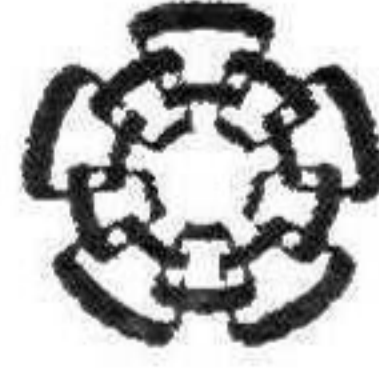


xx(178702.1)



CINVESTAV
BIBLIOTECA CENTRAL

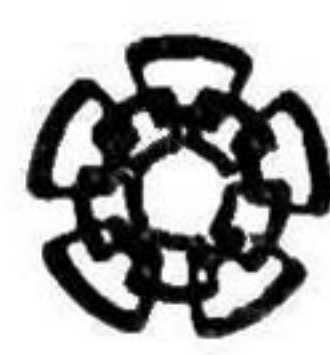


SSIT000004107

TK165, G8

.N86

2009



CENTRO DE INVESTIGACIÓN Y
DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO
NACIONAL

COORDINACIÓN GENERAL DE
SERVICIOS BIBLIOGRÁFICOS

Control por Regulación en Sistemas de Manufactura Flexible usando Redes de Petri Coloreadas

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Saúl Alonso Nuño Sánchez
Ingeniero Industrial

Instituto Tecnológico y de Estudios Superiores de Occidente
2003-2007

Becario de CONACYT, expediente no. 13493

Director de Tesis
Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2009.

**CINVESTAV
IPN
ADQUISICION
DE LIBROS**

CLASIF.:	TK 165.62 N8A 2009
ADQUIS.	SS 1-572
FECHA:	19 III 2010
PROCED.	Don-2010
	\$ _____

ID 163342-1001

Abstract.

This thesis is concerned with the regulation control problem in the Flexible Manufacturing Systems (*FMS*) modeled by Petri Nets (*PN*). Since real *FMS* lead to large *PN* models, this work uses Colored *PN* (*CPN*) to obtain reduced models. Unfortunately, when *CPN* are used, the techniques used to analyze the *FMS* are complex in the general case.

In order to tackle this problem, this work provides a modeling methodology to build *CPN* models where the workflow is captured and the arc functions belong to bisymmetric matrices. Based on this methodology, polynomial algorithms to control the *FMS* are posed.

Since in this work we assume that the specification is less detailed than the *FMS* model, i.e. it is a system model abstraction, then our proposal focuses in finding out a system state partition in order to make comparable the system model with its specification. Afterwards, the controller that force the system to visit the partitions in the order stated by the specification is designed. Finally, the controller is translated to Ladder Diagrams (*LD*) and is implemented into a *PLC*.

Resumen.

Esta tesis aborda el problema de control por regulación en los Sistemas de Manufactura Flexible (*SMF*) modelados por Redes de Petri (*RP*). Puesto que los *SMF* generan grandes modelos en *RP*, este trabajo utiliza *RP* Coloreadas (*RPC*) para obtener modelos reducidos. Desafortunadamente, cuando se usan las *RPC*, las técnicas utilizadas para analizar los *SMF* son complejas en el caso general.

Para enfrentar este problema, este trabajo proporciona una metodología de modelado para construir modelos en *RPC* donde el flujo de trabajo es capturado y las funciones de los arcos pertenecen a las matrices bisimétricas. En base a esta metodología, son propuestos algoritmos polinomiales para controlar el *SMF*.

Como en este trabajo se supone que la especificación es menos detallada que el modelo del *SMF*, es decir, es una abstracción del modelo del sistema, entonces nuestra propuesta se enfoca en encontrar una partición del estado del sistema para hacer comparable el modelo del sistema con su especificación. Después, se diseña el controlador que fuerza al sistema a visitar las particiones en el orden que lo indique la especificación. Finalmente, el controlador es traducido a Diagramas de Escalera (*LD*) e implementado en un *PLC*.

ÍNDICE GENERAL

Abreviaturas	7
Glosario	8
1 Introducción.	11
1.1 Automatización de Procesos Industriales.	11
1.1.1 Controlador Lógico Programable.	12
1.2 Planteamiento del Problema.	12
1.3 Revisión de la Literatura.	13
1.3.1 Lenguajes de programación.	16
1.4 Panorama General de la Tesis.	17
1.5 Objetivos.	18
1.6 Organización de la Tesis.	18
2 Conceptos Básicos.	19
2.1 Introducción.	19
2.2 Redes de Petri Generalizadas.	20
2.2.1 Reglas de Evolución y Ecuación de Estado.	21
2.2.2 Propiedades de las <i>RP</i>	23
2.2.3 Clasificación de las <i>RP</i>	24
2.3 Redes de Petri Coloreadas.	24
2.3.1 Reglas de Evolución y Marcado.	25
2.4 Ejemplo.	26
2.5 Componentes de Sistemas de Manufactura Flexible.	30
2.6 Conclusiones.	32
3 Modelado de elementos físicos y operaciones.	33
3.1 Introducción.	33
3.2 Métodos de Modelado.	34
3.2.1 Métodos Ascendentes.	34
3.2.2 Métodos Descendentes.	37

3.2.3	Métodos Híbridos.	39
3.3	Módulos Elementales.	39
3.3.1	Módulo en <i>RPC</i> representando el Robot.	39
3.3.2	Módulo en <i>RPC</i> representando una Banda Transportadora.	41
3.3.3	Módulo en <i>RPC</i> representando Máquinas.	43
3.3.4	Módulo en <i>RPC</i> representando la operación de reconocimiento.	44
3.3.5	Módulo en <i>RPC</i> representando los sitios de almacenamiento.	44
3.3.6	Módulo en <i>RPC</i> representando un ensamble.	46
3.4	Funciones.	46
3.4.1	Funciones en los módulos.	49
3.5	Conclusiones.	50
4	Metodología de Modelado para SMF.	51
4.1	Introducción.	51
4.2	Modelado del sistema.	52
4.3	Modelado de la especificación.	63
4.4	Conclusiones.	67
5	Control por regulación	68
5.1	Introducción.	68
5.2	<i>RPC</i> equivalentes en trayectoria.	69
5.3	Caracterización de la controlabilidad.	74
5.3.1	Diseño del supervisor.	75
5.4	Conclusiones.	79
6	Conversión de <i>RPC</i> a <i>LD</i>.	80
6.1	<i>Introducción</i>	80
6.2	Metodología para convertir una <i>RPC</i> a un <i>LD</i>	81
6.2.1	Disparo de una transición.	81
6.3	Marcado inicial.	85
6.4	Ejemplos.	86
6.5	Conversión del controlador <i>H</i> en <i>LD</i>	97
6.6	Conclusiones.	106
7	Conclusiones y trabajo futuro.	107
7.1	Conclusiones.	107
7.2	Trabajo Futuro.	108

ÍNDICE DE FIGURAS

2.1	Sistema de los vagones.	27
2.2	Modelo en <i>RP</i> del sistema de los vagones.	28
2.3	Grafo de alcanzabilidad de la <i>RP</i> del sistema de los vagones.	29
2.4	Representación en <i>RPC</i> del sistema de los vagones.	30
3.1	Unión de módulos por fusión de lugares.	35
3.2	Unión de módulos por fusión de transiciones.	35
3.3	Unión de módulos por fusión de caminos que inician y terminan en lugares.	36
3.4	Unión de módulos por fusión de caminos que inician y terminan en transiciones.	36
3.5	Estructuras básicas de <i>RPC</i> con módulos. (a) Secuencia de operaciones. (b) Secuencia de operaciones con elección.	37
3.6	Refinamiento de transiciones en el método descendente.	38
3.7	Modelo en <i>RPC</i> de las tareas del robot.	40
3.8	(a) Modelo del comportamiento del brazo del robot con <i>RPC</i> . (b) Modelo del comportamiento de la pinza con <i>RPC</i>	41
3.9	(a) Operación permisiva. (b) Operación de sincronización.	42
3.10	Módulo en <i>RPC</i> representando al robot.	42
3.11	(a) Modelo de la banda transportadora con <i>RP</i> . (b) y (c) Módulo de la banda transportadora con <i>RPC</i>	43
3.12	(a) Reconocimiento con <i>RP</i> . (b) Módulo de reconocimiento con <i>RPC</i>	44
3.13	(a) Almacenamiento tipo <i>FILO</i> con <i>RP</i> . (b) Módulo de almacenamiento tipo <i>FILO</i> con <i>RPC</i>	45
3.14	Modelo de ensamble con <i>RP</i>	46
3.15	(a) Módulo de la operación de ensamble con <i>RPC</i> . (b) Módulo de la especificación de la operación de ensamble con <i>RPC</i>	47
4.1	Identificación de elementos físicos en el sistema de transporte de partes con robot y banda.	54

4.2	Módulos de las operaciones del sistema de transporte de partes.	56
4.3	Modelo final en <i>RPC</i> del sistema de transporte de partes.	56
4.4	Matriz de incidencia representación en funciones.	57
4.5	Representación numérica de la matriz de incidencia.	59
4.6	Identificación de elementos físicos del sistema de ensamble.	60
4.7	Módulos de operaciones y sitios del sistema de ensamble.	60
4.8	Modelo del <i>SMF</i> de ensamble con <i>RPC</i>	61
4.9	Modelo del <i>SMF</i> simplificado	62
4.10	Matriz de incidencia representada con funciones.	63
4.11	Matriz de incidencia representada en forma numérica.	64
4.12	Modelo de la especificación para el sistema de transporte de partes.	65
4.13	Modelo de la especificación del <i>SMF</i> de ensamble	66
5.1	Grafos de alcanzabilidad del modelo del sistema y especificación relacionados por la función π	70
5.2	Equivalencia en trayectoria de grafos de alcanzabilidad.	71
5.3	Esquema de control por regulación.	76
6.1	Representación del disparo de una transición respecto a un color de entrada y salida binario.	82
6.2	Representación del disparo de una transición respecto a colores no binarios.	83
6.3	Representación del disparo de una transición respecto a colores no binarios.	84
6.4	Representación del marcado inicial de una <i>RPC</i>	85
6.5	Matriz de incidencia de la especificación del sistema de transporte de partes.	86
6.6	Matriz construida para encontrar π	86
6.7	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de transporte de partes. Parte I.	89
6.8	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de transporte de partes. Parte II.	90
6.9	Matriz de incidencia para la especificación del sistema de ensamble.	91
6.10	Matriz construida para encontrar π	91
6.11	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de ensamble. Parte I.	93
6.12	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de ensamble. Parte II.	94

6.13	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de ensamble. Parte III.	95
6.14	Representación en <i>LD</i> de la especificación en <i>RPC</i> del sistema de ensamble. Parte IV.	96
6.15	<i>RPC</i> del sistema y el modelo en <i>RPC</i> de la especificación en lazo cerrado.	98
6.16	<i>LD</i> del sistema en lazo cerrado. Parte I.	99
6.17	<i>LD</i> del sistema en lazo cerrado. Parte II.	100
6.18	<i>LD</i> del sistema en lazo cerrado. Parte III.	101
6.19	<i>LD</i> del sistema en lazo cerrado. Parte IV.	102
6.20	<i>LD</i> del sistema en lazo cerrado. Parte V.	103
6.21	<i>LD</i> del sistema en lazo cerrado. Parte VI.	104
6.22	<i>LD</i> del sistema en lazo cerrado. Parte VII.	105

AGRADECIMIENTOS.

Primeramente a Dios quien me ha brindado una vida lo suficientemente larga para culminar con mis estudios.

Al CONACYT y al CINVESTAV por el apoyo económico y demás medios necesarios para el cumplimiento de mi investigación.

Al Dr. Antonio Ramírez Treviño por compartir conmigo su sabiduría y otorgarme de su tiempo para encaminarme en el cumplimiento de los objetivos de la tesis desarrollada.

A mis padres, Sahara Sánchez Jiménez y Luis Nuño García, por haberme inculcado excelentes valores, por haber sido unos buenos modelos a seguir, por velar que mi salud esté en buenas condiciones, por darme techo, comida y sustento, entre muchas otras cosas más, lo cual espero retribuir algún día.

A mi novia Lucía Karina Ramírez Valle por concederme cariño, amor y comprensión, por alentarme a seguir adelante y superarme, por cuidarme cuando ha sido necesario, por la paciencia que ha tenido a lo largo de mis estudios, entre otros.

A mis hermanos Luis, Noé y Benjamín quienes me han ayudado de diversas maneras, tanto en actividades recreativas como en incitarme al trabajo.

A mis amigos Riemann, Lizette, Oscar, Edgar, Iván, Samuel, Arnulfo, Jesús, Molus, Marina, Christian, Eduardo, Gerardo, Hebert, Ileana, Laura y el resto de las personas que me han dado su apoyo y amistad.

A todos ustedes, muchas gracias. . .

ABREVIATURAS.

<i>AMS</i>	Almacén de Materia Prima
<i>AE</i>	Área de Ensamble
<i>ANSA</i>	Organización Internacional de Estandarización
<i>CMF</i>	Celda de Manufactura Flexible
<i>CNC</i>	Controladas Numéricamente por Computadora
<i>FBD</i>	Diagrama Funcional de Bloques
<i>FIFO</i>	Primera Entrada Primera Salida
<i>FILO</i>	Primera Entrada Última Salida
<i>IEC</i>	Comisión Internacional de Técnicos Electricistas
<i>IL</i>	Lista de Instrucciones
<i>ISA</i>	Sociedad de Instrumentación de América
<i>LD</i>	Diagrama de Escalera
<i>PLC</i>	Controlador Lógico Programable
<i>RP</i>	Red de Petri
<i>RPC</i>	Red de Petri Coloreada
<i>RPG</i>	Red de Petri Generalizada
<i>SED</i>	Sistemas de Evento Discreto
<i>SFC</i>	Diagrama Funcional Secuencial
<i>SMF</i>	Sistema de Manufactura Flexible
<i>ST</i>	Texto Estructurado
<i>TCS</i>	Teoría del Control Supervisor

GLOSARIO DE SÍMBOLOS.

(a) Conjuntos

\forall	Para todo (Cuantificador universal)
\exists	Existe (Cuantificador existencial)
\in	Es un elemento de
\notin	No es un elemento de
$>$	Mayor que
\geq	Mayor o igual que
$<$	Menor que
\leq	Menor o igual que
\neq	Diferente de
$\{x \in A P(x)\}$	El conjunto de todos los $x \in A$ tal que satisfacen la condición $P(x)$
$A \cup B$	Unión de los conjuntos A y B
$A \cap B$	Intersección de los conjuntos A y B
$A \times B$	Producto Cartesiano de A y B
$ A $	Cardinalidad del conjunto A (número de elementos de A)
\emptyset	Conjunto Vacío
\mathbb{N}	Conjunto de todos los números naturales
\mathbb{Z}	Conjunto de todos los números enteros
\mathbb{Z}^+	Conjunto de todos los números enteros positivos
$f : X \rightarrow Y$	Es un mapeo de X a Y

(b) Redes de Petri

P	Conjunto de lugares de una RP
p, p_i	Lugar, i -ésimo lugar
T	Conjunto de Transiciones
t, t_j	Transición, j -ésima transición
$I(P, T)$	Función de entrada ($I : T \times P \rightarrow \mathbb{Z}^+$)
$O(P, T)$	Función de salida ($O : T \times P \rightarrow \mathbb{Z}^+$)

GLOSARIO DE SÍMBOLOS

$\bullet t$	Conjunto de lugares de entrada a t
t^\bullet	Conjunto de lugares de salida a t
$\bullet p$	Conjunto de transiciones de entrada a t
p^\bullet	Conjunto de transiciones de salida a t
M	Marcado de una RP
M_0	Marcado inicial de una RP
$M(p_i)$	Marcado del i -ésimo lugar
(N, M_0)	RP marcada
$R(N, M_0)$	Conjunto de alcanzabilidad de una RP
$\mathcal{L}(N, M_0)$	Lenguaje de disparos de una RP
$M_i \xrightarrow{t_j} M_j$	El marcado M_j es alcanzado al disparar la transición t_j
σ	Secuencia de disparos de transiciones ($\sigma = t_i t_j \dots t_q$)
$\vec{\sigma}$	Vector de Parikh de σ
$C(p_i)$	Conjunto de colores asociados al i -ésimo lugar
$C(t_j)$	Conjunto de colores asociados a la i -ésima transición
φ	Es una función de salida que asocia a cada marcado en $R(N, M_0)$ un vector de salida.
A	Matriz de incidencia de un RP
A'	Matriz de incidencia de la especificación
M'_0	Marcado inicial de la especificación
A^T	Matriz transpuesta de la matriz A
■	Fin de la demostración

(c) Funciones

I	Función identidad
Ens	Función de ensamble
$Assign$	Función de asignación de recursos

GLOSARIO DE SÍMBOLOS

(d) LD



Contacto normalmente abierto donde la energía fluye de izquierda a derecha cuando la variable asociada es 1.



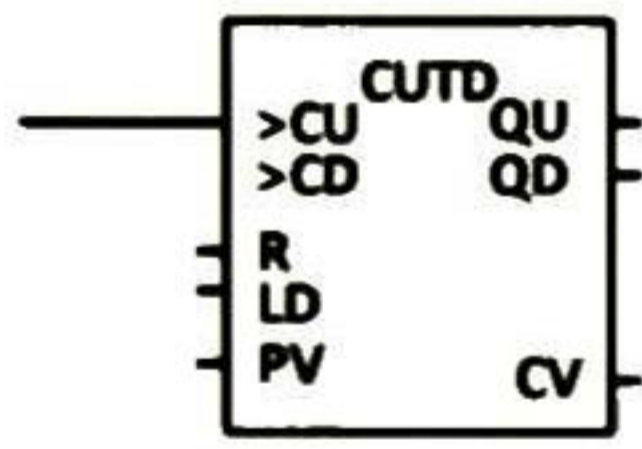
Contacto normalmente cerrado donde la energía fluye de izquierda a derecha cuando la variable asociada es 0.

—(S)

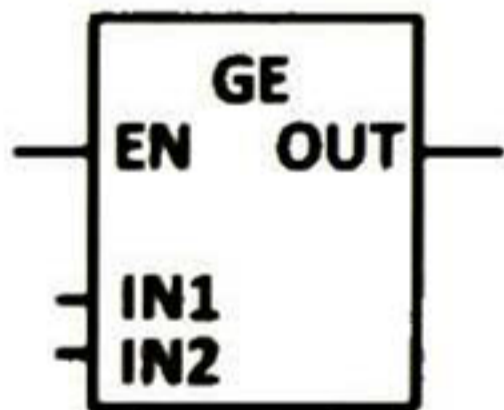
Esta bobina se fija en el estado de encendido cuando la energía provenga desde la izquierda. La bobina permanece en este estado hasta que se desactive.

—(R)

Esta bobina coloca el estado de apagado cuando la energía fluye desde la izquierda. La bobina permanece en este estado hasta que se active.



La operación contar adelante/atrás (*CTUD*) empieza a contar adelante cuando se produce un flanco positivo en la entrada de contaje hacia adelante (*CU*). Cuenta hacia atrás si se produce un flanco positivo en la entrada (*CD*). Si el valor actual (*CV*) es mayor o igual al valor de preselección (*PV*), se activa el bit de contaje (*Cxxx*). El contador se inicializa al activarse la entrada de desactivación (*R*).



La función secuencia monotamente decreciente (*GE*) compara $IN1 \geq IN2$ con el resultado booleano depositado en *OUT*. Los tipos de datos de entrada y salida pueden variar; pero deben ser del mismo tipo.

1. INTRODUCCIÓN

En el transcurso de los últimos años se han dado grandes cambios en diversos ámbitos, sobre todo en el económico, científico y tecnológico. En la actualidad las industrias, en la búsqueda de la ampliación de su mercado, o bien para continuar siendo competitivas en él, se han dado a la tarea de buscar la automatización de sus procesos productivos, la optimización de recursos, la implementación de sistemas estrictos para el control de calidad, la minimización de costos, entre otros. En lo que respecta a la automatización industrial uno de los retos más importantes es controlar la producción mientras que se mantiene una gran flexibilidad en los sistemas productivos, en otras palabras, deben de poseer la capacidad de producir una gran variedad de productos utilizando un tiempo de fabricación muy reducido.

1.1. Automatización de Procesos Industriales.

Un *SMF* es un sistema integrado por máquinas herramientas flexibles como las máquinas controladas numéricamente por computadora (*CNC*), robots, dispositivos automáticos de almacenamiento de materiales, un sistema inteligente de manejo de materiales automatizado (vehículos guiados automáticamente, bandas transportadoras, entre otros), así como de un sistema de control y de toma de decisiones dedicado a coordinar y monitorear a todo el resto de los componentes del *SMF*. A su vez, los *SMF* se pueden descomponer en unidades más pequeñas llamadas células robotizadas o celdas de manufactura (*CMF*) encargadas de obtener un producto o algún subproducto. Los robots utilizados en las *CMF* se caracterizan por la facilidad en su programación y que tienen varios grados de libertad, con lo cual son capaces de efectuar operaciones de gran complejidad. Para alcanzar el objetivo de producción de un *SMF*, un controlador que coordine a todos los componentes y que permita alcanzar los objetivos puestos en la especificación debe ser diseñado.

Tanto los *SMF* como las *CMF* son ejemplos de sistemas dinámicos de eventos discretos (*SED*), donde un *SED* es un sistema dinámico caracterizado por tener

CAPÍTULO 1. INTRODUCCIÓN

un espacio de estados numerables cuyo valor cambia abruptamente de manera asíncrona en respuesta a la ocurrencia de un evento.

Comúnmente el control de un *SED* se realiza por medio del encendido y apagado de interruptores de un controlador lógico programable (*PLC*). Una secuencia adecuada de encendido y apagado de interruptores permite poder confinar el *SMF* o cualquier *SED* a un comportamiento requerido para lograr los objetivos de producción.

1.1.1. Controlador Lógico Programable.

Los *PLC*'s son autómatas encargados de encender o apagar eventos a través de interruptores. Para que el *PLC* cumpla con su objetivo de controlar el sistema, es decir, confinarlo a un comportamiento requerido, es necesario programarlo para que sea capaz de secuenciar los procesos.

Los *PLC*'s se utilizan tanto para el control de procesos continuos como en *SED*. Comúnmente se eligen para el control de *SED* debido a su velocidad, capacidad de programación y de comunicación, aunado a su bajo costo y a la adaptabilidad a casi cualquier proceso de este tipo. Un problema que enfrenta la industria actualmente por el uso de los *PLC* es que por lo general no se cuenta con personal capacitado para programarlos y que garanticen la ausencia de errores. La mayoría de las veces los programadores al desempeñar su labor de forma empírica suelen introducir muchos errores de programación o comportamientos incorrectos debido a especificaciones ambiguas, lo que conlleva a que se tengan pérdidas económicas por los largos tiempos de demora en localizar los conflictos y poner de nuevo en marcha al sistema. Por ello, para evitar los problemas anteriores es preferible contar con el conocimiento de una metodología para modelar el sistema, diseñar y modelar la especificación, así como la traducción del modelo mediante un lenguaje aceptado por el *PLC* para su programación.

1.2. Planteamiento del Problema.

Por lo general, los métodos para controlar *SED* utilizan autómatas y *RP* básicamente. Debido a que todas las trayectorias que el sistema puede realizar son explícitamente descritas en los autómatas y al poco paralelismo que este formalismo permite, esta herramienta deja de ser aplicable en problemas reales de gran dimensión y cuyo comportamiento sea complejo como los *SMF*. Las *RP*, por otro lado, capturan información relevante como el asincronismo, paralelismo y

CAPÍTULO 1. INTRODUCCIÓN

conurrencia, pueden plasmarse en ellas las exclusiones mutuas, nos proporcionan información sobre el bloqueo de un sistema y qué condiciones lo ocasionan, es posible llevar a cabo un análisis matemático para corroborar que un estado es alcanzable. En el caso de los *SMF* nos permiten verificar si es posible obtener un producto dado y pueden llevarse a la simulación para observar y analizar toda la evolución de la red. No obstante las *RP*, cuando son aplicadas al modelado de *SMF*, todavía generan modelos de gran tamaño. Una herramienta que resulta adecuada para este caso son las *RP* de alto nivel, por ejemplo las redes de Petri coloreadas (*RPC*). Aunque el poder descriptivo de las *RP* de alto nivel es el mismo que el de las *RP*, la manera en que codifican la información con funciones (y no con números) hace que los modelos sean más compactos.

Este trabajo incursiona en el modelado de *SMF* con *RPC*, donde existe un compromiso entre el tamaño del modelo obtenido y el conjunto de funciones utilizadas durante el modelado. La idea es que se obtenga un modelo lo suficientemente compacto; pero que las funciones formen un ideal izquierdo, un campo o cualquier otra estructura matemática, para permitir una mayor eficiencia en el diseño de los algoritmos del control supervisor.

Actualmente los controladores supervisores se implementan en *PLC* y la programación de un *PLC* se realiza mediante programación jerárquica, distribuida y con cierto grado de polimorfismo. El control diseñado con sus respectivas funciones debe ser fácilmente codificable en un *PLC* a través de alguno de los lenguajes propuestos en el estándar *IEC1131 – 3*.

1.3. Revisión de la Literatura.

Diversas investigaciones han sido desarrolladas para mostrar la aplicación de las *RP* en el modelado de *SED* [23] y en el estudio de sus propiedades [20]. También se han escrito libros [31], [19], [8] con fines didácticos y pedagógicos para presentar de manera sencilla los conceptos básicos sobre *RP*, aplicaciones, análisis de *SED*, técnicas de modelado, propiedades, entre otros. Asimismo se han publicado libros enfocados a tipos especiales de redes como las de libre elección [7].

El modelo de *SED*, ya sea desarrollado de forma ascendente o descendente, debe capturar la información más relevante del sistema; pero también presentar buenas propiedades, motivo por el cual se han desarrollado diversas metodologías para cumplir con dichos objetivos, un ejemplo es el modelado modular propuesto en [28] el cual permite construir modelos para *SED* con buenas propiedades.

Algunos trabajos se han dedicado especialmente al modelado y síntesis de *RP*

CAPÍTULO 1. INTRODUCCIÓN

para *SMF*. En [33] se presentan las *RP* temporizadas con punto de decisión extendido o *RP* de decisión para controlar el flujo de material, recursos e información de un *SMF*. Este tipo de redes cuenta con puntos de decisión, representados por rectángulos vacíos, con los cuales se determina qué marca tiene prioridad sobre las demás, cuál transición habilitar cuando se da una libre elección o la prioridad en el marcado de lugares. Este tipo de redes permite la simulación; pero presentan una deficiencia para validar la especificación.

En [37] se propone una metodología de modelado descendente para *SMF* y en [34],[35] se aborda el tema de recursos compartidos y exclusiones mutuas secuenciales y paralelas, en tales trabajos se proponen condiciones para que las *RP* estén libres de bloqueos y presenten buenas propiedades como la vivacidad, acotamiento y ciclicidad.

Las *RPC* han permitido la construcción de modelos más compactos que las *RP* Generalizadas (*RPG*), lo cual ha traído consigo una mayor facilidad en procesar la información de los modelos en computadoras y desarrollar el control en tiempo real, además de poder realizar simulaciones. Otra investigación que también se ha enfocado al análisis de los invariantes para las *RPC* es [32].

Al modelar *SMF* con *RP* se generan representaciones de gran tamaño, por ello algunos investigadores han optado por utilizar las *RPC* con la finalidad de obtener modelos reducidos. En [27] se utilizan las *RP* de alto nivel para modelar modularmente las *CMF*. En [9] el *SMF* se modela en dos etapas, primero con *RP* se obtiene la representación de la arquitectura del sistema y del plan de proceso por separado, posteriormente ambos modelos se integran en un modelo único por medio de *RPC*, además la metodología propuesta evita los bloqueos en la red. Sin embargo, en los trabajos anteriores el problema de la controlabilidad no es tratado.

En el estudio de los *SED* no solamente interesa el modelado de los sistemas, sino también el controlar o el restringir el comportamiento del sistema a uno que sea requerido. Existen diversas estrategias para llevar a cabo el control de un *SED* donde se emplean autómatas, *RP*, entre otras herramientas formales.

En [24], [26] P.J. Ramadge y W. M. Wonham proponen la teoría del control supervisor (*TCS*) donde tanto el sistema como la especificación (comportamiento al cual se pretende confinar al sistema de acuerdo a la *TCS*) son descritos por lenguajes formales o por autómatas. En tales trabajos los eventos son clasificados en controlables (si para su ocurrencia requieren de una señal externa) y no controlables (si para la ocurrencia de un evento no requiere de una señal externa). Debido a la explosión de espacios de estados que un autómata puede generar,

CAPÍTULO 1. INTRODUCCIÓN

el determinar la controlabilidad de una especificación podría ser muy laborioso, motivo por el cual en [25] se introduce el control supervisor modular. En este enfoque la especificación global es presentada como una unión de módulos que el sistema debe satisfacer, donde para cada módulo se calcula un control supervisor. Sin embargo, el uso de autómatas como herramienta formal para modelar *SMF*, limita su uso a ejemplos pequeños o académicos.

Con el fin de obtener modelos más compactos que las representaciones por autómatas Guia trasladó la *TCS* a las *RP* en [11], en el cual sistema y especificación son modelados con *RP* acotadas. En [12] presenta una metodología para diseñar el control supervisor usando *RP*. También es considerado el caso cuando las *RP* no son acotadas [10]. En [13] se enuncian las condiciones suficientes y necesarias para que exista un supervisor en *RP* para el sublenguaje supremo controlable y en [14] lleva a cabo un análisis estructural para el control supervisor.

En [38], [36] se presentan controladores con *RP* aumentadas aplicadas a sistemas de manufactura, se le da este nombre cuando la *RP* que representa al código de control es ampliada para la recuperación de errores mientras que las propiedades y comportamiento de la especificación son preservadas. También en [21] se trata el tema del control y simulación de *SMF*.

En [29] se aborda el problema de control usando la técnica de regulación para *SED*, el cual trata de la construcción de un controlador que permita que la salida del sistema sea igual a la salida de la especificación. En este trabajo también se propone un esquema de regulación en el caso de la múltiple especificación.

En [8] se muestra la utilidad de los lugares de control (lugares que no pierden marcas al dispararse su transición de salida, debido a que su estado depende de un sistema de decisión externo) y de los circuitos de proceso y de control, para controlar *SMF*. Estas técnicas pueden ser de gran utilidad cuando un sistema tiene poca flexibilidad; pero cuando se tienen sistemas de gran tamaño y gran flexibilidad como los *SMF* se tienen que aplicar variaciones a las técnicas para poder llevar a cabo tareas que requieran múltiples secuencias y ensambles, aunado a que se generan modelos de gran tamaño y muchos caminos que podrían simplificar mediante el uso de *RPC*.

En [18] se presenta un lenguaje para describir sistemas concurrentes modelados por *RPC*; pero solamente se basa en el modelado del sistema de control y en la metodología propuesta es necesario hacer simplificaciones a la *RPC* obtenida.

En [3], [4], [5], [6] describe como diseñar, validar e implementar controladores lógicos para *SMF* cuando el comportamiento requerido por el sistema es representado con *RPC* ordenadas, sin embargo el problema de controlabilidad no está

incluido.

1.3.1. Lenguajes de programación.

Con la intención de mejorar las técnicas de programación de sistemas de control industriales la Comisión Internacional de Técnicos Electricistas (*IEC*) creó el estándar *IEC1131 – 3* [17] donde proporciona una selección de lenguajes para desarrollar el programa de control, una buena estructura para el diseño de programas ya sea de manera ascendente o descendente, el uso de encapsulamiento y ocultamiento de información, la formalización de programas reusables, entre otros temas. Los lenguajes descritos en el *IEC1131 – 3* son: Texto Estructurado, Diagrama Funcional de Bloques, Diagrama de Escalera, Lista de Instrucciones y Diagrama Secuencial Funcional.

- **Texto Estructurado (ST):** Es un lenguaje textual de alto nivel que permite desarrollar una programación estructurada, cuya sintaxis se asemeja mucho al *PASCAL*. El *ST* tiene un amplio rango de construcciones para asignar valores a variables de diferentes tipos de datos, llamar funciones y bloques funcionales, para crear expresiones (simples o complejas), para evaluación de enunciados condicionales y para iteraciones, es decir, selecciones de código repetido. El *ST* es particularmente útil para cálculos aritméticos o algorítmicos complejos.
- **Diagrama Funcional de Bloques (FBD):** Es un lenguaje gráfico para representar señales y flujos de datos a través de bloques funcionales es semejante a los diagramas de circuitos eléctricos. El *FBD* es un lenguaje utilizado en un amplio rango de problemas incluyendo la lógica booleana y el control de lazo cerrado. Cabe señalar que los bloques pueden ser reutilizados en diversas partes en la programación de un *PLC* y también dentro de otros lenguajes como *LD* o *SFC*.
- **Diagrama de Escalera (LD):** Es un lenguaje gráfico que está basado sobre la lógica de relevadores. El *LD* siempre cuenta con una línea vertical izquierda de energía la cual se encarga de suministrar la energía a través de los contactos ubicado a lo largo de los peldaños. Cada contacto representa un estado de una variable booleana y si el valor de todos los contactos en un escalón es verdadero la energía puede fluir a través del escalón y activar una bobina en el extremo derecho del peldaño. Los contactos pueden ser

CAPÍTULO 1. INTRODUCCIÓN

normalmente abiertos, si presentan un valor verdadero cuando el contacto se cierra, o normalmente cerrados, caso contrario de los anteriores. El *LD* es uno de los lenguajes comúnmente utilizados en la programación de los *PLC* actuales, orientado hacia aplicaciones de control discreto.

- **Lista de instrucciones (IL):** Es un lenguaje de bajo nivel, parecido al ensamblador, el cual se encuentra en un amplio rango de *PLC*'s. Está compuesto de una secuencia de instrucciones la cual opera una instrucción por línea. Una instrucción consiste de un operador seguido por uno o más operandos (el sujeto del operador). Es ideal para resolver problemas pequeños pocos puntos de decisión y donde haya un número limitado de cambios en la ejecución del flujo del programa.
- **Diagrama Funcional Secuencial (SFC):** Es un lenguaje gráfico para representar el comportamiento secuencial de un sistema de control. Un *SFC* está compuesto de cajas rectangulares (simbolizando una serie de pasos), arcos (enlazan pasos y transiciones) y transiciones (barras horizontales). Cada paso representa un estado particular del sistema a controlar. Cada transición tiene asociada una condición, cuando ésta es verdadera el paso anterior a la transición es desactivado y el posterior es activado. Las condiciones en las transiciones pueden ser dadas utilizando cualquiera de los lenguajes anteriormente descritos. El *SFC* generalmente se utiliza para definir secuencias de control que se manejan por eventos en el tiempo y cuando hay presencia paralelismo o toma de decisiones.

Generalmente en los trabajos del ramo los lenguajes más empleados para traducir controladores construidos con *RP*, o alguna de sus extensiones, son el *LD* y el *SFC*. En [16] se presenta una metodología para traducir automáticamente especificaciones modeladas con *RP* a *LD* para la programación de un *PLC*.

1.4. Panorama General de la Tesis.

En este trabajo se presenta la obtención del modelo del *SMF* usando *RPC*, capturando los eventos controlables e incontrolables. La metodología propuesta es ascendente y captura la distribución física de los elementos de un *SMF*, la especificación (o requerimientos) también es capturada como una máquina de estados, tal y como se propone en las recetas establecidas por el estándar *ANSA/ISA-S88.01*

[1], [22]. La metodología propuesta utiliza un conjunto de funciones y colores limitado de forma que pueda modelar los *SMF* y que facilite el diseño de un controlador para un *SMF*. Puesto que la especificación no captura todos los eventos del *SMF*, entonces se propone empotrar el comportamiento de la especificación dentro del modelo del *SMF* usando el concepto de equivalencia en trayectoria [30], por lo que los estados de la especificación se hacen comparables con los estados modelo del *SMF* utilizando una función y cualquier evento de la especificación se simula por una secuencia de eventos del modelo del *SMF*; tomando en cuenta que las secuencias sean controlables. Finalmente, se realiza la traducción del controlador del *SMF* a *LD*.

1.5. Objetivos.

Los objetivos de esta tesis son:

Proponer una metodología de modelado para modelar *SMF* por medio de *RPC*. Los modelos obtenidos deben ser compactos y sus funciones deben permitir diseñar el control por regulación de manera sencilla.

Traducir los controladores obtenidos a *PLC* en base a alguno de los lenguajes propuestos en el estándar *IEC1131 – 3*, en este caso el seleccionado fue *LD*.

1.6. Organización de la Tesis.

Esta tesis se organiza de la siguiente manera. El Capítulo 2 presenta brevemente los conceptos básicos que se necesitan en este trabajo. En el Capítulo 3 se describen los módulos elementales empleados en la metodología de modelado, así como el conjunto de funciones. En el Capítulo 4 se expone la metodología de modelado para el sistema y la especificación, también la metodología de modelado es aplicada a través de dos ejemplos. El Capítulo 5 muestra el diseño del supervisor y como es aplicado para confinar el *SMF* dentro del comportamiento de la especificación. Posteriormente en el capítulo 6 presenta una metodología para traducir el controlador computado en el capítulo anterior a *LD* para implementarse en *PLC*. Además se muestran dos ejemplo donde se aplica el algoritmo para computar π y \vec{w} , asimismo se presenta la traducción de la especificación de estos ejemplos a *LD*. Finalmente, las conclusiones de esta tesis son presentadas en el capítulo 7.

2. CONCEPTOS BÁSICOS.

Resumen: En este capítulo se presentan brevemente los conceptos básicos de *RPC* estableciendo la nomenclatura utilizada a través del presente trabajo. La definición de *RPC* es antecedida por una presentación de las *RPG*, algunas propiedades relevantes de las *RP* y su clasificación. Posteriormente se muestra un ejemplo ilustrativo donde se modela con *RP* y *RPC*. Finalmente se incluyen algunos conceptos asociados a los *SMF*.

2.1. Introducción.

Los formalismos utilizados con mayor frecuencia para modelar *SED's* son los autómatas y las *RP*. Los autómatas son de gran utilidad para obtener representaciones gráficas sencillas, cuando el sistema no presenta paralelismo y exhibe pocos estados. Las *RP* son una herramienta apropiada para modelar los *SED's* cuando es necesario capturar el paralelismo, concurrencia, sincronización, exclusiones mutuas, entre otras características de interés, aunado a que es posible desarrollar un análisis matemático para el estudio de las propiedades de los *SED's*. Sin embargo, las *RP* al ser aplicadas para modelar a los *SMF* reales, todavía producen modelos muy grandes, motivo por el cual se utilizan las *RPC* para obtener representaciones más compactas.

Este capítulo se organiza de la siguiente forma: En la sección 2.2 se introducen formalmente a las redes de Petri generalizadas (*RPG*), así como propiedades básicas de las *RP*. Además se presentan las clases de *RP* más importantes: máquina de estados, grafos marcados y de libre elección. La sección 2.3 presenta la definición de las *RPC*, la herramienta usada para el desarrollo de esta tesis. Posteriormente en la sección 2.4 se muestra un ejemplo ilustrativo de modelado con *RP* y *RPC*, así como el análisis de sus propiedades. Después en la sección 2.5 se introducen algunos conceptos básicos, relacionados con los componentes de los *SMF*, para un mejor entendimiento de los próximos capítulos. Finalmente en la sección 2.6 se presentan las conclusiones.

2.2. Redes de Petri Generalizadas.

Las RP son una herramienta formal para el modelado y análisis de SED , ya que permiten analizar la dinámica del SED por medio de la evolución de marcas dentro de la red. El modelar con RP puede ser complicado cuando no se sigue una metodología, en particular cuando el objetivo es modelar un SMF (por su gran tamaño y complejidad). Para enfrentar este tipo de problemas una estrategia es dividir el sistema en partes y obtener un modelo en RP para cada subsistema. Aunque este procedimiento ayuda a disminuir el número de errores en el modelado, no evita que el modelo final sea de gran tamaño.

A continuación se enuncia la definición formal de una RPG .

Definición 1. Una RPG es un grafo bipartito dirigido, definido por la quintupla $RPG = (P, T, I, O, M_0)$, donde:

$P = \{p_1, p_2, \dots, p_n\}$ es un conjunto finito de n elementos llamados lugares.

$T = \{t_1, t_2, \dots, t_m\}$ es un conjunto finito de m elementos llamados transiciones.

$P \cup T \neq \emptyset$ y $P \cap T = \emptyset$

$I : T \times P \rightarrow \mathbb{Z}^+$, es una función de entrada que define los arcos que van dirigidos de P a T .

$O : T \times P \rightarrow \mathbb{Z}^+$. es una función de salida que define los arcos que van dirigidos de T a P .

$M_0 : P \rightarrow \mathbb{Z}^+$, es una función que define el número de marcas en P .

\mathbb{Z}^+ es el conjunto de los enteros positivos.

A la distribución de marcas dentro de los lugares de la red se le denomina marcado y éste representa el estado de la RPG . Una RPG con un marcado inicial específico se denota por (N, M_0) .

El marcado se representa por un vector de dimensión n , donde el elemento i representa el número de marcas en el lugar i . El marcado en p_i se expresa como $M(p_i)$.

