



Luis Valencia Cabrera

lvalencia@us.es

(<http://www.cs.us.es/~lvalencia>)

Ciencias de la Computacion e IA

(<http://www.cs.us.es/>)

Universidad de Sevilla

Sistemas basados en reglas (2010/2011)

Introducción

- Muchas *situaciones complejas* se rigen 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.
- En esta presentación se muestra cómo los **sistemas probabilísticos** pueden considerarse como una **generalización de los sistemas basados en reglas**.

Contenidos

- Como extensión a la presentación de la sesión anterior, se detallan los contenidos:
 - **base de conocimiento** de los sistemas expertos basados en reglas (definición y ejemplos de reglas)
 - **motor de inferencia** (cómo opera/funciona)
 - subsistema de **control de la coherencia** (cómo trabaja)
 - **Subsistema de explicación** de las conclusiones sacadas por el motor de inferencia
 - ejemplo de **aplicación**
 - **limitaciones** de los sistemas expertos basados en reglas.

Base de conocimiento

- En los sistemas basados en reglas intervienen dos elementos importantes: las **reglas** y los **hechos**.
- **Hechos:**
 - Son conocidos en una situación **particular**.
 - Son **dinámicos**, pueden cambiar de una aplicación a otra.
 - Su naturaleza **no** es **permanente** y
 - Se almacenan en la **memoria de trabajo**.
- **Reglas:**
 - En *situaciones deterministas*, relaciones **generales** entre un conjunto de objetos, y reglas que gobiernan las relaciones
 - Son **estáticos**, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje
 - La información es de naturaleza **permanente**.
 - Almacenada en la **base de conocimiento**.

Ejemplo. Notas

- 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

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

- 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 $\text{Notificar} = \text{sí}$.

Reglas. Componentes

- Cada una de las reglas anteriores relaciona dos o más objetos y está formada por las partes siguientes:
- **Premisa ó antecedente:** 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.
- **Conclusión ó consecuente:** expresión lógica tras la palabra clave *entonces*.

Reglas. Definición

- **Definición de 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**.
 - "Si *premisa*, entonces *conclusión*".
- La premisa y la conclusión pueden contener afirmaciones múltiples objeto-valor.
 - Sólo una afirmación objeto-valor → *expresión lógica simple*.
 - Ejemplo: tanto la premisa como la conclusión de la regla 1.
 - Más de una → *expresión lógica compuesta*.
 - Ejemplo: las expresiones lógicas de las premisas y la conclusión de la Regla 2.
- Una regla que contiene solamente expresiones lógicas simples se denomina una *regla simple* (ejemplo: reglas 2); en otro caso, se llama *regla compuesta* (ejemplo: regla 1).

Ejemplo. Cajero automático

Problema determinista que puede formularse con un conjunto de **reglas**:

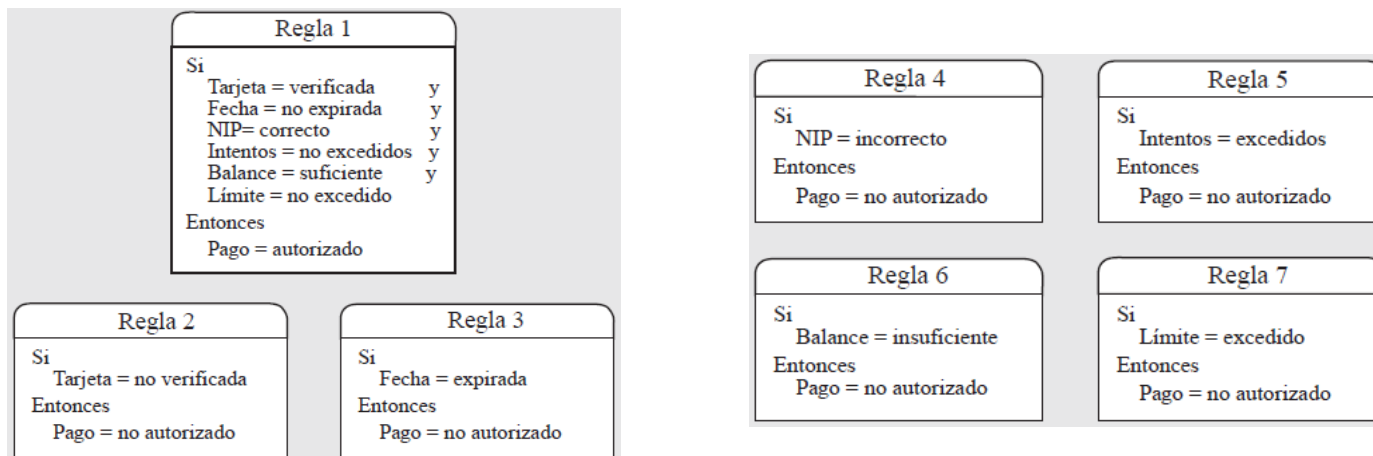
Un cliente desea sacar dinero de un CA:

1. Introduce la tarjeta en el CA, la máquina la lee y la verifica.
2. Si la tarjeta no es verificada con éxito (ej, porque no es legible), el CA devuelve la tarjeta al usuario con el mensaje de error correspondiente.
3. En otro caso, el CA pide al usuario su número de identificación personal (NIP).
4. Si el número fuese incorrecto, se dan tres oportunidades de corregirlo.
5. Si el NIP es correcto, el CA pregunta al usuario cuánto dinero desea sacar.
6. Para que el pago se autorice, la cantidad solicitada no debe exceder de una cierta cantidad límite diaria,
7. además de haber suficiente dinero en su cuenta.

