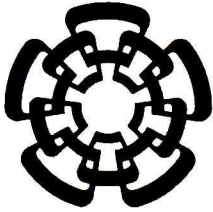


OT-868-SS1

DOM-7015



**Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara**

**Representación y Manipulación
Algebraica de las Creencias, Deseos e
Intenciones de un Agente**

Tesis que presenta:
María Italia Jiménez Ochoa

para obtener el grado de:
Doctora en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Director de Tesis

Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Diciembre de 2014

**CINVESTAV
IPN
ADQUISICION
LIBROS**

Representación y Manipulación Algebraica de las Creencias, Deseos e Intenciones de un agente

RESUMEN

En los sistemas multiagentes interactúan diversos agentes para alcanzar un objetivo en común o resolver tareas complejas. La arquitectura BDI cuyas siglas provienen de las palabras en inglés Belief, Desires, Intentions (BDI) es adecuada para implementar sistemas multiagentes cuando el ambiente no se conoce completamente o cambia con el tiempo. La idea básica de esta arquitectura es representar el estado del agente por medio de las categorías mentales de las creencias, deseos e intenciones. Las creencias representan el conocimiento que el agente tiene sobre el ambiente, sobre sí mismo y sobre otros agentes, los deseos representan los objetivos o las situaciones que el agente quisiera lograr y las intenciones representan las secuencias de acciones que el agente debe realizar para alcanzar uno o más de sus objetivos.

Existen trabajos donde los agentes implementados usando la arquitectura BDI son usados en ambientes cuyo modelo es un sistema de eventos discretos. En este campo, los agentes han sido utilizados para la toma de decisiones y el monitoreo de señales de sistemas. Las decisiones que los agentes hagan, dependen de la información percibida; generalmente incrementando dicha información se realizarán mejores decisiones. Desafortunadamente, el caso cuando el agente conoce toda la información del sistema no es realista, ya que en muchas ocasiones no se cuenta con los sensores para medir toda la información del sistema. Para solucionar este problema, a los agentes se les dota de mecanismos que obtienen un valor para el estado del ambiente a pesar de que la señal percibida no es completa o suficiente para obtener el estado real del ambiente, al mecanismo se le llama función de revisión de creencias.

La función de revisión de creencias genera nuevas creencias dependiendo de la información percibida y de las creencias actuales. La función de revisión de creencias debe evitar conflictos entre las creencias actuales y la información percibida para que el agente realice acciones adecuadas en su entorno y alcanzar uno o más de sus objetivos.

En este trabajo se extiende la arquitectura BDI al ámbito de los Sistemas de Eventos Discretos (SED) modelados con Redes de Petri Interpretadas (RdPI). Las RdPI son utilizadas para modelar el comportamiento del ambiente del agente. Además se introducen las RdPI como herramienta formal para diseñar modelos que representen la función de revisión de creencias ambientales, los deseos e intenciones de un agente. Esos modelos se definen por estructuras dinámicas donde generan y revisan en un tiempo polinomial un tipo de creencia y la secuencia de operaciones que debe realizar el agente. La aproximación propuesta permite al agente ejecutar acciones en el SED sin que se alcancen estados no deseados. Además delimita la cantidad de memoria para representar el conjunto de creencias. Asimismo se realizan cambios mínimos en el conjunto de creencias gracias a que el mecanismo algebraico propuesto efectúa cambios locales. Los generadores de creencias propuestos cumplen con los postulados AGM y satisfacen los axiomas propuestos para la revisión de creencias.

Representation and Algebraic Manipulation of Beliefs, Desires and Intentions of an Agent

ABSTRACT

In multiagent systems several agents are cooperating to achieve a common goal or solving complex task. BDI architecture (Belief, Desires and Intentions) is used to implement multiagent systems working in different and unknown environmental conditions. The BDI architecture represents the agent state by the belief, desires and intentions mental attitudes. Beliefs capture the agent knowledge about the environment state, other agents, and itself. Desires represent agent goals or agent plans. Intentions are the set of actions that the agent performs to meet its goals.

There exists different works where the agent environment is modeled as Discrete Event System. In this field, the agents have been used for to make decisions and the monitoring system signals. Making good decisions, however, depends on the perceived information: in general, as the perceived information is increased, as better decisions are made. Unfortunately, the case when the agent knows the all the system information is not a realistic one, since many system signal cannot be measured. In order to overcome this drawback, BDI agents are endowed with a mechanism to guess the unperceived signals, named the belief revision function.

The belief revision function generates new beliefs depending on the perceived information and the current agent beliefs. The belief revision functions avoid inconsistencies in the current belief data base and the perceived information in order to the agent performs a set of actions to meet its goals.

In this work, the BDI architecture is extended to the scope of Discrete Event Systems (DES) modeled by Interpreted Petri Nets (IPN). The IPN are used for modeling the environment where the agent is working. Moreover the IPN are introduced as formal tools for modeling the environmental belief revision function, desires and intentions of the agent. These models are defined for dynamics models, where in polynomial time they generate beliefs and the sequence of operations that the agent must be perform. This algebraic approach allows to the agent executes actions into the DES without reaching a forbidden states. Moreover, this approach represents belief in bounded structures and every time that new information is perceived by the agent, minimal number of consistent beliefs are generated. Finally, the proposed belief generators meet the axioms and postulates of the AGM theory.

AGRADECIMENTOS

A Dios por guiarme espiritualmente y brindarme la oportunidad de concluir este ciclo de mi vida

Al Dr. Antonio Ramírez Treviño, por su paciencia y dedicación en el desarrollo de este trabajo de investigación

A mi familia por brindarme su apoyo y comprensión

A Luis Ricalde por ser mi soporte y motivación en esta etapa de mi vida

A todos mis amigos y doctores del CINVESTAV unidad Guadalajara que siempre estuvieron alentándome a terminar el doctorado

A CONACYT por su apoyo económico (No. de becario 144099)

..... a todos ellos, muchas gracias.

Índice general

1. Introducción	1
1.1. Panorama general	1
1.2. Revisión de la literatura	3
1.3. Planteamiento del problema	6
1.4. Objetivos generales	6
1.5. Aportaciones	7
2. Conceptos básicos	9
2.1. Redes de Petri	10
2.1.1. Redes de Petri .	10
2.1.2. Redes de Petri Interpretadas	13
2.1.3. Redes de Petri con Ventana de Tiempo	15
2.2. Agentes	16
2.2.1. Agentes	16
2.2.2. Sistemas MultiAgentes	18
2.2.3. Arquitectura Creencias, Deseos e Intenciones (BDI)	19
3. Esquema de la generación de creencias en RdPI	25
3.1. Generador de Creencias Centralizado	26
3.1.1. Ambiente	27
3.1.2. Metodología de modelado en RdPI	27
3.1.3. Percepción	32
3.1.4. Creencias	36
3.1.5. Función de revisión de creencias	37
3.2. Generador de Creencias Distribuido	37
3.2.1. Metodología de Modelado Distribuido en RdPI	38
3.3. Conclusiones .	42

4. Generador de creencias implementado en RdPIVT	43
4.1. Creencias del estado del sistema	44
4.2. Creencias de faltas del sistema	50
4.3. Algoritmo para diseñar la función de revisión de creencias.	58
4.4. Ejemplo	59
4.5. Conclusiones .	62
5. Generador de creencias extendido implementado en RdPIVT	65
5.1. Algoritmo para diseñar la función de revisión de creencias.	73
5.2. Ejemplo	73
5.3. Conclusiones	79
6. Generador de creencias en RdPI no evento detectable	81
6.1. Algoritmo para diseñar la función de revisión de creencias.	85
6.2. Ejemplo	85
6.3. Conclusiones	88
7. Postulados AGM para la revisión de creencias en RdPI	89
7.1. Postulados de revisión para modelos en RdPI evento detectables	94
7.2. Postulados de revisión para modelos en RdPI no evento detectables	100
8. Generador de intenciones	103
8.1. Generador de deseos e intenciones	105
8.1.1. Cálculo de la función Π y las transiciones de disparo del modelo del ambiente del agente	106
8.1.2. Cálculo y prueba de controlabilidad en la secuencia de disparo de transiciones	107
8.1.3. Cálculo de la función parcial H	109
8.1.4. Representación de la función parcial H como una <i>RdPI</i>	110
8.1.5. Definición del conjunto de eventos activos e inactivos	110
8.2. Generador de creencias	111
8.2.1. Cálculo de las posibles creencias iniciales	112
8.2.2. Definición del modelo del observador	113
8.2.3. Cálculo de las creencias equivalentes y matriz de eventos en común	113
8.2.4. Marcado inicial	115

<i>ÍNDICE GENERAL</i>	3
8.2.5. Reglas de disparo	115
8.3. Ejemplo	121
9. Conclusiones y trabajo a futuro	125
9.1. Conclusiones	125
9.2. Trabajo a futuro	126

Capítulo 1

Introducción

1.1. Panorama general

El estudio de los sistemas multiagentes comenzó en el campo de la inteligencia artificial distribuida [44] a mediados de los años 70's. La inteligencia artificial distribuida estudia la distribución de procesos inteligentes a través de entidades independientes llamadas agentes.

Diversas disciplinas como filosofía, matemáticas, economía, psicología, teoría de control, entre otras, han aportado técnicas, puntos de vista y teorías [6], [41] para constituir lo que hoy en día es la inteligencia artificial.

La inteligencia artificial distribuida es muy reconocida en diversas áreas de aplicación como diseño y aplicación de los sistemas de manufactura [4], control autónomo, diagnóstico, robótica. En el área de manufactura, diversos investigadores han aplicado la inteligencia artificial distribuida en la ingeniería concurrente, diseño de ingeniería colaborativa, control, planeación de manufactura y scheduling [42].

Debido a la aplicación multidisciplinaria de la inteligencia artificial distribuida no se ha podido precisar una definición de agentes [13], [7]. Sin embargo, diversos autores [44], [7] lo consideran como una entidad física o virtual capaz de efectuar acciones racionales en un ambiente dependiendo de lo que percibe de su ambiente. Con ello, se ha hecho una clasificación de los tipos de agentes que van desde los más simples hasta los más complejos. Esta clasificación se ha hecho en base a las características del agente, tales como:

- Comunicativo: Capaz de interactuar con otros agentes a través de la comunicación.
- Reactivo. Capaz de reaccionar ante los cambios en su ambiente.
- Deliberativo. Capaz de elegir una tarea a efectuar para cumplir con un objetivo.

Los agentes más sencillos son llamados agentes reactivos ya que simplemente usan un mecanismo que reacciona ante los cambios de su medio ambiente sin comunicarse con otros agentes. En contraste con los agentes sencillos, están los agentes deliberativos o

cognitivos ya que poseen un conocimiento explícito de su medio ambiente y pueden incluir mecanismos para interactuar con otros agentes.

Para implementar las características de un agente es necesario considerar una arquitectura que capture los componentes del agente como son: percepción, ejecución, aprendizaje, razonamiento y control, además de representar la conectividad en los componentes. Los agentes reactivos puede tener solamente un subconjunto de tales componentes mientras que los agentes complejos pueden necesitar un subconjunto mayor.

La arquitectura interna de un agente es esencialmente la descripción, el flujo de control y las operaciones que pueden ejecutar sus componentes [45]. De igual forma que los agentes, las arquitecturas se han clasificado en cuatro categorías: deliberativa, reactiva, colaborativa e híbridas.

Cada una de estas arquitecturas es ideal de acuerdo a los requerimientos para su aplicación. Sin embargo, hoy en día se ha difundido el uso de las arquitecturas deliberativas ya que le permite al agente tener una representación parcial del medio ambiente. Los agentes que son implementados con este tipo de arquitecturas son comúnmente llamados agentes deliberativos o cognitivos.

Una de las arquitecturas deliberativas más conocidas es la arquitectura *BDI*, cuyas siglas provienen de las palabras Belief, Desires, Intentions (*BDI*) [21], [11]. La idea básica de esta arquitectura es describir el estado interno de un agente por medio de un conjunto de categorías mentales (creencias, deseos e intenciones) y definir un mecanismo de control en el cual, el agente seleccione su curso de acción basado en esas representaciones. La arquitectura *BDI* tiene una representación explícita acerca de las creencias, deseos e intenciones de un agente, donde las creencias representan el estado de información del agente es decir, sus conocimientos sobre el ambiente, sobre sí mismo y sobre otros agentes, los deseos representan el estado de motivación del agente, esto es, los objetivos o las situaciones que el agente quisiera lograr y las intenciones representan el estado deliberativo del agente esto es, las secuencias de acciones que un agente puede realizar para alcanzar uno o más de sus deseos.

Los agentes que son implementados con la arquitectura *BDI* en ambientes de sistemas de manufactura flexible son usados, por ejemplo, para la toma de decisiones y el monitoreo de señales de sistemas. Con una buena decisión se logran buenas políticas de producción, tales como justo a tiempo, balance de líneas, entre otras. Sin embargo, las decisiones dependen de la información percibida: incrementando dicha información se generan buenas decisiones. Desafortunadamente, el caso cuando el agente conoce toda la información del sistema no es completamente factible. Esto se debe a que varias señales del sistema no pueden ser medibles por el costo de los sensores o porque no existe la tecnología adecuada para medir ciertas variables del sistema. Para solucionar este problema, los agentes cognitivos están dotados de un mecanismo que estima la señal no percibida, llamado función de revisión de creencias. La función de revisión de creencias es un mecanismo que genera creencias dependiendo de la información percibida, y de las creencias actuales. Este mecanismo también depende en gran parte del modelo del sistema. Este modelo debe ser congruente con el sistema ya que es el conocimiento inicial que el agente tiene acerca del sistema. Actualmente, todas las formalizaciones de la función de revisión de creencias

carecen de una metodología para modelar el ambiente del agente.

Un gran número de trabajos han formalizado las creencias, deseos e intenciones de un agente así como el mecanismo de revisión de creencias usando estructuras temporales ramificadas, lógica de primer orden y lógica modal. Sin embargo, el diseño de sistemas basados en agentes se ha beneficiado muy poco mediante esas formalizaciones debido a que carecen de un análisis de completitud de tales expresiones lógicas o de procedimientos de decisión para la prueba de satisfacibilidad o validación de fórmulas.

Este trabajo aborda la función de revisión de creencias en el caso que un agente interactúe en un ambiente que puede ser modelado con Redes de Petri Interpretadas. Esta investigación utiliza una metodología para modelar las creencias del agente, garantizando un modelo congruente con el ambiente. Además, el mecanismo de revisión de creencias es propuesto como un sistema dinámico que genera creencias a partir de la información percibida, del modelo del ambiente y de las creencias actuales. Este mecanismo permite delimitar la cantidad de memoria para representar las creencias y efectúa cambios mínimos en las creencias gracias a que el mecanismo algebraico origina cambios locales.

1.2. Revisión de la literatura

La arquitectura *BDI* basada en las actitudes de creencias, deseos e intenciones se ha visto como un modelo adecuado para diseñar agentes inteligentes. Dichas actitudes representan la información acerca del ambiente, los objetivos a cumplir y las opciones disponibles para elegir un curso de acción, respectivamente.

Diversos trabajos han sido desarrollados para formalizar las creencias, deseos e intenciones del agente y muestran cómo esas actitudes determinan las acciones de un agente. Por ejemplo, la arquitectura Máquina Inteligente con Recursos Acotados (Intelligent Resource-bounded Machine Architecture IRMA) propuesto por Bratman [9], [10] es una clásica arquitectura descrita en alto nivel. Esta metodología describe cómo un agente selecciona su curso de acción basado en representaciones explícitas de sus percepciones, creencias, deseos e intenciones. La arquitectura incorpora un número de módulos que representan los planes y las intenciones del agente, un mecanismo de razonamiento para determinar el curso de acción que el agente puede tomar para alcanzar un objetivo, un analizador de oportunidades, un proceso de filtro y un procedimiento de deliberación. Tan pronto como las creencias del agente son actualizadas, el analizador de oportunidades es capaz de sugerir cursos de acción en base a las creencias del agente. Todas las opciones disponibles son determinadas a través del proceso de filtro, donde todas las opciones son sometidas a una prueba para verificar la consistencia con las intenciones actuales del agente. Aquellas opciones que aprueben la condición de consistencia, son pasadas al proceso de deliberación, que consiste en elegir una nueva intención modificando la estructura de las intenciones.

Algunos aspectos filosóficos de la teoría de Bratman fueron formalizados por Cohen y Levesque [11]. En su formalismo, establece las condiciones bajo las cuales un agente está enfocado a cumplir sus objetivos dando énfasis al proceso de revisión de las intenciones. Además contempla el caso en que el agente puede abortar sus objetivos, ya sea cuando el

agente cree haber alcanzado dicho objetivo o cree que el objetivo no puede ser alcanzado. El último caso se considera que el agente puede evitar aquellos cursos de acciones que pueden tener efectos no deseados que impiden alcanzar los objetivos.

Por otro lado, Rao y Georgeff [21],[23],[22] presentan un formalismo alternativo para la arquitectura *BDI*. En ese formalismo, hay tres elementos importantes. En el primero las intenciones son consideradas como parte del balance que debe existir entre las creencias y deseos, las creencias son modeladas con estructuras ramificadas. Este modelo permite definir diferentes estrategias para alcanzar un objetivo con respecto a las intenciones del agente. El segundo elemento permite distinguir entre las opciones que el agente tiene sobre las acciones que puede ejecutar y sus creencias acerca de las acciones que considera posibles. En el primer caso, el agente puede elegir su curso de acción y en el segundo caso, el ambiente lo determina. Finalmente, la metodología indica una interrelación entre las creencias, deseos e intenciones que permite evitar algunos problemas que se han detectado en los trabajos previos con respecto a la formalización de la noción de intenciones. Esto le permite al agente evitar aquellos cursos de acción en los cuales pueden tener efectos no deseados al momento de intentar alcanzar sus objetivos.

El formalismo propuesto por Rao y Georgeff es uno de los más populares debido a su gran diversidad de aplicaciones y por su elegante representación. Sin embargo, algunos autores, [3], [16], consideran que existe dificultad en implementarlo en un modelo computacional debido a la complejidad en la semántica que se utiliza para describir las creencias, deseos e intenciones.

En paralelo a estos trabajos, se han desarrollado diversos trabajos [18], [27], [20], enfocados al estudio del mecanismo de la revisión de creencias, deseos e intenciones.

Para el caso de la revisión de creencias, la razón de requerir una función de revisión de creencias se debe al hecho de que se tiene la posibilidad de conflicto entre las creencias actuales con las que se genere cada vez que se percibe nueva información. Agregando simplemente la nueva información percibida al conjunto de creencias podría causar inconsistencias, lo cual no es lo deseable. Entonces, se necesita revisar el conjunto de creencias para hacerlo consistente.

Existen diversos trabajos que abordan este problema de inconsistencias en el conjunto de creencias. El más importante es el presentado por [2] ya que introduce el mecanismo de revisión de creencias hoy llamado *AGM*. En este enfoque tanto el conjunto de creencias, (llamado estado epistémico del agente) como la información percibida se representa por un conjunto de fórmulas en un lenguaje lógico y se define un operador de revisión descrito mediante un conjunto de postulados basados en la contracción y expansión de creencias. Los postulados de expansión están enfocados a delimitar las condiciones que permiten agregar cada vez que el agente percibe nueva información de su ambiente. Los postulados de contracción delimitan las condiciones que permitan eliminar una creencia del conjunto de creencias. Mediante estos postulados se garantiza que el operador de revisión genera cambios consistentes en el conjunto de creencias. Sin embargo, el conjunto de creencias no está acotado por lo que hace compleja su implementación.

Algunos otros trabajos han formalizado y mejorado el enfoque propuesto de *AGM* [5],

[15], [18]. En el enfoque de Boutiliers [5], el estado epistémico del agente se representa por mundos posibles. Los mundos posibles son un conjunto de fórmulas consistentes que son ordenados mediante una relación de orden parcial y el operador de revisión es llamado operador de revisión natural. Ese operador mapea un conjunto de mundos posibles ordenados y una información percibida del medio ambiente a un mundo posible ordenado revisado. La revisión del mundo posible ordenado es el resultado de efectuar un número mínimo de cambios en el estado epistémico del agente.

En el mismo contexto, el enfoque de Freund y Lehmann [15] trabaja con conjuntos de creencias y usan el operador de revisión natural propuesto por Boutilier. Anexa un postulado que incluye el caso cuando la información percibida es inconsistente con el conjunto de creencias, en tal caso, ellos proponen no retener ninguna creencia y comenzar con un conjunto de creencias vacío.

En general, los trabajos anteriores no prestan atención al número de creencias en el conjunto de creencias. Para resolver este problema, se introduce el concepto de base de creencias y la revisión de la base de creencias permitiendo una representación finita de creencias. Varios trabajos [19], [24] usan la base de creencias e introducen el concepto de revisión local, considerando que no todas las creencias del agente son relevantes para una operación de revisión de creencias. El uso de revisión local mejora la ejecución de los algoritmos propuestos para la revisión de creencias.

Hasta el momento, los trabajos mencionados no contemplan algún método para modelar el conocimiento inicial que el agente tiene acerca de su ambiente, es decir, estos enfoques nacen con la base de creencias ya establecidas. Asimismo no existe una manera sistemática para diseñar la función de revisión de creencias y es una tarea difícil de probar que este conjunto cumple con sus especificaciones. Además, muy pocos trabajos contemplan procedimientos para verificar la satisfacibilidad y completitud de las fórmulas que representan las creencias del agente y la información percibida por el ambiente.

Este trabajo propone un nuevo enfoque para definir e implementar la función de revisión de creencias donde el agente interactúa en ambientes que pueden ser modelados por Redes de Petri Interpretadas (*RdPI*), es decir, sistemas dinámicos con una estructura fija donde el valor de algunas variables de sistema es desconocido o cambia con el tiempo. Las *RdPI* son una herramienta formal y adecuada para modelar el ambiente del agente, por tanto, las creencias de tipo ambiental del agente son representadas por la estructura y el marcado del modelo del ambiente y el mecanismo para generar nuevas creencias se representa por las reglas de disparo de dicho modelo. Además, este trabajo aborda la representación de la generación de creencias en un sistema multiagente donde se propone una metodología de modelado para distribuir el comportamiento del ambiente del agente. La ventaja de este planteamiento es que las creencias de los agentes son acotadas y la revisión de creencias actúa localmente, es decir, en base a la percepción de señales de entrada, creencias actuales y la estructura de la *RdPI* se realizan mínimos cambios para generar nuevas creencias. Además, este trabajo permite el uso del álgebra lineal para introducir el concepto de error de creencia (error de estado) el cual es utilizado para implementar controladores y sistemas confiables. La ventaja del enfoque algebraico es que este trabajo cumple con los postulados *AGM* [2] y satisface con los axiomas propuestos para las creencias [28], [45]:

el agente nunca debe contradecir lo que realmente conoce del sistema y que conforme el agente está activo percibiendo y actuando dentro del ambiente, la revisión de creencias debe proporcionar una información cada vez más exacta del ambiente.

1.3. Planteamiento del problema

Este trabajo consiste en obtener modelos algebraicos para representar la generación de creencias de un agente que interactúa en un ambiente de Sistemas de Eventos Discretos (*SED*). Estos modelos algebraicos deben incluir un mecanismo de revisión de creencias para evitar inconsistencias en las creencias del agente con las siguientes características:

1. Las creencias que se generen deben ser consistentes antes y después de la revisión.
2. Debe ser mínimo el cambio de creencias durante la revisión de creencias.
3. Cumplir con los postulados llamados *AGM* propuestos por Alchourron, Makinson y Ganderfors.
4. Disponer de una ecuación de error que nos permita cuantificar la calidad de las creencias.

1.4. Objetivos generales

Los objetivos del presente trabajo de investigación son:

- Demostrar que las redes de Petri son una herramienta formal y adecuada para representar las creencias de un agente.
- Proponer una metodología de modelado distribuido para diseñar el ambiente de un sistema multiagente
- Adaptar los observadores y diagnosticadores diseñados con redes de Petri para ser utilizados como mecanismo para generar creencias del ambiente de un agente.
- Obtener un modelo de observador extendido utilizando la teoría de observador y diagnosticador para representar el mecanismo de la función de revisión de creencias.
- Mostrar que el mecanismo de la función de revisión de creencias modelado con redes de Petri Interpretadas cumple con los axiomas de la función de revisión de creencias.
- Demostrar que la representación de la función de revisión de creencias modelado con redes de Petri Interpretadas cumple con los postulados propuestos por Alchourron, Makinson y Ganderfors.

- Adaptar la teoría de control por regulación con redes de Petri para ser utilizada como mecanismo para generar intenciones en base a las creencias del ambiente y deseos del agente.

1.5. Aportaciones

Las aportaciones de este trabajo de investigación son:

- Se demuestra que las redes de Petri constituyen una herramienta formal y adecuada para representar las creencias de un agente que interactúa en un ambiente en Sistemas de Eventos Discretos.
- Se diseña una metodología para el modelado distribuido, en redes de Petri Interpretadas, del sistema que representa el ambiente y la base de creencias del agente.
- Definición del mecanismo de la revisión de creencias bajo dos enfoques: centralizado y distribuido, los cuales están compuestos por diversos sistemas dinámicos enfocados a generar diferentes tipos de creencias acerca del ambiente del agente.
- Tres esquemas de la generación de creencias:
 - El primer esquema está compuesto por dos sistemas dinámicos para generar creencias del ambiente del agente. El primer sistema dinámico llamado observador, es una copia del modelo del sistema con estructura adicional para generar creencias acerca del estado del sistema y el segundo sistema dinámico, encargado de generar creencias de las faltas del sistema, es llamado diagnosticador y es una copia del modelo del observador con funciones de tiempo adicionales.
 - El segundo esquema lo comprende un sistema dinámico llamado observador extendido donde se fusiona la teoría del observador y diagnosticador, es decir, el sistema dinámico es una copia del modelo del sistema con estructura y funciones de tiempo adicionales para generar creencias del estado y faltas del sistema.
 - El tercer esquema es un sistema dinámico llamado observador para generar creencias en ambientes que son modelados en Redes de Petri Interpretadas y que tienen la propiedad no evento detectable.
- Definición de algoritmos polinomiales para diseñar cada uno de los esquemas de generación de creencias, propuestos en este trabajo.
- Definición de conceptos de la revisión de creencias desde un enfoque algebraico: creencias, base de conocimiento, operador de revisión de creencias y las operaciones de cambios en las creencias.
- Demostración de que la propiedad evento detectabilidad en el modelo en *RdPI* del ambiente del agente, permite que, conforme el agente interactúa con el sistema, las

creencias del agente mejoran y tienden al estado del sistema, es decir, las creencias llegan a ser hechos.

- Demostración de que la propiedad no evento detectabilidad en el modelo en *RdPI* del ambiente del agente, permite que, conforme el agente interactúa con el sistema no contradice con lo que percibe del ambiente.
- Demostración que el mecanismo de la función de revisión de creencias modelado con redes de Petri Interpretadas cumple con los axiomas que cumple cualquier operador de revisión de creencias
- Demostración que la representación algebraica de la función de revisión de creencias (modelado con redes de Petri Interpretadas) cumple con los postulados de revisión, propuestos por Alchourron, Makinson y Ganderfors.
- Adicionalmente, si los modelos de generación de creencias se aplican a *RdPI* observables, entonces el generador de creencias se convierte en un observador en el sentido clásico de la teoría de control, donde los métodos propuestos mejoran la velocidad de convergencia de los observadores.
- Demostración que la teoría de control por regulación se puede utilizar para representar los deseos e intenciones de un agente mediante el modelo de referencia y el modelo del controlador por regulación ya que el modelo de referencia se considera como los objetivos que el agente debe alcanzar y la ley de control es el curso de acción que debe seguir el agente para lograr dichos objetivos.

Capítulo 2

Conceptos básicos

Resumen: Este capítulo presenta los conceptos básicos de agentes, multiagentes, la arquitectura *BDI* así como los postulados propuestos por Alchourron, Ganderfors y Makinson, llamado *AGM*. Además, se definen los conceptos básicos relacionados a las redes de Petri (*RdP*), redes de Petri Interpretadas (*RdPI*) y una extensión de las *RdPI* que incluye ventana de tiempo (*RdPIVT*). Estos conceptos son útiles porque las creencias y el mecanismo para generar creencias de un agente que interactúa en un ambiente de Sistemas de Eventos Discretos (*SED*) son modelados con *RdPIVT*.

2.1. Redes de Petri

Las Redes de Petri (*RdP*) han demostrado ser una potente herramienta de modelado para los Sistemas de Eventos Discretos (*SED*) debido a que capturan la mayoría de las características de este tipo de sistemas como la sincronización y el paralelismo de procesos, el paso de mensajes, la exclusión mutua, precondiciones y poscondiciones en la ocurrencia de eventos, entre otras.

Las principales virtudes de las *RdP* son su representación gráfica intuitiva y una sólida base matemática para el análisis de sus propiedades [32]. Las *RdP* se usan en los sistemas de manufactura flexible, sistemas distribuidos, protocolos de comunicación, etc.

2.1.1. Redes de Petri

Definición 2.1 *La estructura de una RdP, G , es una 4-tupla $G = (P, T, I, O)$ donde,*

- $P = \{p_1, p_2, \dots, p_n\}$ es un conjunto finito de elementos llamados lugares, los cuales se representan gráficamente mediante círculos.
- $T = \{t_1, t_2, \dots, t_m\}$ es un conjunto finito de elementos llamados transiciones, las cuales se representan mediante barras o segmentos de línea.
- $I : P \times T \longrightarrow \mathbb{Z}^+$ es una función que representa la relación entrada que guardan los lugares con respecto a las transiciones. Gráficamente se presenta mediante arcos dirigidos que van desde los lugares a las transiciones.
- $O : P \times T \longrightarrow \mathbb{Z}^+$ es una función que representa la relación de salida que guardan los lugares con respecto a las transiciones. Gráficamente se representa con arcos dirigidos que van de las transiciones a los lugares.

Note que $P \cap T = \emptyset$ y que nunca existe un arco entre nodos del mismo tipo, por lo cual se dice que una *RdP* es un grafo bipartito.

El símbolo \mathbb{Z}^+ representa al conjunto de los enteros no-negativos. Si $I : P \times T \longrightarrow \{0, 1\}$ y $O : P \times T \longrightarrow \{0, 1\}$, es decir, todos los arcos tienen peso 1, se dice que la estructura de *RdP* es ordinaria.

La estructura G de una *RdP* también se puede representar mediante las siguientes matrices:

- La matriz de incidencia previa $C^+ = [c'_{ij}]_{m \times n}$, donde $c'_{ij} = I(p_i, t_j)$,
- La matriz de incidencia posterior $C^- = [c^-_{ij}]_{m \times n}$, donde $c^-_{ij} = O(p_i, t_j)$,

- La matriz de incidencia $C = [c_{ij}]_{m \times x}$, donde $c_{ij} = c^-_{ij} - c^+_{ij}$.

Definición 2.2 Sea $t_j \in T$, entonces

$$\bar{\cdot}(t_j) = \{p_i \in P \mid I(p_i, t_j) \neq 0\} \quad \text{y} \quad (t_j)^\bullet = \{p_i \in P \mid O(p_i, t_j) \neq 0\}$$

son los conjuntos de lugares de entrada y salida de la transición t_j , respectivamente.

De manera similar, si $p_i \in P$ entonces

$$\bullet(p_i) = \{t_j \in T \mid O(p_i, t_j) \neq 0\} \quad \text{y} \quad (p_i)^\circ = \{t_j \in T \mid I(p_i, t_j) \neq 0\}$$

son los conjuntos de transiciones de entrada y salida del lugar p_i , respectivamente.

El estado del sistema se representa mediante el marcado de la *RdP*, el cual se define de la siguiente forma:

Definición 2.3 El marcado de una *RdP* es una función $M : P \longrightarrow (\mathbb{Z}^+)^n$ que asigna un número entero no negativo a cada lugar de la red, es decir un número dado de marcas.

Gráficamente cada marca se representa mediante un círculo oscuro en el interior del lugar a que ha sido asignada. Así, una *RdP* está compuesta de una estructura y de un marcado, la primera indica las relaciones entre los elementos del sistema, mientras que el último representa el estado de éste.

Definición 2.4 Una red con marcado inicial se llama sistema de *RdP*, red o simplemente *RdP* y se denota por $N = (G, M_0)$, donde G es la estructura de la *RdP* y M_0 es su marcado inicial o distribución inicial de marcas. Una *RdP* es ordinaria si su estructura de *RdP* es ordinaria.

La evolución del estado de un SED está representada por la evolución del marcado de la *RdP* que lo modela de acuerdo a las siguientes reglas:

1. Regla de habilitación: una transición t_j está habilitada (puede dispararse) en un marcado M_k si y sólo si $\forall p_i \in P, M_k(p_i) \geq I(p_i, t_j)$, es decir, si todos sus lugares de entrada contienen el número suficiente de marcas.
2. Regla de disparo: una transición habilitada t_j puede dispararse, en cuyo caso se alcanza un nuevo marcado M_{k+1} , el cual puede ser calculado mediante la ecuación de estados de una *RP*:

$$M_{k+1} = M_k + C \cdot v \quad (2.1)$$

donde v es el vector de disparo definido de la siguiente forma:

$$v(i) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{en caso contrario} \end{cases}$$

es decir el vector de disparo contiene un 1 en la posición correspondiente a la transición que se dispara y 0 en el resto de las posiciones. En este caso se dice que M_{k+1} es un *marcado alcanzable* desde M_k a través del disparo de t_j y se representa por $M_k \xrightarrow{t_j} M_{k+1}$.

La ecuación 2.1 es conocida como la ecuación de estados de la *RdP*.

Definición 2.5 Una secuencia de disparo $\sigma = t_i t_j \dots t_k$ en M_i es una secuencia de transiciones tal que

$$M_i \xrightarrow{t_i} M_j \xrightarrow{t_j} \dots M_k \xrightarrow{t_k} M_r$$

o por simplicidad, $M_i \xrightarrow{\sigma} M_r$, donde cada transición t_u está habilitada en el marcado M_u .

Definición 2.6 Sea $\sigma = t_i t_j \dots t_k$ una secuencia de disparo de transiciones. El vector de Parikh $\vec{\sigma} : T \rightarrow (\mathbb{Z}^+)^m$ mapea cada transición $t_i \in T$ al número de sus ocurrencias en σ , donde $m = |T|$.

Definición 2.7 El conjunto de alcanzabilidad de una *RdP*, $R(G, M_0)$, es el conjunto de todos los marcados alcanzables a partir del marcado inicial de la red.

El conjunto de alcanzabilidad de una *RdP*, es el conjunto de todos los posibles marcados que se pueden obtener evaluando la ecuación 2.1, iniciando con M_0 y disparando exclusivamente transiciones habilitadas en cada momento.

Definición 2.8 Un marcado $M_k \in R(G, M_0)$ es un marcado alcanzable si se alcanza desde cualquier marcado $M_j \in R(G, M_0)$, es decir

$$\forall M_j \in R(G, M_0), \exists \sigma \text{ tal que } M_j \xrightarrow{\sigma} M_k$$

Propiedades de las redes de Petri

Las propiedades de las *RdP* se pueden dividir en dos tipos: propiedades estructurales, las cuales sólo dependen de las relaciones entre los nodos de la red (es decir de la estructura de la red); y propiedades dinámicas, las cuales dependen del marcado inicial M_0 . A continuación se mencionan algunas de estas propiedades, las cuales son de interés para el presente trabajo (otras propiedades de las *RdP* se pueden consultar en [43] y [12]).

Propiedades estructurales

- **Repetitividad.** Una *RdP* es repetitiva si y sólo si existe un vector $X \in \mathbb{N}^n$ tal que $CX = \vec{0}$, es decir, una *RdP* es repetitiva si el marcado inicial puede volver a ser alcanzado. Al vector X se le llama *t-semiflujo*.
- **Conservatividad.** Una *RdP* es conservativa si y sólo si existe un vector $Y \in \mathbb{N}^m$ tal que $Y^T C = \vec{0}$, es decir, una *RdP* es conservativa si el número de marcas total de la red se mantiene constante para cualquier marcado alcanzado. Al vector Y se le llama *p-semiflujo*.

Propiedades dinámicas

- **Vivacidad.** Una transición $t_k \in T$ es viva, para un marcado M_0 , si $\forall M \in \mathbf{R}(G, M_0)$, $\exists M_k$ sucesor de M a partir del cual t_k esté habilitada. La red es viva si todas sus transiciones son vivas, es decir una red es viva si todas sus transiciones pueden llegar a dispararse.
- **Acotamiento.** Una red R es b -acotada (b -limitada) para un marcado M_0 , si $\forall P_i \in P$, $\exists b \in \mathbf{N}$, tal que $\forall M_k \in \mathbf{R}(G, M_0)$, se cumple que $M_k(P_i) \leq b$, es decir si el número total de marcas en la red no crece infinitamente.
- **Binaridad.** Una red R es binaria (1-acotada) para un marcado M_0 , si el número de marcas de cualquiera de sus lugares es igual a cero o uno en cualquier marcado alcanzable desde M_0 . Esto es, $\forall M \in \mathbf{R}(G, M_0)$ y $\forall P_i \in P$ $M(P_i) \in \{0, 1\}$.

2.1.2. Redes de Petri Interpretadas

Las *RdP Interpretadas (RdPI)* son una extensión a la definición de las *RdP*, las cuales permiten la representación de señales de entrada y salida para un *SED* [33],[32].

Definición 2.9 Una Red de Petri Interpretada (*RdPI*) es una 4-tupla $Q = (N, \Sigma, \lambda, \varphi)$, donde

- $N = (G, M_0)$ es un sistema de *RdP* con marcado inicial M_0 .
- $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_r\}$ es el alfabeto de entrada de la red, donde cada σ_i es un símbolo de entrada.
- $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$ es la función de etiquetado de las transiciones con la siguiente restricción:
 $\forall t_j, t_k \in T, j \neq k$, si $I(p_i, t_j) = I(p_i, t_k) \neq 0$ y ambos $\lambda(t_j) \neq \varepsilon, \lambda(t_k) \neq \varepsilon$, entonces $\lambda(t_j) \neq \lambda(t_k)$;
 donde ε representa un evento interno del sistema.
- $\varphi : \mathbf{R}(G, M_0) \rightarrow (\mathbb{Z}^+)^q$ es la función de salida que asocia a cada marcado en $\mathbf{R}(G, M_0)$ un vector en $(\mathbb{Z}^+)^q$

Comentario 2.1 Las siguientes suposiciones se utilizarán a lo largo de este trabajo.

1. Para enfatizar el hecho de que existe un marcado inicial en una *RdPI*, se escribirá (Q, M_0) en vez de $Q = (N, \Sigma, \lambda, \varphi)$, para denotar a una *RdPI*

2. Cuando la función φ es una matriz $q \times n$ donde q es el número de lugares que representan estados medibles en el SED y n es el número de lugares en el modelo (Q, M_0) , cada columna de esta matriz es un vector elemental o nulo. Si el símbolo de salida i se presenta (se enciende) cada vez que $M(p_j) \geq 1$, entonces $\varphi(i, j) = 1$, de lo contrario $\varphi(i, j) = 0$.
3. Transiciones equivalentes no son permitidas, esto es, suponga que $\forall t_i, t_j$ tal que $t_i \neq t_j$, $\lambda(t_i) = \lambda(t_j)$, esto cumple que $C(\bullet, i) \neq C(\bullet, j)$. Este tipo de transiciones son redundantes.
4. Note que por definición de λ , las RdPI son deterministas en relación a las transiciones etiquetadas, es decir, dos transiciones con la misma señal de entrada asociada (diferente de ε) no puede tener los mismo lugares de entrada [30]. Sin embargo pueden ser no deterministas en relación a la transiciones no etiquetadas (aquellas t_j tal que $\lambda(t_j) = \varepsilon$).

Los alfabetos de entrada y salida se pueden ver como los conjuntos de señales discretas de los actuadores y de los sensores del sistema que se está modelando, respectivamente. En este contexto, se puede hacer una distinción entre nodos medibles y no medibles y entre transiciones controlables y no controlables, como indican las siguientes definiciones.

Definición 2.10 Sea (Q, M_0) una RdPI. Si $\lambda(t_i) \neq \varepsilon$, para una $t_i \in T$, entonces se dice que t_i es una transición manipulable; en otro caso se dice no manipulable. Un lugar $p_i \in P$ se dice medible si la i -ésima columna de φ es no nula, es decir, $\varphi(\bullet, i) \neq 0$, en otro caso se dice no medible.

