

Enrique Castillo,
José Manuel Gutiérrez, y
Ali S. Hadi

Sistemas Expertos y Modelos de Redes Probabilísticas

Con más de 150 ejemplos y 250 figuras

Enrique Castillo
Universidad de Cantabria
39005 Santander, España
E-mail: castie@ccaix3.unican.es

José Manuel Gutiérrez
Universidad de Cantabria
39005 Santander, España
E-mail: gutierjm@ccaix3.unican.es

Ali S. Hadi
Universidad de Cornell
358 Ives Hall
Ithaca, NY 14853-3901, USA
E-mail: ali-hadi@cornell.edu

A todo el pueblo de la desaparecida Yugoslavia con la esperanza de que vivan juntos en paz y sean amigos, como lo son los autores de este libro, a pesar de sus diferencias en religiones, lenguas y nacionalidades.

— This is page vi
— Printer: Opaque this

Prefacio

En las dos últimas décadas se ha producido un notable desarrollo en el área de la inteligencia artificial y, en particular, en la de los sistemas expertos. Debido a su carácter multidisciplinar, muchos de los resultados obtenidos en esta disciplina han sido publicados en diversas revistas de numerosos campos: ciencias de la computación, ingeniería, matemáticas, estadística, etc. Este libro trata de reunir, organizar y presentar estos resultados de forma clara y progresiva. Se ha tratado de mantener la información actualizada al máximo, de tal forma que incluso algunos de los conceptos presentados en el libro no han sido publicados previamente (por ejemplo, algunos resultados de los Capítulos 7, 11 y 12).

Este libro está destinado a estudiantes e investigadores de áreas teóricas y aplicadas de disciplinas tales como ciencias de la computación, ingeniería, medicina, matemáticas, economía y ciencias sociales. Dado este carácter multidisciplinar, se han intentado mantener al mínimo los conocimientos previos necesarios para leer este libro. Así, sólo se requieren algunas nociones básicas de estadística y probabilidad y estar familiarizado con los conceptos elementales del álgebra lineal (ver, por ejemplo, Hadi (1996)). En algunas ocasiones los conceptos se ilustran utilizando algunos programas de *Mathematica*. Para un completo entendimiento de estos programas, se requiere cierto conocimiento del programa *Mathematica* (ver Castillo y otros (1993)).

Este libro puede ser utilizado como libro de consulta, o como libro de texto en cursos de postgrado o en últimos cursos de carrera. El libro contiene numerosos ejemplos ilustrativos y ejercicios al final de cada capítulo. También se han desarrollado varios programas que implementan los algoritmos y metodologías presentadas. Estos programas, junto con los manuales de usuario correspondientes, pueden obtenerse de la dirección World Wide Web (WWW) <http://ccaix3.unican.es/~AIGroup>. Creemos que pueden ayudar a los lectores a entender los conceptos presentados y a aplicar esta metodología a sus respectivos ámbitos profesionales y de estudio. Por ejemplo, estos programas han sido utilizados para resolver algunos de los ejemplos y ejercicios del libro, así como para analizar varios ejemplos prácticos reales (Capítulo 12). Finalmente, al final del libro se incluye una extensa bibliografía para consultas adicionales.

Aunque en el libro se presentan tanto la teoría como las aplicaciones prácticas de esta disciplina, se ha puesto un interés especial en las aplicaciones prácticas. Por eso, muchos de los teoremas presentados se incluyen sin demostración, refiriéndose a la bibliografía para aquellos lectores interesados. Así mismo, se introducen numerosos ejemplos para ilustrar cada uno de los conceptos presentados.

Este libro está organizado de la forma siguiente. El Capítulo 1 presenta una introducción al área de la inteligencia artificial y los sistemas expertos que, entre otras cosas, analiza algunos ejemplos ilustrativos, describe los componentes de un sistema experto, así como las etapas necesarias para su desarrollo, y analiza la relación de los sistemas expertos con otras áreas de la inteligencia artificial. Los Capítulos 2 y 3 describen los dos tipos principales de sistemas expertos: los sistemas expertos basados en reglas y los basados en probabilidad. Aunque estos dos tipos de sistemas se introducen de forma separada, en el Capítulo 3 se muestra que los sistemas expertos basados en reglas se pueden considerar como un tipo particular de sistemas expertos probabilísticos. También se muestra que dos de las componentes más complejas e importantes de un sistema experto son el “subsistema de control de coherencia” y el “motor de inferencia” y estos son, quizás, los dos componentes más débiles de los sistemas expertos desarrollados hasta la fecha. En los Capítulos 5–10 se muestra la forma de implementar estos componentes de forma eficiente.

A partir del Capítulo 5 se requieren algunos conceptos de la teoría de grafos, ya que éstos serán la base para construir las redes probabilísticas. Los conceptos necesarios en este libro, que son un requisito básico para entender los capítulos siguientes, se presentan en el Capítulo 4. Los Capítulos 5–7 analizan el problema de la construcción de modelos probabilísticos, desde varias perspectivas. En particular, los conceptos de dependencia e independencia condicional, necesarios para definir la estructura de las redes probabilísticas, se introducen y analizan con detalle en el Capítulo 5. El Capítulo 6 presenta los dos modelos más importantes de redes probabilísticas, las redes de Markov y las redes Bayesianas, que se definen a partir de una estructura gráfica no dirigida y dirigida, respectivamente. El Capítulo 7 presenta algunas extensiones de los modelos gráficos para definir modelos probabilísticos más generales a partir de multigrafos, conjuntos de relaciones de independencia condicional, modelos multifactorizados y modelos definidos condicionalmente.

Los Capítulos 8 y 9 presentan los métodos de propagación exacta y aproximada más importantes, respectivamente. El Capítulo 10 analiza la propagación simbólica que es uno de los avances más recientes de la propagación en redes probabilísticas. El Capítulo 11 está dedicado al problema del aprendizaje; en concreto, al problema del aprendizaje de redes Bayesianas a partir de un conjunto de datos (una base de datos, etc.). Finalmente, el Capítulo 12 ilustra la aplicación de todos los conceptos presentados en el libro mediante su aplicación a ejemplos reales.

Muchos de nuestros colegas y estudiantes han leído versiones anteriores de este libro y nos han proporcionado muchas sugerencias que han ayudado a mejorar notablemente distintas partes del mismo. En particular, agradecemos la inestimable ayuda de (en orden alfabético): Noha Adly, Remco Bouckaert, Federico Ceballos, Jong Wang Chow, Javier Díez, Dan

Geiger, Joseph Halpern, Judea Pearl, Julius Reiner, José María Sarabia,
Milan Studený, y Jana Zvárová.

Enrique Castillo
Jose Manuel Gutiérrez
Ali S. Hadi

— This is page x
— Printer: Opaque this

Tabla de Contenidos

1	Introducción	1
1.1	Introducción	1
1.2	¿Qué es un Sistema Experto?	2
1.3	Ejemplos Ilustrativos	4
1.4	¿Por Qué los Sistemas Expertos?	7
1.5	Tipos de Sistemas Expertos	8
1.6	Componentes de un Sistema Experto	9
1.7	Desarrollo de un Sistema Experto	15
1.8	Otras Áreas de la IA	16
1.9	Conclusiones	21
2	Sistemas Basados en Reglas	23
2.1	Introducción	23
2.2	La Base de Conocimiento	24
2.3	El Motor de Inferencia	30
2.4	Control de la Coherencia	51
2.5	Explicando Conclusiones	59
2.6	Ejemplo de Aplicación	59
2.7	Introduciendo Incertidumbre	64
	Ejercicios	65
3	Sistemas Expertos Basados en Probabilidad	69
3.1	Introducción	69
3.2	Algunos Conceptos Básicos de la Teoría de la Probabilidad	71
3.3	Reglas Generalizadas	85
3.4	Introduciendo los Sistemas Expertos Basados en Probabilidad	87
3.5	La Base de Conocimiento	92
3.6	El Motor de Inferencia	104
3.7	Control de la Coherencia	106
3.8	Comparando los dos Tipos de Sistemas Expertos	108
	Ejercicios	109

4	Algunos Conceptos sobre Grafos	115
4.1	Introducción	115
4.2	Conceptos Básicos y Definiciones	116
4.3	Características de los Grafos no Dirigidos	120
4.4	Características de los Grafos Dirigidos	124
4.5	Grafos Triangulados	131
4.6	Grafos de Aglomerados	142
4.7	Representación de Grafos	148
4.8	Algunos Algoritmos para Grafos	162
	Ejercicios	175
5	Construcción de Modelos Probabilísticos	179
5.1	Introducción	179
5.2	Criterios de Separación Gráfica	181
5.3	Algunas Propiedades de la Independencia Condicional	188
5.4	Modelos de Dependencia	197
5.5	Factorizaciones de una Función de Probabilidad	199
5.6	Construcción de un Modelo Probabilístico	206
	Apéndice al Capítulo 5	211
	Ejercicios	213
6	Modelos Definidos Gráficamente	217
6.1	Introducción	217
6.2	Algunas Definiciones y Problemas	219
6.3	Modelos de Dependencia Gráficos no Dirigidos	225
6.4	Modelos de Dependencia en Gráficos Dirigidos	243
6.5	Modelos Gráficos Equivalentes	262
6.6	Expresividad de los Modelos Gráficos	269
	Ejercicios	272
7	Extensiones de los Modelos Gráficos	277
7.1	Introducción	277
7.2	Modelos Definidos por Multigrafos	279
7.3	Modelos Definidos por Listas de Independencias	286
7.4	Modelos probabilísticos Multifactorizados	290
7.5	Modelos Multinomiales Multifactorizados	291
7.6	Modelos Normales Multifactorizados	304
7.7	Modelos probabilísticos definidos Condicionalmente	311
	Ejercicios	326
8	Propagación Exacta en Redes Probabilísticas	331
8.1	Introducción	331
8.2	Propagación de Evidencia	332
8.3	Propagación en Poliárboles	336
8.4	Propagación en Redes Múltiplemente Conexas	356

8.5	Método de Condicionamiento	358
8.6	Métodos de Agrupamiento	367
8.7	Propagación en Árboles de Conglomerados	383
8.8	Propagación Orientada a un Objetivo	395
8.9	Propagación Exacta en Redes Bayesianas Gaussianas	400
	Ejercicios	405
9	Métodos de Propagación Aproximada	411
9.1	Introducción	411
9.2	Base Intuitiva de los Métodos de Simulación	412
9.3	Metodología General para los Métodos de Simulación	418
9.4	El Método de Aceptación-Rechazo	425
9.5	Método del Muestreo Uniforme	429
9.6	El Método de la Función de Verosimilitud Pesante	430
9.7	El Muestreo Hacia Adelante y Hacia Atrás	432
9.8	Método de Muestreo de Markov	435
9.9	Método del Muestreo Sistemático	438
9.10	Método de Búsqueda de la Máxima Probabilidad	450
9.11	Análisis de Complejidad	460
	Ejercicios	460
10	Propagación Simbólica de Evidencia	463
10.1	Introducción	463
10.2	Notación y Conceptos Preliminares	465
10.3	Generación Automática de Código Simbólico	467
10.4	Estructura Algebraica de las Probabilidades	474
10.5	Propagación Simbólica Mediante Métodos Numéricos	475
10.6	Propagación Simbólica Orientada a un Objetivo	485
10.7	Tratamiento Simbólico de la Evidencia Aleatoria	491
10.8	Análisis de Sensibilidad	493
10.9	Propagación Simbólica en Redes Bayesianas Normales	496
	Ejercicios	500
11	Aprendizaje en Redes Bayesianas	503
11.1	Introducción	503
11.2	Midiendo la Calidad de una Red Bayesiana	506
11.3	Medidas de Calidad Bayesianas	509
11.4	Medidas Bayesianas para Redes Multinomiales	513
11.5	Medidas Bayesianas para Redes Multinormales	522
11.6	Medidas de Mínimo Requerimiento Descriptivo	529
11.7	Medidas de Información	532
11.8	Análisis Posterior de las Medidas de Calidad	533
11.9	Algoritmos de Búsqueda de Redes Bayesianas	534
11.10	El Caso de Datos Incompletos	536
	Apéndice al Capítulo 11: Estadística Bayesiana	538

Ejercicios	548
12 Ejemplos de Aplicación	551
12.1 Introducción	551
12.2 El Sistema del Tanque de Presión	552
12.3 Sistema de Distribución de Energía	565
12.4 Daño en Vigas de Hormigón Armado	572
12.5 Daño en Vigas de Hormigón Armado: El Modelo Normal . .	585
Ejercicios	590
Notación	595
Referencias	603
Índice	619

Capítulo 1

Introducción

1.1 Introducción

No hace mucho tiempo, se creía que algunos problemas como la demostración de teoremas, el reconocimiento de la voz y el de patrones, ciertos juegos (como el ajedrez o las damas), y sistemas altamente complejos de tipo determinista o estocástico, debían ser resueltos por personas, dado que su formulación y resolución requieren ciertas habilidades que sólo se encuentran en los seres humanos (por ejemplo, la habilidad de pensar, observar, memorizar, aprender, ver, oler, etc.). Sin embargo, el trabajo realizado en las tres últimas décadas por investigadores procedentes de varios campos, muestra que muchos de estos problemas pueden ser formulados y resueltos por máquinas.

El amplio campo que se conoce como *inteligencia artificial* (IA) trata de estos problemas, que en un principio parecían imposibles, intratables y difíciles de formular utilizando ordenadores. A. Barr y E. A. Feigenbaum, dos de los pioneros de la investigación en IA, definen ésta como sigue: (véase Barr y Feigenbaum (1981), página 4):

La Inteligencia Artificial es la parte de la Ciencia que se ocupa del diseño de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la

comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc.

Hoy en día, el campo de la IA engloba varias subáreas tales como los sistemas expertos, la demostración automática de teoremas, el juego automático, el reconocimiento de la voz y de patrones, el procesamiento del lenguaje natural, la visión artificial, la robótica, las redes neuronales, etc. Este libro está dedicado a los *sistemas expertos*. Aunque los sistemas expertos constituyen una de las áreas de investigación en el campo de la IA, la mayor parte de las restantes áreas, si no todas, disponen de una componente de sistemas expertos formando parte de ellas.

Este capítulo presenta una introducción a los sistemas expertos. Se comienza con algunas definiciones de sistemas expertos en la Sección 1.2. La Sección 1.3 da algunos ejemplos que sirven para motivar los sistemas expertos en varios campos de aplicación. Estos ejemplos muestran la importancia y la amplia aplicabilidad de los sistemas expertos en la práctica. Algunas de las razones para utilizar los sistemas expertos se indican en la Sección 1.4. Los principales tipos de sistemas expertos se presentan en la Sección 1.5. La Sección 1.6 discute y analiza la estructura de los sistemas expertos y sus principales componentes. Las diferentes etapas necesarias para el diseño, desarrollo e implementación de los sistemas expertos se analizan en la Sección 1.7. Finalmente, la Sección 1.8 se dedica a mencionar algunas de las restantes áreas de investigación de la IA y suministran al lector interesado algunas de las referencias más importantes, revistas, y direcciones de acceso (WWW).

1.2 ¿Qué es un Sistema Experto?

En la literatura existente se pueden encontrar muchas definiciones de sistema experto. Por ejemplo, Stevens (1984), página 40, da la definición siguiente:

Los sistemas expertos son máquinas que piensan y razonan como un experto lo haría en una cierta especialidad o campo. Por ejemplo, un sistema experto en diagnóstico médico requeriría como datos los síntomas del paciente, los resultados de análisis clínicos y otros hechos relevantes, y, utilizando éstos, buscaría en una base de datos la información necesaria para poder identificar la correspondiente enfermedad. [...] Un Sistema Experto de verdad, no sólo realiza las funciones tradicionales de manejar grandes cantidades de datos, sino que también manipula esos datos de forma tal que el resultado sea inteligible y tenga significado para responder a preguntas incluso no completamente especificadas.

Aunque la anterior es todavía una definición razonable de un sistema experto, han surgido desde entonces otras definiciones, debido al rápido desarrollo de la tecnología (ver, por ejemplo, Castillo y Álvarez (1991) y Durkin (1994)). El sentido de estas definiciones puede resumirse como sigue:

Definición 1.1 Sistema Experto. *Un sistema experto puede definirse como un sistema informático (hardware y software) que simula a los expertos humanos en un área de especialización dada.*

Como tal, un sistema experto debería ser capaz de procesar y memorizar información, aprender y razonar en situaciones deterministas e inciertas, comunicar con los hombres y/u otros sistemas expertos, tomar decisiones apropiadas, y explicar por qué se han tomado tales decisiones. Se puede pensar también en un sistema experto como un *consultor* que puede suministrar ayuda a (o en algunos casos sustituir completamente) los expertos humanos con un grado razonable de fiabilidad.

Durante la última década se han desarrollado muy rápidamente numerosas aplicaciones de sistemas expertos a muchos campos (ver, por ejemplo, Quinlan (1987, 1989)). Durkin (1994) examina unos 2,500 sistemas expertos y los clasifica por criterios, tales como áreas de aplicación, tareas realizadas, etc. Tal como puede verse en la Figura 1.1, la economía, la industria y la medicina continúan siendo los campos dominantes entre aquellos en los que se utilizan los sistemas expertos. La sección siguiente muestra algunos ejemplos que motivan la aplicación de los sistemas expertos en algunos de estos campos.

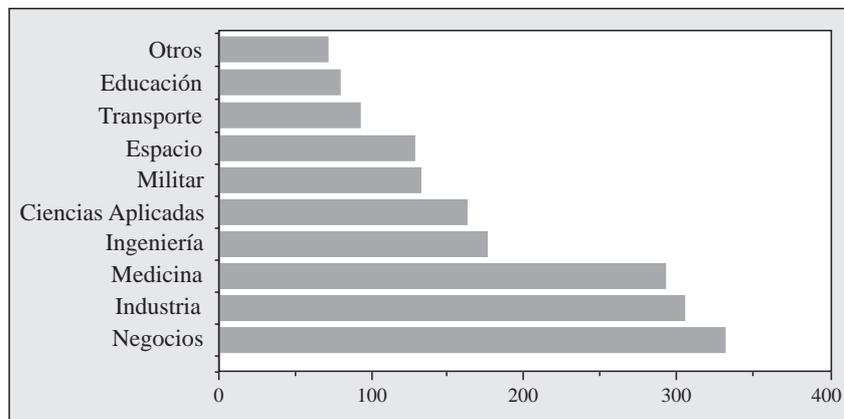


FIGURA 1.1. Campos de aplicación de los sistemas expertos. Adaptado de Durkin (1994) y Castillo, Gutiérrez, y Hadi (1995a).

1.3 Ejemplos Ilustrativos

Los sistemas expertos tienen muchas aplicaciones. En esta sección se dan unos pocos ejemplos ilustrativos del tipo de problemas que pueden resolverse mediante sistemas expertos. Otros ejemplos prácticos se dan a lo largo del libro.

Ejemplo 1.1 Transacciones bancarias. No hace mucho, para hacer una transacción bancaria, tal como depositar o sacar dinero de una cuenta, uno tenía que visitar el banco en horas de oficina. Hoy en día, esas y otras muchas transacciones pueden realizarse en cualquier momento del día o de la noche usando los cajeros automáticos que son ejemplos sencillos de sistemas expertos. De hecho, se pueden realizar estas transacciones desde casa comunicándose con el sistema experto mediante la línea telefónica. ■

Ejemplo 1.2 Control de tráfico. El control de tráfico es una de las aplicaciones más importantes de los sistemas expertos. No hace mucho tiempo, el flujo de tráfico en las calles de una ciudad se controlaba mediante guardias de tráfico que controlaban el mismo en las intersecciones. Hoy se utilizan sistemas expertos que operan automáticamente los semáforos y regulan el flujo del tráfico en las calles de una ciudad y en los ferrocarriles. En la Sección 2.6 y en los ejercicios del Capítulo 2 se dan ejemplos de estos sistemas. ■

Ejemplo 1.3 Problemas de planificación. Los sistemas expertos pueden utilizarse también para resolver problemas complicados de planificación de forma que se optimicen ciertos objetivos como, por ejemplo, la organización y asignación de aulas para la realización de exámenes finales en una gran universidad, de forma tal que se logren los objetivos siguientes:

- Eliminar las coincidencias de asignación simultánea de aulas: Sólo se puede relizar un examen en cada aula al mismo tiempo.
- Asientos suficientes: Un aula asignada para un examen debe tener al menos dos asientos por estudiante.
- Minimizar los conflictos temporales: Minimizar el número de alumnos que tienen exámenes coincidentes.
- Eliminar la sobrecarga de trabajo: Ningún alumno debe tener más de dos exámenes en un periodo de 24 horas.
- Minimizar el número de exámenes realizados durante las tardes.