Se tienen siete objetos, y cada objeto puede tomar uno y sólo un valor de entre sus posibles valores.

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}

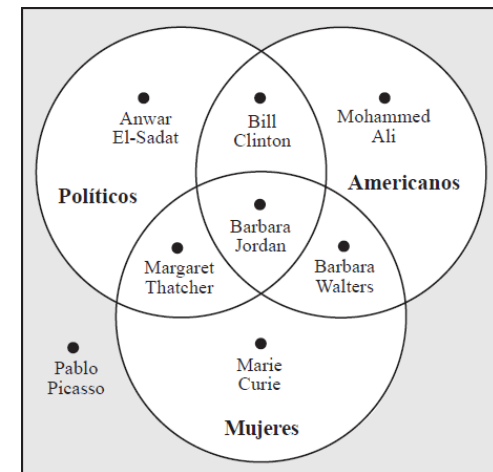
Ejemplo. Cajero automático



- **Estrategia del CA** cuando un usuario intenta sacar dinero.
- En la Regla 1, la premisa consiste en seis afirmaciones objeto-valor conectadas mediante el operador lógico y → la premisa es cierta si las seis afirmaciones lo son. Relaciona el objeto Pago (en la conclusión) con los demás objetos. 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 de las restantes reglas constan de una sola afirmación.
- La Regla 1 indica cuándo debe **permitirse** el pago, y las restantes cuándo debe **rechazarse**.

Ejemplo. Gente famosa

- Se dispone de una base de datos consistente en **N** individuos.
- Para cada individuo, la base de datos contiene **4 atributos**: nombre, sexo, nacionalidad y profesión.
- Muestra si una persona es americana, política y/o si es mujer, siendo atributos **binarios** (sólo 2 valores posibles).
 - Como mucho, $2^3 = 8$ conjuntos disjuntos.
- Se tiene el nombre de una persona en cada subconjunto.
- Ejemplo: $N = 8$ 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 los demás 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:
 - Regla 1: Si *Nombre* = *Bill Clinton*, entonces *Americano* = sí y *Político* = sí y *Mujer* = no.
 - ...



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

Restricciones

- Algunos sistemas imponen ciertas restricciones a las reglas. Por ejemplo:
 - No permitir en la premisa el operador lógico **o**.
 - Limitar las conclusiones a expresiones lógicas simples
 - Utilizar **formas normales** (Ej. Forma Normal Conjuntiva).
- 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.

Sustitución de reglas

- A la derecha vemos ejemplos de sustitución de reglas.
- Cada regla de la primera columna puede ser sustituida por el conjunto de reglas de la segunda columna
- Todas las reglas de la 2ª columna satisfacen las condiciones anteriores.
- Ejemplo, la primera regla compuesta:
 - Regla 1: Si $A \text{ o } B$, entonces C ,
 - puede ser reemplazada por las dos reglas simples:
 - Regla 1a: Si A , entonces C .
 - Regla 1b: Si B , entonces C .

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

Sustitución de reglas

- Como ejemplo adicional de sustitución según las reglas anteriores, la tabla de abajo muestra que:
 - Regla 2: Si $\overline{A \circ B}$, entonces C,
- puede ser reemplazada por la regla
 - Regla 2: Si \bar{A} y \bar{B} , entonces C,
- donde \bar{A} significa no A.

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

- Esta **tabla de verdad** muestra que las expresiones lógicas $\overline{A \circ B}$ y \bar{A} y \bar{B} son equivalentes. Los símbolos C y F se utilizan para cierto y falso, respectivamente.

Motor de Inferencia

- Hemos visto que hay dos tipos de elementos:
 - los **hechos**, información declarativa sobre el mundo.
 - las **reglas**, conocimiento declarativo aplicable a la gestión de la base de hechos.
- El motor de inferencia usa ambos para obtener conclusiones o hechos:
 - En lógica monótona: los únicos hechos de la base de conocimiento son los iniciales y los deducidos.
 - En lógica no-monótona: puede insertarse, modificarse y eliminarse hechos.
- Ejemplo: si la **premisa** de una regla es **cierta** → la **conclusión también**.
 - Se incorpora a los datos, de modo que tanto los hechos iniciales como las conclusiones derivadas forman parte de los hechos o datos en un instante.
- Las conclusiones pueden clasificarse en dos tipos:
 - *Simples* (son las que resultan de una regla simple).
 - *Compuestas* (son las que resultan de más de una regla).

Motor de Inferencia (cont.)

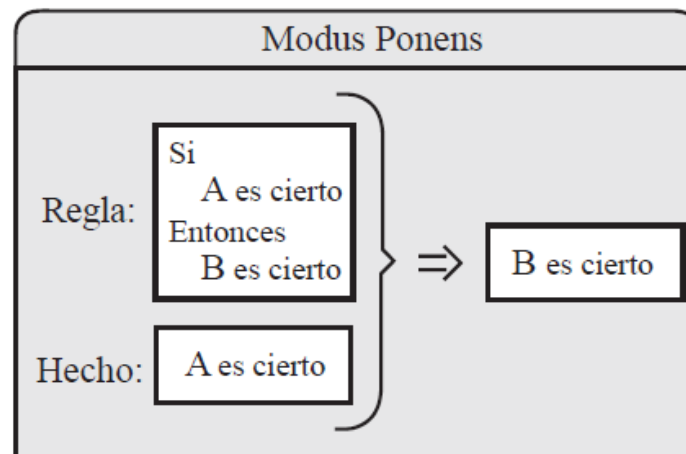
- La interpretación de las reglas conlleva los siguientes pasos básicos:
 - **Reconocimiento de patrones:** comparación de los patrones en las reglas con los elementos de la memoria de trabajo. Aquellas reglas que pueden aplicarse se almacenan en la **Agenda**.
 - **Resolución de conflictos:** elección de una regla entre las satisfechas por la memoria de trabajo y ejecución de su parte ENTONCES (conclusión).
 - **Ejecución:** La ejecución de las reglas da lugar a cambios en la memoria de trabajo. (También podrían añadirse nuevas reglas)
- Para obtener conclusiones → **reglas y estrategias de inferencia y control**.
- La estrategia de control especifica la forma de resolver conflictos