Las funciones $I(P, T)$ y $O(P, T)$ son matrices de dimensión $n \times m$ con elementos $I(p_i, t_j)$ y $O(p_i, t_j)$ arcos ponderados. Una RPG puede representarse asimismo por su matriz de incidencia $A = [A_{ij}]_{n \times m}$, donde:

$$A_{ij} = O(p_i, t_j) - I(p_i, t_j) \quad (2.1)$$

Definición 2. Sean $x, y \in T \cup P$, entonces $\bullet(x) = \{y \in T \cup P \mid \text{existe un arco de } y \text{ a } x\}$; $(x)^\bullet = \{y \in T \cup P \mid \text{existe un arco de } x \text{ a } y\}$ son el conjunto de predecesores y sucesores del nodo x , respectivamente.

CAPÍTULO 2. CONCEPTOS BÁSICOS

2.2.1. Reglas de Evolución y Ecuación de Estado.

El marcado de una *RPG* evoluciona de acuerdo con las siguientes reglas:

Una transición $t_j \in T$ está habilitada en el marcado M_k si y sólo si $\forall p_i \in P$, $M_k(p_i) \geq I(p_i, t_j)$.

Si t_j está habilitada en un marcado M_k entonces puede ser disparada. En caso de que esto suceda se alcanza un nuevo marcado M_{k+1} . El marcado M_{k+1} se puede calcular usando la ecuación de estado:

$$M_{k+1} = M_k + Av_k \quad (2.2)$$

donde v_k es el vector ($m \times 1$) de disparo de la transición habilitada t_j , donde el j -ésimo elemento es 1 y el resto son cero.

Realizando varios disparos, todos ellos legales (si y sólo si las transiciones están habilitadas), se tiene:

$$\begin{aligned} M_1 &= M_0 + Av_0 \\ M_2 &= M_1 + Av_1 = M_0 + Av_0 + Av_1 = M_0 + A(v_0 + v_1) \\ &\vdots \\ M_{k+1} &= M_k + Av_k. \end{aligned}$$

Así, combinando las ecuaciones anteriores se obtiene que:

$$M_{k+1} = M_0 + Av \quad (2.3)$$

Donde v es la suma de todos los $v_i, i \in [0, \dots, k]$.

Si bajo un v se llega a un M_{k+1} con valores negativos, entonces los disparos no son legales.

La notación $M_k \xrightarrow{t_j} M_{k+1}$ representa el hecho de que el marcado M_{k+1} es alcanzado al disparar transición habilitada t_j desde el marcado M_k .

Definición 3. Una secuencia de disparos de una *RPG* (N, M_0) es una secuencia $\sigma = t_i t_j \dots t_k$ tal que $M_j \xrightarrow{t_i} M_i \xrightarrow{t_j} \dots \xrightarrow{t_k} M_k$. El hecho de alcanzar M_k desde M_j por el disparo de la secuencia σ es denotado por $M_j \xrightarrow{\sigma} M_k$.

CAPÍTULO 2. CONCEPTOS BÁSICOS

Definición 4. Sea (N, M_0) una RPG. El conjunto de todas las secuencias de disparo de (N, M_0) $\mathcal{L}(N, M_0) = \{\sigma = t_i t_j \dots t_k | M_j \xrightarrow{t_i} M_i \xrightarrow{t_j} \dots \xrightarrow{t_k} M_k\}$ es denominado el lenguaje de disparos de (N, M_0) .

Definición 5. Una secuencia de disparos $\sigma \in \mathcal{L}(N, M_j)$ donde $M_j \xrightarrow{\sigma} M_k$ es llamada secuencia de bloqueo si M_k es un marcado bloqueante, es decir, ninguna transición puede habilitarse en M_k .

Definición 6. El conjunto $R(N, M_0) = \{M_k | M_0 \xrightarrow{\sigma} M_k\}$ de (N, M_0) es nombrado conjunto de alcanzabilidad de una RPG (N, M_0) .

Un marcado se dice alcanzable si está dentro del conjunto $R(N, M_0)$.

Definición 7. Un P – invariante X es un vector de enteros no negativos que satisface la ecuación $X^T A = 0$.

Definición 8. El soporte de un P – invariante X_i es el conjunto

$$\langle X_i \rangle = \{p_j | X_i(p) \neq 0\}.$$

Si multiplicamos ambos lados de la ecuación de estados (2.3) por un P – invariante se tiene que:

$$X^T M_{k+1} = X^T M_0 \tag{2.4}$$

lo cual significa que bajo algún vector de ponderación X^T , el número de marcas en la red se ha conservado.

Definición 9. Un T – invariante Y es un vector de enteros no negativos que cumple con la ecuación $AY = 0$.

Definición 10. El soporte de un T – invariante Y_i es el conjunto

$$\langle Y_i \rangle = \{t_j | Y_i(p) \neq 0\}$$

Ahora bien, si en la ecuación de estados (2.3) $v = Y$ resulta que:

$$M_{k+1} = M_0 \tag{2.5}$$

indica que la red regresa al estado inicial después de una secuencia de disparos dados por Y , si y sólo si los disparos son legales.

CAPÍTULO 2. CONCEPTOS BÁSICOS

2.2.2. Propiedades de las RP .

Conflicto. Si para un M_j existen varias transiciones habilitadas $\{t_1, \dots, t_q\}$, y el disparo de una de ellas ocasiona que el resto se deshabilite, entonces se dice que las transiciones $\{t_1, \dots, t_q\}$ de la RP están en conflicto. A $\{t_1, \dots, t_q\}$ se le denomina conjunto de conflicto.

Ausencia de bloqueos. Una RP presenta ausencia de bloqueos si $\forall M_j \in R(N, M_0)$ existe al menos una transición habilitada.

Acotamiento. Una RP es acotada si no existen sobreflujos, es decir, si para cualquier M_0 finito entonces $M_k(p)$ es finito. Obviamente, una RP es acotada si y sólo si $R(N, M_0)$ es finito.

Definición 11. Un lugar p_i de una RP se dice b -acotado, si $\forall M_k \in R(N, M_0)$, $M_k(p_i) \leq b$, donde $b \in \mathbb{Z}^+$. Una RP es b -acotada, si todos sus lugares son b -acotados.

Seguridad. Una RP es segura o binaria si es 1-acotada.

Definición 12. Una $RP (N, M_0)$ es estrictamente conservativa si y sólo si $\forall M_k \in R(N, M_0)$

$$\sum_{i=1}^n M_k(p_i) = \sum_{i=1}^n M_0(p_i) \quad (2.6)$$

Definición 13. Una RP con un marcado inicial M_0 es conservativa si existe un vector X que lleve a la RP al marcado M_{K+1} y un $X \neq 0$ tal que:

$$X^T M_{k+1} = X^T M_0 \quad (2.7)$$

Consistencia. Una RP es consistente si $\exists \sigma$ tal que $M_k \xrightarrow{\sigma} M_0$, y todas las transiciones deben dispararse al menos una vez.

Definición 14. Una transición $t_k \in T$ es viva para un marcado M_0 , si $\forall M_k \in R(N, M_0)$, $\exists \sigma$ tal que $M_k \xrightarrow{\sigma} M_n$ y $t_k \in \sigma$. Una RP es viva si todas sus transiciones son vivas.

Ciclicidad. Una RP es cíclica si $\forall M_k \in R(N, M_0)$, $\exists \sigma$ tal que $M_k \xrightarrow{\sigma} M_0$.

Repetitividad. Una RP es repetitiva a partir de un marcado M_k si existe un σ tal que $M_k \xrightarrow{\sigma} M_k$ y el vector de Parikh de σ , $\vec{\sigma}$, construido a partir de σ no debe contener elementos nulos. Si el vector de disparo $\vec{\sigma}$ contiene elementos nulos se dice ser parcialmente repetitiva.

CAPÍTULO 2. CONCEPTOS BÁSICOS

2.2.3. Clasificación de las RP .

Con el fin de estudiar las RP éstas se pueden agrupar de acuerdo a su estructura en las siguientes clases:

- **Grafos marcados.** Un grafo marcado es una RP donde cada lugar tiene solamente una transición de entrada y una transición de salida, es decir, $\forall p_i \in P, |\bullet p_i| = |p_i \bullet| = 1$.
- **Máquina de estado.** Una máquina de estado es una RP donde cada transición tiene sólo un lugar de entrada y un lugar de salida, esto es, $\forall t_i \in T, |\bullet t_i| = |t_i \bullet| = 1$.
- **Redes de libre elección.** Una RP de libre elección es aquella en la que si dos transiciones t_i y t_j comparten el mismo lugar de entrada p_i , entonces sus conjuntos de lugares de entrada (predecesor) tiene que ser igual. En otras palabras $\forall p_i \in P$, si $|p_i \bullet| > 1$, entonces $\forall t_i, t_j \in (p_i)^\bullet, \bullet(t_i) = \bullet(t_j)$.
- **Otras redes.** Aquellas que no están incluidas en las definiciones anteriores.

2.3. Redes de Petri Coloreadas.

Cuando se representa un SMF mediante las RPG por su complejidad y dimensión resulta en modelos con un gran número de lugares y transiciones. Una alternativa para obtener modelos más compactos es usando las RPC , propuestas por K. Jensen en 1981. Gracias al nivel de abstracción de las RPC los modelos obtenidos resultan bastante condensados. En las RPC las marcas contenidas en los lugares pueden distinguirse entre sí, puesto que poseen una identidad simbólica; estas marcas reciben el nombre de colores, por ello se le otorga este nombre a esta extensión de las RP . En las RPC las transiciones se pueden habilitar con un conjunto de colores específico. Además, los arcos relacionan los colores de las transiciones con los de los lugares y viceversa; relación que está dada por funciones.

A continuación se enuncia la definición formal de una RPC .

Definición 15. Una RCP es una séxtupla $RPC = (P, T, C, I, O, M_0)$ donde: P y T se definen como en una RPG .

$C = C(p_i) \cup C(t_j)$, $C(p_i)$ y $C(t_j)$ son un conjunto de elementos llamados colores, asociados a cada $p_i \in P$ y $t_j \in T$ respectivamente, que es

$$C(p_i) = \{a_{i1}, a_{i2}, \dots, a_{iu_i}\}, u_i = C|(p_i)|, C(t_j) = \{b_{j1}, b_{j2}, \dots, b_{jv_j}\}, v_j = C|(t_j)|.$$

CAPÍTULO 2. CONCEPTOS BÁSICOS

$I : C(T) \times C(P) \rightarrow \mathbb{N}$, $O : C(T) \times C(P) \rightarrow \mathbb{N}$ son las funciones de entrada y salida, respectivamente. Ambas funciones son representadas por matrices $I = [I(p_i, t_j)]$, $O = [O(p_i, t_j)]$ donde:

$$I(p_i, t_j) = [I(p_i, t_j)_{(b_{j1})} \cdots I(p_i, t_j)_{(b_{jk})} \cdots I(p_i, t_j)_{(b_{jv_j})}]$$

$$e I(p_i, t_j)_{(b_{jk})} = \sum_{h=1}^{u_i} \gamma_{ih} a_{ih}.$$

y

$$O(p_i, t_j) = [O(p_i, t_j)_{(b_{j1})} \cdots O(p_i, t_j)_{(b_{jk})} \cdots O(p_i, t_j)_{(b_{jv_j})}]$$

$$y O(p_i, t_j)_{(b_{jk})} = \sum_{h=1}^{u_i} \gamma_{ih} a_{ih}.$$

M_0 es el mercado inicial de la RPC. Es una función definida sobre P tal que $M(p_i) = \sum_{h=1}^{u_i} \alpha_{ih} a_{ih}$, $\forall p_i \in P$ donde α_{ih} es el número de marcas coloreadas a_{ih} en p_i .

A la distribución de marcas coloreadas dentro de los lugares de la RPC se le denomina **mercado** y éste representa el estado de la RPC.

2.3.1. Reglas de Evolución y Mercado.

La regla de disparo de transiciones para una RPC se define de la siguiente manera:

Definición 16. Una transición t_j está habilitada con respecto a un color b_{jk} si y sólo si $\forall p_i \in P$, $M(p_i) \geq I(p_i, t_j)_{(b_{jk})}$; t_j puede ser disparada si y sólo si está habilitada.

El disparo de la transición habilitada t_j desde un mercado M con respecto a un color b_{jk} origina un nuevo mercado M' definido por:

$$\forall p_i \in P, M'(p) = M(p) + O(p_i, t_j)_{(b_{jk})} - I(p_i, t_j)_{(b_{jk})} \quad (2.8)$$

Definición 17. La matriz de incidencia A se define como:

CAPÍTULO 2. CONCEPTOS BÁSICOS

$A = O(p_i, t_j)_{(b_{jk})} - I(p_i, t_j)_{(b_{jk})}$
Entonces A es una función de $C(t) \times C(p) \rightarrow \mathbb{Z}$.
La ecuación (2.8) puede reescribirse como:

$$\forall p_i \in P, M'(p) = M(p) + Au \quad (2.9)$$

donde u es el vector de disparos que selecciona una columna de A , en otras palabras elige un color de la transición t_j que se dispara.

Si en una RPC se realiza una secuencia de disparos v , en este caso una secuencia de colores, desde M_0 , el marcado final M_{k+1} se calcula con la siguiente ecuación:

$$M_{k+1} = M_0 + Av \quad (2.10)$$

Las RPC presentan las mismas propiedades antes señaladas para las RP .

2.4. Ejemplo.

Un sistema está integrado por dos vagones V_1 y V_2 , una vía para cada vagón, un botón M y cuatro sensores indicadores de posición (a, b, c, d) . Este sistema es mostrado en la figura 2.1.

Los vagones V_1 y V_2 se desplazan sobre vías independientes por acción de las señales D_i (movimiento hacia la derecha) e I_i (movimiento hacia la izquierda), donde $i = 1, 2$. Las posiciones extremas son detectadas por los sensores a, b para la vía 1 y c, d para la vía 2. Un botón M sirve para iniciar la operación del sistema desde su posición inicial que es cuando ambos vagones se localizan en el extremo izquierdo.

Funcionamiento del sistema: cuando el botón M es oprimido ambos vagones parten hacia la derecha; al llegar éstos a su extremo derecho deben regresarse y detenerse al alcanzar su posición inicial. Solo se puede iniciar un nuevo ciclo cuando los dos vagones estén detenidos.

El modelo en RP para este sistema se exhibe en la figura 2.2.

A la red se le puede asociar la siguiente interpretación:

Para los lugares:

p_1 : Vagón 1 en reposo.

p_2 : Vagón 2 en reposo.

p_3 : Vagón 1 en avanzando hacia la derecha.

CAPÍTULO 2. CONCEPTOS BÁSICOS

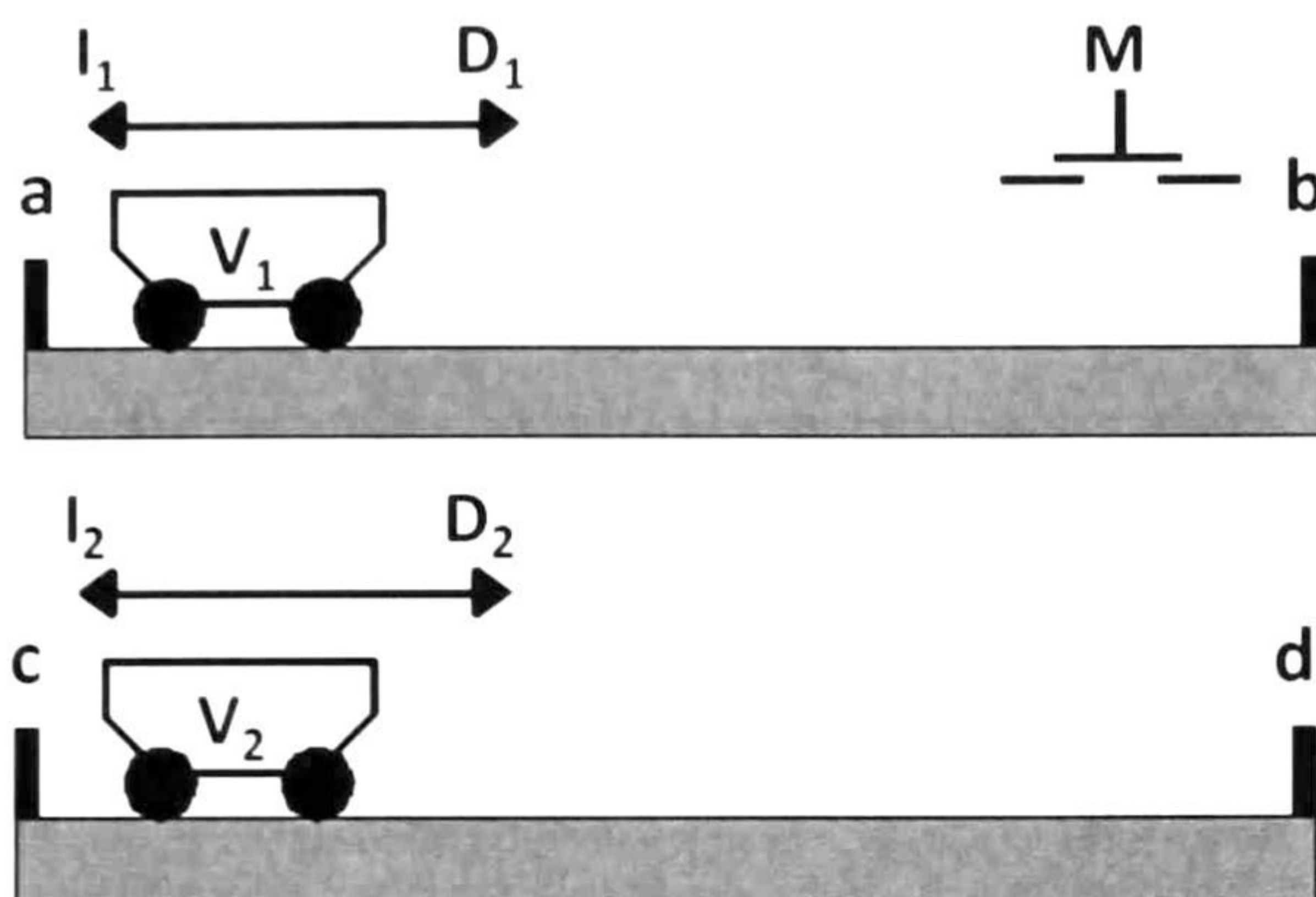


Figura 2.1: Sistema de los vagones.

p_4 : Vagón 2 en avanzando hacia la derecha.

p_5 : Vagón 1 en desplazándose hacia la izquierda.

P_6 : Vagón 2 en desplazándose hacia la izquierda.

Para las transiciones:

t_1 : Se oprime el botón M.

t_2 : El vagón 1 llega al final de su recorrido y es detectado por el sensor b .

t_3 : El vagón 2 llega al final de su recorrido y es detectado por el sensor d .

t_4 : El vagón 1 regresa a su posición inicial y es detectado por el sensor a .

t_5 : El vagón 2 regresa a su posición inicial y es detectado por el sensor c .

La matriz de incidencia de esta red es:

$$A = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{pmatrix}$$

Donde $M_0 = (1 \ 1 \ 0 \ 0 \ 0 \ 0)^T$

El conjunto de predecesores y sucesores $\forall x, y \in T \cup P$ es:

CAPÍTULO 2. CONCEPTOS BÁSICOS

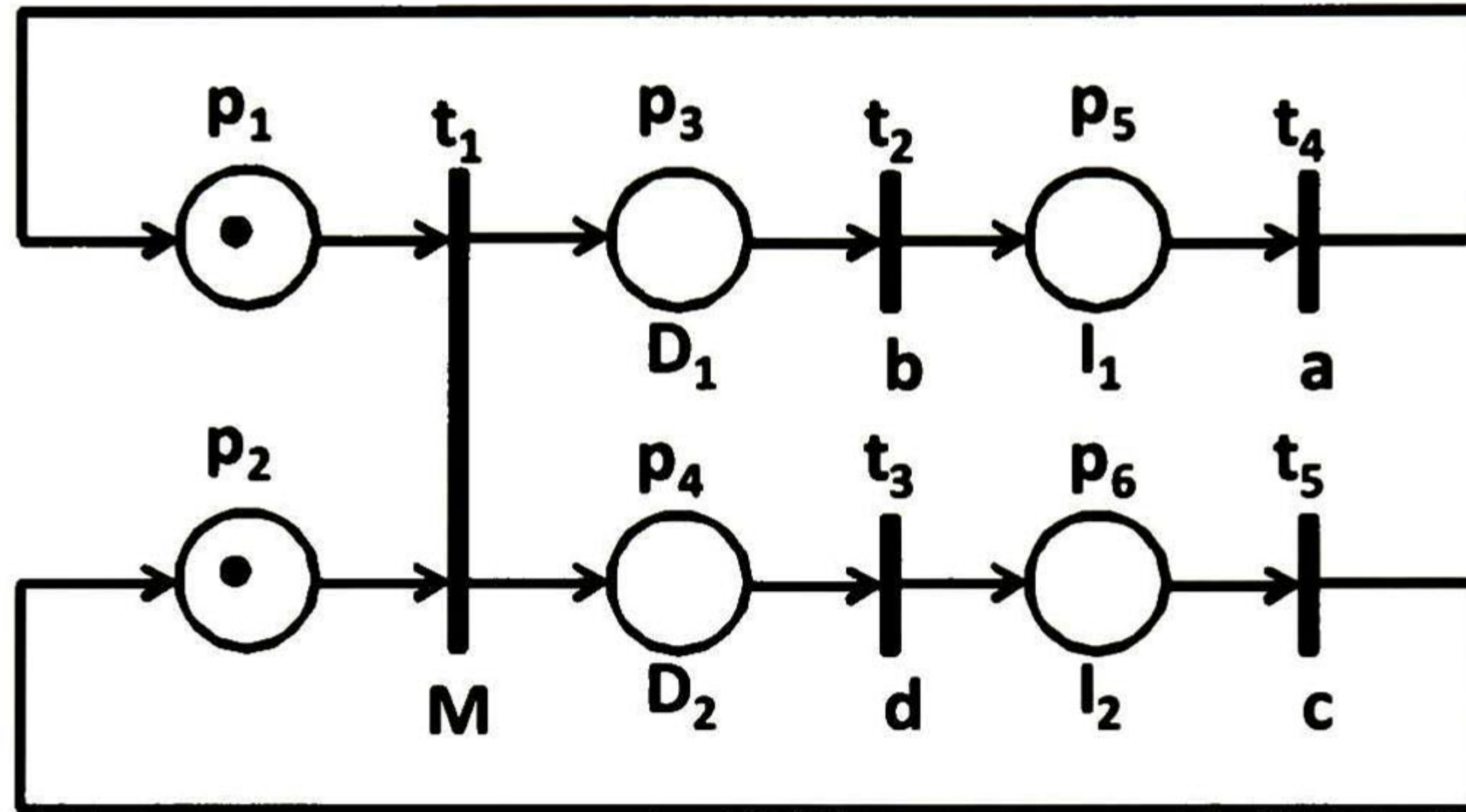


Figura 2.2: Modelo en RP del sistema de los vagones.

<i>Predecesores</i>	<i>Sucesores</i>
$\bullet(p_1) = \{t_4\}$	$(p_1)^\bullet = \{t_1\}$
$\bullet(p_2) = \{t_5\}$	$(p_2)^\bullet = \{t_1\}$
$\bullet(p_3) = \{t_1\}$	$(p_3)^\bullet = \{t_2\}$
$\bullet(p_4) = \{t_1\}$	$(p_4)^\bullet = \{t_3\}$
$\bullet(p_5) = \{t_2\}$	$(p_5)^\bullet = \{t_4\}$
$\bullet(p_6) = \{t_3\}$	$(p_6)^\bullet = \{t_5\}$
$\bullet(t_1) = \{p_1, p_2\}$	$(t_1)^\bullet = \{p_3, p_4\}$
$\bullet(t_2) = \{p_3\}$	$(t_2)^\bullet = \{p_5\}$
$\bullet(t_3) = \{p_4\}$	$(t_3)^\bullet = \{p_6\}$
$\bullet(t_4) = \{p_5\}$	$(t_4)^\bullet = \{p_1\}$
$\bullet(t_5) = \{p_6\}$	$(t_5)^\bullet = \{p_2\}$

Los P y T-invariantes de la RP son:

<i>P-invariantes</i>	<i>T-invariantes</i>
$X_1 = (1 \ 0 \ 1 \ 0 \ 1 \ 0)$	$Y_1 = (1 \ 1 \ 1 \ 1 \ 1)^T$
$X_2 = (0 \ 1 \ 0 \ 1 \ 0 \ 1)$	

Los soportes de los P y T-invariantes son:

$$\langle X_1 \rangle = \{p_1, p_3, p_5\}$$

$$\langle X_2 \rangle = \{p_2, p_4, p_6\}$$

CAPÍTULO 2. CONCEPTOS BÁSICOS

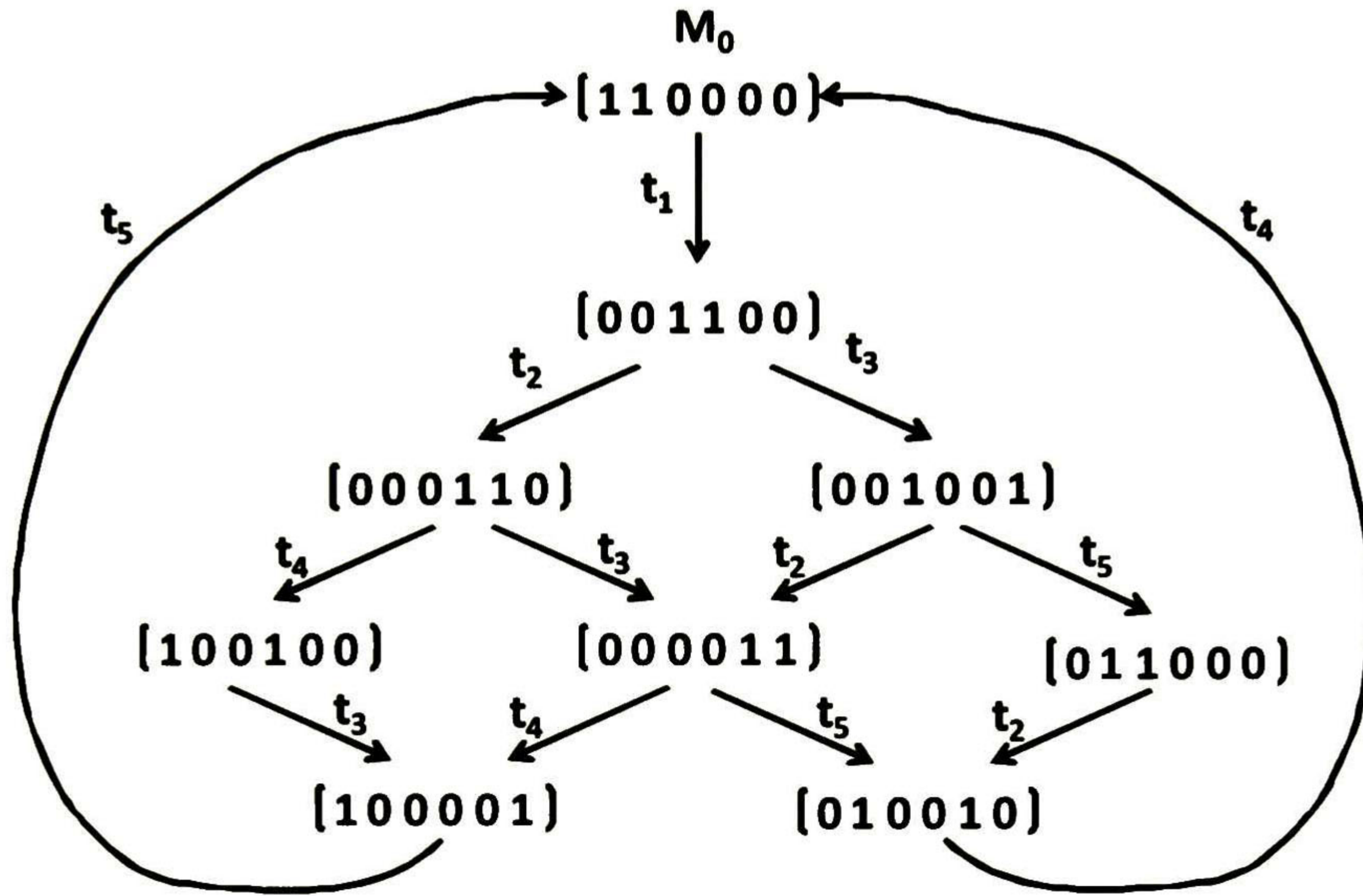


Figura 2.3: Grafo de alcanzabilidad de la RP del sistema de los vagones.

$$\langle Y_1 \rangle = \{t_1, t_2, t_3, t_4, t_5\}$$

Esta RP presenta ausencia de bloqueos, no tiene conflictos, es binaria, es viva, es cíclica, es conservativa y consistente.

Tanto la ciclicidad como el acotamiento se pueden apreciar claramente en su grafo de alcanzabilidad mostrado en la figura 2.3.

Debido a que cumple con $\forall p_i \in P, |\bullet p_i| = |p_i^\bullet| = 1$, entonces es un grafo marcado.

Como se puede apreciar la RP anterior está constituida básicamente por dos circuitos iguales por ello la representación del sistema en RPC es la siguiente:

Se debe definir el conjunto de colores para cada lugar y transición, que en este caso es:

$$C(p_1) = \{\langle V_1 \rangle, \langle V_2 \rangle\} = C(p_2) = C(p_3)$$

$$C(T_1) = \{\langle V_1, V_2 \rangle\}$$

$$C(T_2) = C(T_3) = C(P_1)$$

Claramente se puede observar que T_1 requiere de los colores V_1 y V_2 para habilitarse, en cambio T_2 y T_3 necesitan de V_1 o V_2 . De acuerdo al marcado inicial

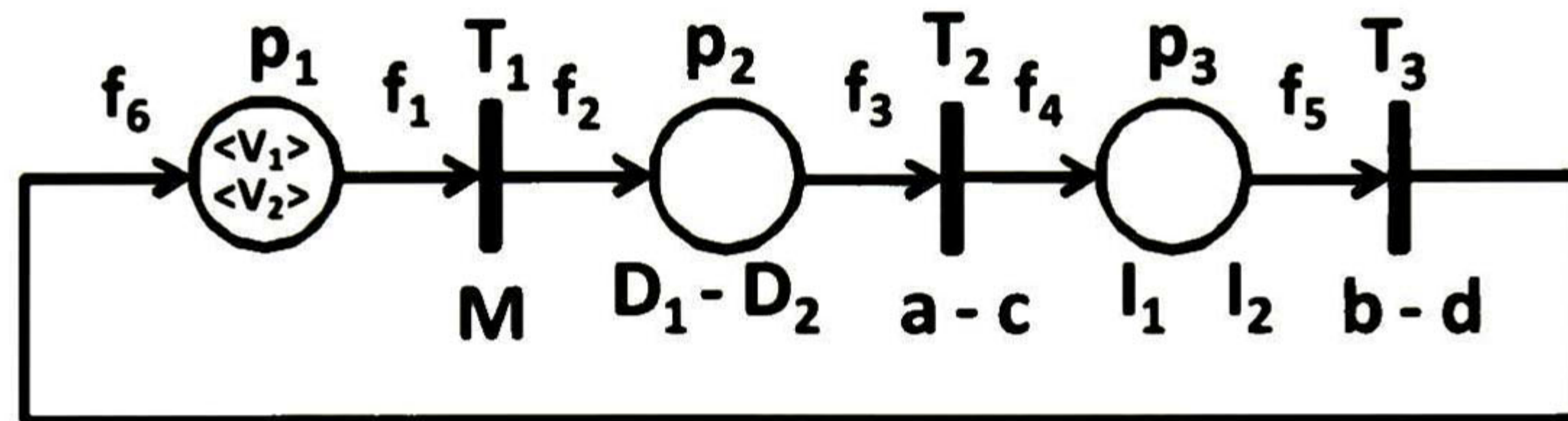


Figura 2.4: Representación en *RPC* del sistema de los vagones.

de la *RPC* ambos vagones se localizan en p_1 .

En este caso la interpretación es la siguiente:

Para los lugares coloreados:

p_1 : Vagón 1 o 2 en reposo, esto conforme al colores que estén dentro del lugar.

p_2 : Vagón 1o 2 avanzando hacia la derecha, de acuerdo a los colores que estén dentro del lugar.

p_3 : Vagón 1o 2 avanzando hacia la izquierda, de acuerdo a los colores que estén dentro del lugar.

Para las transiciones coloreadas:

T_1 : Se oprime el botón M.

T_2 : El vagón 1 (2) llega al final de su recorrido y es detectado por el sensor b (d).

T_3 : El vagón 1 (2) regresa a su posición inicial y es detectado por el sensor a (c).

Las funciones se definirán posteriormente debido a que las usadas en este trabajo difieren de las que son empleadas por otros autores.

Cabe señalar que con las *RPC* es posible reducir aún más el modelo, en este caso llegando a tener solamente un lugar y una transición, sin embargo debe existir un compromiso entre el nivel de abstracción que se quiere obtener y la claridad de representación del sistema, en el caso de esta tesis es la distribución física o layout del *SMF*, pues se puede perder visualmente lo que representaba el modelo. De ahí que en la metodología propuesta se definirá hasta que nivel es conveniente hacer la abstracción.

2.5. Componentes de Sistemas de Manufactura Flexible.

Un *SMF* está compuesto básicamente de los siguientes elementos físicos:

CAPÍTULO 2. CONCEPTOS BÁSICOS

- **Efectores.** Son todos los elementos capaces de transportar, temporalmente sostener, tomar o colocar alguna parte en un sitio específico, por ejemplo los robots, bandas transportadoras, vehículos autoguiados, líneas de transferencia, entre otros.
- **Sitios de Almacenamiento.** Son los lugares donde las partes pueden ser depositadas, temporal o permanentemente, sin recibir ningún tratamiento adicional. Las partes pueden ser almacenadas manual o automáticamente; pero para retirarse las partes deben seguir una política como la política *FIFO* o la *FILO*.
- **Máquinas herramientas.** Son aquellos componentes capaces de realizar tareas de transformación sobre las partes.
- **Sensores y actuadores.** Estos dispositivos proporcionan una interfaz entre el controlador y la máquina o proceso a controlar. Los sensores traducen alguna variable en la máquina o proceso a una señal (en general eléctrica) que puede, a su vez, ser procesada por el equipo de control en forma subsecuente, para obtener el valor actual de la variable de interés. Los sensores pueden ser usados para el control de la seguridad de un *SMF*, inspección de una pieza para el control de calidad, posición e información relacionada con las piezas en el *SMF*, reconocimiento de partes, entre otros. Los actuadores aceptan una señal del sistema de control y, de alguna manera, imponen esta señal sobre la máquina o el proceso a controlar. Tanto los sensores como los actuadores pueden ser continuos o discretos.

Todos los componentes de un *SMF* son controlados por un sistema de decisiones con el objetivo de producir bienes que cumplan con los criterios de calidad establecidos para satisfacer las necesidades del cliente. La siguiente tabla muestra algunos componentes de un *SMF* y sus respectivas operaciones que pueden efectuar.

<i>Componente de la Célula</i>	<i>Posibles operaciones</i>
Robot	Mover, tomar, colocar.
Banda Transportadora	Avanzar, detener.
Máquina	Ensamblar, taladrar, cortar, torneado, fresar, etc.
Sistema de visión	Reconocimiento.

Enseguida se describe brevemente en qué consisten las operaciones anteriormente señaladas.

CAPÍTULO 2. CONCEPTOS BÁSICOS

- **Tomar.** Es la operación consiste en que el robot sujete una parte en su pinza y la mantenga sostenida para poder desplazarla o efectuar otra operación.
- **Colocar.** Esta acción significa que el robot debe liberar la parte que retenía en su pinza sobre un sitio particular.
- **Mover (Coordenadas).** Indica que el robot tiene que desplazarse desde su posición actual o una posición de seguridad hasta una posición indicada mediante coordenadas.
- **Avanzar.** Esta operación se lleva a cabo mientras una banda transportadora esté en movimiento, la cual permite el transporte de partes.
- **Detener.** Indica que la banda transportadora suspendió temporalmente su movimiento.
- **Reconocimiento.** Esta operación la desarrollan los sensores para identificar una parte de entre un conjunto establecido.
- **Alimentar y descargar.** Esta acción es ejecutada por los robots o por las bandas transportadoras al introducir (retirar) partes a (de) la célula o alguna máquina herramienta.

Las otras operaciones son claramente indicadas por su nombre.

Otro de elemento importante de los *SMF* son los sitios. Un sitio es un lugar donde una parte recibe un tratamiento especial, por ejemplo ensamblado, cromado, pintura, barnizado, cobertura con resina, entre otros. Este tipo de componentes también pueden almacenar o mantener temporalmente a las partes, su capacidad depende del número de partes que puede albergar, por ejemplo un sitio de n lugares tiene la capacidad de recibir hasta n partes.

2.6. Conclusiones.

Este capítulo presentó una definición formal de las *RPG* así como algunas de las propiedades básicas y clasificación de las *RP*. También la definición formal de las *RPC*, como una extensión de las *RP* para el modelado de sistemas de grandes dimensiones y reducción de representaciones repetitivas, fue introducida en este capítulo. Asimismo se presentaron algunos conceptos básicos, relacionados con los componentes de los *SMF*, los cuales serán de ayuda a lo largo de este trabajo.

3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES.

Resumen: En este capítulo se presentan los métodos de síntesis de aplicación general para el modelado de SED con RP. También se definen los módulos elementales en RPC y sus respectivas funciones las cuales se utilizarán en la metodología de modelado ascendente propuesta. El propósito de estos módulos es contar con una librería para modelar fácilmente las operaciones de un SMF, sus funciones tienen la característica de ser matrices bisimétricas, las cuales permitirán resolver las ecuaciones implicadas en el diseño del control supervisor más rápidamente.

3.1. Introducción.

Como con la utilización de cualquier formalismo de modelado de SED's, la representación obtenida de un sistema mediante la utilización de RP no es única; la simplicidad del modelo resultante depende en su mayor parte de la experiencia del diseñador y de la cantidad de información que se requiera capturar del sistema para que éste pueda ser analizado. Asimismo es importante tener en consideración si se pretende lograr una minimización en la representación del sistema, o bien, describir detalladamente las operaciones en el modelo.

El tratar de obtener el modelo de un SED de gran talla solamente en un paso puede llegar a ser demasiado complicado y existe una mayor probabilidad de introducir errores. Por tal motivo se han diseñado metodologías de modelado modulares, las cuales han permitido atacar el problema por partes y de esta manera lograr obtener con éxito y de manera más sencilla el modelo total de un SED.

Este capítulo se divide de la siguiente manera: En la sección 3.2 se presentan brevemente los métodos de síntesis para la elaboración de modelos utilizando las RP. En base a los componentes de los SMF que se presentaron en el capítulo anterior y a las operaciones, se obtiene los módulos elementales en RPC que se

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

requieren para modelar un *SMF* y las características que estos deben capturar, lo cual es expuesto en la sección 3.3. Posteriormente en la sección 3.4 se muestran las funciones asociadas a cada uno de los módulos construidos en la sección anterior. Por último las conclusiones son presentadas en la sección 3.5.

3.2. Métodos de Modelado.

En esta sección se presentan los métodos más comunes para la construcción de modelos utilizando las *RP*.

3.2.1. Métodos Ascendentes.

Este tipo de técnicas se basan en la descomposición del sistema total en subsistemas. La estrategia consiste principalmente en modelar cada una de las partes obtenidas al descomponer el sistema para finalmente unir las y así formar el modelo del sistema total. Cada subsistema se modela de acuerdo con el nivel de detalle requerido por de las operaciones dentro del subsistema; en este caso la preocupación principal recae sobre el modelado del subsistema ignorando, en gran parte, la interacción con los demás subsistemas.

Los modelos de los diferentes subsistemas pueden tener lugares o transiciones que representan recursos o tareas en común con otros subsistemas, estos nodos en común constituyen las interacciones que existen entre los subsistemas. Después de que se han construido los modelos de todos los subsistemas, el siguiente paso es unir los modelos por medio de la fusión de lugares, de transiciones o un camino. La fusión de lugares generalmente se emplea cuando dos subsistemas comparten un mismo recurso como se aprecia en la figura 3.1, en cambio cuando se presentan actividades en común entre subsistemas o el fin de una tarea es pauta para el comienzo de otra se lleva a cabo la fusión de transiciones mostrado en la figura 3.2. Cuando los modelos de subsistemas se construyen a través de circuitos elementales es cuando pueden fusionarse por caminos. estos caminos en común deben comenzar y terminar en lugares, como en el ejemplo mostrado en la figura 3.3, o iniciar y finalizar en transiciones, como se exhibe en la figura 3.4.

