figura 4.2: Preasignación desde una perspectiva local frente a una global.

dispararse, ya que ambas tienen sus lugares de entrada (buffers) vacíos, es decir,  $b_1 \in \bullet t_1$  y  $b_4 \in \bullet t_6$ .

En la parte de la derecha si se ha realizado una preasignación considerando los T-semiflujos globales. Primero se ha preasignado  $\mathbf{x}_g^3$  y luego  $\mathbf{x}_g^4$ . Esta parte si es viva, ya que siempre podrá dispararse  $\mathbf{x}_g^1$  o  $\mathbf{x}_g^2$  (T-semiflujos locales de  $\mathbf{x}_g^3$ ). Además, los buffers de entrada de  $\mathbf{x}_g^4$  ( $b_7$  y  $b_8$ ) siempre habilitaran el disparo de  $\mathbf{x}_g^1$  o  $\mathbf{x}_g^2$  (T-semiflujos locales de  $\mathbf{x}_g^4$ ), aunque estos están también condicionados por la posibilidad de disparar  $\mathbf{x}_g^1$  y  $\mathbf{x}_g^2$  respectivamente.

El Alg. 4.6 proporciona los pasos a seguir para la preasignación de buf-

fers en una red DSSP con  $TSF_1$ . En primer lugar es necesario calcular todos los T-semiflujos locales de cada máquina. Luego se calculan los globales y se comprueba que son  $TSF_1$  (Def. 4.5). Conociendo cuales son los T-semiflujos locales que forman parte de alguno global, se calcula el lugar de espera  $p_e$  (Def. 4.4) de cada máquina. Se ordenan las secuencias de disparo de todos los T-semiflujos locales partiendo desde el  $p_e$  de su máquina. De forma secuencial, en cada  $TSF_1$ , se realiza lo siguiente: para cada  $\mathbf{x}_l$  que lo compone se preasignan los buffer ( $b_i$ ) de entrada a la primera transición ( $t_k$ ) de la secuencia ordenada de  $\mathbf{x}_l$  que no haya sido preasignada previamente. Cada una de las secuencias ordenadas  $\mathbf{x}_l^j$  que forman parte de la misma máquina ( $\mathcal{N}_j$ ) que  $\mathbf{x}_l$  y que contienen la transición ( $t_k$ ) son comparadas transición a transición con la secuencia ordenada de  $\mathbf{x}_l$  a partir de ( $t_k$ ). La primera transición ( $t_p \in \mathbf{x}_l^j$ ) que no coincide, devolverá las marcas a los buffers ( $b_i$ ) preasignados.

**Algorithm 4.6.** *procedure* PREASIGNAR BUFFERS EN REDES DSSP  
 CON  $\mathbf{x}_g$   $TSF_1$

*Input:*  $\{\mathcal{N}; \mathcal{N}_1, \dots, \mathcal{N}_i\}$

*Calcular T-semiflujo locales( $\mathbf{x}_l$ ), globales completos ( $\mathbf{x}_g$ ) y lugares de espera ( $p_e$ )*

*Ordenar las secuencias de transiciones de los  $\mathbf{x}_l$  partiendo de ( $p_e$ ).*

**for** todo  $\mathbf{x}_g$  **do**

**for** todo  $\mathbf{x}_l$  que compone  $\mathbf{x}_g$  **do**

*Preasignar todos buffer ( $b_i$ ) de entrada de  $\mathbf{x}_l$  a la primera transición ( $t_k$ ) vista desde el lugar de espera que no haya sido preasignada previamente.*

*Sea  $\mathcal{N}_j$  la máquina a la que pertenece  $\mathbf{x}_l$*

**for** todo  $\mathbf{x}_l^j$  de  $\mathcal{N}_j$  menos  $\mathbf{x}_l$  **do**

**if**  $t_k \in \mathbf{x}_l^j$  **then**

*Comparar las secuencias de transiciones  $\mathbf{x}_l$  con  $\mathbf{x}_l^j$  a partir de ( $t_k$ )*

*La primera transición ( $t_p \in \mathbf{x}_l^j$ ) que no coincide, devolverá las marcas a los buffers ( $b_i$ ) preasignados.*

**end if**

**end for**

**end for**

**end for**

**end procedure**

Se va a aplicar el Alg. 4.6 a la parte derecha de la red de la Fig. 4.2.

Los T-semiflujos locales y globales aparecen en la Tab. 4.2. Como todos los T-semiflujos locales tanto de  $\mathcal{N}_1$  como de  $\mathcal{N}_2$  pertenecen a T-semiflujos globales, sus circuitos se utilizan para calcular los lugares de espera. Los lugares  $p_1$  y  $p_8$  son los lugares de espera de  $\mathcal{N}_1$  y  $\mathcal{N}_2$  respectivamente ya que  $p_1$  es el único lugar común de los circuitos generados por  $\mathbf{x}_1^1$ ,  $\mathbf{x}_2^1$ ,  $\mathbf{x}_3^1$  y  $\mathbf{x}_4^1$  para  $\mathcal{N}_1$  y  $p_8$  es el único lugar común de los circuitos generados por  $\mathbf{x}_1^2$ ,  $\mathbf{x}_2^2$ ,  $\mathbf{x}_3^2$  y  $\mathbf{x}_4^2$  para  $\mathcal{N}_2$ . Las secuencias ordenadas partiendo de los lugares de espera y sus buffers de entrada/salida son:

1. Secuencias de los T-semiflujos de  $\mathcal{N}_1$  partiendo de  $p_1$

- $\mathbf{x}_1^1 \Rightarrow \{t_1 t_2 t_3\} \implies b_1/b_2$  (buffer entrada/buffer salida)
- $\mathbf{x}_2^1 \Rightarrow \{t_1 t_{10} t_9\} \implies b_3/b_4$
- $\mathbf{x}_3^1 \Rightarrow \{t_{11} t_{12} t_{13}\} \implies b_5/b_6$
- $\mathbf{x}_4^1 \Rightarrow \{t_{11} t_{20} t_{19}\} \implies b_7/b_8$

2. Secuencias de los T-semiflujos de  $\mathcal{N}_2$  partiendo de  $p_8$

- $\mathbf{x}_1^2 \Rightarrow \{t_6 t_5 t_4\} \implies b_2/b_1$
- $\mathbf{x}_2^2 \Rightarrow \{t_6 t_7 t_8\} \implies b_4/b_3$
- $\mathbf{x}_3^2 \Rightarrow \{t_{16} t_{15} t_{14}\} \implies b_6/b_5$
- $\mathbf{x}_4^2 \Rightarrow \{t_{16} t_{17} t_{18}\} \implies b_8/b_7$

Se elige realizar la preasignación primero a  $\mathbf{x}_g^3$  y luego a  $\mathbf{x}_g^4$ . El  $TSF_1$   $\mathbf{x}_g^3 = \mathbf{x}_3^1 + \mathbf{x}_3^2$ , por lo que habrá que preasignar los buffer de entrada de ambos T-semiflujos locales  $\mathbf{x}_3^1$  y  $\mathbf{x}_3^2$ . Para la preasignación de  $\mathbf{x}_3^1$  tendremos que fijarnos en sus buffers de entrada ( $b_5$ ) y en su secuencia de disparo. De este modo, se elimina el arco ( $b_5, t_{13}$ ) y se añade un nuevo arco ( $b_5, t_{11}$ ) desde el buffer de entrada ( $b_5$ ) a la primera transición de la secuencia ordenada  $\mathbf{x}_3^1$  que no haya sido preasignada ( $t_{11}$ ). Ahora, con el objetivo de recargar el marcado del buffer  $b_5$  si se inicia el disparo de  $\mathbf{x}_3^1$ , pero finalmente no se completa, nos fijaremos en las demás secuencias de disparo de la máquina  $\mathcal{N}_1$  que contengan  $t_{11}$ . Como la secuencia generada por  $\mathbf{x}_4^1$  contiene la transición ( $t_{11}$ ), se comparan las siguientes transiciones de las secuencias  $\mathbf{x}_3^1$  y  $\mathbf{x}_4^1$ . Al no coincidir ( $t_{12}$  con  $t_{20}$ ), existirá un arco ( $t_{11}, b_5$ ) que recargará el marcado de  $b_5$ . Además como la transición  $t_{11}$  ya ha sido preasignada, se eliminará de la secuencia ordenada de  $\mathbf{x}_4^1$ .

Las modificaciones pueden representarse gráficamente del siguiente modo: cada color representa los cambios realizados en las preasignaciones de

cada  $TSF_1$ . Un círculo rodeando una transición indica que existe un arco desde el buffer de ese color, hasta la transición. Una raya sobre la transición indica que el arco va desde la transición hasta el buffer. En caso de que una transición esté tachada significa que ya ha sido preasignada. En rojo aparece la preasignación de  $x_3^1$  en la máquina  $\mathcal{N}_1$ :

- $x_1^1 \Rightarrow \{t_1 t_2 t_3\} \implies b_1/b_2$
- $x_2^1 \Rightarrow \{t_1 t_{10} t_9\} \implies b_3/b_4$
- $x_3^1 \Rightarrow \{t_{11} \text{ (red circle)} t_{12} t_{13}\} \implies b_5/b_6$
- $x_4^1 \Rightarrow \{t_{11} \text{ (green circle)} \bar{t}_{20} t_{19}\} \implies b_7/b_8$

Ahora habrá que preasignar  $x_3^2$ , el otro T-semiflujo local que compone  $x_g^3$ . Su preasignación (rojo) queda del siguiente modo:

- $x_1^2 \Rightarrow \{t_6 t_5 t_4\} \implies b_2/b_1$
- $x_2^2 \Rightarrow \{t_6 t_7 t_8\} \implies b_4/b_3$
- $x_3^2 \Rightarrow \{t_{16} \text{ (red circle)} t_{15} t_{14}\} \implies b_6/b_5$
- $x_4^2 \Rightarrow \{t_{16} \text{ (green circle)} \bar{t}_{17} t_{18}\} \implies b_8/b_7$

Una vez todos los T-semiflujos locales que componen  $x_g^3$  han sido preasignados se procede a preasignar otro T-semiflujo global. En este caso  $x_g^4$  que está compuesto por  $x_4^1$  y  $x_4^2$ . Las modificaciones realizadas aparecen en color verde. En la Fig. 4.2 se pueden observar las modificaciones realizadas en los  $TSF_1$   $x_g^3$  y  $x_g^4$ . Los arcos añadidos aparecen en azul, mientras los arcos rojos han sido eliminados.