Otros ejemplos de problemas de planificación que pueden ser resueltos mediante sistemas expertos son la planificación de doctores y enfermeras en un gran hospital, la planificación en una gran factoría, y la planificación de autobuses para las horas de congestión o de días festivos. ■

Ejemplo 1.4 Diagnóstico médico. Una de las aplicaciones más importantes de los sistemas expertos tiene lugar en el campo médico, donde éstos pueden ser utilizados para contestar a las siguientes preguntas:

1. ¿Cómo se puede recoger, organizar, almacenar, poner al día y recuperar la información médica (por ejemplo, registros de pacientes) de una forma eficiente y rápida? Por ejemplo, supóngase que un doctor en un centro médico está interesado en conocer información sobre una cierta enfermedad (E) y tres síntomas asociados (S_1 , S_2 , y S_3). Se puede utilizar un sistema experto para buscar en la base de datos, extraer y organizar la información deseada. Esta información puede resumirse en tablas tales como la dada en la Tabla 1.1 o en gráficos como el de la Figura 1.2.
2. ¿Cómo se puede aprender de la experiencia? Es decir, cómo se actualiza el conocimiento de los doctores en medicina cuando el número de pacientes que éstos tratan aumenta?
3. Supuesto que un paciente presenta un conjunto de síntomas, ¿cómo se decide qué enfermedad es la que más probablemente tiene el paciente?
4. ¿Cuáles son las relaciones entre un conjunto (normalmente no observable) de enfermedades y un conjunto (observable) de síntomas? En otras palabras, ¿qué modelos pueden utilizarse para describir las relaciones entre los síntomas y las enfermedades?
5. Dado que el conjunto de síntomas conocidos no es suficiente para diagnosticar la enfermedad con cierto grado de certeza, ¿qué información adicional debe ser obtenida (por ejemplo, ¿qué síntomas adicionales deben ser identificados? o ¿qué pruebas médicas deben realizarse?).
6. ¿Cuál es el valor de cada una de éstas piezas de información? En otras palabras, ¿cuál es la contribución de cada uno de los síntomas adicionales o pruebas a la toma de decisión? ■

Ejemplo 1.5 Agentes secretos. Alberto, Luisa, Carmen, y Tomás son agentes secretos, cada uno está en uno de los cuatro países: Egipto, Francia, Japón y España. No se sabe dónde está cada uno de ellos. Por tanto, se ha pedido información y se han recibido los cuatro telegramas siguientes:

- Desde Francia: Luisa está en España.
- Desde España: Alberto está en Francia.
- Desde Egipto: Carmen está en Egipto.
- Desde Japón: Carmen está en Francia.

No se sabe quién es el que ha mandado cada uno de los mensajes, pero se sabe que Tomás miente (¿un agente doble?) y que los demás agentes dicen la verdad.

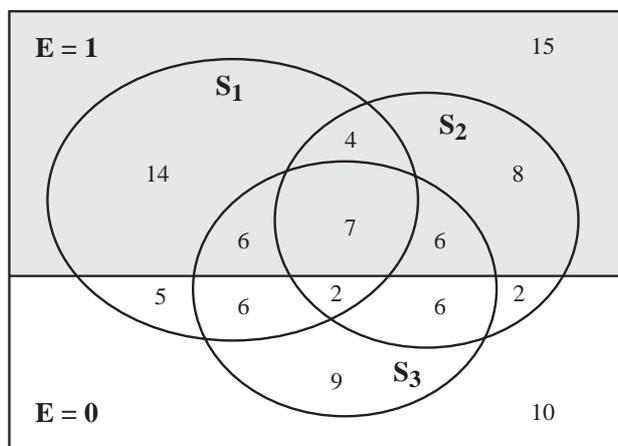


FIGURA 1.2. Una representación gráfica de la distribución de frecuencias de una enfermedad (D) y tres síntomas binarios (S_1 , S_2 , y S_3) en una base de datos médica.

E	S_1	S_2	S_3	Frecuencia
1	1	1	1	7
1	1	1	0	4
1	1	0	1	6
1	1	0	0	14
1	0	1	1	6
1	0	1	0	8
1	0	0	1	0
1	0	0	0	15
0	1	1	1	2
0	1	1	0	0
0	1	0	1	6
0	1	0	0	5
0	0	1	1	6
0	0	1	0	2
0	0	0	1	9
0	0	0	0	10

TABLA 1.1. Una representación tabular de la distribución de frecuencias de una enfermedad (D) y tres síntomas binarios (S_1 , S_2 , y S_3) en una base de datos médica (1 representa la presencia y 0 representa la ausencia de la enfermedad o el síntoma indicado).

La cuestión que se desea responder es: ¿Qué agente está en cada país? Aunque se trata de un problema de lógica, que contiene afirmaciones muy simples, su solución no es fácil de obtener a simple vista. En la Sección 2.4.2 se muestra la forma de resolver automáticamente este problema utilizando un conjunto de reglas. ■

1.4 ¿Por Qué los Sistemas Expertos?

El desarrollo o la adquisición de un sistema experto es generalmente caro, pero el mantenimiento y el coste marginal de su uso repetido es relativamente bajo. Por otra parte, la ganancia en términos monetarios, tiempo, y precisión resultantes del uso de los sistemas expertos son muy altas, y la amortización es muy rápida. Sin embargo, antes de desarrollar o adquirir un sistema experto debe realizarse un análisis de factibilidad y de coste-beneficio. Hay varias razones para utilizar sistemas expertos. Las más importantes son:

1. Con la ayuda de un sistema experto, personal con poca experiencia puede resolver problemas que requieren un conocimiento de experto. Esto es también importante en casos en los que hay pocos expertos humanos. Además, el número de personas con acceso al conocimiento aumenta con el uso de sistemas expertos.
2. El conocimiento de varios expertos humanos puede combinarse, lo que da lugar a sistemas expertos más fiables, ya que se obtiene un sistema experto que combina la sabiduría colectiva de varios expertos humanos en lugar de la de uno solo.
3. Los sistemas expertos pueden responder a preguntas y resolver problemas mucho más rápidamente que un experto humano. Por ello, los sistemas son muy valiosos en casos en los que el tiempo de respuesta es crítico.
4. En algunos casos, la complejidad del problema impide al experto humano resolverlo. En otros casos la solución de los expertos humanos no es fiable. Debido a la capacidad de los ordenadores de procesar un elevadísimo número de operaciones complejas de forma rápida y aproximada, los sistemas expertos suministran respuestas rápidas y fiables en situaciones en las que los expertos humanos no pueden.
5. Los sistemas expertos pueden ser utilizados para realizar operaciones monótonas, aburridas e incómodas para los humanos. En verdad, los sistemas expertos pueden ser la única solución viable en una situación en la que la tarea a realizar desborda al ser humano (por ejemplo, un avión o una cápsula espacial dirigida por un sistema experto).

6. Se pueden obtener enormes ahorros mediante el uso de sistemas expertos.

El uso de los sistemas expertos se recomienda especialmente en las situaciones siguientes:

- Cuando el conocimiento es difícil de adquirir o se basa en reglas que sólo pueden ser aprendidas de la experiencia.
- Cuando la mejora continua del conocimiento es esencial y/o cuando el problema está sujeto a reglas o códigos cambiantes.
- Cuando los expertos humanos son caros o difíciles de encontrar.
- Cuando el conocimiento de los usuarios sobre el tema es limitado.

1.5 Tipos de Sistemas Expertos

Los problemas con los que pueden tratar los sistemas expertos pueden clasificarse en dos tipos: problemas esencialmente deterministas y problemas esencialmente estocásticos. Por ejemplo, aunque el ejemplo 1.1 (transacciones bancarias) y el Ejemplo 1.2 (control de tráfico) pueden contener algunos elementos de incertidumbre, son esencialmente problemas deterministas. Por otra parte, en el campo médico (ver Ejemplo 1.4) las relaciones entre síntomas y enfermedades se conocen sólo con un cierto grado de certeza (la presencia de un conjunto de síntomas no siempre implica la presencia de una enfermedad). Estos tipos de problemas pueden también incluir algunos elementos deterministas, pero se trata fundamentalmente de problemas estocásticos.

Consecuentemente, los sistemas expertos pueden clasificarse en dos tipos principales según la naturaleza de problemas para los que están diseñados: deterministas y estocásticos.

Los problemas de tipo determinista pueden ser formulados usando un conjunto de reglas que relacionen varios objetos bien definidos. Los sistemas expertos que tratan problemas deterministas son conocidos como *sistemas basados en reglas*, porque sacan sus conclusiones basándose en un conjunto de reglas utilizando un mecanismo de *razonamiento lógico*. El Capítulo 2 se dedica a los sistemas expertos basados en reglas.

En situaciones inciertas, es necesario introducir algunos medios para tratar la incertidumbre. Por ejemplo, algunos sistemas expertos usan la misma estructura de los sistemas basados en reglas, pero introducen una medida asociada a la incertidumbre de las reglas y a la de sus premisas. En este caso se pueden utilizar algunas fórmulas de propagación para calcular la incertidumbre asociada a las conclusiones. Durante las últimas décadas han sido propuestas algunas medidas de incertidumbre. Algunos ejemplos

de estas medidas son *los factores de certeza*, usados en las conchas para generar sistemas expertos tales como el sistema experto MYCIN (véase Buchanan y Shortliffe (1984)); la *lógica difusa* (véase, por ejemplo, Zadeh (1983) y Buckley, Siler, y Tucker (1986)); y la *teoría de la evidencia* de Dempster y Shafer (véase Shafer (1976)).

Otra medida intuitiva de incertidumbre es la *probabilidad*, en la que la *distribución conjunta* de un conjunto de variables se usa para describir las relaciones de dependencia entre ellas, y se sacan conclusiones usando fórmulas muy conocidas de la teoría de la probabilidad. Este es el caso del sistema experto PROSPECTOR (véase Duda, Gaschnig, y Hart (1980)), que utiliza el teorema de Bayes para la exploración de mineral.

Los sistemas expertos que utilizan la probabilidad como medida de incertidumbre se conocen como sistemas expertos *probabilísticos* y la estrategia de razonamiento que usan se conoce como *razonamiento probabilístico*, o *inferencia probabilística*. Este libro está dedicado a los sistemas expertos de tipo probabilístico. Otros libros que sirven para introducirse de forma general en otras medidas de incertidumbre son Buchanan y Shortliffe (1984), Waterman (1985), Pearl (1988), Jackson (1990), Neapolitan (1990), Castillo y Álvarez (1991), Durkin (1994) y Jensen (1996).

En los comienzos de los sistemas expertos de tipo probabilístico surgieron varios obstáculos, debido a las dificultades encontradas para definir la distribución de probabilidad conjunta de las variables. Ello ha ralentizado su desarrollo. Con la introducción de los *modelos de redes probabilísticas*, estos obstáculos se han superado y los sistemas expertos probabilísticos han vuelto de forma espectacular durante las dos últimas décadas. Estos modelos, que incluyen las redes de Markov y las Bayesianas, se basan en una representación gráfica de las relaciones entre las variables. Esta representación conduce no sólo a formas más eficientes de definir la distribución conjunta de probabilidad sino también a una propagación de incertidumbre muy eficiente, que permite sacar conclusiones. Ejemplos de tales conchas para el desarrollo de sistemas expertos son el sistema HUGIN (véase Andersen y otros (1989)) y *X-pert Nets*,¹ que ha sido desarrollado por los autores de este libro.

1.6 Componentes de un Sistema Experto

Las definiciones de sistemas expertos dadas en la Sección 1.2 se entienden mejor cuando se examinan las principales componentes de los sistemas expertos. Estas componentes se muestran esquemáticamente en la Figura 1.3 y se explican seguidamente.

¹Ésta y otras conchas para sistemas expertos pueden obtenerse de la dirección WWW <http://ccaix3.unican.es/~AIGroup>.

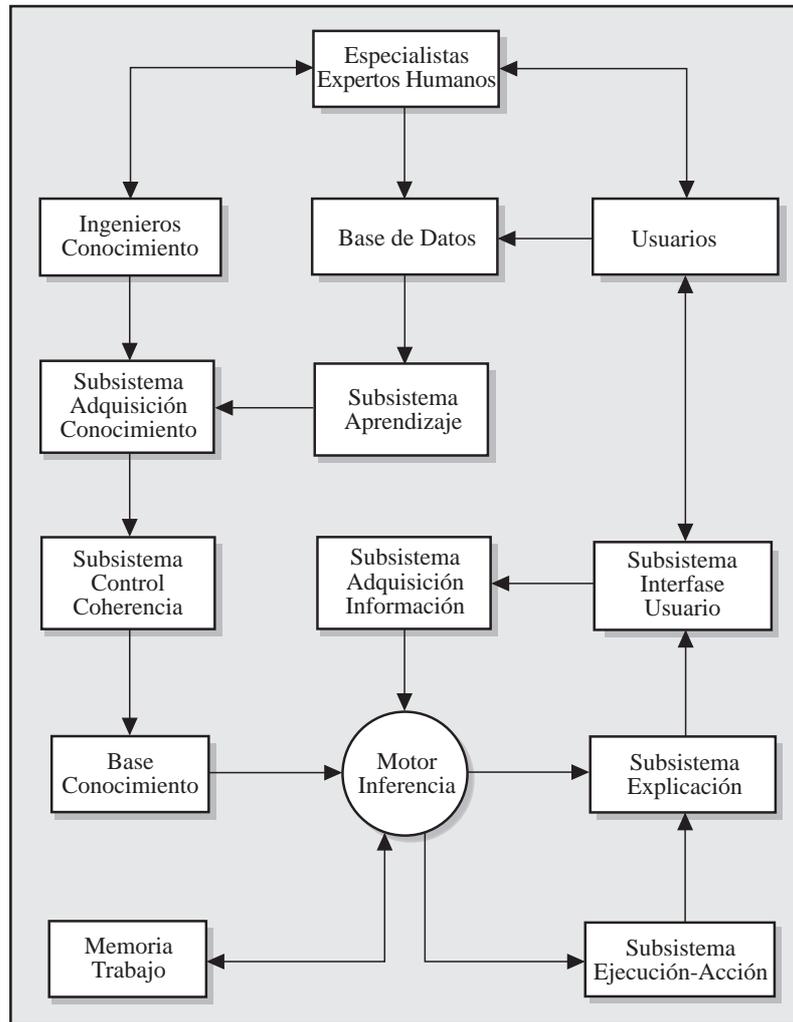


FIGURA 1.3. Componentes típicos de un sistema experto. Las flechas representan el flujo de la información.

1.6.1 La Componente Humana

Un sistema experto es generalmente el resultado de la colaboración de uno o varios *expertos humanos especialistas en el tema de estudio* y los *ingenieros del conocimiento*, con los *usuarios* en mente. Los expertos humanos suministran el conocimiento básico en el tema de interés, y los ingenieros del conocimiento trasladan este conocimiento a un lenguaje, que el sistema experto pueda entender. La colaboración de los expertos humanos, los ingenieros del conocimiento y los usuarios es, quizás, el elemento más

importante en el desarrollo de un sistema experto. Esta etapa requiere una enorme dedicación y un gran esfuerzo debido a los diferentes lenguajes que hablan las distintas partes y a las diferentes experiencias que tienen.

1.6.2 *La Base de Conocimiento*

Los especialistas son responsables de suministrar a los ingenieros del conocimiento una base de conocimiento ordenada y estructurada, y un conjunto de relaciones bien definidas y explicadas. Esta forma estructurada de pensar requiere que los expertos humanos repiensen, reorganicen, y reestructuren la base de conocimiento y, como resultado, el especialista se convierte en un mejor conocedor de su propio campo de especialidad.

Hay que diferenciar entre *datos* y *conocimiento*. El conocimiento se refiere a afirmaciones de validez general tales como reglas, distribuciones de probabilidad, etc. Los datos se refieren a la información relacionada con una aplicación particular. Por ejemplo, en diagnóstico médico, los síntomas, las enfermedades y las relaciones entre ellos, forman parte del conocimiento, mientras los síntomas particulares de un paciente dado forman parte de los datos. Mientras el conocimiento es permanente, los datos son efímeros, es decir, no forman parte de la componente permanente de un sistema y son destruidos después de usarlos. El conocimiento se almacena en la base de conocimiento y los datos se almacenan en la *memoria de trabajo*. Todos los procedimientos de los diferentes sistemas y subsistemas que son de carácter transitorio se almacenan también en la memoria de trabajo.

1.6.3 *Subsistema de Adquisición de Conocimiento*

El subsistema de adquisición de conocimiento controla el flujo del nuevo conocimiento que fluye del experto humano a la base de datos. El sistema determina qué nuevo conocimiento se necesita, o si el conocimiento recibido es en realidad nuevo, es decir, si debe incluirse en la base de datos y, en caso necesario, incorpora estos conocimientos a la misma.

1.6.4 *Control de la Coherencia*

El subsistema de control de la coherencia ha aparecido en los sistemas expertos muy recientemente. Sin embargo, es una componente esencial de un sistema experto. Este subsistema controla la consistencia de la base de datos y evita que unidades de conocimiento inconsistentes entren en la misma. En situaciones complejas incluso un experto humano puede formular afirmaciones inconsistentes. Por ello, sin un subsistema de control de la coherencia, unidades de conocimiento contradictorio pueden formar parte de la base de conocimiento, dando lugar a un comportamiento insatisfactorio del sistema. Es también bastante común, especialmente en sistemas con

mecanismos de propagación de incertidumbre, que se llegue a conclusiones absurdas o en conflicto como, por ejemplo, situaciones en las que el sistema genera probabilidades mayores que la unidad o negativas. Por ello, el subsistema de control de la coherencia comprueba e informa a los expertos de las inconsistencias. Por otra parte, cuando se solicita información de los expertos humanos, éste subsistema informa sobre las restricciones que ésta debe cumplir para ser coherente con la existente en la base de conocimiento. De esta forma, ayuda a los expertos humanos a dar información fiable.

1.6.5 *El Motor de Inferencia*

El motor de inferencia es el corazón de todo sistema experto. El cometido principal de esta componente es el de sacar conclusiones aplicando el conocimiento a los datos. Por ejemplo, en diagnóstico médico, los síntomas de un paciente (datos) son analizados a la luz de los síntomas y las enfermedades y de sus relaciones (conocimiento).

Las conclusiones del motor de inferencia pueden estar basadas en *conocimiento determinista* o *conocimiento probabilístico*. Como puede esperarse, el tratamiento de situaciones de incertidumbre (probabilísticas) puede ser considerablemente más difícil que el tratamiento de situaciones ciertas (deterministas). En muchos casos, algunos hechos (datos) no se conocen con absoluta certeza. Por ejemplo, piénsese en un paciente que no está seguro de sus síntomas. Puede darse el caso de tener que trabajar con conocimiento de tipo no determinista, es decir, de casos en los que se dispone sólo de información aleatoria o difusa. El motor de inferencia es también responsable de la propagación de este conocimiento incierto. De hecho, en los sistemas expertos basados en probabilidad, la propagación de incertidumbre es la tarea principal del motor de inferencia, que permite sacar conclusiones bajo incertidumbre. Esta tarea es tan compleja que da lugar a que ésta sea probablemente la componente más débil de casi todos los sistemas expertos existentes. Por esta razón, la mayor parte de este libro se dedica al análisis y resolución del problema de la propagación de incertidumbre.

1.6.6 *El Subsistema de Adquisición de Conocimiento*

Si el conocimiento inicial es muy limitado y no se pueden sacar conclusiones, el motor de inferencia utiliza el *subsistema de adquisición de conocimiento* para obtener el conocimiento necesario y continuar con el proceso de inferencia hasta que se hayan sacado conclusiones. En algunos casos, el usuario puede suministrar la información requerida para éste y otros objetivos. De ello resulta la necesidad de una *interfase de usuario* y de una comprobación de la consistencia de la información suministrada por el usuario antes de introducirla en la memoria de trabajo.