Note que $T = T_m \cup T_{nm}$ y $P = P_m \cup P_{nm}$ con $T_m \cap T_{nm} = \emptyset$ y $P_m \cap P_{nm} = \emptyset$, donde T_m y P_m son los conjuntos de transiciones manipulables y lugares medibles, respectivamente; mientras que T_{nm} y P_{nm} son los conjuntos de transiciones no manipulables y lugares no medibles, respectivamente.

Gráficamente, los lugares no medibles y transiciones no manipulables se rellenan con color oscuro.

Las reglas de evolución de una RdPI son similares a las reglas de una red ordinaria y se enuncian de la siguiente forma:

1. **Regla de habilitación:** una transición $t_j \in T$ de una RdPI está habilitada en un marcado M_k si

$$\forall p_i \in P, M_k(p_i) \geq I(p_i, t_j).$$

2. **Regla de disparo:** Si la transición t_j está habilitada y el símbolo de entrada $\lambda(t_j) = a_i \neq \varepsilon$ asociado a ella está presente, entonces t_j debe dispararse. En el caso que $\lambda(t_j) = \varepsilon$, entonces t_j puede ser o no disparada. En ambos casos, al ser disparada t_j en un marcado M_k , un nuevo marcado M_{k+1} es alcanzado, el cual puede ser calculado usando la siguiente ecuación de estados:

$$\begin{aligned} M_{k+1} &= M_k + Cv_k \\ y_k &= \varphi(M_k) \end{aligned} \quad (2.2)$$

donde C y v_k son definidos como en la RdP y $y_k \in (\mathbb{Z}^+)^q$ es el k -ésimo vector de salida de la $RdPI$.

Definición 2.11 Una secuencia de disparo de transiciones en una $RdPI$ (Q, M_0) es una secuencia de transiciones $\sigma = t_i t_j \dots t_k$ tal que $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \dots M_w \xrightarrow{t_k} \dots$

Definición 2.12 El conjunto de todas las secuencias de disparo de transiciones de una $RdPI$ es llamado lenguaje de disparo:

$$\mathcal{L}(Q, M_0) = \{\sigma \mid \sigma = t_i t_j \dots t_k \wedge M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \dots M_w \xrightarrow{t_k} \dots\}.$$

Definición 2.13 El lenguaje de entrada de una $RdPI$ es el conjunto

$$\mathcal{L}_{in}(Q, M_0) = \{\lambda(t_i)\lambda(t_j)\dots\lambda(t_k) \mid t_i t_j \dots t_k \in \mathcal{L}(Q, M_0)\}$$

mientras que el lenguaje de salida es

$$\mathcal{L}_{out}(Q, M_0) = \{\varphi(M_0)\varphi(M_1)\dots\varphi(M_w)\dots \mid M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \dots M_w \xrightarrow{t_k} \dots \wedge t_i t_j \dots t_k \in \mathcal{L}(Q, M_0)\}$$

Definición 2.14 Sea (G, M_0) una $RdPI$ y $M(p_j)$ cualquier marcado de un lugar p_j de (G, M_0) . El conjunto de s ecuaciones

$$LCM = \left\{ \sum_{j=1}^n \gamma_j^i M(p_j) = k_i \mid i \in [1, \dots, s] \text{ y } \gamma_j^i \in \mathbb{Z}^+ \right\}$$

forman un conjunto de leyes conservativas de marcado (LCM) si $\forall \gamma_j^i \neq 0$ cumple que k_i/γ_j^i es un valor entero y $\forall p_j \in P_{nm}$ están en al menos una ecuación del conjunto LCM .

El conjunto LCM se puede redefinir como

$$\Gamma M = K$$

2.1.3. Redes de Petri con Ventana de Tiempo

Las $RdPI$ con Ventana de Tiempo $RdPIVT$ son una extensión a la definición de las $RdPI$, las cuales permiten asociar tiempo a las transiciones para representar el período de tiempo en que éstas deben dispararse.

Definición 2.15 Una *RdPIVT* es la 4-tupla $Q_{\pm}^+ = (RdPI, H^-, H^+, T_k^L)$, donde:

- *RdPI* es una Red de Petri Interpretada.
- $H^- : T \rightarrow \mathbb{R}^+$ es una función que asocia a cada transición un número real positivo que representa el mínimo tiempo de disparo de una transición después de que ésta se habilita.
- $H^+ : T \rightarrow \mathbb{R}^+$ es una función que asocia a cada transición un número real positivo que representa el máximo tiempo de disparo de una transición después de que ésta se habilita.
- $T_k^L : T \rightarrow \{\text{normal}, \text{falta}\}$ es una función que mapea cada transición $t_j \in T$ a una etiqueta tipo normal o falta como sigue:

$$T_k^L(t_j) = \begin{cases} \text{normal} & \text{Si } t_j \text{ se dispara en el tiempo } \tau_{t_j} \\ & \text{y } H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j} \\ \text{falta} & \text{otro caso} \end{cases}$$

donde χ_{t_j} es el tiempo cuando t_j se habilita en el mercado M_k y τ_{t_j} el tiempo actual.

En otras palabras, las funciones H^- y H^+ representan el intervalo de tiempo cuando una transición habilitada debe ser disparada. La función T_k^L etiqueta a una transición $t_j \in T$ como normal o falta en relación a las funciones H^- y H^+ y del tiempo en que se dispara.

Por definición, todas las transiciones son etiquetadas como normal, esto es

$$T_0^L(t_j) = \text{normal}, \forall t_j \in T$$

2.2. Agentes

2.2.1. Agentes

Debido a la naturaleza interdisciplinaria de los agentes es difícil precisar la definición de los sistemas multiagentes. Sin embargo varios autores [13], [7] definen a un agente como una entidad física (un robot, un avión o un auto) o virtual (componente de software y un módulo de software) con las siguientes características:

- Capaz de actuar en un ambiente. El concepto de acción, el cual es fundamental en los sistemas multiagentes, está basado en el hecho de que los agentes llevan a cabo acciones con el fin de modificar su ambiente y que influyen a futuro en la toma de decisiones.
- Puede comunicarse directamente con otros agentes.

- Capaz de percibir su ambiente
- Representación parcial, esto es, los agentes no tienen una percepción completa de lo que pasa en su ambiente.

Los agentes son clasificados de acuerdo a sus características o habilidades que poseen. Antes de resumir dicha clasificación, se presenta una lista de características que denotan a un agente.

- Comunicativo: capaz de interactuar con otros agentes a través de la comunicación.
- Semiautónomo: ni están bajo el control humano o de otros agentes ni son completamente autónomos, es decir, el humano debe ser capaz de controlar la autonomía del agente.
- Reactivo: capaz de reaccionar a tiempo ante los cambios en su ambiente
- Deliberativo: capaz de elegir una tarea a efectuar para cumplir con un objetivo.
- Colaborativo: capaz de interactuar con humanos y otros agentes.
- Pro-activo: capaz de tomar la iniciativa, sin percibir cambios en su ambiente.
- Adaptativo: capaz de acomodar cambios y tareas de su ambiente

Los agentes se clasifican como agentes de software, agentes móviles, agentes reactivos y agentes cognitivos.

- Los agentes software son programas de software que pueden ejecutar una tarea en específico y posee un grado de inteligencia que le permite ejecutar parte de sus tareas de forma autónoma e interactuar con su ambiente de una manera conveniente.
- Los agentes móviles son agentes de software que pueden emigrar de una máquina a otra.
- Los agentes reactivos son simples agentes de software (y/o hardware) equipados con un mecanismo que reaccionan a los cambios del ambiente.
- Los agentes cognitivos son agentes autónomos que siguen sus propios objetivos mediante un mecanismo complejo de razonamiento.

El tipo de agente que se considera en este trabajo de investigación son los agentes cognitivos.

Definición 2.16 *Los agentes cognitivos (llamados también agentes deliberativos) poseen un conocimiento explícito del medio ambiente. Adicionalmente al mecanismo tradicional de los agentes reactivos, pueden interactuar y cooperar con otros agentes. Los agentes cognitivos pueden ser usados para:*

- *Resolver problemas distribuidos*
- *Resolver problemas extensos para ser resueltos por un agente simple*
- *Construir sistemas modulares*

2.2.2. Sistemas MultiAgentes

Los sistemas multiagentes se pueden definir como un sistema donde interactúan diversos agentes para alcanzar un objetivo en común o para ejecutar alguna tarea. Los sistemas multiagentes están compuestos de los siguientes elementos:

1. Un ambiente, que es un espacio que generalmente tiene volumen.
2. Un conjunto de objetos. Estos objetos están situados en el ambiente y que pueden ser percibidos, creados, destruidos y modificados por los agentes.
3. Un conjunto de agentes, los cuales son un tipo de objetos en el ambiente, que representan las entidades activas del sistema.
4. Un conjunto de operaciones que hacen posible al agente, percibir, consumir y manipular objetos del ambiente.
5. Operadores con la tarea de representar las aplicaciones de esas operaciones.

Cuando los agentes son situados en el ambiente, el ambiente es generalmente un espacio, y los agentes son capaces de

- percibir su ambiente, esto es, de reconocer los objetos situados en el ambiente a través de las capacidad perceptiva;
- actuar, esto es, de transformar el estado del sistema para modificar posiciones y las relaciones que existen entre los objetos.

Actualmente, en los sistemas multiagentes se considera fundamental el concepto de ambiente para la coordinación de acciones entre los agentes.

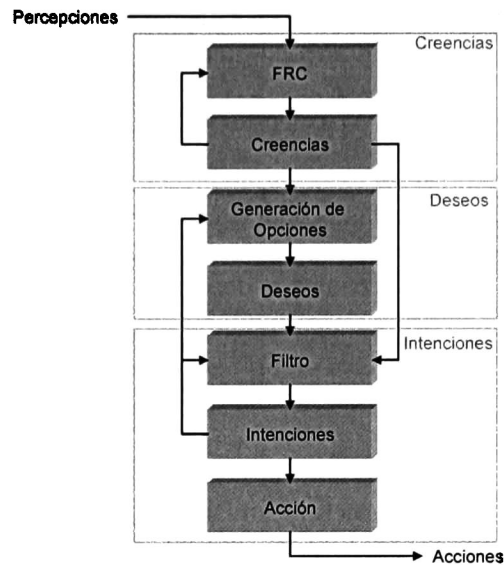


Figura 2.1: Componentes de la arquitectura BDI.

2.2.3. Arquitectura Creencias, Deseos e Intenciones (BDI)

El principio de esta arquitectura es el enfoque tradicional del razonamiento práctico, es decir, del proceso de decidir, momento a momento qué acción ejecutar con la finalidad de cumplir los objetivos. El razonamiento práctico involucra dos procesos importantes: decidir *qué* objetivo quiere alcanzar y *cómo* el agente ha de alcanzar esos objetivos.

El proceso del razonamiento práctico de la arquitectura *BDI* se muestra en la figura 2.1 y lo conforma siete componentes:

- 1. Un conjunto de *creencias* actuales, que representan la información que el agente tiene acerca del ambiente.
- 2. Una *función de revisión de creencias* (FRC), toma una percepción y las creencias actuales del agente y en base a ellos, determina un nuevo conjunto de creencias.
- 3. Una *función de generación de opciones*, determina los objetivos disponibles del agente (sus deseos), en base a sus creencias actuales acerca de su ambiente y sus intenciones actuales.
- 4. Un conjunto de *deseos* actuales, representan los objetivos o las situaciones que el agente quisiera lograr.
- 5. Una *función filtro*, representa el proceso de deliberar, es decir, determina las intenciones del agente en base a sus creencias actuales, deseos e intenciones.
- 6. Un conjunto de *intenciones* actuales, que representan los posibles cursos de acción disponibles al agente para alcanzar uno o más de sus deseos.

7. Una *función de selección de acción* (acción), el cual determina una acción para ejecutar en base a su intenciones actuales.

Para explicar de una manera general los componentes de la arquitectura BDI suponga que *Bel* es el conjunto de todas las posibles creencias, *Des* el conjunto de todos los posibles deseos e *Int* el conjunto de todas las posibles intenciones de un agente. Solo para ejemplificar, el contenido de estos conjuntos no es importante, sin embargo, en la formalización de dicha arquitectura tantos las creencias, deseos e intenciones son representados, en su mayoría, mediante fórmulas de la lógica modal o de primer orden.

La función de revisión de creencias de un agente es un mapeo

$$frc : \varrho(Bel) \times P \rightarrow \varrho(Bel)$$

el cual, en base a las percepciones actuales (*P*) y las creencias actuales (*Bel*), determina un nuevo conjunto de creencias *Bel*. La revisión de creencias puede incluir un modelo de su ambiente si el agente está situado en un ambiente dinámico. Dicho modelo representa el conocimiento inicial que tiene acerca del ambiente donde ha de llevar a cabo acciones. Las condiciones que debe cumplir la función de revisión de creencias se detallada en la sección 2.2.3.

La función de generación de opciones, *opciones*, mapea un conjunto de creencias y un conjunto de intenciones a un conjunto de deseos

$$opciones : \varrho(Bel) \times \varrho(Int) \rightarrow \varrho(Des)$$

Esta función debe ser responsable del proceso de decidir cómo alcanzar los objetivos. Así, una vez que el agente ha formado una intención, éste debe considerar las opciones para alcanzar dicho objetivo. Tiene dos propósitos la función de generación de opciones. El primero es que debe ser consistente: cualquier opción generada debe ser consistente con las creencias actuales y con las intenciones actuales. El segundo es que debe ser oportuno, en el sentido de que debe de reconocer cuándo las circunstancias del ambiente cambian repentinamente para ofrecer al agente nuevas formas de alcanzar los objetivos.

El proceso de deliberación, decidir qué hacer, se representa por la función filtro

$$filtro : \varrho(Bel) \times \varrho(Des) \times \varrho(Int) \rightarrow \varrho(Int)$$

el cuál actualiza las intenciones del agente en base a las creencias, deseos e intenciones actuales. Esta función debe cumplir con dos condiciones. La primera debe abortar cualquier intención que no puede ser alcanzable y la segunda es que debe retener aquellas intenciones que no son alcanzables y que aún pueden ser consideradas en un proceso posterior. |

La función de ejecución se asume que simplemente retorna una intención ejecutable

$$ejecución : \varrho(Int) \rightarrow A$$

La función de revisión de creencias.

La razón de requerir una función de revisión de creencias se debe al hecho de que tenemos la posibilidad de conflicto entre las creencias actuales con las que se genere cada que se percibe nueva información. Simplemente agregando la nueva información percibida al conjunto de creencias podría causar inconsistencias, y por supuesto no es lo deseable. Entonces, se necesita revisar la base de creencias para hacerla consistente.

El enfoque sobre el cual muchos trabajos se han basado para abordar este problema es el propuesto por Alchourron, Ganderfors y Makinson, llamado *AGM*. En ese enfoque la base de creencias es llamado estado epistémico del agente, el cual se representa por un conjunto de fórmulas de la lógica proposicional L :

- L es cerrado bajo $\neg, \wedge, \vee, \longrightarrow$
- A, B, C son variables sobre las sentencias de L .
- \perp, T , falsedad y verdad, respectivamente
- Relación de consecuencia, Cn .

Como la idea central de la revisión de creencias es evitar inconsistencias en las creencias del agente, proponen la definición de un operador llamado *operador de revisión* (*) que efectúa la revisión de una manera racional, lógica y consistente. Algunos de los principios básicos del operador de revisión son:

1. Las creencias deben ser consistentes, antes y después de la revisión.
2. La manera en como se representen las creencias debe ser igual, antes y después de la revisión.
3. Debe ser mínima la pérdida de información durante el cambio de creencias.
4. Las creencias menos importantes deben ser eliminadas en preferencia a otras creencias.

El mecanismo de la revisión está basado en la contracción (-) y expansión (+). Primero elimina todas las creencias inconsistentes, provocando cambios mínimos en la base de creencias y después agrega la nueva información percibida.

Definición 2.17 *El conjunto de creencias se representa por un conjunto de sentencias $K \subseteq L$, el cual es cerrado bajo la consecuencia lógica, esto es, cualquier subconjunto de fórmulas en un conjunto de creencias también está en el conjunto de creencias. Este conjunto puede ser visto como una teoría o como una descripción parcial del ambiente del agente.*

Definición 2.18 (*Teoría o Conjunto de creencias*) Un conjunto de fórmulas cerrado y deductivo X es una teoría si y solo si $Cons(X) = X$.

La teoría de expansión es fácil de definir en un conjunto de creencias.

Definición 2.19 (*Teoría de Expansión*). Sea T una teoría y ϕ una fórmula. Entonces $T + \phi = Cons(T \cup \phi)$. Adecuado cuando la fórmula agregada es consistente con la teoría inicial.

Definición 2.20 (*La identidad de Levi*) La identidad de Levi se define como

$$T * \phi = Cons((T - \neg\phi) \cup \phi)$$

$$T * \phi = Cons((T - \neg\phi) + \phi)$$

Contracción y revisión AGM

Alchourron, Makinson y Ganderfors desarrollaron un conjunto de postulados que cualquier revisión de creencias debe cumplir. Estos postulados son:

Nombre	Postulados de revisión
(K*1)	Para cualquier sentencia ϕ y para cualquier conjunto de creencias K , $K * \phi$ es un conjunto de creencias.
(K*2)	$\phi \in K * \phi$
(K*3)	$K * \phi \subseteq K + \phi$
(K*4)	Si $\neg\phi \notin K$, entonces $K + \phi \subseteq K * \phi$
(K*5)	$K * \phi = K_{\perp}$ si y solo si $\vdash \neg\phi$
(K*6)	Si $\vdash \phi \leftrightarrow \psi$, entonces $K * \phi = K * \psi$
(K*7)	$K * \phi \wedge \psi \subseteq (K * \phi) + \psi$
(K*8)	Si $\neg\psi \notin K * \phi$, entonces $(K * \phi) + \psi \subseteq K * \phi \wedge \psi$

donde (K*1) es la familia de postulados de la lógica proposicional, los cuales, son cerrados bajo la consecuencia lógica.

(K*2) Establece que la nueva información percibida debe estar incluida en K .

(K*3) Expresa que la expansión representa una “cota máxima” cuando se incorpora nuevas creencias.

(K*4) Establece que, cuando la negación de la información percibida no está en K , entonces, la operación de revisión se reduce sólo a efectuar la operación de expansión.

(K*5) La única situación en la que la revisión permite tener inconsistencias en el conjunto de creencias ocurre cuando el agente acepta información contradictoria.

(K*6) Es una expresión del Principio de Irrelevancia de Sintaxis.

(K*7) y (K*8) La idea es que, si $K * \phi$ es una revisión de K y $K * \phi$ cambia por otra sentencia ψ , tal cambio debe ser hecho por la expansión de $K * \phi$ cuando sea posible. $K * \phi \wedge \psi$, debe ser el mismo como la expansión de $K * \phi$ por ψ siempre y cuando ψ no contradiga las creencias en $K * \phi$.

Nombre	Postulados de contracción
(K-1)	Para cualquier sentencia ϕ y para cualquier conjunto de creencias K , $K - \phi$ es un conjunto de creencias.
(K-2)	$K - \phi \subseteq K$
(K-3)	If $\phi \notin K$ then $K - \phi = K$
(K-4)	Si no $\vdash \phi$, entonces $\phi \notin K - \phi$
(K-5)	Si $\phi \in K$, entonces $K \subseteq (K - \phi) + \phi$
(K-6)	Si $\vdash \phi \leftrightarrow \psi$, entonces $K - \phi = K - \psi$
(K-7)	$K - \phi \cap K - \psi \subseteq K - \phi \wedge \psi$
(K-8)	Si $\phi \notin K - \phi \wedge \psi$, entonces $K - \phi \wedge \psi \subseteq K - \psi$

donde (K-1) es la familia de postulados de la lógica proposicional los cuales son cerrados bajo la consecuencia lógica.

(K-2) como $K - \phi$ se forma a partir de K eliminando algunas creencias, se requiere que nuevas creencias no ocurran en $K - \phi$.

(K-3) cuando $\phi \notin K$, el criterio del principio de la economía de la información (conservar lo más posible las viejas creencias), requiere que nada se debe eliminar en K .

(K-4) la sentencia a ser eliminada no debe ser una consecuencia lógica de las creencias del conjunto $K - \phi$. A menos que ϕ sea lógicamente válida, en tal caso, ésta nunca debe ser eliminada por la restricción de integridad (si K deriva lógicamente ϕ , entonces $\phi \in K$.)

(K-5) una forma de garantizar el principio de la economía de la información, es expandir $K - \phi$ con ϕ . Esto debe dar el mismo conjunto de creencias antes de la contracción.

(K-6) es una expresión del Principio de Irrelevancia de Sintaxis.

(K-7) La contracción de K con $\phi \wedge \psi$ es la misma si primero se elimina K con ϕ y después se elimina K con ψ .

(K-8) si la sentencia ψ no es parte de $K - \phi \wedge \psi$ entonces $K - \phi \wedge \psi$, debe ser el mismo como la contracción de $K - \psi$.

Capítulo 3

Esquema de la generación de creencias en RdPI

Resumen. Este capítulo formaliza cada uno de los elementos que comprenden la generación de creencias bajo dos enfoques: centralizado y distribuido.

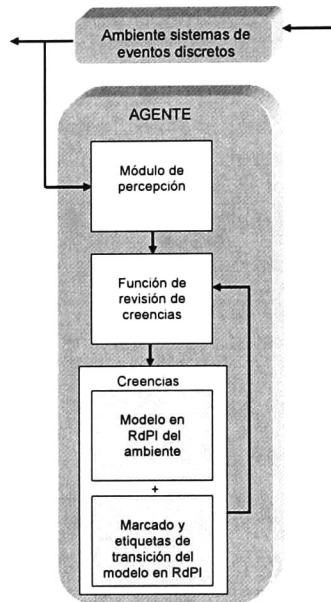


Figura 3.1: Esquema de generación de creencias.

Los agentes que son implementados con la arquitectura *BDI* en ambientes de sistemas de manufactura flexible son usados, por ejemplo, para la toma de decisiones y el monitoreo de señales de sistemas. Las decisiones dependen de la información percibida: incrementando dicha información se generan buenas decisiones. Desafortunadamente, el caso cuando el agente conoce toda la información del sistema no es del todo factible. Esto se debe a que varias señales del sistema no pueden ser medibles por el costo de los sensores o porque no existe la tecnología adecuada para medir ciertas variables del sistema. Para solucionar este problema, los agentes cognitivos están dotados de un mecanismo que supone la señal no percibida, llamado *función de revisión de creencias*.

3.1. Generador de Creencias Centralizado

El esquema del generador de creencias que se propone en este trabajo consiste de un modelo del ambiente del agente, un módulo de percepción, creencias y una función de revisión de creencias. El ambiente del agente es modelado con una Red de Petri Interpretada (*RdPI*) que representa el comportamiento de un Sistema de Eventos Discretos (*SED*). El módulo de percepción es una función dedicada a obtener la transición disparada en el modelo del ambiente del agente. La función de revisión de creencia está enfocada a generar creencias acerca del ambiente del agente, dependiendo de la información percibida del ambiente y las creencias actuales. Finalmente, las creencias representan la información que el agente tiene acerca del estado del sistema y las faltas del sistema con el que interactúa.

La figura 3.1 ilustra de una manera abstracta los módulos que integran el esquema de

generación de creencias y son definidos a continuación.

3.1.1. Ambiente

Los agentes por lo general tienen una representación de su mundo externo y consiste de una representación del ambiente en el cual ellos interactúan y de otros agentes que son parte de dicho ambiente [4]. Los modelos correspondientes al estado del ambiente y al modelo social pueden ser vistos como modelos estructurales, los cuales, eventualmente serán complementados por estructuras que representan las tareas globales ó los objetivos a alcanzar (deseos) y las acciones a realizar para lograr dichos objetivos (intenciones).

En algunos sistemas multiagentes, cada agente tiene su propio modelo del ambiente y lo actualizan analizando las salidas de sus sensores o leyendo los mensajes que reciben de otros agentes. Cuando cada agente va a construir su propio modelo del ambiente, el problema es cómo hacerlo y cómo mantener el modelo del ambiente de forma dinámica. En esta situación, la representación del ambiente de cada agente puede ser diferente y la consistencia global no se garantiza. Por otro lado, algunos sistemas multiagentes, el ambiente es modelado por un agente en específico, el cual simplifica la representación del ambiente que requieren el resto de los agentes.

En ambos casos, el modelo del ambiente debe ser congruente con el ambiente del agente para que el comportamiento del agente en base a sus creencias, deseos e intenciones sea exitoso.

En este trabajo, el ambiente del agente es un Sistema de Eventos Discretos (*SED*) y se modela con *RdPI*. El modelo del ambiente es el conocimiento inicial que tiene el agente.

3.1.2. Metodología de modelado en *RdPI*

La siguiente metodología tipo bottom-up, presentado en [1],[35], permite modelar en *RdPI* el comportamiento del *SED*.

A. Construcción de módulos en *RdP*

Algoritmo 3.1 Construcción de *RdP*

1. *Componentes del sistema.- Identificar y nombrar los componentes del sistema, que no son sensores. Como resultado de esta etapa, se debe obtener el conjunto finito COMPONENTES_SISTEMA = $\{cs_1, cs_2, \dots, cs_n\}$ con los nombres dados a los componentes. Un componente del sistema puede ser una válvula, un motor, un recurso del sistema, etc.*
2. *Variables de estado. Para cada componente del sistema, seleccionar las variables necesarias para representar su comportamiento. En otras palabras, construir el conjunto finito VARIABLES_ESTADO_i = $\{ve_1^i, ve_2^i, \dots, ve_m^i\}$ asociado al componente del sistema $cs_i \in$ COMPONENTES_SISTEMA. Estas variables pueden representar la*

posición (por ejemplo, la posición del vástago de una válvula), velocidad, voltaje, etc. de cada componente del sistema. Debe existir al menos una variable de estado por cada componente del sistema.

3. *Conjuntos de valores.*- Para cada variable de estado $ve_j^i \in \text{VARIABLES_ESTADO}_i$, identificar el conjunto de posibles valores de ve_j^i , $\text{VALORES}_{ve_j^i} = \{val_1^{ij}, val_2^{ij}, \dots, val_p^{ij}\}$
 Por ejemplo, la variable "posición_de_válvula" podría tomar dos valores "abierta", "cerrada"
4. *Codificación de valores de variables de estado.*- Representar en términos de marcado en RdP los valores de cada conjunto $\text{VALORES}_{ve_j^i}$. Este punto se puede lograr fácilmente si se usan lugares binarios. Para cada $ve_j^i \in \text{VARIABLES_ESTADO}_i$, crear el conjunto $P_{ve_j^i} = \{p_1^{ij}, p_2^{ij}, \dots, p_n^{ij}\}$ de lugares, tal que $|\text{VALORES}_{ve_j^i}| = |P_{ve_j^i}|$. El marcado de estos lugares debe ser binario y mutuamente excluyente. Entonces, cuando $M(p_z^{ij}) = 1$ significa que la variable ve_j^i tiene el valor val_z^{ij} .
5. *Modelado de eventos.*- Para cada par de valores val_m^{ij}, val_n^{ij} tal que la variable de estado ve_j^i podría cambiar del valor val_m^{ij} al valor val_n^{ij} , crear una transición t_{mn}^{ij} . Agregar un arco del lugar p_m^{ij} a la transición t_{mn}^{ij} y un arco de la transición t_{mn}^{ij} al lugar p_n^{ij}
6. *Marcado inicial.*- Si val_n^{ij} es el valor inicial de la variable de estado ve_j^i , entonces $M_0(p_m^{ij}) = 1$, y $M_0(p_n^{ij}) = 0$ de otra manera.
7. *Salida.* La salida de este algoritmo es un conjunto de módulos aislados de RdP, \mathfrak{M}_j^i , cada uno modela el comportamiento de la variable de estado ve_j^i .

Hasta ahora han sido creados los módulos aislados G^{ij} en RdP, uno por cada variable de estado ve_j^i . El próximo apartado define dos operaciones en RdP que sirven para fusionar los módulos y obtener un modelo global.

B. Composición de módulos de RdP

La composición de módulos de RdP se hace a través de dos operadores básicos: la composición sincrónica y permisiva. La composición sincrónica se usa para establecer una relación entre módulos que tienen transiciones que representan el mismo evento en el sistema; esta composición fusiona dos o más transiciones en una única transición, siempre y cuando ellas representen el mismo evento en el sistema. Por lo tanto, esta composición representa el hecho de que las variables de estado evolucionan simultáneamente. La composición permisiva entre dos módulos se usa para establecer una relación entre dos o más módulos cuando el marcado de un lugar de un módulo permite el disparo de una transición de otro(s) módulo(s). El disparo de estas transiciones no modifica el marcado de estos lugares. Esta composición se puede representar con autolazos entre los módulos. Las siguientes definiciones ayudarán a formalizar estos conceptos.

Definición 3.1 Sea (G, M_0) una RdP y $\text{ETIQUETAS} = \{e_1, \dots, e_w\}$ un conjunto de etiquetas arbitrariamente definido. (G, M_0) es una RdP etiquetada si existe una función $tEt : T \rightarrow \text{ETIQUETAS}$ que asocia una etiqueta a cada transición y una función $pEt : P \rightarrow 2^{\text{ETIQUETAS}}$ que asocia un subconjunto de etiquetas a cada lugar.

Definición 3.2 Sean (G_1, M_0^1) y (G_2, M_0^2) dos RdP etiquetadas con conjuntos de etiquetas ETIQUETAS_1 y ETIQUETAS_2 definidos en cada RdP respectivamente. Además ETIQUETAS_1 no es necesariamente diferente de ETIQUETAS_2 . La composición sincrónica de (G_1, M_0^1) y (G_2, M_0^2) es la RdP etiquetada $(G_{1\parallel 2}, M_0^{1\parallel 2})$ dada por:

$$P_{1\parallel 2} = P_1 \cup P_2,$$

$$T_{1\parallel 2} = (T_1 - cT_1) \cup (T_2 - cT_2) \cup cT,$$

$$M_0^{1\parallel 2} = M_0^1 \cup M_0^2,$$

$$pEt_{1\parallel 2} = pEt_1 \cup pEt_2,$$

$$tEt_{1\parallel 2}(t_j) = \begin{cases} tEt_1(t_j) & \text{si } t_j \in T_1 \cup cT \\ tEt_2(t_j) & \text{si } t_j \in T_2 \end{cases},$$

$$I_{1\parallel 2} = I_1|_{\{(p_i, t_j) | t_j \in T_1 - cT_1\}} \cup I_2|_{\{(p_i, t_j) | t_j \in T_2 - cT_2\}} \cup \left\{ ((p_i, t_j), 1) \left| \begin{array}{l} t_j \in cT_1 \text{ y } I_1(p_i, t_j) = 1 \\ \text{or} \\ t_j \in cT_2 \text{ y } I_2(p_i, t_j) = 1 \end{array} \right. \right\}$$

$$O_{1\parallel 2} = O_1|_{\{(p_i, t_j) | t_j \in T_1 - cT_1\}} \cup O_2|_{\{(p_i, t_j) | t_j \in T_2 - cT_2\}} \cup \left\{ ((p_i, t_j), 1) \left| \begin{array}{l} t_j \in cT_1 \text{ y } O_1(p_i, t_j) = 1 \\ \text{or} \\ t_j \in cT_2 \text{ y } O_2(p_i, t_j) = 1 \end{array} \right. \right\}$$

donde

$$cT_1 = \{t_k \in T_1 | \exists t_i \in T_2, tEt_1(t_k) = tEt_2(t_i)\},$$

$$cT_2 = \{t_k \in T_2 | \exists t_i \in T_1, tEt_2(t_k) = tEt_1(t_i)\}.$$

$$cT = t_k | \exists t_i \in cT_1, \exists t_j \in cT_2, tEt_1(t_i) = tEt_2(t_j), \text{ y}$$

$f|_A$ significa que la función f está restringida al conjunto A .

De la definición se observa que la composición sincrónica es conmutativa y asociativa.

Definición 3.3 Sean (G_1, M_0^1) y (G_2, M_0^2) dos RdP etiquetadas con conjuntos de etiquetas ETIQUETAS_1 y ETIQUETAS_2 definidos en cada RdP respectivamente. Además ETIQUETAS_1 no es necesariamente diferente de ETIQUETAS_2 . La composición permisiva de (G_1, M_0^1) y (G_2, M_0^2) es la RdP etiquetada $(G_{1\circ 2}, M_0^{1\circ 2})$ dada por

$$P_{1\circ 2} = P_1 \cup P_2,$$

$$T_{1\circ 2} = T_1 \cup T_2,$$

$$M_0^{1\circ 2} = M_0^1$$

$$\begin{aligned}
 I_{1 \circ 2} &= I_1 \cup I_2 \cup \left\{ ((p_k, t_j), 1) \left| \begin{array}{l} p_k \in P_{1 \circ 2}, t_j \in T_{1 \circ 2} \text{ y} \\ tEt_1(t_j) \in pEt_2(p_k) \text{ o} \\ tEt_2(t_j) \in pEt_1(p_k) \end{array} \right. \right\} \\
 O_{1 \circ 2} &= O_1 \cup O_2 \cup \left\{ ((p_k, t_j), 1) \left| \begin{array}{l} p_k \in P_{1 \circ 2}, t_j \in T_{1 \circ 2} \text{ y} \\ tEt_1(t_j) \in pEt_2(p_k) \text{ o} \\ tEt_2(t_j) \in pEt_1(p_k) \end{array} \right. \right\} \\
 tEt_{1 \circ 2} &= tEt_1 \cup tEt_2, \text{ y} \\
 pEt_{1 \circ 2} &= pEt_1 \cup pEt_2.
 \end{aligned}$$

De la definición se observa que la composición permisiva es conmutativa y asociativa.

Continuando con la metodología de modelado, la composición de módulos se realiza como se indica en el siguiente algoritmo.

Algoritmo 3.2 *Composición de módulos en RdP*

Entradas: Los módulos \mathfrak{M}_i^j

Salidas: El modelo en RdP (G, M_0)

1. **Etiquetado de módulos.**- Para cada módulo \mathfrak{M}_i^j de RdP, se definen las funciones tEt_i^j y pEt_i^j de etiquetado. Ambas funciones se definen de manera arbitraria, sin embargo, tEt_i^j debe ser inyectiva. El proceso de etiquetado asocia un significado físico con los nodos de la RdP.

Si una transición tiene un significado físico (es decir, el evento del sistema que la transición representa) que se activa por la ocurrencia de otras variables del sistema (esto es, cuyo disparo se restringe por el valor de otras variables de estado del sistema), entonces esta transición se debe replicar tantas veces como su significado físico se activa y cada réplica de transición debe tener una etiqueta diferente para preservar que la función sea inyectiva. Por ejemplo, si una máquina es descargada por robot 1 o robot 2, entonces la transición que representa la acción de descargar se debe duplicar.

El conjunto de etiquetas de un lugar p_m del módulo \mathfrak{M}_i^j debe contener la etiqueta de una transición t_n del módulo \mathfrak{M}_i^k si la ocurrencia del evento representado por t_n requiere que el valor de la variable ve_j^i sea igual al valor representado por el lugar p_m .

2. **Composición de módulos.**- Obtener la composición sincrónica \mathfrak{M}^{cs} de todos los módulos \mathfrak{M}_i^j y después aplicar la composición permisiva al modelo resultante \mathfrak{M}^{cs} consigo mismo para obtener el modelo (G, M_0) .

Finalmente una RdPI se construye agregando las señales de entrada y salida a la RdP. Esta tarea se realiza aplicando el siguiente algoritmo.

C. Definición de la función de entrada y salida de una RdPI

Algoritmo 3.3 Construcción de una RdPI

Entradas: El modelo en RdP (G, M_0)

Salidas: El modelo en RdPI (G, M_0)

1. **Función de entrada.**- La función λ se establece asociando a cada transición t_{mn}^{ij} del modelo, el símbolo de entrada α_{mn}^{ij} (si existe) y se usa para cambiar la variable de estado ve_j^i del valor val_m^{ij} al valor val_n^{ij} .
2. **Función de salida.**- La función φ se define como una matriz de dimensión $q \times n$, donde q es el número de señales de salida disponibles en el SED y n es el número de lugares en el modelo (G, M_0) . Cada columna de esta matriz es un vector elemental o nulo, y $\varphi(i, j) = 1$ cuando la salida i se presenta (activada) cada vez que $M(p_j) = 1$. Se permite que $\varphi(m, l) = \varphi(m, k) = 1$ solo cuando p_l y p_k pertenecen al mismo conjunto $P_{ve_j^i}$.

La metodología completa para obtener un modelo en RdPI consiste en aplicar los algoritmos en el siguiente orden: Construcción de RdP, Composición de módulos en RdP y Construcción de una RdPI.

Ejemplo 3.1 Considere el sistema de manufactura de la figura 3.2. Este sistema está compuesto de dos máquinas $m1$ y $m2$, un robot $r1$ y un almacén $a1$ donde se depositan los productos procesados.

El comportamiento del sistema es el siguiente. La máquina $m1$ aplicará varias tareas sobre la pieza tipo A , finalmente cuando la pieza es procesada por la máquina, el robot $r1$ la tomará y la colocará en la máquina $m2$. La máquina $m2$, se encargará de procesar la pieza A para obtener un producto final B . Finalmente cuando el producto B es obtenido será recogido por el robot $r1$ y almacenado en $a1$.

De acuerdo con la metodología de modelado, se identifican cuatro componentes: $m1$, $m2$, $r1$ y $a1$. Las variables de estado de interés son las siguientes: "actividades_m1", para describir el comportamiento de la $m1$; "actividades_m2", para describir el comportamiento de la $m2$; "actividades_r1" y "actividades_a1", describen el comportamiento del $r1$ y $a1$ respectivamente.

Los valores de cada una de las variables de estado son las siguientes:

- $Valores_actividades_m1 = \{disponible, procesando, produciendo\}$
- $Valores_actividades_m2 = \{disponible, procesando, produciendo\}$
- $Valores_actividades_r1 = \{libre, trasladando, almacenando\}$
- $Valores_actividades_a1 = \{vacío, ocupado\}$

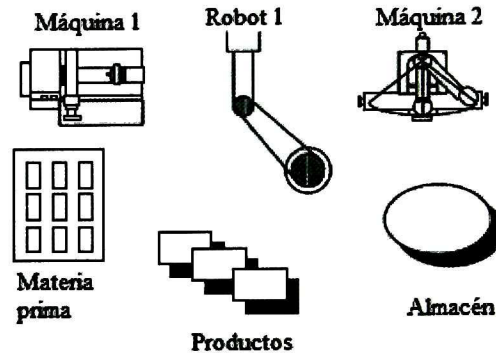


Figura 3.2: Sistema de una celda de manufactura.

Estos conjuntos de valores de las variables de estado son codificados usando los siguientes lugares:

- $Lugares_actividades_m1 = \{p_1, p_2, p_3\}$
- $Lugares_actividades_m2 = \{p_7, p_8, p_9\}$
- $Lugares_actividades_r1 = \{p_4, p_5, p_6\}$
- $Lugares_actividades_a1 = \{p_{10}, p_{11}, p_{10+2i}, p_{11+2i}\}$ donde $i \in \{0 \dots k - 1\}$

La salida del algoritmo Construcción de RdP da como resultado módulos aislados en RdP que representan el comportamiento de cada uno de los componentes identificados. Esos módulos son mostrados en la figura 3.3.

Aplicando los algoritmos Composición de módulos en RdP y Construcción de una RdPI obtenemos el modelo global de RdPI de la figura 3.4.

3.1.3. Percepción

Los agentes tienen la capacidad de percibir su ambiente y ésta se representa mediante una función [45]. La función percibir puede ser implementada en hardware en el caso de un agente situado en un mundo físico, por ejemplo, puede ser una cámara de video, sensor infrarrojo en un robot o cualquier tipo de sensor. Para agentes de software, los sensores, pueden ser comandos que obtienen información del ambiente del software. La salida de la función percibir es una percepción [44].

En este trabajo, el agente recibe señales de salida del sistema y mediante la función de selección de transición determina qué transición del modelo del sistema fue disparada. La función de selección se muestra en la figura 3.5 y se define a continuación.