Reglas y estrategias de inferencia y control

- Reglas de inferencia
 - Modus Ponens
 - Modus Tollens
 - Resolución
- Estrategias de inferencia
 - • Encadenamiento de reglas
 - • Encadenamiento de reglas orientado a un objetivo
 - • Compilación de reglas
- Son **utilizadas** por el motor de inferencia **para obtener conclusiones**:
 - Las dos **primeras** reglas de inferencia se usan para obtener **conclusiones simples**.
 - El **resto** de reglas y estrategias para obtener **conclusiones compuestas**.
 - Ninguna de las estrategias por sí solas conduce a todas las conclusiones posibles.
 - **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**.

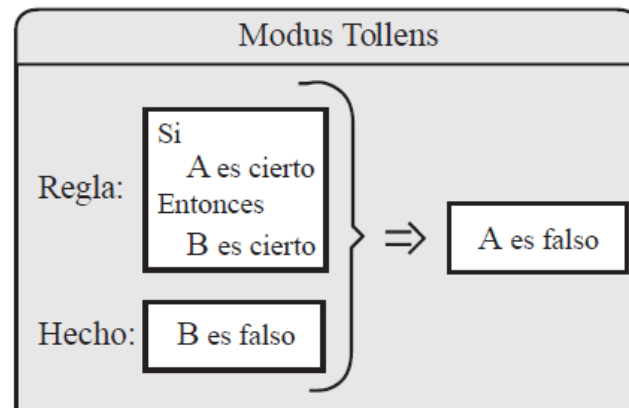
Modus Ponens

- Regla de inferencia más comúnmente utilizada.
- Se utiliza para obtener conclusiones simples.
- Si la premisa de la regla es cierta, la conclusión pasa a formar parte del conocimiento.
- Supongamos que se tiene la regla: "Si **A es cierto**, entonces **B es cierto**" y que se sabe además que "**A es cierto**." Entonces, la regla Modus Ponens concluye que "**B es cierto**."
- Esta regla de inferencia es la base de un gran número de sistemas expertos.



Modus Tollens

- Se utiliza también para obtener conclusiones simples.
- Se examina la conclusión y si es falsa, se concluye que la premisa también es falsa.
- Supongamos que se tiene la regla: "Si **A es cierto**, entonces **B es cierto**" pero se sabe que "**B es falso**."
- Utilizando Modus Ponens no se puede obtener ninguna conclusión, pero Modus Tollens concluye que "**A es falso**."
- Muy simple y con muchas aplicaciones útiles, pero menos utilizada que la Modus Ponens.



Modus Ponens vs Modus Tollens.

Diferentes comportamientos

- La regla Modus Ponens se mueve **hacia adelante**, de la premisa a la conclusión de una regla.
- La regla Modus Ponens **necesita información de** los objetos de **la premisa** para concluir
- La regla Modus Tollens se mueve **hacia atrás**, de la conclusión a la premisa.
- La regla Modus Tollens **necesita información** sobre los objetos **de la conclusión**.
- Las dos **reglas de inferencia** no deben ser vistas como alternativas sino como **complementarias**.
- La combinación de ambas estrategias proporciona más conocimiento que cada una por separado.

Modus Tollens como expansión de la base de conocimiento (BC)

- Supongamos que la BC consiste sólo en la Regla 1.

Regla 1		
Si		
Tarjeta = verificada	y	
Fecha = no expirada	y	
NIP = correcto	y	
Intentos = no excedidos	y	
Balance = suficiente	y	
Límite = no excedido		
Entonces		
Pago = autorizado		

- Se puede utilizar Modus Tollens para “invertir” la Regla 1 y obtener conclusión cuando se tiene información sobre los objetos de su conclusión. Entonces, aplicar Modus Tollens a la regla “Si A, entonces B” es equivalente a aplicar Modus Ponens a la regla “Si \bar{B} , entonces \bar{A} .”

- En este caso de Regla 1, utilizando la equivalencia,

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

- se obtiene la Regla 1b:

Regla 1b		
Si		
Pago = no autorizado		
Entonces		
Tarjeta = no verificada	o	
Fecha = expirada	o	
NIP = incorrecto	o	
Intentos = excedidos	o	
Balance = insuficiente	o	
Límite = excedido		

- Utilizar ambas Modus Ponens y Modus Tollens cuando la BC contiene sólo la Regla 1 equivale a usar la regla Modus Ponens cuando la base de conocimiento contiene ambas, la Regla 1 y la Regla 1b.
- El rendimiento del motor de inferencia depende del conjunto de reglas en su BC. 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).