1.6.7 Interfase de Usuario

La interfase de usuario es el enlace entre el sistema experto y el usuario. Por ello, para que un sistema experto sea una herramienta efectiva, debe incorporar mecanismos eficientes para mostrar y obtener información de forma fácil y agradable. Un ejemplo de la información que tiene que ser mostrada tras el trabajo del motor de inferencia, es el de las conclusiones, las razones que expliquen tales conclusiones y una explicación de las acciones iniciadas por el sistema experto. Por otra parte, cuando el motor de inferencia no puede concluir debido, por ejemplo, a la ausencia de información, la interfase de usuario es un vehículo para obtener la información necesaria del usuario. Consecuentemente, una implementación inadecuada de la interfase de usuario que no facilite este proceso minaría notablemente la calidad de un sistema experto. Otra razón de la importancia de la interfase de usuario es que los usuarios evalúan comúnmente los sistemas expertos y otros sistemas por la calidad de dicha interfase más que por la del sistema experto mismo, aunque no se debería juzgar la calidad de un libro por su portada. Los lectores que estén interesados en el diseño de una interfase de usuario pueden consultar los libros de Shneiderman (1987) y Brown y Cunningham (1989).

1.6.8 El Subsistema de Ejecución de Órdenes

El *subsistema de ejecución de órdenes* es la componente que permite al sistema experto iniciar acciones. Estas acciones se basan en las conclusiones sacadas por el motor de inferencia. Como ejemplos, un sistema experto diseñado para analizar el tráfico ferroviario puede decidir retrasar o parar ciertos trenes para optimizar el tráfico global, o un sistema para controlar una central nuclear puede abrir o cerrar ciertas válvulas, mover barras, etc., para evitar un accidente. La explicación de las razones por las que se inician estas acciones pueden darse al usuario mediante el *subsistema de explicación*.

1.6.9 El Subsistema de Explicación

El usuario puede pedir una explicación de las conclusiones sacadas o de las acciones iniciadas por el sistema experto. Por ello, es necesario un subsistema que explique el proceso seguido por el motor de inferencia o por el subsistema de ejecución. Por ejemplo, si un cajero automático decide rechazar la palabra clave (una acción), la máquina puede mostrar un mensaje (una explicación) como la siguiente:

¡Lo siento!, su palabra clave es todavía incorrecta tras tres intentos.

Retenemos su tarjeta de crédito, para garantizar su seguridad.

Por favor, póngase en contacto con su banco en horas de oficina.

En muchos dominios de aplicaciones, es necesaria la explicación de las conclusiones debido a los riesgos asociados con las acciones a ejecutar. Por ejemplo, en el campo del diagnóstico médico, los doctores son responsable últimos de los diagnósticos, independientemente de las herramientas técnicas utilizadas para sacar conclusiones. En estas situaciones, sin un subsistema de explicación, los doctores pueden no ser capaces de explicar a sus pacientes las razones de su diagnóstico.

1.6.10 El Subsistema de Aprendizaje

Una de las principales características de un sistema experto es su capacidad para aprender. Diferenciaremos entre aprendizaje estructural y aprendizaje paramétrico. Por *aprendizaje estructural* nos referimos a algunos aspectos relacionados con la estructura del conocimiento (reglas, distribuciones de probabilidad, etc.). Por ello, el descubrimiento de nuevos síntomas relevantes para una enfermedad o la inclusión de una nueva regla en la base de conocimiento son ejemplos de aprendizaje estructural. Por *aprendizaje paramétrico* nos referimos a estimar los parámetros necesarios para construir la base de conocimiento. Por ello, la estimación de frecuencias o probabilidades asociadas a síntomas o enfermedades es un ejemplo de aprendizaje paramétrico.

Otra característica de los sistemas expertos es su habilidad para obtener *experiencia* a partir de los *datos* disponibles. Estos datos pueden ser obtenidos por expertos y no expertos y pueden utilizarse por el *subsistema de adquisición de conocimiento* y por el *subsistema de aprendizaje*.

De las componentes antes mencionadas puede verse que los sistemas expertos pueden realizar varias tareas. Estas tareas incluyen, pero no se limitan a, las siguientes:

- Adquisición de conocimiento y la verificación de su coherencia; por lo que el sistema experto puede ayudar a los expertos humanos a dar conocimiento coherente.
- Almacenar (memorizar) conocimiento.
- Preguntar cuándo se requiere nuevo conocimiento.
- Aprender de la base de conocimiento y de los datos disponibles.
- Realizar inferencia y razonamiento en situaciones deterministas y de incertidumbre.
- Explicar conclusiones o acciones tomadas.
- Comunicar con los expertos y no expertos humanos y con otros sistemas expertos.

1.7 Desarrollo de un Sistema Experto

Weiss y Kulikowski (1984) sugieren las etapas siguientes para el diseño e implementación de un sistema experto (ver también Hayes-Roth, Waterman, y Lenat (1983), Luger y Stubblefield (1989), y la Figura 1.4):

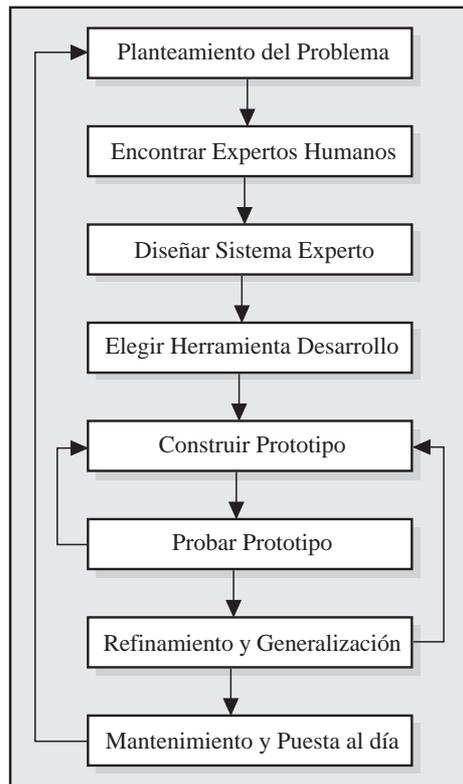


FIGURA 1.4. Etapas en el desarrollo de un sistema experto.

1. **Planteamiento del problema.** La primera etapa en cualquier proyecto es normalmente la definición del problema a resolver. Puesto que el objetivo principal de un sistema experto es responder a preguntas y resolver problemas, esta etapa es quizás la más importante en el desarrollo de un sistema experto. Si el sistema está mal definido, se espera que el sistema suministre respuestas erróneas.
2. **Encontrar expertos humanos que puedan resolver el problema.** En algunos casos, sin embargo, las bases de datos pueden jugar el papel del experto humano.

3. **Diseño de un sistema experto.** Esta etapa incluye el diseño de estructuras para almacenar el conocimiento, el motor de inferencia, el subsistema de explicación, la interfase de usuario, etc.
4. **Elección de la herramienta de desarrollo, concha, o lenguaje de programación.** Debe decidirse si realizar un sistema experto a medida, o utilizar una concha, una herramienta, o un lenguaje de programación. Si existiera una concha satisfaciendo todos los requerimientos del diseño, ésta debería ser la elección, no sólo por razones de tipo financiero sino también por razones de fiabilidad. Las conchas y herramientas comerciales están sujetas a controles de calidad, a los que otros programas no lo están.
5. **Desarrollo y prueba de un prototipo.** Si el prototipo no pasa las pruebas requeridas, las etapas anteriores (con las modificaciones apropiadas) deben ser repetidas hasta que se obtenga un prototipo satisfactorio.
6. **Refinamiento y generalización.** En esta etapa se corrigen los fallos y se incluyen nuevas posibilidades no incorporadas en el diseño inicial.
7. **Mantenimiento y puesta al día.** En esta etapa el usuario plantea problemas o defectos del prototipo, corrige errores, actualiza el producto con nuevos avances, etc.

Todas estas etapas influyen en la calidad del sistema experto resultante, que siempre debe ser evaluado en función de las aportaciones de los usuarios. Para el lector interesado en estos temas recomendamos la lectura de los trabajos de O'Keefe, Balci y Smith (1987), Chandrasekaran (1988) y Preece (1990).

1.8 Otras Áreas de la IA

En esta sección se da una breve descripción panorámica del ámbito y dominio de algunas áreas de la IA distintas de la de los sistemas expertos. Puesto que este libro está dedicado exclusivamente a sistemas expertos, se dan algunas referencias para que el lector interesado pueda acceder a otras áreas de la IA. Debe tenerse en mente que ésta no es una lista exhaustiva de todas las áreas de la IA y que la IA es un campo que se desarrolla muy rápidamente, y emergen continuamente nuevas ramas para tratar las nuevas situaciones de esta ciencia que no para de crecer.

Hay varios libros que dan una visión general de la mayoría de los temas incluidos en la IA. El multivolumen *Handbook of Artificial Intelligence* editado por Barr y Feigenbaum (1981, 1982) (volúmenes 1 y 2) y por Cohen y Feigenbaum (1982) (volumen 3), y la *Encyclopedia of Artificial Intelligence*,

editado por Shapiro (1987) contienen discusiones detalladas de varios de los temas de la IA. Hay otros muchos libros que cubren las áreas de IA. Por mencionar unos pocos, citamos a: Charniak y McDermott (1985), Rich y Knight (1991), Winston (1992), Ginsberg (1993), Russell y Norvig (1995).

Como consecuencia de la intensa investigación realizada en el área de la IA, hay también un número creciente de revistas que publican artículos en los distintos campos de la IA y temas relacionados con ella. Algunas de estas revistas son: *Applied Artificial Intelligence*, *Applied Intelligence*, *Artificial Intelligence*, *Artificial Intelligence Magazine*, *International Journal of Intelligent Systems*.

Por otra parte, revistas tales como *Artificial Intelligence in Medicine*, *Biocybernetics and Biomedical Engineering*, *Cybernetics and Systems*, *Fuzzy Sets and Systems*, *IEEE Expert*, *IEEE Transactions on Systems, Man and Cybernetics*, *International Journal for Artificial Intelligence in Engineering*, *International Journal of Approximate Reasoning*, *International Journal of Computer Vision*, *International Journal of Expert Systems*, *Machine Learning*, *Networks*, *Neural Networks*, y *Pattern Recognition Letters* se especializan en un tema o en un cierto dominio de aplicaciones.²

1.8.1 Representación del Conocimiento

Hay muchas fuentes de información o conocimiento relacionadas con la IA. El campo de la representación del conocimiento se refiere a los mecanismos para representar y manipular esta información. Los esquemas de representación resultantes deberían permitir una búsqueda o una operación eficiente de los mecanismos de inferencia. Por ejemplo, en algunos casos la información puede ser representada mediante objetos (o variables) y por reglas lógicas (que expresan relaciones entre los objetos). Por ello, esta representación puede manipularse usando análisis lógico. Éste es el mecanismo de representación del conocimiento utilizado, por ejemplo, en los sistemas expertos basados en reglas (Capítulo 2). Para tener una visión general de las diferentes metodologías de representación del conocimiento véase, por ejemplo, Bachman, Levesque, y Reiter (1991), Bench-Capon (1990), y los *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning* (KR-89, 91, 92, y 94) publicados por Morgan y Kaufmann Publishers.

²Una lista que contiene la mayoría de revistas en el campo de la IA se puede obtener en la dirección WWW “<http://ai.iit.nrc.ca/ai-journals.html>”; véase también “<http://www.bus.orst.edu/faculty/brownc/aies/journals.htm>.”

1.8.2 Planificación

Cuando se trata con problemas complejos, es importante dividir las tareas en partes más pequeñas que sean más fáciles de manejar. Los métodos de planificación analizan diferentes estrategias para descomponer un problema dado, resolver cada una de sus partes, y llegar a una solución final. La interacción entre las partes dependerá del grado de descomponibilidad del problema. Por otra parte, el comienzo de la computación paralela, capaz de realizar varias tareas simultáneamente, da lugar a nuevos problemas que requieren estrategias especiales de planificación. En esta situación, el objetivo consiste en dividir las tareas de forma adecuada para resolver muchas partes simultáneamente. El trabajo editado por Allen, Hendler, y Tate (1990), da una descripción general de este campo. Por otra parte, la colección de artículos editada por Bond y Gasser (1988) está dedicada al *razonamiento paralelo*, también conocido como *razonamiento distribuido*.

1.8.3 Demostración Automática de Teoremas

La capacidad de hacer deducciones lógicas fue considerada durante mucho tiempo como una posibilidad reservada a la mente humana. La investigación desarrollada en los años 1960 en el área de la demostración automática de teoremas ha mostrado que esta tarea puede ser realizada por máquinas programables. Tales máquinas son capaces no sólo de modificar el conocimiento existente, sino también de obtener conclusiones nuevas. En primer lugar, *los demostradores de teoremas* han sido utilizados en varios campos de las matemáticas, tales como la Lógica, la Geometría, etc. El campo de la Matemática constituye un área natural para esta metodología por la existencia de mecanismos de deducción y de una extensa base de conocimiento. Sin embargo, los demostradores de teoremas, pueden ser adaptados para resolver problemas de otras áreas de conocimiento con estas dos mismas características. Una introducción general a este tema se da en Wos y otros (1984) y Bundy (1983), que contiene el código en Prolog de un demostrador de teoremas muy simple. Referencias más recientes son las de Newborn (1994), Almula (1995) y las incluídas en ellos.

1.8.4 Los Juegos Automatizados

Los juegos automatizados constituyen un ejemplo de una de las más antiguas y fascinantes áreas de la IA (véase, por ejemplo, Newell, Shaw, y Simon (1963)). Los juegos por computador (tales como el ajedrez, backgammon, y los de cartas) han visto un desarrollo masivo en los últimos años. Por ejemplo, los programas de juegos de ajedrez son capaces de competir e incluso vencer a bien conocidos maestros. El juego automático requiere un estudio teórico profundo y tiene varias aplicaciones en otras áreas tales como *métodos de búsqueda*, *optimización*, etc. Una buena discusión

de este campo, que incluye además referencias históricas de interés, puede encontrarse en Levy (1988).

1.8.5 *Reconocimiento de patrones*

El reconocimiento de patrones trata de diferentes técnicas de clasificación para identificar los subgrupos, o conglomerados, con características comunes en cada grupo. El grado de asociación de cualquiera de los objetos con cada uno de los grupos suministra un modo de sacar conclusiones. Por ello, los algoritmos desarrollados en este área son herramientas útiles para tratar con varios problemas de muchos campos tales como reconocimiento de imágenes, reconocimiento de señales, diagnóstico de fallos de equipos, control de procesos, etc. Para una introducción general a este campo véase Sing-Tze (1984) y Niemann (1990) y Patrick y Fattu (1984) para una discusión de tipo estadístico.

1.8.6 *Reconocimiento de la Voz*

La voz es, con mucho, el medio de comunicación más usado por el hombre. El reconocimiento de voz trata del problema de procesar el lenguaje hablado y capturar los diferentes elementos semánticos que forman la conversación. Los problemas asociados con las diferentes pronunciaciones y tonos de voz son los principales obstáculos que esta disciplina tiene que afrontar. Una introducción general al problema del reconocimiento de voz se da en Rabiner y Juang (1993).

1.8.7 *Procesamiento de Lenguaje Natural*

Un objetivo del procesamiento del lenguaje natural consiste en extraer tanta información como sea posible de un texto escrito. Con el uso creciente de los ordenadores en el tratamiento de la información, el lenguaje escrito está jugando un papel muy importante como medio de comunicación. Puesto que el reconocimiento de la voz es inherentemente un problema más difícil, resulta necesario disponer de un procesado eficiente del lenguaje escrito. El problema inverso del procesamiento del lenguaje es el de la *generación de lenguaje*, es decir, dotar a los computadores de capacidades para generar sentencias de lenguaje natural en vez de mensajes de tipo telegráfico. La combinación de estas dos tareas permitiría, por ejemplo, la posibilidad de traducir textos escritos en diferentes idiomas, lo que se conoce como *traducción asistida por ordenador*. Una referencia clásica a este campo es Schank y Abelson (1977). También se pueden encontrar descripciones interesantes de procesamiento del lenguaje natural en Allen (1995) y McKeown (1985).

1.8.8 *Visión Artificial*

Uno de los objetivos de la visión artificial es la posibilidad de usar ordenadores para localizar y reconocer automáticamente objetos en tres dimensiones. Muchas otras áreas de la IA tales como la representación del conocimiento, el reconocimiento de patrones, y las redes neuronales juegan un papel esencial en la visión artificial. Los muy significativos avances técnicos producidos durante la última década han sido aplicados a varios sistemas comerciales utilizados en fabricación, inspección, tareas de guía, etc. Para una introducción general a este área, véase Fischler y Firschein (1987) y Shapiro y Rosenfeld (1992).

1.8.9 *Robótica*

La robótica es una de las áreas de la IA más populares. Los robots combinan elementos mecánicos, sensores, y ordenadores que les permiten tratar con objetos reales y realizar muchas tareas de forma precisa, rápida y cómoda. Por ello, se puede pensar en los robots como ordenadores que interactúan con el mundo real. Una revisión general de la robótica se presenta en McKerrow (1991), mientras Jones y Flynn (1993) tratan el tema de las aplicaciones prácticas.

1.8.10 *Redes Neuronales*

Las redes neuronales se crearon con el objetivo de reproducir de forma básica las funciones elementales del cerebro humano. Las arquitecturas en red con un gran número de conexiones entre varias capas de procesadores fueron introducidas para reproducir la estructura del cerebro humano. La información contenida en una red neuronal se codifica en la estructura de la red y en los pesos de las conexiones. Por tanto, en una situación particular, los pesos de las conexiones tienen que modificarse para reproducir la salida deseada. Esta tarea de aprendizaje se consigue mediante una técnica de aprender por analogía, es decir, el modelo se entrena para reproducir las salidas de un conjunto de señales de entrenamiento con el objetivo de codificar de esta forma la estructura del fenómeno. La aparición de ordenadores rápidos en los que pudieran simularse redes grandes y complejas, y el descubrimiento de potentes algoritmos de aprendizaje han sido las causas que han posibilitado el desarrollo rápido de este área de conocimiento. Para una introducción ver, por ejemplo, Freeman y Skapura (1991) y Lisboa (1992).

1.9 Conclusiones

A partir de la breve descripción de las variadas áreas de la IA mostradas en este capítulo, se puede ver que éstas están interrelacionadas. Por ejemplo, la robótica utiliza otras áreas de la IA tales como la visión automática y el reconocimiento de patrones o de la voz. El área de la IA, como un todo, es altamente interdisciplinar. Por ejemplo, los sistemas expertos requieren varios conceptos de la ciencia del computador, la lógica matemática, la teoría de grafos, la teoría de la probabilidad y la estadística. Por ello, el trabajo en este campo requiere la colaboración de muchos investigadores en diferentes áreas de especialización.

Capítulo 2

Sistemas Basados en Reglas

2.1 Introducción

En nuestra vida diaria encontramos muchas situaciones complejas gobernadas por reglas deterministas: sistemas de control de tráfico, sistemas de seguridad, transacciones bancarias, etc. Los sistemas basados en reglas son una herramienta eficiente para tratar estos problemas. Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. La base de conocimiento contiene el conjunto de reglas que definen el problema, y el motor de inferencia saca las conclusiones aplicando la lógica clásica a estas reglas. Una introducción general a los sistemas expertos basados en reglas, puede encontrarse, por ejemplo, en Buchanan y Shortliffe (1984), Castillo y Álvarez (1991), Durkin (1994), Hayes-Roth (1985), Waterman (1985), y también en el trabajo editado por García y Chien (1991). El libro de Pedersen (1989) muestra un enfoque práctico e incluye varios algoritmos.

Este capítulo presenta los conceptos básicos que forman parte de los sistemas expertos basados en reglas. No se pretende realizar una descripción detallada de estos sistemas, para la que hay libros mucho más completos que éste, sino sólo introducir al lector, de forma simple e intuitiva, en esta metodología. La intención de este capítulo es mostrar cómo los sistemas probabilísticos pueden considerarse como una generalización de los sistemas basados en reglas. La Sección 2.2 describe la base de conocimiento de los sistemas expertos basados en reglas y da una definición y ejemplos de reglas, que constituyen el corazón de la base de conocimiento. Seguidamente, se

Objeto	Conjunto de valores posibles
Nota	{0, 1, ..., 10}
Calificación	{sobresaliente, notable, aprobado, suspenso}
Puesto	{0, 1, ..., 100}
Admitir	{sí, pendiente, no}
Notificar	{sí, no}

TABLA 2.1. Un ejemplo de objetos con sus posibles valores.

discute cómo opera el motor de inferencia (Sección 2.3), cómo trabaja el subsistema de control de la coherencia (Sección 2.4), y cómo se explican las conclusiones sacadas por el motor de inferencia (Sección 2.5). La Sección 2.6, muestra un ejemplo de aplicación. Finalmente, la Sección 2.7 muestra algunas de las limitaciones de los sistemas expertos basados en reglas.