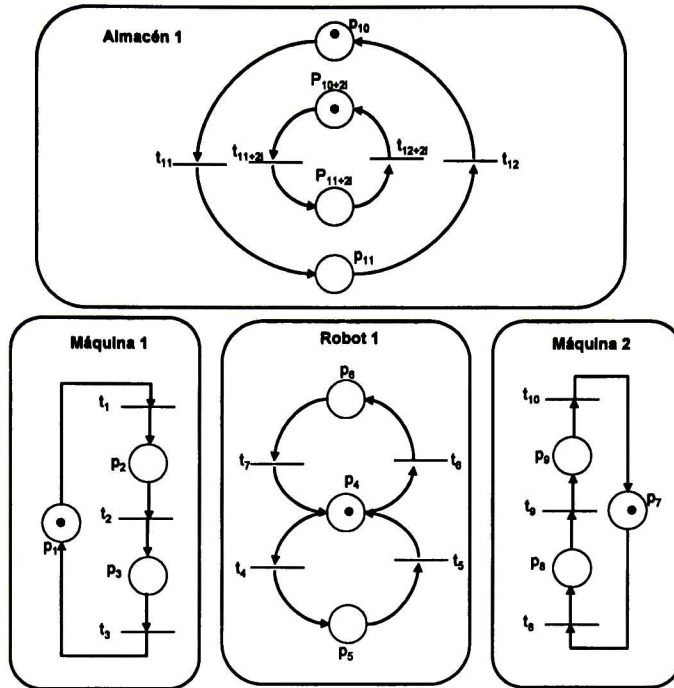


Figura 3.3: Módulos en RdP de cada uno de los componentes del sistema de la celda de manufactura.

Definición 3.4 Sea $N_S = (Q, M_0)$ un modelo $RdPI$ del ambiente del agente. La función de selección de transición es la función: $FST : Z^n \rightarrow \mathbb{N}$. Esta función se calcula como:

$$FST(V_k^{k+1}) = t_j^{\varphi C}$$

donde

$$V_k^{k+1} = \varphi(M_{k+1}) - \varphi(M_k) \text{ y}$$

$t_j^{\varphi C}$ es la columna de φC tal que $\varphi C(\bullet, j) = \varphi(M_{k+1}) - \varphi(M_k)$

La propiedad de evento detectabilidad en el modelo del sistema del ambiente del agente permite distinguir el disparo de la transición en dicho modelo. Esta propiedad se define formalmente como sigue:

Definición 3.5 Una $RdPI (Q, M_0) = (P, T, I, O, \Sigma, \lambda, \varphi)$ es evento detectable si el disparo de alguna transición puede estar determinada únicamente por el conocimiento de las señales de entradas y salida que (Q, M_0) produce.

El siguiente lema, demostrado en [36], establece que la caracterización estructural de la propiedad evento detectabilidad de una $RdPI$ puede llevarse a cabo utilizando algoritmos polinomiales.

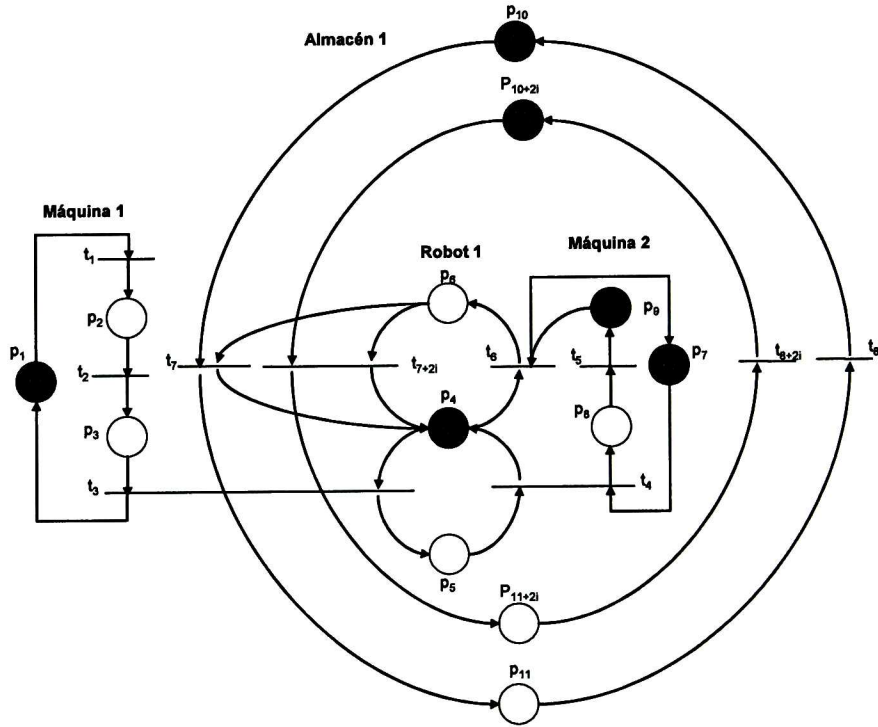


Figura 3.4: Modelo en *RdPI* del sistema de la celda de manufactura.

Lema 3.1 Sea $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$ una *RdPI* viva. (Q, M_0) es evento detectable si y solo si:

1. $\forall t_i, t_j \in T$ tal que $\lambda(t_i) = \lambda(t_j)$ o $\lambda(t_i) = \varepsilon$, cumple que $\varphi C(\bullet, t_i) \neq \varphi C(\bullet, t_j)$ y
2. $\forall t_k \in T$ cumple que $\varphi C(\bullet, t_k) \neq 0$.

Comentario 3.1 Si el modelo del ambiente del agente en *RdPI* (Q, M_0) no tiene la propiedad evento detectabilidad, es decir, el disparo de una transición t_i en (Q, M_0) no se distingue de otra transición t_j ($\varphi C(\bullet, i) = \varphi C(\bullet, j)$), la función de selección de transición daría como resultado un conjunto de transiciones las cuales se confunden bajo la señal de salida del sistema. $(\{t_i, t_j | t_i \neq t_j \text{ y } \varphi C(\bullet, i) = \varphi C(\bullet, j)\})$ o bien daría como resultado una de las transiciones con las que se confunde.

Ejemplo 3.2 Para ejemplificar la función de selección de la transición del agente, retomamos el sistema de la celda de manufactura modelado en *RdPI* de la figura 3.4. En este modelo, el alfabeto de entrada es $\Sigma = \{\}$ porque no hay señales de entrada en el sistema. La función λ para este modelo es $\forall t_j, \lambda(t_j) = \varepsilon$. Como los lugares $p_2, p_3, p_5, p_6, p_8, p_{11}, p_{11+2i}$ representan estados que tienen diferente señales de salida y el resto de los lugares no tienen señales de salida, la matriz de salida

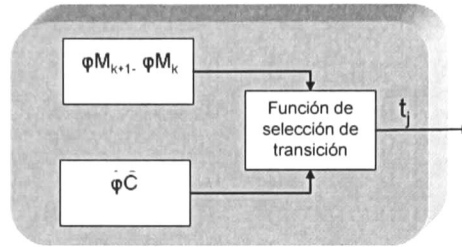


Figura 3.5: Función de selección de transición.

$$\varphi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & 0 \\ & & & & & & & & & & & \dots & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Suponga que el marcado inicial del sistema es $M_0 = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ y que representa que las máquinas $m1$ y $m2$ están disponibles, el robot $r1$ esta libre y el almacén $a1$ en la posición 1 está vacío. Ahora suponga que la máquina $m1$ comienza a procesar la pieza A . Entonces el nuevo marcado del modelo del sistema es $M_1 = [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ El agente percibe la señal de salida

$$\varphi(M_1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

La función de selección de transición calcula

$$\varphi M_1 - \varphi M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\varphi C = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Observe que las columnas de la matriz φC no son nulas y diferentes entre si. El modelo en RdPI del sistema de la celda de manufactura es evento detectable, por lo tanto la función de selección de transición genera como salida la percepción de que la transición t_1 del modelo del sistema fue la que se disparó, esto es:

$$\varphi C(\bullet, 1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

3.1.4. Creencias

Las creencias se pueden definir como la información y representación desde el punto de vista de un agente acerca de su ambiente, otros agentes y sobre sí mismo [44], [45].

Las creencias sociales están relacionadas a la organización y estructura de un grupo de agentes (roles, funciones, etc.). Las creencias personales son un tipo de creencias que tiene el agente sobre sí mismo y por lo tanto incluye la representación de sus roles, actividades, planes, etc. Por último, las creencias ambientales están relacionadas al estado del ambiente, las características de los objetos y agentes, es decir, el conjunto de características externas tales como posiciones, valores, etc. [7]. Por ejemplo, en un sistema de control distribuido, las creencias están relacionadas al estado del proceso y el valor de los parámetros de control. Obviamente, esas creencias por naturaleza, pueden ser erróneas y pueden darse falsas interpretaciones y decisiones incorrectas. Por esta razón, es fundamental incluir un mecanismo de revisión, el cual puede tener en cuenta la evolución de esos datos.

En este trabajo, las creencias del agente son de tipo ambiental porque el ambiente del agente en el que interactúa es un *SED*. El agente está encargado de generar creencias acerca del estado y las faltas del sistema a partir de las creencias actuales y de la información que percibe de su ambiente. Las creencias del estado del sistema son representadas por el marcado del modelo del ambiente del agente y las creencias de las faltas del sistema son representadas por el etiquetado de las transiciones de dicho modelo. En los siguientes capítulos se definen formalmente las creencias de un agente.

3.1.5. Función de revisión de creencias

El esquema de la función de revisión de creencias propuesto en este trabajo se ilustra en la figura 3.6. Éste esquema está compuesto por varias funciones de revisión de creencias FRC_i ; donde ellos comparten información. Cada FRC_i es un sistema dinámico dedicado a generar creencias de cualquiera de los tipos de creencias: ambientales, sociales, relacionales, personales, etc.

El modelo en $RdPI$ del ambiente del agente está disponible para todas las FRC_i . El modelo en $RdPI$ del ambiente del agente puede ser el conocimiento inicial del agente o bien puede ser obtenido por una FRC_i que genere creencias estructurales.

El mecanismo de la FRC_i consiste en generar creencias dependiendo de las señales percibidas del ambiente, el modelo en $RdPI$ del SED y de las creencias actuales. En general, la FRC_i también requiere de las creencias generadas por otros sistemas dinámicos.

Las salidas de las FRC_i son creencias que son usadas por otras estructuras, tales como las intenciones de los agentes, como lo muestra la figura 2.1. Estas estructuras utilizan las creencias para la toma de decisiones que influyan en el comportamiento del SED . Como las estructuras de las intenciones del agente son los planificadores del SED , el agente necesita conocer la calidad de las creencias (lo cercano a los valores actuales del sistema) para que las acciones que ejecute en el SED eviten situaciones peligrosas o riesgos innecesarios para los humanos. Así, las creencias debe cumplir con los siguientes axiomas [28], [45].

A1. Las creencias son acorde a las señales del sistema percibidas.

A2. Nueva información se utiliza para mejorar las creencias, es decir, las creencias están cerca de los hechos cuando se obtiene nueva información.

Este trabajo presenta diversos enfoques algebraicos para representar la FRC que genera creencias del ambiente del agente. Además, estos enfoques algebraicos cumplen con los axiomas porque es posible definir una ecuación de error entre los valores actuales del sistema y la FRC interna. Este error es acotado y nunca incrementa su valor.

El mecanismo para representar la FRC se define en los siguientes capítulos.

3.2. Generador de Creencias Distribuido

El esquema distribuido del generador de creencias que se propone en este trabajo se muestra en la figura 3.7, es una extensión del esquema de la generación de creencias centralizada de la figura 3.1. En este esquema existen tantas FRC como el número de agentes que forman el sistema multiagente. Cada agente está compuesto de los siguientes módulos: un modelo local del SED , función de revisión de creencias, módulo de percepción y un procedimiento de comunicación con los agentes. El modelo local del SED es una división del SED , por lo que cada agente genera creencias acerca del ambiente correspondientes a dicha división. Cada modelo local del SED puede tener eventos comunes, por lo que

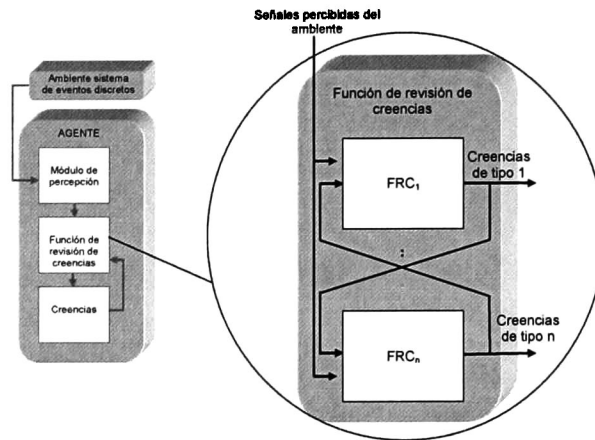


Figura 3.6: Función de revisión de creencias.

los agentes utilizan el procedimiento de comunicación para establecer sincronización e informar las creencias locales.

3.2.1. Metodología de Modelado Distribuido en RdPI

Para construir una generación de creencias descentralizada, se debe dividir el comportamiento del *SED* que representa el ambiente del agente.

En este trabajo, cada modelo local modelado en *RdPI* correspondiente a una partición del *SED* $((G_i, M_0^i)_i)$ representa al comportamiento de una variable del sistema.

Cada modelo $(G_i, M_0^i)_i$ puede tener lugares en común, pero no existen arcos entre cada $(G_i, M_0^i)_i$. Los lugares en común (p_k) existirán por dos casos:

1. Cuando las transiciones de salida de cada p_k representan los eventos que se sincronizaron entre los $(G_i, M_0^i)_i$.
2. Cuando el lugar p_k de un módulo $(G_i, M_0^i)_i$ permite el disparo de una transición de otro(s) módulo(s) $(G_i, M_0^i)_k$ sin perder su marcado.

Estos lugares en común permiten conservar el comportamiento global del *SED* a pesar de que estén representados en varios módulos. Los arcos que entran a estos lugares en común son representados gráficamente como líneas punteadas si vienen de otros módulos e indican la comunicación entre los agentes.

Las siguientes definiciones ayudarán a formalizar los operadores: la composición sincrónica y permisiva distribuida, que permiten modelar los lugares en común.

Definición 3.6 Sean $(G_1, M_0^1)_1$ y $(G_2, M_0^2)_2$ dos módulos locales en *RdP* etiquetados con conjuntos de etiquetas $ETIQUETAS_1$ y $ETIQUETAS_2$ y definidos como lo establece

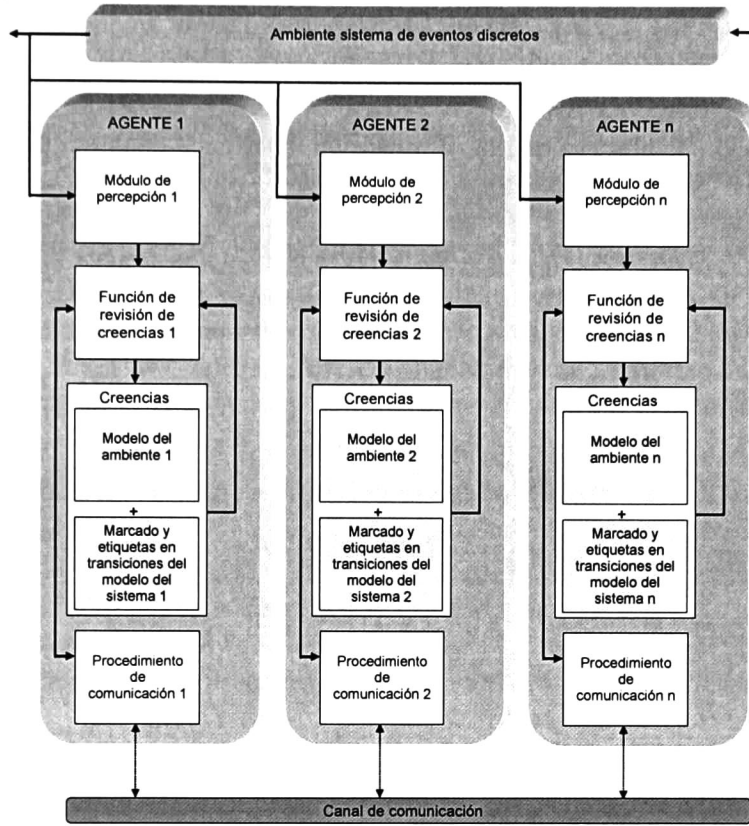


Figura 3.7: Esquema distribuido de la generación de creencias

la definición 3.1. Además $ETIQUETAS_1$ no es necesariamente diferente de $ETIQUETAS_2$. La composición sincrónica distribuida de $(G_1, M_0^1)_1$ y $(G_2, M_0^2)_2$ es la RdP etiquetada $(G_{1||2}, M_0^{1||2})^D$ dada por:

$$P_{1||2}^D = P_1 \cup \bullet cT_2 \cup cT_2 \bullet,$$

$$T_{1||2}^D = T_1 \cup cT_2,$$

$$M_0^{1||2D} = M_0^1,$$

$$pEt_{1||2}^D = pEt_1$$

$$tEt_{1||2}^D(t_j) = \begin{cases} tEt_1(t_j) & \text{si } t_j \in T_1 \\ tEt_2(t_j) & \text{si } t_j \in cT_2 \end{cases}$$

$$I_{1||2}^D = I_1 | \{(p_i, t_j) | t_j \in T_1\} \cup I_2 | \{(p_i, t_j) | t_j \in cT_2 \text{ y } I_2(p_i, t_j) = 1\},$$

$$O_{1||2}^D = O_1 | \{(p_i, t_j) | t_j \in T_1\} \cup O_2 | \{(p_i, t_j) | t_j \in cT_2 \text{ y } O_2(p_i, t_j) = 1\},$$

donde

$cT_2 = \{t_k \in T_2 | \exists t_i \in T_1, tEt_2(t_k) = tEt_1(t_i)\}$ y
 $f|_A$ significa que la función f está restringida al conjunto A .

De la definición se observa que la composición sincrónica distribuida es conmutativa y asociativa.

Definición 3.7 Sean $(G_1, M_0^1)_1$ y $(G_2, M_0^2)_2$ dos módulos locales en RdP etiquetados con conjuntos de etiquetas $ETIQUETAS_1$ y $ETIQUETAS_2$ y definidos como lo establece la definición 3.1 y Además $ETIQUETAS_1$ no es necesariamente diferente de $ETIQUETAS_2$. La composición permisiva distribuida de $(G_1, M_0^1)_1$ y $(G_2, M_0^2)_2$ es la RdP etiquetada $(G_{1\circ 2}, M_0^{1\circ 2})^D$ dada por

$$\begin{aligned} P_{1\circ 2}^D &= P_1 \cup P_2^p, \\ T_{1\circ 2}^D &= T_1, \\ M_0^{1\circ 2} &= M_0^1 \\ I_{1\circ 2} &= I_1 \cup I_2 \cup \left\{ ((p_k, t_j), 1) \mid \begin{array}{l} p_k \in P_{1\circ 2}^D, t_j \in T_{1\circ 2}^D \text{ y} \\ tEt_1(t_j) \in pEt_2(p_k) \end{array} \right\}, \\ O_{1\circ 2} &= O_1 \cup O_2 \cup \left\{ ((p_k, t_j), 1) \mid \begin{array}{l} p_k \in P_{1\circ 2}, t_j \in T_{1\circ 2} \text{ y} \\ tEt_1(t_j) \in pEt_2(p_k) \end{array} \right\}, \\ tEt_{1\circ 2} &= tEt_1, \text{ y} \\ pEt_{1\circ 2} &= pEt_1 \cup pEt_2. \end{aligned}$$

donde

$$P_2^p = \{p_k | p_k \in P_2 \text{ y } tEt_1(t_j) \in pEt_2(p_k) \text{ donde } t_j = p_k \bullet\}$$

De la definición se observa que la composición permisiva distribuida es conmutativa y asociativa.

Para construir un modelo local en $RdPI$ correspondiente a cada variable del sistema que comprende el ambiente del agente, se realiza como se indica el siguiente algoritmo.

Algoritmo 3.4 Distribución del SED.

1. Obtener los módulos aislados en RdP \mathfrak{M}_j^i que representan cada variable de estado del sistema, usando el algoritmo 3.1, Construcción de RdP
2. Para cada módulo \mathfrak{M}_j^i en RdP realizar:
 - **Etiquetado de módulos.**- Definir las funciones tEt_j^i y pEt_j^i de etiquetado (definición 3.1). Ambas funciones se definen de manera arbitraria, sin embargo, tEt_j^i debe ser inyectiva. El proceso de etiquetado asocia un significado físico con los nodos de la RdP .

Si una transición tiene un significado físico que se activa por la ocurrencia de otras variables del sistema, entonces esta transición se debe replicar tantas veces como su significado físico se activa y cada réplica de transición debe tener una etiqueta diferente para preservar que la función sea inyectiva.

El conjunto de etiquetas de un lugar p_m del módulo \mathfrak{M}_j^i debe contener la etiqueta de una transición t_n del módulo \mathfrak{M}_i^k si la ocurrencia del evento representado por t_n requiere que el valor de la variable ve_j^i sea igual al valor representado por el lugar p_m .

- **Composición de módulos.**- *Obtener la composición sincrónica distribuida \mathfrak{M}^{csd} de todos los módulos \mathfrak{M}_i^j y después aplicar la composición permisiva distribuida al modelo resultante \mathfrak{M}^{csd} consigo mismo para obtener el modelo (G, M_0) *i*.*
- *Obtener el modelo local en RdPI, ejecutando el algoritmo 3.3 Construcción de una RdPI*

La distribución del *SED* puede ser tan fina que cada modelo local representa el comportamiento de una variable del *SED* como que el modelo local represente el modelo global del *SED*. La comunicación entre agentes queda fuera del alcance de este trabajo.

Ejemplo 3.3 *Para ejemplificar el algoritmo de la Distribución del SED, considere el sistema de la celda de manufactura de la figura 3.2.*

1. *Los módulos aislados en RdP que representan cada variable de estado del sistema, se muestra en la figura 3.3.*
2. *El etiquetado y la composición de los módulos en RdP son los siguientes:*
 - *La transición t_7 se activa por la ocurrencia de la variable de estado que representa el almacén. Esta transición se replica por cada posición del almacén.*
 - *Aplicando la composición sincrónica y permisiva distribuida obtenemos la figura 3.8. Existen cuatro agentes encargados de generar creencias del ambiente, uno por cada variable de estado del sistema. El modelo local en RdP del sistema de la celda de manufactura de cada agente, tiene lugares adicionales que permiten preservar la dinámica del sistema. Para el caso del modelo en RdP que tiene el agente 2, tiene cinco lugares adicionales ($p_3, p_7, p_9, p_{10}, p_{10+2i}$) al comportamiento de la variable de estado del $r1$. Por medio de los procedimientos de comunicación de los agentes (líneas punteadas), estos lugares adicionales permiten la sincronización de eventos del $r1$ con la $m1, m2$ y el $a1$. Por ejemplo, el disparo de la transición t_7 representa el hecho que si hay una posición disponible en el $a1$ (representado por el lugar p_{10+2i}) y el $r1$ tiene la pieza tipo B (representado por el lugar p_6), entonces el $r1$ almacena la pieza en la posición i (lugar p_{11+2i}) y queda libre para otra operación (lugar p_4).*

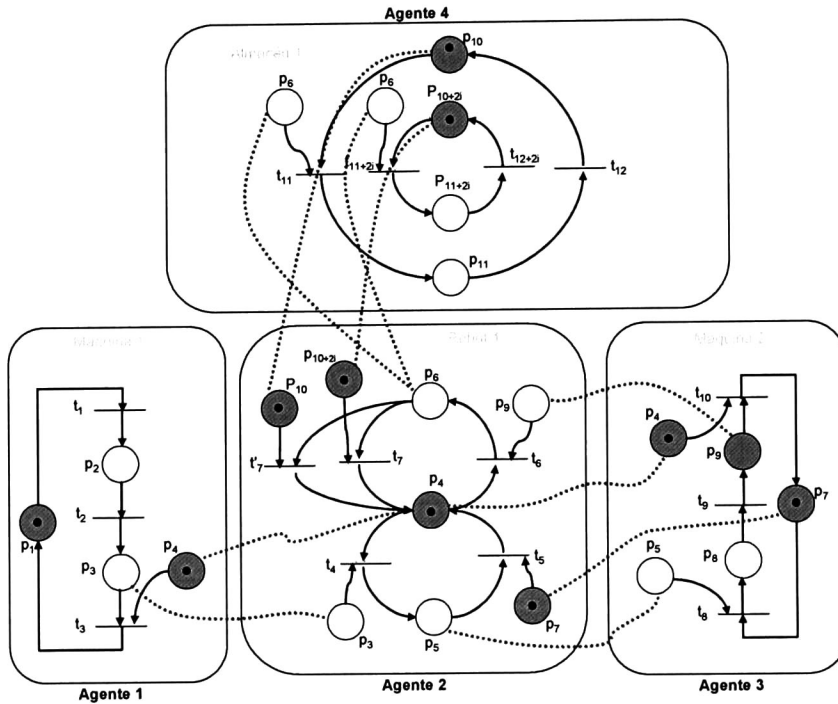


Figura 3.8: Distribución del sistema de la celda de manufactura.

- Para obtener el modelo en RdPI de cada modelo local del agente, se asocia la función de entrada λ y salida φ . El alfabeto de entrada es $\Sigma = \{\}$ porque no hay señales de entrada en el sistema. La función λ para cada modelo en RdPI es $\forall t_j, \lambda(t_j) = \varepsilon$. Como los lugares $p_1, p_4, p_7, p_9, p_{10}, p_{10+2i}$ representan estados que no tienen asociadas señales de salida, gráficamente estos lugares no medibles se representan con círculos oscuros.

3.3. Conclusiones

En este capítulo se definió de forma algebraica los elementos necesarios para modelar la generación de creencias de un agente en un ambiente de SED. La primera parte presentó una metodología para el modelo del SED centralizado. La segunda parte se propone una metodología de modelado para descentralizar la generación de creencias. En un sistema multiagente, cada agente enfocado a generar creencias ambientales, tiene un modelo local en RdPI del SED. Este modelo en RdPI es una copia del modelo del comportamiento de un componente del SED más lugares adicionales. Esta estructura adicional permite conservar la dinámica del SED. Mediante protocolos de comunicación, los agentes se informan entre sí para coordinar sincronización de eventos entre los modelos locales en RdPI y para informar las correspondientes creencias del ambiente.

Capítulo 4

Generador de creencias implementado en RdPIVT

Resumen. Este capítulo presenta el primer enfoque propuesto en este trabajo y consiste en definir la función de revisión de creencias como un mecanismo para generar creencias del estado del sistema y creencias acerca de las posibles faltas del sistema.

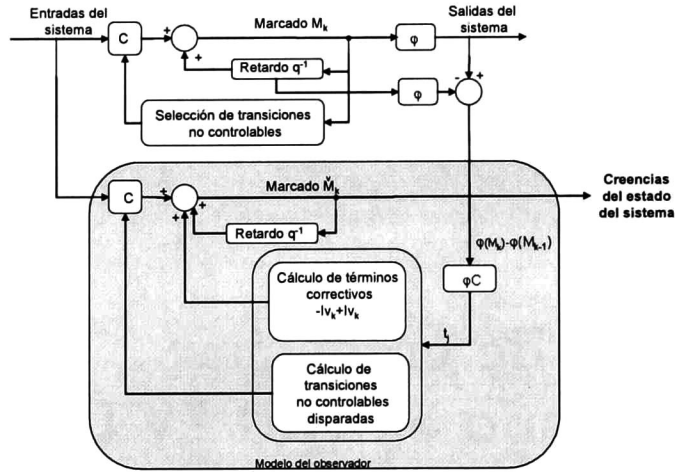


Figura 4.1: Esquema del modelo del observador

En este capítulo se presenta el generador de creencias llamado *Observador–Diagnosticador* el cual consiste de una función de revisión de creencias compuesto por dos sistemas dinámicos para generar creencias del estado del ambiente. El primer sistema dinámico llamado observador es una copia del modelo del sistema más estructura adicional. Con una adecuada regla de disparo, el observador genera creencias acerca del estado del Sistema de Eventos Discretos *SED*. El segundo sistema dinámico llamado diagnosticador es una copia del modelo del observador más unas funciones de etiquetado en los lugares como en las transiciones para generar creencias de las faltas intermitentes del *SED*.

4.1. Creencias del estado del sistema

En general, el agente percibe un subconjunto de la información del sistema, la función de revisión de creencias encargada de generar creencias relacionadas al estado del sistema, tiene que adivinar la información no percibida. Para asegurar que la función de revisión de creencias genere creencias que cumpla con los axiomas $A_1 - A_2$ mencionados en el capítulo anterior, proponemos la arquitectura del *observador* ilustrada en la figura 4.1.

El observador se define a continuación:

Definición 4.1 Sea $N = (Q, M_0)$ un sistema modelado en *RdPI viva, binaria, acotada y evento detectable del sistema* Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$. El observador $S = (Q_S, \check{M}_0)$ de (Q, M_0) es la *RdPI* donde:

1. $Q_S = (P_S, T_S, I_S, O_S, \Sigma, Id, Id)$, donde $P_S = P$, $T_S = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$, la

ecuación de estados es:

$$\begin{aligned}\check{M}_{k+1} &= \check{M}_k + [C \quad -I \quad I] v_k \\ \check{y}_k &= Id(\check{M}_k)\end{aligned}$$

y Id es la función identidad e I es la matriz de identidad. I_S y O_S son las funciones de entrada y salida de la RdPI del modelo del sistema

2. El mercado inicial \check{M}_0 de (Q_S, \check{M}_0) es

$$\blacksquare \check{M}_0(p_i) = \begin{cases} M_0(p_i) & \text{si } p_i \text{ es medible.} \\ k, M^{LB}(p_i) \leq k \leq M^{UB}(p_i), & \text{otro caso.} \end{cases}$$

3. La regla de disparo es el vector de disparo que resulta del siguiente algoritmo. Sea Γ la matriz $\Gamma [i, j] = \gamma_j^i$ que representa el conjunto de las LCM y K es el vector $K(i) = k_i$

Algoritmo 4.1 Cálculo del vector de disparo y mercado actual.

Entradas: C - Matriz de incidencia del modelo del sistema (Q, M_0) ,

t_j Transición que se dispara en el modelo del sistema (Q, M_0)

\check{M}_{act} - mercado actual del modelo del observador

Salidas:

\check{M}_k mercado actual del modelo del observador

v_k el vector de disparo

Si $(\check{M}_{act} + C(t_j) \geq 0)$ entonces

$$\Omega = \begin{cases} v_k(t_j) = 1 \\ v_k(t_i) \geq 0 \\ \forall t_i \in T, \text{ tal que } \varphi C(\bullet, i) \neq \bar{0} \\ v_k(t_i) \leq 1 \\ \forall t_i \in T, \text{ tal que } \varphi C(\bullet, i) \neq \bar{0} \\ \Delta = 0 \end{cases}$$

de lo contrario

$$\Delta = \begin{cases} v_k(i) = 0 \\ \varphi C(\bullet, i) \neq \bar{0} \\ \Omega = 0 \end{cases}$$

$$PPL_1 = \left\{ \begin{array}{l}
 \text{mín } \sum v_k \\
 \text{s.a.} \\
 \check{M}_k = \check{M}_{act} + C \sum v_k \\
 \check{M}_k(p_l) \geq M^{LB}(p_l) \\
 \varphi^{(\bullet, l)} = \bar{0} \\
 \check{M}_k(p_l) \leq M^{UB}(p_l) \\
 \varphi^{(\bullet, l)} = \bar{0} \\
 \sum \check{M}_k(l) = \varphi M_k(m) \\
 \varphi^{(m, l)} \neq \bar{0} \\
 v_k(i) \geq 0 \\
 v_k(i) \leq 1 \\
 v_k(i) \geq 0 \\
 i > |T| \\
 v_k(i) \leq 1 \\
 i > |T| \\
 v_k(i) \geq 0 \\
 \varphi^{C(\bullet, i)} = 0 \\
 v_k(i) \leq 1 \\
 \varphi^{C(\bullet, i)} = 0 \\
 \Omega \\
 \Delta
 \end{array} \right.$$

Esta definición establece que el vector de disparo se calcula bajo los siguientes casos:

1. cuando la transición t_j se dispara en el modelo del sistema y ésta está habilitada en el modelo del observador.
2. cuando el marcado de un lugar excede su cota máxima, esas marcas son removidas por el disparo de las transiciones extras de salida en S .
3. cuando el número de marcas en S no son suficientes para disparar la misma transición t_j , (aquella que se dispara en el modelo del sistema), entonces la transición adicional de entrada en S será disparada para agregar marcas en el lugar de salida de t_j .

Comentario 4.1 *Cuando el modelo del observador trabaja junto con el modelo del diagnosticador, las transiciones extras de entrada y salida del diagnosticador son disparadas cuando la respectiva transición se dispara en el observador.*

Ahora, demostraremos que la arquitectura del observador cumple con los axiomas $A_1 - A_2$, esto es, el observador puede ser utilizado como una función de revisión de creencias para generar creencias relacionadas al estado del sistema. Para probar esto, suponemos que el modelo del sistema en $RdPI$ es evento detectable, esto es, la información percibida del ambiente es suficiente para determinar cuál transición fue disparada en el modelo del sistema en $RdPI$. Esta propiedad se definió formalmente en la sección 3.1.3.

Teorema 4.1 Sea $N = (Q, M_0)$ un sistema modelado en RdPI viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$ y $S = (Q_S, \check{M}_0)$ el observador de (Q, M_0) . Sea $\beta_k [j] = v_k [k]$ el vector de transiciones extras de salida tal que $m < k \leq m + n$ y $\delta_k [k] = v_k [k]$ el vector de transiciones extras de entrada tal que $m < k \leq m + 2n$, entonces las siguientes sentencias se cumplen:

1. $\forall k, M_k \in R(Q, M_0), \check{M}_k \in R(Q_S, \check{M}_0), \varphi(M_k) = \varphi(\check{M}_k)$
2. $\forall k, \|\check{M}_{k+1} - M_{k+1}\|_1 \leq \|\check{M}_k - M_k\|_1$

donde $\|\check{M}_k - M_k\|_1$ es la norma 1 del vector, esto es, $\sum_{i=1}^n |\check{M}_k(i) - M_k(i)|$

Demostración. Por inducción en la longitud de disparo de la palabra σ .

1. Si $|\sigma| = 0$ entonces $k = 0$, \check{M}_0 se establece como en la definición 4.1. Si un lugar p_i es medible, entonces $M_0(p_i) = \check{M}_0(p_i)$, si p_i es no medible, entonces, por lo general $M_0(p_i) \neq \check{M}_0(p_i)$. Además, $\varphi(\bullet, i) = \vec{0}$ si p_i es no medible. Por lo tanto $\varphi(M_0) = \varphi(\check{M}_0)$ (lugares con diferentes marcados son mapeados a cero). Así, la salida del observador es la misma que la salida del sistema.
2. Si $|\sigma| = 1$ entonces $k = 1$. Suponga que $M_0 \xrightarrow{t_j} M_1$ ocurre en N , entonces $\check{M}_0 \xrightarrow{\sigma_1} \check{M}_1$ ocurre en S y que $E_k = \|\check{M}_k - M_k\|_1$ es el error de marcado en el paso 1. De acuerdo a la definición de S , $\check{M}_0 = M_0 + \Delta\check{M}_0$ y $\check{M}_1 = M_1 + \Delta\check{M}_1$. Entonces los siguientes casos son posibles:

a) La transición \vec{t}_j también se dispara en S .

- $v_1(i) = v_1(l) = \vec{0}$, entonces, de la ecuación de estados:

$$M_1 = M_0 + C\vec{t}_j$$

$$\check{M}_1 = (M_0 + \Delta\check{M}_0) + C\vec{t}_j$$

así

$$E_0 = \|\check{M}_0 - M_0\|_1 = \|M_0 + \Delta\check{M}_0 - M_0\|_1 = \|\Delta\check{M}_0\|_1$$

y

$$E_1 = \|\check{M}_1 - M_1\|_1 = \left\| \left[(M_0 + \Delta\check{M}_0) + C\vec{t}_j \right] - \left[M_0 + C\vec{t}_j \right] \right\|_1 = \|\Delta\check{M}_0\|_1$$

por lo tanto $E_1 = E_0$, por lo que $E_1 \leq E_0$.

- $v_1(i) = 1$ para alguna i . Entonces esa transición elimina una marca $m < i \leq m+n$ de p_i cuando $\check{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j > M^{UB}(p_i)$. Entonces, de la ecuación de estados:

$$M_1(p_i) = M_0(p_i) + C \vec{t}_j$$

$$\check{M}_1(p_i) = (M_0(p_i) + \Delta \check{M}_0(p_i)) + C(p_i, \cdot) \vec{t}_j - v_1(i)$$

así

$$E_0(p_i) = \|\Delta \check{M}_0(p_i)\|_1$$

y

$$E_1(p_i) = \|\Delta \check{M}_0(p_i) - v_1(i)\|_1$$

como $(M_0(p_i) + \Delta \check{M}_0(p_i)) + C(p_i, \cdot) \vec{t}_j > M^{UB}(p_i)$ entonces $\Delta \check{M}_0(p_i) > 0$, por tanto $E_1(p_i) \leq E_0(p_i)$.

- $v_1(l) = 1$ para alguna l . Entonces esa transición agrega una marca $m < l \leq m+2n$ en p_i cuando $\check{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j < M^{LB}(p_i)$. Entonces, de la ecuación de estados:

$$M_1 = M_0 + C \vec{t}_j$$

$$\check{M}_1 = (M_0 + \Delta \check{M}_0) + C \vec{t}_j + I v_1$$

así

$$E_0 = \|\Delta \check{M}_0\|_1$$

y

$$E_1 = \|\Delta \check{M}_0 + I v_1\|_1$$

como $(M_0(p_i) + \Delta \check{M}_0(p_i)) + C(p_i, \cdot) \vec{t}_j < M^{LB}(p_i)$ entonces $\Delta \check{M}_0(p_i) < 0$ y por tanto $E_1(p_i) \leq E_0(p_i)$.

- Note que no es posible que ambos $v_1(i) = 1$ y $v_k(l) = 1$.

b) La transición \vec{t}_j no se dispara en S .

- $v_1(i) = v_1(l) = \vec{0}$. Los lugares $p_z \in \bullet t_j$ no tienen suficientes marcas para disparar t_j y el marcado $\check{M}_1(p_z)$ no puede reducirse porque todos los marcados en esos lugares tienen la cota mínima. Entonces, de la ecuación de estados:

$$M_1(p_z) = M_0(p_z) - 1$$

$$\check{M}_1(p_z) = (\check{M}_0(p_z) + \Delta \check{M}_0(p_z)) = M^{LB}(p_z)$$

Así $\Delta \check{M}_0(p_z) < 0$ y

$$E_1(p_z) = \|\Delta \check{M}_0(p_z) + 1\|_1$$

Por tanto $E_1(p_z) \leq E_0(p_z)$.

Los lugares $p_z \in t_j^*$ tiene marcas igual a la cota máxima. Entonces, de la ecuación de estados:

$$M_1(p_z) = M_0(p_z) + 1$$

$$\check{M}_1(p_z) = (\check{M}_0(p_z) + \Delta\check{M}_0(p_z)) = M^{UB}(p_z)$$

Así $\Delta\check{M}_0(p_z) > 0$ y

$$E_1(p_z) = \|\Delta\check{M}_0(p_z) - 1\|_1$$

Por tanto $E_1(p_z) \leq E_0(p_z)$.

- $v_1(i)$, $v_1(l) \neq \vec{0}$. Los lugares Places $p_z \in t_j^*$ no tiene suficientes marcas para disparar t_j y el marcado en S puede ser reducido. Entonces, de la ecuación de estados:

$$M_1(p_z) = M_0(p_z) - 1$$

Como $v_1(z) = 1$, entonces
 $m < z \leq m+n$

$$\check{M}_1(p_z) = (\check{M}_0(p_z) + \Delta\check{M}_0(p_z)) - 1$$

Así

$$E_1(p_z) = \|\Delta\check{M}_0(p_z)\|_1$$

por lo tanto $E_1(p_z) \leq E_0(p_z)$.

Los lugares $p_z \in t_j^*$ tiene marcas y el marcado en S puede ser incrementado. Entonces, de la ecuación de estados:

$$M_1(p_z) = M_0(p_z) + 1$$

Como $v_1(z) = 1$, entonces
 $m < z \leq m+2n$

$$\check{M}_1(p_z) = (\check{M}_0(p_z) + \Delta\check{M}_0(p_z)) + 1$$

Así

$$E_1(p_z) = \|\Delta\check{M}_0(p_z) - 1\|_1$$

Por lo tanto $E_1(p_z) \leq E_0(p_z)$.