Las técnicas ascendentes también permiten la construcción de estructuras en *RP* que formen cadenas de operaciones o una secuencia con elección, lo cual es exhibido en la figura 3.5.a) y 3.5.b) respectivamente, para este ejemplo se supondrá

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

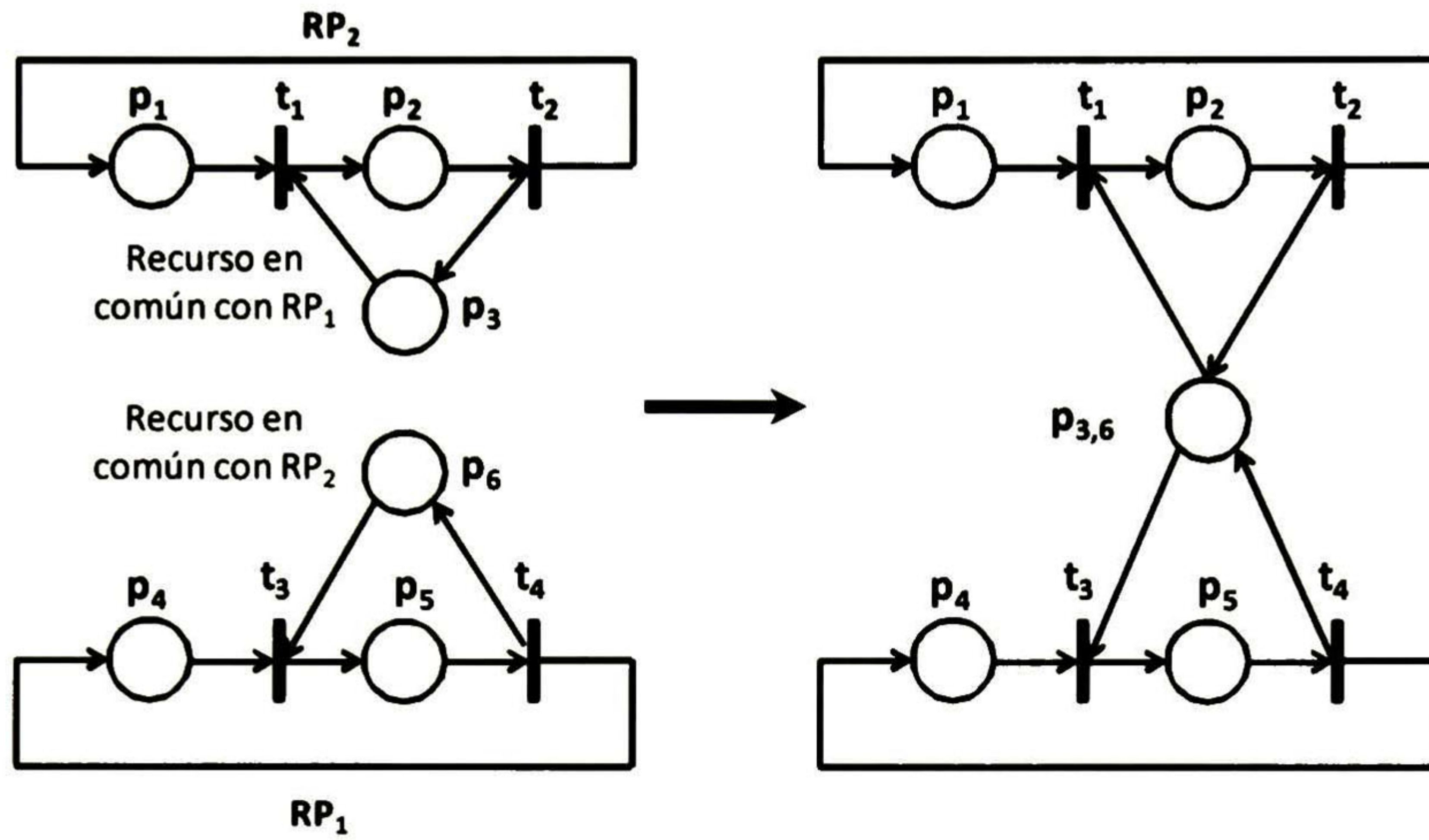


Figura 3.1: Unión de módulos por fusión de lugares.

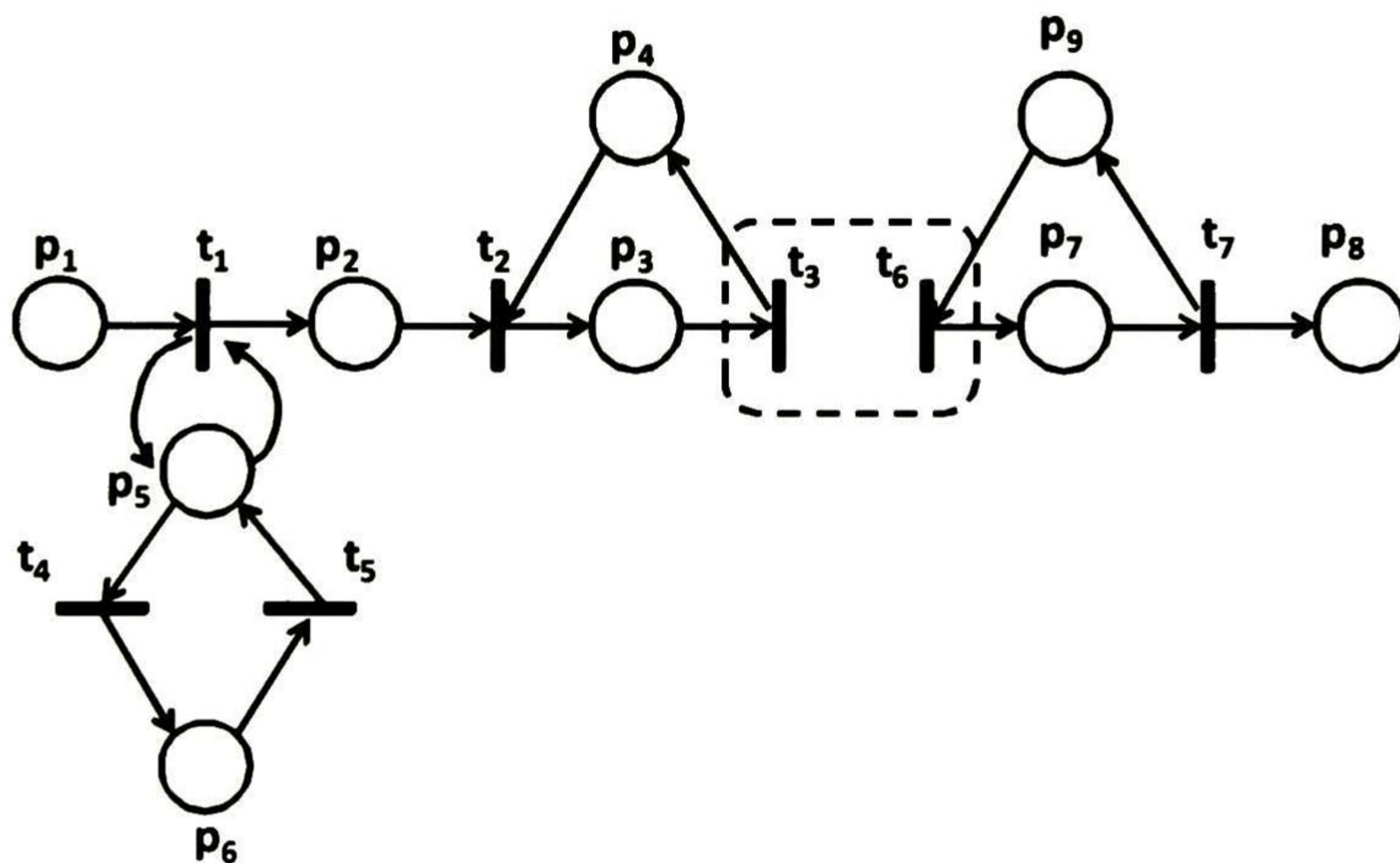


Figura 3.2: Unión de módulos por fusión de transiciones.

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

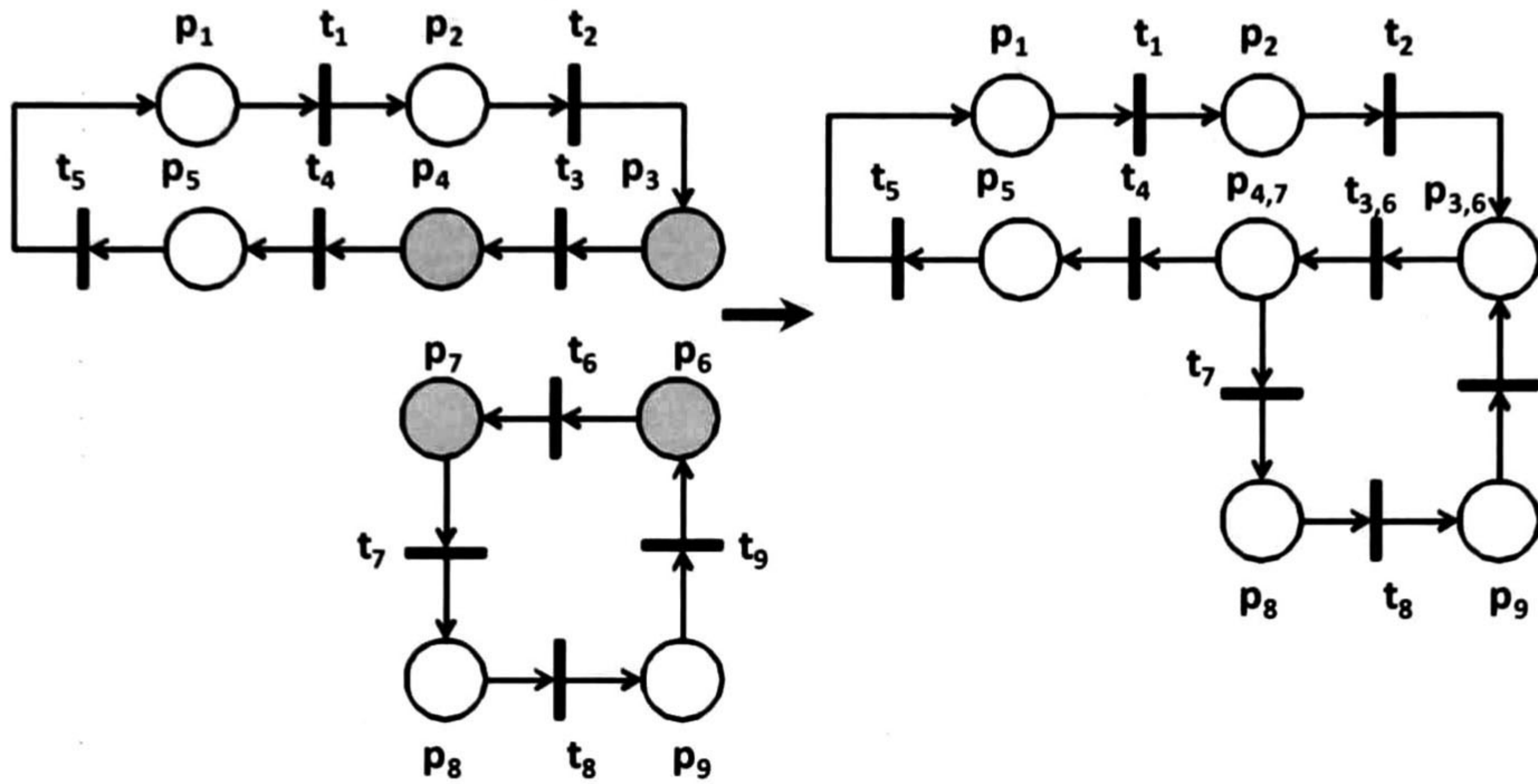


Figura 3.3: Unión de módulos por fusión de caminos que incian y terminan en lugares.

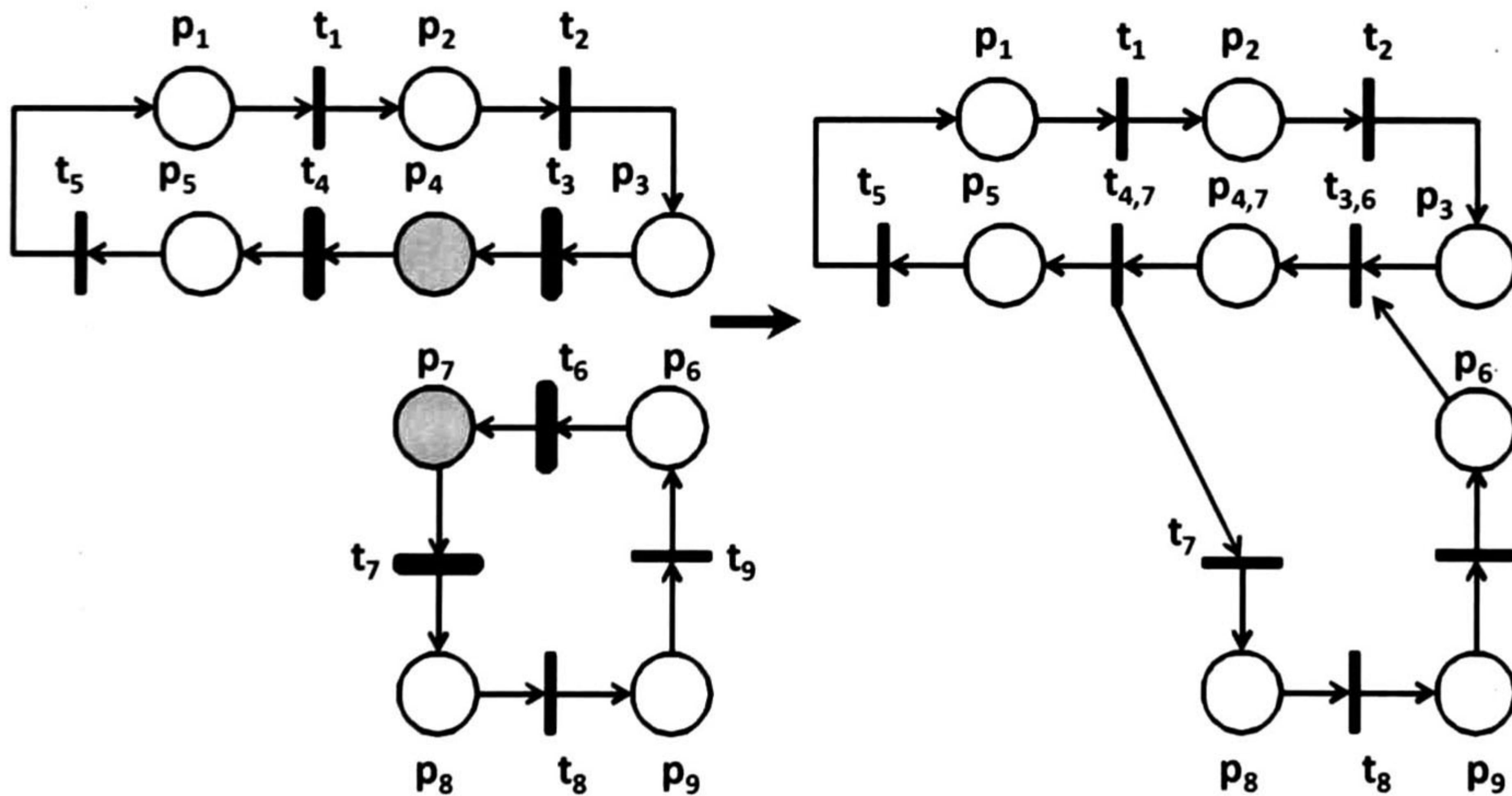


Figura 3.4: Unión de módulos por fusión de caminos que incian y terminan en transiciones.

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

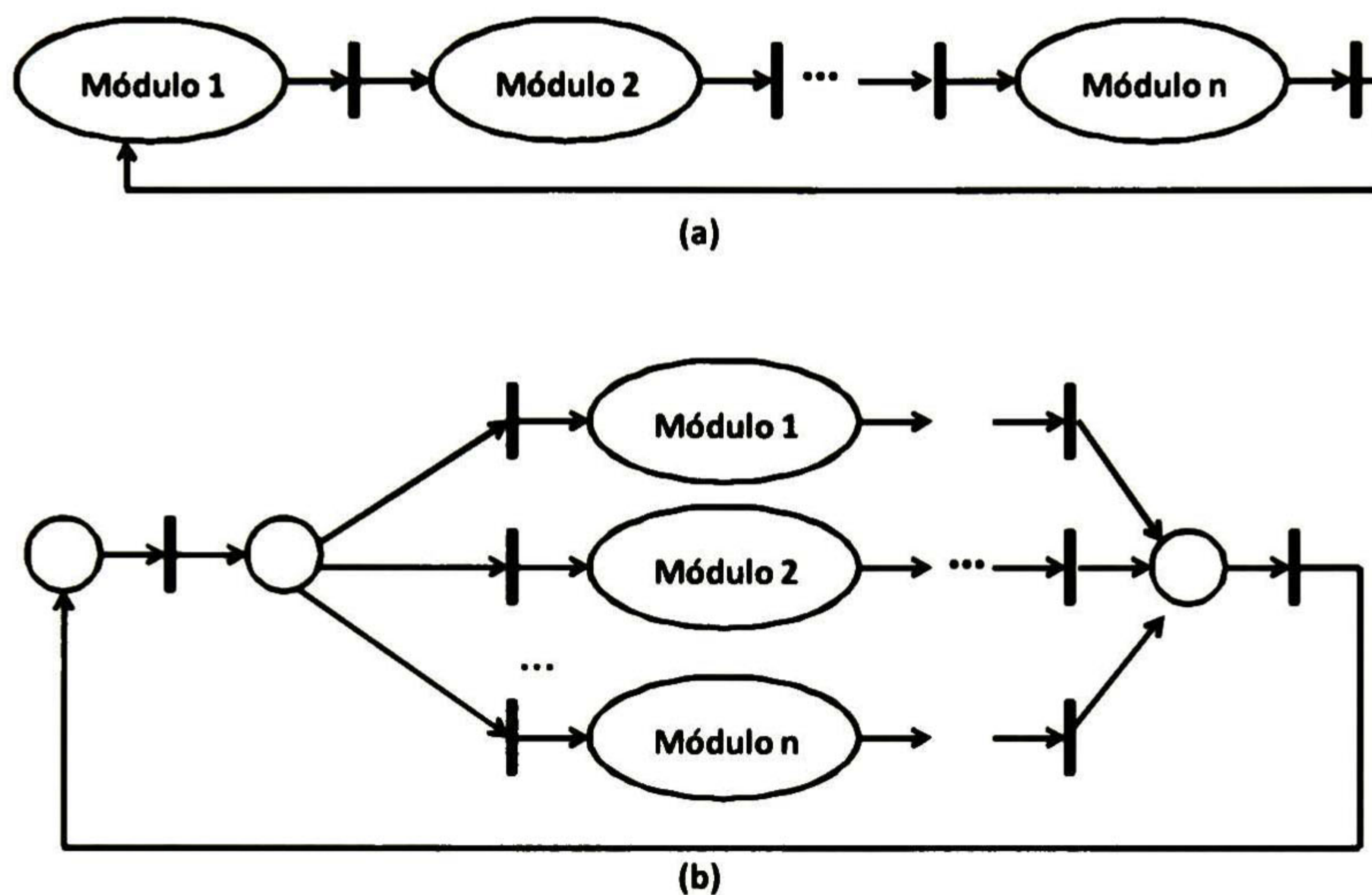


Figura 3.5: Estructuras básicas de *RPC* con módulos. (a) Secuencia de operaciones. (b) Secuencia de operaciones con elección.

que los módulos tienen un lugar inicial y uno final para poder concatenarlos como indica la figura.

Ventajas del Método Ascendente.

El construir el modelo del sistema en *RP* basado en una metodología ascendente permite especificar detalladamente y con mayor facilidad el sistema, al considerarlo como una composición de subsistemas independientes. Otra ventaja es que puede facilitar e incluso, en algunos casos, evitar el análisis al llevar a cabo la composición de subredes.

3.2.2. Métodos Descendentes.

Estos métodos consisten en generar un modelo en *RP* con un gran nivel de abstracción, el cual represente a un enfoque global del sistema, sin importar particularidades de bajo nivel. La siguiente etapa es llevar a cabo una expansión de lugares y transiciones para dar detalle al modelo del sistema, a lo cual se le

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

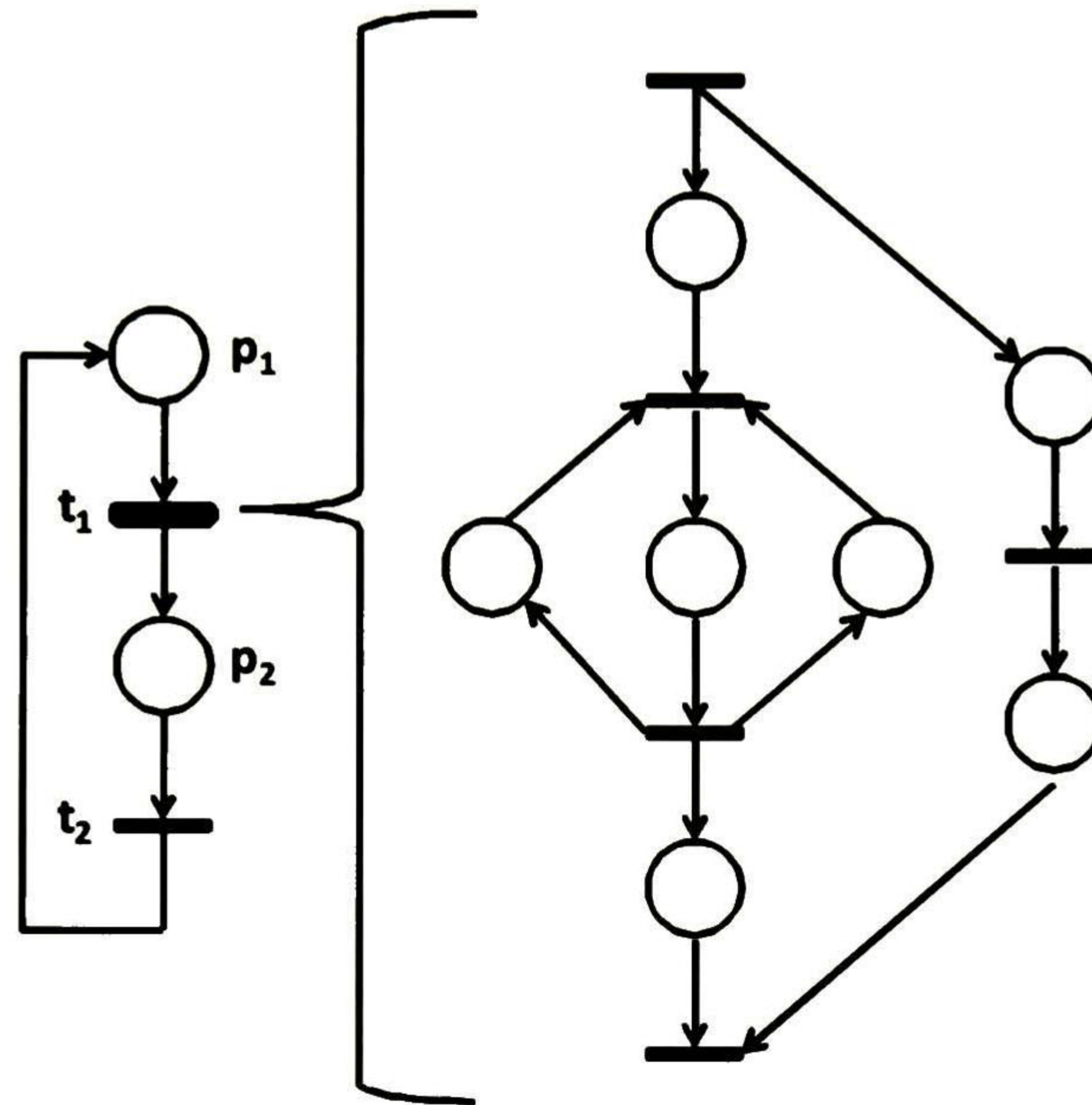


Figura 3.6: Refinamiento de transiciones en el método descendente.

conoce como refinamiento. La operación de refinamiento debe aplicarse hasta que el modelo cumpla con la información que se quiere capturar del sistema.

Por ejemplo en la figura 3.6 se muestra un modelo inicial del sistema global bastante y al refinar la transición t_1 se agrega detalle al modelo del sistema, pues ahora se cuenta con dos caminos que representan actividades en paralelo, donde una de estas actividades requiere de dos recursos para ser completada. En un siguiente paso se puede refinar algún otro lugar o transición hasta llegar al nivel de especificación requerido.

Debido a que esta técnica de síntesis le es más complicado expresar una exclusión mutua durante la etapa de refinamiento, ésta es mejor para la construcción de modelos de sistemas secuenciales.

Ventajas del Método Descendente.

En este procedimiento para elaborar modelos con *RP* la principal ventaja es que al tener una visión global del sistema permite tomar en cuenta las interacciones

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

entre los posibles subsistemas, así como los elementos requeridos para especificar dichas interacciones. Cuando se trata de sistemas secuenciales que no compiten por recursos, el modelado con esta técnica puede ser sencillo.

3.2.3. Métodos Híbridos.

Esta técnica de modelado es una combinación de los dos métodos anteriores. En este caso los refinamientos de lugares y transiciones que se lleven a cabo son por medio de módulos elementales que representan subsistemas bien definidos.

El modelo inicial en este método es una red viva, acotada y cíclica. Debe ser de este modo debido a que esta técnica busca que estas propiedades se conserven en el modelo final del sistema. Este método no tiene problemas en representar exclusiones mutuas puesto que se pueden agregar arcos y lugares para representar a los recursos compartidos.

3.3. Módulos Elementales.

En este trabajo se propone una metodología de modelado ascendente, la cual está basada principalmente en la obtención de módulos básicos y después unirlos para obtener el modelo del sistema. En esta sección se presentan los módulos elementales usados para modelar los elementos físicos y operaciones de los *SMF*.

3.3.1. Módulo en *RPC* representando el Robot.

Un robot es un manipulador programable y multifuncional, diseñado para mover materiales, partes, herramientas, entre otros, a través de movimientos variables, o bien son programados para desarrollar diversas actividades sobre las partes.

Un robot está compuesto por un brazo robótico, una pinza para poder efectuar las operaciones de tomar, mover y colocar, o bien en lugar de la pinza puede tener en su extremo una herramienta para llevar a cabo una operación sobre las partes. El módulo que se presentará captura el comportamiento cuando el robot cuenta con una pinza el extremo de su brazo.

Los módulos con *RPC* que representan las tareas que el robot debe desarrollar son mostrados en la figura 3.7.a), 3.7.b) y 3.7.c), y el comportamiento de la pinza y el robot es modelado en la figura 3.8.a) y 3.8.b). Cabe señalar que los elementos del robot son representados por P-invariantes, sin embargo, el modelado de las tareas es arbitrario. Los modelos del brazo y la pinza están compuestos por un

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

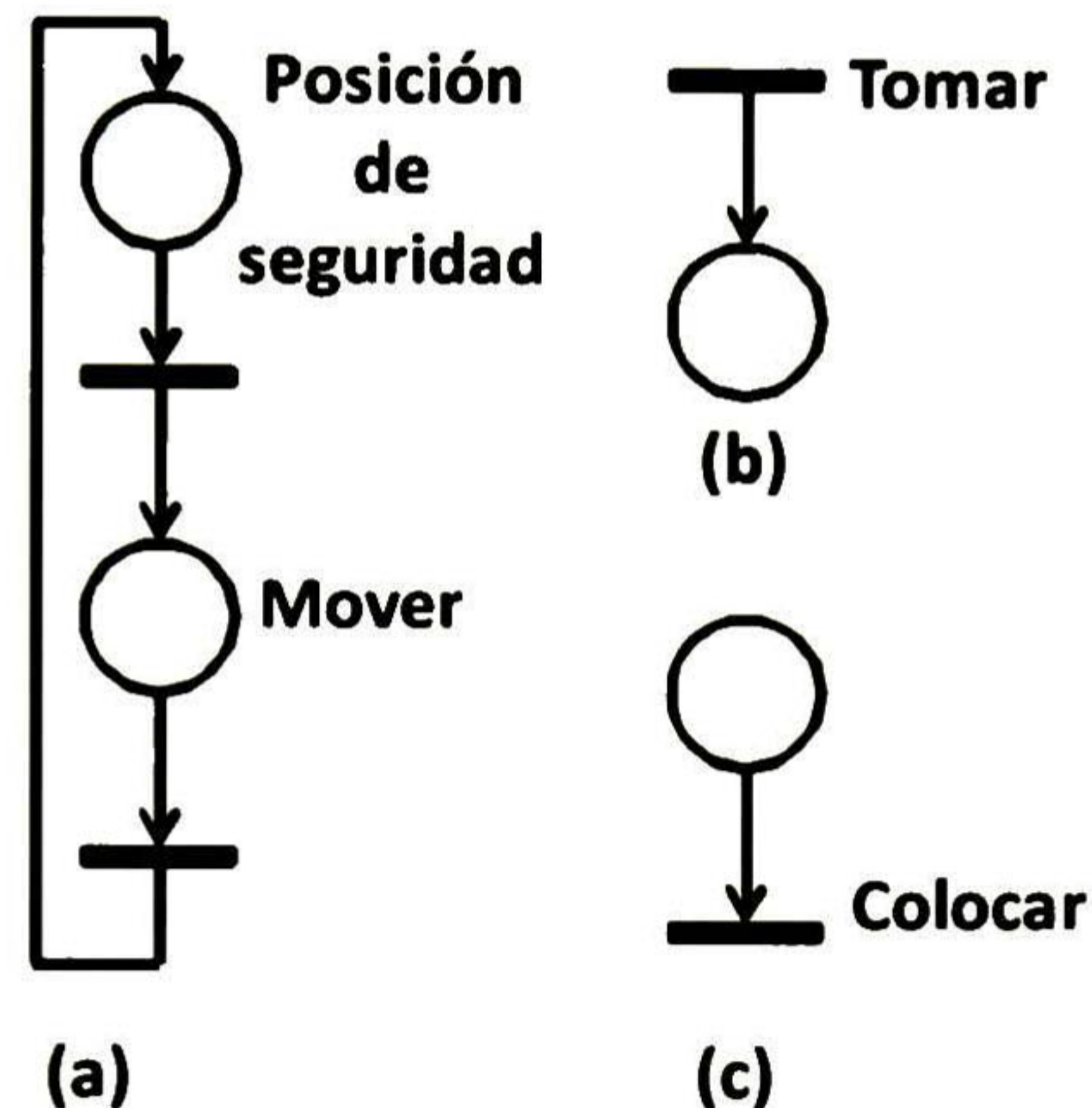


Figura 3.7: Modelo en *RPC* de las tareas del robot.

lugar que representa cuando el elemento está ocioso (cuando el brazo se localiza en la posición de seguridad o cuando la pinza se encuentra abierta) y otro lugar que simboliza al elemento en actividad. Cabe destacar que si la tarea es cíclica el modelo también es un P-invariante, si la tarea consiste en tomar partes de un sitio externo al sistema, entonces debe representarse por una transición de entrada que pueda suministrar marcas indefinidamente a un lugar, y si la tarea consiste en liberar partes en un sitio externo al sistema, entonces tiene que representarse por una transición de salida que pueda eliminar marcas del modelo.

El siguiente paso consiste en unir los módulos. Para realizar este paso existen dos operaciones. La primera establece que las tareas y los elementos modelados pueden mezclarse usando operaciones permisivas y en el segundo las tareas y los elementos modelados pueden fusionarse usando operaciones de sincronización. El modelo obtenido al aplicar operaciones permisivas es más realista y representa todos los posibles comportamientos concurrentes; no obstante el análisis de un modelo puede ser complejo. Sin embargo, por medio de operaciones de sincronización se fusionan transiciones que representan a la misma operación y el modelo obtenido al usar esta operación representa el comportamiento del sistema bajo condiciones normales y algunas evoluciones concurrentes son removidas.

La figura 3.9.a) representa el modelo del sistema del robot al aplicar operaciones permisivas y la figura 3.9.b) muestra el modelo obtenido cuando se llevan

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

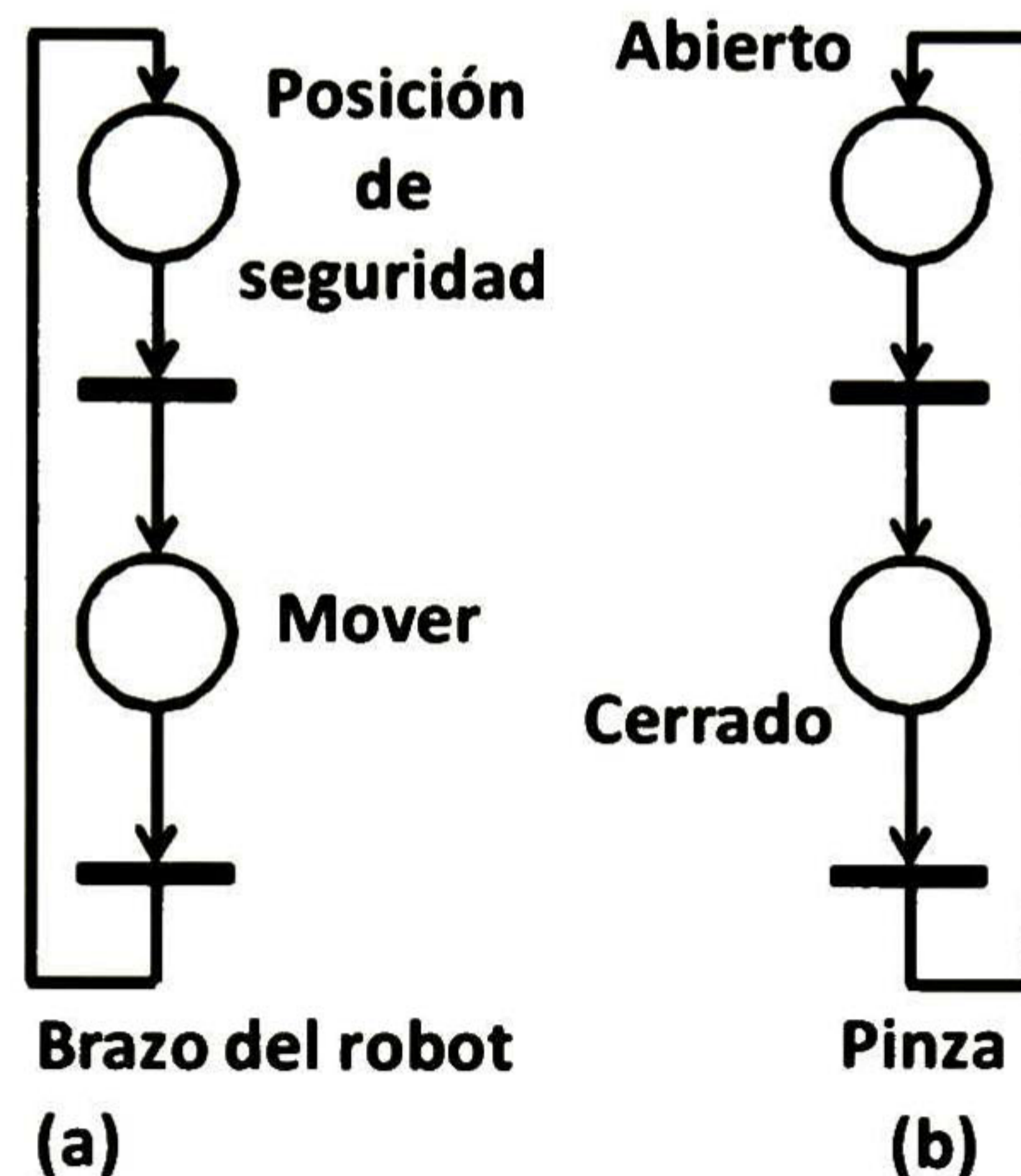


Figura 3.8: (a) Modelo del comportamiento del brazo del robot con *RPC*. (b) Modelo del comportamiento de la pinza con *RPC*.

a cabo operaciones de sincronización. En ambos casos el modelo en *RPC* puede ser simplificado eliminando lugares implícitos, obteniendo así el modelo expuesto en la figura 3.10.

3.3.2. Módulo en *RPC* representando una Banda Transportadora.

Una banda transportadora es capaz de llevar a cabo operaciones de avanzar las partes que estén sobre ella y detener su movimiento. Obviamente si la banda transportadora está llena una parte no puede entrar y las partes dentro de ella no podrán avanzar hasta que se retire la parte que se localice al final de la banda transportadora; por lo tanto es necesario un módulo en *RPC* que tome en cuenta estas consideraciones.

En la figura 3.11.a) se muestra un modelo en *RP* que considera las características previamente señaladas por medio de una asignación sucesiva de recursos, donde el recurso asignado representa un lugar de la banda. Es posible ver a la banda como una sucesión de sitios si está discretizada.

El modelo en *RP* puede ser visto como la unión de un conjunto de módulos de asignación de recursos, éstos pueden ser fusionados en un módulo único utilizando

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

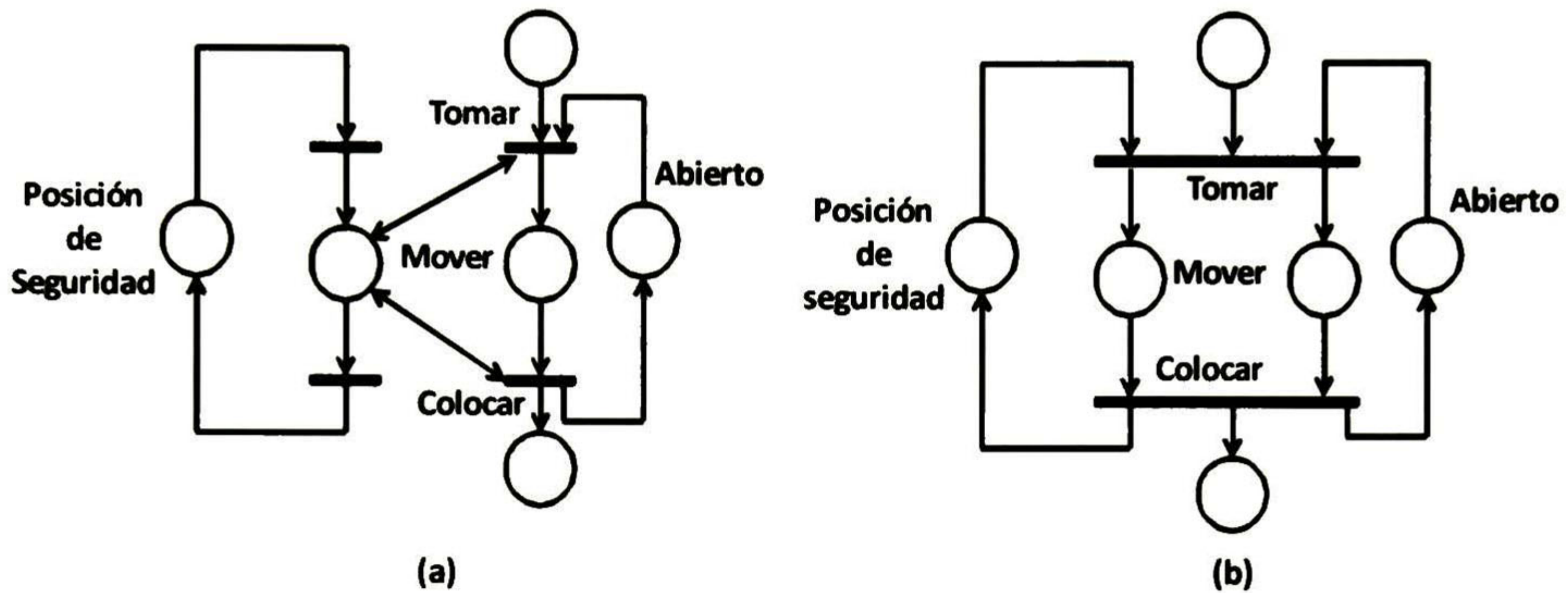


Figura 3.9: (a) Operación permisiva. (b) Operación de sincronización.

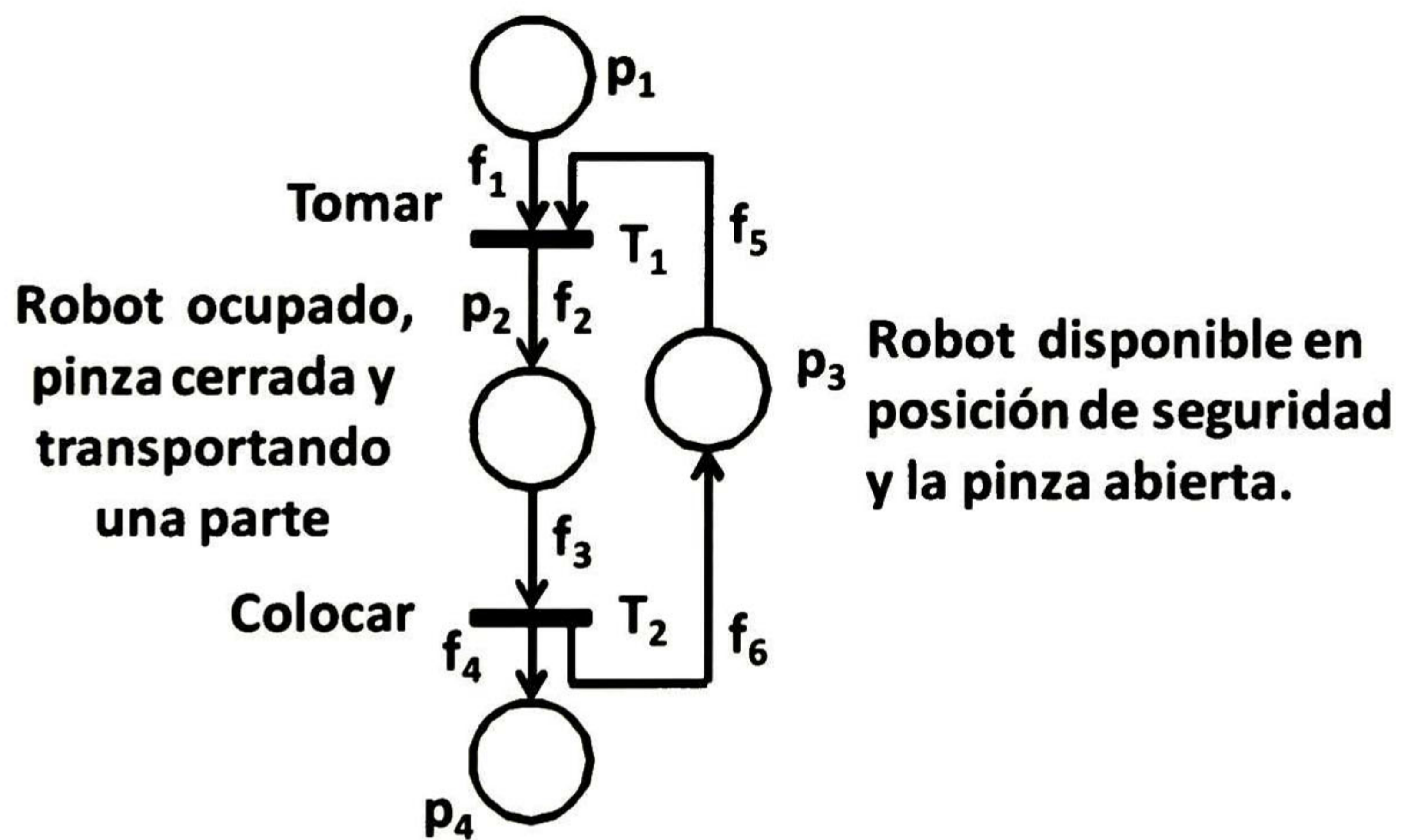


Figura 3.10: Módulo en *RPC* representando al robot.