2.2 La Base de Conocimiento

En los sistemas basados en reglas intervienen dos elementos importantes: la base de conocimiento y los datos. Los datos están formados por la evidencia o los hechos conocidos en una situación particular. Este elemento es dinámico, es decir, puede cambiar de una aplicación a otra. Por esta razón, no es de naturaleza permanente y se almacena en la memoria de trabajo.

En situaciones deterministas, las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas. El conocimiento se almacena en la base de conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos. La información almacenada en la base de conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje.

Para dar una idea intuitiva de lo que es una regla, supóngase que se tiene un conjunto de *objetos* y, por simplicidad, que cada objeto puede tener uno y sólo uno de un conjunto de posibles valores. Ejemplos de objetos con sus posibles valores se dan en la Tabla 2.1.

Seguidamente se dan unos pocos ejemplos de reglas:

Regla 1: Si $\text{nota} > 9$, entonces $\text{calificación} = \text{sobresaliente}$.

Regla 2: Si $\text{puesto} < 20$ o $\text{nota} > 7$, entonces $\text{Admitir} = \text{sí}$ y $\text{Notificar} = \text{sí}$.

Cada una de las reglas anteriores relaciona dos o más objetos y está formada por las partes siguientes:

- La *premisa* de la regla, que es la *expresión lógica* entre las palabras clave *si* y *entonces*. La premisa puede contener una o más afirmaciones

objeto-valor conectadas con operadores lógicos *y*, *o*, o *no*. Por ejemplo, la premisa de la Regla 1 consta de una única afirmación objeto-valor, mientras que las premisas de la Regla 2 constan de dos afirmaciones objeto-valor conectadas por un operador lógico.

- La *conclusión* de la regla, que es la expresión lógica tras la palabra clave *entonces*.

Los ejemplos anteriores facilitan la definición siguiente de regla.

Definición 2.1 Regla. *Una regla es una afirmación lógica que relaciona dos o más objetos e incluye dos partes, la premisa y la conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante los operadores lógicos *y*, *o*, o *no*.*

Una regla se escribe normalmente como “Si *premisa*, entonces *conclusión*”. En general, ambas, la premisa y la conclusión de una regla, pueden contener afirmaciones múltiples objeto-valor. Una expresión lógica que contiene sólo una afirmación objeto-valor se denomina *expresión lógica simple*; en caso contrario, la expresión se dice *expresión lógica compuesta*. Por ejemplo, las expresiones lógicas en ambas, premisa y conclusión de la Regla 1, son simples, mientras que las expresiones lógicas de las premisas y la conclusión de la Regla 2 es compuesta. Correspondientemente, una regla que contiene solamente expresiones lógicas simples se denomina una *regla simple*; en otro caso, se llama *regla compuesta*. Por ejemplo, la Regla 1 es simple, mientras que la Reglas 2 es compuesta.

Ejemplo 2.1 Cajero Automático. Como ejemplo de problema determinista que puede ser formulado usando un conjunto de reglas, considérese una situación en la que un usuario (por ejemplo, un cliente) desea sacar dinero de su cuenta corriente mediante un cajero automático (CA). En cuanto el usuario introduce la tarjeta en el CA, la máquina la lee y la verifica. Si la tarjeta no es verificada con éxito (por ejemplo, porque no es legible), el CA devuelve la tarjeta al usuario con el mensaje de error correspondiente. En otro caso, el CA pide al usuario su número de identificación personal (NIP). Si el número fuese incorrecto, se dan tres oportunidades de corregirlo. Si el NIP es correcto, el CA pregunta al usuario cuánto dinero desea sacar. Para que el pago se autorice, la cantidad solicitada no debe exceder de una cierta cantidad límite diaria, además de haber suficiente dinero en su cuenta.

En este caso se tienen siete objetos, y cada objeto puede tomar uno y sólo un valor de entre sus posibles valores. La Tabla 2.2 muestra estos objetos y sus posibles valores.

La Figura 2.1 muestra siete reglas que gobiernan la estrategia que el CA debe seguir cuando un usuario intenta sacar dinero de su cuenta. En la Regla 1, por ejemplo, la premisa consiste en seis afirmaciones objeto-valor conectadas mediante el operador lógico *y*, lo que indica que la premisa

Objeto	Conjunto de posibles valores
Tarjeta	{verificada, no verificada}
Fecha	{expirada, no expirada}
NIP	{correcto, incorrecto}
Intentos	{excedidos, no excedidos}
Balance	{suficiente, insuficiente}
Límite	{excedido, no excedido}
Pago	{autorizado, no autorizado}

TABLA 2.2. Objetos y posibles valores para el ejemplo del cajero automático.

es cierta si las seis afirmaciones lo son. Por ello, la Regla 1 relaciona el objeto *Pago* (en la conclusión) con los demás objetos. Según la Regla 1, la acción que debe iniciar el CA es dar el dinero al usuario si la tarjeta se ha verificado correctamente, la fecha no ha expirado, el NIP es correcto, el número de intentos para dar el NIP correcto no se ha excedido y la cantidad solicitada no excede ni la cantidad disponible ni el límite máximo diario. Las expresiones lógicas en cada una de las restantes reglas de la Figura 2.1 constan de una sola afirmación. Nótese que la Regla 1 indica cuándo debe permitirse el pago, y las restantes cuándo debe rechazarse. ■

Ejemplo 2.2 Gente famosa. Supóngase que se dispone de una base de datos consistente en N individuos. Para cada individuo, la base de datos contiene cuatro atributos: nombre, sexo, nacionalidad y profesión. Supóngase que la base de datos muestra sólo si una persona es americana, política y/o si es mujer. Cada uno de estos atributos es binario (toma sólo dos valores posibles). En este caso, la base de datos puede contener, como mucho, $2^3 = 8$ conjuntos disjuntos. Estos conjuntos se muestran en la Figura 2.2. La figura muestra también el nombre de una persona en cada subconjunto. La Tabla 2.3 da un ejemplo de una base de datos que contiene $N = 8$ personas famosas. En este caso se tienen cuatro objetos: *Nombre*, *Americano*, *Político*, y *Mujer*. El primer objeto puede tomar uno de N posibles valores (los nombres de cada persona) y cada uno de los tres últimos objetos pueden tomar el valor *sí* o el valor *no*.

A partir de la Tabla 2.3 se pueden construir reglas para identificar a cada persona, resultando un total de ocho reglas. Por ejemplo, la regla siguiente corresponde al presidente Clinton:

- Regla 1: Si *Nombre = Clinton*, entonces *Americano = sí* y *Político = sí* y *Mujer = no*.

Las restantes siete reglas pueden construirse de forma análoga. ■

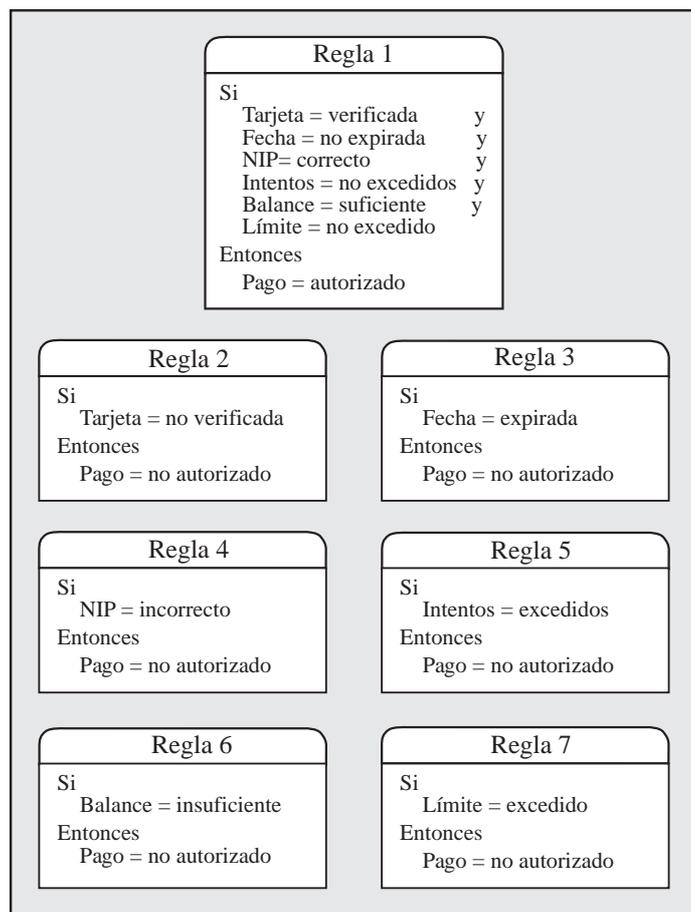


FIGURA 2.1. Ejemplos de reglas para sacar dinero de un cajero automático.

Nombre	Americano	Político	Mujer
Barbara Jordan	sí	sí	sí
Bill Clinton	sí	sí	no
Barbara Walters	sí	no	sí
Mohammed Ali	sí	no	no
Margaret Thatcher	no	sí	sí
Anwar El-Sadat	no	sí	no
Marie Curie	no	no	sí
Pablo Picasso	no	no	no

TABLA 2.3. Una base de datos mostrando cuatro objetos y sus valores correspondientes para el ejemplo de las personas famosas.

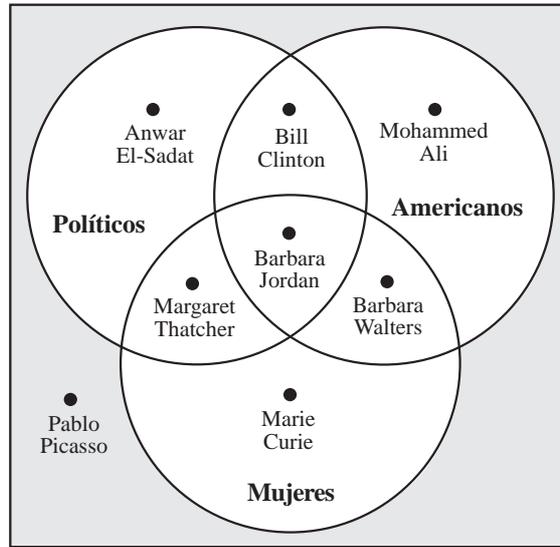


FIGURA 2.2. Un ejemplo de una base de datos con tres atributos binarios que dividen la población en ocho conjuntos disjuntos.

Los Ejemplos 2.1 y 2.2 se utilizarán posteriormente en este capítulo para ilustrar varios conceptos relacionados con los sistemas expertos basados en reglas.

Algunos sistemas imponen ciertas restricciones a las reglas. Por ejemplo:

- No permitir en la premisa el operador lógico *o*, y
- Limitar las conclusiones a expresiones lógicas simples.

Hay buenas razones para imponer estas restricciones. En primer lugar, las reglas que satisfacen estas restricciones son fáciles de tratar a la hora de escribir un programa de ordenador. En segundo lugar, las dos restricciones anteriores no dan lugar a una pérdida de generalidad, puesto que reglas mucho más generales pueden ser reemplazadas por conjuntos de reglas de esta forma. A esto se le llama *sustitución de reglas*. Por tanto, el conjunto de reglas especificado inicialmente por el experto humano puede requerir una sustitución posterior por un conjunto de reglas equivalente para satisfacer estas restricciones.

La Tabla 2.4 da ejemplos de sustitución de reglas. Nótese que cada regla de la primera columna puede ser sustituida por el correspondiente conjunto de reglas de la segunda columna y que todas las reglas de ésta satisfacen las condiciones anteriores. Por ejemplo, la primera regla compuesta de la Tabla 2.4:

- Regla 1: Si *A* o *B*, entonces *C*,

puede ser reemplazada por las dos reglas simples:

Regla	Reglas Equivalentes
Si A o B , entonces C	Si A , entonces C Si B , entonces C
Si \overline{A} o \overline{B} , entonces C	Si \bar{A} y \bar{B} , entonces C
Si \overline{A} y \overline{B} , entonces C	Si \bar{A} , entonces C Si \bar{B} , entonces C
Si $(A$ o $B)$ y C , entonces D	Si A y C , entonces D Si B y C , entonces D
Si $\overline{(A$ o $B)}$ y C , entonces D	Si \bar{A} y \bar{B} y C , entonces D
Si \bar{A} y \bar{B} y C , entonces D	Si \bar{A} y C , entonces D Si \bar{B} y C , entonces D
Si A , entonces B y C	Si A , entonces B Si A , entonces C
Si A , entonces B o C	Si A y \bar{B} , entonces C Si A y \bar{C} , entonces B
Si A , entonces \overline{B} y \overline{C}	Si A y B , entonces \bar{C} Si A y C , entonces \bar{B}
Si A , entonces \overline{B} o \overline{C}	Si A , entonces \bar{B} Si A , entonces \bar{C}

TABLA 2.4. Ejemplos de sustitución de reglas: Las reglas en la primera columna son equivalentes a las reglas de la segunda columna. Nótese que en los seis primeros ejemplos las sustituciones se aplican a la premisa y en los cuatro últimos, a la conclusión.

- Regla 1a: Si A , entonces C .
- Regla 1b: Si B , entonces C .

Como ejemplo adicional, la Tabla 2.5 muestra que

- Regla 2: Si \overline{A} o \overline{B} , entonces C ,

puede ser reemplazada por la regla

- Regla 2: Si \bar{A} y \bar{B} , entonces C ,

donde \bar{A} significa *no* A . La Tabla 2.5 se llama *tabla de verdad*.

A	B	\bar{A}	\bar{B}	$\overline{A \circ B}$	$\bar{A} \text{ y } \bar{B}$
C	C	F	F	F	F
C	F	F	C	F	F
F	C	C	F	F	F
F	F	C	C	C	C

TABLA 2.5. Una tabla de verdad mostrando que las expresiones lógicas $\overline{A \circ B}$ y $\bar{A} \text{ y } \bar{B}$ son equivalentes. Los símbolos C y F se utilizan para cierto y falso, respectivamente.

2.3 El Motor de Inferencia

Tal como se ha mencionado en la sección anterior, hay dos tipos de elementos: los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento). El motor de inferencia usa ambos para obtener nuevas conclusiones o hechos. Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla debe ser también cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones. Por ello, tanto los hechos iniciales o datos de partida como las conclusiones derivadas de ellos forman parte de los hechos o datos de que se dispone en un instante dado.

Las conclusiones pueden clasificarse en dos tipos: *simples* y *compuestas*. Las conclusiones simples son las que resultan de una regla simple. Las conclusiones compuestas son las que resultan de más de una regla. Para obtener conclusiones, los expertos utilizan diferentes tipos de reglas y estrategias de inferencia y control (véase, por ejemplo, Castillo y Álvarez (1991), Durkin (1994), Shapiro (1987), Waterman (1985)). En el resto de esta sección se discuten las reglas de inferencia

- Modus Ponens,
- Modus Tollens,
- Resolución,

y las estrategias de inferencia

- Encadenamiento de reglas,
- Encadenamiento de reglas orientado a un objetivo,
- Compilación de reglas,

que son utilizadas por el motor de inferencia para obtener conclusiones simples y compuestas. Las dos primeras reglas de inferencia se usan para obtener conclusiones simples y el resto de reglas y estrategias para obtener conclusiones compuestas.

Nótese, sin embargo, que ninguna de las estrategias anteriores, si se implementan solas, conduce a todas las conclusiones posibles. Por ello, deben implementarse varias reglas y estrategias en el sistema experto para que el motor de inferencia sea capaz de obtener tantas conclusiones como sea posible.

2.3.1 *Modus Ponens y Modus Tollens*

El *Modus Ponens* es quizás la regla de inferencia más comúnmente utilizada. Se utiliza para obtener conclusiones simples. En ella, se examina la premisa de la regla, y si es cierta, la conclusión pasa a formar parte del conocimiento. Como ilustración, supóngase que se tiene la regla, “Si A es cierto, entonces B es cierto” y que se sabe además que “ A es cierto.” Entonces, tal como muestra la Figura 2.3, la regla Modus Ponens concluye que “ B es cierto.” Esta regla de inferencia, que parece trivial, debido a su familiaridad, es la base de un gran número de sistemas expertos.

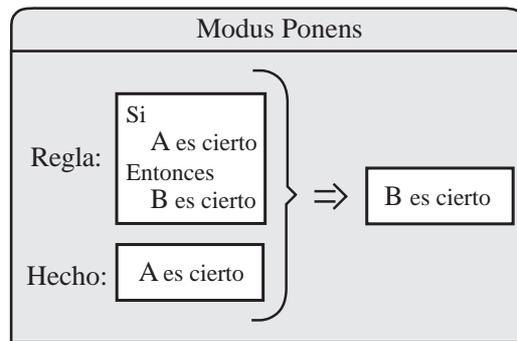


FIGURA 2.3. Una ilustración de la regla de inferencia Modus Ponens.

La regla de inferencia *Modus Tollens* se utiliza también para obtener conclusiones simples. En este caso se examina la conclusión y si es falsa, se concluye que la premisa también es falsa. Por ejemplo, supóngase de nuevo que se tiene la regla, “Si A es cierto, entonces B es cierto” pero se sabe que “ B es falso.” Entonces, utilizando la regla Modus Ponens no se puede obtener ninguna conclusión, pero, tal como se muestra en la Figura 2.4, la regla Modus Tollens concluye que “ A es falso.” Aunque muy simple y con muchas aplicaciones útiles, la regla Modus Tollens es menos utilizada que la Modus Ponens.

Por ello, la regla Modus Ponens se mueve hacia adelante, es decir, de la premisa a la conclusión de una regla, mientras que la regla Modus Tollens se mueve hacia atrás, es decir, de la conclusión a la premisa. Las dos reglas de inferencia no deben ser vistas como alternativas sino como complementarias. La regla Modus Ponens necesita información de los objetos

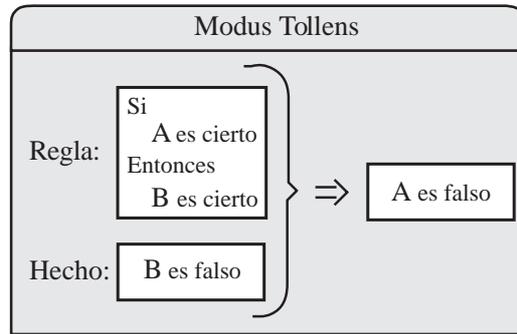


FIGURA 2.4. Una ilustración de la regla Modus Tollens.

de la premisa para concluir, mientras que la regla Modus Tollens necesita información sobre los objetos de la conclusión. De hecho, para un motor de inferencia que solamente utiliza Modus Ponens, la incorporación de la regla de inferencia Modus Tollens puede ser considerada como una expansión de la base de conocimiento mediante la adición de reglas, tal como ilustra el ejemplo que sigue.

Ejemplo 2.3 La regla Modus Tollens equivale a una expansión de la base de conocimiento. Supóngase que la base de conocimiento consiste sólo en la Regla 1, que se muestra en la Figura 2.5. Se puede utilizar la regla de inferencia Modus Tollens para “invertir” la Regla 1 y obtener alguna conclusión cuando se tiene información sobre los objetos de su conclusión. Entonces, aplicar la regla Modus Tollens a la regla “Si A , entonces B ” es equivalente a aplicar la regla Modus Ponens a la regla “Si \bar{B} , entonces \bar{A} .” En este caso de Regla 1, utilizando la equivalencia

$$\overline{A = C \text{ y } B = C} \Leftrightarrow \overline{A = F \text{ o } B = F},$$

se obtiene la Regla 1b, que se muestra en la Figura 2.6. Por ello, utilizar ambas, las reglas Modus Ponens y Modus Tollens cuando la base de conocimiento contiene sólo la Regla 1, es equivalente a usar la regla Modus Ponens cuando la base de conocimiento contiene ambas, la Regla 1 y la Regla 1b. ■

Por otra parte, el rendimiento del motor de inferencia depende del conjunto de reglas en su base de conocimiento. Hay situaciones en las que el motor de inferencia puede concluir utilizando un conjunto de reglas, pero no puede, utilizando otro (aunque éstos sean lógicamente equivalentes). A continuación se da un ejemplo ilustrativo.

Ejemplo 2.4 Inferencia con dos conjuntos equivalentes de reglas. Supóngase de nuevo que se tienen dos motores de inferencia: El motor E_1 , cuya base de conocimiento contiene las siete reglas de la Figura 2.1, y el

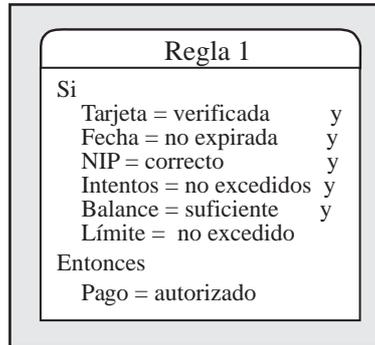


FIGURA 2.5. Regla 1 tomada de la Figura 2.1.