Inferencia con dos conjuntos equivalentes de reglas

- Supongamos dos motores de inferencia:
 - El motor E1, cuya base de conocimiento contiene las siete reglas que vimos.
 - El motor E2, cuya base de conocimiento contiene las siete reglas de la derecha.
- Los dos conjuntos son lógicamente equivalentes.
- Supongamos que se sabe que el valor de NIP es incorrecto.
- Si E1 y E2 utilizan sólo Modus Ponens, E1 será capaz de concluir que Pago = no autorizado (por la Regla 4), pero E2 no concluirá.
- Por tanto, algunas conclusiones lógicamente derivables pueden no ser obtenidas usando sólo la regla de inferencia Modus Ponens.
- Si ambos motores usan Modus Tollens, entonces ambos concluirán.

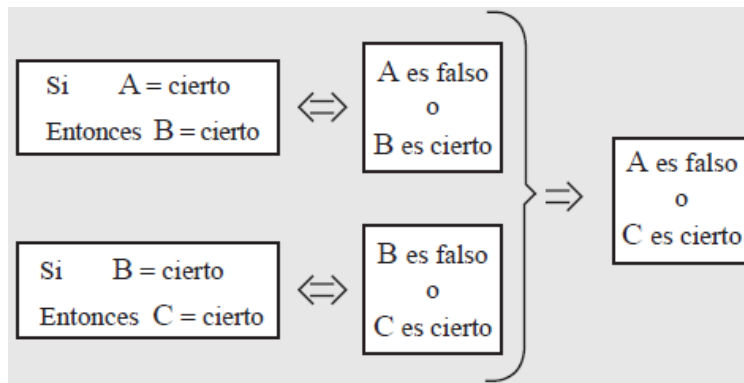
<table><tr><th colspan="2">Regla 1</th></tr><tr><td>Si</td><td></td></tr><tr><td>Tarjeta = verificada</td><td>y</td></tr><tr><td>Fecha = no expirada</td><td>y</td></tr><tr><td>NIP= correcto</td><td>y</td></tr><tr><td>Intentos = no excedidos</td><td>y</td></tr><tr><td>Balance = suficiente</td><td>y</td></tr><tr><td>Límite = no excedido</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr></table>		Regla 1		Si		Tarjeta = verificada	y	Fecha = no expirada	y	NIP= correcto	y	Intentos = no excedidos	y	Balance = suficiente	y	Límite = no excedido		Entonces		Pago = autorizado	
Regla 1																					
Si																					
Tarjeta = verificada	y																				
Fecha = no expirada	y																				
NIP= correcto	y																				
Intentos = no excedidos	y																				
Balance = suficiente	y																				
Límite = no excedido																					
Entonces																					
Pago = autorizado																					
<table><tr><th colspan="2">Regla 2b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Tarjeta = verificada</td><td></td></tr></table>	Regla 2b		Si		Pago = autorizado		Entonces		Tarjeta = verificada		<table><tr><th colspan="2">Regla 3b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Fecha = no expirada</td><td></td></tr></table>	Regla 3b		Si		Pago = autorizado		Entonces		Fecha = no expirada	
Regla 2b																					
Si																					
Pago = autorizado																					
Entonces																					
Tarjeta = verificada																					
Regla 3b																					
Si																					
Pago = autorizado																					
Entonces																					
Fecha = no expirada																					
<table><tr><th colspan="2">Regla 4b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>NIP = correcto</td><td></td></tr></table>	Regla 4b		Si		Pago = autorizado		Entonces		NIP = correcto		<table><tr><th colspan="2">Regla 5b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Intentos = no excedidos</td><td></td></tr></table>	Regla 5b		Si		Pago = autorizado		Entonces		Intentos = no excedidos	
Regla 4b																					
Si																					
Pago = autorizado																					
Entonces																					
NIP = correcto																					
Regla 5b																					
Si																					
Pago = autorizado																					
Entonces																					
Intentos = no excedidos																					
<table><tr><th colspan="2">Regla 6b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Balance = suficiente</td><td></td></tr></table>	Regla 6b		Si		Pago = autorizado		Entonces		Balance = suficiente		<table><tr><th colspan="2">Regla 7b</th></tr><tr><td>Si</td><td></td></tr><tr><td>Pago = autorizado</td><td></td></tr><tr><td>Entonces</td><td></td></tr><tr><td>Límite = no excedido</td><td></td></tr></table>	Regla 7b		Si		Pago = autorizado		Entonces		Límite = no excedido	
Regla 6b																					
Si																					
Pago = autorizado																					
Entonces																					
Balance = suficiente																					
Regla 7b																					
Si																					
Pago = autorizado																					
Entonces																					
Límite = no excedido																					

El Mecanismo de Resolución

- Las reglas de inferencia Modus Ponens y Modus Tollens pueden ser utilizadas para obtener conclusiones simples.
- Las conclusiones compuestas, basadas en dos o más reglas, se obtienen usando el **mecanismo de resolución**.
- Esta regla de inferencia consiste en las **etapas** siguientes:
 1. Las reglas son sustituidas por expresiones lógicas equivalentes.
 2. Éstas expresiones se combinan en otra expresión lógica.
 3. Esta última expresión se utiliza para obtener la conclusión.
- Involucra conceptos como la combinación y simplificación de expresiones lógicas que se ilustran a continuación.

Mecanismo de resolución 1

- Supongamos 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 regla por una expresión lógica equivalente:
 - La Regla 1 es equivalente a la expresión lógica: “ A es falso o B es cierto.” (Verificar tabla verdad)
 - La Regla 2 es equivalente a la expresión lógica: “ B es falso o C es cierto.”