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

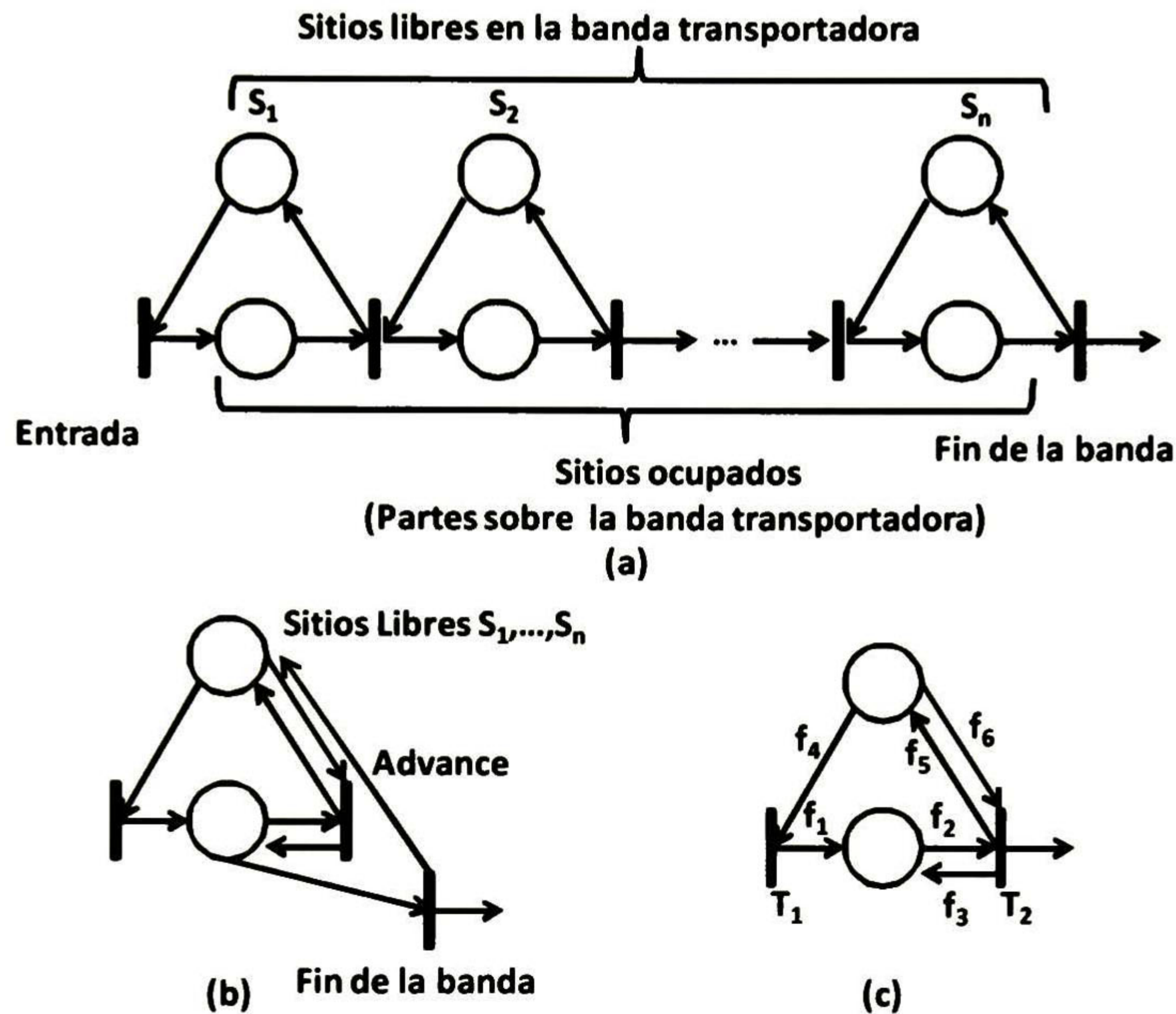


Figura 3.11: (a) Modelo de la banda transportadora con *RP*. (b) y (c) Módulo de la banda transportadora con *RPC*.

las *RPC*, donde un lugar representa los diferentes sitios (lugares desocupados) y el otro las partes sobre la banda transportadora, lo cual es mostrado en la figura 3.11.b). Finalmente si la transición que representa el final de la banda y el avance son fusionadas en una sola se obtiene el modelo expuesto en la figura 3.11.c) donde la parte solamente puede abandonar la banda con la combinación correcta de colores.

3.3.3. Módulo en *RPC* representando Máquinas.

El módulo que representa a una máquina en *RPC* puede ser el mismo que el del robot, debido a que sigue el mismo concepto de asignación de recurso. En este caso un lugar representa a la máquina ocupada y el otro lugar cuando está ociosa; pero el lugar que simboliza a la máquina ociosa debe contener tantas marcas coloreadas de acuerdo a como lo señale la capacidad de la máquina, es decir, el número de

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

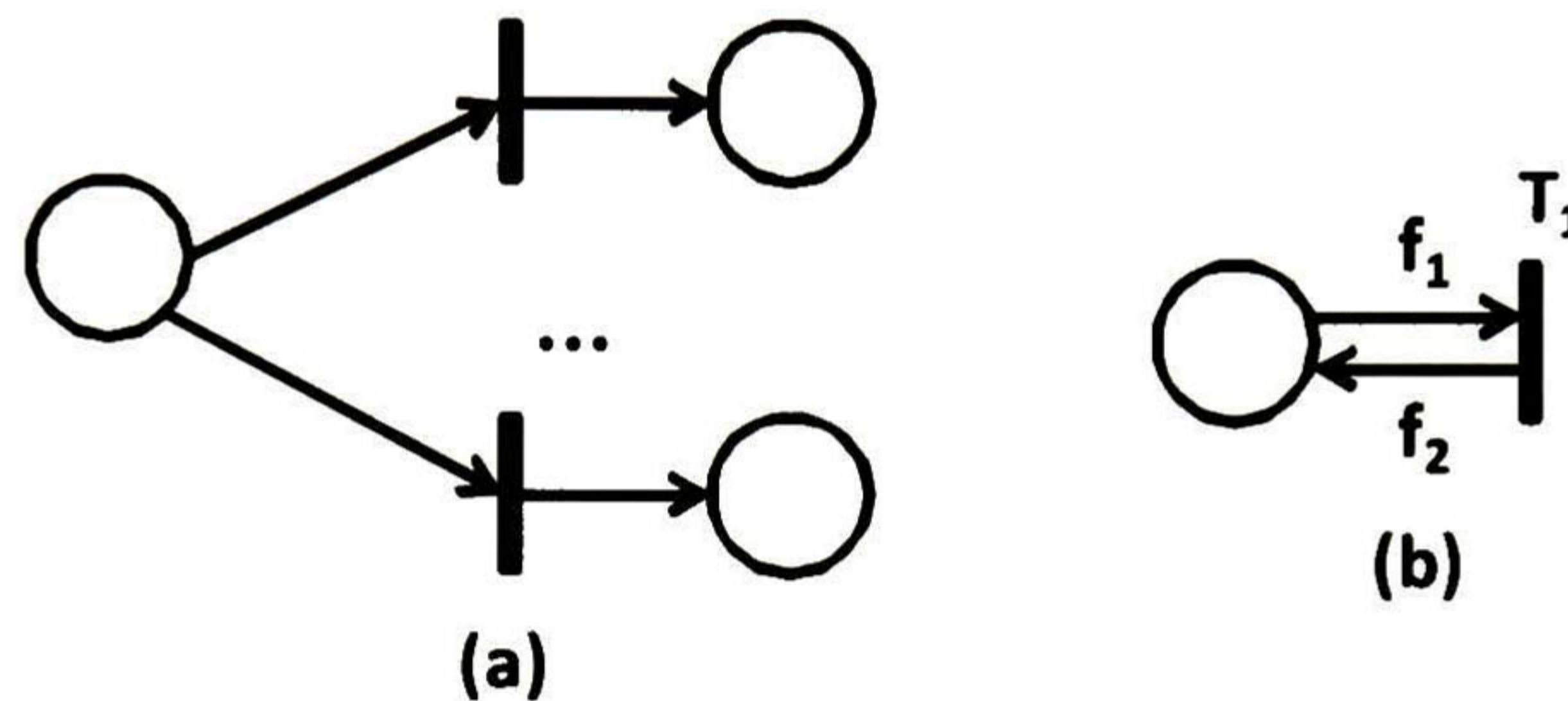


Figura 3.12: (a) Reconocimiento con *RP*.(b) Módulo de reconocimiento con *RPC*.

partes que pueda procesar a la vez.

3.3.4. Módulo en *RPC* representando la operación de reconocimiento.

Como se mencionó en el capítulo anterior la operación de reconocimiento es ejecutada por sensores o un sistema de visión artificial. El modelo en *RP* de esta operación es representado en la figura 3.12.a) donde dependiendo de la transición que se dispare es el elemento que el sensor reconoció de entre el conjunto previamente establecido.

Si se fusionan todas las transiciones en una sola, así como los lugares de entrada y salida en un único lugar se obtiene el módulo en *RPC* mostrado en la figura 3.12.b) donde la transición T_1 recibe marcas sin color por medio de f_1 y a través de f_2 regresa marcas coloreadas.

3.3.5. Módulo en *RPC* representando los sitios de almacenamiento.

Un almacén puede guardar o liberar partes; pero con el fin de construir su respectivo modelo es necesario tomar en cuenta otras consideraciones como la regla de almacenaje.

En un almacén tipo *FIFO* (Primero en entrar, primero en salir) la primer parte en ser almacenada tiene que ser la primera parte en ser liberada del almacén. Esta política sigue el mismo comportamiento que una banda transportadora motivo por el cual el modelo puede ser el mismo.

Un almacén tipo *FILO* (Primero en entrar, último en salir) es aquel en el que la primera parte en ser guardada tiene que ser la última en ser retirada del almacén. El modelo en *RP* de este tipo de almacén es mostrado en la figura

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

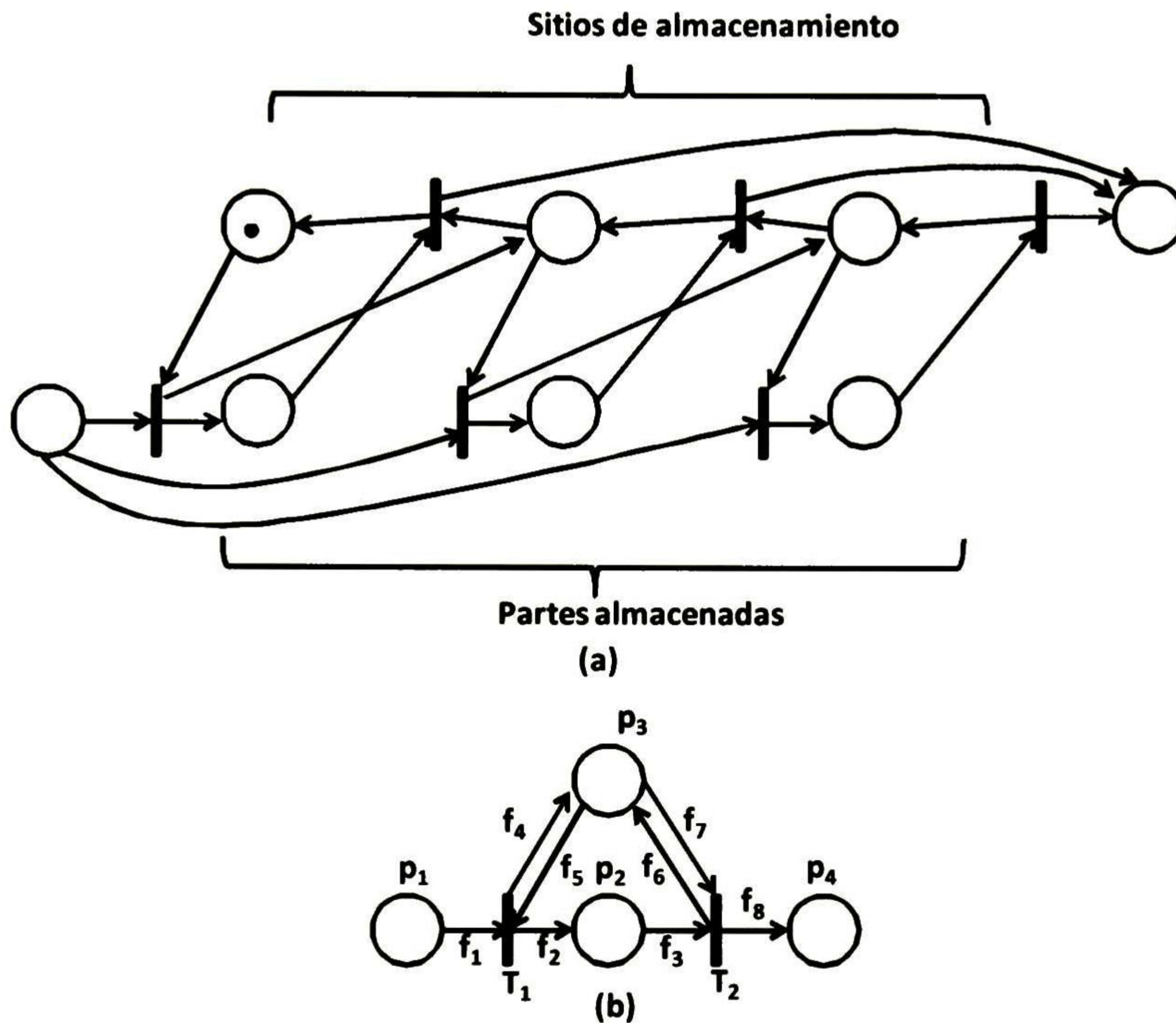


Figura 3.13: (a) Almacenamiento tipo *FILO* con *RP* (b) Módulo de almacenamiento tipo *FILO* con *RPC*.

3.13.a). Si los módulos repetitivos son fusionados se obtiene el módulo único en *RPC* expuesto en la figura 3.13.b).

Un almacén de acceso aleatorio puede tener el mismo modelo en *RPC* que el robot o la máquina puesto que es una asignación de un recurso, en este caso el sitio de almacenamiento.

Un almacén temporal de acceso aleatorio puede ser modelado por un único lugar sólo si su capacidad de almacenamiento nunca es excedida.

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

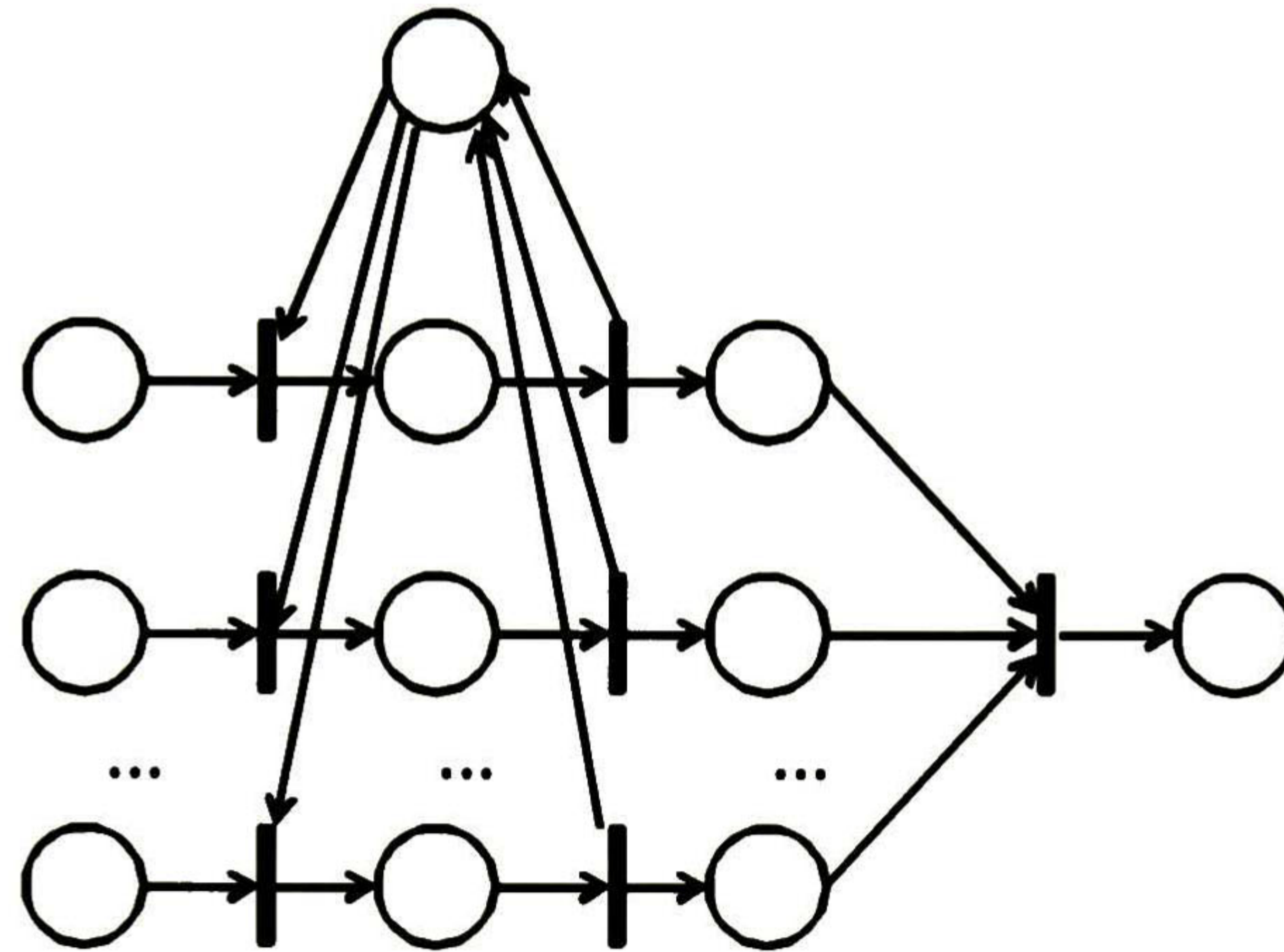


Figura 3.14: Modelo de ensamblaje con *RP*.

3.3.6. Módulo en *RPC* representando un ensamblaje.

La operación de ensamblado puede ser realizada por un robot, una máquina o manualmente por un ser humano. Siguiendo el mismo razonamiento que en los casos anteriores es posible construir un módulo en *RP* que represente el ensamblaje, mostrado en la figura 3.14, y después simplificarlo para obtener el modelo en *RPC* presentado en la figura 3.15.a). En el módulo que representa la especificación de la operación de ensamblaje, mostrado en la figura 3.15.b), es necesario un arco bidireccional y un lugar adicional que ayudan para dar el secuenciamiento necesario en la operación de ensamblaje, también se requiere agregar los arcos con las funciones f_7 y f_8 donde la suma de ellos es igual a una de las funciones propuestas para llevar a cabo el ensamblaje.

Es importante aclarar que el módulo de la especificación de la operación de ensamblaje es uno de muchos que se podrían construir; pero éste se hizo de esta manera con el propósito de utilizar las funciones propuestas.

3.4. Funciones.

En esta sección son definidas las funciones asociadas a los módulos introducidos en la sección anterior, las cuales serán utilizadas en la metodología de modelado

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

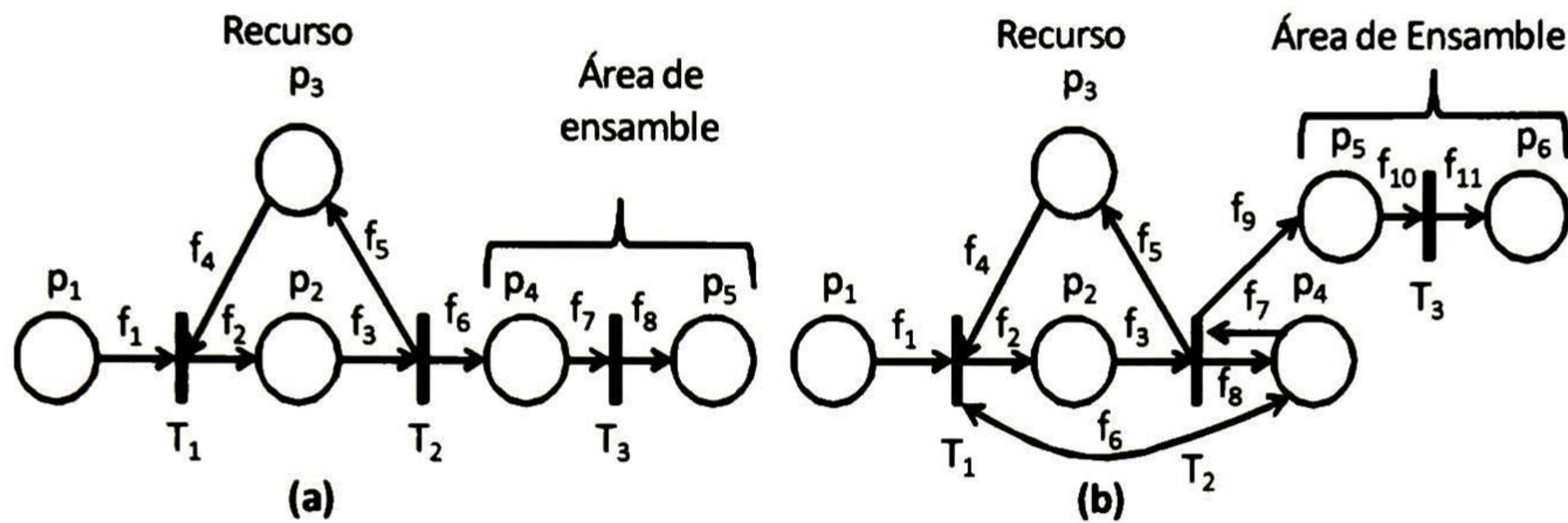


Figura 3.15: (a) Módulo de la operación de ensamblaje con *RPC*. (b) Módulo de la especificación de la operación de ensamblaje con *RPC*.

propuesta.

Es posible transformar los bloques matriciales de la matriz de incidencia en bloques matriciales bisimétricos pertenecientes a un anillo conmutativo con identidad.

Se les llamará colores principales a los colores asignados a las partes, porque el flujo de éstas a través del sistema es lo que nos interesa y estos colores nos ayudarán a determinar la dimensión de los bloques matriciales.

- **Función identidad.** Esta función aparece cuando las partes están usando uno o más recursos, para ser movidas o para que se ejecute una acción sobre ellas (el movimiento de los colores principales). Esta función es representada con la matriz identidad.
- **Función de Asignación de recursos.** Esta función se presenta cuando un recurso es tomado para poder llevar a cabo una operación. Cuando es asignado un solo recurso, para cualquiera de las partes, la matriz bisimétrica que se presenta en el modelo tomará la forma:

$$Asign_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

donde el subíndice 4 representa la dimensión del bloque matricial. También esta función se utiliza cuando se requiere de una operación de re-

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

conocimiento, donde la dimensión del bloque dependerá de la cantidad de los colores que existan en el conjunto de identificación. Como se puede apreciar este bloque matricial es redundante porque únicamente se requiere de un solo renglón (en el caso de la asignación del robot u operación de identificación) o una columna (en caso de la operación de ensamble); pero con el fin de igualar la dimensión del bloque con la dimensión de otras funciones y poder realizar cálculos más rápidos se utiliza de esta manera.

En caso de que los recursos sean asignados por pares a un elemento (como el caso de los filósofos comensales localizado en [15] o [19] donde dos tenedores específicos son asignados a cada filósofo) la función de asignación toma la forma de:

$$Asign_5^2 = \begin{pmatrix} 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 \end{pmatrix}$$

donde el superíndice 2 denota que dos recursos son asignados simultáneamente.

- **Función de ensamble.** La función de ensamble se presenta cuando nosotros construimos la especificación para esta operación, la cual es una de las más complejas debido a que su forma depende de la dimensión del bloque, y de acuerdo a la secuencia de ensamblado cambia el marcado inicial que se le asigna a este bloque y la matriz especial de ceros que la acompaña a esta función, dicho bloque de ceros debe aparecer en el mismo renglón de la matriz de incidencia donde se localice la función de ensamble.

De acuerdo a la dimensión del bloque la función de ensamble es:

$$Ens_2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad Ens_3 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

$$Ens_4 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

Ahora tomemos por ejemplo el bloque con dimensión 4. Si se requiere la secuencia (a, b, c, d) , se necesita del siguiente marcado inicial y bloque de ceros (El cual se expresará con la matriz $O(p_i, t_j)$ para mayor claridad):

$$O(p_i, t_j) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad M_0 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Con el mismo bloque de ceros; pero cambiando el marcado inicial las secuencias que pueden ser obtenidas son: (b, c, d, a) , (c, d, a, b) y (d, a, b, c) . Una secuencia de orden inverso (d, c, b, a) puede obtenerse con:

$$O(p_i, t_j) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad M_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

De la misma manera si se cambia el marcado inicial es posible obtener los ensambles (c, b, a, d) , (b, a, d, c) y (a, d, c, b) .

Para lograr obtener otras permutaciones en la secuencia de ensamble se debe trabajar con la función de ensamble para encontrar su bloque de ceros y marcado inicial correspondientes.

3.4.1. Funciones en los módulos.

Ahora se especificarán las funciones de los módulos previamente establecidos.

- **Funciones en el módulo que representa al robot.** En la figura 3.10 las funciones son $f_1 = f_2 = f_3 = f_4 = I_n$ y $f_5 = f_6 = Asign_n$
- **Funciones en el módulo que representa a la banda transportadora.** En la figura 3.11.c) las funciones son $f_1 = f_2 = f_3 = I_n$, $f_4 = f_5 = f_6 = Asign_n$.
- **Funciones en el módulo representando reconocimiento.** En la figura 3.12.b) se tiene que $f_1 = Asign_n$ y la función $f_2 = I_n$.
- **Funciones en el módulo de almacenamiento tipo FILO.** En la figura 3.13.b) las funciones son $f_1 = f_2 = f_3 = f_4 = I_n$ y $f_5 = f_6 = f_7 = f_8 = Asign_n$.

CAPÍTULO 3. MODELADO DE ELEMENTOS FÍSICOS Y OPERACIONES

- **Funciones en el módulo de ensamble.** En la figura 3.15.a) las funciones son $f_1 = f_2 = f_3 = f_6 = I_n$, $f_4 = f_5 = f_{10} = A\text{sign}_n$ y $f_8 = r(A\text{sign}_n)$ (función que representa a un solo renglón de la función $A\text{sign}_n$). En la figura 3.15.b) las funciones son $f_1 = f_2 = f_3 = f_9 = I_n$, $f_4 = f_5 = f_{10} = A\text{sign}_n$, $f_7 + f_8 = E\text{ns}_n$, f_6 es una matriz de ceros (0_n^*) que cambia de acuerdo a la secuencia de ensamble.

Nota 18. Con las matrices bisimétricas es posible construir diferentes anillos conmutativos con identidad; pero no cualquier matriz bisimétrica puede pertenecer a estos anillos, sin embargo cuentan con elementos en común. Por ejemplo, un anillo puede ser el $\text{span}\{I_n, A\text{sign}_n, E\text{ns}_n\}$ y otro anillo el $\text{span}\{I_n, A\text{sign}_n^2, A\text{sign}_n\}$.

Hasta ahora se han presentado los módulos elementales para operaciones básicas de los SMF , existen más que pueden ser modeladas por medio de módulos siguiendo el mismo razonamiento que para la construcción de los módulos aquí presentados. También otras funciones pueden ser construidas como la siguiente matriz bisimétrica:

$$Diag0_4 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

que no aparece en ninguno de los módulos presentados; pero puede ser de ayuda en el problema de regulación.

3.5. Conclusiones.

Las técnicas de síntesis para construir modelos en RP , ascendentes y descendentes, fueron presentadas en este capítulo. También se establecieron los módulos elementales de las operaciones básicas y componentes de los SMF , que serán utilizados en la metodología de modelado ascendente con RPC propuesta en el próximo capítulo. Asimismo se definieron las funciones que serán empleadas a lo largo de este trabajo, las cuales son matrices bisimétricas que pueden formar anillos conmutativos con identidad, éstas fueron elaboradas de este modo con el fin de efectuar cálculos posteriores con mayor rapidez.

4. METODOLOGÍA DE MODELADO PARA SMF

Resumen: En este capítulo se propone una metodología de modelado ascendente para representar SMF con RPC. Esta metodología de modelado está basada en la identificación de los componentes y operaciones que figuran en el SMF para después ser sustituidas por los módulos propuestos en el capítulo anterior y posteriormente unirlos para construir el modelo de todo el sistema.

También la metodología es aplicada para el modelado de la especificación del sistema con RPC y así poder confinar su comportamiento.

4.1. Introducción.

Hoy en día, se han propuesto diversas metodologías para modelar *SMF* mediante *RP* [36]; desafortunadamente los modelos obtenidos con *RP* resultan ser de grandes dimensiones. Las *RPC* al aumentar el nivel de abstracción permiten que se logren obtener modelos compactos. También se han presentado metodologías de modelado con *RPC*; pero para lograr sintetizar el modelo final del sistema primero deben de pasar por un modelado con *RP* [9], o bien, las funciones que utilizan no permiten llevar a cabo cálculos de invariantes, sumas o multiplicaciones de manera sencilla para resolver el problema de control por regulación.

La metodología de modelado que aquí se presenta permite construir modelos compactos de *SMF* con *RPC*. Es una metodología ascendente la cual captura el comportamiento de las operaciones y componentes elementales de los *SMF* por medio de módulos elaborados con *RPC*. Estos módulos son unidos secuencialmente, de acuerdo al layout del *SMF*, para obtener el modelo del sistema completo.

También en la metodología de modelado se especifica el nivel de abstracción al cual se llegará, pues con las *RPC* los modelos resultantes pueden ser tan compactos que nos impidan identificar al sistema modelado. Por ello, el modelo

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

obtenido al aplicar la metodología propuesta debe reflejar al layout del *SMF* para poder dar seguimiento visual al flujo de trabajo a través del sistema.

Este capítulo está estructurado de la siguiente forma: en la sección 4.2 se presenta la metodología de modelado para *SMF* con *RPC* y aplicada a dos ejemplos aquí presentados. Posteriormente, la sección 4.3 muestra el modelado de la especificación para poder confinar el comportamiento del modelo en *RPC* del sistema. Finalmente las conclusiones son presentadas en la sección 4.4.

4.2. Modelado del sistema.

Un *SMF* está compuesto por elementos programables como robots, bandas transportadoras, vehículos guiados automáticamente, máquinas herramientas y un controlador. La metodología propuesta, la cual puede ser aplicada para obtener el modelo del sistema y de la especificación, consta de los siguientes pasos:

1. Identificar todos los elementos del *SMF* El usuario debe distinguir y clasificar todos los elementos del *SMF*, efectores y sitios del *SMF* bajo estudio. El conjunto $E = \{e_1, \dots, e_k\}$ de elementos del *SMF*, efectores y sitios, debe construirse.
2. Operaciones Elementales. Las tareas u operaciones j que cada elemento e_i tiene que ejecutar deben ser consideradas. El conjunto $O_j^i = \{o_1^i, \dots, o_j^i\}$ de las operaciones del elemento e_i tiene que ser construido. Las tareas y operaciones tienen argumentos, indicando las partes y sitios involucrados durante las operaciones.
3. La información anterior se debe resumir por medio de tablas que relacionen sitios y operaciones (ver tabla 4.1.a), así como efectores y operaciones (ver tabla 4.1.b), para conocer el flujo de trabajo.

<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="padding: 5px;"><i>Sitios</i></th> <th style="padding: 5px;"><i>Operaciones</i></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px; text-align: center;">S_1</td> <td style="padding: 5px; text-align: center;">$O_j^{S_1}$</td> </tr> <tr> <td style="padding: 5px; text-align: center;">S_2</td> <td style="padding: 5px; text-align: center;">$O_j^{S_2}$</td> </tr> <tr> <td style="padding: 5px; text-align: center;">\vdots</td> <td style="padding: 5px; text-align: center;">\vdots</td> </tr> <tr> <td style="padding: 5px; text-align: center;">S_n</td> <td style="padding: 5px; text-align: center;">$O_j^{S_n}$</td> </tr> </tbody> </table> <p style="text-align: center;">(a)</p>	<i>Sitios</i>	<i>Operaciones</i>	S_1	$O_j^{S_1}$	S_2	$O_j^{S_2}$	\vdots	\vdots	S_n	$O_j^{S_n}$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="padding: 5px;"><i>Efectores</i></th> <th style="padding: 5px;"><i>Operaciones</i></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px; text-align: center;">E_1</td> <td style="padding: 5px; text-align: center;">$O_j^{E_1}$ u $O_k^{E_1}$ entonces $O_h^{E_1}$ u $O_m^{E_1}$</td> </tr> <tr> <td style="padding: 5px; text-align: center;">\vdots</td> <td style="padding: 5px; text-align: center;">\vdots</td> </tr> <tr> <td style="padding: 5px; text-align: center;">E_p</td> <td style="padding: 5px; text-align: center;">$O_j^{E_p}$ u $O_k^{E_p}$ entonces $O_h^{E_p}$ u $O_m^{E_p}$</td> </tr> </tbody> </table> <p style="text-align: center;">(b)</p>	<i>Efectores</i>	<i>Operaciones</i>	E_1	$O_j^{E_1}$ u $O_k^{E_1}$ entonces $O_h^{E_1}$ u $O_m^{E_1}$	\vdots	\vdots	E_p	$O_j^{E_p}$ u $O_k^{E_p}$ entonces $O_h^{E_p}$ u $O_m^{E_p}$
<i>Sitios</i>	<i>Operaciones</i>																		
S_1	$O_j^{S_1}$																		
S_2	$O_j^{S_2}$																		
\vdots	\vdots																		
S_n	$O_j^{S_n}$																		
<i>Efectores</i>	<i>Operaciones</i>																		
E_1	$O_j^{E_1}$ u $O_k^{E_1}$ entonces $O_h^{E_1}$ u $O_m^{E_1}$																		
\vdots	\vdots																		
E_p	$O_j^{E_p}$ u $O_k^{E_p}$ entonces $O_h^{E_p}$ u $O_m^{E_p}$																		

(4.1)

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

Cuando el usuario tiene experiencia modelando y el sistema es relativamente pequeño este paso puede ser omitido.

4. Cada operación debe ser reemplazada por su módulo correspondiente en *RPC*.
5. Unir los módulos secuencialmente en el orden definido por la tabla 4.1.b). En algunas ocasiones es necesario añadir arcos cuando un recurso es compartido por varias operaciones y agregar transiciones para concatenar los módulos de las operaciones. Lo que se pretende es que se respete la arquitectura física del *SMF*, es decir, aun siendo posible fusionar módulos de operaciones iguales se recomienda no hacerlo, de este modo el modelo del sistema podrá reflejar la arquitectura física del sistema. Por ejemplo, dos bandas transportadoras tienen la misma estructura en su módulo construido con *RPC*; pero es preferible, en lugar de fusionar los modelos, que cada una tenga su propio módulo en *RPC*, pues de esta manera es más fácil distinguirlas entre sí. Otro ejemplo es cuando se realiza más de una operación de ensamblado en el modelo de la especificación, nosotros podemos fusionar el lugar que representa al robot sólo si hay un único robot que lleve a cabo la tarea de ensamble; pero el lugar que simboliza que la parte que está en la pinza y el resto del módulo no debe ser fusionado. Con las *RPC* se puede alcanzar un alto nivel de abstracción; pero a mayor nivel de abstracción se puede perder el seguimiento visual del flujo de trabajo dentro de la *CMF*. Por lo tanto la reducción máxima hasta que el modelo sea compatible con la arquitectura física del *SMF*.
6. Obtener el conjunto total de colores de los recursos con los que cuenta el *SMF*. Se le asignará un color a cada una de las partes que fluyan a través del sistema (los cuales serán los colores principales) y también un color a cada recurso. El tener esta sincronización entre la arquitectura física del *SMF* y su modelo permite omitir la asignación de colores a algunos sitios; pero cuando las partes necesiten de un lugar específico dentro del sitio, sin importar el orden o la regla de almacenamiento, es necesario dar un color a cada sitio de acuerdo a su capacidad. En base al conjunto total de colores se define el color de cada lugar y cada transición. Si una transición para ser disparada necesita el mismo color desde lugares diferentes, es necesario

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

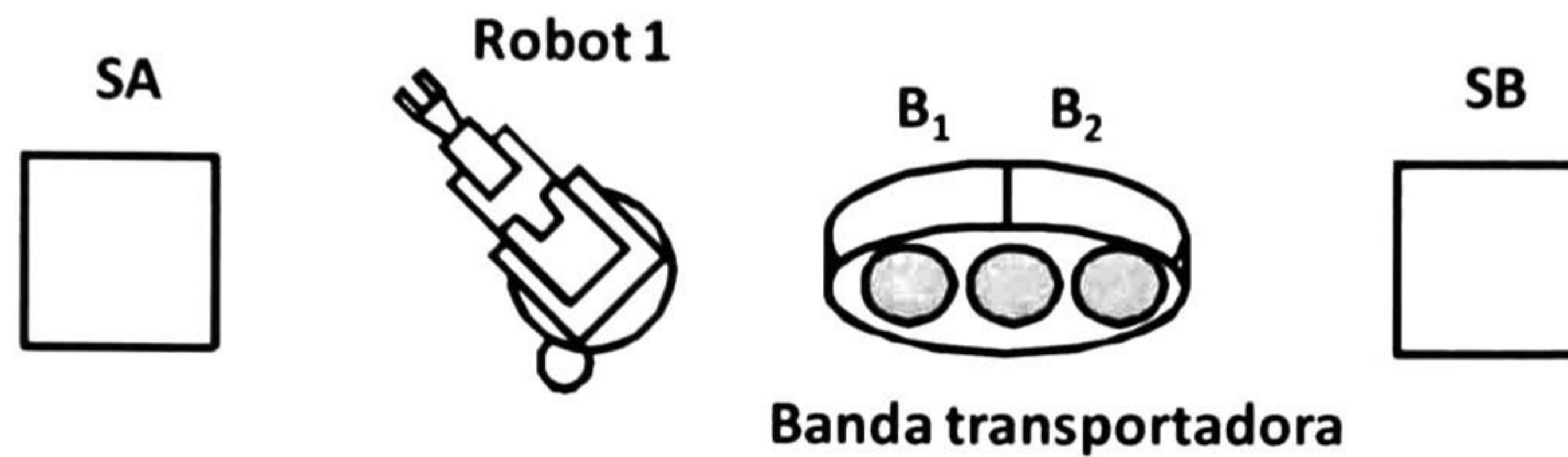


Figura 4.1: Identificación de elementos físicos en el sistema de transporte de partes con robot y banda.

redefinir el color de uno o más lugares con el fin de evitar confusiones.

7. Definir las funciones de cada arco, así como su dimensión, ya sea dentro del modelo del *SMF* o por separado.
8. Construir la matriz de incidencia del modelo en *RPC* del sistema y definir el marcado inicial de la red.

El resultado de aplicar la metodología de modelado propuesta es una *RPC* sincronizada con la arquitectura física del *SMF*, donde los circuitos principales por donde fluyen las partes son grafos marcados (sin tomar en cuenta los lugares redundantes inducidos por la función As_n) o una máquina de estados (si en la ruta de trabajo de las partes no existe ninguna operación de ensamble).

Ejemplo 19. Existe una celda compuesta por dos sitios de almacenamiento, un equipo de visión, un robot y una banda como se ilustra en la figura 4.1. La celda opera de la siguiente forma: Las partes son identificadas dentro del sitio A por el equipo de visión, posteriormente el robot toma las partes y las coloca sobre la banda transportadora, la cual se encarga de llevar las partes hasta el sitio B. Para no hacer un ejemplo exhaustivo se considerará que la banda solamente tiene dos lugares discretizados. Se supondrá que inicialmente hay cinco partes y los almacenes cuentan con una capacidad ilimitada.