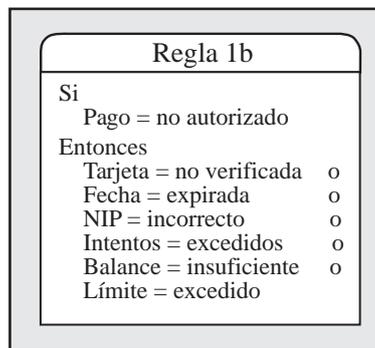


FIGURA 2.6. La Regla 1b puede obtenerse de la Regla 1 utilizando la regla de inferencia Modus Tollens.

motor E_2 , cuya base de conocimiento contiene las siete reglas de la Figura 2.7. Nótese que los dos conjuntos de reglas son lógicamente equivalentes. Supóngase además que se sabe que el valor de NIP es *incorrecto*. Si ambos E_1 y E_2 utilizan sólo la regla de inferencia Modus Ponens, entonces E_1 será capaz de concluir que $Pago = no\ autorizado$ (por la Regla 4), pero E_2 no concluirá. Por ello, algunas de las conclusiones lógicamente derivables pueden no ser obtenidas usando sólo la regla de inferencia Modus Ponens. Por otra parte, si ambos motores usan la regla Modus Tollens, entonces ambos concluirán. ■

2.3.2 El Mecanismo de Resolución

Las reglas de inferencia Modus Ponens y Modus Tollens pueden ser utilizadas para obtener conclusiones simples. Por otra parte, las conclusiones compuestas, que se basan en dos o más reglas, se obtienen usando el lla-

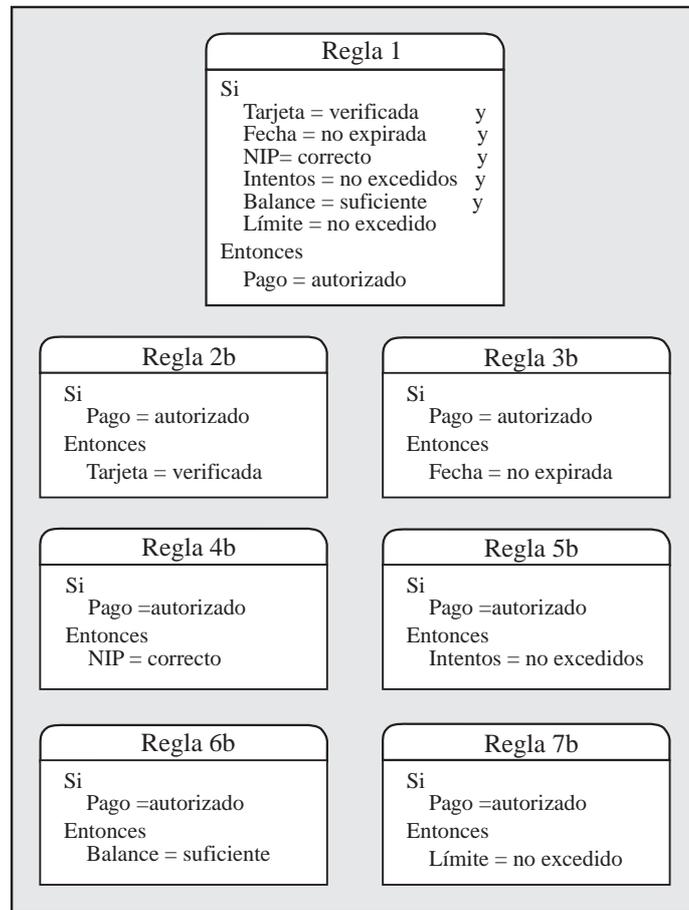


FIGURA 2.7. Un conjunto de reglas lógicamente equivalentes al conjunto de reglas de la Figura 2.1).

mado *mecanismo de resolución*. Esta regla de inferencia consiste en las etapas siguientes:

1. Las Reglas son sustituidas por expresiones lógicas equivalentes.
2. Estas expresiones lógicas se combinan en otra expresión lógica.
3. Esta última expresión se utiliza para obtener la conclusión.

Estas etapas involucran conceptos tales como la combinación y simplificación de expresiones lógicas, que se ilustran de un modo intuitivo en los ejemplos que siguen. Para un tratamiento detallado de esta regla de inferencia el lector puede consultar alguno de los libros específicos citados en la introducción de este capítulo.

A	B	\bar{A}	Si A , entonces B	$\bar{A} \text{ o } B$
C	C	F	C	C
C	F	F	F	F
F	C	C	C	C
F	F	C	C	C

TABLA 2.6. Una tabla de verdad mostrando que la regla “Si A es cierto, entonces B es cierto” es equivalente a la expresión lógica “ A es falso o B es cierto.”

Ejemplo 2.5 Mecanismo de resolución 1. Supóngase que se tienen las dos reglas:

- Regla 1: Si A es cierto, entonces B es cierto.
- Regla 2: Si B es cierto, entonces C es cierto.

La primera etapa en el mecanismo de resolución consiste en sustituir cada una de las dos reglas por expresiones lógicas equivalentes. Esto se hace como sigue (véase la Figura 2.8):

- La Regla 1 es equivalente a la expresión lógica: “ A es falso o B es cierto.” Una prueba de esta equivalencia se muestra en la tabla de verdad que se muestra en la Tabla 2.6.
- Similarmente, la Regla 2 es equivalente a la expresión lógica: “ B es falso o C es cierto.”

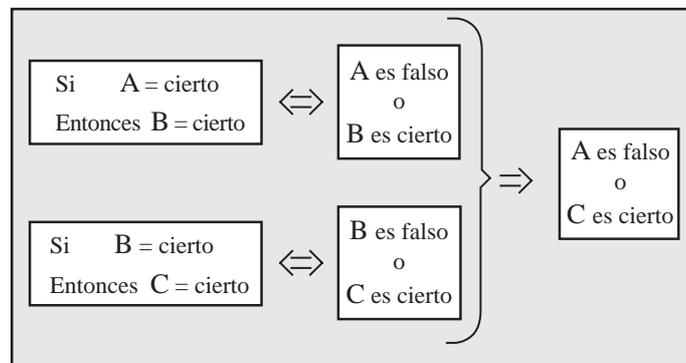


FIGURA 2.8. Un ejemplo ilustrando la regla de inferencia correspondiente al mecanismo de resolución.

La segunda etapa consiste en combinar las dos expresiones anteriores en una, tal como sigue: Las expresiones lógicas “ A es falso o B es cierto” y “ B es falso o C es cierto” implican la expresión “ A es falso o C es

A	B	C	$\bar{A} \circ B$	$\bar{B} \circ C$	$(\bar{A} \circ B)$ y $(\bar{B} \circ C)$	$\bar{A} \circ C$
C	C	C	C	C	C	C
C	C	F	C	F	F	F
C	F	C	F	C	F	C
C	F	F	F	C	F	F
F	C	C	C	C	C	C
F	C	F	C	F	F	C
F	F	C	C	C	C	C
F	F	F	C	C	C	C

TABLA 2.7. Una tabla de verdad que muestra que las expresiones lógicas “ A es falso o B es cierto” y “ B es falso o C es cierto” implican la expresión lógica “ A es falso o C es cierto.”

cierto.” Una prueba de esta equivalencia se muestra en la Tabla 2.7. Esta última expresión se utiliza seguidamente en la tercera etapa para obtener la conclusión. Las etapas anteriores se ilustran en la Figura 2.8. ■

Ejemplo 2.6 Mecanismo de resolución 2. Considérese de nuevo el ejemplo del CA con el objeto añadido *Explicar*, que puede tomar los valores {sí, no}, indicando si se necesita explicar las acciones del CA. Apliquemos ahora el mecanismo de resolución a la evidencia $NIP = incorrecto$ y a las reglas siguientes:

- Si $NIP = incorrecto$ entonces $Pago = no\ autorizado$.
- Si $Pago = no\ autorizado$ entonces $Explicar = sí$.

Tal como se ilustra en la Figura 2.9, la regla de inferencia correspondiente al mecanismo de resolución conduce a la conclusión $Explicar = sí$. En efecto, siguiendo los pasos indicados, se tiene

1. Las dos reglas se sustituyen por las expresiones equivalentes:
 - $NIP = correcto$ o $Pago = no\ autorizado$
 - $Pago = autorizado$ o $Explicar = sí$
2. Las dos expresiones anteriores se combinan de la forma indicada para dar la expresión $NIP = correcto$ o $Explicar = sí$, y
3. Esta última expresión se combina con la evidencia $NIP = incorrecto$, y se obtiene la conclusión compuesta, $Explicar = sí$. ■

Es importante señalar que la regla de inferencia correspondiente al mecanismo de resolución no siempre conduce a conclusiones, pues, de hecho,

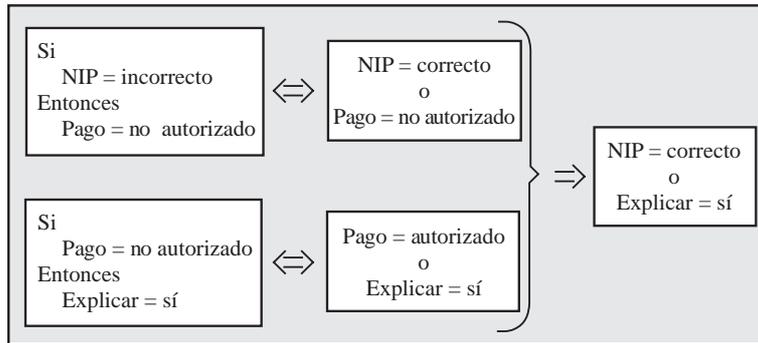


FIGURA 2.9. La regla de inferencia del mecanismo de resolución aplicada al ejemplo del CA.

puede no conocerse la verdad o falsedad de ciertas expresiones. Si esto ocurre, el sistema experto, o más precisamente, su motor de inferencia, debe decidir entre:

- Abandonar la regla, dada la imposibilidad de obtener conclusiones, o
- Preguntar al usuario, mediante el subsistema de demanda de información, sobre la verdad o falsedad de una o varias expresiones para poder continuar el proceso de inferencia hasta que se obtenga una conclusión.

2.3.3 Encadenamiento de Reglas

Una de las estrategias de inferencia más utilizadas para obtener conclusiones compuestas es el llamado *encadenamiento de reglas*. Esta estrategia puede utilizarse cuando las premisas de ciertas reglas coinciden con las conclusiones de otras. Cuando se encadenan las reglas, los hechos pueden utilizarse para dar lugar a nuevos hechos. Esto se repite sucesivamente hasta que no pueden obtenerse más conclusiones. El tiempo que consume este proceso hasta su terminación depende, por una parte, de los hechos conocidos, y, por otra, de las reglas que se activan. La estrategia de encadenamiento de reglas se da en el algoritmo siguiente:

Algoritmo 2.1 Encadenamiento de reglas.

- **Datos:** Una base de conocimiento (objetos y reglas) y algunos hechos iniciales.
 - **Resultado:** El conjunto de hechos derivados lógicamente de ellos.
1. Asignar a los objetos sus valores conocidos tales como los dan los hechos conocidos o la evidencia

2. Ejecutar cada regla de la base de conocimiento y concluir nuevos hechos si es posible.
3. Repetir la Etapa 2 hasta que no puedan ser obtenidos nuevos hechos. ■

Este algoritmo puede ser implementado de muchas formas. Una de ellas comienza con las reglas cuyas premisas tienen valores conocidos. Estas reglas deben concluir y sus conclusiones dan lugar a nuevos hechos. Estos nuevos hechos se añaden al conjunto de hechos conocidos, y el proceso continúa hasta que no pueden obtenerse nuevos hechos. Este proceso se ilustra, a continuación, con dos ejemplos.

Ejemplo 2.7 Encadenamiento de Reglas 1. La Figura 2.10 muestra un ejemplo de seis reglas que relacionan 13 objetos, del A al M . Las relaciones entre estos objetos implicadas por las seis reglas pueden representarse gráficamente, tal como se muestra en la Figura 2.11, donde cada objeto se representa por un nodo. Las aristas representan la conexión entre los objetos de la premisa de la regla y el objeto de su conclusión. Nótese que las premisas de algunas reglas coinciden con las conclusiones de otras reglas. Por ejemplo, las conclusiones de las Reglas 1 y 2 (objetos C y G) son las premisas de la Regla 4.

Supóngase que se sabe que los objetos A, B, D, E, F, H e I son *ciertos* y los restantes objetos son de valor desconocido. La Figura 2.12 distingue entre objetos con valor conocido (los hechos) y objetos con valores desconocidos. En este caso, el algoritmo de encadenamiento de reglas procede como sigue:

- La Regla 1 concluye que $C = \text{cierto}$.
- La Regla 2 concluye que $G = \text{cierto}$.
- La Regla 3 concluye que $J = \text{cierto}$.
- La Regla 4 concluye que $K = \text{cierto}$.
- La Regla 5 concluye que $L = \text{cierto}$.
- La Regla 6 concluye que $M = \text{cierto}$.

Puesto que no pueden obtenerse más conclusiones, el proceso se detiene. Este proceso se ilustra en la Figura 2.12, donde los números en el interior de los nodos indican el orden en el que se concluyen los hechos. ■

Ejemplo 2.8 Encadenamiento de reglas 2. Considérense de nuevo las seis reglas de la Figura 2.10 y supóngase ahora que se dan los hechos $H = \text{cierto}$, $I = \text{cierto}$, $K = \text{cierto}$ y $M = \text{falso}$. Esto se ilustra en la Figura 2.13, donde los objetos con valores conocidos (los hechos) aparecen

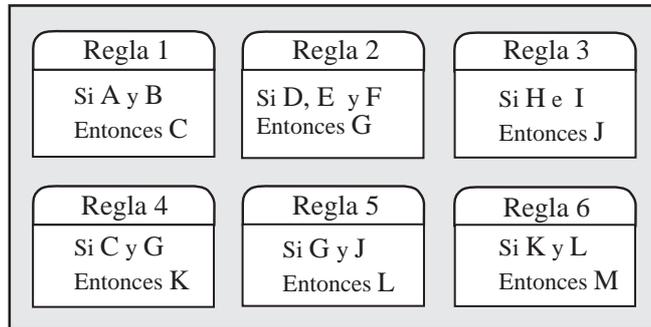


FIGURA 2.10. Un ejemplo de un conjunto de seis reglas relacionando 13 objetos.

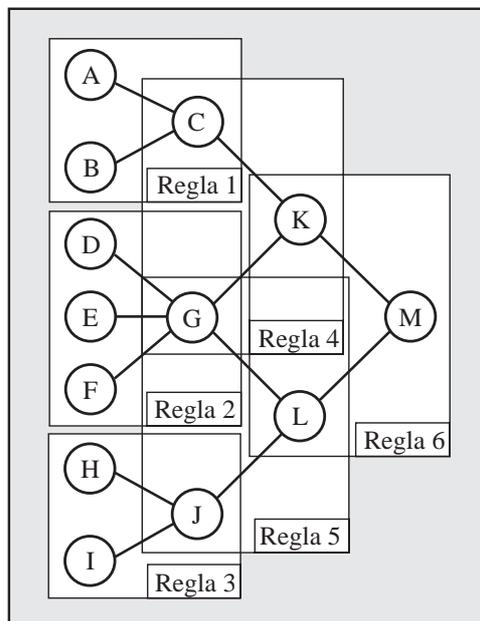


FIGURA 2.11. Una representación gráfica de las relaciones entre las seis reglas de la Figura 2.10.

sombreados y la variable objetivo se muestra rodeada por una circunferencia. Supóngase, en primer lugar, que el motor de inferencia usa las dos reglas de inferencia Modus Ponens y Modus Tollens. Entonces, aplicando el Algoritmo 2.1, se obtiene

1. La Regla 3 concluye que $J = \text{cierto}$ (Modus Ponens).
2. La Regla 6 concluye (Modus Tollens) que $K = \text{falso}$ o $L = \text{falso}$, pero, puesto que $K = \text{cierto}$, deberá ser $L = \text{falso}$.

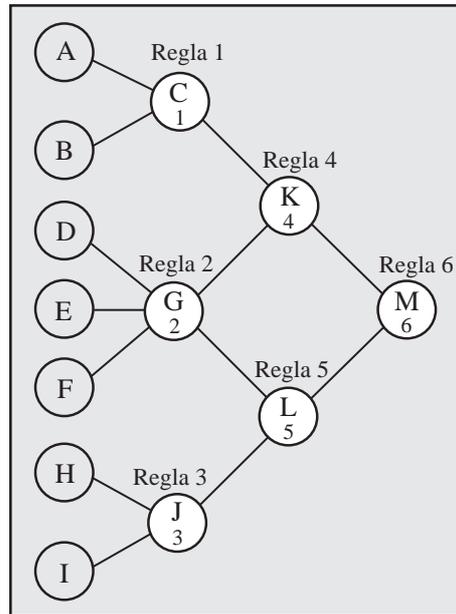


FIGURA 2.12. Un ejemplo que ilustra la estrategia de encadenamiento de reglas. Los nodos con valores conocidos aparecen sombreados y los números en su interior indican el orden en el que se concluyen los hechos.

3. La Regla 5 concluye (Modus Tollens) que $G = falso$ o $J = falso$, pero, puesto que $J = cierto$, deberá ser $G = falso$.

En consecuencia, se obtiene la conclusión $G = falso$. Sin embargo, si el motor de inferencia sólo utiliza la regla de inferencia Modus Ponens, el algoritmo se detendrá en la Etapa 1, y no se concluirá nada para el objeto G . Este es otro ejemplo que ilustra la utilidad de la regla de inferencia Modus Tollens. ■

Nótese que la estrategia de encadenamiento de reglas diferencia claramente entre la memoria de trabajo y la base de conocimiento. La memoria de trabajo contiene datos que surgen durante el periodo de consulta. Las premisas de las reglas se comparan con los contenidos de la memoria de trabajo y cuando se obtienen nuevas conclusiones son pasadas también a la memoria de trabajo.

2.3.4 Encadenamiento de Reglas Orientado a un Objetivo

El algoritmo de encadenamiento de reglas orientado a un objetivo requiere del usuario seleccionar, en primer lugar, una variable o nodo objetivo; entonces el algoritmo navega a través de las reglas en búsqueda de una conclusión para el nodo objetivo. Si no se obtiene ninguna conclusión con la

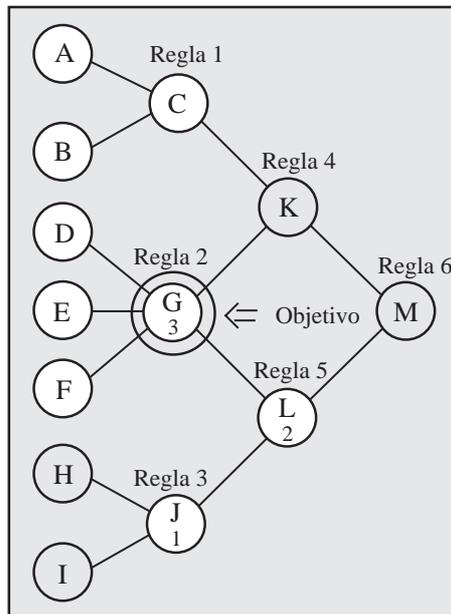


FIGURA 2.13. Otro ejemplo que ilustra el algoritmo de encadenamiento de reglas. Los nodos con valores conocidos aparecen sombreados, la variable objetivo se muestra rodeada por una circunferencia, y los números del interior de los nodos indican el orden en el que se concluyen los hechos.

información existente, entonces el algoritmo fuerza a preguntar al usuario en busca de nueva información sobre los elementos que son relevantes para obtener información sobre el objetivo.

Algunos autores llaman a los algoritmos de encadenamiento y de encadenamiento orientado a un objetivo *encadenamiento hacia adelante* y *encadenamiento hacia atrás*, respectivamente. Pero esta terminología puede ser confusa, puesto que ambos algoritmos pueden, en realidad, utilizar las dos reglas de inferencia Modus Ponens (hacia adelante) y Modus Tollens (hacia atrás).

El algoritmo de encadenamiento de reglas orientado a un objetivo se describe a continuación.

Algoritmo 2.2 Encadenamiento de reglas orientado a un objetivo.