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

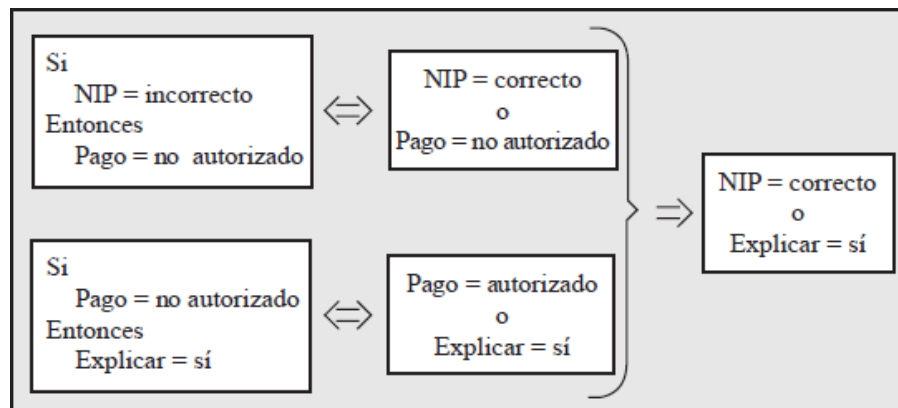
Mecanismo de resolución 1

- La segunda etapa consiste en combinar las dos expresiones anteriores en 1:
 - 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 cierto.” (Ver tabla de verdad)
- Esta última expresión se utiliza seguidamente en la tercera etapa para obtener la conclusión.

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

Mecanismo de resolución 2

- En el ejemplo del CA añadimos el objeto **Explicar**, que puede tomar los valores **{sí, no}**, indicando si se necesita explicar las acciones del CA.
- Aplicamos el mecanismo de **resolución** a la evidencia *NIP = incorrecto* y a las reglas:
 - Si *NIP = incorrecto* entonces *Pago = no autorizado*.
 - Si *Pago = no autorizado* entonces *Explicar = sí*.
- La regla de inferencia correspondiente al mecanismo de resolución conduce a la conclusión *Explicar = sí*.



Mecanismo de resolución 2

- 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 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*.
- El mecanismo de resolución no siempre conduce a conclusiones.
- **Puede no conocerse la verdad o falsedad de ciertas expresiones.** Si ocurre, el motor de inferencia debe decidir entre:
 - **Abandonar la regla**, dada la imposibilidad de obtener conclusiones, o
 - **Preguntar al usuario**, mediante el subsistema de adquisición de información, sobre la verdad o falsedad de una o varias expresiones para continuar el proceso de inferencia hasta obtener una conclusión.
 - **Hipótesis del mundo cerrado**: Si algo no puede demostrarse que sea cierto, entonces es falso.

Encadenamiento de reglas

- Una de las estrategias de inferencia más utilizadas para obtener **conclusiones compuestas**.
- 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**.
- Se repite hasta que no pueden obtenerse más conclusiones.
- El tiempo de este proceso hasta su terminación depende de:
 - Los hechos conocidos,
 - Las reglas que se activan.

Algoritmo

Encadenamiento de reglas

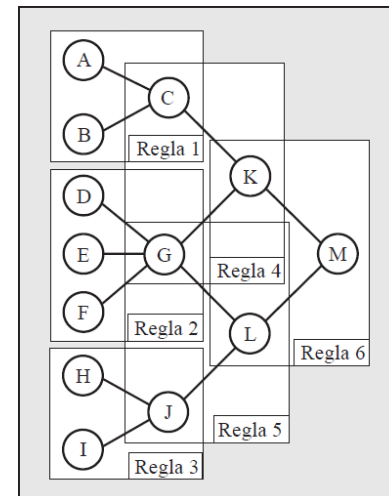
- **Datos:** Una base de conocimiento (objetos y reglas) y algunos hechos iniciales.
- **Resultado:** El conjunto de hechos derivados lógicamente de ellos.
 1. **Asignar a los objetos sus valores conocidos** tales como los dan los hechos conocidos o la evidencia
 2. **Ejecutar cada regla de la base de conocimiento** y concluir nuevos hechos si es posible.
 3. **Repetir la Etapa 2** hasta que no puedan ser obtenidos nuevos hechos.
- Puede implementarse de muchas formas.
 - Comienza con las reglas cuyas premisas tienen valores conocidos.
 - Las conclusiones de éstas dan lugar a nuevos hechos.
 - Éstos se añaden al conjunto de hechos conocidos,
 - El proceso continúa hasta que no pueden obtenerse nuevos hechos.

Ejemplo

Encadenamiento de reglas

- Sea la BC con 6 reglas relacionando 13 objetos como la de la derecha.
- Las relaciones entre estos objetos implicadas por las seis reglas pueden representarse con un árbol:
 - 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.
- Como vemos 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.

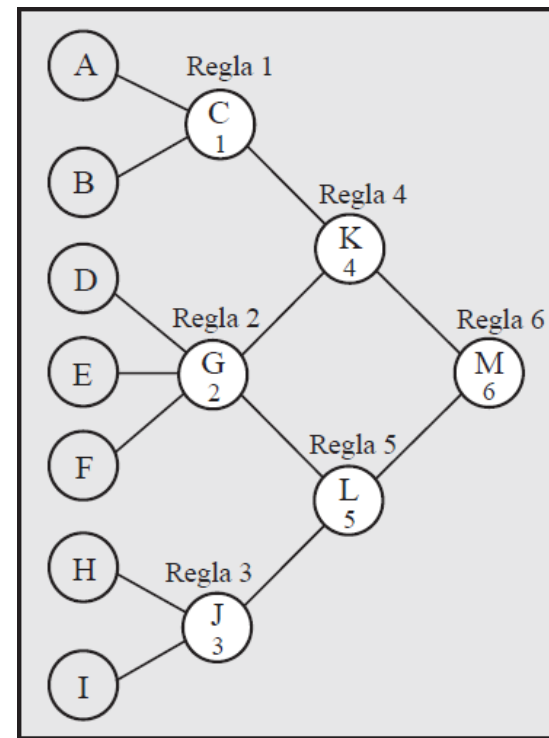
Regla 1 Si A y B Entonces C	Regla 2 Si D, E y F Entonces G	Regla 3 Si H e I Entonces J
Regla 4 Si C y G Entonces K	Regla 5 Si G y J Entonces L	Regla 6 Si K y L Entonces M