1. Los elementos físicos que pueden ser identificados de la figura 4.1 son $E = \{S1, R1, S2, Banda, SV\}$ donde *SA* representa el sitio A, *SB* el sitio B, R_1 es el robot y *SV* simboliza el sistema de visión. Las partes del conjunto de

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

reconocimiento son A_1, A_2, A_3 y los sitios SA, SB y dos lugares discretizados de la banda B_1 y B_2 .

2. Las operaciones son las siguientes:

$$O_j^{R_1} = \{Tomar(SA, parte), Colocar(Banda, parte)\}, O_j^{Banda} = \{Avanzar\} \text{ y } O_j^{SV} = \{Reconocer(Partes)\}.$$

3. Al resumir la información anterior en tablas se obtiene:

Sitios	Operaciones
SA	Desalmacenar
SB	Almacenar
Banda (B_1 y B_2)	Avanzar

(a)

Efectores	Operaciones
Robot	Tomar parte de SA y después colocarla sobre la banda transportadora.
Banda	Avanzar (Mover partes de B_1 a B_2 y después de B_2 a SB)
Equipo de visión	Reconocimiento.

(b)

(4.2)

4. En la figura 4.2 se muestran los módulos correspondientes a cada operación.

5. Los módulos anteriores deben ser unidos de acuerdo a la secuencia indicada en la tabla 4.2 para obtener el modelo final expuesto en la figura 4.3.

6. El conjunto total de colores para este ejemplo es $\{A_i, B_j, R_1, \bullet\}$ para $i = 1, 2, 3$ y $j = 1, 2$. A continuación se definen los colores de cada lugar y transición.

$$C(p_1) = \{\langle \bullet \rangle, \langle A_1 \rangle, \langle A_2 \rangle, \langle A_3 \rangle\}$$

$$C(p_2) = \{\langle A_1, R_1 \rangle, \langle A_2, R_1 \rangle, \langle A_3, R_1 \rangle\}$$

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

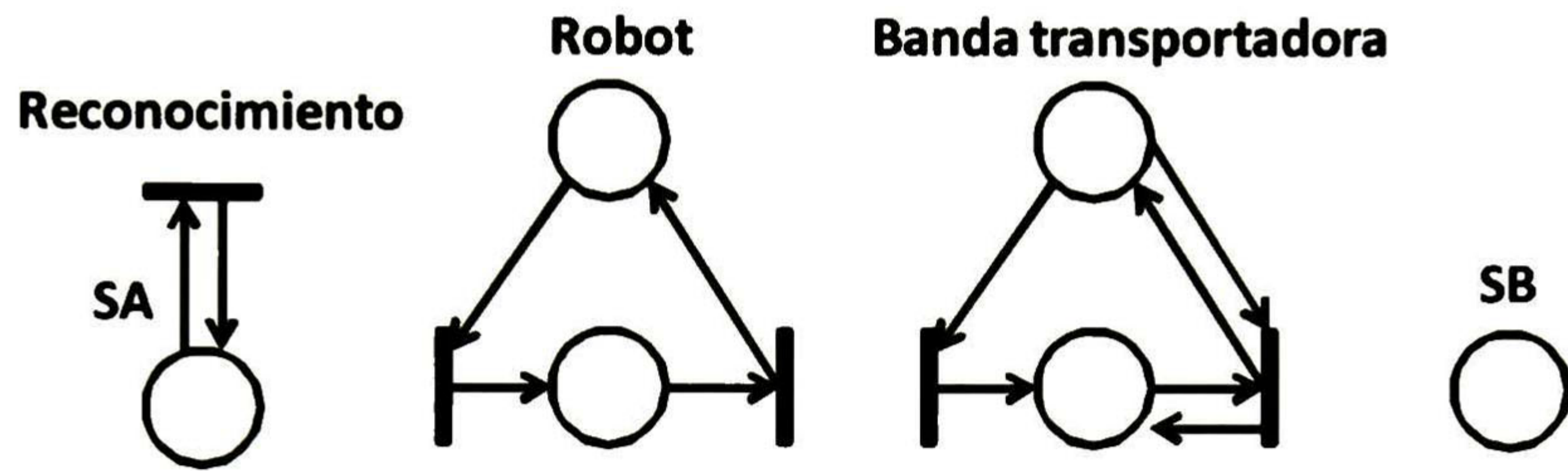


Figura 4.2: Módulos de las operaciones del sistema de transporte de partes.

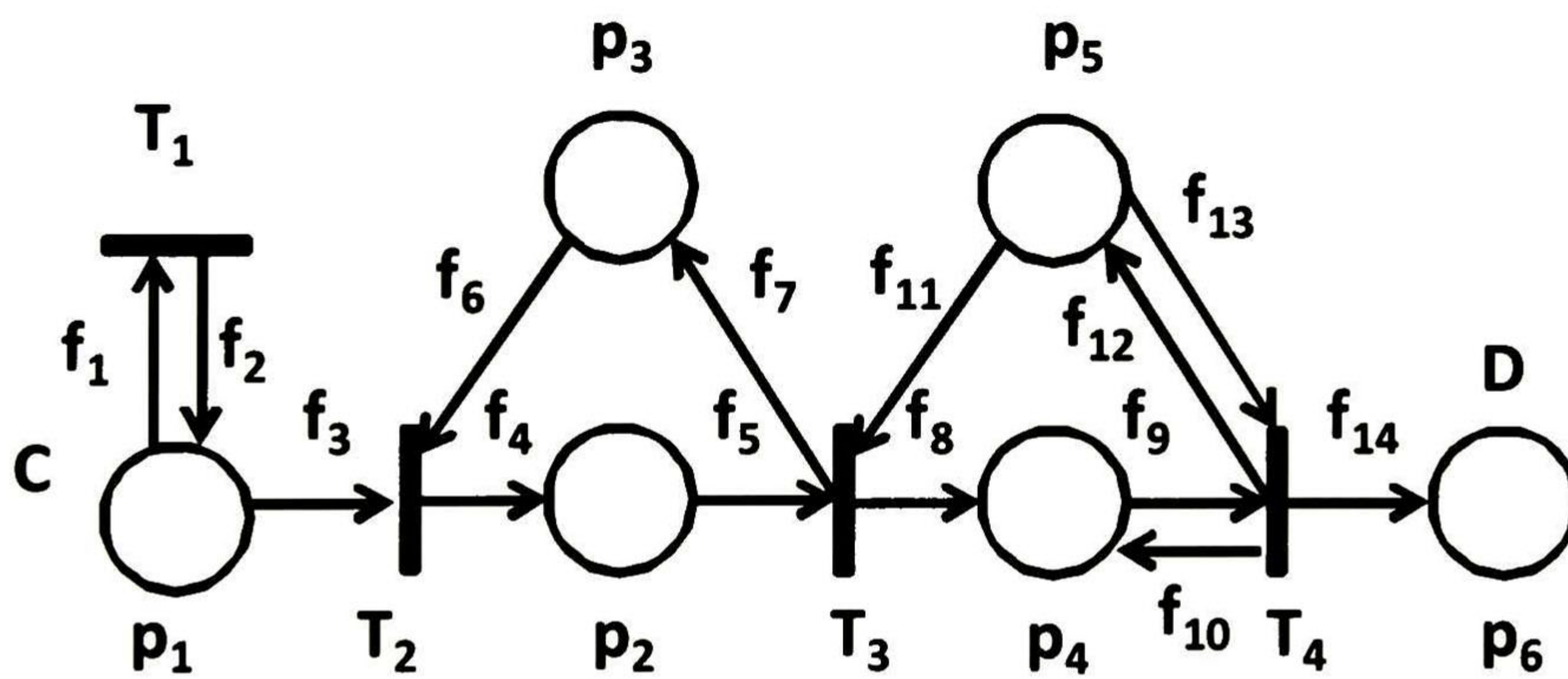


Figura 4.3: Modelo final en *RPC* del sistema de transporte de partes.

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

	T_1	T_2	T_3	T_4	
P_1	$-Asig_3$	0_3	0_3	0_3	0_3
	I_3	$-I_3$	0_3	0_3	0_3
P_2	0_3	I_3	$-I_3$	0_3	0_3
P_3	0_3	$-Asig_3$	$Asig_3$	0_3	0_3
P_4	0_3	0_3	I_3	$-I_3$	0_3
	0_3	0_3	0_3	I_3	$-I_3$
P_5	0_3	0_3	$-Asig_3$	$Asig_3$	0_3
	0_3	0_3	0_3	$-Asig_3$	$Asig_3$
P_6	0_3	0_3	0_3	0_3	I_3

Figura 4.4: Matriz de incidencia representación en funciones.

$$C(p_3) = \{\langle R_1 \rangle\}$$

$$C(p_4) = \{\langle A_1, B_1 \rangle, \langle A_2, B_1 \rangle, \langle A_3, B_1 \rangle, \langle A_1, B_2 \rangle, \langle A_2, B_2 \rangle, \langle A_3, B_2 \rangle\}$$

$$C(p_5) = \{\langle B_1 \rangle, \langle B_2 \rangle\}$$

$$C(p_6) = \{\langle A_1 \rangle, \langle A_2 \rangle, \langle A_3 \rangle\}$$

$$C(T_1) = \{\langle \bullet \rangle\}$$

$$C(T_2) = C(p_2)$$

$$C(T_3) = \{\langle A_1, R_1, B_1 \rangle, \langle A_2, R_1, B_1 \rangle, \langle A_3, R_1, B_1 \rangle\}$$

$$C(T_4) = \{\langle A_1, B_1, B_2 \rangle, \langle A_2, B_1, B_2 \rangle, \langle A_3, B_1, B_2 \rangle, \langle A_1, B_2 \rangle, \langle A_2, B_2 \rangle, \langle A_3, B_2 \rangle\}$$

7. Para este ejemplo las funciones asociadas al modelo son:

$$f_1 = f_6 = f_7 = f_{11} = f_{12} = f_{13} = Asign_3$$

$$f_2 = f_3 = f_4 = f_5 = f_8 = f_9 = f_{10} = f_{14} = I_3$$

8. La matriz de incidencia puede ser construida en base a las funciones (ver figura 4.4) o en forma numérica, como se expone en la figura 4.5. Asimismo en la figura 4.5 se muestra el marcado inicial del sistema.

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

Ejemplo 20. *Un SMF está integrado por un robot, un almacén de materia prima y un área de ensamble, el cual es mostrado en la figura 4.6. El robot toma las partes del almacén de materia prima y las coloca en alguno de los dos sitios de ensamble. El ensamble se realiza en forma vertical (por apilamiento de partes), por ende los distintos tipos de ensamble son obtenidos al cambiar el orden de apilamiento y la longitud de la pila en el área de ensamble. El ensamble requerido para este caso son las secuencias (A_1, A_2, A_3) o la (A_2, A_3, A_1) . Se supondrá que inicialmente existen dos partes de cada tipo.*

1. La identificación de los elementos físicos está representado en la figura 4.6 son $E = \{AMP, R_1, AE\}$ donde AMP es el almacén de materia prima, R_1 es el robot 1 y AE simboliza al área de ensamble. Las partes son A_1, A_2, A_3 y los sitios son AMP y AE .
2. En este ejemplo las operaciones son únicamente llevadas a cabo por el robot 1, $O_j^{R_1} = \{Tomar(RMS, parte), Colocar(AS, parte)\}$.
3. Resumiendo la información se tiene que:

Sitios	Operaciones	Efectores	Operaciones	(4.3)
AMP	Desalmacenar	Robot	Tomar una parte de AMP y entonces colocarla en AE	
AE	Ensamblar			
(a)		(b)		

4. La figura 4.7 muestra la sustitución de cada operación por su módulo respectivo.

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

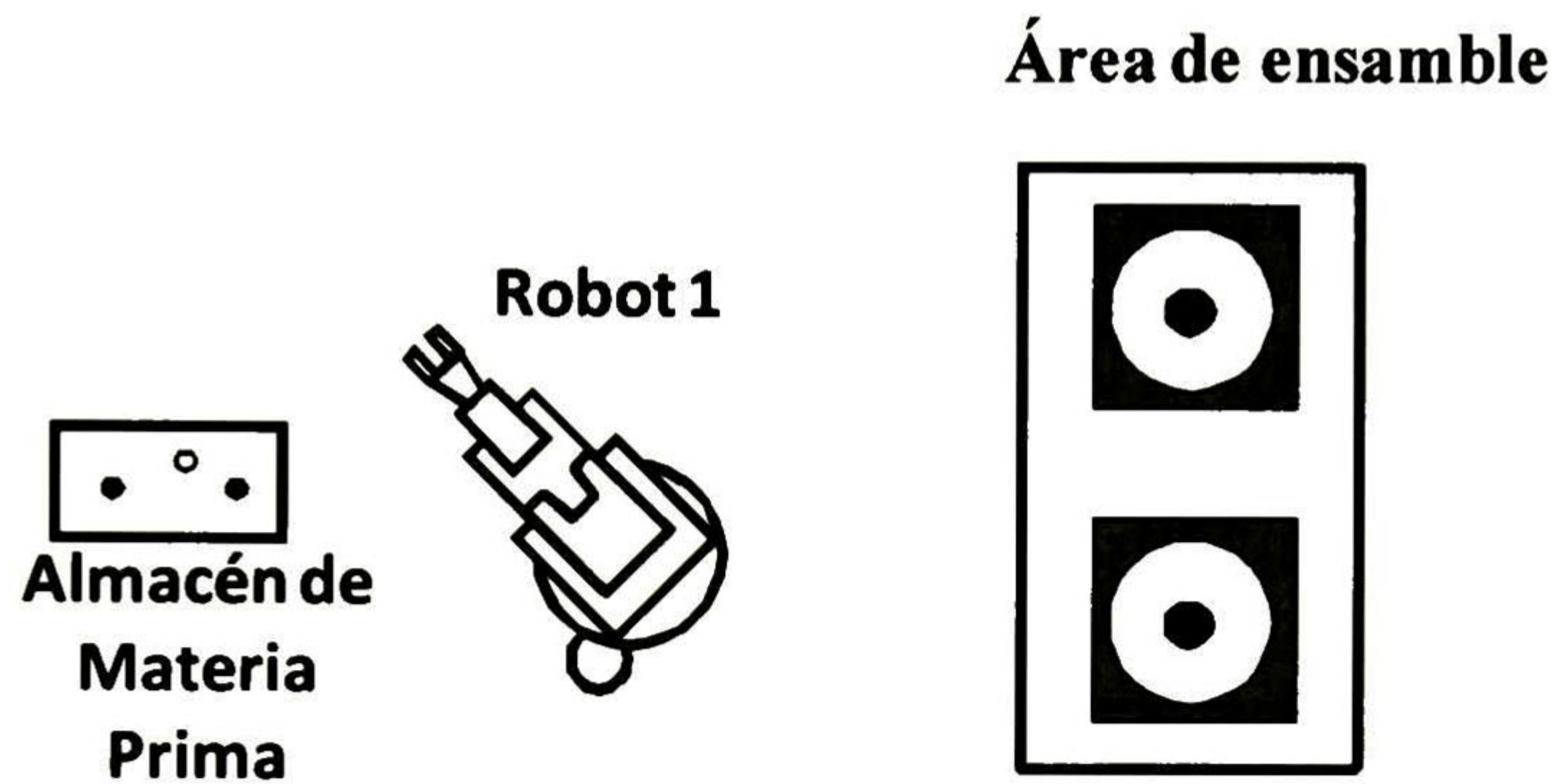


Figura 4.6: Identificación de elementos físicos del sistema de ensamble.

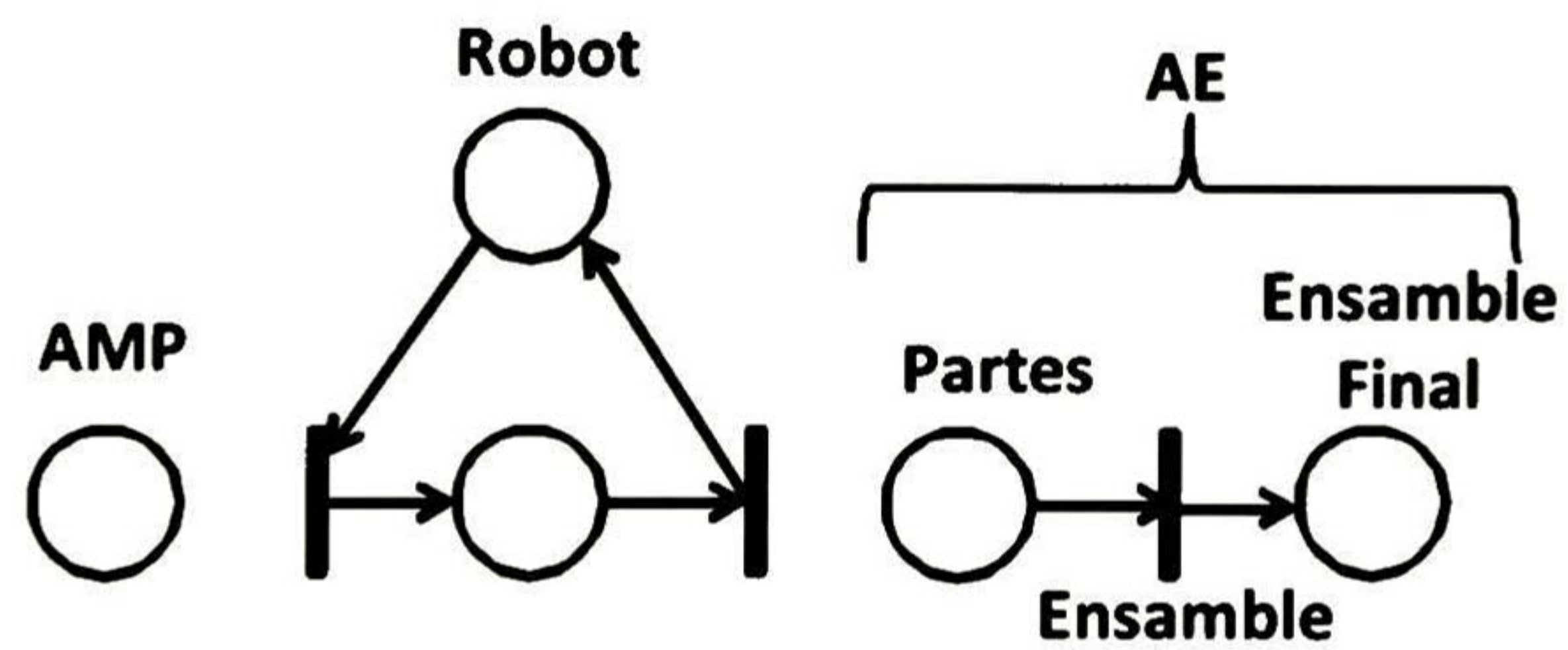


Figura 4.7: Módulos de operaciones y sitios del sistema de ensamble.

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

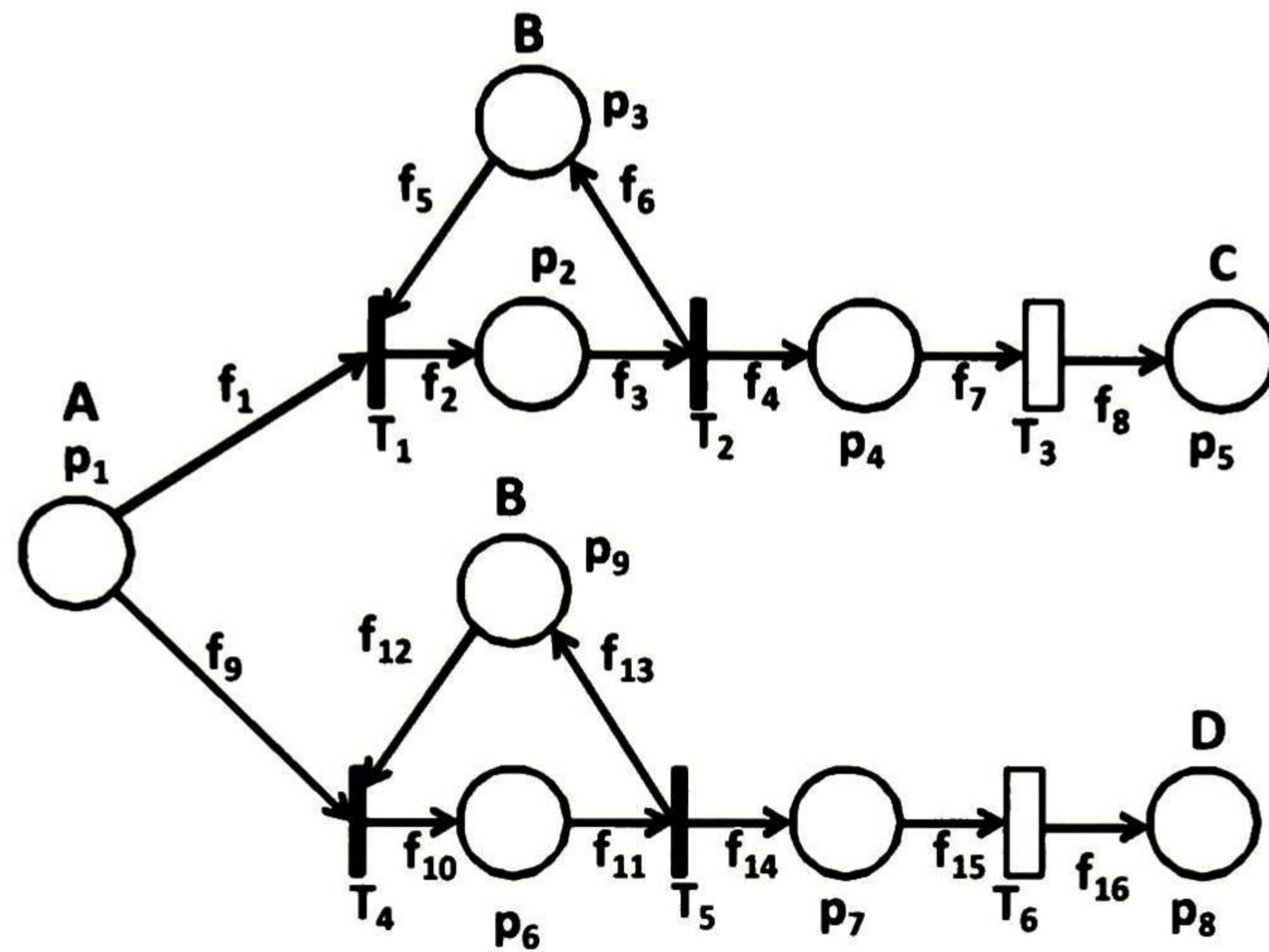


Figura 4.8: Modelo del *SMF* de ensamble con *RPC*

5. Usando la información contenida en la tabla 4.3 y con los módulos mostrados anteriormente es posible construir el modelo presentado en la figura 4.8. Este modelo se obtuvo sin utilizar directamente el módulo de ensamble para corroborar que hay diversas formas de obtener el módulo.

Como el lugar p_3 y p_9 representan al robot, estos pueden ser fusionados para así tener el modelo presentado en la figura 4.9, el cual está sincronizado con la arquitectura física del sistema.

En este caso se está considerando que las transiciones T_3 y T_6 son incontrolables debido a que una vez que se apilan tres partes queda completo un ensamble

6. Para este sistema el conjunto total de colores es $\{A_i, R_1, E_j\}$ para $i = 1, 2, 3$ y $j = 1, 2$ donde A_i denota a las partes, R al robot y E_j a los ensambles. Asignando colores a cada lugar y transición se obtienen los siguientes resul-

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

	T_1	T_2	T_3	T_4	T_5	T_6
P_1	$-I_3$	0_3	0_3	$-I_3$	0_3	0_3
P_2	I_3	$-I_3$	0_3	0_3	0_3	0_3
P_3	$-Asig_3$	$Asig_3$	0_3	$-Asig_3$	$Asig_3$	0_3
P_4	0_3	I_3	$-Asig_3$	0_3	0_3	0_3
P_5	000	000	1 1 1	000	000	000
P_6	0_3	0_3	0_3	I_3	$-I_3$	0_3
P_7	0_3	0_3	0_3	0_3	I_3	$-Asig_3$
P_8	000	000	000	000	000	1 1 1

Figura 4.10: Matriz de incidencia representada con funciones.

Analizando el marcado inicial del sistema, el robot está disponible y la secuencia de ensamble requerida puede ser desarrollada diseñando un controlador.

4.3. Modelado de la especificación.

Esta sección muestra la manera en que el modelo en *RPC* del sistema se puede confinar a un comportamiento requerido llamado especificación. La especificación puede ser una máquina de estados o un modelo construido a partir de los módulos propuestos, especialmente cuando se presenta una operación de ensamble se puede usar el modulo expuesto en la figura 3.15.b) que está compuesto por un conjunto de secuencias simples de operaciones con algunos arcos de prueba (representados por arcos bidireccionales). Los arcos de prueba detectan el estado actual del lugar (colores) y, dependiendo de su valor, algunos colores que habilitan a la transición pueden ser evitados. La figura 3.15.b) muestra cómo la materia prima es transformada en bienes o en un producto terminado (p_6). Cabe señalar que los lugares p_4 y p_5 de esta figura representan subensambles. Por lo tanto, la

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

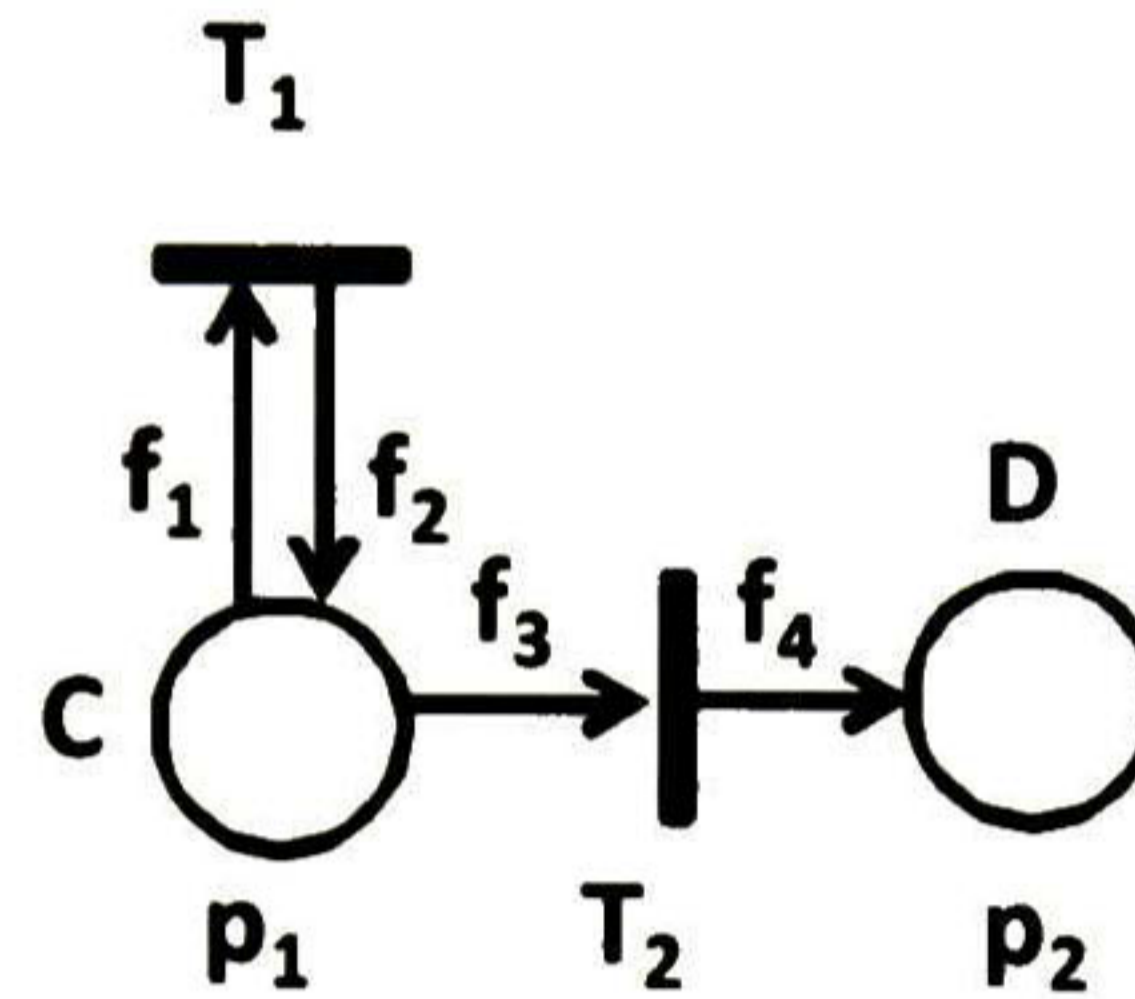


Figura 4.12: Modelo de la especificación para el sistema de transporte de partes.

materia prima requerida depende del subensamble a realizar, si el lugar p_4 es eliminado de la especificación, el controlador supervisor aun puede ser computado, sin embargo su computación será menos eficiente debido a que se necesitará una gran exploración del grafo de alcanzabilidad. Este hecho será explicado en el próximo capítulo en la sección del diseño del supervisor.

La máquina de estados mostrada en la figura 4.12 puede ser una especificación dada para el sistema del ejemplo 19.

El conjunto de colores para el modelo de la especificación mostrada en la figura 4.12 es:

$$C(p_1) = \{\langle \bullet \rangle, \langle A_1 \rangle, \langle A_2 \rangle, \langle A_3 \rangle\}$$

$$C(p_2) = \{\langle A_1 \rangle, \langle A_2 \rangle, \langle A_3 \rangle\}$$

$$C(T_1) = \{\langle \bullet \rangle\}$$

$$C(T_2) = C(p_2)$$

Para el ejemplo 20 uniendo dos módulos de ensamble se tiene como modelo de la especificación el que se aprecia en la figura 4.13.

Para el modelo de la especificación mostrado en la figura 4.13 los siguientes colores son definidos:

$$C(p_1) = \{\langle A_1 \rangle, \langle A_2 \rangle, \langle A_3 \rangle\} = C(p_5) = C(p_9)$$

$$C(p_2) = \{\langle A_1, R_1 \rangle, \langle A_2, R_1 \rangle, \langle A_3, R_1 \rangle\} = C(p_7)$$

$$C(p_3) = \{\langle R_1 \rangle\}$$

$$C(p_4) = \{\langle A'_1 \rangle, \langle A_1 A_2 \rangle, \langle E_1 \rangle\}$$

$$C(p_6) = \{\langle E_1 \rangle\}$$

$$C(p_8) = \{\langle E_2 \rangle, \langle A'_2 \rangle, \langle A_2 A_3 \rangle\}$$

$$C(p_{10}) = \{\langle E_2 \rangle\}$$

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

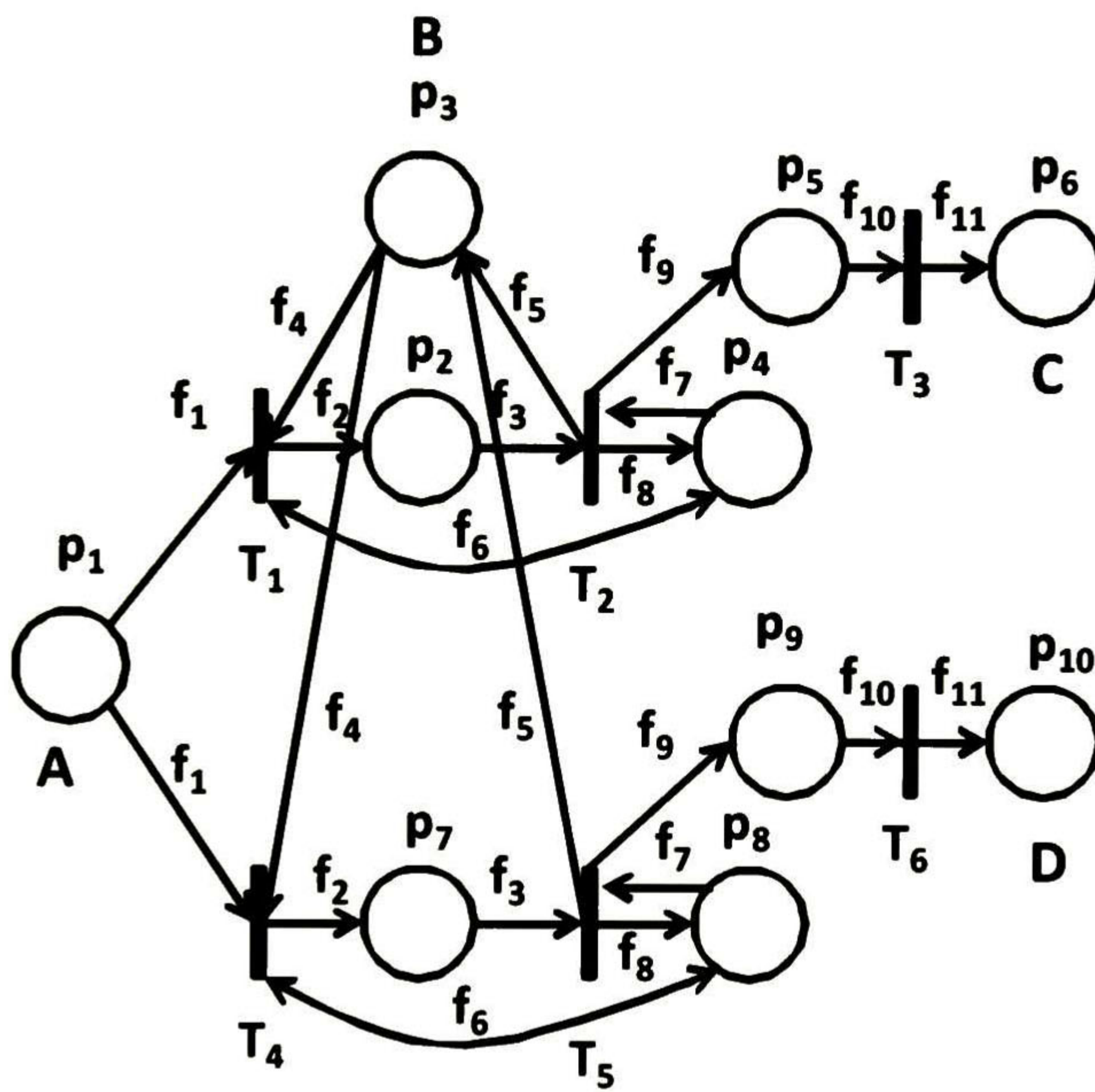


Figura 4.13: Modelo de la especificación del *SMF* de ensamblaje

CAPÍTULO 4. METODOLOGÍA DE MODELADO PARA SMF

$$C(T_1) = \{\langle A_1, R_1, A_1A_2, E_1 \rangle, \langle A_2, R_1, A'_1 \rangle, \langle A_3, R_1, 2A_1A_2 \rangle\}$$

$$C(T_2) = \{\langle A_1, R_1, A_1A_2 \rangle, \langle A_2, R_1, A'_1, E_1 \rangle, \langle A_3, R_1, A_1A_2 \rangle\}$$

$$C(T_3) = \{\langle A_1, A_2, A_3 \rangle\}$$

$$C(T_4) = \{\langle A_1, R_1, A'_2, A_2A_3 \rangle, \langle A_2, R_1, E_2 \rangle, \langle A_3, R_1, 2A'_2 \rangle\}$$

$$C(T_5) = \{\langle A_1, R_1, A'_2 \rangle, \langle A_2, R_1, A_2A_3, E_2 \rangle, \langle A_3, R_1, A'_2 \rangle\}$$

$$C(T_6) = \{\langle A_1, A_2, A_3 \rangle\}$$

4.4. Conclusiones.

En este capítulo se presentó una metodología de modelado ascendente para modelar *SMF* con *RPC*. Al aplicar la metodología de modelado se obtienen modelos compactos sincronizados con la distribución física del sistema, lo cuales capturan el flujo de las partes a través del *SMF*. También las matrices de incidencia pueden ser reducidas de tamaño si son presentadas en forma de funciones, dicha forma es la que se utilizará para resolver las ecuaciones implicadas en el diseño del control supervisor más rápidamente. La especificación, para confinar el comportamiento del modelo del *SMF*, puede ser construida a través de una máquina de estados con *RPC* o por medio de los módulos propuestos.

5. CONTROL POR REGULACIÓN

Resumen: en este capítulo se presentan los conceptos de equivalencia en trayectoria e igualdad a la salida que son utilizados en el control por regulación. Asimismo, se define el problema de control por regulación, el cual consiste en el diseño de un controlador que permita que la salida de un sistema y una especificación dados, sean iguales. En nuestro caso diremos que es confinar el comportamiento del SMF al comportamiento de la especificación cuando ambos son modelados con RPC. Además se presenta el conjunto de condiciones para resolver el problema de control por regulación, caracterización que se basa en la existencia una función que permita relacionar los estados del sistema con los de la especificación. También se propone un algoritmo para computar la función π , la cual permite empotrar el comportamiento de la especificación dentro del modelo del SMF, y otro algoritmo para probar la controlabilidad de los marcados alcanzables.

5.1. Introducción.

El problema de control para *SED* está fundamentado en la idea de cómo garantizar que el comportamiento del sistema logre ser confinado a un comportamiento requerido, y también cómo diseñar un controlador para llevar a cabo tal labor. En los *SED's*, como lo son los *SMF*, el primer paso es la obtención de la representación del sistema y de la especificación. Posteriormente, se debe comprobar que el comportamiento del sistema pueda ser confinado al comportamiento de la especificación, y en caso de ser posible se diseña el controlador. Desafortunadamente, cuando los *SMF* se modelan con *RP*, el verificar si una representación puede ser confinada al comportamiento de una especificación llega a ser complicado por el tamaño del modelo y la cantidad de operaciones matemáticas que se necesitan para lograrlo. Con el fin de reducir el tamaño del modelo del sistema y de la especificación esta tesis empleó las *RPC*. Por tanto el tema de este capítulo es proponer un algoritmo con el cual sea posible encontrar fácilmente una transformación lineal que permita confinar el comportamiento del sistema al de una

CAPÍTULO 5. CONTROL POR REGULACIÓN

especificación.

Este capítulo está ordenado de la siguiente manera: En la sección 5.2 se introducen los conceptos de equivalencia en trayectoria e igualación a la salida necesarios para resolver el problema del control por regulación y así hacer equivalentes el espacio de estados del modelo en RPC del SMF y el modelo de su especificación. Después en la sección 5.3 se da el conjunto de condiciones para resolver el problema del control por regulación. También es propuesto un algoritmo para computar π y \vec{w} , y un algoritmo para verificar la controlabilidad de los marcados alcanzables. Por último, las conclusiones son presentadas en la sección 5.4.

5.2. RPC equivalentes en trayectoria.

En la figura 5.1 se muestra el conjunto de alcanzabilidad de dos RPC , (S_f, M_0) y (R_m, M'_0) que representan el modelo del sistema y de la especificación, respectivamente. Se puede observar que existe una función π que relaciona los marcados de $R(S_f, M_0)$, de forma que dos marcados consecutivos, $M'_i \xrightarrow{t'_i} M'_j$ en $R(R_m, M'_0)$ son transformados en dos marcados $\pi(M'_i)$ y $\pi(M'_j)$ que pertenecen a $R(S_f, M_0)$ y $R(R_m, M'_0)$, y además existe una secuencia de disparo de transiciones con respecto a un color $w \in \mathcal{L}(S_f, \pi(M'_i))$ que satisface $\pi(M'_i) \xrightarrow{w} \pi(M'_j)$. Cuando dicha función π existe, entonces se dice que (S_f, M_0) es equivalente en trayectoria con respecto a (R_m, M'_0) . A continuación se define formalmente este concepto.

Definición 21. Sea (Q, M_0) una RPC y $R(Q, M_0)$ su grafo de alcanzabilidad. $\forall M_i, M_j \in R(Q, M_0)$, M_i está $T(Q, M_0)$ – relacionado con M_j , denotado por $M_i \sim_{T(Q, M_0)} M_j$, si existe una transición $t_q \in T$ (el conjunto de transiciones de (Q, M_0)) tal que $M_i \xrightarrow{t_q} M_j$.