**Lemma 4.7.** *Dada una estructura DSSP consistente y conservativa en la cual todos los T-semiflujos son  $TSF_1$ , aplicando el Alg. 4.6, la nueva red SSP es estructuralmente viva*

*Demostración.* Si la red es consistente y conservativa para cada  $TSF_1$ , existe una secuencia de disparo de sus T-semiflujos locales que permiten a la estructura volver a su marcado inicial, habiendo disparado todos los T-semiflujos locales que lo componen. Además los  $TSF_1$  tienen buffers privados, es decir, sus marcas solo pueden ser consumidas/producidas por T-semiflujos locales del mismo  $TSF_1$ . Mediante la preasignación y recarga de

los buffers, no se modifica el "juego de marcas" ya que estas son consumidas en transiciones previas, pero si finalmente no se dispara la transición a la que el buffer estaba realmente asignado, se recarga el marcado del mismo. La preasignación se realiza de forma secuencial a cada  $TSF_1$ , de modo que, al preasignar el primer  $TSF_1$  ( $\mathbf{x}_g^i$ ) estamos asegurando que solo podrán dispararse los T-semiflujos locales pertenecientes a ( $\mathbf{x}_g^i$ ) que tenga suficientes marcas en sus buffers de entrada para completar su disparo. Además con un marcado inicial adecuado, podrá repetirse indefinidamente el disparo secuencial de todos los T-semiflujos locales de ( $\mathbf{x}_g^i$ ), siendo este  $TSF_1$  vivo. En la próxima preasignación de otro  $TSF_1$  ( $\mathbf{x}_g^k$ ), del mismo modo, se está asegurando que solo podrán dispararse los T-semiflujos locales pertenecientes a ( $\mathbf{x}_g^k$ ) que tenga suficientes marcas en sus buffers de entrada para completar su disparo y que se dispararan secuencialmente sus T-semiflujos locales. Si en alguna máquina, un T-semifluido local  $\mathbf{x}_1$  de  $\mathbf{x}_g^k$  comparte alguna transición con otro T-semifluido local  $\mathbf{x}_2$  que pertenece a un  $TSF_1$  ya preasignado ( $\mathbf{x}_g^i$ ) el disparo de  $\mathbf{x}_1$  está condicionado a que  $\mathbf{x}_2$  esté habilitado. Como  $\mathbf{x}_2$  es un T-semifluido local vivo de  $\mathbf{x}_g^i$ , siempre podrá estar habilitado en alguna futura evolución del sistema. Así queda demostrado que una vez preasignando ( $\mathbf{x}_g^k$ ) también es vivo. Aplicando la misma lógica, se puede deducir que preasignando secuencialmente todos los  $TSF_1$ , se obtendrá un sistema SSP vivo.  $\square$

## 4.2. Aplicación del algoritmo a $TSF_2$

Si todos los  $\mathbf{x}_g$  de una red no son  $TSF_1$  es necesario realizar algunos pasos adicionales en el algoritmo. En esta sección se define una nueva estructura denominada T-semifluido tipo 2 ( $TSF_2$ ), cuya definición es la misma que en los  $TSF_1$  (Def. 4.5) pero permitiendo que los buffers no sean privados. Esta no privacidad de los buffer implica que las marcas pueden ser consumidas por distintos  $TSF_2$ . Por esto, en algunas ocasiones, aunque se preasigne un  $TSF_2$ , no se está garantizando que siempre haya un T-semifluido local de este  $TSF_2$  habilitado por sus buffers de entrada. Para solucionar este problema, se van a añadir unos arcos de lectura a la red, que bajo ciertas condiciones impidan el disparo de algunas transiciones.

Para poder entender el problema que aparece nos fijaremos en red de la Fig. 4.3. La red formada por los arcos negros y rojos, es una DSSP compuesta por 2 agentes ( $\mathcal{N}_1$  y  $\mathcal{N}_2$ ) que cooperan a través de 5 buffers. Cada máquina contiene 3 T-semiflujos locales y a su vez el sistema consta de 3

T-semiflujos globales. Los secuencias ordenas a partir del lugar de espera de los T-semiflujos locales de ambas máquinas, así como sus buffers de entra/salida y la composición de los T-semiflujos globales son:

1. Secuencias de los T-semiflujos de  $\mathcal{N}_1$  partiendo de  $p_1$

- $\mathbf{x}_1^1 \Rightarrow \{t_1 t_2\} \Rightarrow b_2/b_1$  (buffer entrada/buffer salida)
- $\mathbf{x}_2^1 \Rightarrow \{t_3 t_4\} \Rightarrow b_2/b_3$
- $\mathbf{x}_3^1 \Rightarrow \{t_6 t_5\} \Rightarrow b_4/b_5$

2. Secuencias de los T-semiflujos de  $\mathcal{N}_2$  partiendo de  $p_2$

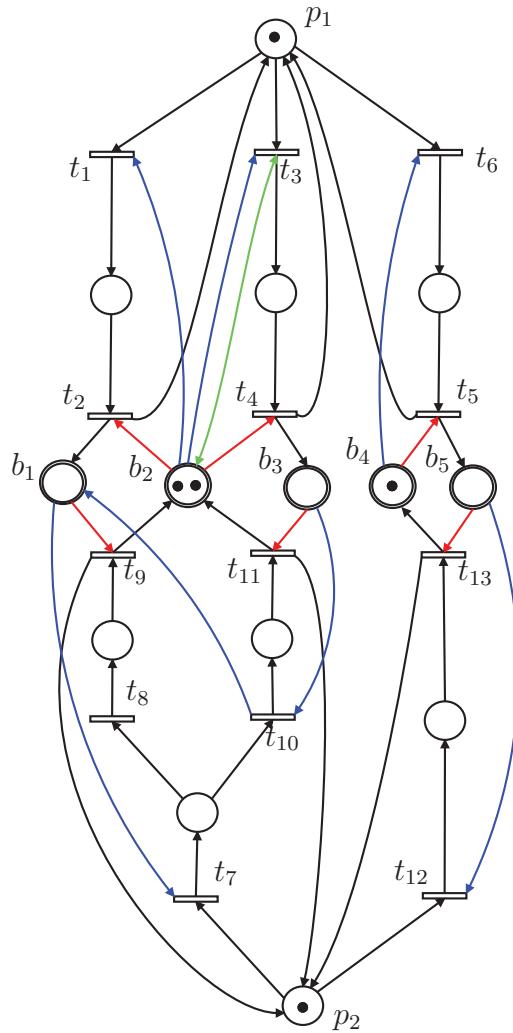
- $\mathbf{x}_1^2 \Rightarrow \{t_7 t_8 t_9\} \Rightarrow b_1/b_2$
- $\mathbf{x}_2^2 \Rightarrow \{t_7 t_{10} t_{11}\} \Rightarrow b_3/b_2$
- $\mathbf{x}_3^2 \Rightarrow \{t_{12} t_{13}\} \Rightarrow b_5/b_4$