Note que el marcado de error nunca incrementa. Así $E_1(p_z) \leq E_0(p_z)$.

3. Ahora, suponga que para $|\sigma| = r$, $E_r \leq E_{r-1}$ y $\varphi(M_r) = \varphi(\check{M}_r)$.
4. Similar al paso 2) $\forall p \in P$, $E_{r+1}(p_z) \leq E_r(p_z)$. Además, como $E_{r+1}(p_z) \leq E_r(p_z)$ y $E_0(p_i) = 0$ para lugar medibles, entonces $E_{r+1}(p_i) = 0$. Por lo tanto, $M_{r+1}(p_i) = \check{M}_{r+1}(p_i)$ para lugar medibles, así $\varphi(M_{r+1}) = \varphi(\check{M}_{r+1})$.

Corolario 4.1 *Sea N un sistema modelado con una RdPI viva, binaria, acotada y evento detectable del sistema Υ . Sea S el observador de N . Entonces $\check{M}_{k+1}(p_i) - M_{k+1}(p_i) = 0$ cuando p_i es un lugar medible en N .*

Corolario 4.2 *Sea N un sistema modelado con una RdPI viva, binaria, acotada y evento detectable del sistema Υ y A un agente interactuando con el sistema Υ . Sea S el observador. Entonces S es una FRC para A .*

Demostración. Del teorema 4.1, el mercado de S cumple que $\check{M}_{k+1} - M_{k+1} \leq \check{M}_k - M_k$, esto es, las creencias son mejoradas cada que el sistema evoluciona. $\varphi(M_k) = \varphi(\check{M}_k)$, esto es, las creencias cumplen con los valores del estado percibidos del sistema.

Así, A puede usar S como una FRC. ■

Comentario 4.2 *La propiedad evento detectable del modelo del ambiente en RdPI permite que, conforme el SED evolucione, las creencias del agente tiendan a ser un hecho, es decir, el mercado del modelo del observador llegue a ser igual al mercado del sistema ($\check{M}_{k+1} = M_{k+1}$).*

4.2. Creencias de faltas del sistema

En esta sección se presenta un sistema dinámico llamado *diagnosticador* para generar creencias acerca de las faltas intermitentes del SED [38]. El *diagnosticador* representa el mecanismo que detecta defectos que tienen el potencial de generar errores en un sistema de duración limitada, causadas por un mal funcionamiento temporal del sistema o debido a alguna interferencia externa. Una arquitectura típica para el modelo del diagnosticador es ilustrado en la figura 4.2. El diagnosticador lo comprende de tres partes: un modelo del sistema, un detector de error y un diagnosticador de falta.

El modelo captura el comportamiento específico del sistema. Cuando el comportamiento del modelo es diferente al comportamiento del sistema, el detector de error activa una señal que el diagnosticador de falta utiliza para identificar la falta del sistema. En este trabajo asumimos que la simulación del modelo del sistema nunca falla. Además capturamos el tiempo del comportamiento de los eventos del sistema, añadiendo dos números reales positivos a las transiciones. Esos números representan el mínimo $H^-(t_i)$ y el máximo $H^+(t_i)$ tiempo de disparo de una transición t_i . Suponga que una transición t_i se habilita en el tiempo τ_i , ésta debe ser disparada en el tiempo π_i , donde $H^-(t_i) + \tau_i \leq \pi_i \leq H^+(t_i) + \tau_i$.

Este trabajo propone un diagnosticador que detecta cuando una transición t_i del sistema se habilita, si t_i no se dispara en el tiempo π_i o se dispara antes de π_i , el diagnosticador determina que esta transición representa la falta en el sistema. Formalmente, los bloques del diagnosticador son definidos a continuación.

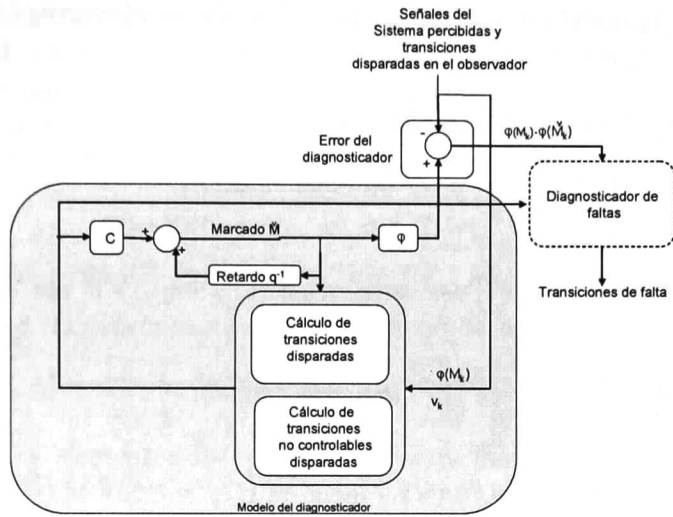


Figura 4.2: Esquema del modelo del diagnosticador

Definición 4.2 Sea $N = (Q, M_0)$ un sistema modelado con RdPI viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$ y $S = (Q_S, \check{M}_0)$ observador de N . El modelo diagnosticador $D = (Q_D, \bar{M}_0)$ de (Q, M_0) , es la RdPIVT donde:

1. $Q_D = (P_D, T_D, I_D, O_D, \Sigma, Id, Id, H^-, H^+, T_k^L)$,

- donde $P_D = P$, $T_D = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$, su ecuación de estados es:

$$\begin{aligned} \bar{M}_{k+1} &= \bar{M}_k + \begin{bmatrix} C & -I & I \end{bmatrix} v_k \\ \bar{y}_k &= Id(\bar{M}_k) \end{aligned}$$

Id es la función identidad y I es la matriz identidad. I_D y O_D son las funciones de entrada y salida de la RdPI del modelo del sistema.

- $H^- : T \rightarrow \mathbb{R}^+$ es una función que asocia a cada transición un número real positivo que representa el tiempo menor de disparo después de que se habilita. Por definición $H^-(t_j^D) = -\infty$, $\forall t_j^D \in T_D - T$.
- $H^+ : T \rightarrow \mathbb{R}^+$ es una función que asocia a cada transición un número real positivo que representa el tiempo mayor de disparo después de que se habilita. Por definición $H^+(t_j^D) = +\infty$, $\forall t_j^D \in T_D - T$
- $T_k^L : T_D \rightarrow \{\text{normal}, \text{falta}\}$ es una función que mapea cada transición $t_j \in T$ a una etiqueta tipo normal o falta como sigue:

$$T_k^L(t_j) = \begin{cases} \text{normal} & \text{Si } t_j \text{ se dispara en el tiempo } \tau_{t_j} \\ & \text{y } H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j} \\ \text{falta} & \text{otro caso} \end{cases}$$

donde χ_{t_j} es el tiempo cuando t_j se habilita en el mercado M_k y τ_{t_j} el tiempo actual del sistema.

2. El mercado inicial \bar{M}_0 de (Q_S, \bar{M}_0) es

$$\blacksquare \bar{M}_0(p_i) = \begin{cases} M_0(p_i) & \text{si } p_i \text{ es un lugar medible.} \\ k, M^{LB}(p_i) \leq k \leq M^{UB}(p_i), & \text{otro caso} \end{cases}$$

3. Sea τ el tiempo cuando t_j^D se habilita en (Q_D, \bar{M}_0) , y ϑ sea el tiempo actual del sistema.

- Entonces $t_j^D \in T$ debe ser disparada cuando:
 - $\vartheta > H^+(t_j^D) + \tau$ o
 - t_j fue disparada en (Q_S, \bar{M}_0) y $H^-(t_j^D) + \tau \leq \vartheta \leq H^+(t_j^D) + \tau$
- Entonces $t_j^D \in T_D - T$ debe ser disparada cuando:
 - la respectiva transición se dispara en el observador.

Comentario 4.3 Los siguientes puntos son a considerar:

1. El mercado inicial del modelo del diagnosticador es el mismo mercado inicial del modelo del observador, esto es, $\bar{M}_0(p_i) = \check{M}_0(p_i)$ si p_i es un lugar no medible, $\bar{M}_0(p_i) = M_0(p_i)$ en otro caso.
2. En este trabajo, las transiciones extras de entrada y salida del diagnosticador, $t_j^D \in T_D - T$, se disparan si se disparan en el modelo del observador, esto es, la ventana de tiempo de t_j^D es infinita, $H^-(t_j^D) = -\infty$, $H^+(t_j^D) = +\infty$.
3. En caso que t_j se dispare antes del mínimo tiempo en el modelo del sistema, esto es, $\vartheta < H^-(t_j^D) + \tau$, la correspondiente transición en el modelo diagnosticador $t_j^D \in T$ debe ser disparada después de que el diagnosticador identifica la transición de falta del sistema.
4. Después del disparo de $t_j^D \in T$ cuando $\vartheta > H^+(t_j^D) + \tau$, el modelo diagnosticador debe regresar al mercado anterior del disparo de t_j^D y los tiempos mínimos y máximos de disparo de t_j^D cambia como se indica a continuación:
 - $H^-(t_j^D) = 0$ y $H^+(t_j^D) = \infty$.
 - $\bar{M}_k = \bar{M}_{k+1} - [C][\gamma_k]$.

Existen marcas en el modelo diagnosticador que representan los valores actuales de las variables del estado del sistema (esos que pertenecen a los lugares medibles). Sin embargo, existen otras marcas que el modelo diagnosticador estima (esos que pertenece a

los lugares no medibles). Para diferenciar esas marcas, el modelo del diagnosticador asocia una etiqueta llamada cierta al primer tipo de marcas y una etiqueta llamada incierta para el segundo tipo de marcas. El módulo diagnosticador de falta utiliza esta información para detectar faltas o advertencias que ocurren en el sistema. La siguiente definición introduce la función de etiquetado de las marcas y su dinámica.

Definición 4.3 Sea $N = (Q, M_0)$ un sistema modelado con una RdPI viva, binaria, acotada y evento detectable del sistema Υ y $D = (Q_D, \bar{M}_0)$ el modelo diagnosticador de (Q, M_0) . Sea $Tokens$ un conjunto de elementos, definido como $Tokens = \bigcup_{i=1}^n Tokens_i$ y $Tokens_i = \{l_1^i, \dots, l_{q_i}^i\}$, donde $q_i = M^{UB}(p_i)$. La función de etiquetado del mercado es $\bar{M}^L : P_D \rightarrow 2^{Tokens}$, tal que $\bar{M}^L(p_i) \subseteq Tokens_i$ y $\bar{M}_k(p_i) = |\bar{M}_k^L(p_i)|$, asocia a cada lugar p_i un número de elementos de $Tokens_i$ igual a su mercado.

Definición 4.4 La función de posibilidad $\Theta : Tokens \rightarrow \{cierta, incierta\}$ determina cuál marca del mercado \bar{M}_k está representando los valores de las variables de estado actuales del sistema

$$\Theta_0(l_q^i) = \begin{cases} \text{cierta si } p_i \text{ es un lugar medible.} \\ \text{incierta otro caso} \end{cases}$$

donde $q = |Tokens_i|$

La definición anterior establece el etiquetado inicial de las marcas de los lugares del modelo del diagnosticador. Las marcas de los lugares medibles tienen la etiqueta cierta y las marcas de los lugares no medibles se les asocia la etiqueta incierta. Gráficamente, las marcas con etiqueta tipo incierta se representan por círculos claros en el interior del lugar.

Definición 4.5 La regla de disparo de la función de posibilidad Θ , definida anteriormente, es la siguiente

- Sea $\bar{M}_k^L(p_i) = \{l_1^i, \dots, l_h^i\} \cup \{l_{h+1}^i, \dots, l_u^i\}$ el valor de la función de etiquetado de mercado, donde $\forall l_j^i \in \{l_1^i, \dots, l_h^i\}$, $\Theta_k(l_j^i) = \text{cierta}$, $\forall l_g^i \in \{l_{h+1}^i, \dots, l_u^i\}$, $\Theta_k(l_g^i) = \text{incierta}$. Suponga que $M_k \xrightarrow{t_j} M_{k+1}$ ocurre en el modelo del sistema y $q = \min(|\{l_1^i, \dots, l_h^i\}|, I(p_i, t_j))$, $q_A = |\{l_1^i, \dots, l_h^i\}| - q$, $q_R = I(p_i, t_j) - q$, entonces
- $\bar{M}_{k+1}^L(p_i) = \{l_1^i, \dots, l_r^i\} \cup \{l_{r+1}^i, \dots, l_u^i\}$, donde
- $|\{l_1^i, \dots, l_r^i\}| = O(p_i, t_j) + \delta_k(i) + q_A$
- $|\{l_{r+1}^i, \dots, l_u^i\}| = |\{l_{h+1}^i, \dots, l_u^i\}| - q_R - \beta_k(i)$
- $\forall l_j^i \in \{l_1^i, \dots, l_r^i\}$, $\Theta_k(l_j^i) = \text{cierta}$, $\forall l_g^i \in \{l_{r+1}^i, \dots, l_u^i\}$, $\Theta_k(l_g^i) = \text{incierta}$.

La regla de disparo de la función de posibilidad Θ establece que cuando $M_k \xrightarrow{t_j} M_{k+1}$ ocurre en el modelo del sistema:

1. $t_j \in T_D$ se dispara en el modelo diagnosticador:
 - Se remueven $q + q_R$ marcas del lugar p_i , donde $|q + q_R| = I(p_i, t_j)$ y q son marcas con etiqueta tipo cierta y q_R son marcas con etiqueta tipo incierta.
 - Se depositan en el lugar p_i , $O(p_i, t_j)$ marcas con etiqueta tipo cierta.
2. $t_j \in T_D - T$ se dispara en el modelo diagnosticador:
 - Se remueven $\beta_k(i)$ marcas con etiqueta tipo incierta del lugar p_i .
3. $t_j \in T_D$ no se dispara en el modelo diagnosticador:
 - Se depositan en el lugar p_i , $\delta_k(i)$ marcas con etiqueta tipo cierta.

El módulo del diagnosticador de error (DE) es una función que está enfocada en detectar cuándo la salida del sistema es diferente de la salida del modelo del diagnosticador. La propiedad de evento detectabilidad del modelo del sistema se utiliza en DE : cuando un evento ocurre, entonces la salida del sistema y/o la salida del modelo del diagnosticador cambia; si esos cambios son diferentes entonces existe un error.

Definición 4.6 Sea $N = (Q, M_0)$ un sistema modelado en una RdPI viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$ y $D = (Q_D, \bar{M}_0)$ sea el modelo diagnosticador de (Q, M_0) . El diagnosticador de error es la función $DE : \mathbb{Z}^n \times \mathbb{Z}^n \longrightarrow \mathbb{Z}^n$ tal que $DE(\varphi(M_k), \varphi(\bar{M}_k)) = \varphi(M_k) - \varphi(\bar{M}_k)$

Una vez que DE detecta diferencia entre la salida del sistema y del modelo diagnosticador, el módulo diagnosticador de falta (DF) identifica la transición t_k y le asigna una etiqueta. Si un marca tipo cierta se usa para disparar la transición t_k , es decir, la marca del modelo del diagnosticador representa un recurso actual del sistema, entonces se determina que se ha producido un error. Sin embargo, si una marca tipo incierta se usa para disparar la transición t_k , es decir, la marca del modelo del diagnosticador es supuesta, entonces se produce una advertencia de error.

Definición 4.7 Sea $N = (Q, M_0)$ un sistema modelado con una RdPI viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$. Sea $S = (Q_S, \bar{M}_0)$ el modelo del observador de N y $D = (Q_D, \bar{M}_0)$ el modelo del diagnosticador de (Q, M_0) . El diagnosticador de falta es la función $DF : T \longrightarrow \{\text{advertencia, error}\}$. Esta función se calcula de la siguiente manera:

1. si el evento representado por la transición t_k se dispara en S y t_k no se dispara en el modelo diagnosticador entonces

- si t_k está habilitada en el modelo diagnosticador, entonces $DF(t_k) = \text{error}$
2. si el evento representado por la transición t_k no se dispara en S y t_k se dispara en el modelo diagnosticador, entonces
- si una marca etiquetada como cierta se usa para disparar t_k entonces $DF(t_k) = \text{error}$
 - si una marca etiquetada como incierta se usa para disparar t_k entonces $DF(t_k) = \text{advertencia}$

Como existen marcas que el diagnosticador estima, entonces las transiciones habilitadas en el diagnosticador y en el sistema no son las mismas. Afortunadamente, durante el comportamiento normal del sistema (cuando un error no se detecta), el marcado del modelo diagnosticador tiende al marcado del sistema, así, las transiciones habilitadas en el diagnosticador y las transiciones habilitadas en el sistema llegan a ser las mismas.

De esta forma, se puede usar el diagnosticador como una función de revisión de creencias. Para demostrar este hecho, se introducen las siguientes definiciones.

Definición 4.8 Sea $N = (Q, M_0)$ un sistema modelado en RdPI viva, binaria, acotada y evento detectable del sistema Υ y $D = (Q_D, \bar{M}_0)$ su modelo diagnosticador. Sea $T_N^F(M_k, \tau)$ el conjunto de transiciones de falta de N en el marcado M_k y tiempo τ , y $T_D^F(\bar{M}_k, \tau)$ el conjunto de transiciones de faltas de D , entonces $E_k(M_k, \tau) = T_N^F(M_k, \tau) - T_D^F(\bar{M}_k, \tau)$ es el error de transiciones de faltas en el marcado M_k y tiempo τ .

Si t_i se dispara en el marcado M_k alcanzando el marcado M_{k+1} en el tiempo τ , entonces, la notación $T_N^F(M_k, \tau^+)$ denota el conjunto de transiciones de faltas antes del disparo de t_i y $T_N^F(M_{k+1}, \tau^-)$ denota el conjunto de transiciones de faltas después del disparo de t_i . La misma notación se utiliza para el error de transiciones de faltas que se define a continuación.

Teorema 4.2 Sea $N = (Q, M_0)$ un sistema modelado en RdPI viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$ y $D = (Q_D, \bar{M}_0)$ el modelo diagnosticador de N . Sea $E_k = T_N^F(M_k, \tau) - T_D^F(\bar{M}_k, \tau)$ el error de transiciones de falta en el marcado k y tiempo τ . Entonces las siguientes sentencias se cumplen:

1. $\forall k, M_k \in R(Q, M_0), \bar{M}_k \in R(Q_S, \bar{M}_0), \varphi(M_k) = \varphi(\bar{M}_k)$
2. $\forall k, E_k(M_k, \tau) = T_N^F(M_k, \tau) - T_D^F(\bar{M}_k, \tau) = \emptyset$, donde $\alpha_k^- < \tau < \alpha_{k+1}^+$ y α_k^- es el tiempo cuando la red evoluciona de M_{k-1} a M_k y α_{k+1}^+ es el tiempo cuando la red evoluciona de M_k a M_{k+1} .

Demostración. La demostración de la sentencia 1 es similar a la que se demostró en el teorema 4.1.

Para demostrar la sentencia 2, comenzamos por inducción sobre la longitud de la secuencia de disparo σ .

1. Si $|\sigma| = 0$ entonces $k = 0$. Suponga que el tiempo actual del sistema es cero. Entonces, por definición $T_D^F(\bar{M}_0, 0) = T_N^F(M_0, 0) = \{\}$. Así $E_0(M_0, \alpha_0^-) = E_0(M_0, 0) = \emptyset$. Ahora, suponga que el tiempo del sistema se incrementa a τ_0 y uno de los siguientes casos se presenta:
 - Si una transición t_i^N está habilitada en el mercado inicial M_0 y es la primera transición disparada en τ_0 , donde $H^-(t_i^N) \leq \tau_0 \leq H^+(t_i^N)$, alcanzando el mercado M_1 , entonces esta transición no se incluye en el conjunto de transiciones de faltas $T_N^F(M_1, \tau_0)$. De acuerdo a las reglas de disparo del modelo del diagnosticador, esta transición también se dispara en el diagnosticador en el tiempo τ_0 . Así, esta transición no se incluye en $T_D^F(\bar{M}_1, \tau_0)$ y $E_1(M_1, \tau_0) = \emptyset$. Como otra transición no se dispara antes de t_i^N entonces $E_0(M_0, \tau_0^+) = \emptyset$. Note que otra transición no se dispara en el mercado M_0 , ya que el mercado actual es M_1 .
 - Existe al menos una transición t_i^N habilitada en el mercado inicial M_0 y $\tau_0 > H^+(t_i^N)$, pero t_i^N no se dispara en el sistema. Así $T_N^F(M_0, H^+(t_i^N)) = \{t_i^N\}$. Como $\bar{M}_0 \geq M_0$, entonces, por el lema de monoticidad [12] t_i^D también está habilitada en el mercado \bar{M}_0 . De acuerdo a las reglas de disparo del modelo del diagnosticador, t_i^D se dispara en el tiempo del sistema $H^+(t_i^N)$ alcanzando el mercado \bar{M}_1 . Como ambas redes, el diagnosticador y el modelo del sistema son evento detectable, $\varphi(M_0) \neq \varphi(\bar{M}_1)$. Entonces, el diagnosticador de error (DE) detecta esta diferencia y el diagnosticador regresa al mercado \bar{M}_0 , los tiempos mínimos y máximos de disparo de t_i^D cambian a $H^-(t_i^D) = 0$ y $H^+(t_i^D) = +\infty$, respectivamente y determina que $T_D^F(\bar{M}_0, H^+(t_i^N)) = \{t_i^D\}$. Así, $E_0(M_0, H^+(t_i^N)) = \emptyset$. Así, este procedimiento puede ser aplicado en varias ocasiones y podemos demostrar que $E_0(M_0, \tau_0) = \emptyset$, mientras M_0 no cambia.
 - Existe al menos una transición t_i^N habilitada en el mercado inicial M_0 , esta transición se dispara en el tiempo $\tau_0 < H^-(t_i^N)$ cambiando al mercado M_1 . Así $T_N^F(M_0, H^+(t_i^N)) = \{t_i^N\}$. Como $\tau_0 < H^-(t_i^N)$, t_i^D no se dispara en el modelo diagnosticador y $\varphi(M_1) \neq \varphi(\bar{M}_0)$. Entonces, el diagnosticador de error (DE) detecta esta diferencia, el diagnosticador evoluciona al mercado \bar{M}_1 y determina que $T_D^F(\bar{M}_0, \tau_0) = \{t_i^D\}$. Así $E_1(M_1, \tau_0) = \emptyset$. En otras palabras, al inicio del siguiente paso del sistema $|\sigma| = 1$, $E_1(M_1, \tau_0) = \emptyset$. Como otra transición no se dispara antes de t_i^N entonces $E_0(M_0, \tau_0^+) = \emptyset$. Note que otra transición no se dispara en el mercado M_0 , ya que el mercado actual es M_1 .

Observe que podemos demostrar $E_1(M_1, \alpha_1^-) = \emptyset$, es decir, cuando el mercado cambia a M_1 .

2. Por el paso inductivo, $|\sigma| = n$, $E_n(M_n, \alpha_n^-) = \emptyset$, $E_n(M_n, \tau) = \emptyset$, donde $\alpha_n^- < \tau < \alpha_{n+1}^+$, y $E_{n+1}(M_{n+1}, \alpha_{n+1}) = \emptyset$.
3. Si $|\sigma| = n + 1$ entonces $k = n + 1$. Del paso inductivo $E_{n+1}(M_{n+1}, \alpha_{n+1}) = \emptyset$. Uno de los siguientes casos se presenta:
 - Si una transición t_j^N está habilitada en el marcado M_{n+1} y se dispara en el tiempo τ_{n+1} , donde $H^-(t_j^N) + \alpha_{n+1} \leq \tau_{n+1} \leq H^+(t_j^N) + \alpha_{n+1}$, alcanzando el marcado M_{n+2} , entonces esta transición no se incluye en $T_N^F(M_{n+2}, \tau_{n+1} + \alpha_{n+1})$. De acuerdo a las reglas de disparo del diagnosticador, esta transición también se dispara en el diagnosticador en el tiempo τ_{n+1} . Así, esta transición no se incluye en $T_D^F(\bar{M}_{n+2}, \tau_{n+1})$ y $E_{n+2}(M_{n+2}, \tau_{n+1}) = \emptyset$. Como otra transición no se dispara antes de t_j^N , entonces $E_{n+1}(M_{n+1}, \tau_{n+1}^+) = \emptyset$. Observe que otra transición no puede ser disparada en el marcado M_{n+1} , ya que el marcado actual es M_{n+2} .
 - Existe al menos una transición t_j^N habilitada en el marcado M_{n+1} , y $\tau_{n+1} > H^+(t_j^N) + \alpha_{n+1}$, pero t_j^N no se dispara en el sistema. Así $T_N^F(M_{n+1}, H^+(t_j^N) + \alpha_{n+1}) = \{t_j^N\}$. Como $\bar{M}_{n+1} \geq M_{n+1}$, entonces, por el lema de monotonicidad, t_j^D es también habilitada en el marcado \bar{M}_{n+1} . De acuerdo a las reglas de disparo del diagnosticador, t_j^D se dispara en el tiempo del sistema $H^+(t_j^N) + \alpha_{n+1}$ alcanzando el marcado \bar{M}_{n+2} . Como ambas redes, el diagnosticador y el modelo del sistema son evento detectable $\varphi(M_{n+1}) \neq \varphi(\bar{M}_{n+2})$. Entonces, el diagnosticador de error (DE) detecta esta diferencia y el diagnosticador regresa al marcado \bar{M}_{n+1} . El mínimo y máximo período de tiempo de disparo de t_j^D cambia a $H^-(t_j^D) = 0$ y $H^+(t_j^D) = +\infty$ y determina que $T_D^F(\bar{M}_{n+1}, H^+(t_j^N) + \alpha_{n+1}) = \{t_j^D\}$. Así $E_{n+1}(M_{n+1}, H^+(t_j^N) + \alpha_{n+1}) = \emptyset$. Varias transiciones pueden ser disparadas cuando su máximo período de tiempo haya transcurrido. Así este procedimiento puede ser aplicado varias ocasiones y podemos demostrar que $E_{n+1}(M_{n+1}, \tau_{n+1}) = \emptyset$, mientras M_{n+1} no cambie.
 - Existe al menos una transición t_j^N habilitada en el marcado M_{n+1} , esta transición se dispara y $\tau_{n+1} < H^-(t_j^N) + \alpha_{n+1}$ alcanzando el marcado M_{n+2} . Así $T_N^F(M_{n+1}, H^-(t_j^N) + \alpha_{n+1}) = \{t_j^N\}$. Como $\tau_{n+1} < H^-(t_j^N) + \alpha_{n+1}$, t_j^D no se dispara en el modelo diagnosticador y $\varphi(M_{n+2}) \neq \varphi(\bar{M}_{n+1})$. Entonces, el diagnosticador de error (DE) detecta esta diferencia, el diagnosticador evoluciona al marcado \bar{M}_{n+2} y determina que $T_D^F(\bar{M}_{n+1}, \tau_{n+1}) = \{t_j^D\}$. Así $E_{n+2}(M_{n+2}, \tau_{n+1}) = \emptyset$. En otras palabras, al inicio del siguiente paso del sistema $|\sigma| = n + 2$, $E_{n+2}(M_{n+2}, \tau_{n+1}) = \emptyset$. Como otra transición no se dispara antes de t_j^N , entonces $E_{n+1}(M_{n+1}, \tau_{n+1}^+) = \emptyset$. Observe que otra transición no puede ser disparada en el marcado M_{n+1} , ya que ahora el marcado es M_{n+2} .

■

Podemos combinar los dos sistemas dinámicos, la función de revisión de creencias para generar creencias del estado del sistema y la función de revisión de creencias para generar creencias de las faltas del sistema y así obtener la arquitectura que se ilustra en la figura 4.3.

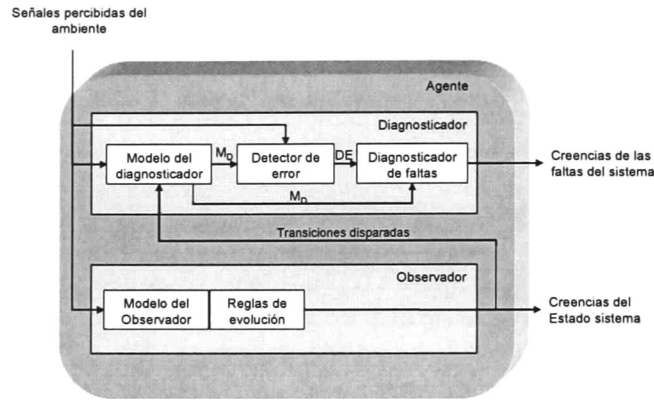


Figura 4.3: Función de revisión de creencias para generar creencias del estado y faltas del sistema.

4.3. Algoritmo para diseñar la función de revisión de creencias.

En esta sección se presenta un algoritmo para obtener el diseño de la función de revisión de creencias el cual está integrado por los modelos del observador y diagnosticador dedicados a generar creencias del estado del sistema y de las faltas del sistema, respectivamente

Algoritmo 4.2 *Diseño de la función de revisión de creencias de faltas del sistema*

Entradas: El modelo en *RdPI* que representa el ambiente del agente *A*.

Salidas: La función de revisión de creencias.

- Modelo observador
1. Construir la red del observador Q_S . Este es una copia del modelo en *RdPI* del ambiente del agente donde transiciones extras son incluidas $T_S = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$. En este caso, cada lugar tiene una transición extra de entrada y de salida. Las nuevas transiciones de entrada no tienen lugares de entrada y las nuevas transiciones de salida no tienen lugares de salida.
 2. Calcular el vector de disparo v_k con el algoritmo 4.1

3. La red Q_S y v_k conforman el modelo observador.
 - Modelo diagnosticador.
 1. Construir la red del diagnosticador Q_D . Este es una copia de la red Q_O . Se incluyen las funciones H^- , H^+ asociadas a las transiciones t_j que representan el período de tiempo de disparo.
 2. Definir las funciones del detector de error y diagnosticador de falta.
 3. Definir la función de etiquetado de marcas y transiciones.
 4. La red Q_D y las funciones detector de error, diagnosticador de falta, función de etiquetado de marcas y transiciones conforman al modelo diagnosticador.
 - Función de revisión de creencias.
 1. La función de revisión de creencias conformada por el observador y diagnosticador es la salida del algoritmo.

4.4. Ejemplo

Continuando con el ejemplo del sistema de la celda de manufactura de la figura 3.2, el agente tiene un modelo en *RdPI* del sistema. Como el sistema modelado en *RdPI* comprende lugares que generan una señal de salida cuando éstos están marcados (lugares medibles), se distingue cuál transición fue disparada. Como se mencionó en la sección de ejemplo 3.1.3, la red es evento detectable y el modelo del observador presentado en la figura 4.4, puede ser usado para generar creencias acerca del estado del sistema. Como los lugares p_1 , p_7 , p_4 , p_9 , p_{10} y p_{10+2i} son no medibles, el agente no puede percibir cuándo las máquinas 1 y 2 están disponibles, el robot 1 está libre, la máquina 2 está produciendo o el almacén en la posición i está vacío, respectivamente. Así, el modelo del observador es una copia del modelo del ambiente donde cada lugar tiene una transición extra de entrada y salida y el vector de disparo se calcula siguiendo el algoritmo 4.1.

El marcado inicial del modelo del observador se establece de tal forma que las señales de salida del observador sea igual que las señales del sistema, $\varphi\check{M}_0 = \varphi M_0$ donde $\varphi(M_0) = [0 \ 0 \ 1 \ 0 \ 0 \ 1]^T$. Además, los lugares no medibles inicialmente están marcados como lo establece la definición 4.1.

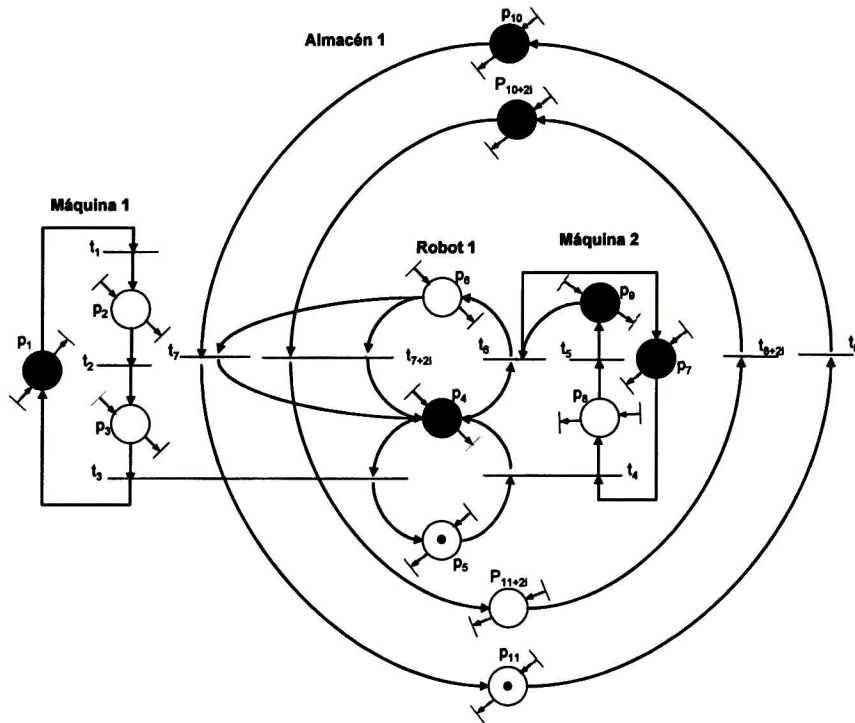


Figura 4.4: Modelo en *RdPI* del observador del sistema de la celda de manufactura.

El modelo del diagnosticador, representado en la figura 4.5, puede ser usado para generar creencias acerca de las faltas de las transiciones, esto es, faltas en las máquinas, robot o en el almacén. El agente está dedicado a detectar dos tipos de errores y determinar el elemento de falta. El primer tipo de error es cuando un sensor no se desactiva después del máximo tiempo para hacerlo. En este caso, este error se interpreta como si el componente del sistema está descompuesto o el sensor no está trabajando correctamente. El otro tipo de error es cuando un sensor se desactiva antes del mínimo tiempo para hacerlo, esto es, la transición se dispara antes del mínimo tiempo. Como las transiciones del modelo diagnosticador tiene una función que representa el mínimo y máximo tiempo de disparo, el agente conoce cuándo una transición debe ser disparada en el tiempo correcto. Cuando esto no sucede, el agente detecta el error y determina el elemento de falta usando la la función de error y la función de falta, respectivamente.

El marcado inicial del modelo del diagnosticador es igual que el modelo del observador. Las marcas de los lugares tienen una etiqueta de acuerdo a la definición 4.4. Las marcas de los lugares medibles tienen la etiqueta cierta y las marcas de los lugares no medibles se les asocia la etiqueta incierta. Gráficamente, las marcas con etiqueta tipo incierta se representan por círculos claros en el interior del lugar. De acuerdo a la definición de las *RdPIVT*, inicialmente las transiciones del modelo del diagnosticador tienen la etiqueta tipo normal.

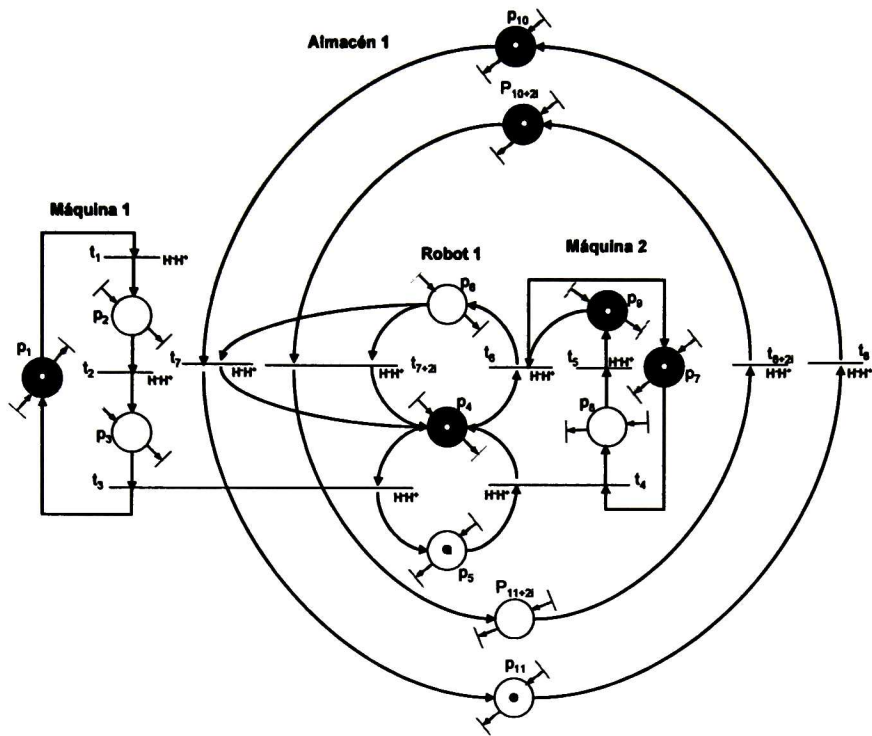


Figura 4.5: Modelo en *RdPIVT* del diagnosticador del sistema de la celda de manufactura.

El marcado inicial del observador y las etiquetas de las transiciones del modelo del diagnosticador representan las creencias iniciales que tiene el agente acerca del sistema de la celda de manufactura.

El agente cree que el robot 1 está libre. Si ésta no es la situación actual del sistema, eventualmente el $r1$ estará libre después de trasladar la pieza A de la $m1$ a la $m2$ o de almacenar la pieza B que produce la $m2$. Suponga que el $r1$ coloca la pieza tipo A en la $m2$, entonces la transición t_4 del modelo del sistema se dispara en el tiempo \mathbb{T}_j . El agente percibe la señal $\varphi M_1 = [0 \ 0 \ 0 \ 0 \ 1 \ 1]^T$ y calcula $\varphi M_1 - \varphi M_0 = [0 \ 0 \ -1 \ 0 \ 1 \ 0]^T$ para determinar qué transición se disparó en el modelo del sistema. Como $\varphi M_1 - \varphi M_0 = \varphi C(\bullet, 4)$ el agente distingue que la transición t_4 se disparó en el modelo del sistema. El modelo del observador dispara la transición t_4 . Como el agente percibe la señal de salida φM_1 en el tiempo \mathbb{T}_j y $H^-(t_4) + \tau_j \leq \mathbb{T}_j \leq H^+(t_4) + \tau_j$, donde τ_j es el tiempo en que se habilitó la transición t_4 , el diagnosticador dispara la transición y la etiqueta como normal. En el modelo del diagnosticador, al disparo de la transición t_4 , se remueve la marca tipo cierta del lugar p_5 y la marca con etiqueta tipo incierta del lugar p_7 ; se depositan marcas con etiqueta cierta en los lugares p_4 y p_8 . Como el lugar p_4 excede su cota máxima, es decir, $\bar{M}1(p_4) > 1$, existen dos marcas, una con etiqueta tipo cierta y la otra con etiqueta tipo incierta. En este caso, se dispara la transición extra de salida del lugar p_4 para remover una marca con etiqueta tipo incierta, ver figura 4.6.

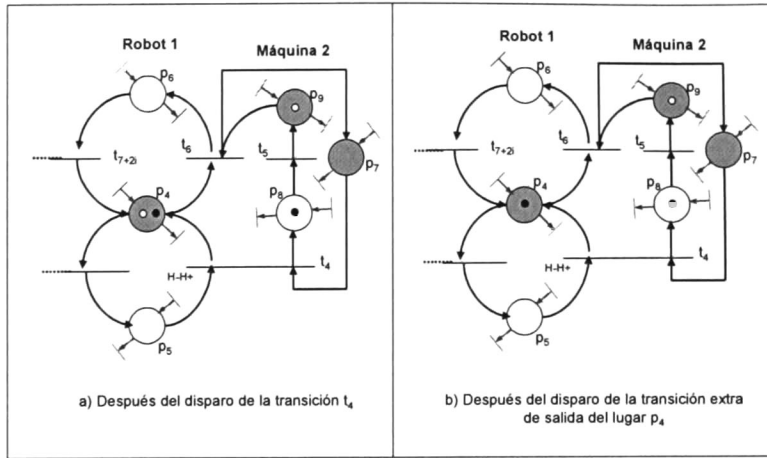


Figura 4.6: Regla de disparo de la función de posibilidad.