Si existen transiciones $t_i, t_j \in T$ con respecto a un color tal que $M_i \xrightarrow{t_i} \dots \xrightarrow{t_j} M_j$, se dice que M_i está $T(Q, M_0)^*$ – relacionado con M_j . Esto se denota por $M_i \sim_{T(Q, M_0)^*} M_j$

Definición 22. Sean (S_f, M_0) y (R_m, M_0) dos RPC y $R(S_f, M_0)$, $R(R_m, M_0)$ sus grafos de alcanzabilidad, respectivamente. Se dice que (S_f, M_0) es equivalente en trayectoria con respecto a (R_m, M_0) si existe una función $\pi : (R_m, M_0) \rightarrow R(S_f, M_0)$ tal que:

$$\sim_{T(S_f, M_0)^*} \circ \pi = \pi \circ \sim_{T(R_m, M'_0)} \quad (5.1)$$

CAPÍTULO 5. CONTROL POR REGULACIÓN

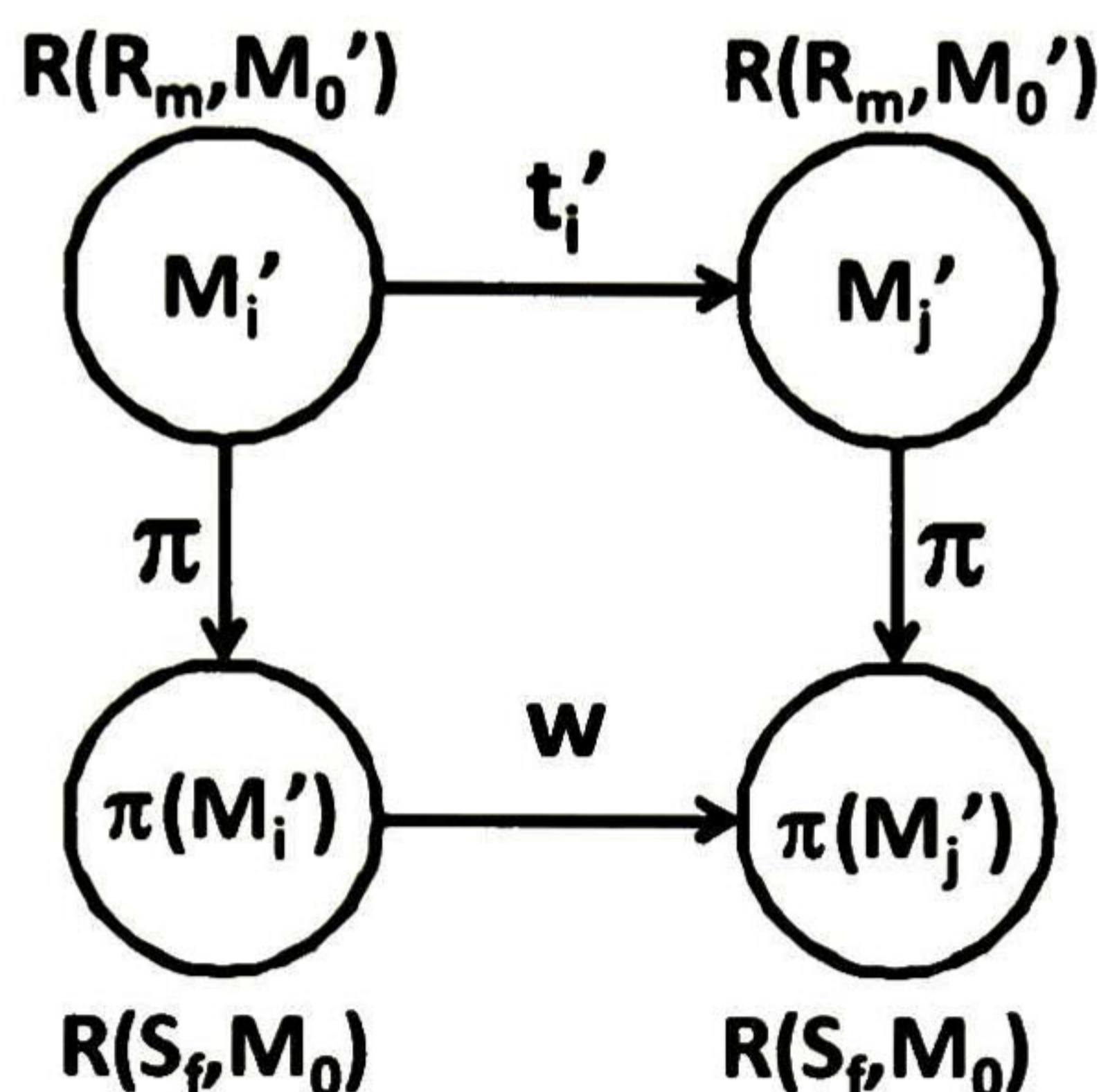


Figura 5.1: Grafos de alcanzabilidad del modelo del sistema y especificación relacionados por la función π .

La ecuación 5.1 hace que el diagrama de la figura 5.2 conmute. Un lector interesado en profundizar en el tema de relaciones puede consultar [2].

Para llevar a cabo la regulación también se requiere garantizar que la salida de ambos modelos, sistema y especificación, sean las mismas en todo instante. Por tal motivo, es necesario agregar la condición $\varphi_{R_m}(M_i') = \varphi_{S_f}(M_i)$ cuando $\pi(M_i') = M_i$ al concepto de equivalencia en trayectoria. Esta condición indica que el lenguaje de salida de ambas *RPC* debe ser igual. No obstante, aún falta asegurar que el sistema pueda ser confinado a únicamente producir estas señales de salida, es decir falta probar la propiedad de controlabilidad. A continuación se presenta formalmente el concepto de igualdad a la salida.

Definición 23. Sean (S_f, M_0) y (R_m, M_0) dos *RPC*, y $R(S_f, M_0)$, $R(R_m, M_0)$ sus conjuntos de alcanzabilidad, respectivamente. (S_f, M_0) es equivalente en la trayectoria de salida con respecto a (R_m, M_0) si (S_f, M_0) es equivalente en trayectoria con respecto a (R_m, M_0) y $\forall M_j' \in R(R_m, M_0)$, $\varphi_{R_m}(M_j') = \varphi_{S_f}(\pi(M_j'))$.

Proposición 24. Sean (S_f, M_0) y (R_m, M_0') dos *RPC* representadas por las ecuaciones de estado

CAPÍTULO 5. CONTROL POR REGULACIÓN

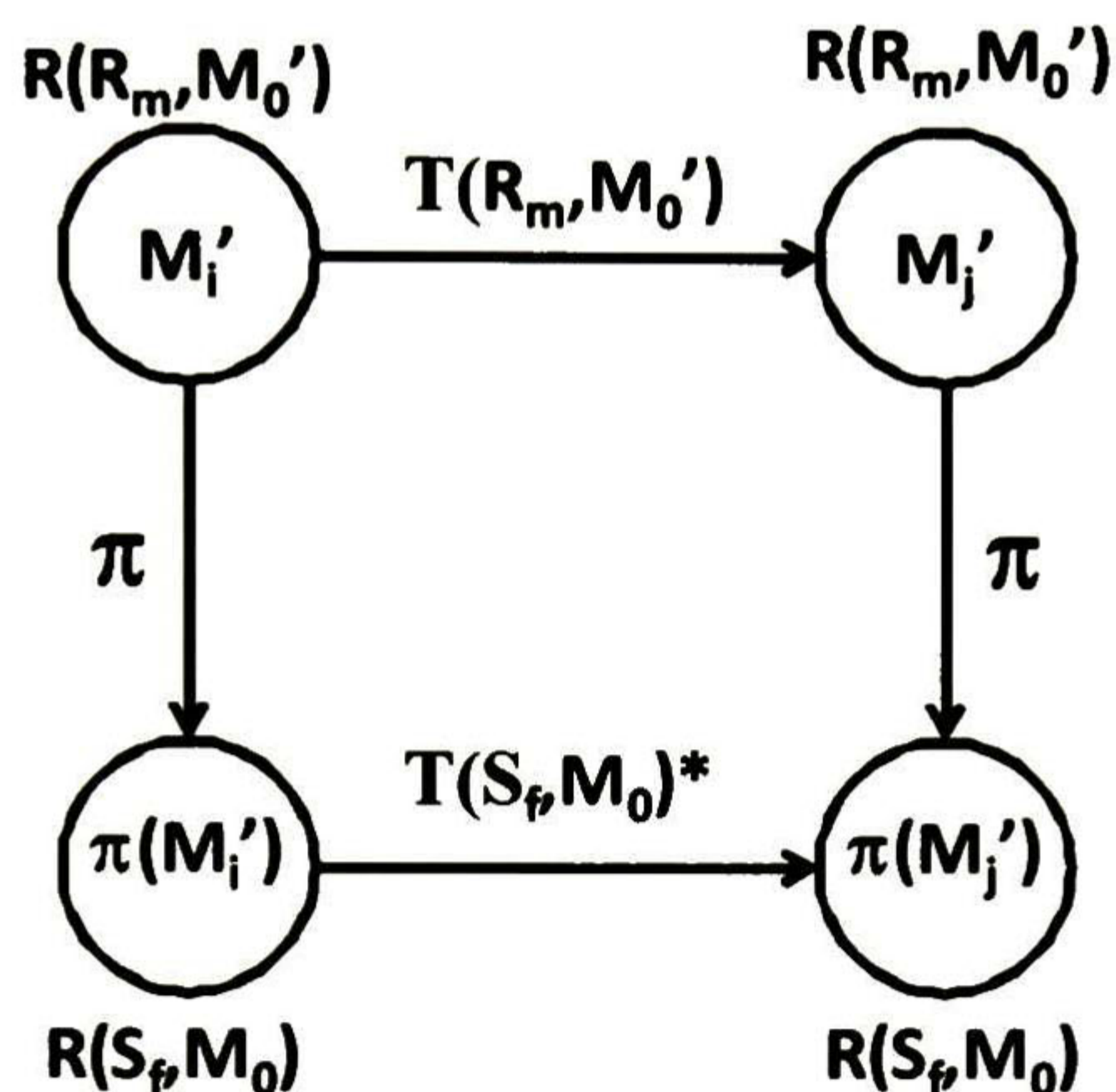


Figura 5.2: Equivalencia en trayectoria de grafos de alcanzabilidad.

$$\begin{aligned}
 S_f &= \begin{cases} M_{k+1} = M_k + A \vec{v}_k \\ y_k = \varphi_{S_f}(M_k) \end{cases} \\
 R_m &= \begin{cases} M'_{k+1} = M'_k + A' \vec{z}_k \\ y'_k = \varphi_{R_m}(M'_k) = M'_k \end{cases}
 \end{aligned} \tag{5.2}$$

respectivamente. (S_f, M_0) es equivalente en trayectoria de salida con respecto a (R_m, M'_0) si y sólo si:

- a) $\exists M_0 \in R(S_f, M_0)$ tal que $\pi(M'_0) = M_0$.
- b) $\forall M'_{j-1} \in R(R_m, M_0)$ y $\forall t'_k \in T'$ habilitados en M'_{j-1} , $\exists w_k \in \mathcal{L}(S_f, \pi(M'_{j-1}))$ tal que $\pi(M'_{j-1} + A' \vec{t}_j) = \pi(M'_{j-1}) + A \vec{w}_k$.
- c) $\forall M'_j \in R(R_m, M'_0)$, $\varphi_{R_m}(M'_j) = \varphi_{S_f} \circ \pi(M'_j)$.

Demostración. Necesidad. Como S_f es equivalente en trayectoria a la salida con respecto a R_m , entonces $\forall M'_j \in R(R_m, M_0)$, $\pi(M'_j) \in R(S_f, M_0)$. En consecuencia existe $M_0 \in R(S_f, M_0)$ tal que

CAPÍTULO 5. CONTROL POR REGULACIÓN

$$\pi(M'_0) = M_0 \quad (5.3)$$

Además, equivalente en trayectoria a la salida implica que $\forall t'_k \in T'$ tal que $M'_{j-1} \xrightarrow{t'_k} M'_j$ entonces $\pi(M'_{j-1}) \xrightarrow{w_k} \pi(M'_j)$ y $\varphi'(M'_j) = \varphi_{S_f} \pi(M'_j)$. En términos de la ecuación de estados:

$\forall M'_{j-1} \in R(R_m, M'_0)$, $\forall t'_k \in T'$ habilitado en M'_{j-1} , $\exists w_k \in \mathcal{L}^{mid}(S_f, M_0)$ tal que

$$M'_{j-1} + A' \vec{t}_j = M'_j \text{ entonces } \pi(M'_j) = \pi(M'_{j-1}) + A \vec{w}_k \quad (5.4)$$

así

$$\pi(M'_j) = \pi(M'_{j-1} + A' \vec{t}_j) \quad (5.5)$$

por lo que

$\forall M'_{j-1} \in R(R_m, M'_0)$, $\forall t'_k \in T'$ habilitado con respecto a algún color en M'_{j-1} , $\exists w_k \in \mathcal{L}(S_f, \pi(M'_{j-1}))$ tal que

$$\pi(M'_{j-1} + A' \vec{t}_j) = \pi(M'_{j-1}) + A \vec{w}_k \quad (5.6)$$

La definición de equivalente en trayectoria a la salida implica que:

$$\forall M'_j \in R(R_m, M'_0), \varphi_{R_m}(M'_j) = \varphi_{S_f} \circ \pi(M'_j) \quad (5.7)$$

Suficiencia. La condición a) establece que:

$$\pi(M'_0) = M_0 \text{ y } M_0 \in R(S_f, M_0) \quad (5.8)$$

Usando inducción en la secuencia de disparos $\sigma \in \mathcal{L}(R_m, M'_0)$ se tiene que:

• $|\sigma| = 1$. Entonces $M'_0 \xrightarrow{t'_a} M'_1$, así la condición b) mantiene que:

$$\pi(M'_0 + A' \vec{t}_a) = M_0 + A \vec{w}_a \quad (5.9)$$

y $w_a \in \mathcal{L}(S_f, M_0)$, por lo que $(M_0 + A \vec{w}_a) = M_1 \in R(S_f, M_0)$. Además, como

CAPÍTULO 5. CONTROL POR REGULACIÓN

$M'_0 + A'\vec{t}_a = M'_1$ y $M_0 + A\vec{w}_a = M_1$, entonces $\pi(M'_1) = M_1$. En este caso:

$$(M'_0, M'_1) \in \sim_{T(R_m, M'_0)} \text{ y } (M_0, M_1) \in \sim_{T(S_f, M_0)^*} \quad (5.10)$$

así,

$$\pi \circ \sim_{T(R_m, M'_0)} (M'_0) = \pi(M'_1) = \sim_{T(S_f, M_0)^*} \circ \pi(M'_0) \quad (5.11)$$

• $|\sigma| = k$. Entonces $M'_{k-1} \xrightarrow{t'_i} M'_k$, se sigue que:

$$\pi \circ \sim_{T(R_m, M'_0)} (M'_{k-1}) = \pi(M'_k) = \sim_{T(S_f, M_0)^*} \circ \pi(M'_{k-1}) \quad (5.12)$$

• $|\sigma| = k + 1$. Entonces $M'_k \xrightarrow{t'_j} M'_{k+1}$, así la condición *b*) mantiene que:

$$\pi(M'_k + A'\vec{t}_j) = M_k + A\vec{w}_j$$

y $w_j \in \mathcal{L}^{mid}(S_f, \pi(M'_k))$, de esta forma $(M_k + A\vec{w}_j) = M_{k+1} \in R(S_f, M_0)$. Además, como $M'_k + A'\vec{t}_j = M'_{k+1}$ y $M_k + A\vec{w}_j = M_{k+1}$, entonces $\pi(M'_{k+1}) = M_{k+1}$. En este caso:

$$(M'_k, M'_{k+1}) \in \sim_{T(R_m, M'_0)} \text{ y } (M_k, M_{k+1}) \in \sim_{T(S_f, M_0)^*} \quad (5.13)$$

así,

$$\pi \circ \sim_{T(R_m, M'_0)} (M'_k) = \pi(M'_{k+1}) = \sim_{T(S_f, M_0)^*} \circ \pi(M'_k) \quad (5.14)$$

De esta manera,

$$\sim_{T(S_f, M_0)^*} \circ \pi = \pi \circ \sim_{T(R_m, M'_0)} \quad (5.15)$$

conmuta para todas las secuencias disparables en $\mathcal{L}(R_m, M'_0)$, además conmuta para todos los marcados alcanzables en $R(R_m, M'_0)$, así S_f es equivalente en trayectoria con respecto a R_m . También, la condición *c*) indica que S_f es equivalente en trayectoria a la salida con respecto a R_m .

■

CAPÍTULO 5. CONTROL POR REGULACIÓN

La proposición 24 no permite tener una caracterización estructural de equivalencia en la trayectoria de salida, debido a que en la condición *b*) es necesario comprobar que $w_k \in \mathcal{L}^{mid}(S_f, M_0)$. Es decir, se debe comprobar que:

$$\forall t'_k \in T' \text{ tal que } M'_{j-1} \xrightarrow{t'_k} M'_j \text{ entonces } \pi(M'_{j-1}) \xrightarrow{w_k} \pi(M'_j)$$

Sin embargo, es posible utilizar la estrategia ya conocida de divide y vencerás en la que, al saber que:

$$\pi(M'_{j-1} + A'\vec{t}_j) = \pi(M'_{j-1}) + A\vec{w}_k$$

se busca una secuencia de disparos para \vec{w}_k , la cual puede ser implementada usando una expansión parcial del grafo de alcanzabilidad de $R(S_f, M_0)$, de $\pi(M'_{j-1})$ a $\pi(M'_j)$.

5.3. Caracterización de la controlabilidad.

El disparo de una transición con respecto a un color puede ser controlable o incontrolable. Una transición se dice ser controlable si el disparo puede ser evitado e incontrolable en caso contrario. Una *RPC* es controlable en el mercado M_k si el disparo, con respecto a un color, de las transiciones que no son requeridas puede ser evitado.

Para verificar cual mercado debe ser alcanzado por el modelo en *RPC* del *SMF*, la equivalencia en trayectoria mostrada en la figura 5.2 tiene que ser considerada. Si cada mercado M'_i de la especificación es mapeado en un mercado $\pi(M'_i)$ del modelo en *RPC* del *SMF*, y cuando la especificación evoluciona de $M'_i \xrightarrow{t'_i} M'_j$ cuando t'_i es disparado con respecto a un color, entonces existe una secuencia de disparos $t_a \dots t_q$ con respecto a un color en el modelo en *RPC* del *SMF* tal que $\pi(M'_i) \xrightarrow{t_a} \dots \xrightarrow{t_q} \pi(M'_j)$ y además todos los mercados $\pi(M'_i) \dots \pi(M'_j)$ son controlables, entonces el comportamiento del *SMF* puede ser confinado dentro del comportamiento de la especificación. Por otra parte, si M'_i, M'_j, \dots representa los mismos subproductos que $\pi(M'_i), \pi(M'_j), \dots$ entonces el *SMF* está produciendo los mismos productos indicados por la especificación.

Proposición 25. Sean (S_f, M_0) y (R_m, M'_0) dos *RPC* representando el sistema y la especificación, respectivamente. Existe un supervisor tal que el modelo en *RPC* del *SMF* produce los mismos subproductos indicados por la especificación si existe una función lineal $\pi : R(R_m, M'_0) \rightarrow R(S_f, M_0)$, tal que:

1. $\pi M'_0 = M_0$

CAPÍTULO 5. CONTROL POR REGULACIÓN

2. para cada transición disparada t' con respecto a un color en la especificación $M'_i \xrightarrow{t'} M'_j$ entonces existe una secuencia de disparos $t_a \dots t_q$ con respecto a un color en el modelo en RPC del SMF tal que $\pi(M'_i) \xrightarrow{t_a} \dots \xrightarrow{t_q} \pi(M'_j)$ y todos los marcados $\pi(M'_i), \dots, \pi(M'_j)$ son controlables.
3. $\varphi' = \varphi\pi$, donde φ', φ son las funciones que mapean la especificación y el modelo del SMF en RPC en subproductos del SMF.

La prueba es similar a la presentada en [30] y aunque las condiciones 1 y 3 se resuelven algebraicamente, la condición 2 necesita de la expansión del grafo de alcanzabilidad. Afortunadamente, sólo es necesario realizar expansiones parciales, desde $\pi(M'_i)$ a $\pi(M'_j)$, donde los marcados M'_i, M'_j provienen de una especificación (máquina de estados) cuyo grafo de alcanzabilidad es fácil de obtener.

El problema de control por regulación con información completa es propuesto en la siguiente definición.

Definición 26. Sea (S_f, M_0) la RPC del sistema a controlar, sea (R_m, M'_0) el modelo en RPC de la especificación, donde la función π hace que (S_f, M_0) sea equivalente en trayectoria de salida con respecto a (R_m, M'_0) . Los conjuntos $R(S_f, M_0)$ y $R(R_m, M'_0)$ son los conjuntos de alcanzabilidad de (S_f, M_0) y (R_m, M'_0) respectivamente y las ecuaciones (5.2) son las ecuaciones de estado de (S_f, M_0) y (R_m, M'_0) respectivamente.

El problema de control por regulación con información completa consiste en encontrar una función parcial (controlador)

$$H : R(S_f, M_0) \times R(R_m, M'_0) \times T' \rightarrow \mathcal{L}^{mid}(S_f, M_0)$$

donde $H(\pi(M'_i), M'_i, t'_k) = w_k$ y w_k es controlable en $(S_f, \pi(M'_i))$.

La figura 5.3 muestra un esquema donde se implementa el control por regulación con información completa. La acción de control para (S_f, M_0) es calculada por un controlador el cual procesa la información recibida de la planta y la especificación para generar la señal de entrada adecuada en el sistema. Desde el punto de vista del problema de regulación, es necesario que la ley de control mantenga la salida de ambos sistemas iguales (es decir $y_k = y'_k$).

5.3.1. Diseño del supervisor.

El diseño del supervisor es llevado a cabo en dos etapas, en la primera el conjunto de ecuaciones

CAPÍTULO 5. CONTROL POR REGULACIÓN

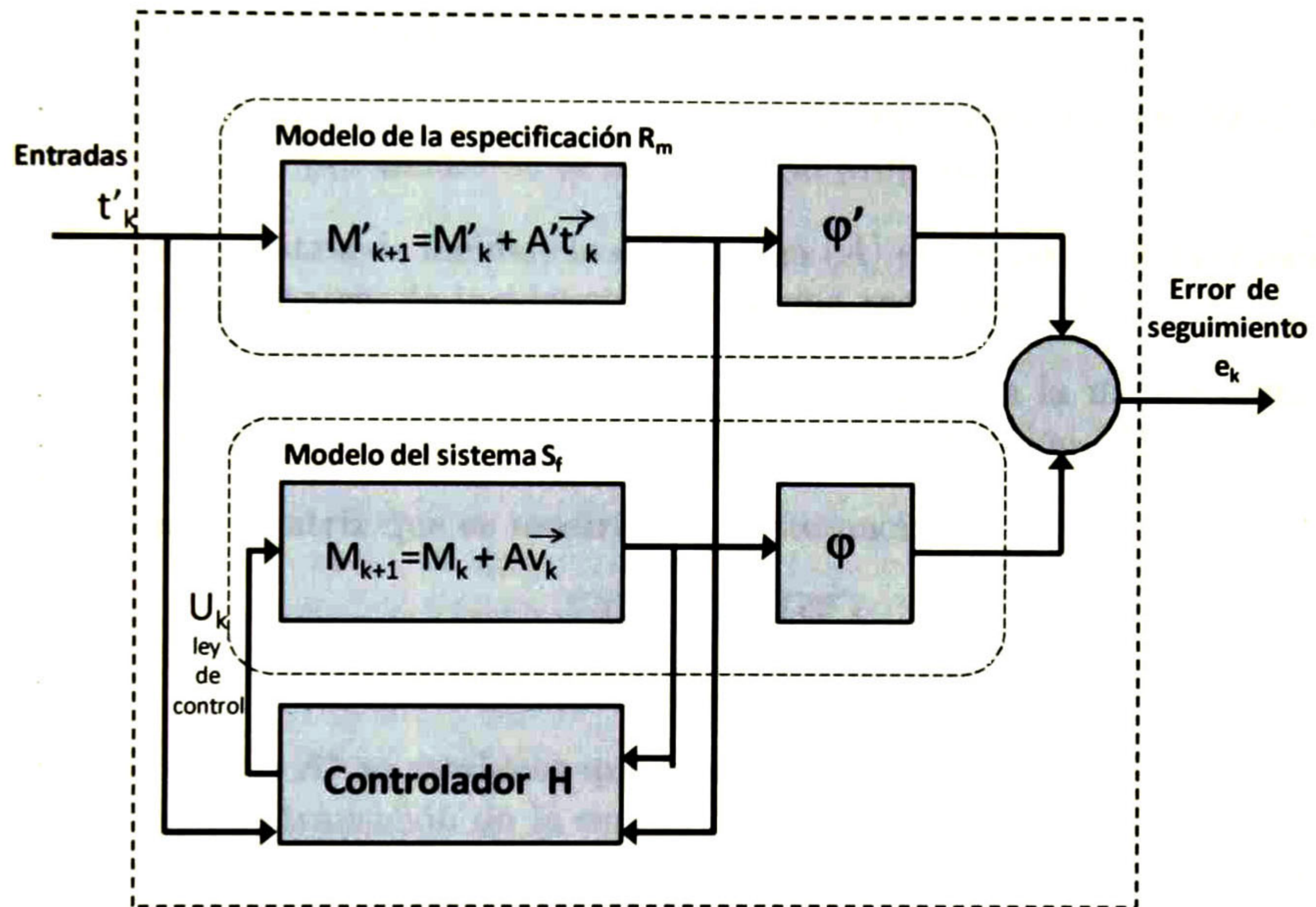


Figura 5.3: Esquema de control por regulación.

CAPÍTULO 5. CONTROL POR REGULACIÓN

$$\begin{aligned} \pi M'_0 &= M_0 \\ \forall t'_j, \pi A' t'_j &= A \vec{w}_j \\ \varphi' &= \varphi \pi \end{aligned} \tag{5.16}$$

de la proposición 25 tiene que ser resuelto, entonces en la segunda etapa se tiene que probar la controlabilidad de los marcados alcanzados cuando se disparan las secuencias $w_j = t_a \dots t_q$.

Algoritmo para el cálculo de π y \vec{w}_j

1. Definir el modelo en *RPC* del sistema y la representación en *RPC* de la especificación, por medio de la metodología propuesta.
2. Obtener la matriz de incidencia del sistema (A) en términos de las funciones y obtener la matriz de incidencia del sistema requerido (A').
3. Calcular el marcado inicial (M_0) para el sistema, con la metodología propuesta, y obtener el marcado inicial para la especificación (M'_0).
4. Elaborar la matriz que es mostrada a continuación

$$\begin{array}{|c|c|c|} \hline M'_0{}^T & & M_0{}^T \\ \hline A'^T & & A^*{}^T \\ \hline \end{array} \tag{5.17}$$

Con la matriz A^* se establece qué transiciones del sistema (\vec{w}_j) son equivalentes a cada transición de la especificación, en base a los lugares medibles por los sensores. Por ejemplo, si la transición t'_1 tiene que ser disparada para ir del lugar medible por el sensor A al lugar medible B en el modelo de la especificación, es equivalente a disparar una secuencia de transiciones en el modelo del sistema para realizar la misma acción; pero esta secuencia es fácil de encontrar debido a que cuando se construyó el modelo de la especificación se hizo mediante una máquina de estados o con los módulos propuestos en la metodología de modelado, en los cuales cada parte tiene su propia ruta de trabajo. Por ello, si el disparo de la secuencia de transiciones t_1, t_2, \dots, t_m es equivalente al disparo de t'_1 (para ir del lugar medible por el sensor A al lugar medible B), entonces la primera columna de la matriz A^* (primer renglón de $A^*{}^T$) será igual a $\sum_{i=1}^m t_i$ para que se satisfaga la segunda condición de la ecuación (5.16). Cabe señalar que las transiciones sumadas

CAPÍTULO 5. CONTROL POR REGULACIÓN

no necesariamente deben de ser consecutivas. Lo mismo se tiene que realizar para el resto de las transiciones de la especificación.

Mientras la especificación preserve la misma dimensión de las funciones del sistema, o bien, que las funciones de A^* sean de la misma dimensión que las de A' es posible encontrar π operando con las funciones propuestas debido a que el producto punto se puede realizar por bloques. En caso de que las funciones de A^* y A' no sean de la misma dimensión no es posible llevar a cabo el producto punto por bloques para el cálculo de π , sin embargo aún es posible aplicar el algoritmo desdoblado las funciones para encontrar π a través del producto punto; pero si se tiene que seguir este camino se pueden eliminar los renglones o columnas redundantes de la matriz de incidencia (producto de una función As_n) para facilitar el trabajo.

5. Calcular la función π por medio del producto punto. Se tiene que calcular cada renglón de π , de forma tal que el primer renglón de la función π multiplicado por medio del producto punto con cada renglón del bloque matricial izquierdo de 5.17 tiene que ser igual a la primera columna del bloque matricial derecho de 5.17 y así sucesivamente para satisfacer simultáneamente las dos primeras condiciones de la ecuación (5.16). Como se mencionó anteriormente sólo si A^{*T} y A'^T preservan la misma dimensión en las funciones es posible hacer el producto punto por bloques.

Es posible analizar este algoritmo desde el punto de vista siguiente:

Sean $A=(a_{ij})$ y $B=(b_{ij})$ matrices de dimensión $m \times n$ y $n \times p$, respectivamente, sobre el campo F . El producto AB es la matriz de dimensión $m \times p$

$C = (c_{ij})$, donde $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$, $(i, j) \in m \times p$. También $c_{ij} = A_i \cdot B^j$, pero

si ahora se transpone la matriz B tal que $D = B^T$ entonces $c_{ji} = A_i \cdot D_j =$

$$(c_{ij})^T \text{ por tal razón } \pi \begin{bmatrix} M_0^T \\ A'^T \end{bmatrix} = \begin{bmatrix} M_0^T \\ A^{*T} \end{bmatrix}$$

Algoritmo para probar la controlabilidad de los marcados alcanzables.

1. La función π y los vectores de Parikh \vec{w}_j son convertidos en funciones de colores y vectores de Parikh coloreados.
2. Una expansión parcial del grafo de alcanzabilidad del modelo del SMF en RPC , desde el marcado $\pi(M'_i)$ al $\pi(M'_j)$, donde la búsqueda se restringe a la transición y colores ejecutados en \vec{w}_j .

CAPÍTULO 5. CONTROL POR REGULACIÓN

3. Si la expansión parcial no es posible, entonces finalizar y el supervisor no puede ser computado, de lo contrario computar los marcados alcanzables.
4. Si los marcados alcanzables no son controlables, entonces finalizar y el algoritmo no puede ser computado, de lo contrario computar el supervisor.

5.4. Conclusiones.

Este capítulo se centró en el problema de control por regulación, por ello se dio el conjunto de condiciones para encontrar la función que permita empotrar el comportamiento de la especificación dentro del modelo del *SMF* en *RPC* y así resolver dicho problema. También se mostró que mediante la utilización del algoritmo propuesto es posible encontrar de manera sencilla la función π a través de las funciones propuestas, en caso de que las matrices contengan funciones de la misma dimensión, en caso contrario el algoritmo continúa siendo útil para resolver el problema sólo es necesario desdoblar las funciones y eliminar renglones o columnas redundantes.

6. CONVERSIÓN DE RPC A LD.

Resumen: Este capítulo presenta una metodología que permite convertir un modelo en RPC a un LD. El principio de esta metodología está fundamentado en que a cada color de una transición de la RPC es equivalente a una red de LD. Los elementos básicos de LD que se asocian a los lugares de la RPC son las bobinas y relevadores; cuando el lugar no es binario también se le asocian contadores y condicionales. También son presentados dos ejemplos donde se aplica esta metodología para traducir especificaciones en RPC a LD y se aplican los algoritmos propuestos en el capítulo anterior para encontrar π y verificar la controlabilidad. Además, se convierte un controlador H en LD.

6.1. Introducción.

Para el control de los SMF no basta el saber cómo obtener una representación compacta del sistema y su especificación, así como el diseño del controlador que iguale las salidas de ambos modelos para confinar el comportamiento del sistema dentro de la especificación; también es necesario la implementación del controlador en un PLC.

Uno de los problemas que enfrenta el personal encargado de programar el PLC es que no está correctamente capacitado para llevar a cabo dicha labor. Por tal motivo los programadores de PLC desempeñan su labor de forma empírica, por lo que suelen introducir muchos errores de programación o comportamientos incorrectos debido a especificaciones ambiguas, lo que trae como consecuencia pérdidas económicas por los largos tiempos de demora en localizar los conflictos y poner de nuevo en marcha al sistema. Siguiendo métodos formales para programar un PLC se puede evitar este tipo de inconvenientes, sin embargo es necesario desarrollar un método para la traducción de las RPC a un lenguaje aceptado por los PLC.

Con el fin de mejorar las técnicas de programación de sistemas de control industriales se creó el estándar IEC1131-3 [17], donde se proporciona una selección

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

de lenguajes (*ST*, *IL*, *LD*, *FBD* y *SFC*) para desarrollar el programa de control, pero ellos no proporcionan una metodología de modelado. Algunas investigaciones se han enfocado al desarrollo de metodologías para convertir un modelo a uno de los lenguajes establecidos en el *IEC1131 – 3* para la programación de un *PLC*, y así poder implementar los controladores a las redes industriales.

Este capítulo está estructurado de la siguiente manera: En la sección 6.2 se propone una metodología para convertir *RPC* a *LD*. Después, en la sección 6.4, se presentan ejemplos donde se aplica el algoritmo del capítulo anterior para encontrar la función π que permita empotrar el comportamiento de la especificación dentro del modelo del *SMF*. Además las especificaciones dadas en *RPC* son convertidas a *LD* por medio de la metodología propuesta. Posteriormente en la sección 6.5 se muestra un ejemplo donde se construye el *LD* de un controlador para que pueda ser implementado en *PLC* y así el sistema se comporte conforme a la especificación dada. Por último, las conclusiones de este capítulo son presentadas en la sección 6.6.

6.2. Metodología para convertir una RPC a un LD.

Esta sección presenta una metodología para convertir modelos en *RPC* a *LD*, siendo el *LD* uno de los cinco lenguajes aceptados por el estándar *IEC1131* para programar un *PLC*. Esta metodología se basa en el trabajo de [16] para convertir *RP* no binarias en *LD*; pero en este caso la metodología se fundamenta en que cada color de una transición es equivalente a una red de *LD*. También se usan funciones paramétricas que los *PLC's* modernos aceptan en su programación para mover los robots a una posición específica.

Para las *RPC* cada color asociado a un lugar o a una transición tiene una interpretación del sistema físico como se observó en el ejemplo de la sección 2.4. Así cada señal de entrada al sistema (color de una transición) se representa en *LD* mediante un contacto. Los elementos e instrucciones de *LD* que esta metodología emplea son los descritos por la norma *IEC1131*.

6.2.1. Disparo de una transición.

La característica principal de la metodología para traducir de *RPC* a *LD* es que se deben de considerar todos los colores de entrada que habilitan un color de una transición y sus los colores de salida cuando la transición sea dispara con respecto a dicho color. Una red de *LD* es un renglón que está compuesto básicamente por

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

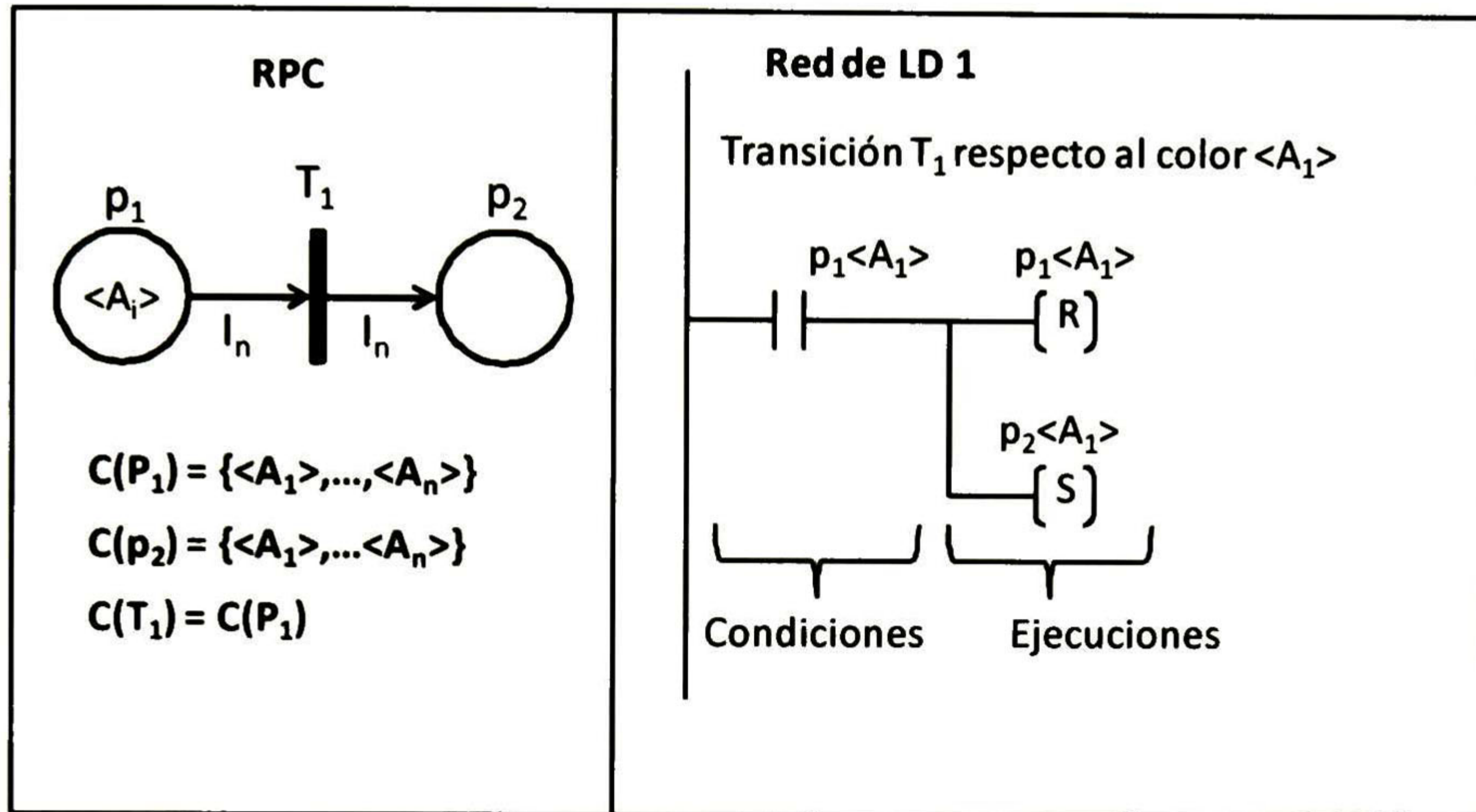


Figura 6.1: Representación del disparo de una transición respecto a un color de entrada y salida binario.

2 partes: La condiciones y las ejecuciones. El comportamiento de una *RPC* puede ser representado en tales partes, es decir, hay una analogía entre las condiciones y las ejecuciones de un *LD* con las reglas de habilitación y disparo de una transición de una *RPC*.

La representación de una transición se consigue asociando cada color de entrada y salida a un elemento básico de *LD*. Así, a cada color de entrada que permita habilitar un color de la transición es representado con un contacto normalmente abierto y una bobina que reinicie el estado (*R*), , si la entrada no es binaria se representa con un comparador y con un contador descendente. Los colores de salida son representados con bobinas activas (*S*) y si los colores de salida no son binarios son representados con un contador ascendente. Si la transición tiene asociada una señal externa, está se representa mediante un contacto normalmente abierto.