3. T-semiflujos globales, buffers relacionados y tipo

- $\mathbf{x}_g^1 = \mathbf{x}_1^1 + \mathbf{x}_1^2 \quad (b_1, b_2) \Rightarrow b_2$  no privado ( $TSF_2$ )
- $\mathbf{x}_g^2 = \mathbf{x}_2^1 + \mathbf{x}_2^2 \quad (b_2, b_3) \Rightarrow b_2$  no privado ( $TSF_2$ )
- $\mathbf{x}_g^3 = \mathbf{x}_3^1 + \mathbf{x}_3^2 \quad (b_4, b_5) \Rightarrow$  todos privados ( $TSF_1$ )

Tras aplicar el Alg. 4.6 a la red siguiendo la siguiente secuencia:  $\mathbf{x}_g^3$ ,  $\mathbf{x}_g^1$  y  $\mathbf{x}_g^2$ , se obtiene la estructura formada por los arcos negros y rojos. Esta estructura no es viva, ya que disparando la secuencia  $\sigma = t_3 t_4 t_3 t_4$  los T-semiflujos  $\mathbf{x}_g^1$  y  $\mathbf{x}_g^2$  se mueren. Esto es debido a que el buffer  $b_2$  de  $\mathbf{x}_g^1$  no es privado y sus marcas pueden ser consumidas por otro T-semiflujo global ( $\mathbf{x}_g^2$ ). De este modo, a pesar de haber preasignado en primer lugar los T-semiflujos locales de  $\mathbf{x}_g^1$ , no se puede garantizar que siempre tenga un T-semiflujo local habilitado. Para impedir esta situación, es necesario que la ultima marca de  $b_2$  sea siempre consumida por el  $\mathbf{x}_1^1$ , para ello se añade un arco de lectura desde  $b_2$  a  $t_3$ . En la Fig. 4.3 el arco de lectura aparece en color verde.

Para añadir estos arcos es necesario realizar algunas modificaciones en el Alg. 4.6: Se realizará primero la preasignación a los  $TSF_1$  y posteriormente de  $TSF_2$  guardando el orden en que estos últimos se han preasignado. Además, si durante la preasignación de  $TSF_2$  desde un buffer a una transición existe un arco Pre y otro Post, estos se contrarrestaran según sus pesos dejando: 1) un único arco en caso de que el peso de uno de ellos sea mayo que el del otro o, 2) suprimiéndose si sus pesos son iguales.

Figura 4.3: DSSP con  $TSF_2$ 

Una vez todos los T-semiflujos globales han sido preasignados, se identificarán los grupos de  $TSF_2$  que tienen buffers ( $b_i$ ) de entrada comunes y se observa la secuencia en la que dichos  $TSF_2$  de cada grupo fueron preasignados. Para el  $TSF_2$  preasignado en enésimo lugar, existirá un arco de lectura desde ( $b_i$ ) hasta la transición ( $t_j$ ) de peso n-1. Donde ( $t_j$ ) es la transición a la que ( $b_i$ ) había sido preasignado durante la aplicación del Alg. 4.6 a  $TSF_2$ . Si existen arcos previos entre el buffer y la transición, a diferencia que durante la preasignación, estos no se contrarrestan, sino que se suman los Pre y Post por separado y se dejan ambos arcos.

Siguiendo con el ejemplo de la Fig. 4.3, existe un conjunto de  $TSF_2$   $\{\mathbf{x}_g^1$  y  $\mathbf{x}_g^2\}$  que comparten un buffer de entrada común ( $b_2$ ). El orden en el que