El nuevo marcado del observador y las etiquetas de las transiciones del modelo del diagnosticador representan la nueva creencia del agente. Esta creencia establece que el $r1$ está libre y que la $m2$ está procesando la pieza tipo A .

Ahora, suponga que el tiempo actual del sistema es \mathbb{T}_m , donde $\mathbb{T}_m > H^+(t_5) + \mathbb{T}_j$. Esto significa que la transición t_5 no se disparó dentro de la ventana de tiempo porque la $m2$ tardó más en procesar la pieza o el sensor 5 está descompuesto. Como el agente conoce los tiempos en que se deben disparar las transiciones y en el tiempo \mathbb{T}_m no ha percibido ningún cambio de señal de salida del sistema, el modelo del diagnosticador dispara la transición t_5 . El módulo DE del diagnosticador, detecta que existe una diferencia entre la señal de salida del sistema y del diagnosticador, $DE(\varphi M_1, \varphi \bar{M}_2) \neq 0$, esto es $\varphi M_1 \neq [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$. El módulo DF identifica que la transición t_5 no se disparó ($\varphi M_1 - \varphi \bar{M}_2 = \varphi C(\bullet, 5)$). Como el disparo de la transición t_5 en el modelo del diagnosticador fue con una marca tipo cierta, el DF etiqueta a la transición como falta. Una vez que el agente identifica el elemento de falta, el modelo del diagnosticador regresa al marcado \bar{M}_1 para que $\varphi M_1 = \varphi \bar{M}_1$ y los tiempos mínimos y máximos de disparo de la transición t_5 cambian a $H^-(t_5) = 0$ y $H^+(t_5) = \infty$.

La nueva creencia del agente es que existe una falta en la $m2$.

4.5. Conclusiones

En este capítulo se presentó un enfoque algebraico llamado *observador – diagnosticador* para representar la función de revisión de creencias de un agente. El entorno del agente es un SED el cual es modelado con $RdPI$. El mecanismo para generar creencias ambientales está compuesto por dos sistemas dinámicos enfocados a generar creencias del estado del sistema y creencias acerca de las faltas del sistema.

El primer sistema dinámico llamado observador es una copia del modelo del sistema más transiciones extras de entrada y salida. Esta estructura adicional permite generar creencias consistentes acerca del estado del sistema. El segundo sistema dinámico llamado diagnosticador es una copia del modelo del observador más información adicional. Esa información adicional asocia una etiqueta a las marcas y a las transiciones. Para el primer caso, permite distinguir aquellas marcas que ya conoce el agente (tipo *cierta*) con aquellas que no percibe del ambiente (tipo *incierto*). Esta distinción se hizo con la finalidad de dar mayor certeza a las creencias del agente con respecto a las faltas del sistema. Para el segundo caso, se asocia una función de tiempo a cada una de las transiciones para determinar la duración de eventos del sistema. Si el agente detecta el disparo de una transición en el modelo del sistema fuera de la ventana de tiempo, el agente etiqueta la transición como falta o advertencia dependiendo del tipo de marca con la que se disparó.

Asimismo, se demostró que las *RdPI* son una herramienta adecuada para representar el mecanismo de la función de revisión de creencias y que cumple con los axiomas que debe satisfacer la función de revisión de creencias. Por último, se presentó un algoritmo polinomial para diseñar la función de revisión de creencias compuesto por dos sistemas dinámicos llamados *observador* y *diagnosticador*.

Capítulo 5

Generador de creencias extendido implementado en RdPIVT

Resumen. Este capítulo presenta el segundo enfoque propuesto en este trabajo, el cual es una extensión del primer planteamiento y consiste en diseñar una función de revisión de creencias compuesto por un sistema dinámico llamado *observador extendido*, enfocado a generar creencias acerca del estado del sistema y de las posibles faltas del sistema simultáneamente.

Este capítulo presenta una extensión del enfoque expuesto en el capítulo 4, donde se utiliza un sistema dinámico llamado *observador extendido*. El *observador extendido* fusiona la teoría de observador y diagnosticador para representar el mecanismo de la función de revisión de creencias ambientales.

En este enfoque, el *observador extendido* es una copia del modelo del sistema más una estructura adicional. La estructura adicional consiste en transiciones extras de salida que permiten obtener creencias consistentes con el estado del sistema. El observador extendido está enfocado en generar creencias del estado y faltas intermitentes del sistema simultáneamente. Las creencias del agente son representadas por la estructura de la *RdPI* del modelo del sistema, el marcado y el etiquetado de transiciones del observador extendido. El mecanismo para generar nuevas creencias se representa por una apropiada regla de disparo [30]. Esta regla de disparo está basada en las percepciones del agente, creencias actuales y la estructura de la *RdPI* del observador extendido.

La definición formal del observador extendido es la siguiente:

Definición 5.1 Sea $N_S = (Q, M_0)$ un sistema modelado en *RdPIVT* viva, binaria, acotada y evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi, H^+, H^-)$, $|P| = n$, $|T| = m$. El modelo del observador extendido $N_O = (Q_O, \hat{M}_0)$ de (Q, M_0) es la *RdPIVT* donde:

1. $Q_O = (P_O, T_O, I_O, O_O, \Sigma, Id, Id, H^+, H^-, T_k^L)$, donde

- $P_O = P$,
- $T_O = T \cup F$, donde $F = \{t_1^o, \dots, t_n^o\}$ es el conjunto de transiciones extras de salida.
- H^-, H^+ representan la ventana de tiempo que una transición habilitada $t_j \in T$ debe ser disparada. Por definición $H^-(t_j^O) = -\infty, \forall t_j^O \in T_O - T$. y $H^+(t_j^O) = -\infty, \forall t_j^O \in T_O - T$, respectivamente.
- $T_k^L : T_O \rightarrow \{\text{normal}, \text{falta}\}$ es una función que mapea cada transición $t_j \in T$ a una etiqueta tipo normal o falta como sigue:

$$T_k^L(t_j) = \begin{cases} \text{normal} & \text{Si } t_j \text{ se dispara en el tiempo } \tau_{t_j} \\ & \text{y } H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j} \\ \text{falta} & \text{otro caso} \end{cases}$$

donde χ_{t_j} es el tiempo cuando t_j se habilita en el marcado M_k y τ_{t_j} el tiempo actual del sistema,

2. La ecuación de estados es:

$$\begin{aligned} \hat{M}_{k+1} &= \hat{M}_k + [C \quad -I] v_k \\ \hat{y}_k &= Id(\hat{M}_k) \end{aligned}$$

y Id es la función identidad y I es la matriz identidad. I_S y O_S son las funciones de entrada y salida de la *RdPI*

3. El mercado inicial \hat{M}_0 de (Q_0, \hat{M}_0) es

$$\bullet \hat{M}_0(p_i) = \begin{cases} M_0(p_i) & \text{si } p_i \text{ es un lugar medible} \\ M^{UB}(p_i), & \text{otro caso.} \end{cases}$$

4. La regla de disparo es el vector de disparo que resulta del siguiente algoritmo. Sea Γ la matriz $\Gamma [i, j] = \gamma_j^i$ que representa el conjunto de las LCM y K es el vector $K(i) = k_i$

Algoritmo 5.1 Cálculo del vector de disparo y mercado actual

Entradas: C - Matriz de incidencia del modelo del sistema (Q, M_0) ,

t_j - Transición que se dispara en el modelo del sistema (Q, M_0)

\hat{M}_{act} - mercado actual del modelo del observador extendido

Salidas:

\hat{M}_k mercado actual del modelo del observador extendido

v_k el vector de disparo

Si $(\hat{M}_{act} + C(t_j) \geq 0)$ entonces

$$\Omega = \begin{cases} v_k(t_j) = 1 \\ v_k(t_i) \geq 0 \\ \forall t_i \in T, \text{ tal que } \varphi C(\bullet, i) \neq \vec{0} \\ v_k(t_i) \leq 1 \\ \forall t_i \in T, \text{ tal que } \varphi C(\bullet, i) \neq \vec{0} \\ \Delta = 0 \end{cases}$$

de lo contrario

$$\Delta = \begin{cases} v_k(i) = 0 \\ \varphi C(\bullet, i) \neq \vec{0} \\ \Omega = 0 \end{cases}$$

$$PPL_2 = \left\{ \begin{array}{l} \text{mín } \sum v_k \\ \text{s.a.} \\ \hat{M}_k = \hat{M}_{act} + C \sum v_k \\ \hat{M}_k(p_l) \geq M^{lB}(p_l) \\ \varphi^{(\bullet, l) = \bar{0}} \\ \hat{M}_k(p_l) \leq M^{UB}(p_l) \\ \varphi^{(\bullet, l) = \bar{0}} \\ \sum \hat{M}_k(l) = \varphi M_k(m) \\ \varphi^{(m, l) \neq \bar{0}} \\ v_k(i) \geq 0 \\ v_k(i) \leq 1 \\ v_k(i) \geq 0 \\ i > |T| \\ v_k(i) \leq 1 \\ i > |T| \\ v_k(i) \geq 0 \\ \varphi C^{(\bullet, i) = 0} \\ v_k(i) \leq 1 \\ \varphi C^{(\bullet, i) = 0} \\ \Omega \\ \Delta \end{array} \right.$$

La definición anterior establece que cuando el marcado de un lugar excede la cota máxima de ese lugar, esas marcas son removidas por el disparo de las transiciones extras de salida. Usando este vector de disparo, el estado de N_O tenderá al estado del sistema si éste es evento detectable, como probaremos en el siguiente teorema.

Proposición 5.1 *Sea $N_S = (Q, M_0)$ un sistema modelado en RdPIVT viva, binaria, acotada y evento detectable del sistema Υ y sea $N_O = (Q_O, \hat{M}_0)$ el modelo del observador extendido en RdPIVT, el vector de transiciones extras de salida $\beta_k[j] = v_k[k]$ tal que $m < k \leq m + n$. Entonces*

$$\forall k \in \mathbb{Z}^+ \quad [\hat{M}_k - M_k] \leq [\hat{M}_{k-1} - M_{k-1}]$$

Demostración. Sea $|\sigma| = 0$. Por definición, $\hat{M}_0 = M_0 + \Delta\hat{M}_0$, donde $\Delta\hat{M}_0 \geq 0$. En este caso,

$$\begin{aligned} e_0(p_i) &= \hat{M}_0(p_i) - M_0(p_i) \\ &= [(M_0(p_i) + \Delta\hat{M}_0(p_i))] - M_0(p_i) \\ &= \Delta\hat{M}_0(p_i) \end{aligned}$$

Ahora, sea $|\sigma| = 1$. Suponga que $t_j \in T$ ocurre en N_S , esto es, $M_0 \xrightarrow{t_j} M_1$. Como N_S es evento detectable, el disparo de t_j en N_S se distingue y, como $\hat{M}_0 = M_0 + \Delta\hat{M}_0$, por el lema de monotonicidad [12], t_j está habilitada en el marcado \hat{M}_0 , así que ésta se dispara en N_O , y entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. Los siguientes casos son posibles:

1. a) $\hat{M}_1(p_i) \leq M^{UB}(p_i)$. En este caso, $\beta_1(i) = 0$

$$\begin{aligned}\hat{M}_1 &= \hat{M}_0 + C(p_i, \cdot) \vec{\hat{t}}_j - \beta_1(i) \\ &= \left[(M_0(p_i) + \Delta\hat{M}_0(p_i)) + C(p_i, \cdot) \vec{\hat{t}}_j \right] \\ &= M_1(p_i) + \Delta\hat{M}_0(p_i)\end{aligned}$$

Así,

$$\begin{aligned}e_1(p_i) &= \hat{M}_1(p_i) - M_1(p_i) \\ &= [M_1(p_i) + \Delta\hat{M}_0(p_i)] - M_1(p_i) \\ &= \Delta\hat{M}_0(p_i)\end{aligned}$$

Por lo tanto $e_1 = e_0$.

b) $\hat{M}_1(p_i) > M^{UB}(p_i)$. En este caso, $\beta_1(i) > 0$

$$\begin{aligned}\hat{M}_1(p_i) &= \hat{M}_1(p_i) - \beta_1(i) \\ &= M_1(p_i) + \Delta\hat{M}_0(p_i) - \beta_1(i) \\ &= M_1(p_i) + \Delta\hat{M}_1(p_i)\end{aligned}$$

Así,

$$\begin{aligned}e_1(p_i) &= \hat{M}_1'(p_i) - M_1(p_i) \\ &= [M_1(p_i) + \Delta\hat{M}_1(p_i)] - M_1(p_i) \\ &= [M_1(p_i) + \Delta\hat{M}_0(p_i) - \beta_1(i)] - M_1(p_i) \\ &= \Delta\hat{M}_0(p_i) - \beta_1(i) \\ &= e_0(p_i) - \beta_1(i)\end{aligned}$$

Como $\beta_1(i) \geq 0$ entonces $e_1(p_i) \leq e_0(p_i)$.

Siguiendo un análisis similar, para $|\sigma| = k$, esto es fácil ver que el error nunca se incrementa, por tanto $e_k \leq e_{k-1}$.

■

Esto significa que, conforme el ambiente del agente evoluciona, las creencias del agente del estado del sistema permanecerán igual o serán mejoradas, esto es, cada información del ambiente que se percibe por el agente se incorpora al conjunto de creencias preservando la consistencia.

Por otro lado, el observador extendido está enfocado a generar creencias acerca de las faltas del *SED*, esto es, creencias acerca de la correcta ejecución de los eventos del ambiente. Estas creencias son generadas mediante la función de etiquetado de transición T_K^L que asocia a cada transición una etiqueta tipo falta o normal cuando su disparo está dentro del período de tiempo permitido.

El etiquetado inicial de una transición $t_j \in T$ es definido dependiendo del tipo de lugar de entrada p_i de $t_j \in T$ como sigue:

Definición 5.2 Sea $\hat{T}_k^L : T \rightarrow \{\text{normal}, \text{falta}\}$ la función de etiquetado de transición. El etiquetado inicial de una transición $t_j \in T$ es el siguiente:

$$\hat{T}_0^L(t_j) = \begin{cases} \text{falta} & \text{si } \exists p_i \in \cdot(t_j) \text{ tal que } p_i \text{ es un lugar no medible} \\ \text{normal} & \text{caso contrario} \end{cases}$$

Definición 5.3 Sea $N_S = (Q, M_0)$ un sistema modelado en RdPIVT viva, binaria, acotada y evento detectable del sistema Υ . Sea $N_O = (Q_O, \hat{M}_0)$ el modelo del observador extendido en RdPIVT y $T_O = T \cup F$ el conjunto de transiciones de N_O . La diferencia de etiquetado de transiciones T_k^L en el momento k entre N_S y N_O es

$$[\hat{T}_k^L - T_k^L] = \sum_{i=0}^m \hat{T}_k^L(t_i) \otimes T_k^L(t_i)$$

donde

$$\hat{T}_k^L(t_i) \otimes T_k^L(t_i) = \begin{cases} 1, & \text{si } \hat{T}_k^L(t_i) \neq T_k^L(t_i) \\ 0, & \text{otro caso} \end{cases}$$

y $m=|T|$

Al igual que las creencias del estado del sistema, conforme el ambiente del agente evoluciona, las creencias de las faltas del sistema permanecerán igual o serán mejoradas, esto es, Así, cada que el agente percibe información del ambiente se incorpora al conjunto de creencias preservando la consistencia. Esto se demuestra a

Proposición 5.2 Sea $N_S = (Q, M_0)$ un sistema modelado en RdPIVT viva, binaria, acotada y evento detectable del sistema Υ y sea $N_O = (Q_O, \hat{M}_0)$ el modelo observador RdPIVT.

$$\forall k, \quad [\hat{T}_k^L - T_k^L] \leq [\hat{T}_{k-1}^L - T_{k-1}^L]$$

Demostración. Procedemos por inducción en la longitud de la secuencia de disparo σ .

1. Si $|\sigma| = 1$, entonces $k = 1$. Suponga que t_j está habilitada en el tiempo χ_{t_j} . Como el modelo del observador extendido tiene más marcas que el modelo del sistema, entonces t_j está habilitada en el mismo tiempo. Por definición de la función de etiquetado de transición $\hat{T}_0^L \geq T_0^L$.

a) Suponga que $\hat{T}_0^L(t_j) \neq T_0^L(t_j)$, entonces, los siguientes casos son posibles:

- 1) Si t_j se dispara en el modelo del sistema en el tiempo τ_{t_j} y $H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j}$ entonces usando la función de etiquetado de transición,

$$T_1^L(t_j) = \text{normal}.$$

Por la propiedad evento detectabilidad de N_S , se detecta el disparo de la transición t_j del sistema. Entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. Como t_j fue disparada en el tiempo τ_{t_j} , tal que $H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j}$, entonces

$$\hat{T}_1^L(t_j) = \text{normal}.$$

Así

$$\hat{T}_1^L(t_j) = T_1^L(t_j)$$

Como $\hat{T}_0^L(t_j) \neq T_0^L(t_j)$ entonces $[\hat{T}_1^L - T_1^L] < [\hat{T}_0^L - T_0^L]$.

- 2) Si t_j se dispara en el modelo del sistema en el tiempo τ_{t_j} y $\tau_{t_j} < H^-(t_j) + \chi_{t_j}$ o $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$, entonces usando la función de etiquetado de transición

$$T_1^L(t_j) = \text{falta}$$

Por la propiedad evento detectabilidad de N_S , se distingue el disparo de la transición del sistema. Entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. Como t_j fue disparada en el tiempo τ_{t_j} , tal que $\tau_{t_j} < H^-(t_j) + \chi_{t_j}$ o $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$, entonces

$$\hat{T}_1^L(t_j) = \text{falta}.$$

así

$$\hat{T}_1^L(t_j) = T_1^L(t_j)$$

Cómo $\hat{T}_0^L(t_j) \neq T_0^L(t_j)$ entonces $[\hat{T}_1^L - T_1^L] < [\hat{T}_0^L - T_0^L]$

- 3) Si t_j no se dispara en el modelo del sistema y el tiempo actual es τ_{t_j} , tal que $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$, entonces

$$T_1^L(t_j) = \text{falta}$$

y

$$\hat{T}_1^L(t_j) = \text{falta}.$$

Observe que el error de observación nunca incrementa, por lo tanto $[\hat{T}_1^L - T_1^L] < [\hat{T}_0^L - T_0^L]$

- b) Suponga que $\hat{T}_0^L(t_j) = T_0^L(t_j)$, similar al paso anterior, tres casos son posibles:

- 1) Si t_j se dispara en el modelo del sistema en el tiempo τ_{t_j} y $H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j}$, entonces usando la función de etiquetado de transición

$$T_1^L(t_j) = \text{normal}.$$

Por la propiedad evento detectabilidad de N_S , se distingue el disparo de la transición del sistema, entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. Como t_j fue disparada en el tiempo τ_{t_j} , tal que $H^-(t_j) + \chi_{t_j} \leq \tau_{t_j} \leq H^+(t_j) + \chi_{t_j}$, entonces

$$\hat{T}_1^L(t_j) = \text{normal}.$$

Así

$$\hat{T}_1^L(t_j) = T_1^L(t_j)$$

Como $\hat{T}_0^L(t_j) = T_0^L(t_j)$ entonces $[\hat{T}_1^L - T_1^L] = [\hat{T}_0^L - T_0^L]$.

- 2) Si t_j se dispara en el modelo del sistema en el tiempo τ_{t_j} y $\tau_{t_j} < H^-(t_j) + \chi_{t_j}$ o $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$ entonces usando la función de etiquetado de transición

$$T_1^L(t_j) = \text{falta}$$

Por la propiedad evento detectabilidad de N_S , se distingue el disparo de la transición del sistema, entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. Como t_j fue disparada en el tiempo τ_{t_j} tal que $\tau_{t_j} < H^-(t_j) + \chi_{t_j}$ o $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$ entonces

$$\hat{T}_1^L(t_j) = \text{falta.}$$

así

$$\hat{T}_1^L(t_j) = T_1^L(t_j)$$

Como $\hat{T}_0^L(t_j) = T_0^L(t_j)$ entonces $[\hat{T}_1^L - T_1^L] = [\hat{T}_0^L - T_0^L]$

- 3) Si t_j no se dispara en el modelo del sistema y el tiempo actual es τ_{t_j} , tal que $\tau_{t_j} > H^+(t_j) + \chi_{t_j}$ entonces

$$T_1^L(t_j) = \text{falta}$$

y

$$\hat{T}_1^L(t_j) = \text{falta.}$$

Como $\hat{T}_0^L(t_j) = T_0^L(t_j)$ entonces $[\hat{T}_1^L - T_1^L] = [\hat{T}_0^L - T_0^L]$.

Observe que el error de observación nunca incrementa, por lo tanto $[\hat{T}_1^L - T_1^L] < [\hat{T}_0^L - T_0^L]$

2. Ahora, suponga que $|\sigma| = k$, usando un razonamiento similar a punto anterior, es fácil ver que $[\hat{T}_k^L - T_k^L] \leq [\hat{T}_{k-1}^L - T_{k-1}^L]$

■

Así, como el ambiente del agente evoluciona, las creencias erróneas de las faltas del sistemas son removidas y reemplazadas por nuevas creencias más correctas, preservando la consistencia del conjunto de creencias de faltas.

5.1. Algoritmo para diseñar la función de revisión de creencias.

En esta sección se presenta un algoritmo para obtener el diseño de la función de revisión de creencias el cual está compuesta por el modelo del observador extendido dedicado a generar creencias del estado del sistema y de las faltas del sistema.

Algoritmo 5.2 *Diseño de la función de revisión de creencias*

Entradas: El sistema modelado en $RdPIVT$ que representa el ambiente del agente A .

Salidas: La función de revisión de creencias

1. Construir la red del observador Q_O . Este es una copia del modelo en $RdPIVT$ del ambiente del agente donde transiciones extras son incluidas $T_O = T \cup \{t_1^o, \dots, t_n^o\}$. Cada lugar tiene una transición extra de salida, donde las nuevas transiciones de salida no tienen lugares de salida.
 2. Calcular el vector de disparo v_k con el algoritmo 5.1.
 3. Obtener la función de etiquetado de transición.
 4. La función de revisión de creencias conformada por la red Q_O, v_k y la función de etiquetado de transición es la salida del algoritmo.
-

5.2. Ejemplo

Considere el sistema del estacionamiento automático representado en la figura 5.1 y suponga que existe un agente dedicado a controlar la entrada de automóviles al estacionamiento automático. Este sistema consiste de las siguientes partes. Una entrada al estacionamiento automático (PLE), cien divisiones para el estacionamiento de automóviles ($D_i, 1 \leq i \leq 100$), una salida del estacionamiento automático ($PLEx$), y un camino de llegada a las divisiones del estacionamiento (FRD), el camino es discretizado en sitios; existe un sitio, llamado FRD_i , por cada D_i . Existe un sensor en cada sitio y éste está activo cuando un automóvil está en el sitio; de lo contrario el sensor está desactivado.

El funcionamiento del sistema es el siguiente. Si existe una división disponible para estacionar un automóvil o no hay más de nueve automóviles esperando una división disponible, entonces el agente permite que el automóvil entre al estacionamiento usando la entrada del estacionamiento (PLE). Una vez que el automóvil entra al estacionamiento, el automóvil busca una división vacía. Cuando existen varias divisiones disponibles D_i , entonces los

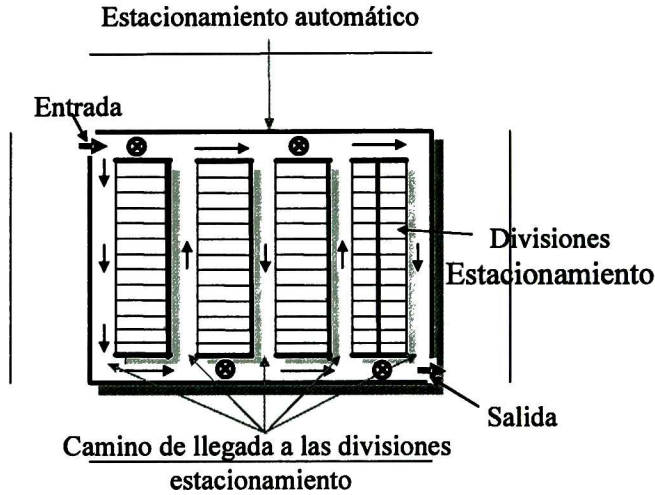


Figura 5.1: Sistema del estacionamiento automático.

automóviles seleccionan una de forma aleatoria. Un automóvil sale del estacionamiento usando la salida ($PLEx$).

Siguiendo la metodología presentada en [37], los siguientes pasos son considerados.

1. Doscientos dos componentes son identificados: la entrada del estacionamiento PLE , el camino de llegada discretizado en 100 sitios FRD_i , cien divisiones para estacionamiento D_i , y la salida del estacionamiento $PLEx$.
2. La siguiente tabla muestra los componentes y las variables de estados necesarios para representar el comportamiento de esos componentes:

Componentes	Variables de estado
PLE	PLE_valor
FRD_i	FRD_valor_i
D_i	D_valor_i
$PLEx$	$PLEx_valor$

3. El rango de esas variables de estado se representa en la siguiente tabla:

Variables de estado	Rango de las variables de estado
PLE_valor	$desocupado, ocupado$
FRD_valor_i	$desocupado_i, ocupado_i$
D_valor_i	$desocupado_i, ocupado_i$
$PLEx_valor$	$desocupado, ocupado$

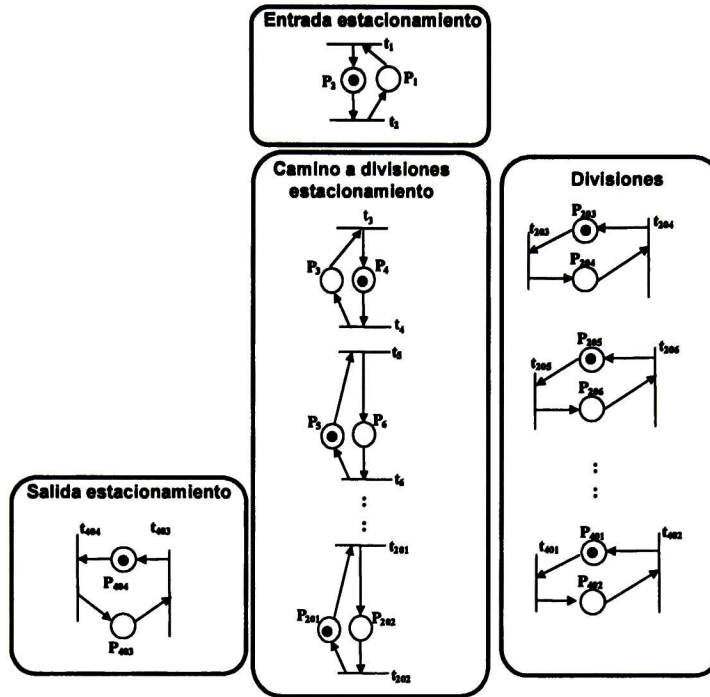


Figura 5.2: Módulos en RdP de los componentes del sistema del estacionamiento automático.

4. Estos rangos son codificados en la $RdPI$, usando el siguiente conjunto de lugares:

Variables de estado	Conjunto de lugares
PLE_valor	$\{p_1, p_2\}$
FRD_valor_i	$\{p_{3+2j}, p_{4+2j}\}$
D_valor_i	$\{p_{203+2j}, p_{204+2j}\}$
$PLEx_valor$	$\{p_{403}, p_{404}\}$

donde $j = 0..,99$

5. Los módulos en RdP representan el comportamiento de las variables de estado y son representados en la figura 5.2.
6. Después, la integración de módulos se realiza usando la composición sincrónica y permisiva.
 - a) Durante la composición sincrónica, las transiciones que modelan la misma tarea deben ser fusionadas en una sola transición. Por ejemplo, la transición t_2 y transición t_3 tienen el mismo significado físico (el automóvil entra al estacionamiento), entonces esas transiciones son fusionadas en una sola.
 - b) En este ejemplo, la composición permisiva no aplica ya que no existen lugares de un módulo que habiliten la transición de otro(s) módulo(s), al disparo de la

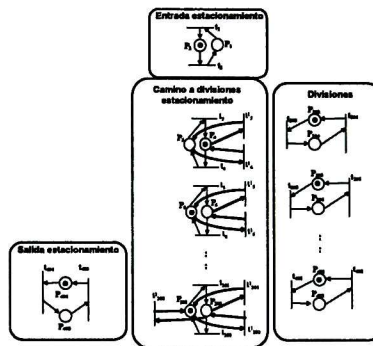


Figura 5.3:

transición no se pierde su marca. Sin embargo, se deben añadir un arco entre transiciones y lugares de diferentes módulos. Por ejemplo, los lugares p_{3+2j} permiten el disparo de t_{203+2j} cuando este lugar está marcado. Así, se añade un arco de p_{3+2j} a t_{203+2j} .

El modelo final en *RdPI*, se representa en la figura 5.4, donde los lugares blancos representan los elementos del sistema con sensores, es decir, el camino de llegada a las divisiones, la entrada y salida del estacionamiento.

Siguiendo con el ejemplo, el agente tiene un modelo en *RdPI* del sistema del estacionamiento automático. Como el sistema modelado en *RdPI* comprende lugares que generan una señal de salida cuando éstos están marcados (lugares medibles), se distingue cuál transición fue disparada. Por lo tanto esta red es evento detectable y el modelo del observador extendido presentado en la figura 5.5, puede ser usado para generar creencias acerca del estado del sistema. Como los lugares p_{203+2j} y p_{204+2j} son no medibles, el agente no puede percibir cuándo una división está ocupada por un automóvil. Así, el modelo del observador extendido es una copia del modelo del ambiente, donde cada lugar tiene una transición extra de salida y el vector de disparo se calcula siguiendo el algoritmo 5.1.

Asimismo, el modelo del observador extendido puede ser usado para generar creencias acerca de las faltas de las transiciones, esto es, faltas en las divisiones o faltas en el camino de llegada a las divisiones. El agente está dedicado a detectar dos tipos de errores y determinar el elemento de falta en el respectivo nivel del estacionamiento automático. El primer tipo de error es cuando un sensor no se desactiva después del máximo tiempo para hacerlo. En este caso, este error se interpreta como si el automóvil está descompuesto en el camino de llegada a las divisiones o el sensor no está trabajando correctamente. El otro tipo de error es cuando un sensor se desactiva antes del mínimo tiempo para hacerlo, esto es, la transición se dispara antes del mínimo tiempo una vez habilitada. Como las transiciones del modelo del observador extendido tienen una función que representan el mínimo y máximo tiempo de disparo una vez que éstas se habilitan, el agente conoce

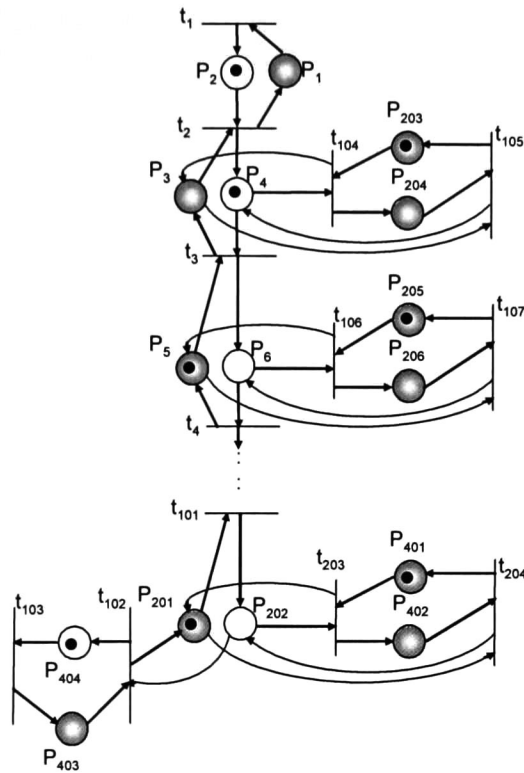


Figura 5.4: *RdPI* obtenida aplicando la composición sincrónica y permisiva.

cuándo una transición debe ser disparada en el tiempo correcto. Cuando esto no sucede, el agente determina el elemento de falta.

Además, el agente está dedicado a controlar la entrada de automóviles en el estacionamiento. Este control está basado en el conocimiento del número total de automóviles en el estacionamiento y el número total de elementos de falta. La información relacionada al número total de automóviles en el camino se percibe por los sensores del sistema.

El marcado inicial del modelo del observador extendido se interpreta que existe un automóvil en la entrada y salida del estacionamiento y otro automóvil en el camino discretizado no. 1 (lugares medibles marcados en p_2 , p_{404} y p_4 respectivamente). Los lugares no medibles están marcados de acuerdo a la definición 5.1.

El marcado inicial del modelo del observador extendido representa la creencia inicial del agente acerca del estado del sistema del estacionamiento automático. Por ejemplo, el agente cree que la división 1 está ocupada. Como no es la condición actual del estacionamiento, eventualmente un automóvil llegará a la división 1. Suponga que el automóvil que está en el camino discretizado no. 1 ocupa la división 1, esto es, la transición t_{104} del modelo del ambiente se dispara en el tiempo \mathcal{T}_j . El agente percibe la señal de salida del

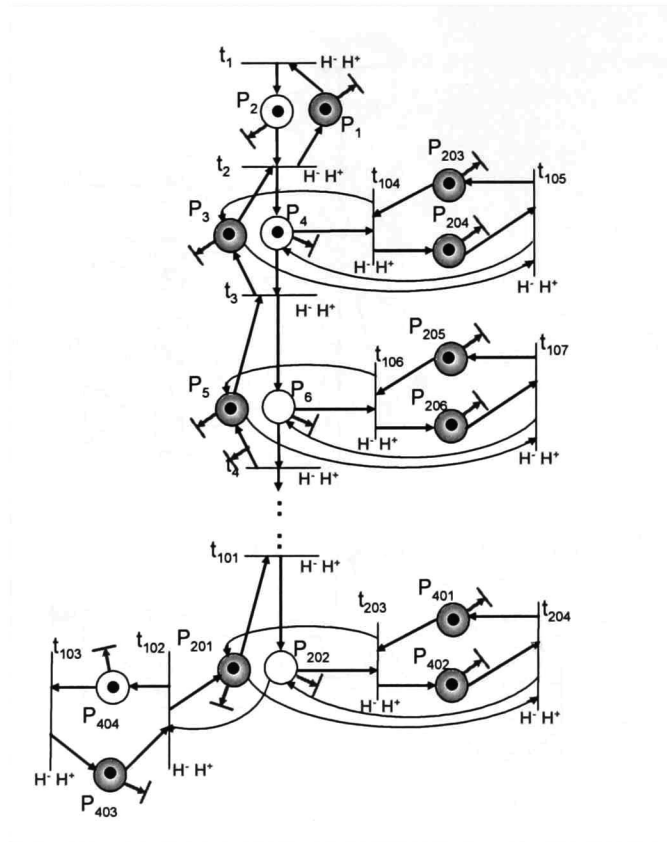


Figura 5.5: El modelo en *RdPIVT* del observador extendido del primer nivel del sistema del estacionamiento automático.

sistema $\varphi M_1 = [1 \ 0 \ \dots \ 1]^T$, y calcula $\varphi M_1 - \varphi M_0 = [0 \ -1 \ \dots \ 0]^T$. El agente distingue que la transición t_{104} fue la que se disparó ya que el modelo del observador extendido es evento detectable. Como el modelo del observador extendido tiene más marcas que el modelo del sistema, la transición t_{104} está habilitada y ésta se dispara. El nuevo marcado del lugar p_{204} es mayor a su cota máxima, es decir, $\hat{M}_1(p_{204}) > 1$, entonces se dispara la transición extra de salida de ese lugar. Además, el disparo de t_{104} en el modelo del sistema en el tiempo $H^-(t_{104}) + \tau_j \leq \mathbb{F}_j \leq H^+(t_{104}) + \tau_j$, donde τ_j es el tiempo en que la transición t_{104} se habilitó, el observador extendido etiqueta la transición como normal. Así, se genera un nuevo marcado y el agente tiene una nueva creencia. Esta nueva creencia establece que existe un automóvil en la división 1 y que el camino 1 está libre.

Ahora, suponga que un automóvil llega hasta el final del camino (se dispara la transición t_{101}), el agente percibe una señal de salida del sistema $\varphi M_k = [1 \ \dots \ 1 \ 1]^T$ en el tiempo τ_m . Además suponga que el agente percibe una señal de salida del sistema $\varphi M_{k+1} = [1 \ \dots \ 0 \ 1]^T$ en el tiempo \mathbb{F}_m . El agente distingue que se disparó la transición t_{102} y ésta se dispara en el modelo del observador extendido (así como la transición extra de salida del lugar p_{404}). Como $\mathbb{F}_m < H^-(t_{102}) + \tau_m$, el modelo del observador extendido

etiqueta la transición t_{102} como falta . El agente tiene la creencia de falta relacionada a la transición t_{102} porque ésta fue disparada en el tiempo mínimo especificado.

5.3. Conclusiones

En este capítulo se presentó el segundo enfoque propuesto en este trabajo y consistió en diseñar una función de revisión de creencias compuesto por un sistema dinámico llamado *observador extendido*. El modelo del observador extendido está enfocado a generar creencias acerca del estado y de las posibles faltas del sistema simultáneamente. El modelo del observador extendido es una copia del modelo en *RdPIVT* del sistema, con transiciones extras en cada lugar. Esta estructura adicional, permite al agente eliminar las creencias inconsistentes con respecto al estado del sistema. El mecanismo para generar creencias del estado y de las faltas del sistema, se representa mediante las reglas de disparo y el etiquetado de las transiciones del observador extendido, respectivamente. Las reglas de disparo se proponen mediante problemas de programación lineal para determinar el vector de disparo de las transiciones del modelo del observador extendido. El etiquetado de las transiciones, se realiza mediante funciones que permiten identificar las faltas en las transiciones en relación al tiempo en que se disparan.

Además se demostró que el modelo del observador extendido cumple con los principales axiomas que debe satisfacer la función de revisión de creencias: el agente nunca debe contradecir lo que realmente conoce del sistema y que conforme el agente está activo percibiendo y actuando dentro del ambiente, la revisión de creencias debe proporcionar una información cada vez más exacta del ambiente. Por último, se presentó un algoritmo en tiempo polinomial para diseñar la función de revisión de creencias.

Capítulo 6

Generador de creencias en RdPI no evento detectable

Resumen. Este capítulo presenta un esquema de generación de creencias para agentes que interactúan en ambientes de sistemas de eventos discretos, cuyo modelo es una *RdPI* no evento detectable.

En general, cuando el modelo en *RdPI* del ambiente del agente no tiene la propiedad de evento detectable, el disparo de una transición se confunde con otra transición, y el disparo de ellas genera la misma señal de salida. Agentes que interactúan con este tipo de sistemas, conforme está activo percibiendo, las creencias del estado del sistema no podrían proporcionar una información exacta del ambiente.

Para este tipo de modelos, se propone utilizar métodos de programación lineal para encontrar una solución al vector de disparo de transiciones del modelo del observador. El marcado alcanzado no contradice la información observada de salida del sistema y es congruente con las leyes conservativas de marcado *LCM*. En la metodología de modelado que se presenta en este trabajo (sección 3.1.2), una variable de estado se representa por un *P*-semiflujo, binario e inicialmente marcado, así las *LCM* representan la cantidad de recursos, piezas de trabajo, mensajes, etc. que se deben preservar en un *P*-semiflujo. Los métodos de programación lineal usados tienen la característica de que el vector de disparo que se encuentra tiene preferencia en disparar las transiciones del modelo del sistema con la finalidad de conservar lo más posible la dinámica del modelo del sistema.

A continuación se define el sistema dinámico que representa el mecanismo para la generación de creencias del estado del sistema, cuyo modelo del ambiente del agente en *RdPI* es no evento detectable.