En la figura 6.1 y en la 6.2 se muestra la representación del disparo del color de una transición. Asimismo en la figura 6.3 se observa la representación de una transición cuando los colores de entrada y salida no son binarios.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

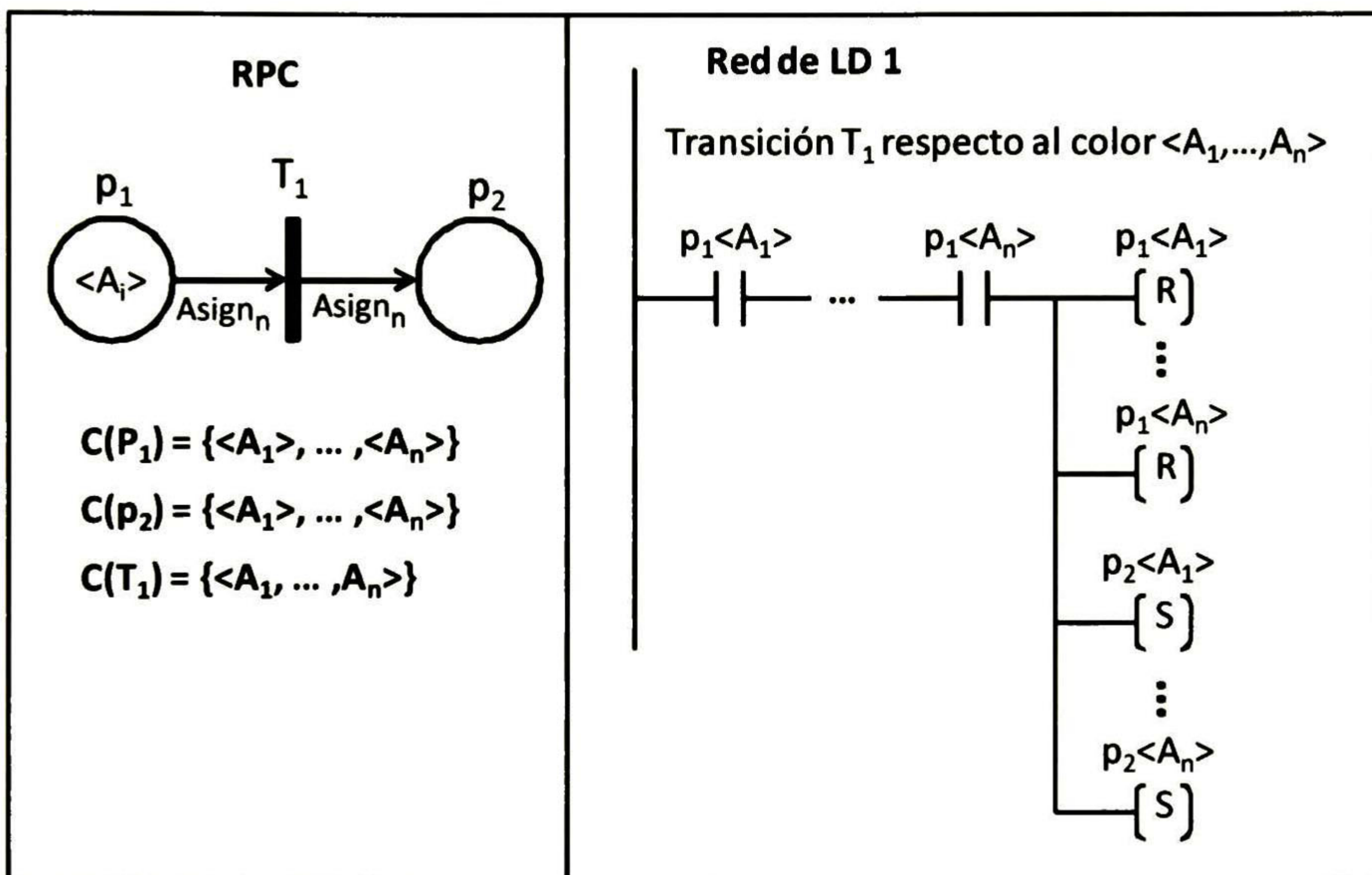


Figura 6.2: Representación del disparo de una transición respecto a colores no binarios.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

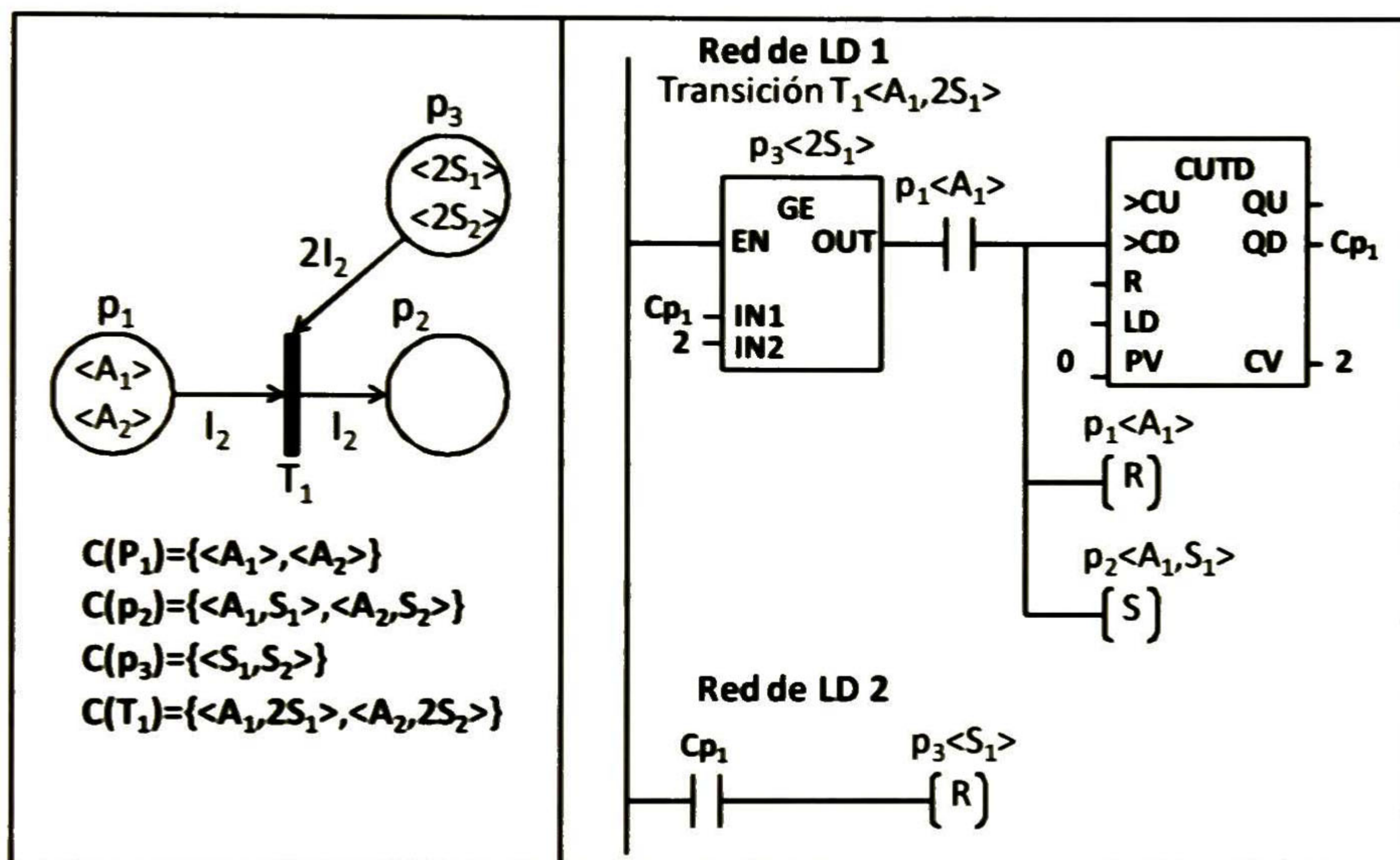


Figura 6.3: Representación del disparo de una transición respecto a colores no binarios.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

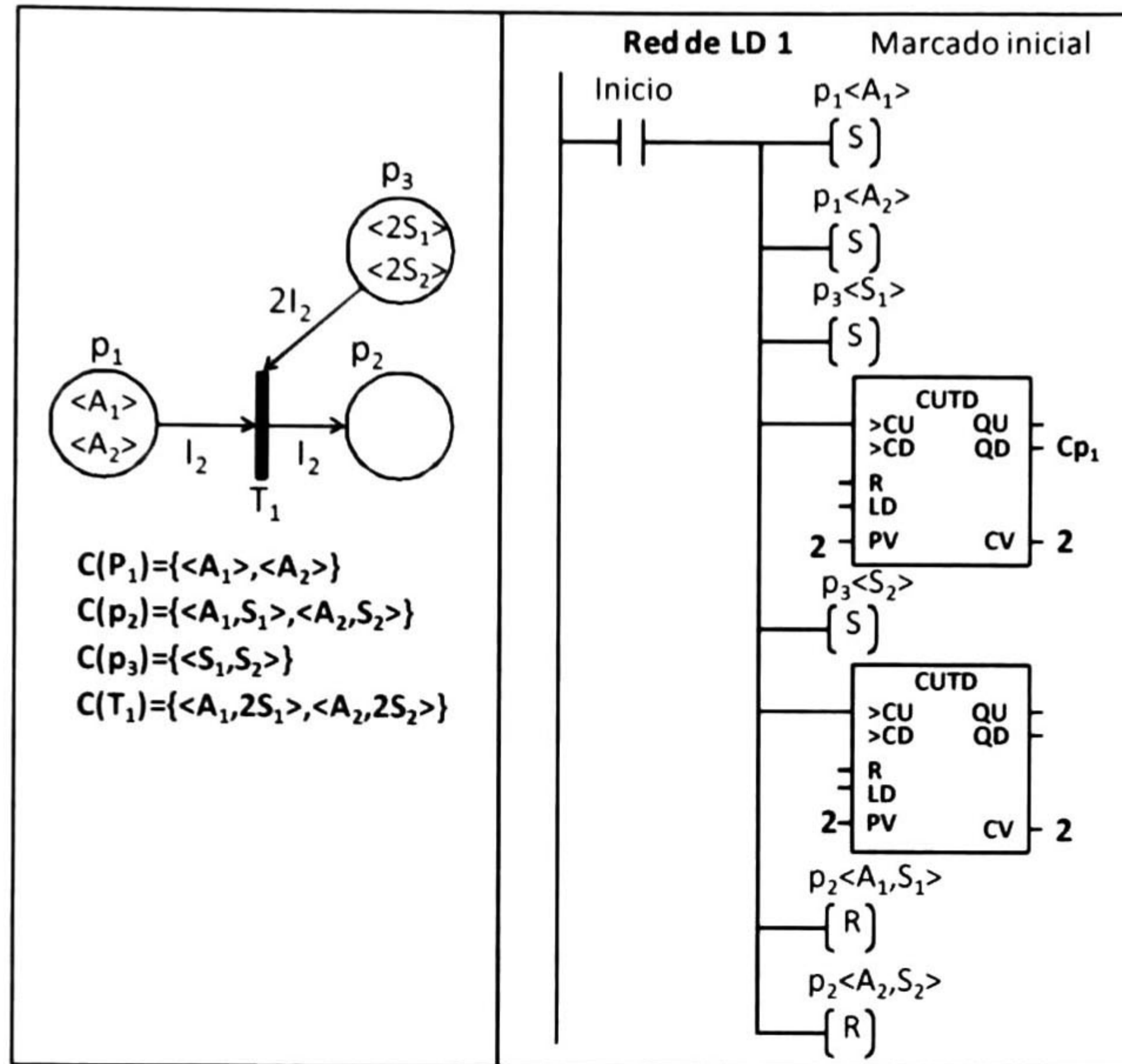


Figura 6.4: Representación del marcado inicial de una RPC

6.3. Marcado inicial.

El marcado inicial de un modelo en *RPC* se establece mediante una sola red de *LD*, la cual consiste de:

- Un contacto normalmente abierto asociado a una señal externa de inicialización.
- Un conjunto de bobinas activas correspondientes a los lugares inicialmente marcados (conjunto inicial de colores dentro de la *RPC*) o un contador si el lugar no es binario.
- Un conjunto de bobinas inactivas para el resto de los colores de los lugares. Si un lugar no marcado no es binario se le debe asignar un contador con una cuenta inicial de cero.

La figura 6.4 muestra como se asigna el marcado inicial de una *RPC* con lugares no binarios a un *LD*.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

	T_1	T_2
P_1	$-As_3$	0_3
	I_3	$-I_3$
P_2	0_3	I_3

Figura 6.5: Matriz de incidencia de la especificación del sistema de transporte de partes.

5 5 5	0 0 0	0 0 0		5 5 5	0 0 0	0 0 0	1 1 1	0 0 0	0 0 0	1 1 1	1 1 1	0 0 0
$-As_3$	I_3	0_3		$-As_3$	I_3	0_3	0_3	0_3	0_3	0_3	0_3	0_3
0_3	$-I_3$	I_3		0_3	$-I_3$	0_3	0_3	0_3	0_3	0_3	0_3	I_3

Figura 6.6: Matriz construida para encontrar π .

6.4. Ejemplos.

A continuación se presentan dos ejemplos donde se muestra en detalle cómo calcular π aplicando el algoritmo propuesto en el capítulo anterior y la traducción de la especificación del sistema a *LD*.

Ejemplo 27. Retomemos el sistema de reconocimiento y transporte de partes del ejemplo 19. La especificación para este sistema fue mostrada en la figura 4.12, a la cual se le asignan las siguientes funciones:

$$f_1 = Asign_3 \quad f_2 = f_3 = f_4 = I_3$$

La matriz de incidencia (ver figura 6.5) y marcado el marcado inicial para esta especificación son:

$$M_0^T = (5 \ 5 \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$$

Construyendo la matriz propuesta en (5.17) se obtiene la matriz que se exhibe en la figura 6.6:

Calculando la matriz π por medio del producto punto se obtiene la siguiente:

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

$$\pi = \begin{array}{|c|c|c|} \hline I_3 & 0_3 & 0_3 \\ \hline 0_3 & I_3 & 0_3 \\ \hline 0_3 & 0_3 & 0_3 \\ \hline \frac{1}{5}I_3 & \frac{1}{5}As_3 & \frac{1}{5}As_3 \\ \hline 0_3 & 0_3 & 0_3 \\ \hline 0_3 & 0_3 & 0_3 \\ \hline \frac{1}{5}I_3 & \frac{1}{5}As_3 & \frac{1}{5}As_3 \\ \hline \frac{1}{5}I_3 & \frac{1}{5}As_3 & \frac{1}{5}As_3 \\ \hline 0_3 & 0_3 & I_3 \\ \hline \end{array}$$

Por la forma en que se construyó la matriz (5.17) si se encuentra una matriz π que satisfaga que $\pi \begin{bmatrix} M_0'^T \\ A'^T \end{bmatrix} = \begin{bmatrix} M_0^T \\ A^{*T} \end{bmatrix}$ se cumple con las condiciones 1 y 2 de la ecuación 5.16, por tal motivo solamente falta verificar que también se cumpla con la tercer condición. Para ello, la matriz φ se construye de acuerdo a los lugares medibles de la figura 4.12, los cuales son los lugares representados con C y D . La matriz φ puede expresarse por medio de funciones identidad como se muestra a continuación.

$$\varphi = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \frac{p_1}{C} & p_2 & p_3 & \frac{p_4}{C} & \frac{p_5}{D} & p_6 & & & \\ \hline I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ \hline 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ \hline 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 \\ \hline \end{array}$$

Y como:

$$\varphi' = \begin{array}{|c|c|c|} \hline I_3 & 0_3 & 0_3 \\ \hline 0_3 & I_3 & 0_3 \\ \hline 0_3 & 0_3 & I_3 \\ \hline \end{array}$$

se cumple con la condición 3 de la ecuación (5.16).

Después de haber encontrado π se debe verificar que los marcados alcanzables son controlables cuando la secuencia w_j es disparada. De la figura 4.12 y 4.5 se puede apreciar que los marcados alcanzados al disparar las secuencias $(t_1, t_4, t_7, t_{10}, t_{13})$, $(t_2, t_5, t_8, t_{11}, t_{14})$ y $(t_3, t_6, t_9, t_{12}, t_{15})$ son controlables.

Por último se convertirá el modelo de la especificación presentado en la figura 4.12 a LD. Las figuras 6.7 y 6.8 muestran el LD de la especificación.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

Ejemplo 28. Consideremos la celda de ensamble del ejemplo 20. El modelo del sistema fue mostrado en la figura 4.9 y el modelo de su especificación en la figura 4.13. Primeramente se definirán la dimensión de las funciones de la especificación y su matriz de incidencia, así como su marcado inicial para que pueda ejecutar los ensambles (A_1, A_2, A_3) o el (A_2, A_3, A_1) .

Las funciones asociadas a la especificación son:

$$f_1 = f_2 = f_3 = f_9 = I_3$$

$$f_4 = f_5 = f_{10} = A_{sign_3}$$

$$f_7 + f_8 = Ens_3$$

$$f_6 = 0_3^*$$

$$f_{11} = r(A_{sign_3})$$

La matriz de incidencia (ver figura 6.9) y el marcado inicial de la especificación son:

$$M_0^T = (22 \ 2 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0)$$

Ahora construyendo la matriz propuesta en (5.17) se tiene la que se muestra en la figura (6.10):

Calculando la matriz π se obtiene la siguiente:

$$\pi = \begin{array}{c} \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0 & 0_3 & 0_3 & 0_3 & 0 \\ \hline 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0 & 0_3 & 0_3 & 0_3 & 0 \\ \hline 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0 & 0_3 & 0_3 & 0_3 & 0 \\ \hline 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0 & 0_3 & 0_3 & 0_3 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0 & I_3 & 0_3 & 0_3 & 0 \\ \hline 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0 & 0_3 & 0_3 & I_3 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array} \end{array}$$

En la figura 4.13 los lugares marcados las letras A , B , C y D representan los lugares medibles por los sensores. Construyendo la matriz φ se tiene que:

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

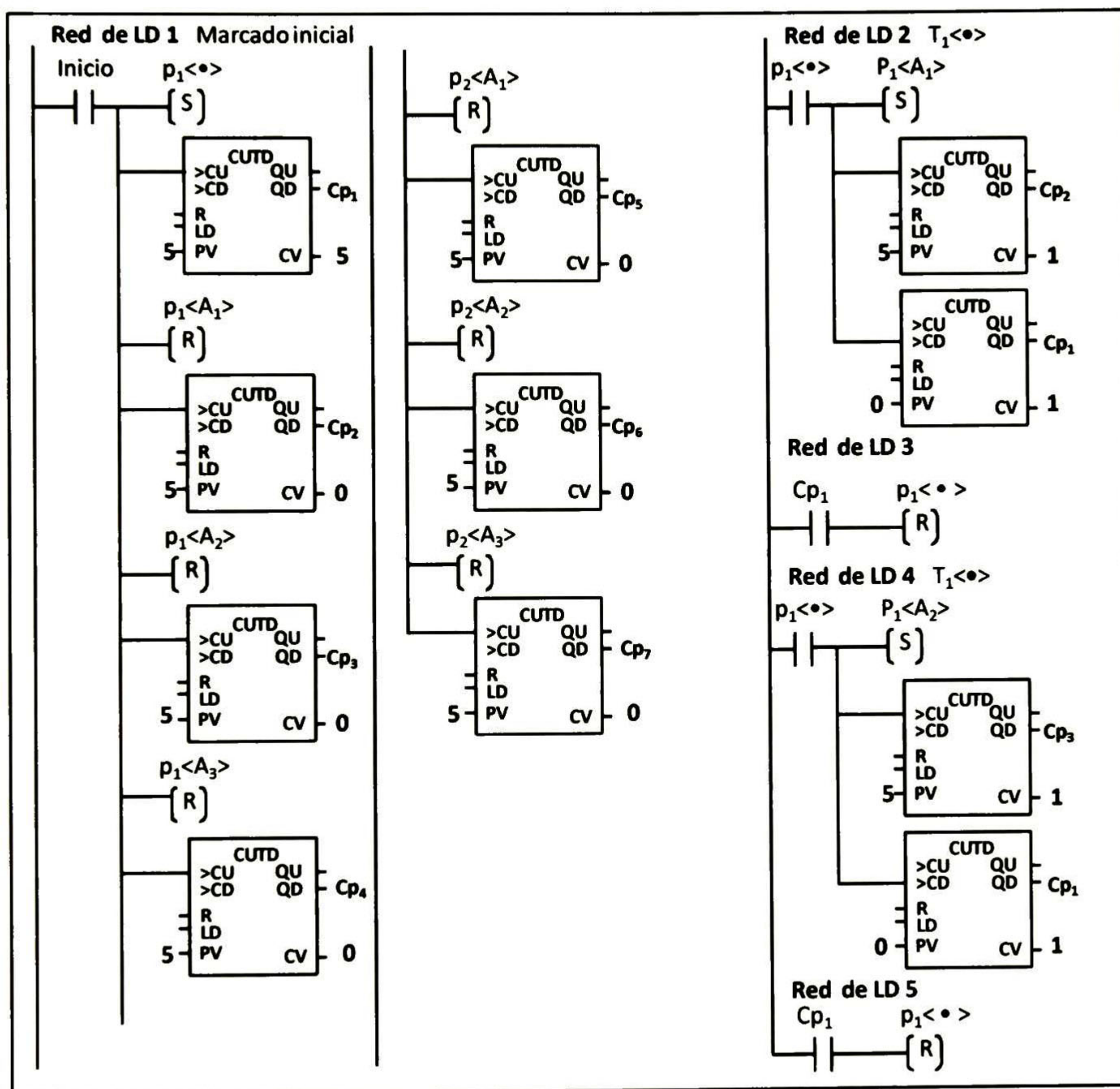


Figura 6.7: Representación en LD de la especificación en RPC del sistema de transporte de partes. Parte I.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

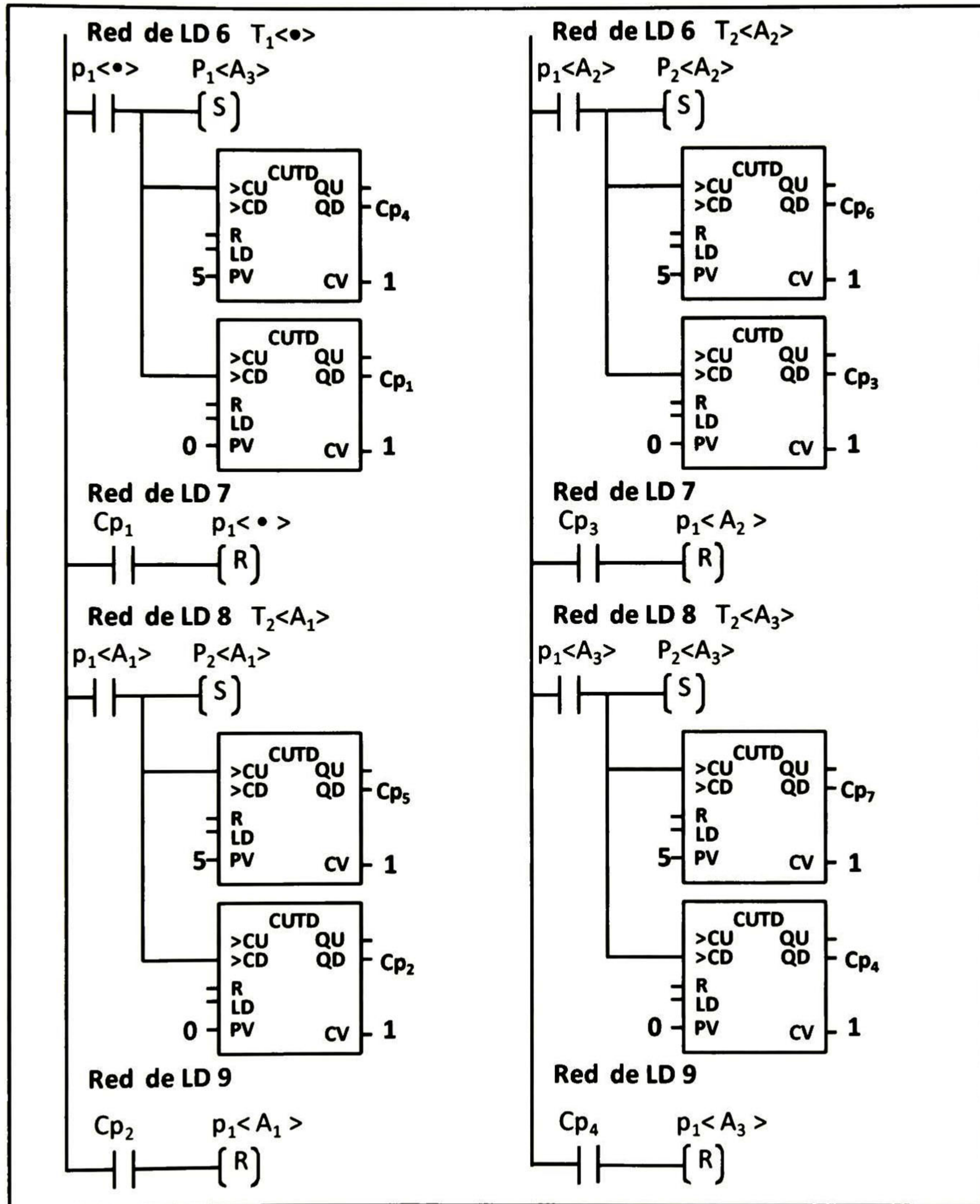


Figura 6.8: Representación en LD de la especificación en RPC del sistema de transporte de partes. Parte II.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

	T_1	T_2	T_3	T_4	T_5	T_6
P_1	$-I_3$	0_3	0_3	$-I_3$	0_3	0_3
P_2	I_3	$-I_3$	0_3	0_3	0_3	0_3
P_3	$-Asig_3$	$Asig_3$	0_3	$-Asig_3$	$Asig_3$	0_3
P_4	0_3	Ens_3	0_3	0_3	0_3	0_3
P_5	0_3	I_3	$-Asig_3$	0_3	0_3	0_3
P_6	000	000	1 1 1	000	000	000
P_7	0_3	0_3	0_3	I_3	$-I_3$	0_3
P_8	0_3	0_3	0_3	0_3	Ens_3	0_3
P_9	0_3	0_3	0_3	0_3	I_3	$-Asig_3$
P_{10}	000	000	000	000	000	1 1 1

Figura 6.9: Matriz de incidencia para la especificación del sistema de ensamble.

2 2 2	0 0 0	1 1 1	0 1 1	0 0 0	0	0 0 0	1 0 1	0 0 0	0	2 2 2	0 0 0	1 1 1	0 0 0	0	0 0 0	0 0 0	0
$-I_3$	I_3	$-Asig_3$	0_3	0_3	0	0_3	0_3	0_3	0	$-I_3$	I_3	$-Asig_3$	0_3	0	0_3	0_3	0
0_3	$-I_3$	$Asig_3$	Ens_3	I_3	0	0_3	0_3	0_3	0	0_3	$-I_3$	$Asig_3$	I_3	0	0_3	0_3	0
0_3	0_3	0_3	0_3	$-Asig_3$	1	0_3	0_3	0_3	0	0_3	0_3	0_3	$-Asig_3$	1	0_3	0_3	0
$-I_3$	0_3	$-Asig_3$	0_3	0_3	0	I_3	0_3	0_3	0	$-I_3$	0_3	$-Asig_3$	0_3	0	I_3	0_3	0
0_3	0_3	$Asig_3$	0_3	0_3	0	$-I_3$	Ens_3	I_3	0	0_3	0_3	$Asig_3$	0_3	0	$-I_3$	I_3	0
0_3	0_3	0_3	0_3	0_3	0	0_3	0_3	$-Asig_3$	1	0_3	0_3	0_3	0_3	0	0_3	$-Asig_3$	1

Figura 6.10: Matriz construida para encontrar π .

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

$$\varphi =$$

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
A	B		C			D	
I_3	O_3	O_3	O_3	0	O_3	O_3	0
O_3	O_3	I_3	O_3	0	O_3	O_3	0
000	000	000	000	1	000	000	0
000	000	000	000	0	000	000	1

Con lo que es muy fácil comprobar que $\varphi' = \varphi\pi$.

Una vez que se encontró π y se verificó la tercera condición de la ecuación (5.16), hay que probar la controlabilidad de los marcados alcanzables cuando la secuencia w_j es disparada. De las figuras 4.11 y 4.13 es fácil observar que el marcado alcanzado por disparar los colores dados por las secuencias $t_1, t_4, t_2, t_5, t_3, t_6$ y (t_7, t_8 o t_9 porque estas transiciones son redundantes) y la secuencia $t_{11}, t_{14}, t_{12}, t_{15}, t_{10}, t_{13}$ y (t_{16}, t_{17} o t_{18}) son secuencias controlables.

Finalmente se convertirá el modelo de la especificación mostrado en la figura 4.13 a LD. En este caso las funciones paramétricas presentadas ayudarán a que el robot se mueva a las coordenadas especificadas para realizar la tarea establecida y finalmente regresará a su posición de seguridad. Las figuras 6.11, 6.12, 6.13 y 6.14 muestran el LD de la especificación.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

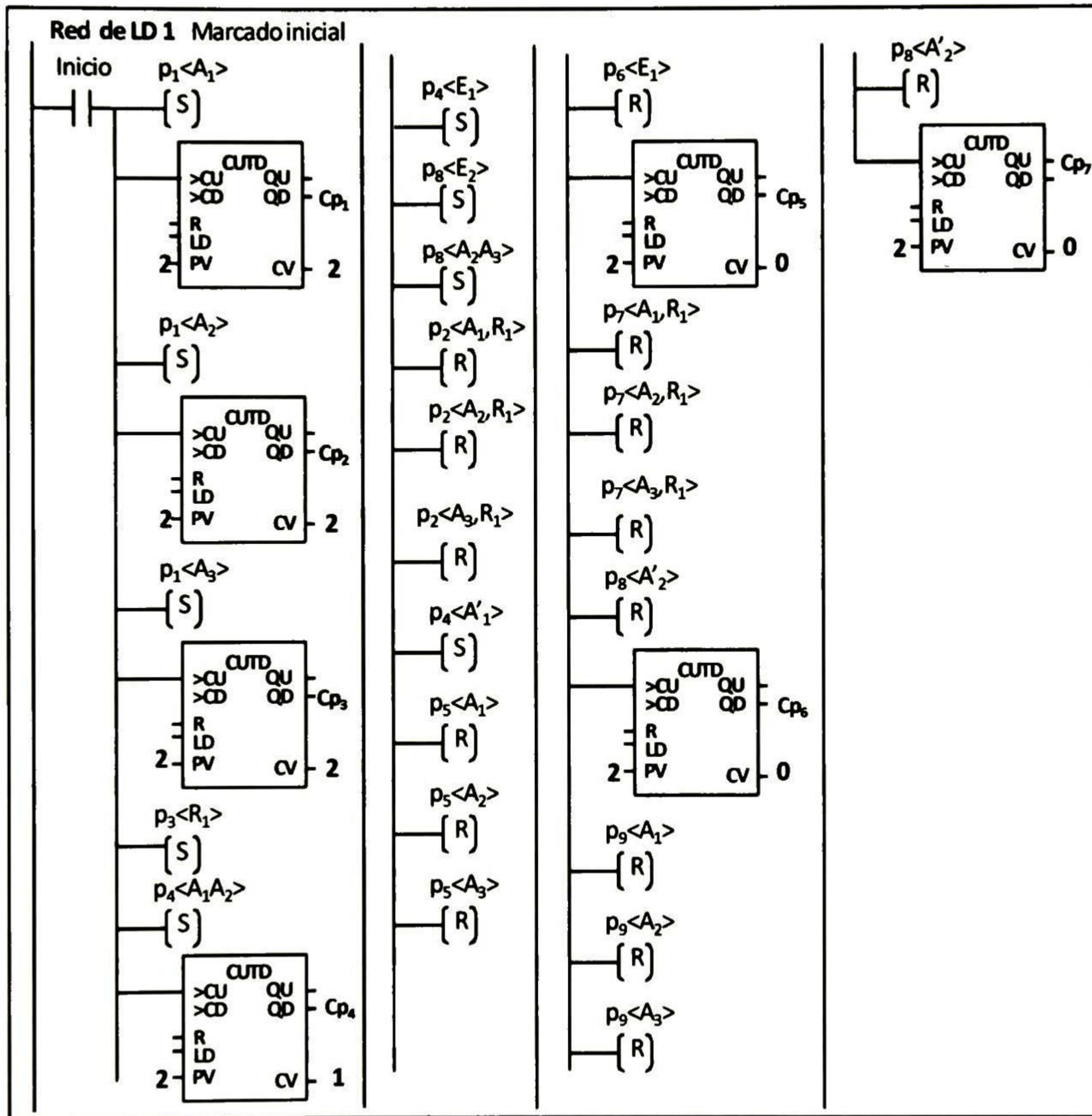


Figura 6.11: Representación en LD de la especificación en RPC del sistema de ensamble. Parte I.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

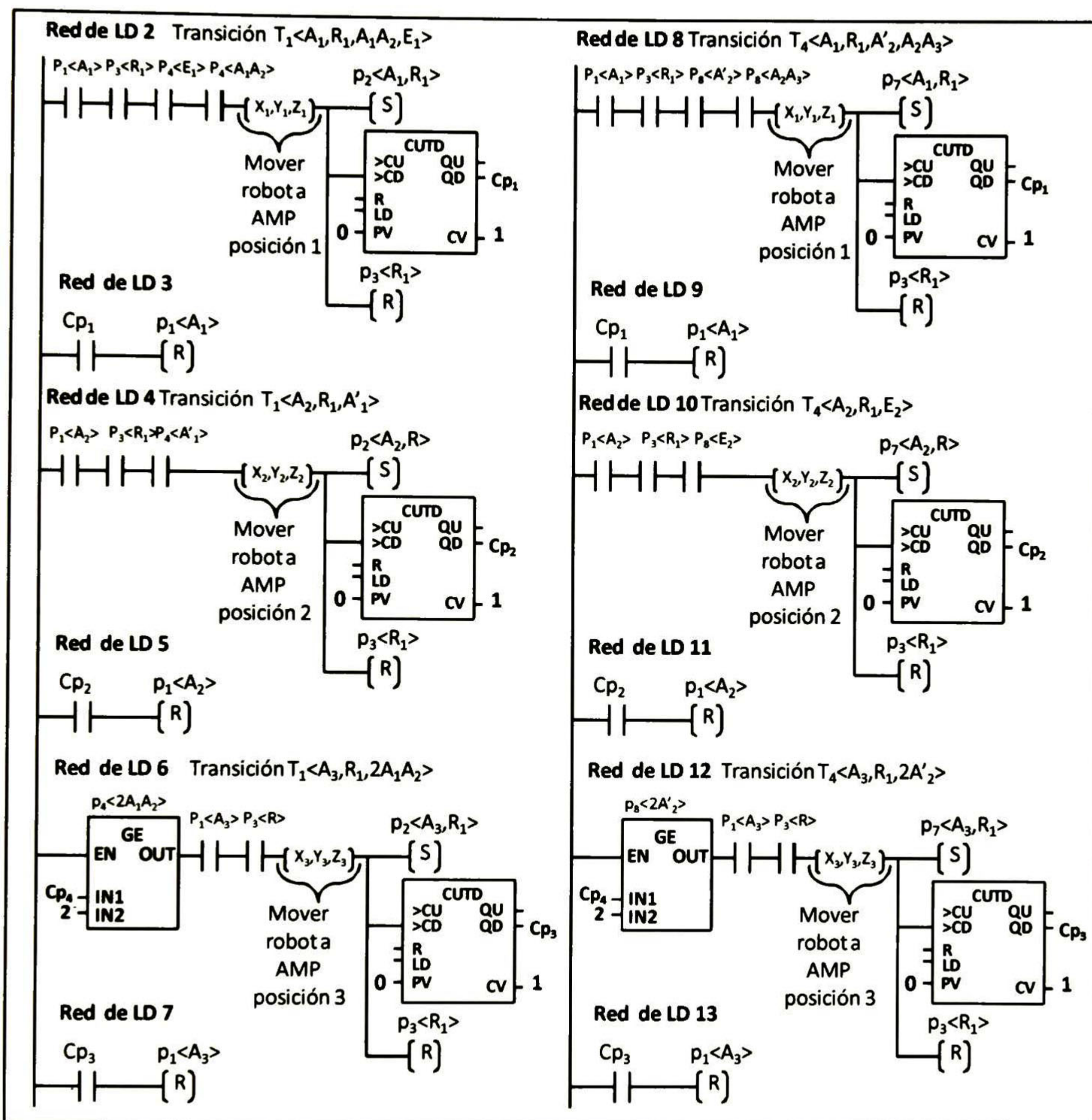


Figura 6.12: Representación en LD de la especificación en RPC del sistema de ensamble. Parte II.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

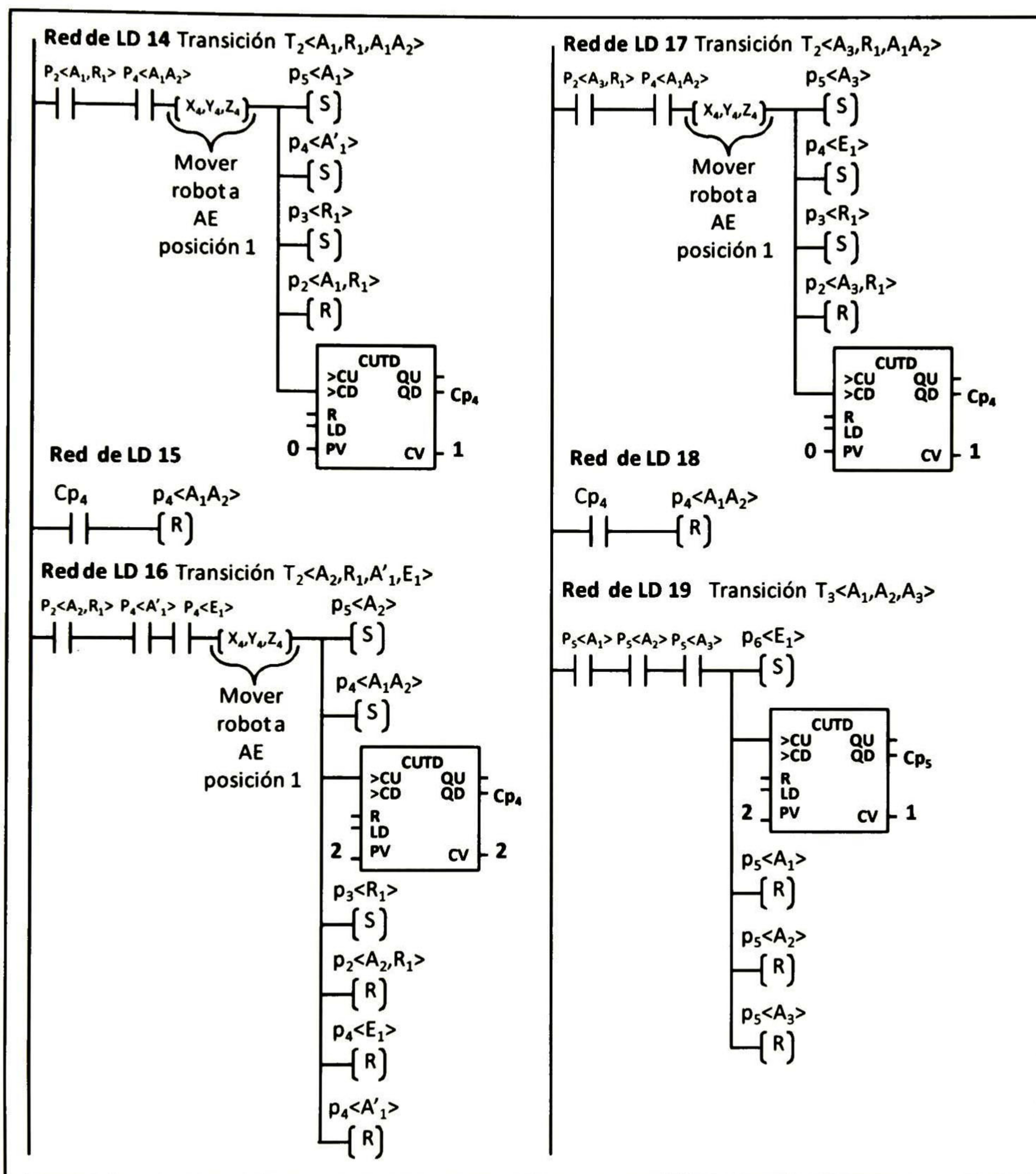


Figura 6.13: Representación en LD de la especificación en RPC del sistema de ensamble. Parte III.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

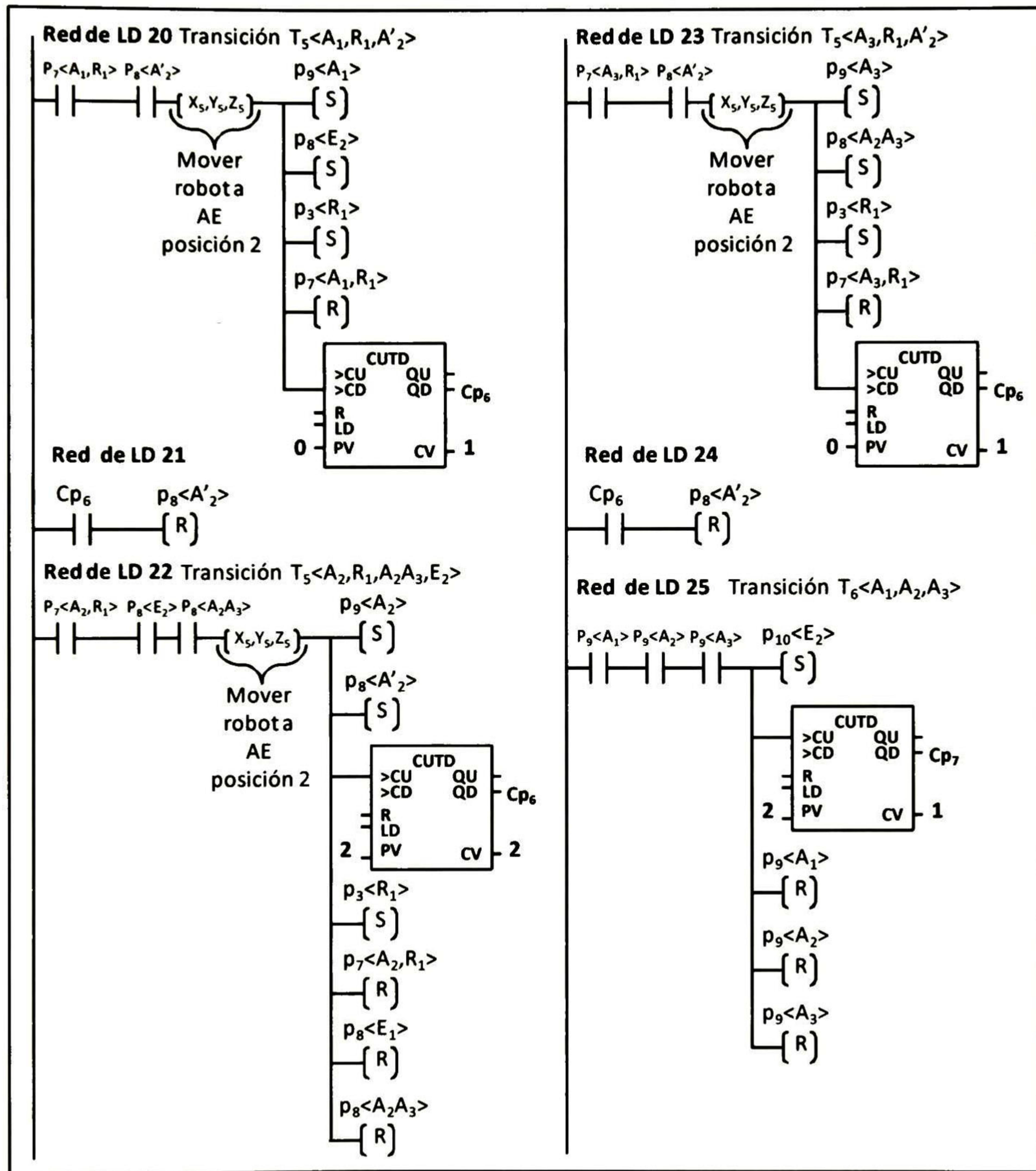


Figura 6.14: Representación en LD de la especificación en RPC del sistema de ensamble. Parte IV.