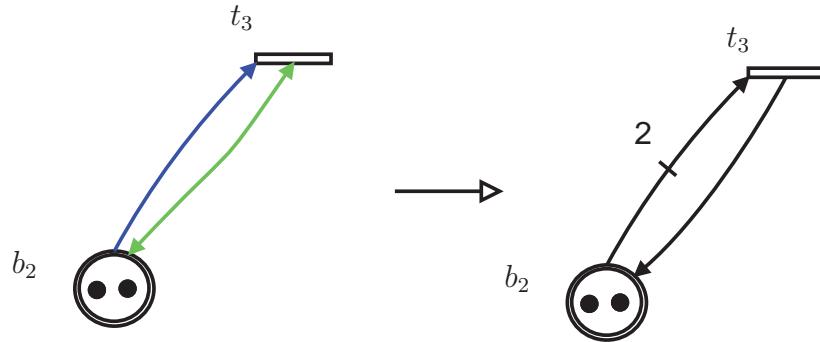


Figura 4.4: Combinación al añadir arcos de lectura

se han preasignado es  $\mathbf{x}_g^1-\mathbf{x}_g^2$ , así que existirá un arco de lectura entre  $b_2$  y  $t_3$  de peso 1 y otro entre  $b_2$  y  $t_1$  de peso 0. El arco de peso 0 no aporta nada a la red, mientras que el arco de lectura entre  $b_2$  y  $t_3$  debe sumarse a el arco ya existente  $(b_2, t_3)$ . Como resultado de esta suma aparece un arco de peso 2  $(b_2, t_3)$  y un arco de peso 1  $(t_3, b_2)$ . En la red de la Fig. 4.3 aparecen los arcos entre  $b_2$  y  $t_3$  sin combinar. La Fig. 4.4 representa como quedaría esta combinación.

**Algorithm 4.8.** *procedure* PREASIGNAR BUFFERS EN REDES DSSP  
 CON  $\mathbf{x}_g$   $TSF_1$  Y  $TSF_2$   
*Input:*  $\{\mathcal{N}; \mathcal{N}_1, \dots, \mathcal{N}_i\}$   
*Aplicar el Alg. 4.6 preasignando secuencialmente los  $TSF_1$  y luego  $TSF_2$  guardando el orden en que han sido preasignados los  $TSF_2$ .*  
*Calcular los conjuntos  $(\Omega)$  de  $TSF_2$  que tienen buffers de entrada común.*  
**for** cada conjunto  $\Omega$  que tiene un buffer  $(b_i)$  de entrada común. **do**  
*Ver el orden ( $n$ ) en que cada  $TSF_2 \in \Omega$  ha sido preasignado*  
**for** todo  $TSF_2 \in \Omega$  **do**  
*Añadir arco de lectura de peso  $(n-1)$  entre  $b_i$  y  $t_k$  donde  $t_k$  es la transición a la que  $b_i$  había sido preasignado al aplicar el Alg. 4.6 en  $TSF_2$  y  $n$  el orden de preasignación de  $TSF_2$  en  $\Omega$ .*  
**end for**  
**end for**  
**end procedure**

### 4.3. Aplicación del algoritmo a $TSF_3$

Desafortunadamente, si todos los T-semiflujos globales de una red no son  $TSF_1$  o  $TSF_2$ , no se puede asegurar que aplicando el algoritmo se consiga una estructura SSP viva, pero se puede establecer una condición para comprobar si la red resultante será estructuralmente viva.

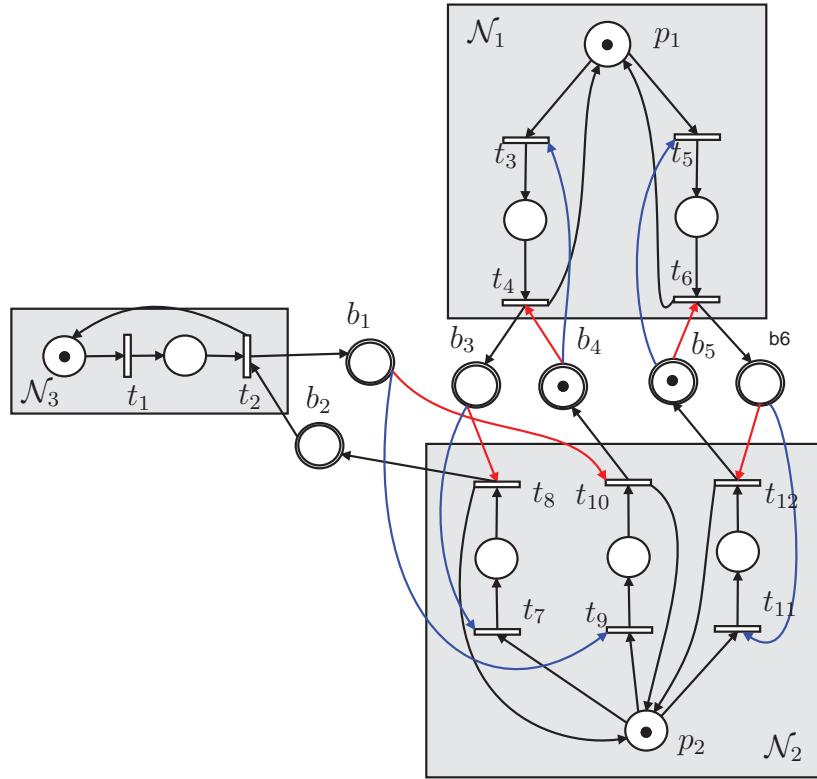
En primer lugar vamos a definir los  $TSF_3$  como  $TSF_1$  (Def. 4.5) que no cumplen la restricción 2, es decir, son T-semiflujos globales en los que sus buffers son privados, pero pueden tener mas de un T-semiflujo local por agente. Estos T-semiflujos globales ( $TSF_3$ ) serán preasignados una vez se hayan preasignado los  $TSF_1$  y los  $TSF_2$ . El problema que aparece en la preasignación de los  $TSF_3$  es debido a la posibilidad de que un mismo T-semiflujo global tenga varios T-semiflujos locales no disjuntos en un mismo agente. Si esto sucede, una vez preasignado, el disparo de un T-semiflujo local perteneciente al  $TSF_3$  puede depender del marcado de un buffer de entrada de otro T-semiflujo local perteneciente al mismo  $TSF_3$  lo cual supone un bloqueo del  $TSF_3$ .