Definición 6.1 Sea $N = (Q, M_0)$ un sistema modelado en *RdPI* viva, binaria, acotada y no evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$, $|P| = n$, $|T| = m$. El modelo diagnosticador $S = (Q_S, M_0)$ de (Q, M_0) es la *RdPI* donde:

1. $Q_S = (P_S, T_S, I_S, O_S, \Sigma, Id, Id)$, donde $P_S = P$, $T_S = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$, la ecuación de estados es:

$$\begin{aligned} \check{M}_{k+1} &= \check{M}_k + \begin{bmatrix} C & -I & I \end{bmatrix} v_k \\ \check{y}_k &= Id(\check{M}_k) \end{aligned}$$

Id es la función identidad e *I* es la matriz de identidad. I_S y O_S son las funciones de entrada y salida de la *RdPI* del modelo del sistema

2. El marcado inicial \check{M}_0 de (Q_S, \check{M}_0) es el vector que se obtiene del siguiente problema de programación lineal.

$$PPL_3 = \begin{cases} \text{mín } \check{M}_0(j) \geq 0 \\ \text{s.a.} \\ \sum_{\varphi(i,j) \neq 0} \check{M}_0(j) = \varphi M_0(i) \\ \Gamma \check{M}_0 = K \end{cases}$$

3. Las reglas de disparo de (Q_S, \check{M}_0) es el vector de disparo que resulta del siguiente algoritmo. Sea Γ la matriz $\Gamma[i, j] = \gamma_j^i$ que representa el conjunto de las *LCM* y K es el vector $K(i) = k_i$

Algoritmo 6.1 Cálculo del vector de disparo y marcado actual

Entradas: C - Matriz de incidencia del modelo del sistema (Q, M_0) ,

t_j - Transición que se dispara en el modelo del sistema (Q, M_0)

M_{act} - marcado actual del modelo del diagnosticador

Salidas:

M_k marcado actual del modelo del diagnosticador

v_k el vector de disparo

Si $(M_{act} + C(t_j) \geq 0)$ entonces

$$\Omega = \begin{cases} \sum v_k(i) = 1 \\ \forall i \in T, \text{ tal que } \varphi C(o,i) = \varphi C(o,j) \\ v_k(i) \geq 0 \\ \forall i \in T, \text{ tal que } \varphi C(o,i) \neq \bar{0} \\ v_k(i) \leq 1 \\ \forall i \in T, \text{ tal que } \varphi C(o,i) \neq \bar{0} \\ \Delta = 0 \end{cases}$$

de lo contrario

$$\Delta = \begin{cases} v_k(i) = 0 \\ \varphi C(o,i) \neq \bar{0} \\ \Omega = 0 \end{cases}$$

$$PPL_4 = \left\{ \begin{array}{l} \text{mín } \sum v_k \\ \text{s.a.} \\ \check{M}_k = \check{M}_{act} + C \sum v_k \\ \check{M}_k(l) \geq 0 \\ \sum \check{M}_k(l) = \varphi M_k(m) \\ \text{tal que } \varphi(m,l) \neq \bar{0} \\ \Gamma M_k = K \\ v_k(i) \geq 0 \\ v_k(i) \leq 1 \\ v_k(i) \geq 0 \\ i > |T| \\ v_k(i) \leq 1 \\ i > |T| \\ v_k(i) \geq 0 \\ \varphi C(o,i) = 0 \\ v_k(i) \leq 1 \\ \varphi C(o,i) = 0 \\ \Omega \\ \Delta \end{array} \right.$$

La definición del modelo del diagnosticador (Q_S, \check{M}_0) establece que el marcado inicial \check{M}_0 de (Q_S, \check{M}_0) se obtiene mediante el problema de programación lineal PPL_3 . El marcado inicial distribuye marcas en el modelo del diagnosticador, de tal forma que se conserva

el marcado en cada P -semiflujo de (Q_S, \check{M}_0) y la salida del diagnosticador es igual al del modelo del sistema ($\varphi\check{M}_0 = \varphi M_0$). Si los lugares de un P -semiflujo no emiten una señal de salida (no hay marca en un lugar medible), entonces marca cualesquiera de los lugares no medibles.

Si la transición t_j se dispara en el modelo del sistema ($M_k \xrightarrow{t_j} M_{k+1}$), entonces el algoritmo 6.1 encuentra un vector de disparo en las siguientes situaciones.

- t_j está habilitada en el modelo del diagnosticador. Se elige una transición de entre aquellas que se confunden con t_j (incluso ella misma). Si no hay transiciones habilitadas que se confunden con t_j , entonces compensa el disparo de t_j a través del disparo de las transiciones extras.
- t_j no está habilitada en el modelo del diagnosticador. Ocurren dos casos. En el primero se calcula una secuencia disparable de transiciones no evento detectables que habilitan a una transición que se confunde con t_j . Entonces se dispara dicha secuencia junto con la transición que se confunde con t_j . En el segundo caso, no existe dicha secuencia de transiciones no evento detectables y entonces el disparo de t_j se compensa disparando transiciones extras.

El algoritmo garantiza en un tiempo polinomial, el cálculo de un vector de disparo cuyo marcado alcanzado preserva las LCM y la salida del modelo del diagnosticador es igual a la salida del sistema ($\varphi\check{M}_k = \varphi M_k$).

A continuación se demuestra que los agentes cuya función de revisión de creencias se implementa con modelos tipo observador descrito anteriormente nunca contradicen las observaciones que se tienen del sistema.

Teorema 6.1 *Sea $N = (Q, M_0)$ un sistema modelado en RdPI binaria, viva, acotada y no evento detectable del sistema Υ , donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$ y $S = (Q_S, \check{M}_0)$ el observador de (Q, M_0) , entonces la siguiente sentencia se cumple:*

- $\forall k, M_k \in R(Q, M_0), \check{M}_k \in R(Q_S, \check{M}_0), \varphi(M_k) = \varphi(\check{M}_k)$

Demostración. Se demuestra trivialmente por definición, el marcado inicial del observador \check{M}_0 se obtiene de tal forma que $\varphi(M_0) = \varphi(\check{M}_0)$. Además, suponga que t_j se dispara en el modelo del sistema $M_{k-1} \xrightarrow{t_j} M_k$, y existen los siguientes casos.

1. $\check{M}_k + C(t_i) \geq 0$. La transición t_i está habilitada en (Q_S, \check{M}_0) , $t_i \in T$, tal que $\varphi C(\bullet, t_i) = \varphi C(\bullet, t_j)$ se dispara t_i , entonces $\check{M}_{k-1} \xrightarrow{t_i} \check{M}_k$, y $\varphi(M_k) = \varphi(\check{M}_k)$.
2. $\check{M}_k + C(t_i) < 0$. La transición t_i no está habilitada en (Q_S, \check{M}_0) , $t_i \in T$, tal que $\varphi C(\bullet, t_i) = \varphi C(\bullet, t_j)$ se obtiene un vector de disparo $\vec{\sigma}$ bajo la restricción de encontrar transiciones extras de entrada y salida tal que las LCM se cumplan en el modelo del diagnosticador y que $\check{M}_{k-1} \xrightarrow{\vec{\sigma}} \check{M}_k$, $\varphi(M_k) = \varphi(\check{M}_k)$.

Entonces para cualquier marcado \check{M}_k de (Q_S, \check{M}_0) , $\varphi(M_k) = \varphi(\check{M}_k)$.

6.1. Algoritmo para diseñar la función de revisión de creencias.

En esta sección se presenta un algoritmo para obtener el diseño de la función de revisión de creencias el cual está compuesta por el modelo del observador dedicado a generar creencias del estado del sistema y de las faltas del sistema.

Algoritmo 6.2 Diseño de la función de revisión de creencias

Entradas: El sistema modelado en $RdPI$ que representa el ambiente del agente A .

Salidas: La función de revisión de creencias

1. Construir la red del observador Q_O . Este es una copia del modelo en $RdPI$ del ambiente del agente donde transiciones extras son incluidas $T_O = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$. Cada lugar tiene una transición extra de entrada y salida, donde las nuevas transiciones de salida no tienen lugares de salida y las transiciones extras de entrada no tienen lugares de entrada
2. Calcular el vector de disparo v_k con el algoritmo 6.1.
3. La función de revisión de creencias conformada por la red Q_O, v_k es la salida del algoritmo.

6.2. Ejemplo

Considere el modelo en $RdPI (Q, M_0)$ de un SED de la figura 6.1, donde el marcado inicial del sistema se indica en la misma figura.

En este modelo, el alfabeto de entrada es $\Sigma = \{\}$ porque no hay señales de entrada en el sistema. La función λ es $\forall t_i, \lambda(t_i) = \varepsilon$. Como los lugares $p_1, p_2, p_4, p_5, p_7, p_8$ y p_{10} representan estados que tienen asociado señales de salida y el resto de los lugares no tienen señales de salida, φ es la matriz de 3×10 .

$$\varphi = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Observe que en la matriz φC existen columnas iguales, por lo que el disparo de una transición en (Q, M_0) , se confunde con otra. Por ejemplo el disparo de t_2 , se confunde con las transiciones t_6, t_9 , ya que $\varphi C(\bullet, 2) = \varphi C(\bullet, 6) = \varphi C(\bullet, 9)$, por lo tanto el modelo en $RdPI$ es no evento detectable.

$$\varphi C = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Suponga que existe un agente encargado de generar creencias del estado del sistema. Entonces, el modelo del observador se muestra en la figura 6.2 y es una copia del modelo del sistema con transiciones extras de entrada y salida. El marcado inicial del modelo del observador se establece de tal forma que existe una marca en cada P-semiflujo y que las señales de salida del observador sea igual que las señales del sistema, $\varphi\check{M}_0 = \varphi M_0$ donde

$$\varphi(M_0) = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

entonces

$$\check{M}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Suponga que se dispara la transición t_6 en el modelo del sistema, entonces $M_0 \xrightarrow{t_6} M_1$. El agente percibe la señal de salida del sistema

$$\varphi M_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

el agente calcula

$$\varphi M_1 - \varphi M_0 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

para determinar qué transición se disparó en el modelo del sistema. Como $\varphi M_1 - \varphi M_0 = \varphi C(\bullet, 2) = \varphi C(\bullet, 6) = \varphi C(\bullet, 9)$ el agente no distingue qué transición se disparó en el modelo del sistema. El agente elige disparar t_2 por lo tanto $\check{M}_0 \xrightarrow{t_2} \check{M}_1$, $\varphi\check{M}_1 = \varphi M_1$. Ahora suponga que se dispara la transición t_7 en el modelo del sistema, $M_1 \xrightarrow{t_7} M_2$, como el modelo del sistema en *RdPI* es no evento detectable, el agente confunde el disparo de la transición t_7 con t_3 y elige disparar t_3 en el modelo del observador $\check{M}_1 \xrightarrow{t_3} \check{M}_2$ y $\varphi\check{M}_2 = \varphi M_2$. Por último, suponga que se dispara la transición t_5 en el modelo del sistema, el agente percibe la señal de salida del sistema

$$\varphi M_3 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

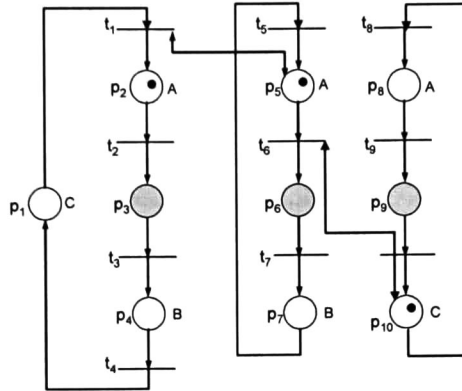


Figura 6.1: Modelo en RdPI no evento detectable.

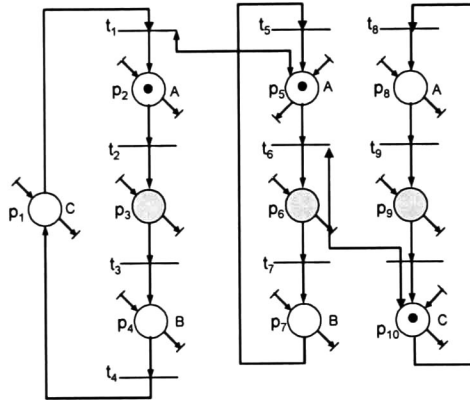


Figura 6.2: Modelo observador en RdPI.

y calcula

$$\varphi M_3 - \varphi M_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Se distingue que se disparó t_5 porque la columna $\varphi C(\bullet, 5) = \varphi M_3 - \varphi M_2$. Por la regla de disparo del observador (definición 4.1), el agente no dispara ninguna transición porque t_5 no está habilitada y $\check{M}_2(p_5) = M^{UB}(p_5)$.

Por lo tanto $\varphi \check{M}_2 = [1 \ 1 \ 1]^T \neq \varphi(M_3)$. Esto significa que la secuencia de disparo de transiciones en el observador no fue la correcta de forma que llegó a un marcado que la creencia que representa contradice con lo que observa.

En este caso el observador calcular un vector de disparo para que la salida que genere sea igual que la salida del sistema. Dispara la transición extra de entrada de p_5 y la transición extra de salida de p_4 para que $\varphi \check{M}_3 = \varphi(M_3)$.

6.3. Conclusiones

En este capítulo se presentó el modelo de un observador de un *SED* no evento detectable para representar la función de generación de creencias del estado del sistema. El observador es implementado con problemas de programación lineal. La ventaja de este tipo de algoritmos lineales es que encuentra un vector de disparo en un tiempo polinomial. El marcado que se alcanza preserva las *LCM* y la salida del modelo del diagnosticador es igual a la salida del sistema ($\varphi\check{M}_k = \varphi M_k$). Finalmente se demostró que los agentes cuya función de revisión de creencias se implementa con este tipo de observadores nunca contradicen las observaciones que se tienen del sistema.

Capítulo 7

Postulados AGM para la revisión de creencias en RdPI

Resumen. Este capítulo define los conceptos relacionados a la revisión de creencias desde un enfoque algebraico: creencias, base de conocimiento, operador de revisión de creencias y las operaciones de cambios en las creencias. Además, se demuestra que los enfoques propuestos en este trabajo que representan la función de revisión de creencias con redes de Petri Interpretadas cumple cada uno de los axiomas propuestos por Alchourron, Makinson and Gärdenfors.

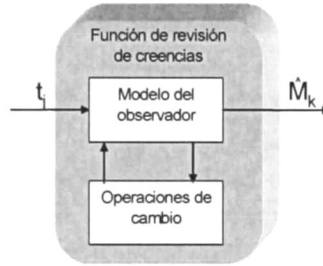


Figura 7.1: Función de revisión de creencias consistente

El mecanismo de la función de revisión de creencias representado por el observador extendido se resume en la figura 7.1. Este esquema está compuesto por dos partes; la primera es un modelo observador extendido y la segunda es el operador de cambios sobre el modelo del observador extendido para generar nuevas creencias. Como ya se ha mencionado, la función de revisión de creencias se considera como un sistema dinámico enfocado a generar nuevas creencias a partir de creencias actuales y de la información del ambiente que el agente percibe. Además, la revisión de creencias está enfocada en el caso cuando la información del ambiente que percibe el agente causa un cambio en las creencias actuales, generando creencias inconsistentes de tal forma que el resultado son nuevas creencias consistentes. Para obtener este resultado, la revisión de creencias elimina creencias inconsistentes con un cambio mínimo, esto es, retiene lo más posible las creencias antiguas.

En este trabajo, las creencias inconsistentes del estado del sistema significan que un lugar excede su cota máxima, así, la función de revisión de creencias utiliza las operaciones de cambio para eliminar las marcas que exceden en tal lugar.

Definición 7.1 Sea $N_O = (Q_O, \hat{M}_0)$ el modelo observador extendido en $RdPIVT$ y sea P_O el conjunto de lugares de N_O . La creencia del estado del sistema de un agente $\hat{M}_k(p_i)$ está definida como:

$$\hat{M}_k(p_i) = \begin{cases} \text{consistente} & \text{si } \hat{M}^{LB}(p_i) \leq \hat{M}_k(p_i) \leq \hat{M}^{UB}(p_i) \\ \text{inconsistente} & \text{si } \hat{M}_k(p_i) > \hat{M}^{UB}(p_i) \end{cases} \quad (7.1)$$

Alchourron, Makinson y Gärdenfors en [2] desarrollaron un conjunto de postulados, llamados *AGM*, que garantizan que el operador de revisión genera cambios consistentes en el conjunto de creencias. Las creencias del agente son representadas por un conjunto de sentencias cercanas al formalismo de la lógica estándar. Además, definen tres tipos de cambios en las creencias: expansión (+), revisión (*) y contracción (-).

Para utilizar la notación de los postulados *AGM*, consideramos la base de conocimiento representada por un conjunto de sentencias algebraicas como sigue:

Definición 7.2 Sea $N_O = (Q_O, \hat{M}_0)$ el modelo observador en $RdPIVT$ y sean P_O y T_O el conjunto de lugares y transiciones de N_O . La base de conocimiento K se llama el

conjunto de creencias tal que

$$K = \left\{ \hat{M}_0(p_i), \vec{t}_j, \hat{T}_0^L(t_j) \right\} \quad (7.2)$$

donde $p_i \in P_O$, $t_j \in T_O$ y \vec{t}_j es el vector de disparo de N_O

Esta definición establece que el conjunto de creencias K son las creencias iniciales del agente. Esto significa que el marcado inicial del modelo del observador extendido en $RdPIVT$ representa el conocimiento inicial que tiene el agente acerca del estado del sistema. Además, la información del ambiente que percibe el agente es parte del conjunto de creencias, es decir, la transición que produce un cambio de marcado en el modelo del sistema. Asimismo, el etiquetado inicial de las transiciones del modelo del observador representan las creencias iniciales con respecto a las faltas del sistema.

Como la información que percibe el agente provoca cambios en las creencias actuales del estado del sistema, se definen tres tipos de operaciones sobre las creencias.

1. **Expansión:** Si la información del ambiente t_j la percibe el agente y t_j está habilitada en el marcado \hat{M}_k , entonces t_j se dispara generando una nueva creencia \hat{M}_{k+1} . La operación de expansión se denota como $K^{+\phi}$.
2. **Revisión:** Si la nueva creencia \hat{M}_{k+1} es inconsistente, para mantener consistencia, algunas marcas de la actual creencia son eliminadas. El resultado de revisar las creencias se denota como $K^{\pm\phi}$
3. **Contracción :** Para mantener consistente las creencias del agente, algunas marcas en \hat{M}_{k+1} son eliminadas, usando la transición extra de salida. Esta operación se denota como $K^{-\phi}$.

Comentario 7.1 1. ϕ expresa la información que el agente percibe de su ambiente (t_j):

$$\phi = \left\{ t_j \mid \varphi C(\bullet, j) = \varphi(M_{k+1}) - \varphi(M_k) \text{ y } M_k \xrightarrow{t_j} M_{k+1} \right.$$

así, $K^{+\phi}$ denota que ϕ se adiciona al conjunto de creencias K generando una nueva creencia, $\hat{M}_k \xrightarrow{t_j} \hat{M}_{k+1}$.

2. Una vez que la nueva creencia \hat{M}_{k+1} se genera, ésta se incorpora al conjunto de creencias y se representa por $K^{+\phi}$ Además, si la nueva creencia \hat{M}_{k+1} es inconsistente, se usa la operación de contracción y la creencia revisada es nuevamente representada en el conjunto de creencias. A esta acción se denota por $K^{\pm\phi}$

Las operaciones de cambio de creencias son implementadas por la siguiente regla de disparo en $RdPI$

Definición 7.3 Sea $N_S = (Q, M_0)$ un sistema modelado en $RdPIVT$ viva, binaria, acotada y evento detectable del sistema Υ y sea $N_O = (Q_O, \hat{M}_0)$ el modelo observador extendido en $RdPIVT$. Entonces, las operaciones de cambio de creencias pueden ser implementadas, definiendo el vector de disparo de transiciones v_k de N_O , usando la regla de disparo que se obtiene del algoritmo 5.1.

La definición anterior establece que cuando el marcado de un lugar excede la cota máxima de ese lugar, esas marcas son removidas por el disparo de las transiciones extras de salida. Esto significa que se elimina la inconsistencia de las creencias del agente acerca del estado del sistema.

El observador extendido utilizado para representar el mecanismo para generar nuevas creencias consistentes bajo las operaciones de expansión, revisión y contracción, cumple con los postulados que proponen Alchourron, Makinson y Gärdenfors.

Los postulados tienen el siguiente significado.

(K*1) es la familia de postulados de la lógica proposicional, los cuales, son cerrados bajo la consecuencia lógica.

(K*2) establece que la nueva información percibida debe estar incluida en K .

$$\phi \in K * \phi$$

(K*3) expresa que la expansión representa una “cota máxima” cuando se incorpora nuevas creencias.

$$K * \phi \subseteq K + \phi$$

(K*4) establece que, cuando la negación de la información percibida no está en K , entonces, la operación de revisión se reduce sólo a efectuar la operación de expansión.

$$\text{Si } \neg\phi \notin K, \text{ entonces } K + \phi \subseteq K * \phi$$

(K*5) la única situación en la que la revisión permite tener inconsistencias en el conjunto de creencias ocurre cuando el agente acepta información contradictoria.

$$K * \phi = K_{\perp} \text{ si y solo si } \vdash \neg\phi$$

(K*6) es una expresión del Principio de Irrelevancia de Sintaxis.

$$\text{Si } \vdash \phi \leftrightarrow \psi, \text{ entonces } K * \phi = K * \psi$$

(K*7) y (K*8) La idea es que, si $K * \phi$ es una revisión de K y $K * \phi$ cambia por otra sentencia ψ , tal cambio debe ser hecho por la expansión de $K * \phi$ cuando sea posible. $K * \phi \wedge \psi$, debe ser el mismo como la expansión de $K * \phi$ por ψ siempre y cuando ψ no contradiga las creencias en $K * \phi$.

$$K * \phi \wedge \psi \subseteq (K * \phi) + \psi$$

$$\text{Si } \neg\psi \notin K * \phi, \text{ entonces } (K * \phi) + \psi \subseteq K * \phi \wedge \psi$$

7.1. Postulados de revisión para modelos en RdPI evento detectables

En esta sección se demuestra que el observador extendido cumple con los postulados de revisión *AGM* desde un enfoque algebraico. Para la demostración supongamos que el vector de disparo de las transiciones extras de salida del modelo del observador extendido es $\beta_k [j] = v_k [k]$ tal que $m < k \leq m + n$ donde $|P| = n$, $|T| = m$.

- El primer postulado requiere que la salida de la función de revisión de creencias sea un conjunto de creencias.

Teorema 7.1 *Para cualquier señal de sistema ϕ y un conjunto de creencias K , $K^{\pm\phi}$ es un conjunto de creencias.*

Demostración. Procedemos por inducción en la longitud de la secuencia de disparo σ .

1. $|\sigma| = 1$, Por definición 2, $\hat{M}_0 = M_0 + \Delta\hat{M}_0$. Suponga que la señal de salida del sistema φM_1 se percibe y que $M_0 \xrightarrow{t_j} M_1$ ocurre en N_S . Como el modelo del observador extendido tiene más marcas que el modelo del sistema y por la propiedad evento detectabilidad de N_S se distingue el disparo de la transición del sistema ($\varphi M_1 - \varphi M_0 = \varphi C(\bullet, j)$), entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. De acuerdo a la definición de N_O , $\hat{M}_0 = M_0 + \Delta\hat{M}_0$ y por el lema de monotonicidad [12], $\hat{M}_1 = M_1 + \Delta\hat{M}_1$. Entonces para cualquier lugar p_i de N_S , los siguientes casos son posible:

- a) Si $\left[\hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \right] \leq M^{UB}(p_i)$ entonces $\hat{M}_1(p_i)$ es una creencia consistente.
- b) Si $\left[\hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \right] > M^{UB}(p_i)$ entonces $\hat{M}_1(p_i)$ es una creencia inconsistente y la transición de salida β_1 elimina una marca del lugar p_i . Entonces de la ecuación de estados

$$\hat{M}_1(p_i) = \left[\hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \right] - \beta_1(i)$$

tenemos que $\hat{M}_1(p_i) = M^{UB}(p_i)$, entonces por definición 7.1, $\hat{M}_1(p_i)$ es una creencia consistente.

Observe que $\forall p_i \in P_O$, $M^{LB}(p_i) \leq \hat{M}_1(p_i) \leq M^{UB}(p_i)$ entonces $\hat{M}_1(p_i)$ es una creencia consistente y $\hat{M}_1(p_i) \in K^{\pm\phi}$. Por lo tanto, $K^{\pm\phi}$ es un conjunto de creencias consistente.

2. Ahora suponga que $|\sigma| = k$, y $\hat{M}_1(p_i) \in K^{\pm\phi}$ es una creencia consistente. Siguiendo un procedimiento similar al paso 1 se demuestra que $\forall p_i \in P_O$, $M^{LB}(p_i) \leq \hat{M}_{k+1}(p_i) \leq M^{UB}(p_i)$. Por lo tanto, usando la definición 7.1 tenemos que $\hat{M}_{k+1}(p_i)$ es una creencia consistente. Por lo tanto $\hat{M}_{k+1}(p_i) \in K^{\pm\phi}$ y $K^{\pm\phi}$ es un conjunto de creencias consistente.

■

- El segundo postulado garantiza que la nueva creencia generada por la información percibida del ambiente del agente sea incluida en $K^{\pm\phi}$

Teorema 7.2 $\phi \in K^{\pm\phi}$

Demostración. Similar a la demostración del teorema 7.1. ■

- El siguiente postulado establece que la aplicación normal de la función de revisión de creencias es cuando la información percibida del ambiente del agente genera creencias inconsistentes.

Teorema 7.3 $K^{\pm\phi} \subseteq K^{+\phi}$

Demostración. Procedemos por inducción en la longitud de la secuencia de disparo σ .

1. $|\sigma| = 1$, entonces \hat{M}_0 se establece como en la definición 2. Si p_i es un lugar medible entonces $\hat{M}_0(p_i) = M_0(p_i)$. Si p_i es un lugar no medible entonces $\hat{M}_0(p_i) = M^{UB}(p_i)$, así $\hat{M}_0 = M_0 + \Delta\hat{M}_0$. Suponga que la señal de salida del sistema φM_1 se percibe y que $M_0 \xrightarrow{t_j} M_1$ ocurre en N_S . Como el modelo del observador extendido tiene mas marcas que el modelo del sistema y por la propiedad de evento detectabilidad de N_S se distingue el disparo de la transición del sistema ($\varphi M_1 - \varphi M_0 = \varphi C(\bullet, j)$), entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. De acuerdo a la definición de N_O , $\hat{M}_0 = M_0 + \Delta\hat{M}_0$ y por el lema de monoticidad [12] $\hat{M}_1 = M_1 + \Delta\hat{M}_0$.

Sea p_i un lugar de N_S . Entonces, de la ecuación de estados

$$M_1(p_i) = M_0(p_i) + C(p_i, \cdot) \vec{t}_j \quad (7.3)$$

y

$$\begin{aligned} \hat{M}_1(p_i) &= \hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \\ &= (M_0(p_i) + \Delta\hat{M}_0(p_i)) + C(p_i, \cdot) \vec{t}_j \\ &= M_1(p_i) + \Delta\hat{M}_0(p_i) \end{aligned} \quad (7.4)$$

$$\hat{M}_1(p_i) \in K^{+\phi}$$

Suponga que $\hat{M}'_1(p_i)$ es la creencia revisada de $\hat{M}_1(p_i)$, esto es, $\hat{M}'_1(p_i) \in K^{\pm\phi}$. Esto significa que por el teorema 7.1, $\hat{M}'_1(p_i)$ es una creencia consistente. Entonces existen dos casos:

- a) si p_i es un lugar medible, entonces por definición, $\Delta\hat{M}_0(p_i) = 0$. Así $\hat{M}_1(p_i) = M_1(p_i)$ y por definición 7.1, $\hat{M}_1(p_i)$ es una creencia consistente. Por lo tanto $\hat{M}'_1(p_i) = \hat{M}_1(p_i)$
- b) si p_i es un lugar no medible, entonces por definición, $\Delta\hat{M}_0(p_i) = M^{UB}(p_i)$, así $\hat{M}_1(p_i) > M^{UB}(p_i)$ y por definición 7.1, $\hat{M}_1(p_i)$ es una creencia inconsistente. Como p_i es un lugar no medible y $\hat{M}_1(p_i)$ es una creencia inconsistente, entonces la transición de salida β_1 elimina una marca de p_i . Entonces de la ecuación de estados

$$\begin{aligned}\hat{M}'_1(p_i) &= \left[\hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \right] - \beta_1(i) \\ &= \left[\left(M_0(p_i) + \Delta\hat{M}_0(p_i) \right) + C(p_i, \cdot) \vec{t}_j \right] - \beta_1(i) \\ &= \left[M_1(p_i) + \Delta\hat{M}_0(p_i) \right] - \beta_1(i)\end{aligned}$$

así

$$\hat{M}'_1(p_i) = \hat{M}_1(p_i) - \beta_1(i)$$

Tenemos que $\hat{M}'_1(p_i) = M^{UB}(p_i)$, entonces por definición 7.1, $\hat{M}'_1(p_i)$ es una creencia consistente.

Como $\hat{M}_1(p_i) > M^{UB}(p_i)$ y $\hat{M}'_1 = \hat{M}_1(p_i) - \beta_1(i)$ entonces $\hat{M}'_1(p_i) < \hat{M}_1(p_i)$.

Concluimos que $\forall p_i \in P_O$, $\hat{M}'_1(p_i) \leq \hat{M}_1(p_i)$, por lo tanto, si $\hat{M}'_1(p_i) \in K^{\pm\phi}$ entonces $\hat{M}'_1(p_i) \in K^{+\phi}$, así $K^{\pm\phi} \subseteq K^{+\phi}$.

2. Ahora suponga que $|\sigma| = k$, entonces $\forall p_i \in P_O$, $\hat{M}'_k(p_i) \leq \hat{M}_k(p_i)$. Siguiendo un procedimiento similar al paso 1 esto se demuestra que $\forall p_i \in P_O$, $\hat{M}_{k+1}(p_i) > M^{UB}(p_i)$ y $\hat{M}'_{k+1} = \hat{M}_{k+1}(p_i) - \beta_{k+1}(i)$ entonces $\hat{M}'_{k+1}(p_i) \leq \hat{M}_{k+1}(p_i)$. Si $\hat{M}'_{k+1}(p_i) \in K^{\pm\phi}$ entonces $\hat{M}'_{k+1}(p_i) \in K^{+\phi}$, Así $K^{\pm\phi} \subseteq K^{+\phi}$

■

- El caso contrario del postulado anterior es cuando la creencia generada por la información percibida del ambiente es una creencia consistente. Así la función de revisión de creencias es identificada con la operación de expansión.

Teorema 7.4 Si $\neg\phi \notin K$, $K^{+\phi} \subseteq K^{\pm\phi}$

Demostración. Procedemos por inducción en la longitud de la secuencia de disparo σ .

1. $|\sigma| = 1$, entonces \hat{M}_0 se establece como en la definición 2. Si p_i es un lugar medible entonces $\hat{M}_0(p_i) = M_0(p_i)$. Si p_i es un lugar no medible entonces $\hat{M}_0(p_i) = M^{UB}(p_i)$, así $\hat{M}_0 = M_0 + \Delta\hat{M}_0$. Suponga que la señal de salida del sistema φM_1 se percibe y que $M_0 \xrightarrow{t_j} M_1$ ocurre en N_S . Como el modelo del observador tiene más marcas

7.1. POSTULADOS DE REVISIÓN PARA MODELOS EN RDPI EVENTO DETECTABLES 97

que el modelo del sistema y por la propiedad evento detectable de N_S se distingue el disparo de la transición del sistema ($\varphi M_1 - \varphi M_0 = \varphi C(\bullet, j)$), entonces $\hat{M}_0 \xrightarrow{t_j} \hat{M}_1$. De acuerdo a la definición de N_O , $\hat{M}_0 = M_0 + \Delta \hat{M}_0$ y por el lema de monotonicidad [12], $\hat{M}_1 = M_1 + \Delta \hat{M}_0$.

Sea p_i un lugar de N_S . Entonces, de la ecuación de estados

$$M_1(p_i) = M_0(p_i) + C(p_i, \cdot) \vec{t}_j \quad (7.5)$$

y

$$\begin{aligned} \hat{M}_1(p_i) &= \hat{M}_0(p_i) + C(p_i, \cdot) \vec{t}_j \\ &= (M_0(p_i) + \Delta \hat{M}_0(p_i)) + C(p_i, \cdot) \vec{t}_j \\ &= M_1(p_i) + \Delta \hat{M}_0(p_i) \end{aligned} \quad (7.6)$$

$$\hat{M}_1(p_i) \in K^{+\phi}$$

Suponga que $\hat{M}_1(p_i)$ es una creencia consistente. Esto significa que p_i es un lugar medible.

Suponga que $\hat{M}'_1(p_i)$ es la creencia revisada de $\hat{M}_1(p_i)$, esto es, $\hat{M}'_1(p_i) \in K^{\pm\phi}$. Como p_i es un lugar medible y por definición, $\Delta \hat{M}_0(p_i) = 0$, entonces $\hat{M}_1(p_i) = M_1(p_i)$. Tenemos que $\hat{M}_1(p_i) = M^{UB}(p_i)$, entonces por definición 7.1, $\hat{M}_1(p_i)$ es una creencia consistencia. Por lo tanto $\hat{M}_1(p_i) = \hat{M}'_1(p_i)$.

Como $\hat{M}_1(p_i)$ es una creencia consistente, concluimos que $\forall p_i \in P^P$ donde P^P es el conjunto de lugares medible, si $\hat{M}_1(p_i) \in K^{+\phi}$ entonces $\hat{M}_1(p_i) \in K^{\pm\phi}$ así $K^{+\phi} \subseteq K^{\pm\phi}$.

2. Ahora supongamos que $|\sigma| = k$ entonces $\forall p_i \in P^P$ $\hat{M}_k(p_i) = M_k(p_i)$, entonces $K^{+\phi} \subseteq K^{\pm\phi}$. Siguiendo un procedimiento similar al paso 1 se demuestra que $\forall p_i \in P^P$, si $\hat{M}_{k+1}(p_i) \in K^{+\phi}$ entonces $\hat{M}_{k+1}(p_i) \in K^{\pm\phi}$ así $K^{+\phi} \subseteq K^{\pm\phi}$

■

- El propósito del quinto postulado es obtener un nuevo conjunto de creencias inconsistentes. La única situación en la que la operación de revisión produce un nuevo conjunto de creencias inconsistentes ocurre cuando es aceptado información lógicamente contradictoria.

Teorema 7.5 $K * \phi = K_{\perp}$ si y solo si $\vdash \neg\phi$

Demostración. Se demuestra trivialmente por la definición 2 y 7.1 porque no es posible tener $\hat{M}_k(p_i) = M^{UB}(p_i)$ y $\hat{M}_k(p_i) > M^{UB}(p_i)$ en N_O . ■

- El siguiente postulado establece que si diferente información del ambiente genera la misma creencia, entonces el resultado de revisar las creencias de ellas es la misma.

Teorema 7.6 Si $\vdash \phi \leftrightarrow \psi$, entonces $K^{\pm\phi} = K^{\pm\psi}$

Demostración. Suponga que $\lambda(t_j) = \lambda(t_i) = \alpha_1 \neq \varepsilon$ y que ambas transiciones están habilitadas en \hat{M}_k . Entonces los siguientes casos son posibles:

1. t_j se dispara alcanzando un nuevo marcado \hat{M}_{k+1} , esto es $\hat{M}_k \xrightarrow{t_j} \hat{M}_{k+1}$
2. t_i se dispara alcanzando un nuevo marcado \hat{M}'_{k+1} , esto es $\hat{M}_k \xrightarrow{t_i} \hat{M}'_{k+1}$

Ahora suponga que $\hat{M}''_{k+1} \neq \hat{M}_{k+1}$ y $\varphi\hat{M}''_{k+1} = \varphi\hat{M}_{k+1}$. Como $\varphi\hat{M}''_{k+1} - \varphi\hat{M}_k = \varphi\hat{M}_{k+1} - \varphi\hat{M}_k$ entonces por la propiedad evento detectable no puede distinguirse cuál transición fue disparada, entonces N_O no es evento detectable.

Como N_O se define como evento detectable, entonces $t_j = t_i$ y $\hat{M}''_{k+1} = \hat{M}_{k+1}$. Además, por el primer postulado 7.1, \hat{M}_{k+1} es una creencia consistente y es una creencia revisada, $\hat{M}'_{k+1} \in K^{\pm\phi}$. Por lo tanto $K^{\pm\phi} = K^{\pm\psi}$, ■

- Los dos últimos postulados establecen que el cambio mínimo de K incluye ambos ϕ y ψ , esto es, $K^{\pm\phi\wedge\psi}$ debería ser la misma como la expansión de $K^{\pm\phi}$ por ψ , siempre y cuando ψ no contradiga las creencias en $K^{\pm\phi}$.

Teorema 7.7 $K^{\pm\phi\wedge\psi} \subseteq (K^{\pm\phi})^{+\psi}$

Demostración. Procedemos por inducción en la longitud de la secuencia de disparo σ .

1. $|\sigma| = 2$, entonces \hat{M}_0 se establece como en la definición 2. Si p_i es un lugar medible entonces $\hat{M}_0(p_i) = M_0(p_i)$. Si p_i es un lugar no medible entonces $\hat{M}_0(p_i) = M^{UB}(p_i)$, así $\hat{M}_0 = M_0 + \Delta\hat{M}_0$. Suponga que la señal de salida del sistema $\varphi M'_0$ se percibe y que $M_0 \xrightarrow{t_i} M'_0$ ocurre en N_S . Como el modelo del observador extendido tiene más marcas que el modelo del sistema y por la propiedad evento detectable de N_S se distingue el disparo de la transición del sistema ($\varphi M_0 - \varphi M'_0 = \varphi C(\bullet, i)$), entonces $\hat{M}_0 \xrightarrow{t_i} \hat{M}'_0$. De acuerdo a la definición de N_O , $\hat{M}_0 = M_0 + \Delta\hat{M}_0$ y por el lema de monotonicidad [12], $\hat{M}'_0 = M'_0 + \Delta\hat{M}_0$.

Sea p_i el lugar de N_S . Entonces $M'_0(p_i) \in K + \phi$. Suponga que $M''_0(p_i)$ es la creencia revisada de $M'_0(p_i)$. Así $M''_0(p_i) \in K \pm \phi$. Esto significa que por el postulado 1, $K \pm \phi$ es un conjunto de creencias consistentes.

Ahora, asuma que otra señal de salida del sistema se percibe, esto es φM_1 se percibe y que $M'_0 \xrightarrow{t_j} M_1$ ocurre en N_S . Entonces $\hat{M}'_0 \xrightarrow{t_j} \hat{M}_1$ y $\hat{M}_1 = M_1 + \Delta\hat{M}_0$. Entonces

7.1. POSTULADOS DE REVISIÓN PARA MODELOS EN RDPI EVENTO DETECTABLES99

$\hat{M}_1(p_i) \in (K \pm \phi) + \psi$. Suponga que $\hat{M}'_1(p_i)$ es la creencia revisada de $\hat{M}_1(p_i)$, entonces $\hat{M}'_1(p_i) \in (K \pm \phi) \pm \psi$. Esto es, $\hat{M}'_1(p_i) \in K \pm \phi \wedge \psi$.

Entonces, demostramos que el conjunto de creencias revisadas de $\phi \wedge \psi$ es un subconjunto del conjunto por la expansión de $K \pm \phi$ por ψ . Entonces existen dos casos:

- a) Si p_i es un lugar no medible entonces por 7.3.1.a $\bar{M}'_1(p_i) = \bar{M}_1(p_i)$.
- b) Si p_i es un lugar no medible entonces por 7.3.1.b $\bar{M}'_1(p_i) < \bar{M}_1(p_i)$.

Así, $\forall p_i \in P_O$, $\hat{M}'_1(p_i) \leq \hat{M}_1(p_i)$. Como $\hat{M}'_1(p_i) \in K \pm \phi \wedge \psi$ y $\hat{M}_1(p_i) \in (K \pm \phi) + \psi$ concluimos que si $\hat{M}'_1(p_i) \in K \pm \phi \wedge \psi$ entonces $\hat{M}'_1(p_i) \in (K \pm \phi) + \psi$, así $K^{\pm\phi \wedge \psi} \subseteq (K^{\pm\phi})^{+\psi}$

■

Proposición 7.1 Si $\neg\psi \notin K^{\pm\phi}$, entonces $(K^{\pm\phi})^{+\psi} \subseteq K^{\pm\phi \wedge \psi}$

Demostración.