Ejemplo

Encadenamiento de reglas

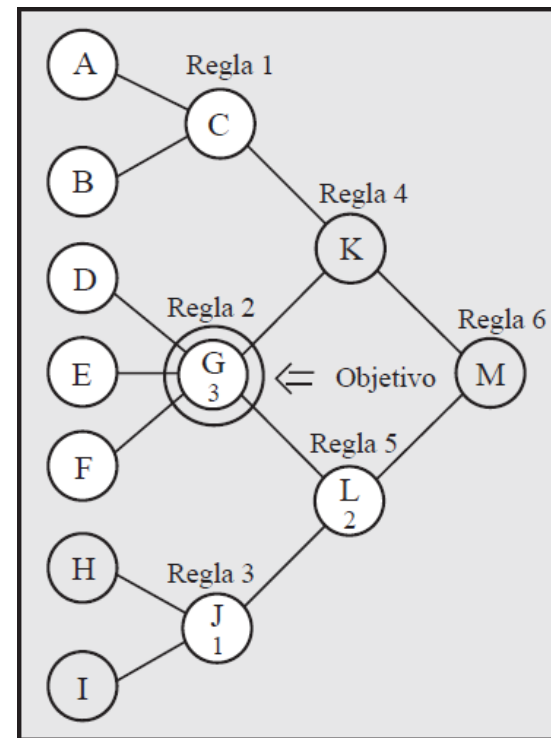
- Supongamos que se sabe que los objetos A, B, D, E, F, H e I son ciertos y los restantes objetos son de valor desconocido.
- La figura distingue entre objetos con valor conocido (los hechos) y objetos con valores desconocidos.
- El algoritmo de encadenamiento de reglas procede como sigue:
 1. La Regla 1 concluye que C = cierto.
 2. La Regla 2 concluye que G = cierto.
 3. La Regla 3 concluye que J = cierto.
 4. La Regla 4 concluye que K = cierto.
 5. La Regla 5 concluye que L = cierto.
 6. La Regla 6 concluye que M = cierto.
- Puesto que no pueden obtenerse más conclusiones, el proceso se detiene.
- Los números dentro de los nodos indican el orden en el que se concluyen los hechos.



Ejemplo

Encadenamiento de reglas 2

- Supongamos los hechos $H = \text{cierto}$, $I = \text{cierto}$, $K = \text{cierto}$ y $M = \text{falso}$.
- Los nodos con valores conocidos aparecen sombreados, la variable objetivo rodeada por una circunferencia, y los números dentro de los nodos indican el orden en el que se concluyen los hechos.
- El motor de inferencia usa Modus Ponens y Modus Tollens y por el algoritmo de encadenamiento:
 - La Regla 3 $\rightarrow J = \text{cierto}$ (Modus Ponens).
 - La Regla 6 $\rightarrow L = \text{falso}$ (Modus Tollens).
 - La Regla 5 $\rightarrow G = \text{falso}$ (Modus Tollens).



Encadenamiento.

Conclusión

- En el ejemplo anterior, por aplicación del algoritmo de encadenamiento con Modus Ponens y Modus Tollens, se obtiene la conclusión $G = \text{falso}$.
- Sin embargo, si el motor de inferencia sólo utiliza la regla de inferencia Modus Ponens, el algoritmo se detendrá en la Etapa 1, no concluyendo nada para el objeto G.
- ***Este 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 hechos que surgen durante el proceso de consulta. Las premisas de las reglas se comparan con los contenidos de la memoria de trabajo y las nuevas conclusiones son pasadas también a la memoria de trabajo.

Encadenamiento de reglas orientado a un objetivo

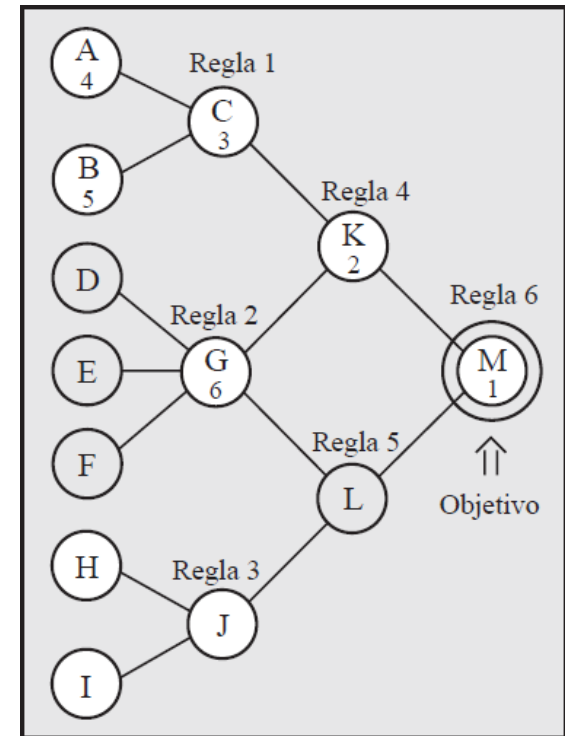
- El usuario **selecciona una variable o nodo objetivo**.
- Entonces el algoritmo navega a través de las **reglas** en **búsqueda de una conclusión para el nodo objetivo**.
- Si no se obtiene ninguna conclusión con la información existente, el algoritmo fuerza a preguntar al usuario en busca de nueva información sobre los elementos relevantes para obtener información sobre el objetivo.
- Algunos autores llaman a los algoritmos:
 - de encadenamiento
 - → **encadenamiento hacia adelante**
 - de encadenamiento orientado a un objetivo
 - → **encadenamiento hacia atrás**.
- Aunque ambos algoritmos pueden utilizar las dos reglas de inferencia Modus Ponens (hacia adelante) y Modus Tollens (hacia atrás).