La red que aparece en la Fig. 4.5 (arcos negros y rojos) es una DSSP compuesta por tres agentes ( $\mathcal{N}_1$ ,  $\mathcal{N}_2$  y  $\mathcal{N}_3$ ) que tiene un  $TSF_1$  ( $\mathbf{x}_g^1 = \mathbf{x}_2^1 + \mathbf{x}_3^1$ ) y un  $TSF_3$  ( $\mathbf{x}_g^2 = \mathbf{x}_1^1 + \mathbf{x}_1^2 + \mathbf{x}_2^2 + \mathbf{x}_1^3$ ) los T-semiflujos locales que componen la estructura se muestran en la Tab. 4.3. Como puede observarse todos los T-semiflujos locales del  $TSF_3$  son disjuntos, así que se consigue una estructura SSP viva al realizar la preasignación de los buffers. En la Fig. 4.5 aparece la red SPP (arcos negros y azules) resultante de la preasignación de  $(\mathbf{x}_g^1)$  y  $(\mathbf{x}_g^2)$ .

Cuadro 4.3: T-semiflujo locales en la red de la Fig 4.5

Id.	Red	Transiciones	$\mathbf{X}_g$ a los que pertenece
$\mathbf{x}_1^1$	$\mathcal{N}_1$	$t_3, t_4$	$\mathbf{x}_g^2$
$\mathbf{x}_2^1$	$\mathcal{N}_1$	$t_5, t_6$	$\mathbf{x}_g^1$
$\mathbf{x}_1^2$	$\mathcal{N}_2$	$t_7, t_8$	$\mathbf{x}_g^2$
$\mathbf{x}_2^2$	$\mathcal{N}_2$	$t_9, t_{10}$	$\mathbf{x}_g^2$
$\mathbf{x}_3^2$	$\mathcal{N}_2$	$t_{11}, t_{12}$	$\mathbf{x}_g^1$
$\mathbf{x}_1^3$	$\mathcal{N}_2$	$t_1, t_2$	$\mathbf{x}_g^2$

Podemos concluir que siempre que los T-semiflujos locales que forman un  $TSF_3$  sean disjuntos se puede aplicar el Alg. 4.9 y la red SSP resultante será estructuralmente viva.

Figura 4.5: DSSP con  $TSF_3$  disjuntos

Vamos a realizar ahora un cambio en la red de la Fig. 4.5 que provoque que los T-semiflujos locales ( $\mathbf{x}_1^2$  y  $\mathbf{x}_2^2$ ) que forman el  $TSF_3$  no sean disjuntos. Para ello se añade la transición  $t_{13}$  en  $\mathcal{N}_2$ , lo que implica que  $\mathbf{x}_1^2 = t_{13} + t_7 + t_8$ , mientras que  $\mathbf{x}_2^2 = t_{13} + t_9 + t_{10}$ . La red DSSP resultante de esta modificación aparece en la Fig. 4.6 (arcos negros y rojos), mientras que la estructura SSP resultante de aplicar el algoritmo también se puede ver en esta misma figura (arcos negros y azules). En este caso la estructura SSP resultante no es viva, ya que disparando la secuencia  $\sigma = t_3 t_4 t_{13} t_7 t_8 t_1 t_2$  hace que el  $TSF_3$  se bloquee y nunca más puedan dispararse sus transiciones.

Sin embargo, no necesariamente siempre que los T-semiflujos locales que forman un  $TSF_3$  sean no disjuntos la red SSP resultante no será estructuralmente viva. Vamos a añadir a la estructura DSSP de la Fig. 4.6 un nuevo  $TSF_1$  ( $\mathbf{x}_g^3$ ) que está compuesto por  $\mathbf{x}_3^1 = t_{16} + t_{17}$  y  $\mathbf{x}_4^2 = t_{13} + t_{14} + t_{15}$ , es decir,  $\mathbf{x}_g^3 = \mathbf{x}_3^1 + \mathbf{x}_4^2$ . La red DSSP resultante de añadir  $\mathbf{x}_g^3$  se ilustra en la Fig. 4.7 (arcos rojos y negros).

Aplicando el Alg. 4.9 se obtiene una SSP estructuralmente viva. Esta SSP está ilustrada en la Fig. 4.7 (arcos negros y azules). A pesar de que el