1. $|\sigma| = 2$, entonces \hat{M}_0 se establece como en la definición 2. Si p_i es un lugar no medible entonces $\hat{M}_0(p_i) = M_0(p_i)$. Si p_i es un lugar no medible entonces $\hat{M}_0(p_i) = M^{UB}(p_i)$, así $\hat{M}_0 = M_0 + \Delta\hat{M}_0$. Suponga que la señal de salida del sistema $\varphi M'_0$ se percibe y que $M_0 \xrightarrow{t_i} M'_0$ ocurre en N_S . Como el modelo del observador extendido tiene más marcas que el modelo del sistema y por la propiedad evento detectable de N_S se detecta el disparo de la transición t_i del sistema ($\varphi M'_0 - \varphi M_0 = \varphi C(\bullet, i)$), entonces $\hat{M}_0 \xrightarrow{t_i} \hat{M}'_0$. De acuerdo a la definición de N_O , $\hat{M}_0 = M_0 + \Delta\hat{M}_0$ y por el lema de monoticidad [12], $\hat{M}'_0 = M'_0 + \Delta\hat{M}_0$.

Sea p_i un lugar de N_S . Entonces $M'_0(p_i) \in K + \phi$. Suponga que $M''_0(p_i)$ es la creencia revisada de $M'_0(p_i)$. Así $M''_0(p_i) \in K \pm \phi$. Esto significa que por el postulado 1, $K \pm \phi$ es un conjunto de creencias consistente.

Ahora, suponga que otra señal de salida del sistema se percibe, esto es φM_1 se percibe y que $M'_0 \xrightarrow{t_j} M_1$ ocurre en N_S . Entonces $\hat{M}'_0 \xrightarrow{t_j} \hat{M}_1$ y $\hat{M}_1 = M_1 + \Delta\hat{M}_0$. Entonces $\hat{M}_1(p_i) \in (K \pm \phi) + \psi$. Suponga que $\hat{M}'_1(p_i)$ es la creencia revisada de $\hat{M}_1(p_i)$, entonces $\hat{M}'_1(p_i) \in (K \pm \phi) \pm \psi$. Esto es, $\hat{M}'_1(p_i) \in K \pm \phi \wedge \psi$.

Probamos que si la expansión de $K \pm \phi$ por ψ no es inconsistente, entonces $(K^{\pm\phi})^{+\psi}$ es un subconjunto de creencias revisadas de $\phi \wedge \psi$. Supongamos que p_i es un lugar medible. Esto significa que $\hat{M}_1(p_i)$ es una creencia consistente. Entonces, por el postulado 7.3.1.a $\bar{M}'_1(p_i) = \bar{M}_1(p_i)$.

Así, $\forall p_i \in P^P$, donde P^P es el conjunto de lugares medibles, $\hat{M}'_1(p_i) = \hat{M}_1(p_i)$. Como $\hat{M}_1(p_i) \in (K \pm \phi) + \psi$ y $\hat{M}'_1(p_i) \in K \pm \phi \wedge \psi$ concluimos que si $\hat{M}_1(p_i) \in (K \pm \phi) + \psi$ entonces $\hat{M}_1(p_i) \in K \pm \phi \wedge \psi$, así si $\neg\psi \notin K^{\pm\phi}$, entonces $(K^{\pm\phi})^{+\psi} \subseteq K^{\pm\phi \wedge \psi}$

2. Ahora suponga que $|\sigma| = k$ entonces $\forall p_i \in P^P$, $\bar{M}_k(p_i) = M_k(p_i)$, entonces $(K^{\pm\phi})^{+\psi} \subseteq K^{\pm\phi \wedge \psi}$. Siguiendo un procedimiento similar al paso 1 esto se demuestra que $\forall p_i \in P^P$ si $\hat{M}_1(p_i) \in (K \pm \phi) + \psi$ entonces $\hat{M}_1(p_i) \in K \pm \phi \wedge \psi$, así, si $\neg\psi \notin K^{\pm\phi}$, entonces $(K^{\pm\phi})^{+\psi} \subseteq K^{\pm\phi \wedge \psi}$

■

La demostración de los postulados de contracción es similar a la demostración de los postulados de revisión.

El modelo del observador-diagnosticador cumplen con los postulados *AGM*, su demostración es muy similar a la que se presentó para el modelo del observador extendido.

Las creencias correspondientes a las faltas del sistema no llegan a ser inconsistentes debido a que se detectan en función al tiempo y por la naturaleza en la que éstas se presentan.

7.2. Postulados de revisión para modelos en RdPI no evento detectables

El modelo del observador en *RdPI* no evento detectables cumplen con los postulados *AGM* y su demostración es diferente a la del observador extendido. Las definiciones de las creencias y las operaciones de cambio de creencias están en función a las señales de salidas del

Definición 7.4 Sea $S = (Q_S, \check{M}_0)$ el modelo observador en *RdPI* y sean P_S y T_S el conjunto de lugares y transiciones de S La base de conocimiento K se llama el conjunto de creencias tal que

$$K = \{\varphi \check{M}_0(p_i), \vec{t}_j\} \quad (7.7)$$

donde $p_i \in P_S$, $t_j \in T_S$ y \vec{t}_j es el vector de disparo de No

Esta definición establece que el conjunto de creencias K son las creencias iniciales del agente. Esto significa que la señal de salida del modelo del observador en *RdPI* representa el conocimiento inicial que tiene el agente acerca del estado del sistema. Además, la información del ambiente que percibe el agente es parte del conjunto de creencias, es decir, la transición que produce un cambio de marcado en el modelo del sistema.

Las operaciones sobre las creencias que se aplican cuando el agente percibe información de su ambiente, es similar a la del modelo del observador extendido.

1. **Expansión:** Si la información del ambiente t_j la percibe el agente y t_j está habilitada en el marcado \hat{M}_k , entonces t_j se dispara alcanzando un nuevo marcado \hat{M}_{k+1} y generando una nueva creencia $\varphi \hat{M}_{k+1}$. La operación de expansión se denota como $K^{+\phi}$.

7.2. POSTULADOS DE REVISIÓN PARA MODELOS EN RDPI NO EVENTO

2. **Revisión:** Si la nueva creencia $\varphi\hat{M}_{k+1}$ es inconsistente, para mantener consistencia, algunas marcas de la actual creencia son eliminadas. El resultado de revisar las creencias se denota como $K^{\pm\phi}$.
3. **Contracción :** Para mantener consistente las creencias del agente, algunas marcas en \hat{M}_{k+1} son eliminadas, usando la transición extra de salida, para generar una creencia consistente $\varphi\hat{M}_{k+1}$ Esta operación se denota como $K^{-\phi}$

Comentario 7.2 1. ϕ expresa la información que el agente percibe de su ambiente (t_j):

$$\phi = \left\{ t_j \mid \varphi C(\bullet, j) = \varphi(M_{k+1}) - \varphi(M_k) \text{ y } M_k \xrightarrow{t_j} M_{k+1} \right.$$

así, $K^{+\phi}$ denota que ϕ se adiciona al conjunto de creencias K generando una nueva creencia, $\hat{M}_k \xrightarrow{t_j} \hat{M}_{k+1}$.

2. Una vez que la nueva creencia \hat{M}_{k+1} se genera, ésta se incorpora al conjunto de creencias y se representa por $K^{+\phi}$ Además, si la nueva creencia \hat{M}_{k+1} es inconsistente, se usa la operación de contracción y la creencia revisada es nuevamente representada en el conjunto de creencias. A esta acción se denota por $K^{\pm\phi}$

Las operaciones de cambio de creencias son implementadas por la siguiente regla de disparo en *RdPI*

Definición 7.5 Sea $N_S = (Q, M_0)$ un sistema modelado en *RdPIVT* viva, binaria, acotada y evento detectable del sistema Υ y sea $N_O = (Q_O, \hat{M}_0)$ el modelo observador extendido en *RdPIVT*. Entonces, las operaciones de cambio de creencias pueden ser implementadas, definiendo el vector de disparo de transiciones v_k de N_O , usando la regla de disparo que se obtiene del algoritmo 5.1.

El observador implementado en *RdPI* no evento detectables cumplen con los postulados *AGM*, su demostración es muy similar a la que se presentó para el modelo del observador extendido.

Capítulo 8

Generador de intenciones

Resumen. Este capítulo presenta el modelo algebraico de las creencias, deseos e intenciones del agente en un entorno de *SED*.

En la arquitectura *BDI*, las creencias se definen como la información que tiene el agente acerca de su entorno, los deseos representan los objetivos o las situaciones que el agente quiere lograr y las intenciones representan los posibles cursos de acción disponibles del agente para alcanzar uno o más de sus objetivos. El principio de esta arquitectura es contar con un proceso para decidir qué curso de acción disponible tiene que ejecutar con la finalidad de cumplir los objetivos en base a las creencias que tenga el agente. Este proceso se compone principalmente de tres funciones.

1. La *función de revisión de creencias*. Esta función toma la percepción y las creencias actuales del agente y en base a ellos, determina un nuevo conjunto de creencias.
2. La *función de generación de opciones*. Esta función determina los objetivos disponibles del agente, en base a sus creencias e intenciones actuales. Esta función tiene dos propósitos.
 - a) Debe ser consistente, cualquier opción generada debe ser consistente con las creencias y con las intenciones actuales.
 - b) Debe ser oportuno, en el sentido de que debe de reconocer cuándo las circunstancias del ambiente cambian para ofrecer al agente nuevas formas de alcanzar los objetivos.
3. La *función de generación de intenciones o filtro*. Esta función genera las acciones del agente en base a sus creencias, deseos e intenciones actuales con dos finalidades.
 - a) Abortar cualquier intención que no puede ser alcanzable.
 - b) Retener aquellas intenciones que no son alcanzables y que aún pueden ser consideradas en un proceso posterior.

En este trabajo, el mecanismo de cada una de estas funciones se representan mediante el control por regulación bajo observación parcial. Así la función de revisión de creencias se implementa como parte del modelo del observador, la función de generación de opciones se representa con el modelo de la referencia y la función de generación de intenciones con el controlador.

El esquema del control por regulación bajo observación parcial que se utiliza en este trabajo se muestra en la figura 8.1. Esta es una extensión del esquema original propuesto por Santoyo [39] y [40] donde se incluye un modelo del observador. Note que la acción de control (ω) al modelo del sistema (S_f, M_0) se calcula por un controlador H , donde H usa la información del observador y del modelo de la referencia o especificación del sistema. En este esquema, es necesario que la ley de control mantenga igual la salida del modelo del sistema y del modelo de referencia.

El principal problema con el esquema del control por regulación bajo observación parcial es que el Principio de Separación no se ha resuelto para los Sistemas de Eventos

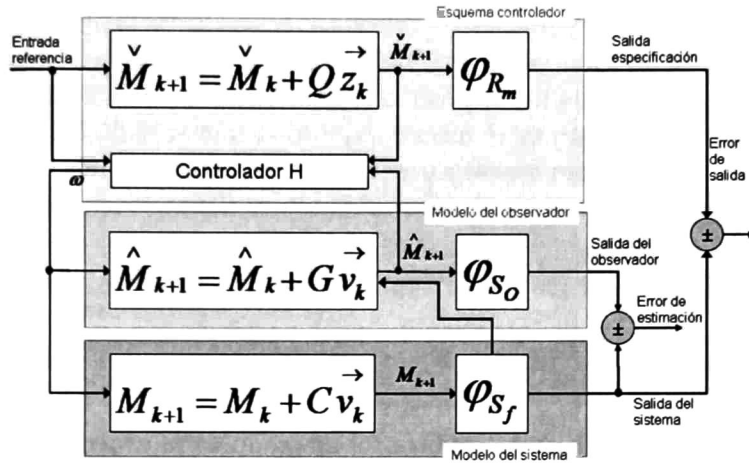


Figura 8.1: Esquema de control por regulación con el modelo del observador.

Discretos, es decir no es posible diseñar por separado el observador y el controlador para posteriormente unirlos, y así obtener un controlador por retroalimentación de estado estimado. En este trabajo se resuelve este problema modificando por separado el controlador presentado en [40] y el observador presentado en [31] y ponerlos a trabajar en conjunto para así obtener un controlador por retroalimentación de estado estimado.

8.1. Generador de deseos e intenciones

La finalidad de un agente implementado con la arquitectura *BDI* en un ambiente *SED* es realizar acciones en su entorno para lograr un objetivo en específico, es decir hacer que el sistema visite algunos marcados en un cierto orden, llamados "secuencias de marcados requeridos". Esas secuencias de marcados requeridos indican las actividades que el sistema debe ejecutar para obtener un producto intermedio. En general, las secuencias de marcados requeridos se representan por secuencias de marcados en otra *RdPI* llamada especificación o referencia.

El modelo de la especificación representa los objetivos que el agente debe alcanzar y tiene, generalmente, un número menor de lugares que el número de lugares del modelo del ambiente del agente. En este caso, la especificación se centra en la representación de algunos estados relevantes, es decir, aquellos que representan los subproductos correspondientes.

Debido a los pocos lugares de la especificación, los marcados del modelo en *RdPI* del sistema y de la especificación se comparan a través de las funciones de salida de los subproductos relevantes. Esas funciones, llamadas $\varphi_{S_f}^{PRS}$ y $\check{\varphi}_{PRS}$ para el sistema y la especificación, respectivamente, deben cumplir la siguiente restricción $\varphi_{S_f}^{PRS}(M_k^i) = \check{\varphi}_{PRS}(\check{M}_k)$ cuando ambos marcados M_k^i y \check{M}_k representan el mismo subproducto relevante

y $\varphi_{S_f}^{PRS}(M_j^i) \neq \check{\varphi}_{PRS}(\check{M}_k)$ cuando no lo estan.

Observe que en el sistema varios marcados podrían representar el mismo subproducto relevante porque el estado de los dispositivos del sistema no son relevantes para definir un subproducto (el subproducto es el mismo, incluso si una válvula está abierta o cerrada, etc). En la especificación, sin embargo, solamente un estado definirá el subproducto (no es una principal limitación ya que la especificación es dada por el usuario).

Bajo este escenario, el modelo del ambiente del agente en *RdPI* debe visitar los estados relevantes en el orden indicado por el modelo de la especificación.

El diseño del generador de intenciones que se propone en este trabajo se obtiene con los siguientes pasos:

1. Calcular la función Π para empotrar el marcado de la especificación en los respectivos marcados relevantes del *SED*.
2. Calcular la transiciones de disparo del modelo del ambiente del agente (vectores de Parikh)
3. Convertir los vectores de Parik en las secuencias de disparo de transiciones
4. Probar la controlabilidad de las secuencias de disparo de transiciones
5. Obtener la función parcial H para generar las acciones a realizar
6. Representar el generador de intenciones en una *RdPI*
7. Definir el conjunto de eventos activos e inactivos de cada marcado relevante del sistema alcanzado por la secuencia de disparo de transiciones.

8.1.1. Cálculo de la función Π y las transiciones de disparo del modelo del ambiente del agente

La definición de marcados equivalentes en trayectoria en salida, presentada en [39] y [40] se usa en este trabajo para calcular la función lineal Π y los vectores de Parikh como sigue.

Definición 8.1 Sea (S_f, M_0) y (R_m, \check{M}_0) el modelo en *RdPI* del sistema y de la especificación, respectivamente. Sea $R(S_f, M_0)$ y $R(R_m, \check{M}_0)$ los grafos de alcanzabilidad de (S_f, M_0) y (R_m, \check{M}_0) , respectivamente. (S_f, M_0) es equivalente en trayectoria en la salida con respecto a (R_m, \check{M}_0) si existe una función $\Pi : R(R_m, \check{M}_0) \rightarrow R(S_f, M_0)$ tal que

$$\sim_{T(S_f, M_0)} \circ \Pi \subseteq \Pi \circ \sim_{T(R_m, \check{M}_0)} \quad (8.1)$$

y

$$\forall \check{M}_j \in R(R_m, \check{M}_0), \varphi_{R_m}^{RC}(\check{M}_j) = \varphi_{S_f}^{RC}(\Pi \check{M}_j) \quad (8.2)$$

donde $\sim_{T(S_f, M_0)} = \left\{ (M_i, M_j) \mid M_i, M_j \in R(S_f, M_0) \text{ y } \exists t_q t_r \dots t_s \in T \text{ tal que } M_i \xrightarrow{t_q} M_l \xrightarrow{t_r} \dots \xrightarrow{t_s} M_j \right\}$
 $\sim_{T(R_m, \check{M}_0)} = \left\{ (\check{M}_i, \check{M}_j) \mid \check{M}_i, \check{M}_j \in R(R_m, \check{M}_0) \text{ y } \exists t_k \in \check{T} \text{ tal que } \check{M}_i \xrightarrow{t_k} \check{M}_j \right\}$ $\varphi_{S_f}^{PRS}$ es la función de salida para (S_f, M_0) .

Note que la ecuación (8.1) establece que el disparo de una transición en el modelo de la referencia puede ser simulado por el disparo de una secuencia de transiciones en el sistema. La ecuación (8.2) establece que los marcados del modelo del sistema y de la especificación deben dar la misma información de salida.

La función Π y los vectores de Parikh \vec{w}_m de la secuencia de transiciones $t_q t_r \dots t_s$ relacionados a la transición \check{t}_a se calculan con el siguiente Problema de Programación Lineal (PPL).

Algoritmo 8.1 Cálculo de la función Π

Entradas: El modelo del sistema y de la referencia $((S_f, M_0), (R_m, \check{M}_0))$

Salidas: La función Π

1.

$$PPL_5 = \begin{cases} \min \sum_{i,j} \Pi_{ij} + \sum_{m=1}^{|T|} \sum_{n=1}^{|\check{T}|} \vec{w}_m(n) \\ \text{s.a.} \\ \Pi \check{M}_0 = M_0 \\ \forall \check{t}_m \in \check{T}, \Pi \cdot Q \check{t}_m = C \vec{w}_m \\ \varphi \cdot \Pi = \check{\varphi} \\ \vec{w}_m(i) \geq 0 \end{cases} \quad (8.3)$$

Si no hay solución, entonces el algoritmo termina con error

La función lineal Π_5 empotra el marcado de la especificación en los respectivos marcados relevantes del *SED*. Como trabajamos en el caso de que algunas señales de salida del modelo del ambiente del agente no son medibles, el PPL_5 debe ejecutarse por cada posible marcado inicial del sistema m_i para obtener su correspondiente función lineal Π_i donde $\Pi_i \check{M}_0 = m_i$ y para cada transición \check{t}_a un vector de Parikh \vec{w}_a . Esto se ilustra en la figura 8.2.

8.1.2. Cálculo y prueba de controlabilidad en la secuencia de disparo de transiciones

Una vez calculada la función parcial Π_i y el vector de Parikh \vec{w}_a^i del marcado m_i correspondientes a la transición \check{t}_a , cada vector de Parikh se transforma en una secuencia de disparo de transiciones σ_a^i expandiendo parcialmente el grafo de alcanzabilidad.

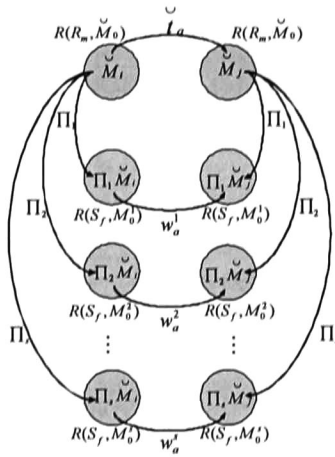


Figura 8.2: La función parcial Π_i y los vectores de Parikh w_a^i por cada t_a .

En este trabajo, se propone expandir el grafo de alcanzabilidad de la siguiente manera. Cuando la transición t_a se dispara en el marcado \check{M}_i , $\check{M}_i \xrightarrow{t_a} \check{M}_j$, entonces el nodo inicial o el nodo raíz del grafo de alcanzabilidad corresponde al marcado $\Pi_i \check{M}_i$. El marcado alcanzado m_k se calcula disparando eventos habilitados de la transición t_j calculada en \check{w}_a^i desde $\Pi_i \check{M}_i$, es decir, $\Pi_i \check{M}_i \xrightarrow{\lambda(t_j)} m_k$ tal que $\check{w}_a^i(t_j) \neq 0$. En el caso de que existan varias transiciones habilitadas t_k en el marcado $\Pi_i \check{M}_i$ con el mismo evento de t_j (evento de una transición manipulable o no manipulable), $\lambda(t_j) = \lambda(t_k)$, entonces cada transición t_k se dispara para alcanzar el marcado m_k , esto es, $\Pi_i \check{M}_i \xrightarrow{t_k} m_k$. Además, si una transición no manipulable t_k está habilitada en $\Pi_i \check{M}_i$ que no está calculada en \check{w}_a^i , $\check{w}_a^i(t_k) = 0$ entonces t_k se dispara. Para cada marcado diferente alcanzado m_k se crea un nodo y se añade un arco desde el nodo que representa el marcado $\Pi_i \check{M}_i$ a m_k . Este proceso se repite para cada marcado $m_k \in R(G, M_0)$ hasta que $m_k = \Pi_i \check{M}_j$ y se complete (opcionalmente) el T -*semiflujo* para cumplir la igualdad $C\check{w}_a^i = \Pi_i Q t_a$ estipulada en el algoritmo (8.3).

De esta manera, el grafo de alcanzabilidad PG_a^i para el vector de Parikh \check{w}_a^i será la secuencia de marcados desde el nodo $\Pi_i \check{M}_i$ hasta $\Pi_i \check{M}_j$, $PG_a^i = \Pi_i \check{M}_i \xrightarrow{t_m} M_a \xrightarrow{t_r} \dots M_h \xrightarrow{t_s} \Pi_i \check{M}_j$ y la secuencia de transiciones son las transiciones de las aristas de PG_a^i , $\sigma_a^i = t_m t_r \dots t_s$, donde $\overrightarrow{t_m t_r \dots t_s} = \check{w}_a^i + X$, X es un T -*semiflujo* del sistema o un vector nulo.

Si en la expansión parcial del grafo de alcanzabilidad existe más de una arista desde el nodo inicial, correspondiente al marcado $\Pi_i \check{M}_i$, significa que existen varios marcados que se pueden alcanzar desde $\Pi_i \check{M}_i$ con el disparo del evento $\lambda(t_m)$. En este caso, la secuencia de disparo de transiciones $\sigma_a^i = t_m t_r \dots t_s$ no es controlable. Otra de las pruebas de controlabilidad que se aplica a σ_a^i es la clásica de controlabilidad introducida en [46] y es escrita en términos de *RdPI*.

Definición 8.2 Sea (S_f, M_0) una RdPI y $K \subseteq \$(S_f, M_0)$ si y sólo si $\forall t_k \in T_{NM}$, $(\lambda(t_k) = \varepsilon)$ se mantiene que $\bar{K}t_k \cap \$(S_f, M_0) \subseteq \bar{K}$.

Significa que si en la secuencia de transiciones $\sigma_a^i = t_m t_r \dots t_s$ existe un marcado alcanzado m_k con el disparo de t_r , existe una transición no manipulable t_k que se puede disparar desde m_k , $m_k \xrightarrow{t_k} m_h$, el resto de la secuencia de transiciones $\dots t_s$ no se pueda realizar desde m_h . En este caso también, la secuencia de disparo de transiciones no es controlable.

8.1.3. Cálculo de la función parcial H

La función parcial H o la función de generación de intenciones, es la función que genera acciones a partir de las creencias y deseos actuales del agente, es decir, a partir del marcado actual del modelo del sistema y de referencia. En este trabajo, la función parcial H propuesta en [39] y [40] se define como sigue.

Definición 8.3 Sea (S_f, M_0) equivalente en trayectoria en la salida con respecto a (R_m, \check{M}_0) . Sea $R(S_f, M_0)$ y $R(R_m, \check{M}_0)$ su respectivo conjunto de alcanzabilidad. El problema de control por regulación de salida consiste en encontrar una función parcial

$$H : R(S_f, M_0) \times R(R_m, \check{M}_0) \times \check{T} \rightarrow \mathcal{L}_{mid}(S_f, M_0)$$

donde $H(\Pi \check{M}_i, \check{M}_i, t_k) = \omega_k$, $\omega_k \subseteq \mathcal{L}_{mid}(S_f, M_0)$ es controlable con respecto a $(S_f, \Pi \check{M}_i)$ y Π es la función que hace (S_f, M_0) equivalente en trayectoria con respecto a (R_m, \check{M}_0) .

El algoritmo para calcular la función parcial H es el siguiente.

Algoritmo 8.2 Cálculo de la función H

Entradas:

La expansión parcial del grafo de alcanzabilidad PG_a^i .

La función lineal Π

El marcado y transición del modelo de referencia $\check{M}_{k+1}, \check{t}_m$ respectivamente

Salidas: La función parcial H

1. Para cada expansión parcial del grafo de alcanzabilidad PG_m correspondiente a la \check{t}_m , la función parcial H se calcula de la siguiente manera:

$$H(\Pi \check{M}_k, \check{M}_{k+1}, \check{t}_m) = \lambda(t_q)$$

$$H(M_a, \check{M}_{k+1}, \varepsilon) = \lambda(t_r)$$

⋮

$$H(M_h, \check{M}_{k+1}, \varepsilon) = \lambda(t_s)$$

$$H(\Pi \check{M}_{k+1}, \check{M}_{k+1}, \varepsilon) = \varepsilon$$

8.1.4. Representación de la función parcial H como una $RdPI$

Usando la función parcial H , el comportamiento del lazo cerrado entre (S_f, M_0) y (R_m, \check{M}_0) se obtiene con el siguiente algoritmo.

Algoritmo 8.3 *Representación de la función parcial H como una $RdPI$*

▪ *Entradas*

(S_f, M_0) y (R_m, \check{M}_0) , El modelo del ambiente del agente en $RdPI$ y el modelo los objetivos del agente

▪ *Salidas*

$$H \left(\Pi \check{M}_i, \check{M}_j, \check{t}_m \right) = t_q t_r \dots t_s$$

1. *Defina un modelo en $RdPI$ del lazo cerrado como:*

$$Q_H = \begin{bmatrix} M_0 \\ \check{M}_0 \end{bmatrix} + \begin{bmatrix} C & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} v_k \\ z_k \end{bmatrix}$$

2. *Calcule los arcos de control como sigue:*

a) *Si $H \left(\Pi \check{M}_i, \check{M}_j, \check{t}_m \right) = \omega_m = t_q t_r \dots t_s$, entonces $\forall t_f \in \omega_m$ tal que t_f es una transición manipulable, calcule el conjunto de marcados habilitados como sigue:*

$$U_{t_f} = \left\{ M_n, \check{M}_j \mid \Pi \check{M}_i \xrightarrow{t_a} \dots M_n \xrightarrow{t_f} M_{n+1} \dots \xrightarrow{t_x} \Pi \check{M}_j \right\}$$

b) *Si $|U_{t_f}| = 2$ entonces se agrega un autolazo desde p_k a t_f , dónde $p_k \in P$ del modelo de la especificación tal que $M_n(p_k) = 1$ y $p_k \notin \{^*t_f\}$.*

c) *Si $|U_{t_f}| > 2$ entonces se agrega un arco inhibidor desde p_k a t_f , donde $p_k \in \check{P}$ tal que $\forall M_n(p_k) = 0$ y $p_k \notin \{^*t_f\}$.*

d) *Si $\check{M}_j \in U_{t_f}$ y $\check{M}_j(\check{p}_k) = 1$ y $\check{p}_k \in \check{P}$ entonces se agrega un autolazo entre \check{p}_k y t_f .*

8.1.5. Definición del conjunto de eventos activos e inactivos

El evento activo es el evento calculado por la función H_i en el marcado $\Pi_i \check{M}_i$ y los eventos inactivos son aquellos eventos habilitados en el marcado $\Pi_i \check{M}_i$ y que no son calculados por la función H_i . Los eventos inactivos son útiles para identificar marcados que pueden alcanzar estados no deseados o que contradigan los objetivos del agente. Estos eventos se definen a continuación.

Definición 8.4 Sea (S_f, M_0) y (R_m, \check{M}_0) el modelo en *RdPI* del sistema y de la referencia, respectivamente. Sea Π_i y H_i la función lineal y el generador de intenciones, respectivamente y \vec{w}_a^i el vector de Parikh correspondiente a \check{t}_a . El evento activo en $\Pi_i \check{M}_i$ es

$$Act_{m_i} = \left\{ H_i \left(\Pi_i \check{M}_i, \check{M}_j, \check{t}_a \right) \mid \Pi_i \check{M}_i = m_i \right\}$$

El conjunto de los eventos inactivos en $\Pi_i \check{M}_i$ es

$$InA_{m_i} = \left\{ \lambda(t_j) \mid \Pi_i \check{M}_i \xrightarrow{t_j}, \vec{w}_a^i(t_j) = 0 \text{ and } \Pi_i \check{M}_i = m_i \right\}$$

Note que $Act_{m_i} \cap InA_{m_i} = \emptyset$ en $\Pi_i \check{M}_i$ porque el modelo del ambiente del agente es determinista [29] sobre los eventos de las transiciones (símbolos de entrada). Además, el posible marcado inicial $\Pi_i \check{M}_0$ no habilita transiciones no manipulables.

Puede existir el caso que $Act_{m_i} \cap InA_{m_i} \neq \emptyset$ en $\Pi_i \check{M}_i$ cuando dos transiciones habilitadas t_i y t_j en $\Pi_i \check{M}_i$ tienen el mismo evento asociado (diferente del símbolo ε) y diferente lugar de entrada, esto es, $\lambda(t_i) = \lambda(t_j) \neq \varepsilon$ y $\cdot t_i \neq \cdot t_j$, donde $\Pi_i \check{M}_i \xrightarrow{t_i}$ y $\Pi_i \check{M}_i \xrightarrow{t_j}$. En este caso, cuando la función de generación de intenciones emite la secuencia de eventos $\lambda(t_i) \lambda(t_r) \dots \lambda(t_s)$, el modelo en *RdPI* del generador de intenciones se vuelve no determinista porque con el símbolo $\lambda(t_i)$ se puede disparar t_i o t_j . Para evitar este problema, se verifica la secuencia de disparo de las transiciones como en la sección anterior.

8.2. Generador de creencias

En esta sección presenta otro mecanismo de la función de revisión de creencias del estado del ambiente del agente, compuesto por un sistema dinámico, llamado observador. El observador es una copia del modelo del ambiente del agente más estructura adicional. La metodología para diseñar el observador utiliza algoritmos polinomiales para definir el marcado inicial, el vector de disparo de transiciones y para calcular el estado del observador. Las reglas de disparo del observador se definen utilizando las *LCM*, la estructura del observador, la matriz de eventos en común y las señales de salida del sistema para actualizar el marcado del observador. La convergencia del marcado estimado (creencias del estado del ambiente del agente) al estado actual del sistema se reduce rápidamente por el uso de algoritmos polinomiales.

El diseño de la función de revisión de creencias del estado del ambiente del agente se obtiene con los siguientes pasos:

1. Calcular las posibles creencias iniciales
2. Definir el modelo del observador
3. Establecer la creencia inicial del estado del ambiente del agente
4. Calcular creencias equivalentes y matriz de eventos en común

5. Definir las reglas de disparo del observador.

8.2.1. Cálculo de las posibles creencias iniciales

Debido a que en el modelo del ambiente del agente en *RdPI* puede haber varios marcados que cumplan con la salida inicial del modelo del sistema, se calculan todos los posibles marcados iniciales del observador. Cada uno de estos posibles marcados iniciales se utiliza para definir la función H que se presentó en la sección anterior y así obtener los eventos activos e inactivos que determinan la creencia inicial adecuada para el agente. El cálculo de los posibles marcados iniciales del observador que cumplen con la salida inicial del modelo del sistema se realiza con el siguiente algoritmo.

Algoritmo 8.4 *Cálculo de los posibles marcados iniciales*

Entradas:

Marcado inicial del modelo del sistema M_0

Las leyes conservativas de marcado $\Gamma_{s \times n}$

El conjunto de transiciones no manipulables $T_{uc} = \{t_i \in T \mid \lambda(t_i) = \varepsilon\}$

Salidas:

El conjunto de posibles marcados iniciales μ_0

Restricción = \emptyset

$\mu_0 = \emptyset$

Hacer

$$PPL_6 = \begin{cases} \text{s.a.} & \min \sum_i \Psi_{ns1}(i) \\ & \varphi\Psi = \varphi M_0 \\ & 0 \leq \Psi(i) \leq 1 \\ & \Gamma\Psi = K \\ & \sum_{p_i \in \bullet t_j} \Psi(p_i) \leq |\bullet t_j| - 1, t_j \in T_{uc} \\ & \text{Restricción} \end{cases}$$

Si PPL_6 tiene solución

entonces Restricción = Restricción $\cup \sum \Psi(p_i) \leq s - 1$, tal que $\Psi(p_i) = 1$, donde s es la cantidad de LCMB

$$\mu_0 = \mu_0 \cup \Psi$$

End_If

Hasta que PPL_6 no tenga solución.

El PPL_6 calcula todos los diferentes posibles marcados que cumplen con la salida inicial del sistema tal que el marcado en los lugares computables es el actual y el marcado en los lugares no computables se calculan para cumplir con las LCMB. Además, los posibles marcados iniciales no habilitan transiciones no manipulables.

8.2.2. Definición del modelo del observador

La estructura propuesta para el modelo del observador es una copia de la estructura del modelo del sistema, con estructura adicional, como sigue.

Definición 8.5 Sea $N = (Q_S, M_0)$ una RdPI viva, binaria y evento detectable, donde $Q_s = (P, T, I, O, \Sigma, \lambda, \varphi)$. El modelo del observador es $S = (Q_O, \hat{M}_0)$ donde:

$Q_O = (\hat{P}, \hat{T}, I_S, O_S, \Sigma, Id, Id)$, donde $\hat{P} = P$, $\hat{T} = T \cup \{t_1^i, \dots, t_n^i\} \cup \{t_1^o, \dots, t_n^o\}$ y la ecuación de estados es

$$\begin{aligned}\hat{M}_{k+1} &= \hat{M}_k + [C \quad -I \quad I] v_k \\ \hat{y}_k &= \varphi(\hat{M}_k)\end{aligned}$$

I es la matriz identidad. I_S, O_S puede ser inferido de su ecuación de estados.

8.2.3. Cálculo de las creencias equivalentes y matriz de eventos en común

Para establecer la creencia inicial del ambiente del agente (marcado inicial del observador), es necesario seleccionar un posible marcado inicial m_i de μ_0 tal que el evento activo de m_i , Act_{m_i} , no sea el común entre los eventos inactivos de otro marcado m_j , InA_{m_j} , $m_j \in \mu_0$, $m_i \neq m_j$, esto es, $Act_{m_i} \cap InA_{m_j} = \emptyset$. Esto significa que si se selecciona el marcado m_i como creencia inicial del agente y el estado actual del sistema es m_j , $Act_{m_i} \cap InA_{m_j} \neq \emptyset$, entonces, el agente al generar el evento Act_{m_i} , esto es, $H(m_i, \check{M}_j, \check{t}_a) = \lambda(t_j) = Act_{m_i}$, hará que el sistema evolucione disparando el evento InA_{m_j} y se alcance un estado no deseado. Note que $Act_{m_i} \cap InA_{m_j} \neq \emptyset$ puede darse para cualquier $m_i = \Pi_i \check{M}_i$. Al marcado m_i tal que $Act_{m_i} \cap InA_{m_j} \neq \emptyset$ se le llama marcado prohibido.

Para identificar estos posibles marcados prohibidos se agrupan todos los marcados equivalentes con respecto al marcado del modelo de referencia.

Definición 8.6 Sea (S_f, M_0) y (R_m, \check{M}_0) el modelo en RdPI del sistema y de la referencia, respectivamente. Sea μ_0 el conjunto de posibles marcados iniciales y Π_i la función lineal correspondiente al marcado $m_i \in \mu_0$.

El conjunto de marcados equivalentes en salida es

$$\mu_a^j = \left\{ \Pi_i \check{M}_j | \check{M}_j \xrightarrow{\check{t}_a}, i \in [1 \dots |\mu_0|] \right\}$$

Observe que μ_a^j contiene todos los marcados equivalentes en salida o creencias equivalentes del estado del ambiente del agente, relacionados con el marcado \check{M}_j que habilita

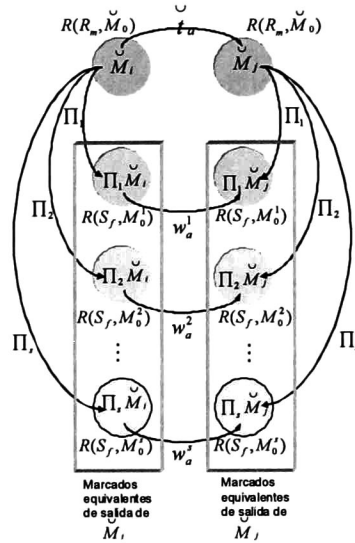


Figura 8.3: Los marcados equivalentes de salida

la transición \check{t}_a . Además existe un conjunto de creencias equivalentes μ_a^j por cada marcado del modelo de referencia, $\check{M}_j \in (R_m, \check{M}_0)$. Los marcados equivalentes de salida se muestran en la figura 8.3.

Para identificar los marcados prohibidos en el marcado $\Pi_i \check{M}_j$, $i \in [1 \dots |\mu_0|]$, se utiliza la matriz de eventos en común $\Omega_{\mu_a^j}$ y se define como sigue.

Definición 8.7 Sea (S_f, M_0) y (R_m, \check{M}_0) el modelo en RdPI del sistema y de la referencia, respectivamente. Sea μ_0 el conjunto de posibles estados iniciales que cumplen la observación inicial del sistema y_0 . Sea μ_a^j el conjunto de marcados equivalentes de salida, y Act_{m_i} , InA_{m_i} los conjuntos de eventos activos e inactivos de $m_i \in \mu_a^j$, respectivamente. Las entradas de la matriz de eventos en común $\Omega_{\mu_a^j}$ de μ_a^j es

$$\Omega_{\mu_a^j}(i, j) = |Act_{m_i} \cap InA_{m_j}|$$

donde $\Omega_{\mu_a^j}(i, j)$ es el número de eventos en común que se deben disparar en el marcado $m_i \in \mu_a^i$ y que se deben evitar en el marcado $m_j \in \mu_a^j$.

Como el modelo de referencia es una máquina binaria fuertemente conexas, por cada transición \check{t}_a hay una matriz de eventos en común $\Omega_{\mu_a^j}$ con dimensión $|\mu_a^i| \times |\mu_a^j|$.

Cuando la transición \check{t}_a se dispara en el marcado \check{M}_{j-1} , esto es, $\check{M}_{j-1} \xrightarrow{\check{t}_a} \check{M}_j$ la función H_i emite los eventos correspondientes a la secuencia de disparo de transiciones $\sigma_a^i = t_m t_r \dots t_s$ y el conjunto de marcados equivalentes de salida μ_b^i , tal que $\forall m_j = \Pi_i \check{M}_j$, $m_j \in \mu_b^i$, se refina, quedando los marcados alcanzados por los eventos de la secuencia σ_a^i

y que son iguales bajo la señal de salida con el modelo de referencia, esto es, $\Pi_i \check{M}_{j-1} \xrightarrow{t_m} M_a \xrightarrow{t_r} \dots M_h \xrightarrow{t_a} \Pi_i \check{M}_j$, $m_j = \Pi_i \check{M}_j$ y $\varphi_{S_j}^{PRS}(m_j) = \check{\varphi}_{RS}(\check{M}_j)$. A la matriz de eventos en común $\Omega_{\mu_a^j}$ correspondiente al refinamiento de μ_b^i por el disparo de σ_a^i se denota como $\Omega_{\mu_a^j | \sigma_a^i}$.

Si entre los marcados prohibidos de μ_a^j forman un ciclo en $\Omega_{\mu_a^j}$ o $\Omega_{\mu_a^j | \sigma_a^i}$, esto es, $m_i, m_{i+1}, \dots, m_q \in \mu_a^j$ tales que $Act_{m_i} \cap InA_{m_{i+1}} \neq \emptyset$, $Act_{m_{i+1}} \cap InA_{m_{i+2}} \neq \emptyset$, \dots , $Act_{m_q} \cap InA_{m_i} \neq \emptyset$ entonces no existe una posible creencias, que si es considerada para generar acciones, el sistema no evoluciona violando los objetivos del agente.

8.2.4. Mercado inicial

El mercado inicial del observador es la creencia inicial del estado del ambiente del agente y se calcula como sigue.

Definición 8.8 Sea $N = (Q, M_0)$ una RdPI viva, binaria y evento detectable donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$. Sea S su observador. La condición inicial del observador \hat{M}_0 se calcula como sigue

$$\hat{M}_0 = m_i \text{ tal que } m_i \in \mu_a^0 \text{ y } \Omega_{\mu_a^0}(i, \cdot) = \vec{0}$$

dónde $\Omega_{\mu_a^0}$ es la matriz de eventos en común para el conjunto de marcados equivalentes de salida μ_a^0 .