Algoritmo 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** como vienen en los hechos de partida y marcarlos. Si el nodo objetivo está marcado, ir a la Etapa 7; en otro caso (eoc):
 - a) Designar como objetivo inicial el objetivo en curso.
 - b) Marcar el objetivo en curso.
 - c) Sea *ObjetivosPrevios* = Φ , con Φ el conjunto vacío.
 - d) Designar todas las reglas como activas (ejecutables).
 - e) Ir a la Etapa 2.
 2. **Encontrar regla activa que incluya objetivo en curso** y ninguno de *ObjetivosPrevios*. Si se encuentra, Etapa 3; eoc, Etapa 5.
 3. Ejecutar la regla del objetivo en curso. Si concluye, asignar conclusión a objetivo en curso, e ir a Etapa 6; eoc Etapa 4.
 4. Si todos los objetos de la regla están marcados, declarar la regla como *inactiva* e ir a la Etapa 2; eoc:
 - a) Añadir objetivo en curso a *ObjetivosPrevios*.
 - b) Designar uno de los objetos no marcados en la regla como objetivo en curso.
 - c) Marcar el objetivo en curso.
 - d) Ir a la Etapa 2.
 5. Si objetivo en curso el inicial, Etapa 7; eoc, preguntar al usuario por el valor del objetivo en curso. Si no se da un valor, Etapa 6; eoc asignar el valor dado e ir a la Etapa 6.
 6. Si objetivo en curso es el inicial, ir a la Etapa 7; eoc, 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.

Ejemplo Encadenamiento de reglas orientado a un objetivo

- Supongamos el nodo M objetivo y D,E,F y L ciertos. Las etapas del algoritmo:
 - Etapa 1: Se asigna el valor cierto a los objetos D,E,F y L y se marcan. Puesto que nodo objetivo M no marcado:
 - Se designa M como objeto en curso.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M\}$.
 - $\text{ObjetivosPrevios} = \Phi$.
 - $\text{ReglasActivas} = \{1, 2, 3, 4, 5, 6\}$.
 - Se va a la Etapa 2.
 - Etapa 2. Se busca una regla con el objetivo M. Se encuentra la Regla 6 → Etapa 3.
 - Etapa 3. La Regla 6 no puede concluir ya que el valor de K desconocido. → Etapa 4.
 - Etapa 4. El objeto K no está marcado.
 - $\text{ObjetivosPrevios} = \{M\}$.
 - Se elige K como objetivo en curso.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M, K\}$.
 - Se va a la Etapa 2.



Ejemplo Encadenamiento de reglas orientado a un objetivo

- Etapa 2. Se busca regla con objetivo K pero no M. Se encuentra Regla 4 → Etapa 3.
- Etapa 3. La Regla 4 no puede concluir puesto que se desconocen los valores de C y G. → Etapa 4.
- Etapa 4. Los objetos C y G no están marcados.
 - $\text{ObjetivosPrevios} = \{M, K\}$.
 - Se elige C o G como objetivo en curso. C.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C\}$.
 - Se continúa con la Etapa 2.
- Etapa 2. Se busca regla activa con el objetivo C pero no los previos $\{M, K\}$. Se encuentra la Regla 1 → Etapa 3.
- Etapa 3. La Regla 1 no puede concluir puesto que se desconocen los valores de A y B. → Etapa 4.
- Etapa 4. Los objetos A y B no están marcados.
 - $\text{ObjetivosPrevios} = \{M, K, C\}$.
 - Se elige A o B como objetivo en curso. A.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C, A\}$.
 - Se continúa con la Etapa 2.
- Etapa 2. Se busca regla activa con el objetivo en curso A pero no $\{M, K, C\}$. No se encuentra ninguna regla → Etapa 5.
- Etapa 5. El objetivo A es $\leftrightarrow M$, se pregunta al usuario por el valor de A. Supongamos A cierto, → Etapa 6.
- Etapa 6. El objetivo A no coincide con el previo M. Por tanto, C se designa como objetivo en curso y se elimina de $\text{ObjetivosPrevios} = \{M, K\}$ → Etapa 2.
- Etapa 2. Se busca regla activa con el objetivo C pero no $\{M, K\}$. Se encuentra Regla 1 → Etapa 3.
- Etapa 3. La Regla 1 no puede concluir porque el valor del objeto B es desconocido. → Etapa 4.
- Etapa 4. El objeto B no está marcado.
 - $\text{ObjetivosPrevios} = \{M, K, C\}$.
 - Se elige como objetivo en curso el único objeto no marcado, B.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C, A, B\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca una regla activa que incluya el objetivo B pero no $\{M, K, C\}$. No se encuentra ninguna → Etapa 5.