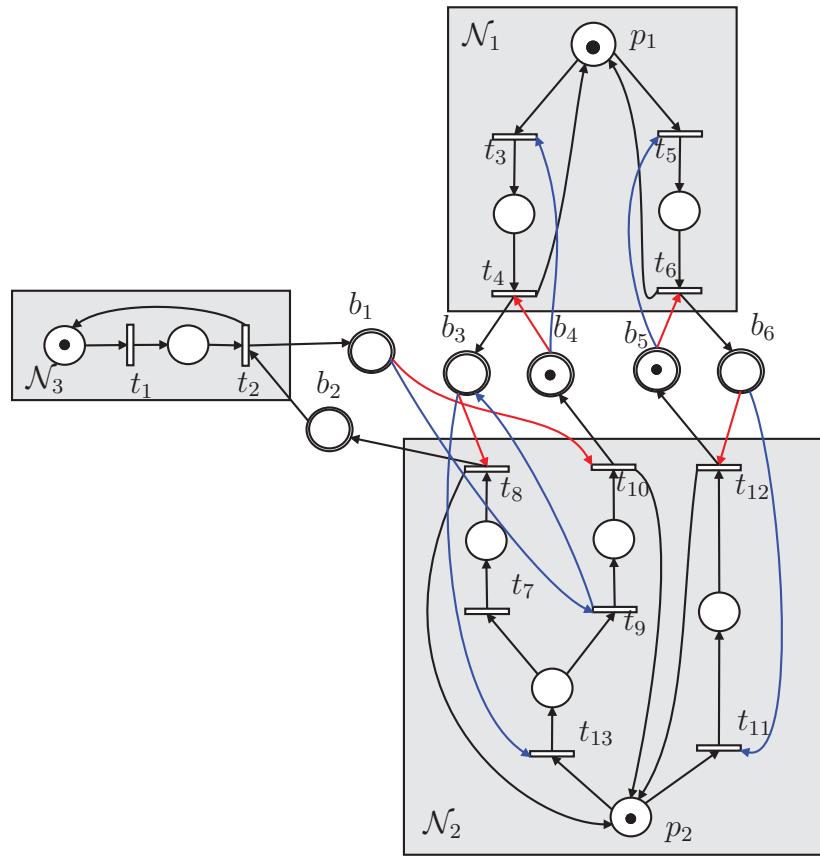


Figura 4.6: DSSP con  $TSF_3$  no disjuntos y SSP estructuralmente no viva

$TSF_3$  ( $\mathbf{x}_g^2$ ) no es disjunto, la red SSP resultante es viva, esto es debido a que durante el proceso de preasignación de  $\mathbf{x}_g^3$  ( $TSF_1$ ), se ha preasignado el buffer  $b_8$  a la transición  $t_{13}$  que era la transición que tenían en común los dos T-semiflujos locales no disjuntos ( $\mathbf{x}_1^2$  y  $\mathbf{x}_2^2$ ) del  $TSF_3$  ( $\mathbf{x}_g^2$ ). Con esto se puede concluir que es posible preasignar  $TSF_3$  que contienen T-semiflujos locales inicialmente no disjuntos y obtener una SSP estructuralmente viva si en el momento que se va a realizar la preasignación del  $TSF_3$  los T-semiflujos locales (sin contar las transiciones que ya han sido preasignadas) que forman el  $TSF_3$  son conjuntos disjuntos.

**Algorithm 4.9.** *procedure PREASIGNAR BUFFERS EN REDES DSSP CON  $\mathbf{x}_g$   $TSF_1$ ,  $TSF_2$  Y  $TSF_3$*

*Input:*  $\{\mathcal{N}; \mathcal{N}_1, \dots, \mathcal{N}_i\}$

*Aplicar el Alg. 4.6 preasignando secuencialmente los  $TSF_1$  y luego  $TSF_2$  guardando el orden en que han sido preasignados los  $TSF_2$ .*

*Calcular los conjuntos ( $\Omega$ ) de  $TSF_2$  que tienen buffers de entrada*

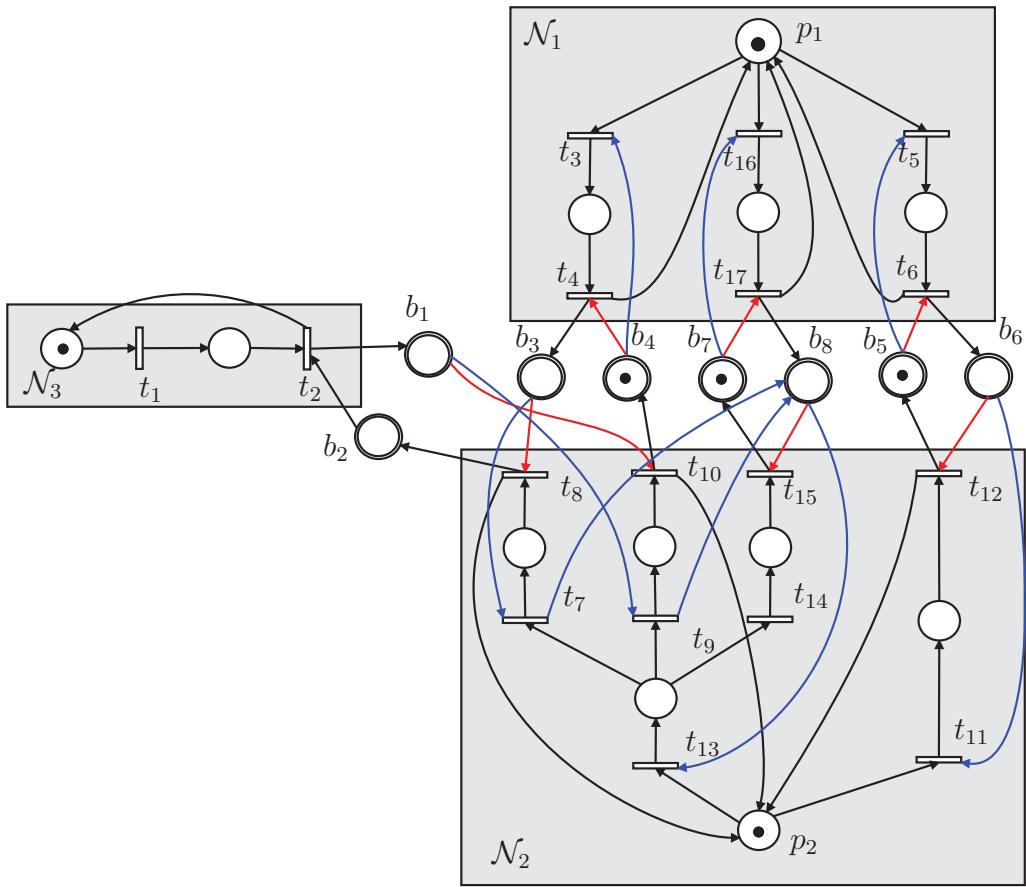


Figura 4.7: DSSP con  $TSF_3$  no disjuntos y SSP estructuralmente viva

común.