La creencia inicial del estado del ambiente del agente es una de las posibles creencias del conjunto μ_a^0 , $m_i \in \mu_a^0$ tal que el evento calculado por $H_i(m_i, \check{M}_j, t_a) = Act_{m_i}$ no es un evento que se debe evitar en otra creencia $m_j \in \mu_a^0, \forall m_j, m_j \neq m$. Así, el agente realiza acciones en el SED en base a la creencia m_i sin contradecir los objetivos o llegue a estados no deseados, esto es, Act_{m_i} no es común con los eventos inactivos de m_j , $Act_{m_i} \cap InA_{m_j} = \emptyset$.

8.2.5. Reglas de disparo

Las reglas de disparo del observador se deben calcular de tal forma que el marcado del observador tienda al marcado actual del sistema cada que se dispara una transición del modelo del sistema. Además, como la función H_i usa el marcado estimado del observador como creencia del estado del ambiente para generar acciones o eventos, los estados que se alcancen no deben contradecir los objetivos o llegue a estados no deseados.

Como el modelo del ambiente del agente en RdPI es evento detectable, el disparo de la transición se distingue. Las reglas de disparo del observador se definen con el siguiente algoritmo, donde los cambios en la salida del sistema y los marcados equivalentes en salida μ_a^j son usados para corregir en línea el marcado del observador.

Algoritmo 8.5 Actualizar el mercado del observador

Entradas:

\hat{M}_k El mercado actual del modelo del observador

y_k La salida actual del sistema

\check{M}_j El mercado actual del modelo de referencia

t_j La transición detectada de disparo en el sistema

σ_j El evento emitido por la función H_i

μ_a^j El mercado equivalente en salida con respecto a \check{M}_j

$\Omega_{\mu_a^j}$ La matriz de eventos en común de μ_a^j .

Salida

\hat{M}_{k+1} El mercado estimado del observador

Si t_j no está habilitada en el mercado M_k y $\lambda(t_j) = \sigma_j$ entonces

Para cada mercado $m_k \in \mu_a^{j-1}$ tal que $m_k \xrightarrow{t_j}$

Borrar el renglón y columna i de $\Omega_{\mu_a^{j-1}}$, tal que i es la posición correspondiente de Act_{m_k} y InA_{m_k} de m_k ,

Borrar de μ_a^{j-1} el mercado $\Pi_i \check{M}_r$, esto es, $\mu_a^{j-1} = \mu_a^{j-1} - \{m_k\}$

• *Fin_Para*

Calcular otro mercado observado $\hat{M}_{k+1} = m_i$ tal que $\Omega_{\mu_a^{j-1}}(i, \cdot) = \vec{0}$

Salida del algoritmo

Fin_Si

Si \hat{t}_j está habilitada en el mercado \hat{M}_k entonces

Para cada $\check{M}_r \in R(R_m, \check{M}_0)$

$\mu_b^r = \mu_b^r - \{\Pi_l \check{M}_r\}$, $\forall \Pi_l \neq \Pi_i$, tal que $\Pi_i \check{M}_{j-1} \xrightarrow{\vec{w}_j^i} \Pi_i \check{M}_j$ y $\vec{w}_j^i(t_j) \neq 0$

Borrar el renglón y columna i de $\Omega_{\mu_b^r}$, tal que i es la posición correspondiente

de $Act_{\Pi_i \check{M}_r}$ y $InA_{\Pi_i \check{M}_r}$ de $\Pi_i \check{M}_r$, $\forall \Pi_l \neq \Pi_i$, donde $\Pi_i \check{M}_{j-1} \xrightarrow{\vec{w}_j^i} \Pi_i \check{M}_j$ y $\vec{w}_j^i(t_j) \neq 0$

Fin_Para

$\hat{M}_{k+1} = \hat{M}_k + C \vec{\hat{t}}_j$

Salir del algoritmo.

De lo contrario

Calcular el conjunto P_i de lugares de entrada de \hat{t}_j

Calcular el conjunto P_o de lugares de salida de \hat{t}_j

Calcular el conjunto T_i^i de transiciones extra de entrada de P_i

Calcular el conjunto T_o^i de transiciones extra de entrada de P_o

Calcular el conjunto T_i^o de transiciones extra de salida de P_i

Calcular el conjunto T_o^o de transiciones extra de salida de P_o

En el observador, el vector de disparo v_k se calcula como sigue:

$$v_k(t_a) = 1 \text{ si } t_a \in T_o^i; v_k(t_b) = 1 \text{ si } t_b \in T_i^o \text{ y } t_b \text{ está habilitado. De lo contrario } v_k(t) = 0$$

Ejecutar el disparo del vector v_k , esto es, $\hat{M}'_{k+1} = \hat{M}_k + [C \quad -I \quad I] v_k$

El marcado del observador \hat{M}_{k+1} se calcula como sigue:

$$PPL_7 = \left\{ \begin{array}{l} \text{mín } \sum_i v_k(i) \\ \text{s.a.} \\ \hat{M}_{k+1} = \hat{M}'_{k+1} + Cv_k \\ \hat{M}_{k+1}(i) \geq 0 \\ \hat{M}_{k+1}(i) \leq 1 \\ \varphi \cdot \hat{M}_{k+1} = \varphi \cdot M_{k+1} \\ \Gamma \hat{M}_{k+1} = K \\ v_k(t_a) = 0 \text{ si } t_a \in T_i^i \cup T_o^i \cup T_i^o \cup T_o^o \cup \{\hat{t}_j\} \end{array} \right.$$

Fin_Si

El anterior algoritmo establece que, cuando la transición \check{t}_a en el marcado \check{M}_{j-1} se dispara en el modelo de referencia, esto es, $\check{M}_{j-1} \xrightarrow{\check{t}_a} \check{M}_j$, la función H_i emite el evento σ_j usando el marcado del observador \hat{M}_k , $H_i(\hat{M}_k, \check{M}_j, \check{t}_a) = \lambda(t_j) = \sigma_j$, donde $\Pi_i \check{M}_{j-1} = \hat{M}_k \in \mu_a^{j-1}$. Si la transición t_j no está habilitada en M_k en el modelo del sistema, entonces el marcado del observador se corrige, borrando de μ_a^{j-1} todos los marcados (y sus correspondientes eventos activos e inactivos de $\Omega_{\mu_a^{j-1}}$) que habilitan la transición t_j . Se selecciona otro marcado $\Pi_j \check{M}_{j-1} = m'$ del conjunto μ_a^{j-1} , $\Pi_j \neq \Pi_i$ ya que es uno de los que el sistema evolucionaría sin violar la especificación porque los eventos calculados por la función H_j usando m' no es igual a los eventos inactivos de otro marcado m'' de μ_a^{j-1} , $\Pi_k \check{M}_{j-1} = m''$, $m'' \neq m'$, esto es, $H_j(m', \check{M}_j, \check{t}_a) \neq InA_{m''}$ y $\Omega_{\mu_a^{j-1}}(j, \cdot) = \vec{0}$, donde j es $Act_{m'}$. Si t_j está habilitada en el marcado M_k entonces t_j se dispara en el modelo del sistema y del observador, esto es, $M_k \xrightarrow{t_j} M_{k+1}$ y $\hat{M}_k \xrightarrow{\hat{t}_j} \hat{M}_{k+1}$, respectivamente. Los marcados $\Pi_j \check{M}_{j-1} \in \mu_a^{j-1}$, $\forall \Pi_j \neq \Pi_i$ tal que $\Pi_j \check{M}_{j-1} \xrightarrow{\sigma_a^j} \Pi_j \check{M}_j$ y σ_a^j no incluye a t_j son borrados. Esto significa que los marcados $\Pi_j \check{M}_{j-1}$ no son el estado actual del sistema y los

marcados equivalentes de salida $\Pi_j \check{M}_{j-1}$ no se alcanzarán. Entonces, los marcados $\Pi_j \check{M}_r \in \mu_b^r$, $\forall \check{M}_r \in R(R_m, \check{M}_0)$ son borrados (y sus correspondientes eventos activos e inactivos de $\Omega_{\mu_b^r}$) logrando así un refinamiento en todos los conjuntos de marcados equivalentes de salida y en las matrices de eventos en común.

De forma similar, si la transición \hat{t}_j no está habilitada en el modelo del observador y t_j se dispara en el modelo del sistema, entonces las transiciones extras de entrada y salida del observador, se utilizan para simular el disparo de \hat{t}_j , alcanzando el marcado \hat{M}'_{k+1} . Observe que el vector de disparo del observador se calcula usando el PPL_7 donde el marcado alcanzado \hat{M}_{k+1} a partir de \hat{M}'_{k+1} cumple con las $LCMB$ y con las señales de salida del modelo del sistema.

Para demostrar que el marcado del modelo del observador propuesto en este trabajo tiende al estado actual del marcado del sistema, el error de observación se define a continuación.

Definición 8.9 Sea $N = (Q_S, M_0)$ una $RdPI$ viva, binaria y evento detectable, donde $Q = (P, T, I, O, \Sigma, \lambda, \varphi)$. Sea $S = (Q_O, \hat{M}_0)$ el observador de (Q_S, M_0) . El error de observación es $|e_k| = \sum_{\substack{p \in P, \\ \hat{p} \in \hat{P}}} |M_k(p) - \hat{M}_k(\hat{p})|$.

El siguiente teorema utiliza las reglas de disparo de las transiciones del observador.

Teorema 8.1 Sea $N = (Q, M_0)$ una $RdPI$ viva, binaria y evento detectable que modela el sistema y $S = (Q_O, \hat{M}_0)$ es su modelo del observador, entonces $\forall k |e_{k+1}| \leq |e_k|$

Demostración. Sea $X = \{x_1, \dots, x_r\}$ el conjunto de T-Componentes elementales de N , $B = \{b_1, \dots, b_s\}$ el conjunto de $LCMB$ definidas en N , P_{x_i} el conjunto de lugares que pertenecen a x_i y P_{b_i} el conjunto de lugares que pertenecen a b_i .

Procedemos a demostrar por inducción en la longitud de disparo de σ en el modelo del sistema.

1. Si $|\sigma| = 0$ entonces $k = 0$, $\Pi_i \check{M}_0 = \hat{M}_0 \in \mu_a^0$ se calcula como en la definición 8.8, entonces el error inicial es $e_0 = M_0 - \hat{M}_0$.
2. Si $|\sigma| = 1$ entonces $k = 1$. Suponga que la transición \check{t}_a en el marcado \check{M}_0 se dispara en el modelo de referencia, esto es, $\check{M}_0 \xrightarrow{\check{t}_a} \check{M}_1$. El controlador H_i emite el evento σ_j usando el marcado del observador \hat{M}_0 , $H_i(\hat{M}_0, \check{M}_1, \check{t}_a) = \lambda(t_j) = \sigma_j$, donde $\Pi_i \check{M}_0 = \hat{M}_0 \in \mu_a^0$. Dos casos son posibles en el sistema:
 - a) La transición t_j está habilitada en M_0 . Así, t_j se dispara en el modelo del sistema, esto es, $M_0 \xrightarrow{t_j} M_1$. Como N es evento detectable, entonces, por el cambio de salida $\varphi(M_1) - \varphi(M_0)$ el disparo de t_j se detecta. También, como N es binaria,

seguro que $M_1(p_i) = 0 \forall p_i \in \bullet t_j$ y $M_1(p_z) = 1 \forall p_z \in t_j \bullet$. Entonces \hat{t}_j se dispara en \hat{M}_0 en el observador S alcanzado el marcado \hat{M}_1 , esto es, $\hat{M}_0 \xrightarrow{\hat{t}_j} \hat{M}_1$. Note que $\hat{M}_1(\hat{p}_z) = 1 \forall \hat{p}_z \in \hat{t}_j \bullet$ porque \hat{M}_0 cumple con las *LCMB*, así, si \hat{t}_j está habilitada en \hat{M}_0 entonces $\hat{M}_0(\hat{p}_z) = 1 \forall \hat{p}_z \in \bullet \hat{t}_j$ y por las *LCMB*, $\hat{M}_0(\hat{p}_z) = 0 \forall \hat{p}_z \in \hat{t}_j \bullet$, así, al disparar \hat{t}_j una marca se agrega a los lugares $\hat{p}_z \in \hat{t}_j \bullet$. Además, los marcados $\Pi_j \check{M}_0 \in \mu_a^0$, $\forall \Pi_j \neq \Pi_i$ tales que $\Pi_j \check{M}_0 \xrightarrow{\sigma_a^0} \Pi_j \check{M}_1$ y σ_a^0 no incluyen a t_j se borran. Entonces, los marcados $\Pi_j \check{M}_r \in \mu_b^r$, $\forall \check{M}_r \in R(R_m, \check{M}_0)$ así como $\Omega_{\mu_b^r}(r, \cdot)$ y $\Omega_{\mu_b^r}(\cdot, r)$ se borran refinando todos los marcados equivalentes de salida y las matrices de eventos comunes. Así \hat{M}_1 cumple con *LCMB* y ninguna otra transición, a parte de \hat{t}_j se debe disparar. Como las mismas marcas fueron removidas en ambas redes, N y S , entonces $|e_1| = |e_0|$.

- b. La transición t_j no está habilitada en M_0 . Esto significa que $\exists p_i \in \bullet t_j$, $M_0(p_i) = 0$, donde $\varphi(r, i) = 1$ y $\check{y}_0(r) = 1$. Entonces existe otro posible marcado $\Pi_j \check{M}_0 = \hat{M}'_0 \in \mu_a^0$ tal que $\Pi_j \neq \Pi_i$. Suponga que el evento calculado por H_j usando \hat{M}'_0 es igual al evento inactivo del marcado \hat{M}_0 , esto es, $H_j(\hat{M}'_0, \check{M}_1, \check{t}_a) = \lambda(t_k) \in InA_{\hat{M}_0}$, y $\Omega_{\mu_a^0}(j, i) \neq \vec{0}$ donde el renglón j es $Act_{\hat{M}'_0}$ y la columna i es $InA_{\hat{M}_0}$. Como el marcado M_0 no habilita la transición t_j entonces todos los marcados $\Pi_k \check{M}_0 \in \mu_a^0$ tales que $\Pi_k \check{M}_0 \xrightarrow{t_j}$ (incluyen a \hat{M}_0) se eliminan de μ_a^0 , esto es, $\mu_a^0 = \mu_a^0 - \{\Pi_k \check{M}_0\}$ así como $\Omega_{\mu_a^0|\Pi_k \check{M}_0}(r, \cdot)$ y $\Omega_{\mu_a^0|\Pi_k \check{M}_0}(\cdot, r)$, donde el renglón y la columna r son $Act_{\mu_a^0|\Pi_k \check{M}_0}$ y $InA_{\mu_a^0|\Pi_k \check{M}_0}$, respectivamente. Así, los eventos calculados por la función H_j usando el posible marcado \hat{M}'_0 no contiene eventos inactivos de otro marcado, esto es, $\Omega_{\mu_a^0}(i, \cdot) = \vec{0}$. Por lo tanto, el nuevo marcado inicial del observador es \hat{M}'_0 , $\hat{M}_0 = \hat{M}'_0$. Como no se disparó la transición t_j , el error es $e_0 = M_0 - \hat{M}_0$

Entonces, en ambos casos $|e_1| \leq |e_0|$

3. Asuma que $|\sigma| = k$ se dispara en el marcado M_0 , esto es, $M_0 \xrightarrow{\sigma} M_k$, entonces la secuencia $\hat{\sigma}_k$ se dispara en el observador, esto es, $\hat{M}_0 \xrightarrow{\hat{\sigma}_k} \hat{M}_k$ así $e_k = M_k - \hat{M}_k$ y por hipótesis de inducción $|e_k| \leq |e_{k-1}|$.
4. Si $|\sigma| = k + 1$ y σt_j . Suponga que la transición \check{t}_b en el marcado \check{M}_k se dispara en el modelo de referencia, esto es, $\check{M}_k \xrightarrow{\check{t}_b} \check{M}_{k+1}$. El controlador H_i emite el evento σ_j usando el marcado del observador \hat{M}_k , esto es, $H_i(\hat{M}_k, \check{M}_{k+1}, \check{t}_b) = \lambda(t_j) = \sigma_j$, donde $\Pi_i \check{M}_k = \hat{M}_k \in \mu_b^k$. Dos casos son posibles en el modelo del sistema.
- a) La transición t_j está habilitada en el marcado M_k . Esto significa que t_j se dispara en el marcado M_k del sistema, esto es, $M_k \xrightarrow{t_j} M_{k+1}$. Como N es evento detectable, entonces, por el cambio de salida $\varphi(M_1) - \varphi(M_0)$ el disparo de t_j se detecta. También, como N es binaria, entonces $M_1(p_i) = 0 \forall p_i \in \bullet t_j$ y $M_1(p_z) = 1 \forall p_z \in t_j \bullet$. Dos casos son posibles en el observador.

- 1) La transición \hat{t}_j está habilitada en S . Así \hat{t}_j se dispara en el mercado \hat{M}_k en S alcanzando el mercado \hat{M}_{k+1} , esto es, $\hat{M}_k \xrightarrow{\hat{t}_j} \hat{M}_{k+1}$. Note que $\hat{M}_{k+1}(\hat{p}_z) = 1 \forall \hat{p}_z \in \hat{t}_j \bullet$ porque \hat{M}_k cumple las *LCMB*, así si \hat{t}_j está habilitada en \hat{M}_k entonces de seguro $\hat{M}_k(\hat{p}_z) = 1 \forall \hat{p}_z \in \bullet \hat{t}_j$ y por las *LCMB*, $\hat{M}_k(\hat{p}_z) = 0 \forall \hat{p}_z \in \hat{t}_j \bullet$, así al disparar \hat{t}_j una marca se agrega al lugar $\hat{p}_z \in \hat{t}_j \bullet$. Además, el mercado $\Pi_j \check{M}_k \in \mu_b^k$, $\forall \Pi_j \neq \Pi_i$ tal que $\Pi_j \check{M}_k \xrightarrow{\sigma_b^k} \Pi_j \check{M}_{k+1}$ y σ_b^k no incluyen a t_j se eliminan. Entonces los marcados $\Pi_j \check{M}_r \in \mu_c^r$, $\forall M_r \in R(R_m, M_0)$ así como $\Omega_{\mu_b^r}(r, \cdot)$ y $\Omega_{\mu_b^r}(\cdot, r)$ se borran refinando todos los marcados equivalentes de salida y las matrices de eventos comunes. Así \hat{M}_{k+1} cumple con *LCMB* y ninguna otra transición, a parte de \hat{t}_j debe dispararse. Como las mismas marcas fueron removidas en ambas redes, N y S entonces $|e_k| = |e_{k-1}|$.
- 2) La transición \hat{t}_j no está habilitada en el mercado \hat{M}_k en S . Esto significa que t_j es una transición incontrolable o bien, una transición no calculada por el controlador H_i y existen lugares $\forall \hat{p}_i \in \bullet \hat{t}_j$ tales que $\hat{M}_k(p_i) = 0$. La transición extra t_z^i de $\hat{p}_z \in \hat{t}_j \bullet$ y t_w^o de $\hat{p}_w \in \bullet \hat{t}_j$ debe dispararse para simular el disparo de \hat{t}_j y la transición extra t_z^o de $\hat{p}_z \in \hat{t}_j \bullet$ y t_w^i de $p_w \in \bullet \hat{t}_j$ no se debe disparar. Entonces, el disparo de esas transiciones extras alcanzan un nuevo mercado \hat{M}'_{k+1} y \hat{M}'_{k+1} no cumple con *LCMB*. Después, algunas transiciones extras diferentes a las mencionadas $t_a t_b t_c \dots$, se deben disparar de tal forma que $\hat{M}'_{k+1} \xrightarrow{t_a t_b t_c \dots} \hat{M}_{k+1}$ y \hat{M}_{k+1} cumple con *LCMB*. Para calcular esas transiciones, observe que para todas las *LCMB* b_q que contienen el lugar $\hat{p}_z \in \hat{t}_j \bullet$, otros lugares \hat{p}_r , $\hat{p}_z \neq \hat{p}_r$, no deben tener marcas, esto es, $\hat{M}_{k+1}(\hat{p}_r) = 0$. Así, si uno de esos lugares \hat{p}_r tienen una marca, entonces la transición extra de salida t_z^o se debe disparar. Entonces, para un lugar $\hat{p}_w \in \bullet \hat{t}_j$ cumple que $|e_k(\hat{p}_w)| = 1$, porque tiene una marca en el sistema y ninguna en el observador. También, para lugares \hat{p}_r con una marca en \hat{M}_k cumple que $|e_k(\hat{p}_z)| = 1$, porque estuvo marcado en el observador pero en mercado actual del sistema no. Después del disparo propuesto de las transiciones extras en el observador, el error en esos lugares son cero, así $|e_k| < |e_{k-1}|$.
Entonces, en ambos casos $|e_{k+1}| \leq |e_k|$.
- b) La transición t_j no está habilitada en M_k . Esto significa que \hat{t}_j está habilitada en el mercado \hat{M}_k en S y la ley de control calculada por la función parcial H_i es el evento σ_j tal que $\lambda(t_j) = \sigma_j$, entonces existen lugares $\forall p_i \in \bullet t_j$, $M_k(p_i) = 0$. Entonces existe otro posible mercado $\Pi_j \check{M}_k = \check{M}'_k \in \mu_b^k$ tal que $\Pi_j \neq \Pi_i$. Suponga que el evento calculado por H_j usando \check{M}'_k es igual al evento inactivo del mercado \hat{M}_k , esto es, $H_j(\check{M}'_k, \check{M}_{k+1}, \check{t}_b) = \lambda(t_k) \in InA_{\hat{M}_k}$, y $\Omega_{\mu_b^k}(j, i) \neq \vec{0}$ donde el renglón j es $Act_{\check{M}'_k}$ y la columna i es $InA_{\hat{M}_k}$. Como el mercado M_k no habilita la transición t_j entonces todos los marcados $\Pi_k \check{M}_k \in \mu_b^k$ tales que $\Pi_k \check{M}_k \xrightarrow{t_j}$ (incluyen a \hat{M}_k) se eliminan de μ_b^k , esto es, $\mu_b^k =$

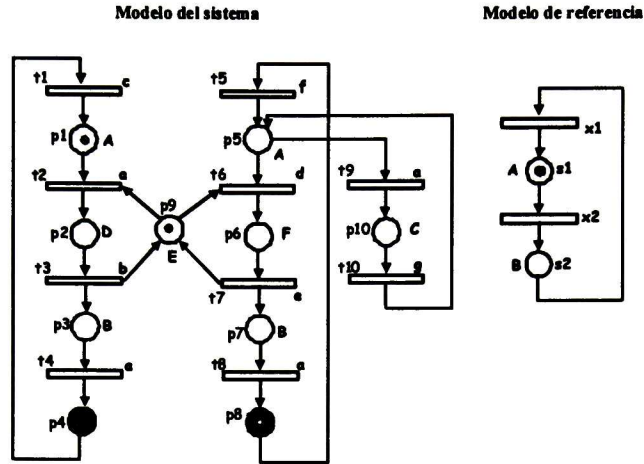


Figura 8.4: Modelo en RdPI del sistema

$\mu_k^b - \{\Pi_k \check{M}_k\}$ así como $\Omega_{\mu_k^b | \Pi_k \check{M}_k}(r, \cdot)$ y $\Omega_{\mu_k^b | \Pi_k \check{M}_k}(\cdot, r)$, donde el renglón y la columna r son $Act_{\mu_k^b | \Pi_k \check{M}_k}$ y $InA_{\mu_k^b | \Pi_k \check{M}_k}$, respectivamente. Así, los eventos calculados por la función H_j usando el posible marcado \hat{M}'_k no contiene eventos inactivos de otro marcado, esto es, $\Omega_{\mu_k^b}(i, \cdot) = \vec{0}$. Por lo tanto, el nuevo marcado del observador es \hat{M}'_k , $\hat{M}_k = \hat{M}'_k$. Como no se disparó la transición t_j , el error es $e_k = M_k - \hat{M}_k$

Entonces, en ambos casos, $|e_{k+1}| \leq |e_k|$.

■

8.3. Ejemplo

Considere la *RdPI* de la figura 8.4. Aplicando el algoritmo 8.4 se obtienen dos posibles marcados iniciales que cumplen con la señal inicial del sistema y cumple con las leyes conservativas de marcado binarias $m_0^1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0]^T$ y $m_0^2 = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ Para cada marcado, se calcula la función lineal y los

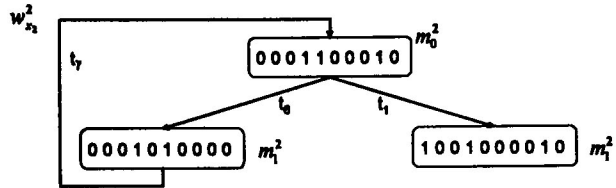


Figura 8.5: Expansión parcial del grafo de alcanzabilidad para el vector de Parikh $w_{x_2}^2$ cuando la transición t_6 y t_1 tienen el mismo símbolo de entrada.

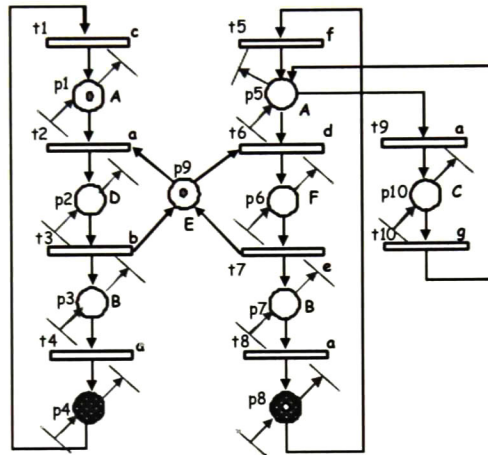


Figura 8.6: Modelo del observador en $RdPI$.

El conjunto de eventos activos e inactivos de la figura 8.4, son los siguientes.

$$\begin{array}{l}
 \Pi_1 \quad \begin{array}{ll}
 \check{M}_0 & \check{M}_1 \\
 Act_{m_0^1} = \{a\} & Act_{m_2^1} = \{a\} \\
 InA_{m_0^1} = \{f\} & InA_{m_2^1} = \{f\}
 \end{array} \\
 \Pi_2 \quad \begin{array}{ll}
 Act_{m_0^2} = \{d\} & Act_{m_2^2} = \{a\} \\
 InA_{m_0^2} = \{c, a\} & InA_{m_2^2} = \{c\}
 \end{array}
 \end{array}$$

El modelo del observador del sistema en $RdPI$ de la figura 8.4 es el de la figura 8.6. Es una copia del modelo del sistema con transiciones extras de entrada y salida

El conjunto de marcados equivalentes de salida de la figura 8.4 es el siguiente

$$\mu_{x_2}^0 = \{m_0^1, m_0^2\} \quad \mu_{x_1}^1 = \{m_2^1, m_2^2\}$$

La matriz de eventos en común, para cada conjunto de marcados equivalentes

$$\Omega_{\mu_{x_2}^0} = \begin{array}{ccc}
 & InA_{m_0^1} & InA_{m_0^2} \\
 Act_{m_0^1} & 0 & 1 \\
 Act_{m_0^2} & 0 & 0
 \end{array}$$

$$\Omega_{\mu_{x_1}^1} = \begin{array}{cc} & \begin{array}{cc} InAm_2^1 & InAm_2^2 \end{array} \\ \begin{array}{c} Act_{m_2^1} \\ Act_{m_2^2} \end{array} & \begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \end{array}$$

Una vez definidas las matrices de eventos en común, se establece el marcado inicial del modelo del observador. El posible marcado inicial del observador es m_0^2 porque el evento activo de ese marcado no es el común con los eventos inactivos del marcado m_0^1 .

$$\Omega_{\mu_{x_2}^0} = \begin{array}{cc} & \begin{array}{cc} InAm_0^1 & InAm_0^2 \end{array} \\ \begin{array}{c} Act_{m_0^1} \\ Act_{m_0^2} \end{array} & \begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array} \end{array}$$

Suponga que el marcado inicial del modelo del observador es $\hat{M}_0 = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]$ y el marcado inicial del modelo del sistema es $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0]^T$

El controlador H_1 emite el evento d , esto es, $H_2(\Pi_2 \check{M}_0, \check{M}_1, x_2) = \lambda(t_6) = d$, como en el sistema no está habilitada la transición t_6 , el observador corrige su marcado, eliminando la segunda columna y reglón de la matriz $\Omega_{\mu_{x_2}^0}$

$$\Omega_{\mu_{x_2}^0} = \begin{array}{cc} & InAm_0^1 \\ Act_{m_0^1} & 0 \end{array}$$

$$\Omega_{\mu_{x_1}^1} = \begin{array}{cc} & InAm_2^1 \\ Act_{m_2^1} & 0 \end{array}$$

El nuevo marcado inicial del observador es m_0^1 .

Capítulo 9

Conclusiones y trabajo a futuro

9.1. Conclusiones

Este trabajo se enfocó en sistemas multiagentes, donde cada agente se implementa con la arquitectura *BDI*, trabajando en ambientes representados por *SED*. La estructura del sistema dinámico es fija y conocida, sin embargo el valor de algunas variables del sistema se desconoce. Este trabajo presentó el uso de las redes de Petri Interpretadas *RdPI* como herramienta formal para diseñar modelos que generen y revisen creencias ambientales del agente. A estos modelos los llamamos genéricamente como observadores, ya que las creencias que generan se interpretan como valores del estado de la *RdPI*. Sin embargo no son observadores clásicos de la teoría de control, ya que el sistema no necesita ser observable para tener una creencia sobre el estado del sistema.

Se propusieron tres modelos diferentes para generar creencias.

El primero trabaja en *SED* evento detectables y se incluye la noción del tiempo como la duración de los eventos. En este caso, las creencias que se generan son sobre el valor del estado y sobre las faltas que ocurren en el sistema. Este modelo está compuesto por dos estructuras, una que genera las creencias sobre el valor del estado y otra que genera creencias sobre las posibles faltas.

El segundo también trabaja con *SED* evento detectables y con tiempo. Pero el modelo está compuesto por una única estructura que genera ambas creencias: sobre el valor del estado y sobre las faltas.

El tercer modelo trabaja en *SED* no evento detectables y sólo genera creencias sobre el valor del estado.

En el primer tipo de modelo se gasta una mayor cantidad de memoria que el segundo, ya que tiene dos estructuras en vez de una. Sin embargo el primer modelo genera creencias sobre las faltas más cercanas a la realidad que el segundo. Ya que por su economía, el segundo modelo, con una sólo estructura, es más lento en su generación de creencias (necesita más eventos para determinar la existencia de una falta).

El tercer tipo ya no es comparable con los anteriores ya que trabaja en *SED* no evento

detectables.

Se demostró que todos los modelos propuestos cumplen con los postulados *AGM* y axiomas de los generadores de creencias. Además, el uso de un sistema algebraico para revisar las creencias nos permitió definir un error el cual es usado para medir la calidad de las creencias.

Adicionalmente, si el *SED* es observable, los generadores de creencias aquí propuestos funcionarán como observadores convencionales de la teoría de control, con la ventaja de que la convergencia al valor real del estado es más rápido.

9.2. Trabajo a futuro

Las áreas a considerar para extender esta investigación son

- Extender el enfoque distribuido para que cada agente tenga su propio modelo del observador y del modelo de control por regulación representando el conocimiento parcial del ambiente así como sus tareas y acciones en específico.
- Adaptar protocolos de comunicación entre los agentes
- Adaptar las métricas de cohesión y acoplamiento para evitar bloqueos.

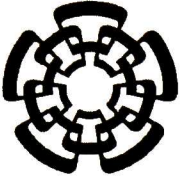
Bibliografía

- [1] M. Alcaraz-Mejía, E. López-Mellado, A. Ramírez-Treviño, I. Rivera-Rangel. Petri Net Based Fault Diagnosis of Discrete Event Systems. *Proceedings of the IEEE Conference on Systems, Man & Cybernetics*, pp. 47
- [2] C. E. Alchourrón, P. Gardemfors, and D. Makinson. On the logic of theory change: partial meet functions for contractions and revision. *Journal of Symbolic Logic*, 50:510-530, 1985.
- [3] L. Ardissono, G. Boella, L. Lesmo: "A Plan-based Formalism to Express Knowledge about Actions", *Proc. 4th Workshop Modelage*, Pontignano (It), 1997, 255-268.
- [4] Weiming Shen, Douglas H. Norrie and Jean Paul A. Barthés. *Multi-agent Systems for concurrent intelligent design and manufacturing*. Ed. Taylor & Francis, 2001.
- [5] C. Boutilier. Unifying default reasoning and belief revision in a modal framework. *Artificial Intelligence*, 68:33-85, 1994.
- [6] R. Davis, H. Shrobe, and P. Szolovits. What is a Knowledge Representation? *AI Magazine*, 14(1):17-33, 1993.
- [7] Jacques Ferber. *Multi-Agent Systems. An introduction to Distributed Artificial Intelligence*. Ed. Addison-Wesley, Great Britian, 1995.
- [8] Klaus Fischer, Jorg P. Muller and Markus Pischel. A pragmatic BDI Architecture. In M. Wooldridge, J.P. Muller, and M. Tambe, editors, *Intelligent Agents II* (LNAI volume 1037), pag. 33-48. Springer-Verlag: Berlin, Germany, 1996.
- [9] M. E. Bratman, *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, MA 1987.
- [10] M. E. Bratman, D. J. Israel and M. E. Pollack. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence*, 4:349-355, 1988.
- [11] P. R. Cohen and H. J. Levesque. Persistence, intention and commitment. In M.P. Georgeff and A. L. Lansky, editors, *Proceedings of the 1986 workshop on Reasoning about Actions and Plans*, pages 297-340. Morgan Kaufmann Publishers, San Mateo, CA, 1987.

- [12] J. Desel and J. Esparza "Free choice Petri Nets." Cambridge University Press, 1995.
- [13] Stan Franklin and Art Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.
- [14] Nir Friedman, Joseph Y. Halpern. Belief Revision: A Critique. *Journal of Logic, Language and Information*, vol. 8, 1999, pp. 401-420.
- [15] M. Freund and D. Lehmann. Belief revision and rational inference. *Technical Report TR 94-16*, Institute of Computer Science, The Hebrew University of Jerusalem, Israel, 1994.
- [16] J. Harland and M. Winikoff. Language Design Issues for Agents based on Linear Logic. In *Proceedings of the Workshop on Computational Logic in Multi-Agent, Systems 3rd International Workshop, CLIMA'02*, Copenhagen, Denmark, 2002.
- [17] H. Katsuno and A. O. Mendelzon. A unified view of propositional knowledge base updates, in *Proceedings IJCAI-89, Detroit, MI (1989)* 1413-1419.
- [18] D. Lehmann. Belief revision, revised. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence (IJCAL'95)*, pages 1534-1540, 1995.
- [19] Sven Ove Hansson and Renata Wassermann. Local change: A preliminar report. In *Fourth Symposium on Logical Formalizations of Commonsense Reasoning*, London, 1998.
- [20] Rohit Parikh. Beliefs, belief revision and splitting languages. In *Logic, Language, and Computation*, Volume 2, 1999, pp. 266-268.
- [21] A.S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI-Architecture. *Artificial Intelligence* 42:213-261, 1990.
- [22] Anand S. Rao. Decision Procedures for propositional linear-time belief-desire-intention logics. In M. Wooldridge, J.P. Muller, and M. Tambe, editors, *Intelligent Agents II (LNAI volume 1037)*, pag. 33-48. Springer-Verlag: Berlin, Germany, 1996.
- [23] A. S. Rao. Means-end plan recognition-towards theory of reactive recognition. In *Proc. 5th Int. Conf. Principles of Knowledge Representation and Reasoning*, page 497-508, 1994.
- [24] Renata Wassermann. On structured belief bases. *Seventh International Workshop on Nonmonotonic Reasoning*, Trento 1998. To appear in Hans Rott and Mary-Anne Williams, editors, *Frontiers in Belief Revision*, Kluwer.
- [25] I. Rivera-Rangel, L. Aguirre-Salas, A. Ramirez-Treviño y E. López-Mellado. "Observer Design for Discrete Event Systems Modeled by Interpreted Petri Nets" *Proc. of 39th IEEE Conference on Decision and Control*, pp 2260-2265, 2000.

- [26] A. Ramirez-Treviño, I. Rivera-Rangel y E. López-Mellado. "Observability of Discrete Event Systems Modeled by Interpreted Petri Nets". *IEEE Transactions on Robotics and Automation*, 19(4), pp 557-565, august 2003.
- [27] Samir Chopra and Rohit Parikh. An inconsistency tolerant model for belief representation and belief revision. *Proceedings of IJCAL 99*, Morgan Kaufmann. A revised version appeared in *Annals of Mathematics and Artificial Intelligence*, vol. 8(1-4), pages 259-285, 2000.
- [28] J. Y. Halpern and Y. O. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319, 1992.
- [29] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages and computation*. Addison-Wesley. 1979.
- [30] I. Jiménez-Ochoa, O. Begovich, A. Ramírez-Treviño, L.I. Aguirre-Salas. Implementing BDI Agents using Petri Nets. *Proc. of the IEEE International Conference Systems, Man and Cybernetics*, October 2003.
- [31] Italia Jiménez y Antonio Ramírez Treviño. A Polynomial Algorithm for Observer Design with Interpreted Petri Nets. World Automation Congress, WAC 2012.
- [32] M. E. Meda and A. Ramírez. Identification in discrete event systems. *Proc. of the IEEE International Conference Systems, Man and Cybernetics*, October 1998.
- [33] A. Ramirez-Treviño, I. Rivera-Rangel and E. Lopez-Mellado. Observer Design for Discrete Event Systems Modeled by Interpreted Petri Nets. *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, USA. 2000.
- [34] Israel Rivera-Rangel, Antonio Ramírez-Treviño, Luis I. Aguirre-Salas and Javier Ruíz-León. Geometrical Characterization of Observability in Interpreted Petri Nets. *Kybernetika*, Vol. 41, No. 5, Página 553-574, 2005.
- [35] A. Ramírez-Treviño, E. Ruíz-Beltrán, I. Rivera-Rangel, and E. López-Mellado. Online Fault Diagnosis of Discrete Event Systems. A Petri Net Based Approach. *Transactions on Automation Science and Engineering*, Vol. 4, no. 1. pp. 31-39, 2007.
- [36] Israel Rivera-Rangel, Antonio Ramírez-Treviño, Luis I. Aguirre-Salas and Javier Ruíz-León. Geometrical Characterization of Observability in Interpreted Petri Nets. *Kybernetika*, Vol. 41, No. 5, Página 553-574, 2005.
- [37] I. Rivera-Rangel, A. Ramírez-Treviño, E. López-Mellado. Observability of Petri Net Reduced Models of Discrete Event Systems. *ETFA '03*.
- [38] E. Ruiz-Beltrán, I. Jiménez-Ochoa, A. Ramírez-Treviño, E. López-Mellado, M. Meda-Campaña. Fault detection and location in DES modeled using Petri Nets. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, pp. 1645-1650, 2005.

- [39] G. Ramirez-Prado, A. Santoyo, A. Ramirez-Trevino, and O. Begovich. Regulation problem in discrete event systems using interpreted petri nets. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 3, pages 2174-2179 vol.3, October 2000.
- [40] A. Santoyo, I. Jiménez-Ochoa and A. Ramírez. "A complete cycle for controller design in Discrete Event System", *IEEE Proc. SMC*, pp. 2688-2693, 2001.
- [41] Stuart Russell, Peter Norving. *Artificial Intelligence a Modern Approach*. Ed. Pearson education. 2da. edición, India, 2003.
- [42] Shen, W. y Norre, D.H. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems, an International Journal*, 1(2):129-156, 1999.
- [43] M. Silva. *Petri nets on automatica and data processing*. Madrid, Spain: AC Ed., 1985, [in Spanish].
- [44] G. Weiss. *Multiagent System, A Modern approach to Distributed Artificial Intelligence*. Cambridge, Massachusetts. London, England 2000.
- [45] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Ed. John Wiley & Sons, LTD, England, 2002.
- [46] P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81-98, 1989. 0018-9219.



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

Representación y Manipulación Algebraica de las Creencias, Deseos
e Intenciones de un Agente

del (la) C.

María Italia JIMÉNEZ OCHOA

el día 15 de Diciembre de 2014.

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

Dr. José Javier Ruíz León
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

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

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Carlos Renato Vazquez Topete
Profesor Investigador
Centro Universitario de los Valles,
UdG



CINVESTAV - IPN
Biblioteca Central



SSIT0012930