- **Datos:** Una base de conocimiento (objetos y reglas), algunos hechos iniciales, y un nodo o variable objetivo.
 - **Resultado:** El valor del nodo o variable objetivo.
1. Asigna a los objetos sus valores conocidos tales como están dados en los hechos de partida, si es que existe alguno. Marcar todos los objetos

cuyo valor ha sido asignado. Si el nodo objetivo está marcado, ir a la Etapa 7; en otro caso:

- (a) Designar como objetivo *inicial* el objetivo *en curso*.
 - (b) Marcar el objetivo en curso.
 - (c) Sea $ObjetivosPrevios = \phi$, donde ϕ es el conjunto vacío.
 - (d) Designar todas las reglas como activas (ejecutables).
 - (e) Ir a la Etapa 2.
2. Encontrar una regla activa que incluya el objetivo en curso y ninguno de los objetos en *ObjetivosPrevios*. Si se encuentra una regla, ir a la Etapa 3; en otro caso, ir a la Etapa 5.
 3. Ejecutar la regla referente al objetivo en curso. Si concluye, asignar el valor concluido al objetivo en curso, e ir a la Etapa 6; en otro caso, ir a la Etapa 4.
 4. Si todos los objetos de la regla están marcados, declarar la regla como *inactiva* e ir a la Etapa 2; en otro caso:
 - (a) Añadir el objetivo en curso a *ObjetivosPrevios*.
 - (b) Designar uno de los objetos no marcados en la regla como el objetivo en curso.
 - (c) Marcar el objetivo en curso.
 - (d) Ir a la Etapa 2.
 5. Si el objetivo en curso es el mismo que el objetivo inicial, ir a la Etapa 7; en otro caso, preguntar al usuario por el valor del objetivo en curso. Si no se da un valor, ir a la Etapa 6; en otro caso asignar al objeto el valor dado e ir a la Etapa 6.
 6. Si el objetivo en curso es el mismo que el objetivo inicial, ir a la Etapa 7; en otro caso, designar el objetivo previo como objetivo en curso, eliminarlo de *ObjetivosPrevios*, e ir a la Etapa 2.
 7. Devolver el valor del objetivo en curso si es conocido. ■

A continuación se ilustra el encadenamiento de reglas orientado a un objetivo mediante algunos ejemplos.

Ejemplo 2.9 Encadenamiento de reglas orientado a un objetivo.

Considérense las seis reglas de las Figuras 2.10 y 2.11. Supóngase que se selecciona el nodo M como nodo objetivo y que se sabe que los objetos D, E, F y L son ciertos. Estos nodos están sombreados en la Figura 2.14. Las etapas del algoritmo de encadenamiento de reglas orientado a un objetivo se ilustran en la Figura 2.14, donde el número en el interior de un nodo indica el orden en el que se visita cada nodo. Estas etapas son:

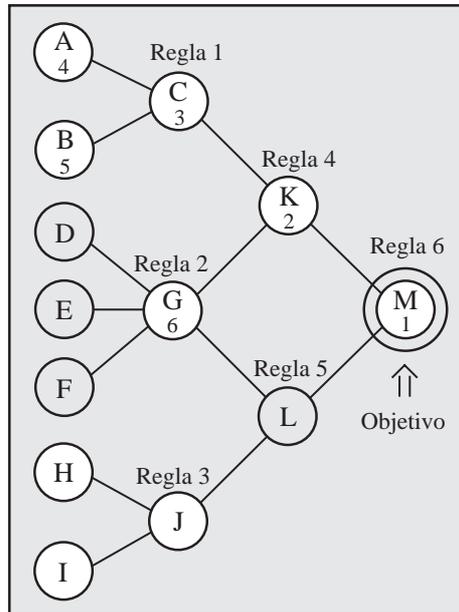


FIGURA 2.14. Un ejemplo que ilustra el algoritmo de encadenamiento de reglas orientado a un objetivo. Los nodos cuyo valor es conocido se han sombreado, el nodo objetivo se ha rodeado por una circunferencia, y el número en el interior de un nodo indica el orden en el que se visita cada nodo.

- Etapa 1: Se asigna el valor *cierto* a los objetos D, E, F y L y se marcan. Puesto que el nodo objetivo M no está marcado, entonces
 - Se designa el objeto M como objeto *en curso*.
 - Se marca el objeto M . Por tanto, se tiene $ObjetosMarcados = \{D, E, F, L, M\}$.
 - $ObjetivosPrevios = \phi$.
 - Las seis reglas están activas. Por tanto, se tiene $ReglasActivas = \{1, 2, 3, 4, 5, 6\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla que incluya el objetivo en curso M . Se encuentra la Regla 6, por lo que se va a la Etapa 3.
- Etapa 3. La Regla 6 no puede concluir puesto que el valor del objeto K es desconocido. Así que se va a la Etapa 4.
- Etapa 4. El objeto K no está marcado. Entonces
 - $ObjetivosPrevios = \{M\}$.

- Se elige el objeto K como objetivo en curso.
- El objeto K está marcado. Por tanto se tiene, $ObjetosMarcados = \{D, E, F, L, M, K\}$.
- Se va a la Etapa 2.
- Etapa 2. Se busca una regla que incluya el objetivo en curso K pero no el anterior M . Se encuentra la Regla 4, y se continúa con la Etapa 3.
- Etapa 3. La Regla 4 no puede concluir puesto que se desconocen los valores de los objetos C y G . Por ello, se continúa con la Etapa 4.
- Etapa 4. Los objetos C y G no están marcados. Entonces
 - $ObjetivosPrevios = \{M, K\}$.
 - Se elige uno de los objetos no marcados C o G como el nuevo objetivo en curso. Supóngase que se elige C .
 - Se marca el objeto C .
Por tanto, se tiene $ObjetosMarcados = \{D, E, F, L, M, K, C\}$.
 - Se continúa con la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso C pero no los objetos previos $\{M, K\}$. Se encuentra la Regla 1, así que se va a la Etapa 3.
- Etapa 3. La Regla 1 no puede concluir puesto que se desconocen los valores de los objetos A y B . Por tanto se continúa con la Etapa 4.
- Etapa 4. Los objetos A y B no están marcados. Entonces
 - $ObjetivosPrevios = \{M, K, C\}$.
 - Se elige uno de los objetos no marcados A y B como nuevo objetivo en curso. Supóngase que se elige A .
 - Se marca el objeto A .
Por ello, $ObjetosMarcados = \{D, E, F, L, M, K, C, A\}$.
 - Se continúa con la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso A pero no los objetivos previos $\{M, K, C\}$. No se encuentra ninguna regla que satisfaga estas condiciones, así que se pasa a la Etapa 5.
- Etapa 5. Puesto que el objetivo en curso A es diferente del inicial M , se pregunta al usuario por el valor del objeto A . Supóngase que A toma el valor cierto, entonces se hace $A = cierto$ y se sigue con la Etapa 6.

- Etapa 6. El objetivo en curso A no coincide con el previo M . Por tanto, el objeto C se designa como objetivo en curso y se elimina de la lista *ObjetivosPrevios*. Por ello, $ObjetivosPrevios = \{M, K\}$ y se continúa con la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo C pero no los anteriores $\{M, K\}$. Se encuentra la Regla 1, por lo que se va a la Etapa 3.
- Etapa 3. La Regla 1 no puede concluir porque el valor del objeto B es desconocido. Así que se va a la Etapa 4.
- Etapa 4. El objeto B no está marcado. Entonces
 - $ObjetivosPrevios = \{M, K, C\}$.
 - Se elige como objetivo en curso el único objeto no marcado, B .
 - Se marca el objeto B .
 - Por ello, $ObjetosMarcados = \{D, E, F, L, M, K, C, A, B\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo B pero no los objetivos previos $\{M, K, C\}$. Como no se encuentra ninguna regla, se va a la Etapa 5.
- Etapa 5. Puesto que el objetivo en curso B no coincide con el inicial M , se pregunta al usuario el valor del objetivo en curso B . Supóngase que se da un valor cierto a B , entonces se hace $B = cierto$ y se va a la Etapa 6.
- Etapa 6. Como el objetivo en curso B no coincide con el inicial M , se designa el objetivo previo C como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello, $ObjetivosPrevios = \{M, K\}$ y se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso C pero no los anteriores $\{M, K\}$. Se encuentra la Regla 1, por lo que se va a la Etapa 3.
- Etapa 3. Puesto que $A = cierto$ y $B = cierto$, entonces $C = cierto$ por la Regla 1. Ahora se va a la Etapa 6.
- Etapa 6. El objetivo en curso C no coincide con el inicial M . Entonces, se designa el objetivo previo K como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello, $ObjetivosPrevios = \{M\}$ y se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso K pero no los anteriores $\{M\}$. Se encuentra la Regla 4, por lo que se va a la Etapa 3.

- Etapa 3. La Regla 4 no puede concluir puesto que el valor del objeto G es desconocido. Por tanto, se va a la Etapa 4.
- Etapa 4. El objeto G no está marcado. Entonces
 - $ObjetivosPrevios = \{M, K\}$.
 - El único objeto no marcado G se elige como objetivo en curso.
 - Se marca el objeto G .
Por ello, $ObjetosMarcados = \{D, E, F, L, M, K, C, A, B, G\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso G pero no los anteriores $\{M, K\}$. Se encuentra la Regla 2, por lo que se va a la Etapa 3.
- Etapa 3. Puesto que $D = cierto$, $E = cierto$ y $F = cierto$, entonces $G = cierto$ por la Regla 2. Ahora se va a la Etapa 6.
- Etapa 6. El objetivo en curso G no coincide con el inicial M . Entonces, se designa el objetivo previo K como objetivo en curso y se elimina de $ObjetivosPrevios$. Por ello, $ObjetivosPrevios = \{M\}$ y se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso K pero no los anteriores $\{M\}$. Se encuentra la Regla 4, por lo que se va a la Etapa 3.
- Etapa 3. Puesto que $C = cierto$ y $G = cierto$, entonces $K = cierto$ por la Regla 4. Seguidamente se va a la Etapa 6.
- Etapa 6. El objetivo en curso K no coincide con el inicial M . Entonces, se designa el objetivo previo M como objetivo en curso y se elimina de $ObjetivosPrevios$. Por ello, $ObjetivosPrevios = \phi$ y se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso M . Se encuentra la Regla 6, por lo que se va a la Etapa 3.
- Etapa 3. Puesto que $K = cierto$ y $L = cierto$, entonces $M = cierto$ por la Regla 6. Ahora se va a la Etapa 6.
- Etapa 6. El objetivo en curso M coincide con el inicial. En consecuencia, se va a la Etapa 7.
- Etapa 7. El algoritmo devuelve el valor $M = cierto$. ■

Nótese que a pesar de que los objetos H, I y J tienen valores desconocidos, el algoritmo orientado a un objetivo ha sido capaz de concluir el valor del objetivo M . La razón de este resultado está en que el conocimiento del objeto L convierte al conocimiento de los objetos H, I y J es irrelevante para el conocimiento del objeto M . ■

Las estrategias de encadenamiento de reglas se utilizan en problemas en los que algunos hechos (por ejemplo, síntomas) se dan por conocidos y se buscan algunas conclusiones (por ejemplo, enfermedades). Por el contrario, las estrategias de encadenamiento de reglas orientadas a un objetivo se utilizan en problemas en los que se dan algunos objetivos (enfermedades) y se buscan los hechos (síntomas) para que éstas sean posibles.

Ejemplo 2.10 Encadenamiento de reglas orientado a un objetivo sin Modus Tollens. Considérense las seis reglas de las Figuras 2.10 y 2.11. Supóngase que se selecciona el nodo J como objetivo y que se dan los siguientes hechos: $G = \text{cierto}$ y $L = \text{falso}$. Esto se ilustra en la Figura 2.15, donde los objetos con valores conocidos (los hechos) aparecen sombreados y el objetivo rodeado por una circunferencia. Supóngase, en primer lugar, que el motor de inferencia utiliza sólo la regla de inferencia Modus Ponens. En este caso, las etapas del Algoritmo 2.2 son como sigue:

- Etapa 1: Se marcan los objetos G y L y se les asignan los valores $G = \text{cierto}$ y $L = \text{falso}$. Puesto que el objetivo J no está marcado, entonces
 - Se designa el objeto J como objetivo *en curso*.
 - Se marca el objeto J . Por ello, $\text{ObjetosMarcados} = \{G, L, J\}$.
 - $\text{ObjetivosPrevios} = \phi$.
 - Todas las reglas están activas. Por tanto, se tiene $\text{ReglasActivas} = \{1, 2, 3, 4, 5, 6\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso J . Puesto que sólo se utiliza la regla de inferencia Modus Ponens, se encuentra la Regla 3 (es la única regla en la que el objetivo en curso J forma parte de su conclusión). Por tanto, se va a la Etapa 3.
- Etapa 3. La Regla 3 no puede concluir puesto que los valores de los objetos H e I son desconocidos. Por tanto, se va a la Etapa 4.
- Etapa 4. Los objetos H y I no están marcados. Entonces
 - $\text{ObjetivosPrevios} = \{J\}$.
 - Se elige uno de los objetos no marcados H o I como objetivo en curso. Supóngase que se elige H .

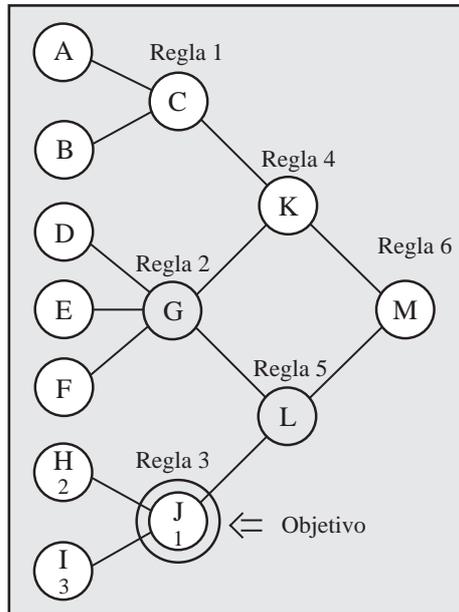


FIGURA 2.15. Un ejemplo que ilustra el encadenamiento de reglas orientado a un objetivo sin Modus Tollens. Los objetos con valores conocidos (los hechos) aparecen sombreados, el objetivo se muestra rodeado por una circunferencia, y el número en el interior de cada nodo indica el orden en el que se visita.

- El objeto H está marcado. Por tanto, se tiene $ObjetosMarcados = \{G, L, J, H\}$.
- Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso H pero no el previo J . No se encuentra tal regla, por lo que se va a la Etapa 5.
- Etapa 5. Puesto que el objetivo en curso H es diferente del objetivo inicial J , se pregunta al usuario el valor del objetivo en curso H . Supóngase que no se da un valor para dicho objeto, entonces se va a la Etapa 6.
- Etapa 6. El objetivo en curso H no coincide con el objetivo inicial J . Entonces, se designa el objetivo previo J como objetivo en curso y se elimina de $ObjetivosPrevios$. Por ello, $ObjetivosPrevios = \phi$ y se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso J . Se encuentra la Regla 3 y se va a la Etapa 3.

- Etapa 3. La Regla 3 no puede concluir puesto que se desconocen los valores de los objetos H e I . Por tanto, se va a la Etapa 4.
- Etapa 4. El objeto I no está marcado. Entonces
 - $ObjetivosPrevios = \{J\}$.
 - Se elige el único objeto no marcado I como el objetivo en curso.
 - Se marca el objeto I . Por ello, $ObjetosMarcados = \{G, L, J, H, I\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso I pero no el previo J . Como no se encuentra tal regla, se va a la Etapa 5.
- Etapa 5. Puesto que el objetivo en curso I es diferente del objetivo inicial J , se pregunta al usuario el valor del objetivo en curso I . Supóngase que no se da un valor para el objeto I , entonces, se va a la Etapa 6.
- Etapa 6. El objetivo en curso I no es el mismo que el inicial J . Por tanto, se designa el objetivo previo J como objetivo en curso y se elimina de la lista $ObjetivosPrevios$. Por ello, se hace $ObjetivosPrevios = \phi$ y se vuelve a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso J . Se encuentra la Regla 3, por lo que se va a la Etapa 3.
- Etapa 3. La Regla 3 no puede concluir puesto que no se conocen los valores de los objetos H e I . Se va a la Etapa 4.
- Etapa 4. Todos los objetos de la Regla 3 están marcados, por lo que la Regla 3 se declara inactiva. Por ello, se hace $ReglasActivas = \{1, 2, 4, 5, 6\}$. Se continúa en la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso J . Puesto que la Regla 3 se ha declarado inactiva y no se utiliza la regla de inferencia Modus Tollens, no se encuentra ninguna regla y se va a la Etapa 5.
- Etapa 5. Puesto que el objetivo en curso J coincide con el inicial, se va a la Etapa 7.
- Etapa 7. El motor de inferencia no puede concluir un valor para el objetivo J . ■

Ejemplo 2.11 Encadenamiento de reglas orientado a un objetivo con Modus Tollens. En el ejemplo 2.10, el motor de inferencia utiliza sólo la regla de inferencia Modus Ponens. Considérense las mismas hipótesis

del Ejemplo 2.10 excepto que ahora el motor de inferencia utiliza ambas reglas de inferencia, la Modus Ponens y la Modus Tollens. Las etapas del Algoritmo 2.2 en este caso son como sigue:

- Etapa 1: Se marcan los objetos G y L y se les asignan los valores $G = \text{cierto}$ y $L = \text{falso}$. Puesto que el objetivo J no está marcado, entonces
 - Se designa el objeto J como objetivo *en curso*.
 - Se marca el objeto J . Por ello, se hace $\text{ObjetosMarcados} = \{G, L, J\}$.
 - $\text{ObjetivosPrevios} = \phi$.
 - Las seis reglas son activas. Por ello, se hace $\text{ReglasActivas} = \{1, 2, 3, 4, 5, 6\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo en curso J . Puesto que se utilizan ambas reglas de inferencia, las Reglas 3 y 5 incluyen el objeto J . Supóngase que se elige la Regla 5. Se va a la Etapa 3. (Si se elige la Regla 3, el algoritmo tardará más tiempo en terminar.)
- Etapa 3. Puesto que se usa la regla de inferencia Modus Tollens, la Regla 5 concluye que $J = \text{falso}$ (puesto que $G = \text{cierto}$ y $L = \text{falso}$). Por tanto, se va a la Etapa 6.
- Etapa 6. El objetivo en curso J coincide con el inicial, por lo que se va a la Etapa 7.
- Etapa 7. Se devuelve $J = \text{falso}$. ■

2.3.5 Compilación de reglas

Otra forma de tratar con reglas encadenadas consiste en comenzar con un conjunto de datos (información) y tratar de alcanzar algunos objetivos. Esto se conoce con el nombre de *compilación de reglas*. Cuando ambos, datos y objetivos, se han determinado previamente, las reglas pueden ser compiladas, es decir, pueden escribirse los objetivos en función de los datos para obtener las llamadas *ecuaciones objetivo*. La compilación de reglas se explica mejor con un ejemplo.

Ejemplo 2.12 Compilación de reglas. Considérese el conjunto de seis reglas de la Figura 2.11 y supóngase que son conocidos los valores de los objetos A, B, D, E, F, H , e I y que los restantes objetos, C, G, J, K, L y M , son objetivos. Denotemos por \wedge el operador lógico *y*; entonces, utilizando las seis reglas, pueden obtenerse las siguientes ecuaciones objetivo:

- La Regla 1 implica $C = A \wedge B$.
- La Regla 2 implica $G = D \wedge E \wedge F$.
- La Regla 3 implica $J = H \wedge I$.
- La Regla 4 implica $K = C \wedge G = (A \wedge B) \wedge (D \wedge E \wedge F)$.
- La Regla 5 implica $L = G \wedge J = (D \wedge E \wedge F) \wedge (H \wedge I)$.
- La Regla 6 implica $M = K \wedge L = A \wedge B \wedge D \wedge E \wedge F \wedge H \wedge I$.