6.5. Conversión del controlador H en LD .

Para este ejemplo se tomará el modelo en RPC de la figura 4.9 y la especificación mostrada en la figura 4.13. Del ejemplo (28) se comprobó que el comportamiento del SMF puede confinarse al de dicha especificación. Para que el sistema se comporte de acuerdo a la especificación establecida es necesario añadir arcos de control (arcos bidireccionales que permiten verificar condiciones) para interconectar los sistemas, como se muestra en la figura 6.15. Los arcos azules permiten que el sistema siga a la especificación sin que el sistema se adelante a disparar alguna transición. Los arcos rojos y verdes permiten comprobar que el sistema haya alcanzado los estados que indicó anteriormente la especificación, sino los ha alcanzado estos arcos detienen el avance de la especificación hasta que el sistema cumpla con las condiciones. También como se aprecia en la figura 6.15 los arcos rojos comprueban las condiciones iniciales del sistema, lo cual se lleva a cabo para estar seguros que el sistema está en las condiciones para desarrollar la tarea que se establezca en la especificación.

Finalmente este modelo en lazo cerrado se traducirá a LD . El marcado inicial para este modelo es igual al que se mostró en la figura 6.11. El resto del diagrama de escalera para el controlador se muestra en las figuras 6.16(6.22. En el LD presentado, las bobinas marcadas con Q permiten activar a las transiciones del sistema, las cuales sólo permanecen activas mientras reciban señal.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

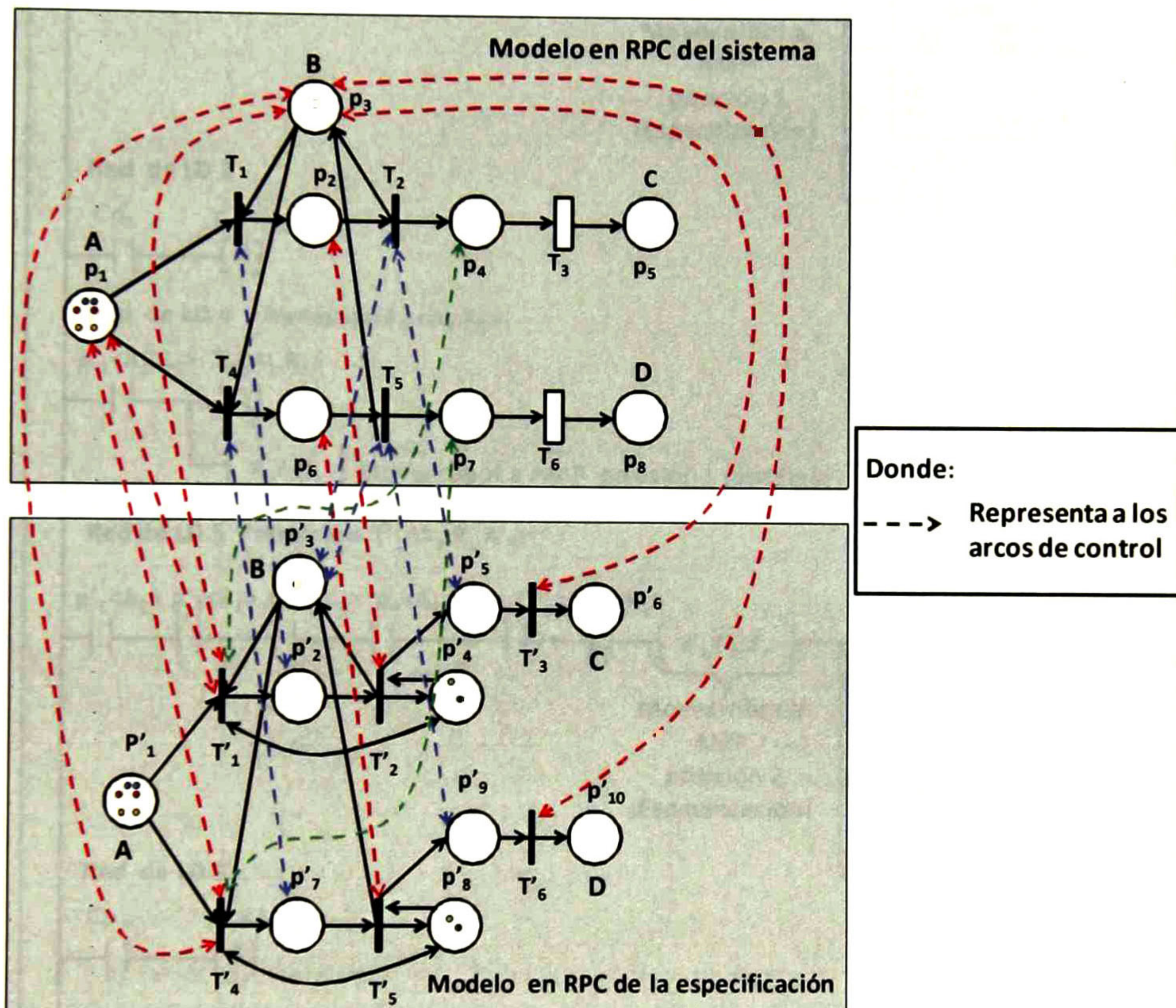


Figura 6.15: RPC del sistema y el modelo en RPC de la especificación en lazo cerrado.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

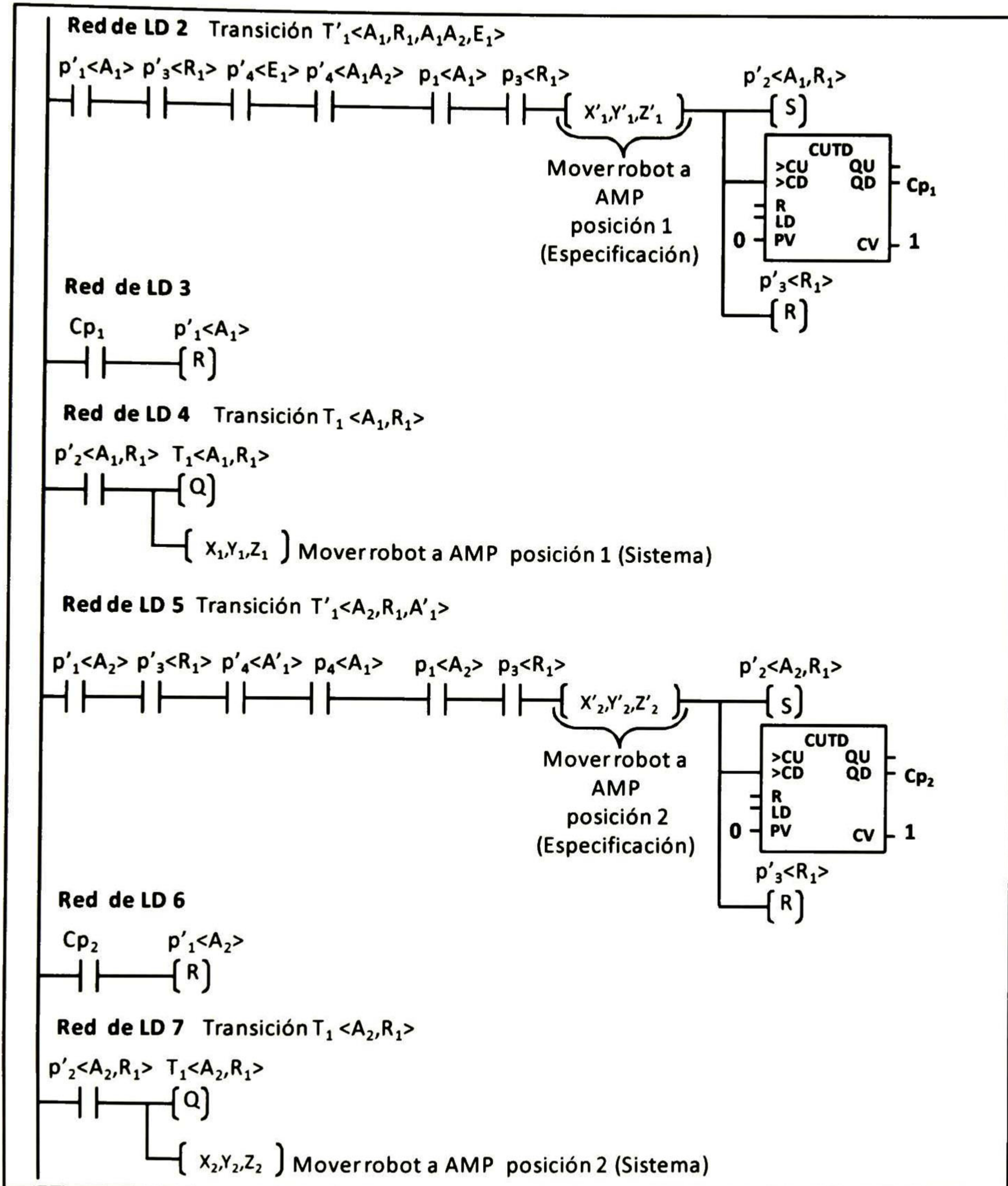


Figura 6.16: LD del sistema en lazo cerrado. Parte I.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

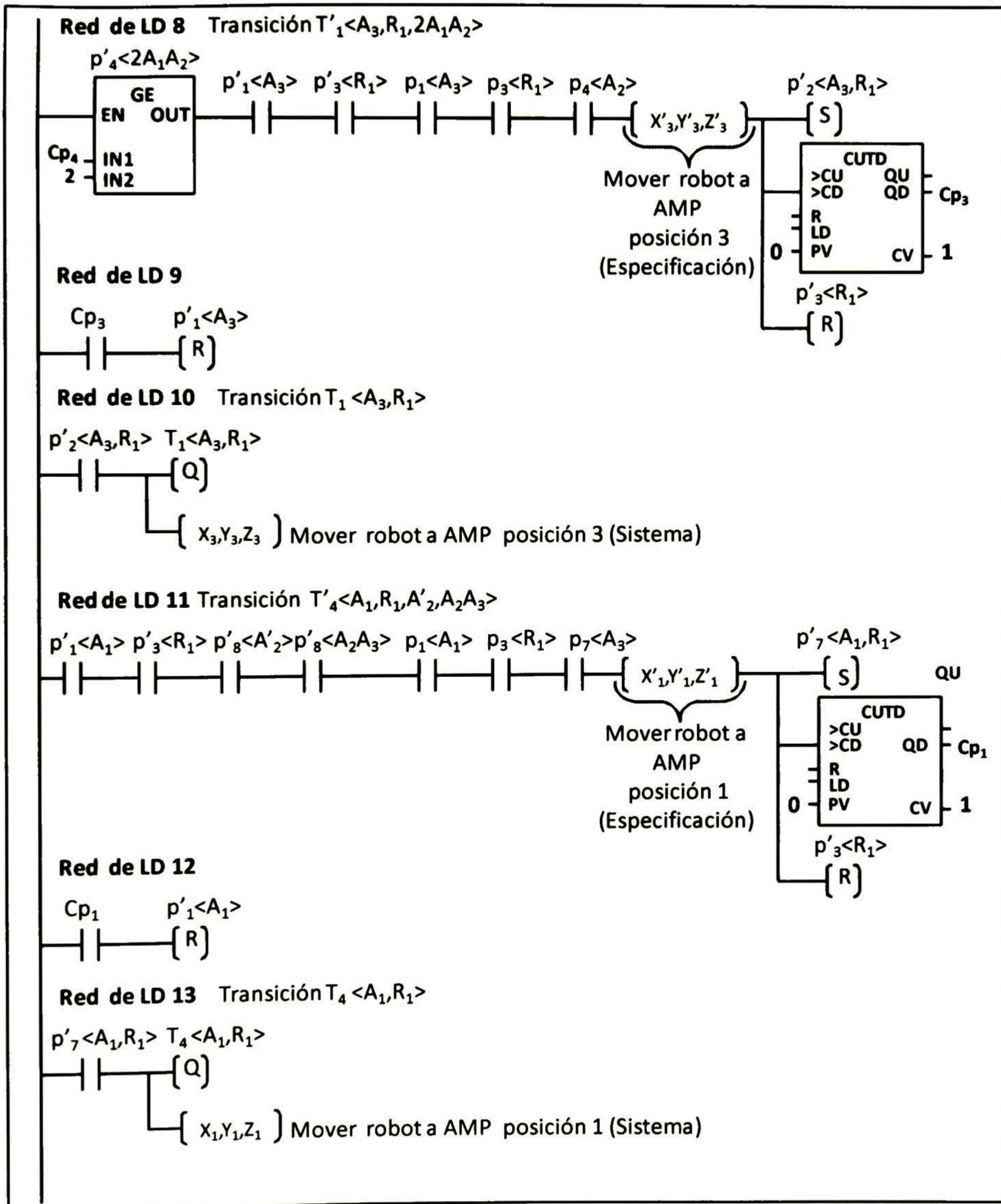


Figura 6.17: LD del sistema en lazo cerrado. Parte II.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

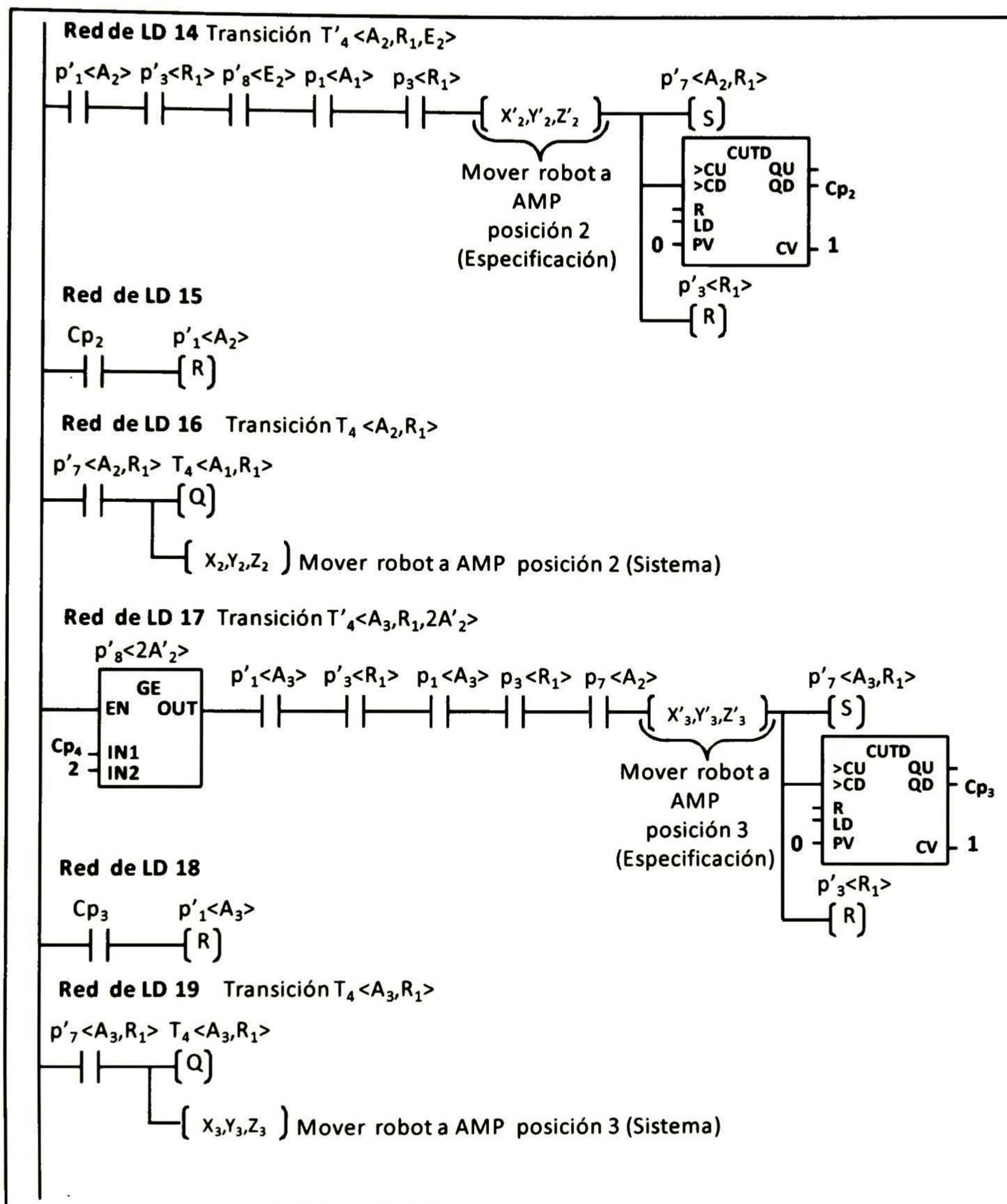


Figura 6.18: LD del sistema en lazo cerrado. Parte III.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

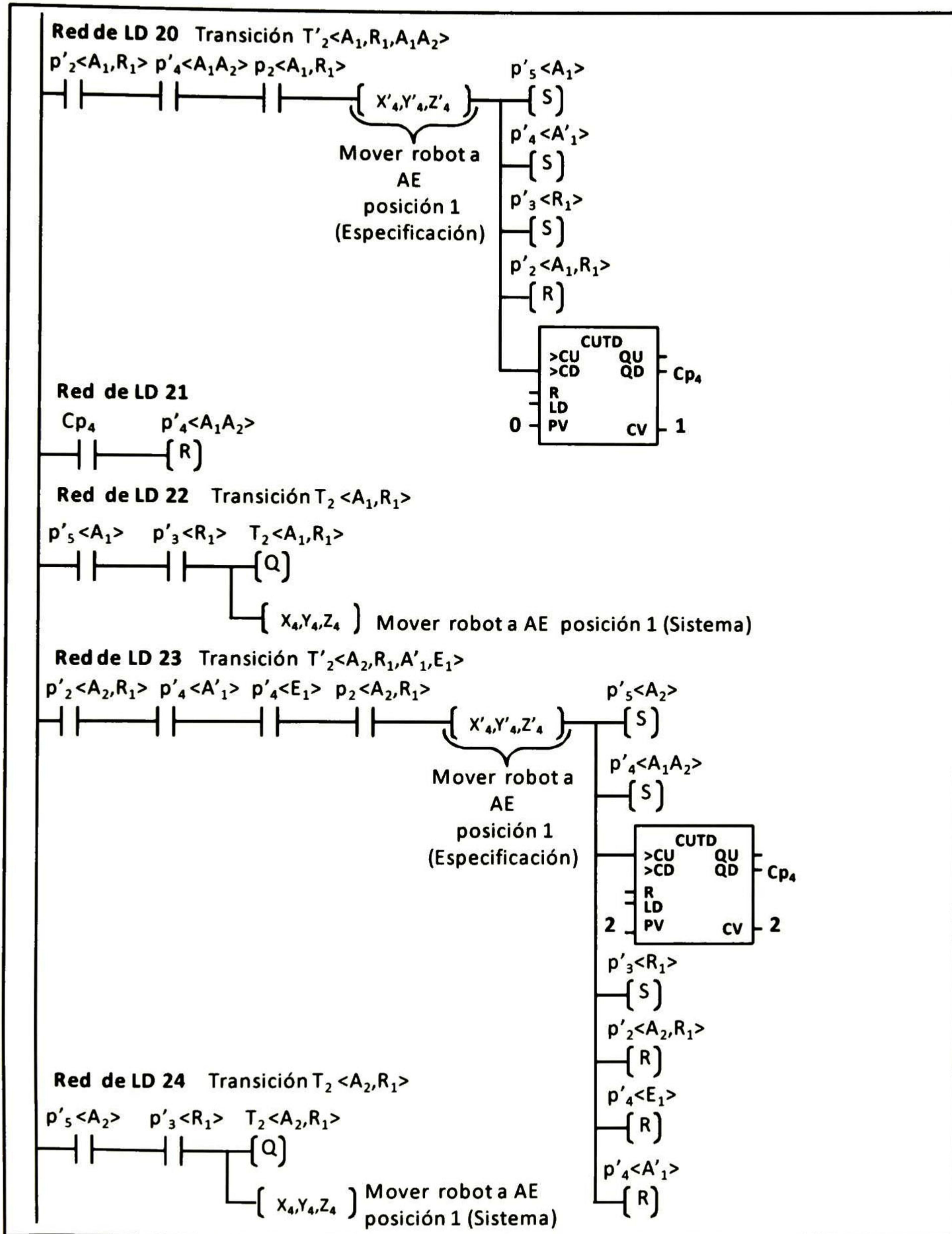


Figura 6.19: LD del sistema en lazo cerrado. Parte IV.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

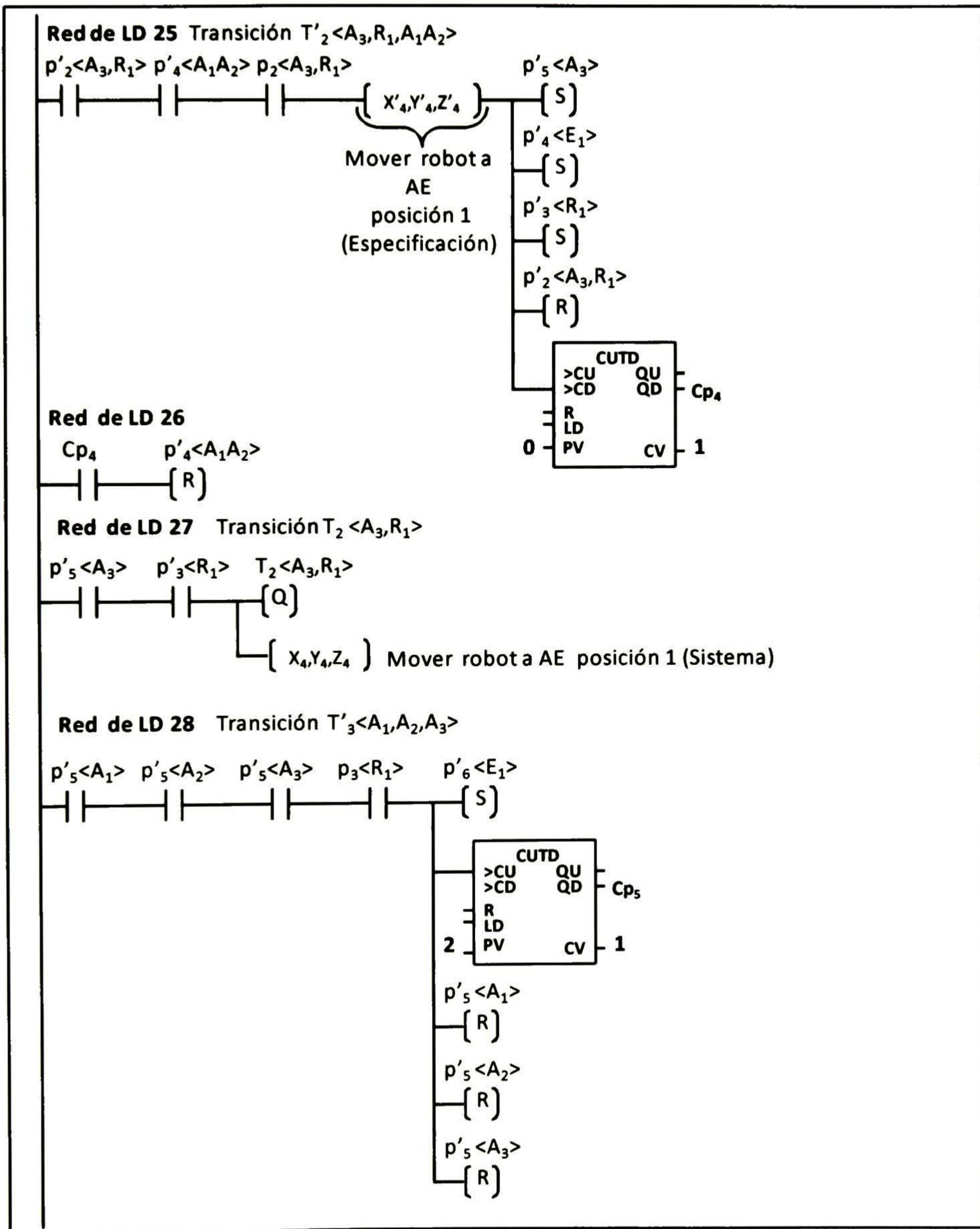


Figura 6.20: LD del sistema en lazo cerrado. Parte V.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

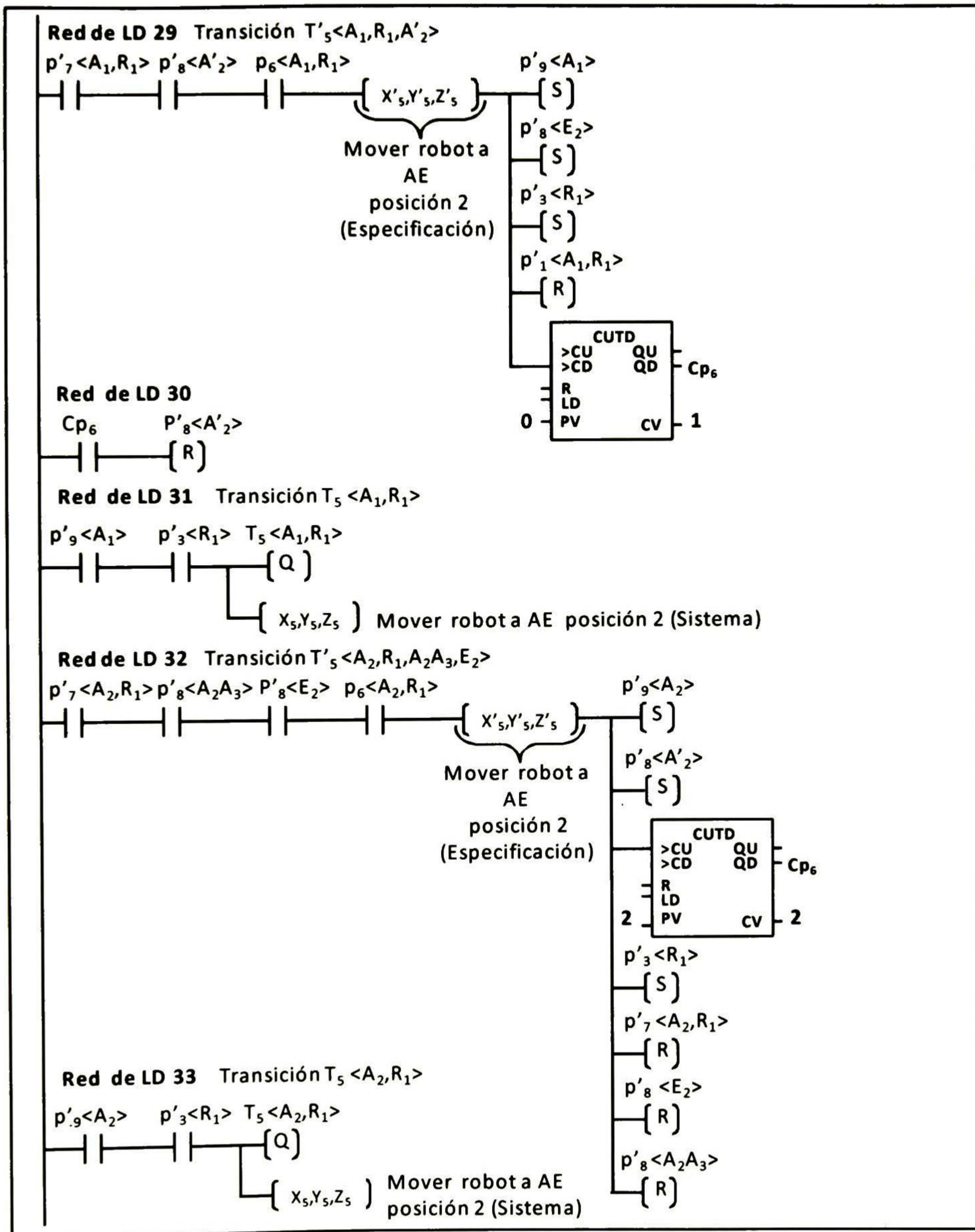


Figura 6.21: LD del sistema en lazo cerrado. Parte VI.

CAPÍTULO 6. CONVERSIÓN DE RPC A LD

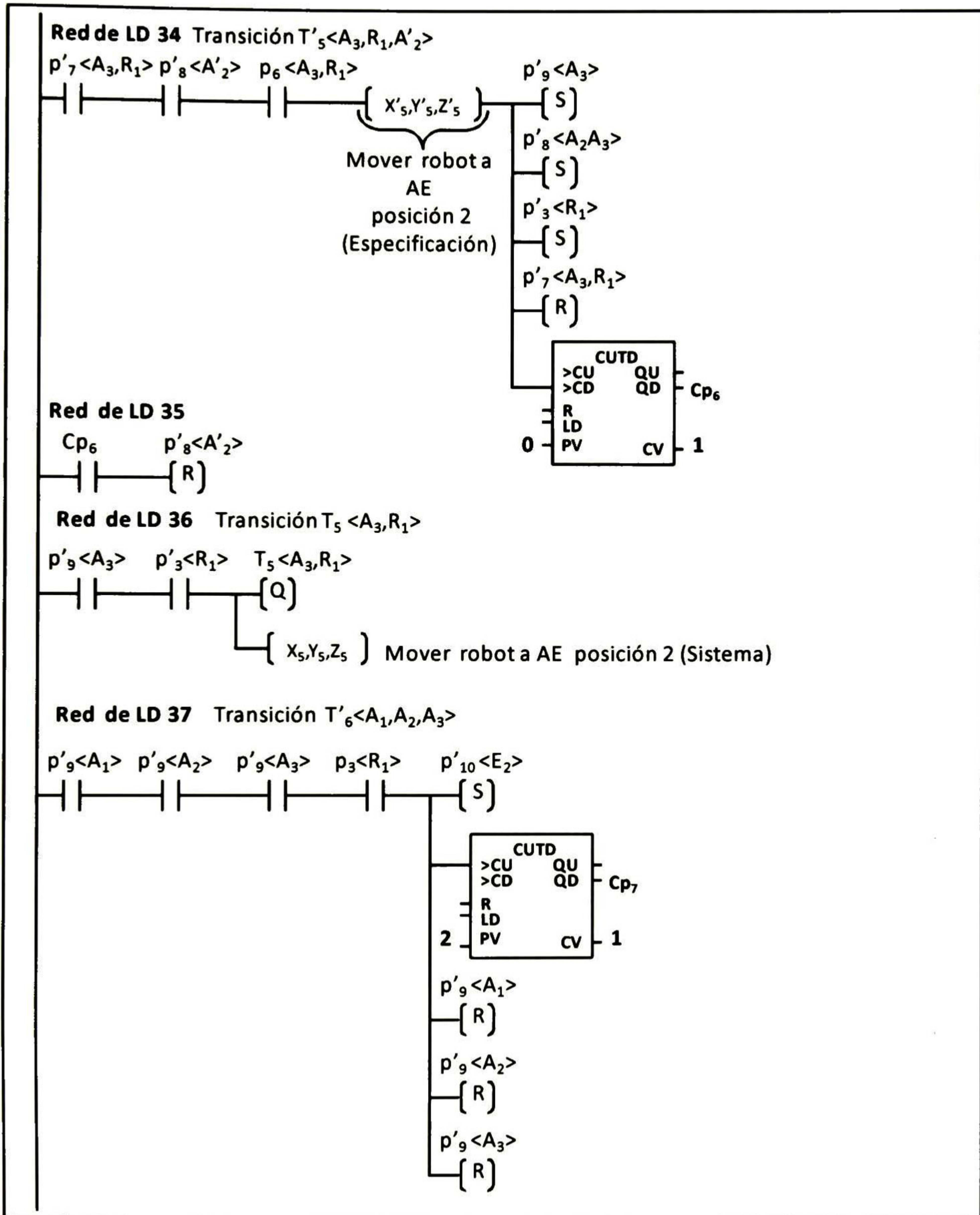


Figura 6.22: LD del sistema en lazo cerrado. Parte VII.

6.6. Conclusiones.

En este capítulo se presentó una metodología para convertir un modelo en *RPC* a *LD* donde se elabora una red de *LD* para cada color de las transiciones y también se contempló el caso cuando los colores de los lugares de entrada y salida a una transición no son binarios. Asimismo, se mostró el uso funciones paramétricas que pueden ser utilizadas en los *PLC* para que un robot se desplace a una posición específica.

Además de que las *RPC* permiten modelar a los *SMF*, se puede aplicar la teoría del control supervisor para saber si es posible confinar el comportamiento del sistema dentro de la especificación y también para diseñar el controlador. El controlador resultante de aplicar la teoría del control supervisor puede ser implementado en un *PLC* comercial.

El propósito de la metodología aquí presentada es proporcionar al ingeniero una herramienta para implementar controladores construidos con *RPC* a *LD* sin que se introduzcan tantos errores en la programación de un *PLC*, para que éste pueda ser utilizado como un controlador.

7 CONCLUSIONES Y TRABAJO FUTURO.

7.1. Conclusiones.

En esta tesis se abordó el problema del modelado de *SMF* y el problema de control usando la técnica de regulación, la cual busca el diseño de un controlador que permita igualar las salidas del sistema y la especificación. En este trabajo se propuso una metodología de modelado con *RPC*, un conjunto de funciones para facilitar la solución de las ecuaciones en el diseño del supervisor y una metodología para convertir modelos en *RPC* a *LD* con la finalidad de implementar controladores a los *PLC*'s comerciales.

El método propuesto para el modelado ascendente de *SMF* permite obtener modelos compactos que reflejan la arquitectura física del *SMF*. Gracias a la forma en que se diseñaron los módulos propuestos en la metodología de modelado permite que las partes circulen a través de circuitos, lo cual permite encontrar más fácilmente las secuencias en las que se deben de disparar las transiciones para seguir el flujo de trabajo de las partes.

Es importante que los módulos en *RPC* tomen en cuenta las tareas de los elementos de un *SMF*, porque permite conocer el orden de las operaciones y así poder unirlos secuencialmente para poder reflejar la arquitectura física y el flujo de trabajo.

Para resolver más rápidamente las ecuaciones implicadas en el diseño del control supervisor se utilizaron funciones que tiene la característica de ser matrices bisimétricas. Para llevar a cabo dicha labor se propuso un algoritmo para encontrar la matriz π a través de las funciones propuestas en caso de que los bloques matriciales que aparecen en 5.17 preserven la misma dimensión, en caso contrario el algoritmo aun puede aplicarse; pero es necesario desdoblar las funciones.

Con la finalidad de que un controlador diseñado con *RPC* pueda ser implementado en un *PLC* se propuso una metodología para convertir una modelo en *RPC* a *LD*. La conversión de un modelo en *RPC* se realizó asignando una red de *LD* a cada color presente en una transición. Además de contactos y bobinas para

convertir una *RPC* a *LD* es necesario utilizar contadores y comparadores cuando se presenten lugares no binarios. Asimismo el uso de funciones ayuda a reducir el número de elementos empleados en el *LD* para llevar un robot a una posición específica por medio de coordenadas.

7.2. Trabajo Futuro.

Existen muchas áreas de investigación en lo que concierne al modelado de *SMF* con *RPC*, algunas de las cuales pueden ser:

- Abordar el problema del control usando la técnica de regulación; pero con múltiples especificaciones para un *SMF*.
- El análisis del modelo obtenido de la especificación y del sistema, el diseño e implementación del esquema de control cuando el sistema es modelado con *RPC* temporizadas.
- El diseño de un programa capaz de permitir construir modelos con *RPC* y que permita convertir automáticamente el modelo en algún lenguaje propuesto en el *IEC1131 – 3*.
- Recuperación de errores de los *SMF* usando las *RPC*.
- La formalización del uso de las *RPC* para el diseño de observadores.

BIBLIOGRAFÍA

- [1] *International Society for Measurement and Control (ISA). Standar ISA-S88.01, Batch Control, Part I: Models and Terminology*, 1995.
- [2] P. Bhattacharya, S. Jain, and S. Nagpaul, *Basic Abstract Algebra*, 2nd ed. Cambridge University Press, 1994.
- [3] A. Colombo, K. Feldmann, C. Schnur, and T. Stöckel, “Specification, design, and implementation of logic controllers based on colored petri net models and the standar IEC 1131 part i: Specification and design,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 6, pp. 657–665, Nov 1999.
- [4] —, “Specification, design, and implementation of logic controllers based on colored petri net models and the standar IEC 1131 part II: Design and implementation,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 6, pp. 666–674, 1999.
- [5] A. Colombo, J. Martínez, and R. Carelli, “Formal validation of complex production systems using coloured petri nets,” *IEEE Transactions on Control Systems Technology*, pp. 8–13, May 1994.
- [6] A. Colombo, R. Neubert, and B. Süßmann, “A coloured petri net-based approach towards a formal specification of agent-controlled production systems,” *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 6–11, Oct 2002.
- [7] J. Desel and J. Esparza, *Free Choice Petri Nets*. Cambridge University Press, 1995.
- [8] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat, *Practice of Petri Nets in Manufacturing*. Chapman Hall, 1993.

BIBLIOGRAFÍA

- [9] K. Ezpeleta and J. M. Colom, "Automatic synthesis of colored petri nets for the control of FMS," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 327–337, Dec 1997.
- [10] A. Guia, "Petri nets as a discrete event models for supervisory control," Ph.D. dissertation, Rensselaer Polytechnic Institute, July 1992.
- [11] A. Guia and F. Dicesare, "Supervisory design using petri nets," *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 1, pp. 92–97, Dec 1991.
- [12] A. Guia and F. DiCesare, "On the existence of petri net supervisors," *IEEE Conference on Decision and Control*, vol. 4, pp. 3380–3385, Dec 1992.
- [13] —, "Blocking and controllability of petri nets in supervisory control," *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 818–823, Apr 1994.
- [14] —, "Petri net structural analysis for supervisory control," *Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 185–195, Apr 1994.
- [15] K. Jensen, "Coloured petri nets and the invariant method," *Theoretical Computer Science*, vol. 14, pp. 317–336, 1981.
- [16] I. Jiménez, E. López, and A. Ramírez, "Synthesis of ladder diagrams from petri nets controller models," *IEEE International Symposium Intelligent Control, ISIC*, pp. 225–230, 2001.
- [17] R. W. Lewis, *Programming Industrial Control Systems Using IEC 1131-3 Revised Edition*. The Institution of Electrical Engineers, 1998.
- [18] J. Martínez and M. Silva, "A language for the description of concurrent systems modeled by coloured petri nets. application to the control of flexible manufacturing systems," *Languages for Automation*, pp. 369–388, 1985.
- [19] E. L. Mellado, *Introducción a Las Redes de Petri*. Facultad de Ciencias Físico-Matemáticas, 1997.
- [20] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of IEEE*, pp. 541–580, Apr 1989.

BIBLIOGRAFÍA

- [21] M. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*, ser. Intelligent Control and Intelligent Automation. World Scientific, 1998.
- [22] A. E. Nisenfeld, “International society for measurement and control (ISA),” in *Practical Guides for Measurement and Control*, 1996.
- [23] J. L. Peterson, “Petri nets,” *Computing Surveys*, pp. 223–252, Sep 1977.
- [24] P. J. Ramadge and W. M. Wonham, “Supervisory control of a class of discrete event processes,” *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206–230, Jan 1987.
- [25] —, “Modular supervisory control of discrete-event systems,” *Mathematics of Control, Signals, and Systems*, vol. 1, pp. 13–30, 1988.
- [26] P. J. G. Ramadge and W. M. Wonham, “The control of discrete event systems,” *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, Jan 1989.
- [27] A. Ramírez-Treviño and E. López-Mellado, “Qualitative modeling of assembly tasks in robot cells using coloured petri nets,” *Proceedings of Rensseslaer’s Second International Conference on Computer Integrated Manufacturing*, pp. 475–482, May 1990.
- [28] J. I. R. Rangel, “Observability and modular synthesis of petri net models of discrete event systems,” Ph.D. dissertation, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2004.
- [29] J. F. Sánchez-Blanco, A. Ramírez-Treviño, and A. Santoyo, “Multiple specification regulation control in interpreted petri nets,” *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4989–4994, 2004.
- [30] —, “Regulation control in interpreted petri nets using trace equivalence,” *IEEE Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1843–1848, 2004.
- [31] M. Silva, *Las Redes de Petri En la Automática Y la Informática*. AC, 1985.
- [32] M. Silva, J. Martínez, H. Alla, and P. Ladet, “Generalized inverses and the calculation of symbolic invariants of coloured petri nets,” *Technique et Science Informatique (TSI)*, vol. 4, no. 1, pp. 113–126, 1985.



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Control por Regulación en Sistemas de Manufactura Flexible usando
Redes de Petri Coloreadas

del (la) C.

Saúl Alonso NUÑO SÁNCHEZ

el día 27 de Agosto de 2009.

Dra. Ofelia Begovich Mendoza
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