**for** cada conjunto  $\Omega$  que tiene un buffer ( $b_i$ ) de entrada común. **do**

Ver el orden ( $n$ ) en que cada  $TSF_2 \in \Omega$  ha sido preasignado

**for** todo  $TSF_2 \in \Omega$  **do**

Añadir arco de lectura de peso ( $n-1$ ) entre  $b_i$  y  $t_k$  donde  $t_k$  es la transición a la que  $b_i$  había sido preasignado al aplicar el Alg. 4.6 en  $TSF_2$  y  $n$  el orden de preasignación de  $TSF_2$  en  $\Omega$ .

**end for**

**end for**

**for** Cada  $TSF_3$  **do**

Evaluar si los T-semiflujos locales (sin tener en cuenta las transiciones ya preasignadas) que lo forman son conjuntos disjuntos.

**end for**

**if** Todos los  $TSF_3$  evaluados son disjuntos **then**

*Preasignar los  $TSF_3$  y la SSP resultante sera viva*

**else**

*No se puede obtener una SSP estructuralmente viva*

**end if**

**end procedure**

El problema que aparece en la preasignación de los  $TSF_3$  es debido a la posibilidad de que un mismo T-semiflujo global tenga varios T-semiflujos locales en un mismo agente. En los  $TSF_1$  y  $TSF_2$  al tener un único T-semiflujo local por agente, se aseguraba que los T-semiflujos locales que pertenecen a uno global eran conjuntos disjuntos es decir, no podían tener transiciones comunes. En estas estructuras, aplicando la preasignación de los buffers, el disparo de un T-semiflujo local solo puede depender del marcado de sus buffers de entrada y de la habilitación de T-semiflujos locales pertenecientes a otro T-semiflujo global preasignado previamente y vivo.

# **Capítulo 5**

---

## **Conclusiones**

---

Las DSSP son subclases de sistemas P/T modulares que se usan para el modelado de sistemas distribuidos formados por la cooperación de procesos secuenciales. Los resultados estructurales representan una ventaja crucial para el análisis de los modelos de esta subclase red. Concretamente, mediante el teorema de rango es posible analizar la vivacidad estructural en redes DSSP. En este trabajo se ha estudiado el problema de “forzar” la vivacidad estructural mediante la preasignación de los buffers. En algunos casos se pueden obtener redes SSP estructuralmente vivas. Se provee un algoritmo que realiza la preasignación de los buffers en los T-semiflujo locales. Mostramos que esta preasignación debe de hacerse desde una perspectiva global, es decir, aplicada secuencialmente a T-semiflujos globales. Se ha aplicado el algoritmo en redes DSSP que modelan sistemas productivos y de salud. En algunas ocasiones, para conseguir un sistema vivo, puede ser necesario aumentar el marcado de los buffers. En redes que modelan guías clínicas donde las marcas de los buffers representan información, aumentar el marcado de un buffer implica añadir información no existente o creada con anticipación, por lo que se puede estar modificando el comportamiento deseado del sistema. Esto no sucede en redes DSSP que modelan sistemas productivos, ya que aumentar el marcado de un buffers significa tener disponibles almacenes de productos intermedios con mayor capacidad.



---

## Bibliografía

---

- [1] S. Bernardi, C. Mahulea, and J. Albareda. Toward a decision support system for the clinical guidelines assessment. Technical report, I3A, University of Zaragoza, 2016. [cited at p. 1]
- [2] EliaEsther Cano, CarlosA. Rovetto, and José Manuel Colom. An algorithm to compute the minimal siphons in  $S^4PR$  nets. *Discrete Event Dynamic Systems*, 22(4):403–428, 2012. [cited at p. 2]
- [3] J. M. Colom. The resource allocation problem in flexible manufacturing systems. In W.v.d. Aalst and E. Best, editors, *Applications and Theory of Petri Nets*, volume 2679 of *Lecture Notes in Computer Science*, pages 23–35. Springer-Verlag, Berlin, Heidelberg, 2003. [cited at p. 2]
- [4] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993. [cited at p. 5]
- [5] ZhiWu Li and MengChu Zhou. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(1):38–51, Jan 2004. [cited at p. 2, 20]
- [6] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989. [cited at p. 5]
- [7] Jonghun Park and S.A. Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 46(10):1572–1583, Oct 2001. [cited at p. 2]
- [8] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981. [cited at p. 5]

- [9] L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of deterministic systems of sequential processes. *Proc. int. Workshop Structures Concurrency Theory (STRICT)*, (1 and 2):279–293, 1995. [cited at p. 14]
- [10] L. Recalde, E. Teruel, and M. Silva. Modeling and analysis of sequential processes that cooperate through buffers. *IEEE Trans. Robotics and Automation*, 14(2):267 – 277, 1998. [cited at p. 1, 2, 14, 15, 16]
- [11] M. Silva, E. Teruel, and J.-M. Colom. Linear algebraic and linear programming techniques for the analysis of P/T net systems. *Lecture on Petri Nets I: Basic Models*, 1491:309–373, 1998. [cited at p. 6]
- [12] E. Teruel and M. Silva. Structure theory os equal conflict systems. *Theoret. Comput. Sci.*, 153(1 and 2):271–300, 1996. [cited at p. 14]