Las tres primeras ecuaciones son equivalentes a las tres primeras reglas. Las tres ecuaciones objetivo son, respectivamente, equivalentes a las reglas siguientes:

- Regla 4a: *Si A y B y D y E y F, entonces K.*
- Regla 5a: *Si D y E y F y H e I, entonces L.*
- Regla 6a: *Si A y B y D y E y F y H e I, entonces M.*

Por ello, si, por ejemplo, cada uno de los objetos $\{A, B, D, E, F, H, I\}$ toma el valor cierto, entonces se obtiene de forma inmediata, a partir de las Reglas 4a, 5a y 6a, que los objetos $\{K, L, M\}$ deben ser ciertos. ■

2.4 Control de la Coherencia

En situaciones complejas, incluso verdaderos expertos pueden dar información inconsistente (por ejemplo, reglas inconsistentes y/o combinaciones de hechos no factibles). Por ello, es muy importante controlar la coherencia del conocimiento tanto durante la construcción de la base de conocimiento como durante los procesos de adquisición de datos y razonamiento. Si la base de conocimiento contiene información inconsistente (por ejemplo, reglas y/o hechos), es muy probable que el sistema experto se comporte de forma poco satisfactoria y obtenga conclusiones absurdas.

El objetivo del control de la coherencia consiste en

1. Ayudar al usuario a no dar hechos inconsistentes, por ejemplo, dándole al usuario las restricciones que debe satisfacer la información demandada.
2. Evitar que entre en la base de conocimiento cualquier tipo de conocimiento inconsistente o contradictorio.

El control de la coherencia debe hacerse controlando la coherencia de las reglas y la de los hechos.

Objetos		Conclusiones		Conclusiones contradictorias
A	B	Regla 1	Regla 2	
C	C	$B = C$	$B = F$	Sí
C	F	$B = C$	$B = F$	Sí
F	C	–	–	No
F	F	–	–	No

TABLA 2.8. Una tabla de verdad que muestra que las Reglas 1 y 2 son coherentes.

2.4.1 Coherencia de Reglas

Definición 2.2 Reglas coherentes. *Un conjunto de reglas se denomina coherente si existe, al menos, un conjunto de valores de todos los objetos que producen conclusiones no contradictorias.*

En consecuencia, un conjunto coherente de reglas no tiene por qué producir conclusiones no contradictorias para todos los posibles conjuntos de valores de los objetos. Es decir, es suficiente que exista un conjunto de valores que conduzcan a conclusiones no contradictorias.

Ejemplo 2.13 Conjunto de reglas incoherentes. Considérense las cuatro reglas siguientes, que relacionan dos objetos A y B binarios $\{C, F\}$:

- Regla 1: Si $A = C$, entonces $B = C$.
- Regla 2: Si $A = C$, entonces $B = F$.
- Regla 3: Si $A = F$, entonces $B = C$.
- Regla 4: Si $A = F$, entonces $B = F$.

Entonces, pueden obtenerse las siguientes conclusiones:

1. Las Reglas 1–2 son coherentes puesto que, tal como se muestra en la Tabla 2.8, para $A = F$, no producen conclusiones.
2. Las Reglas 1–3 son coherentes puesto que para $A = F$ y $B = C$, producen una conclusión ($B = C$) (véase la Tabla 2.9).
3. Las Reglas 1–4 son incoherentes porque producen conclusiones contradictorias para todos los posibles valores de A y B , tal como se ve en la Tabla 2.10. ■

Nótese que un conjunto de reglas puede ser coherente, aunque algunos conjuntos de valores puedan producir conclusiones inconsistentes. Estos conjuntos de valores se llaman *valores no factibles*. Por ejemplo, las Reglas 1–2 son coherentes, aunque producen conclusiones inconsistentes en todos los casos en que $A = C$. En consecuencia, el subsistema de control de

Objetos		Conclusiones			Conclusiones contradictorias
A	B	Regla 1	Regla 2	Regla 3	
C	C	$B = C$	$B = F$	–	Sí
C	F	$B = C$	$B = F$	–	Sí
F	C	–	–	$B = C$	No
F	F	–	–	$B = C$	Sí

TABLA 2.9. Una tabla de verdad que muestra que las Reglas 1–3 son coherentes.

Objetos		Conclusiones				Conclusiones contradictorias
A	B	Regla 1	Regla 2	Regla 3	Regla 4	
C	C	$B = C$	$B = F$	–	–	Sí
C	F	$B = C$	$B = F$	–	–	Sí
F	C	–	–	$B = C$	$B = F$	Sí
F	F	–	–	$B = C$	$B = F$	Sí

TABLA 2.10. Una tabla de verdad que muestra que las Reglas 1–4 son incoherentes.

coherencia eliminará automáticamente el valor C de la lista de posibles valores del objeto A , permitiendo de esta forma al usuario seleccionar sólo valores factibles de los objetos.

Definición 2.3 Valores no factibles. *Se dice que un valor a para el objeto A no es factible si las conclusiones obtenidas al hacer $A = a$ contradicen cualquier combinación de valores del resto de los objetos.*

Por ello, cualquier valor no factible debe ser eliminado de la lista de valores posibles de su correspondiente objeto para eliminar la posibilidad de que el motor de inferencia pueda obtener conclusiones inconsistentes.

Ejemplo 2.14 Valores no factibles. Considérese el conjunto de las cuatro reglas del Ejemplo 2.13. En este caso, el motor de inferencia concluirá lo siguiente:

1. Las dos primeras reglas implican que $A \neq C$, puesto que $A = C$ siempre conduce a conclusiones inconsistentes. Por tanto, el valor $A = C$ deberá ser eliminado automáticamente de la lista de valores factibles de A . Dado que A es binario, entonces resulta $A = F$ (el único valor posible).
2. Las tres primeras reglas implican que $A = F$ y $B = C$. Por tanto, el valor $B = F$ deberá ser eliminado automáticamente de la lista de valores factibles de B .

3. Las primeras cuatro reglas implican que $A \neq C$, $A \neq F$, $B \neq C$ y $B \neq F$. Por tanto, los valores $\{C, F\}$ son eliminados de las listas de valores de A y B , con lo que las listas de valores factibles de todos los objetos están vacías, lo que implica que las cuatro reglas son incoherentes. ■

Nótese que es suficiente realizar la comprobación de la coherencia de las reglas sólo una vez, tras ser introducida cada regla, y que todos los valores no factibles pueden ser eliminados de sus correspondientes listas, nada más ser detectados.

El conjunto de reglas que forman el conocimiento debe ser coherente; en otro caso, el sistema podrá obtener conclusiones erróneas. Por ello, antes de añadir una regla a la base de conocimiento, hay que comprobar la consistencia de esta regla con el resto de ellas, incluidas en la base de conocimiento. Si la regla fuese consistente con el resto de reglas, se añadiría a la base de conocimiento; en caso contrario, se devolvería al experto humano para su corrección.

Ejemplo 2.15 Coherencia de reglas. Supóngase que se tienen los cuatro objetos: $A \in \{0, 1\}$, $B \in \{0, 1\}$, $C \in \{0, 1, 2\}$ y $D \in \{0, 1\}$. Considérense las cuatro reglas:

- Regla 1: Si $A = 0$ y $B = 0$, entonces $C = 0$.
- Regla 2: Si $A = 0$ y $D = 0$, entonces $C = 1$.
- Regla 3: Si $A = 0$ y $B = 0$, entonces $C = 1$.
- Regla 4: Si $A = 0$, entonces $B = 0$.
- Regla 5: Si $B = 0$, entonces $A = 1$.

Supóngase ahora que se desea añadir las tres últimas reglas a una base de conocimiento que contiene las dos primeras reglas. Entonces, las Reglas 1 y 3 son inconsistentes, puesto que tienen la misma premisa pero diferentes conclusiones. Por tanto, la Regla 3 debe ser rechazada y el experto humano informado de la razón del rechazo. El experto humano corregirá la regla en cuestión y/o las reglas existentes si fueran incorrectas. La Regla 4 entrará en la base de conocimiento, puesto que es consistente con las Reglas 1 y 2. La Regla 5 es inconsistente con la Regla 4. Por ello, la consistencia de ambas reglas debe ser comprobada antes de pasar a formar parte de la base de conocimiento. ■

2.4.2 Coherencia de hechos

Los datos o evidencias suministrados por los usuarios deben ser también consistentes en sí y con el conjunto de reglas de la base de datos. Por ello, el sistema no debe aceptar hechos que contradigan el conjunto de reglas y/o

el conjunto de hechos existente en cada instante del proceso. Por ejemplo, con una base de conocimiento que contenga las dos primeras reglas del Ejemplo 2.15, el sistema no debe aceptar el conjunto de hechos $A = 0$, $B = 0$ y $C = 1$ puesto que contradicen la Regla 1.

El sistema debe también comprobar si existe o no, una solución factible e informar al usuario en consecuencia. Si en el ejemplo anterior se trata de dar la información $A = 0$, $B = 0$ y $D = 0$, el sistema debe detectar que no existe ningún valor de C que sea consistente con la base de conocimiento. Nótese que antes de conocer los valores de los objetos, existe una solución factible. Por ejemplo, $A = 0$, $B = 0$, $C = 0$ y $D = 1$ (estos hechos no contradicen la base de conocimiento). Por ello, la inconsistencia surge de que los hechos y las reglas sean inconsistentes.

La coherencia de los hechos puede lograrse mediante las estrategias siguientes:

1. Eliminar todos los valores no factibles (los que contradicen el conjunto de reglas y/o hechos) de los objetos una vez detectados. Cuando se pregunte al usuario por información sobre los valores de un conjunto de objetos, el sistema experto debería aceptar sólo los valores de cada objeto que sean consistentes con las reglas y con el conocimiento previo. Considérese, por ejemplo, la base de conocimiento del Ejemplo 2.15 y supóngase que al sistema experto se le ha dado la información $A = 0$ y $C = 1$; entonces el sistema debe saber que $B \neq 0$. Por ello, este valor debe ser eliminado de la lista de posibles valores del objeto B .
2. El motor de inferencia debe comprobar que los hechos conocidos no contradicen el conjunto de reglas. En la situación anterior, por ejemplo, el sistema no debe aceptar el conjunto de hechos $A = 1$, $B = 1$ y $C = 2$. Si el sistema no elimina los valores no factibles, entonces el usuario podrá dar evidencias contradictorias tales como $Pago = autorizado$ y $NIP = incorrecto$ en el Ejemplo 2.1 del CA. Por ello, tan pronto como se de la primera evidencia, $Pago = autorizado$, el sistema debe seleccionar sólo los valores del NIP que no conduzcan a conclusiones contradictorias.
3. Suministrar al usuario una lista de objetos a los que no se ha asignado valores previamente.
4. Para cada uno de los objetos, mostrar y aceptar sólo sus valores factibles.
5. Actualizar continuamente la base de conocimiento, es decir, tan pronto como se dé un hecho o se obtenga una conclusión, y eliminar los valores no factibles. El motor de inferencia obtiene todas las conclusiones posibles examinando, y posiblemente concluyendo, las reglas tan pronto como una simple unidad de información llega al sistema.

Nótese que dar varias unidades de información simultáneamente puede conducir a inconsistencias en la base de datos. Por ejemplo, dado $A = 0$, no se puede dar la información combinada $B = 0$ y $C = 1$. En este caso, el orden de la información afecta a los posibles valores futuros de los objetos que conducen a compatibilidad, es decir, tras dar $A = 0$ se puede dar $B = 0$ ó $C = 1$, pero estas dos opciones imponen restricciones diferentes a los posibles futuros valores de los restantes objetos.

La actualización continua de la base de conocimiento es muy importante puesto que no actualizar implica la posibilidad de que evidencias contradictorias puedan convivir en la base de conocimiento. Por ello, el conocimiento debe ser actualizado inmediatamente tras la incorporación de cada hecho.

Por ello, tanto la eliminación automática de valores no factibles como la actualización continua del conocimiento aseguran la coherencia de la base de conocimiento. El ejemplo siguiente ilustra la aplicación de esta técnica al problema de los agentes secretos presentado en el Capítulo 1.

Ejemplo 2.16 Los Agentes Secretos. En este ejemplo se retoma el problema de los agentes secretos introducido en el Ejemplo 1.5, en el que cada uno de los cuatro agentes secretos, Alberto, Luisa, Carmen y Tomás, está en uno de los cuatro países: Egipto, Francia, Japón y España. Se han recibido los siguientes telegramas de los agentes:

- De Francia: Luisa está en España.
- De España: Alberto está en Francia.
- De Egipto: Carmen está en Egipto.
- De Japón: Carmen está en Francia.

El problema radica en que no se sabe quién ha enviado cada uno de los mensajes, pero es conocido que Tomás miente (¿es un agente doble?) y que los demás agentes dicen la verdad. El misterio que trata de escudriñarse es el de responder a la pregunta ¿quién está en cada país?

Seguidamente se diseña un sistema experto para resolver este problema. Se tienen cuatro objetos: Alberto, Luisa, Carmen y Tomás. Cada objeto puede tomar uno de cuatro valores: Egipto, Francia, Japón o España. Puesto que Tomás es el único que miente, se considera que un telegrama suyo es siempre falso. Esto da lugar a dos reglas por cada mensaje:

1. El mensaje de Francia (Luisa está en España) da lugar a:
 - Regla 1: Si Tomás está en Francia, entonces Luisa no está en España.

- Regla 2: Si Tomás no está en Francia, entonces Luisa está en España.
2. El mensaje de España (Alberto está en Francia) da lugar a:
 - Regla 3: Si Tomás está en España, entonces Alberto no está en Francia.
 - Regla 4: Si Tomás no está en España, entonces Alberto está en Francia.
 3. El mensaje de Egipto (Carmen está en Egipto) da lugar a:
 - Regla 5: Si Tomás está en Egipto, entonces Carmen no está en Egipto.
 - Regla 6: Si Tomás no está en Egipto, entonces Carmen está en Egipto.
 4. El mensaje de Japón (Carmen está en Francia) da lugar a:
 - Regla 7: Si Tomás está en Japón, entonces Carmen no está en Francia.
 - Regla 8: Si Tomás no está en Japón, entonces Carmen está en Francia.

Utilizando sólo estas ocho reglas, se intentará ahora averiguar el valor que toma el objeto Tomás:

1. Tomás está posiblemente en Egipto. Si Tomás está en Egipto, se obtienen las conclusiones siguientes:
 - Luisa está en España, por la Regla 2.
 - Alberto está en Francia, por la Regla 4.
 - Carmen no está en Egipto, por la Regla 5.
 - Carmen está en Francia, por la Regla 8.

Se ve que con esta hipótesis se llega a la conclusión de que tanto Alberto como Carmen están en Francia, lo que contradice la información de que sólo un agente puede estar en cada país (pero el conjunto de las ocho reglas anteriores no contiene esta información). Por tanto, se concluye que Egipto es un valor imposible para el objeto Tomás, es decir, Tomás no puede estar en Egipto.

2. Tomás está posiblemente en Japón. Si Tomás está Japón, se obtienen las conclusiones siguientes:
 - Luisa está en España, por la Regla 2.
 - Alberto está en Francia, por la Regla 4.

- Carmen está en Egipto, por la Regla 6.

En este caso no hay una contradicción, lo que significa que Japón es un valor posible para el objeto Tomás.

Con las ocho reglas anteriores, el motor de inferencia no puede concluir en qué país está cada uno de los agentes, puesto que las reglas no contienen la información “sólo un agente puede estar en cada país.” Seguidamente se considera esta situación y se obtiene un conjunto de reglas adicionales que tienen en cuenta esta información.

Puesto que cada país puede estar ocupado por exactamente un agente, supóngase que un agente está en un país dado. Entonces, se necesitan tres reglas para garantizar que ninguno de los restantes agentes está en ese mismo país. Dado que se tienen cuatro agentes, resultan un total de 12 reglas (3 reglas \times 4 agentes). Sin embargo, si se utiliza la regla de inferencia Modus Tollens, sólo son necesarias seis reglas, pues las restantes resultan redundantes. Por ejemplo, para Egipto se tienen las reglas:

- Regla 9: Si Alberto está en Egipto, entonces Luisa no está en Egipto.
- Regla 10: Si Alberto está en Egipto, entonces Carmen no está en Egipto.
- Regla 11: Si Alberto está en Egipto, entonces Tomás no está en Egipto.
- Regla 12: Si Luisa está en Egipto, entonces Carmen no está en Egipto.
- Regla 13: Si Luisa está en Egipto, entonces Tomás no está en Egipto.
- Regla 14: Si Carmen está en Egipto, entonces Tomás no está en Egipto.

Nótese que existen un conjunto de seis reglas equivalentes a las anteriores. Por ejemplo, la regla:

- Regla 14a: Si Tomás está en Egipto, entonces Carmen no está en Egipto,

es equivalente a (Modus Tollens) la Regla 14. Por tanto, se necesitan sólo seis reglas por país.

Los conjuntos de seis reglas para cada uno de los restantes países se generan de forma similar. Por tanto, se tienen un total de 24 reglas adicionales que representan el hecho de que exactamente un agente puede estar en cada país. ■

2.5 Explicando Conclusiones

Tal como se ha indicado en el Capítulo 1, las conclusiones no bastan para satisfacer al usuario de un sistema experto. Normalmente, los usuarios esperan que el sistema les dé algún tipo de explicación que indique el por qué de las conclusiones. Durante el proceso realizado por el motor de inferencia, las reglas activas (las que han concluído) forman la base del mecanismo de explicación, que es regulado por el subsistema de explicación.

En los sistemas expertos basados en reglas, es fácil dar explicaciones de las conclusiones obtenidas. El motor de inferencia obtiene conclusiones basándose en un conjunto de reglas y, por tanto, conoce de qué regla procede cada conclusión. Por ello, el sistema puede dar al usuario la lista de hechos concluidos junto con las reglas que se han utilizado para obtenerlos.

Ejemplo 2.17 Explicando conclusiones. Considérense las seis reglas de las Figuras 2.10 y 2.11. Como en el Ejemplo 2.7, supóngase que se sabe que los objetos A, B, D, E, F, H , y I son *ciertos* y que los restantes objetos toman valores desconocidos. Entonces, aplicando el Algoritmo 2.1 y examinando las reglas que han sido ejecutadas, el sistema experto puede suministrar la explicación siguiente a las conclusiones obtenidas:

1. Hechos dados:

$$\begin{aligned} A = \textit{cierto}, \quad B = \textit{cierto}, \quad D = \textit{cierto}, \quad E = \textit{cierto}, \\ F = \textit{cierto}, \quad H = \textit{cierto}, \quad I = \textit{cierto}. \end{aligned}$$

2. Conclusiones y explicaciones:

- $C = \textit{cierto}$, basada en la Regla 1.
- $G = \textit{cierto}$, basada en la Regla 2.
- $J = \textit{cierto}$, basada en la Regla 3.
- $K = \textit{cierto}$, basada en la Regla 4.
- $L = \textit{cierto}$, basada en la Regla 5.
- $M = \textit{cierto}$, basada en la Regla 6. ■

2.6 Ejemplo de Aplicación

Los sistemas de control de tráfico actuales son necesariamente complejos. En esta sección se muestra un ejemplo para ilustrar cómo puede diseñarse un sistema experto basado en reglas para resolver un problema de control de tráfico muy simple. La Figura 2.16 muestra un trazado de ferrocarril en el que varios trenes pueden circular en las dos direcciones. Hay cinco vías, S_1, \dots, S_5 , y 14 señales de tráfico, ocho en la parte superior del diagrama,

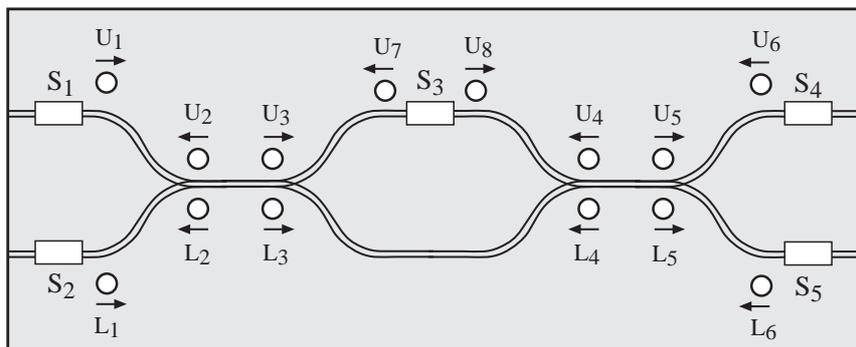


FIGURA 2.16. Trazado de ferrocarril con cinco vías.

Objeto	Valor
U_1 a U_8	$\{verde, rojo\}$
L_1 a L_6	$\{verde, rojo\}$
S_1 a S_5	$\{libre, ocupada\}$

TABLA 2.11. Objetos y sus correspondientes valores para el ejemplo del control de tráfico ferroviario.

U_1, \dots, U_8 , y seis en la parte inferior, L_1, \dots, L_6 . Todos los objetos y sus posibles valores se muestran en la Tabla 2.11.

El objetivo de este sistema es diseñar un conjunto de reglas que eviten la colisión de los trenes. Estas reglas pueden obtenerse como sigue:

1. Si la señal de tráfico U_1 está *verde*, entonces puede permitirse la salida de un tren que esté en la vía S_1 y no debe permitirse la salida de los trenes de la vía S_2 , por lo que L_1 tiene que estar en *rojo*. Lo mismo es cierto para las vías S_4 y S_5 . Esto da las dos primeras reglas de la Tabla 2.12. Nótese que si el motor de inferencia utiliza la regla de inferencia Modus Tollens, estas reglas garantizan también que cuando las señales de la parte baja de las vías estén en verde, las señales de sus correspondientes partes superiores estén en rojo. Es decir, las dos primeras reglas de la Tabla 2.12 implican las dos reglas siguientes:
 - Regla 1a: Si $L_1 = verde$, entonces $U_1 = rojo$.
 - Regla 2a: Si $L_6 = verde$, entonces $U_6 = rojo$.
2. Si la vía S_1 está ocupada, entonces la señal U_2 debe estar en *rojo* para evitar que un tren entre en la vía ocupada. Similarmente, para las demás vías. Esto da lugar a las seis reglas adicionales (Reglas 3–8) en la Tabla 2.12.

3. Si ambas señales U_3 y L_3 están en rojo, entonces ningún tren puede salir de la vía S_1 . La misma condición vale para las señales U_5 y L_5 . Por ello, se tiene la regla
- Regla 9: Si $(U_3 = rojo \text{ y } L_3 = rojo)$ o $(U_5 = rojo \text{ o } L_5 = rojo)$, entonces $U_1 = rojo$.

Las cinco reglas asociadas a las otras cinco vías pueden ser obtenidas de forma análoga. Todas las reglas se muestran en la Tabla 2.12 como las Reglas 9–14.

4. Para evitar la colisión de los trenes procedentes de las vías $S_1 - S_2$ y $S_4 - S_5$, son necesarias las reglas siguientes:
- Regla 15: Si $U_3 = verde$, entonces $U_4 = rojo$
 - Regla 16: Si $L_3 = verde$, entonces $L_4 = rojo$.
5. Para evitar que las señales de la parte alta y sus correspondientes señales de la parte baja estén simultáneamente en verde, es necesario incluir las Reglas 17–20 de la Tabla 2.12.
6. Finalmente, para evitar la colisión de un tren de la vía S_3 con un tren de las otras cuatro vías, se imponen las dos últimas reglas de la Tabla 2.12.

Para mantener la coherencia de los hechos, es necesario actualizar automáticamente el conocimiento tan pronto como se conozca un nuevo hecho o conclusión.

Seguidamente se considera un ejemplo para ilustrar el comportamiento de un sistema experto cuya base de conocimiento consta de los objetos de la Tabla 2.11 y el conjunto de reglas de la Tabla 2.12.

Ejemplo 2.18 Control de tráfico ferroviario. En este ejemplo se usará la concha *X-pert Reglas*. En primer lugar, se necesita escribir un fichero que contenga una descripción de la base de conocimiento anterior. Este fichero es leído por la concha *X-pert Reglas*. Puesto que *X-pert Reglas* no permite el uso del operador *o* en la premisa de las reglas, es necesario reemplazar las Reglas 9–12 por el conjunto siguiente de reglas equivalente (ver las equivalencias de la Tabla 2.4):

- Regla 9a: Si $(U_3 = rojo \text{ y } L_3 = rojo)$, entonces $U_1 = rojo$.
- Regla 9b: Si $(U_5 = rojo \text{ y } L_5 = rojo)$, entonces $U_1 = rojo$.
- Regla 10a: Si $(U_3 = rojo \text{ y } L_3 = rojo)$, entonces $L_1 = rojo$.
- Regla 10b: Si $(U_5 = rojo \text{ y } L_5 = rojo)$, entonces $L_1 = rojo$.
- Regla 11a: Si $(U_2 = rojo \text{ y } L_2 = rojo)$, entonces $U_6 = rojo$.

Regla	Premisa	Conclusión
Regla 1	$U_1 = verde$	$L_1 = rojo$
Regla 2	$U_6 = verde$	$L_6 = rojo$
Regla 3	$S_1 = ocupada$	$U_2 = rojo$
Regla 4	$S_2 = ocupada$	$L_2 = rojo$
Regla 5	$S_3 = ocupada$	$U_3 = rojo$
Regla 6	$S_3 = ocupada$	$U_4 = rojo$
Regla 7	$S_4 = ocupada$	$U_5 = rojo$
Regla 8	$S_5 = ocupada$	$L_5 = rojo$
Regla 9	$(U_3 = rojo \text{ y } L_3 = rojo) \text{ o } (U_5 = rojo \text{ y } L_5 = rojo)$	$U_1 = rojo$
Regla 10	$(U_3 = rojo \text{ y } L_3 = rojo) \text{ o } (U_5 = rojo \text{ y } L_5 = rojo)$	$L_1 = rojo$
Regla 11	$(U_2 = rojo \text{ y } L_2 = rojo) \text{ o } (U_4 = rojo \text{ y } L_4 = rojo)$	$U_6 = rojo$
Regla 12	$(U_2 = rojo \text{ y } L_2 = rojo) \text{ o } (U_4 = rojo \text{ y } L_4 = rojo)$	$L_6 = rojo$
Regla 13	$U_2 = rojo \text{ y } L_2 = rojo$	$U_7 = rojo$
Regla 14	$U_5 = rojo \text{ y } L_5 = rojo$	$U_8 = rojo$
Regla 15	$U_3 = verde$	$U_4 = rojo$
Regla 16	$L_3 = verde$	$L_4 = rojo$
Regla 17	$U_2 = verde$	$L_2 = rojo$
Regla 18	$U_3 = verde$	$L_3 = rojo$
Regla 19	$U_4 = verde$	$L_4 = rojo$
Regla 20	$U_5 = verde$	$L_5 = rojo$
Regla 21	$U_1 = verde \text{ o } L_1 = verde$	$U_7 = rojo$
Regla 22	$U_6 = verde \text{ o } L_6 = verde$	$U_8 = rojo$

TABLA 2.12. Reglas para el ejemplo del control de tráfico ferroviario.

- Regla 11b: Si $(U_4 = rojo \text{ y } L_4 = rojo)$, entonces $U_6 = rojo$.
- Regla 12a: Si $(U_2 = rojo \text{ y } L_2 = rojo)$, entonces $L_6 = rojo$.
- Regla 12b: Si $(U_4 = rojo \text{ y } L_4 = rojo)$, entonces $L_6 = rojo$.

Por ello, se crea el fichero de texto “TrafficControl.txt,” que contiene todos los objetos y las 26 reglas.¹ Supóngase que inicialmente se tienen trenes

¹El fichero “TrafficControl.txt” con la base de conocimiento y la concha para construir sistemas expertos *X-pert Reglas* puede obtenerse de la dirección de World Wide Web <http://ccaix3.unican.es/~AIGroup>.

esperando en las vías S_1 , S_2 y S_3 tal como se indica en la Figura 2.17. Lo que sigue muestra una sesión interactiva utilizando *X-pert Reglas* y tras leer el fichero de texto “TrafficControl.txt”. En primer lugar se especifican los hechos: $S_1 = ocupada$, $S_3 = ocupada$ y $S_5 = ocupada$. Entonces se obtienen los siguientes hechos (datos) y las conclusiones (hechos concluidos):

1. Hechos:

- $S_1 = ocupada$.
- $S_3 = ocupada$.
- $S_5 = ocupada$.

2. Conclusiones:

- $U_2 = rojo$ (basada en la Regla 3).
- $U_3 = rojo$ (basada en la Regla 5).
- $U_4 = rojo$ (basada en la Regla 6).
- $L_5 = rojo$ (basada en la Regla 8).

Por ello, se ponen en rojo cuatro señales para evitar la colisión de los trenes que esperan en las vías. El resto de los objetos toman valores desconocidos. La Figura 2.17 muestra esta información en forma gráfica.

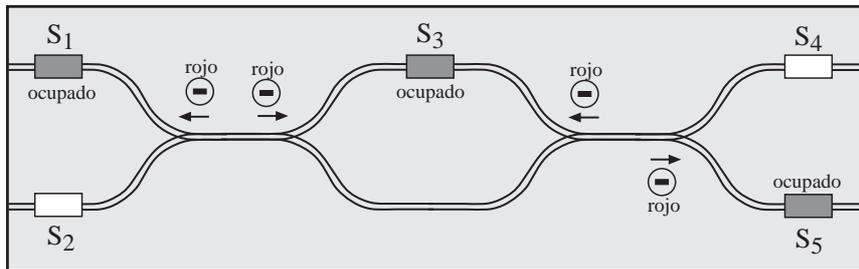


FIGURA 2.17. Nuevas conclusiones resultantes de los hechos $S_1 = S_3 = S_5 = ocupada$.

Supóngase ahora que se desea permitir al tren en la vía S_1 salir en dirección Este. Para ello, se hace $U_1 = verde$. Entonces se tienen los siguientes hechos y conclusiones:

1. Hechos:

- $S_1 = ocupada$.
- $S_3 = ocupada$.
- $S_5 = ocupada$.
- $U_1 = verde$.

2. Conclusiones:

- $U_2 = rojo$ (basada en la Regla 3).
- $U_3 = rojo$ (basada en la Regla 5).
- $U_4 = rojo$ (basada en la Regla 6).
- $L_5 = rojo$ (basada en la Regla 8).
- $L_1 = rojo$ (basada en la Regla 1).
- $U_7 = rojo$ (basada en la Regla 21).
- $L_3 = verde$ (basada en la Regla 9a).
- $U_5 = verde$ (basada en la Regla 9b).
- $L_4 = rojo$ (basada en la Regla 16).
- $S_4 \neq ocupada$ (basada en la Regla 7).
- $S_4 = free$ (es el único valor posible).
- $U_6 = rojo$ (basada en la Regla 11b).
- $L_6 = rojo$ (basada en la Regla 12b).

La Figura 2.18 muestra las conclusiones resultantes. Nótese que el tren que está en la vía S_1 puede ahora partir y dirigirse a la vía S_4 . Este camino se muestra en la Figura 2.18. ■

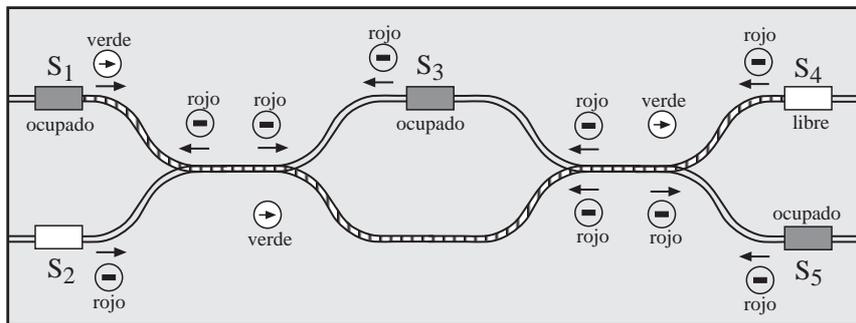


FIGURA 2.18. Nuevas conclusiones resultantes de los hechos $S_1 = S_3 = S_5 = ocupada$ and $U_1 = verde$.

2.7 Introduciendo Incertidumbre

Los sistemas basados en reglas descritos en este capítulo pueden aplicarse sólo a situaciones deterministas. Sin embargo, hay muchos casos prácticos que implican incertidumbre. Por ejemplo, en el Ejemplo 1.4 del diagnóstico

médico, la presencia de algunos síntomas no siempre implica la existencia de una enfermedad dada, incluso aunque haya una fuerte evidencia sobre la existencia de esa enfermedad. Por ello, es útil extender la lógica clásica para incorporar incertidumbre. Esto ha sido realizado mediante la introducción de varias medidas para tratar la incertidumbre. Castillo y Álvarez (1990, 1991) describen la aplicación de estas medidas para mejorar los sistemas expertos basados en reglas. Por otra parte, Johnson y Keravnou (1988) describen algunos prototipos de sistemas expertos basados en lógicas inciertas. El Capítulo 3 describe en detalle los *sistemas expertos basados en probabilidad*, que incorporan la incertidumbre.

Ejercicios

- 2.1 En el Ejemplo 2.3, se usan dos objetos binarios A y B y se da un ejemplo en el que la regla de inferencia Modus Tollens expande la base de conocimiento. Utilizando objetos no binarios, dar un ejemplo similar. Por ejemplo, cuando A y B puedan tomar los valores $\{0, 1, 2\}$.
- 2.2 Mostrar que los dos conjuntos de reglas de las Figuras 2.1 y 2.7 son lógicamente equivalentes.
- 2.3 En algún momento del Ejemplo 2.11, se ha buscado una regla activa que incluyera el objeto en curso J . Se encontraron las Reglas 3 y 5, y se eligió la Regla 5. Completar las etapas del algoritmo si se hubiera elegido la Regla 3 en vez de la Regla 5.
- 2.4 Considérese una intersección de dos calles de doble sentido, tal como se indica en la Figura 2.19, en la que se muestran los giros permitidos. Sean $T_1 - T_3$, $R_1 - R_3$, $B_1 - B_3$ y $L_1 - L_3$ los semáforos asociados a dichos carriles. Definir un conjunto de reglas que regulen la intersección de forma que no pueda producirse ninguna colisión.
- 2.5 Considérese la línea ferroviaria con seis vías dada en la Figura 2.20. Completar el conjunto de reglas dado en la Sección 2.6 para incorporar la nueva vía S_6 .
- 2.6 Considérese la línea ferroviaria de la Figura 2.21 con cuatro vías $\{S_1, \dots, S_4\}$. Diseñar un sistema de control de tráfico ferroviario para controlar las señales. Obtener un conjunto de reglas que garanticen la imposibilidad de colisiones de trenes.
- 2.7 Supóngase que se tienen las seis reglas del Ejemplo 2.7. Siguiendo el proceso dado en los Ejemplos 2.9 y 2.10, aplicar un algoritmo de encadenamiento orientado a un objetivo para concluir un valor para los objetivos dados en las Figuras 2.22(a) y 2.22(b). Los objetos que

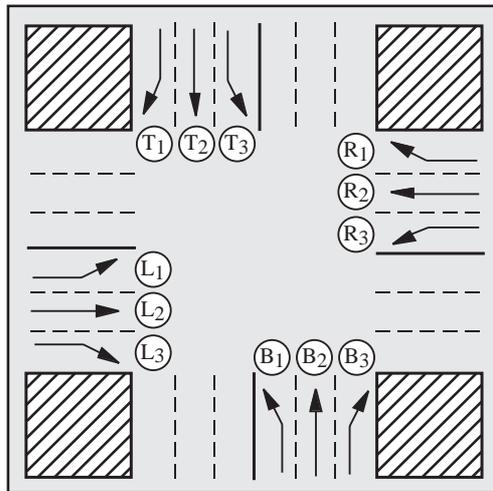


FIGURA 2.19. Intersección mostrando los giros permitidos.

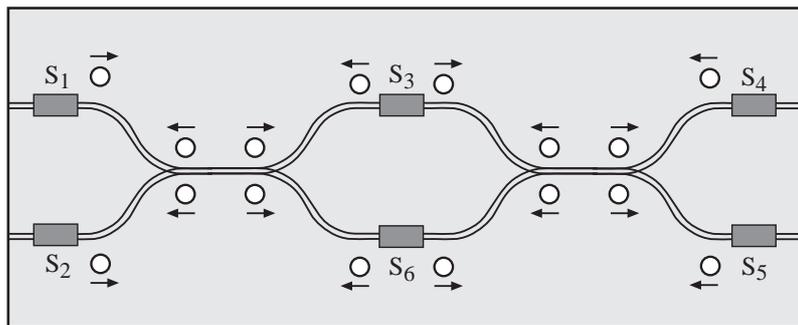


FIGURA 2.20. Una línea ferroviaria con seis vías.

se indican en gris son objetos con valores asignados. Los correspondientes valores se muestran próximos a los objetos. ¿Cuáles serían las conclusiones con un motor de inferencia que sólo incluya la regla de inferencia Modus Ponens?

2.8 Diseñar un sistema experto basado en reglas que sirva para jugar al “Tres en Raya”. Por turno, dos jugadores ponen una de sus piezas en un tablero de 9 cuadrados (3×3) (véase la Figura 2.23). Gana el jugador que consiga poner sus 3 piezas en columna (Figura 2.23(a)), en fila (Figura 2.23(b)), o en diagonal (Figura 2.23(c)). Considérense las estrategias siguientes:

- Estrategia defensiva: Definir las reglas para evitar que el contrario gane.

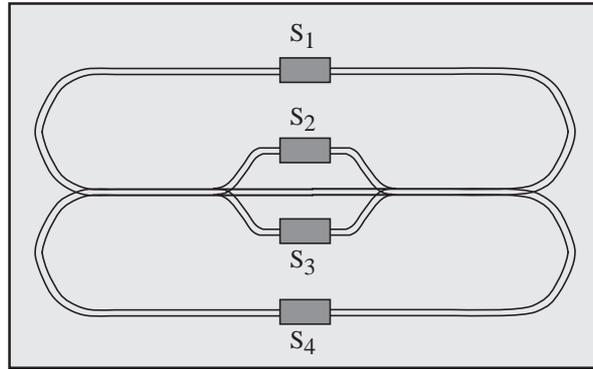


FIGURA 2.21. Una línea ferroviaria con cuatro vías.

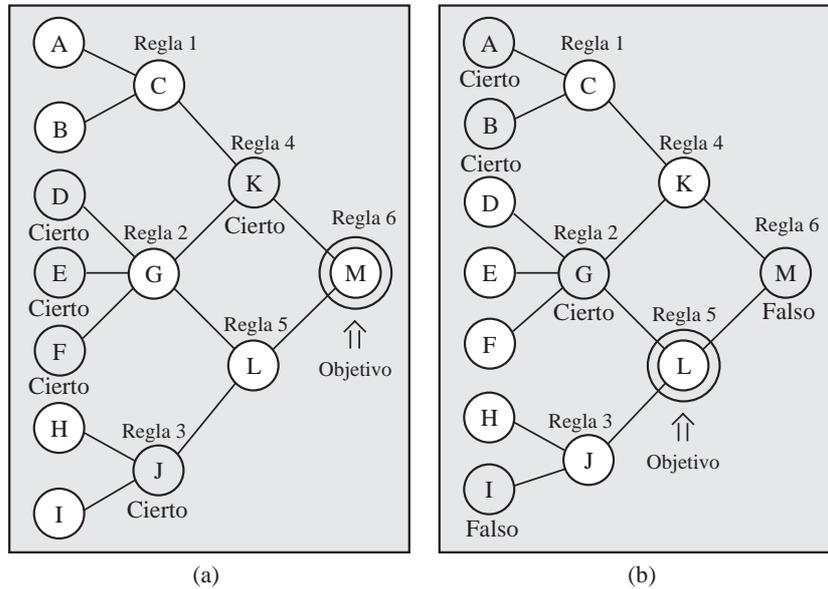


FIGURA 2.22. Hechos iniciales y objetivos para un algoritmo de encadenamiento de reglas orientado a un objetivo.

- Estrategia atacante: Añadir el conjunto de reglas que definan la estrategia para ganar.

2.9 Diseñar un sistema experto basado en reglas para clasificar animales o plantas basándose en un conjunto mínimo de características. Síganse las etapas siguientes:

- Decidir el conjunto de animales o plantas a clasificar.
- Elegir las características diferenciales.

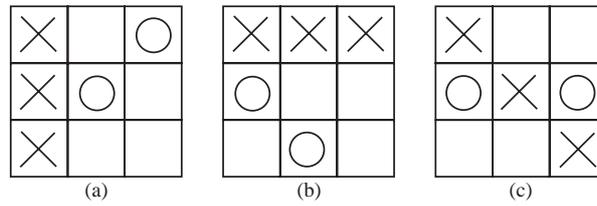


FIGURA 2.23. Tableros del juego del “Tres en Raya”: Tres ejemplos en los que el jugador “X” es el ganador.

- Definir las reglas necesarias para identificar cada animal o planta.
- Eliminar las características innecesarias.
- Escribir las reglas en el sistema.
- Comprobar exhaustivamente el sistema experto.
- Rediseñar el sistema a la vista de lo anterior.

2.10 En el ejemplo de los agentes secretos del Ejemplo 2.16, ¿qué conclusiones pueden sacarse utilizando sólo las ocho primeras reglas cuando (a) se da Francia como valor posible del objeto Tomás y (b) se da España como valor posible para el mismo objeto?.