Ejemplo Encadenamiento de reglas orientado a un objetivo

- Etapa 5. El objeto $B \neq M$, \rightarrow se pregunta al usuario por el valor de B . Supongamos B cierto \rightarrow Etapa 6.
- Etapa 6. Como B no coincide con el inicial M , se designa C como objetivo en curso y se elimina de $\text{ObjetivosPrevios} = \{M, K\} \rightarrow$ Etapa 2.
- Etapa 2. Se busca regla activa con el objetivo C pero no $\{M, K\}$. Se encuentra la Regla 1, por lo que se va a la Etapa 3.
- Etapa 3. Puesto que $A = \text{cierto}$ y $B = \text{cierto}$, entonces $C = \text{cierto}$ por Regla 1. \rightarrow Etapa 6.
- Etapa 6. Como C no coincide con inicial M , se designa objetivo previo K en curso y se elimina de $\text{ObjetivosPrevios} = \{M\} \rightarrow$ Etapa 2.
- Etapa 2. Se busca regla activa con K pero no $\{M\}$. Se encuentra la Regla 4, \rightarrow Etapa 3.
- Etapa 3. La Regla 4 no puede concluir puesto que el valor de G es desconocido. \rightarrow Etapa 4.
- Etapa 4. El objeto G no está marcado.
 - $\text{ObjetivosPrevios} = \{M, K\}$.
 - El único objeto no marcado G se elige como objetivo en curso.
 - $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C, A, B, G\}$.
 - Se va a la Etapa 2.
- Etapa 2. Se busca regla activa con G pero no $\{M, K\}$. Se encuentra Regla 2 \rightarrow Etapa 3.
- Etapa 3. Puesto que $D = \text{cierto}$, $E = \text{cierto}$ y $F = \text{cierto}$, $G = \text{cierto}$ por Regla 2. \rightarrow Etapa 6.
- Etapa 6. Como G no coincide con el inicial M , se designa objetivo previo K como en curso y se elimina de $\text{ObjetivosPrevios} = \{M\} \rightarrow$ Etapa 2.
- Etapa 2. Se busca regla activa con K pero no $\{M\}$. Se encuentra Regla 4 \rightarrow Etapa 3.
- Etapa 3. Puesto que $C = \text{cierto}$ y $G = \text{cierto}$, entonces $K = \text{cierto}$ por Regla 4 \rightarrow Etapa 6.
- Etapa 6. Como K no coincide con M , se designa objetivo previo M en curso y se elimina de $\text{ObjetivosPrevios} = \emptyset \rightarrow$ Etapa 2.
- Etapa 2. Se busca regla activa con el objetivo en curso M . Se encuentra Regla 6 \rightarrow Etapa 3.
- Etapa 3. Puesto que $K = \text{cierto}$ y $L = \text{cierto}$, $M = \text{cierto}$ por Regla 6. \rightarrow Etapa 6.
- Etapa 6. El objetivo en curso M coincide con el inicial. \rightarrow Etapa 7.
- Etapa 7. El algoritmo devuelve el valor:
 $M = \text{cierto}$.

Encadenamiento orientado a objetivo. Conclusión

- En el ejemplo, 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 está en que el conocimiento del objeto L convierte al conocimiento de H , I y J en irrelevante para el conocimiento del objeto M .
- Las estrategias de **encadenamiento de reglas** se utilizan en problemas en los que algunos hechos (ej: síntomas) se dan por conocidos y se buscan algunas conclusiones (ej: 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.
- Como se ilustra en otro ejemplo en el libro, el Modus Tollens es necesario para poder deducir cierto tipo de objetivos.

Compilación de reglas

- Se trata de otra forma de tratar con reglas encadenadas **comenzando con un conjunto de datos (información)** y **tratando de alcanzar algunos objetivos**.
- Cuando ambos, datos y objetivos, se han determinado previamente, las reglas pueden ser compiladas, pueden escribirse los objetivos en función de los datos para obtener las llamadas **ecuaciones objetivo**.

Ejemplo

Compilación de reglas

- Considérese el conjunto de seis reglas anterior 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.

Control de la Coherencia

- En situaciones complejas, incluso los expertos pueden dar **información incoherente** (reglas inconsistentes y/o combinaciones de hechos no factibles).
- Es muy importante **controlar la coherencia del conocimiento** durante:
 - la **construcción** de la base de conocimiento y
 - los procesos de **adquisición** de datos y razonamiento.
- Si la base de conocimiento contiene información **inconsistente** (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** (ej: dándole 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**.

Coherencia de Reglas

- **Definición 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.**

- Un conjunto coherente de reglas puede producir conclusiones contradictorias para algunos conjuntos de valores de los objetos. Basta con que exista un conjunto de valores que conduzcan a conclusiones no contradictorias.

- **Ejemplo Conjunto de reglas incoherentes.** Sean las reglas que relacionan 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.

- Pueden obtenerse las conclusiones:

1. Las Reglas 1–2 son coherentes puesto que, 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).
3. Las Reglas 1–4 son incoherentes porque producen conclusiones contradictorias para todos los posibles valores de A y B.

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

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

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

Coherencia de Reglas

- Un conjunto de reglas puede ser coherente, aunque algunos conjuntos de valores produzcan conclusiones incoherentes. Estos conjuntos se llaman *valores no factibles*. Ej: las Reglas 1–2 son coherentes, aunque producen conclusiones incoherentes en todos los casos en que $A = C$.
 - → el subsistema de control de coherencia eliminará el valor C de la lista de posibles valores del objeto A , permitiendo al usuario seleccionar sólo valores factibles de los objetos.
- **Definición 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.

Así, cualquier valor no factible debe eliminarse 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 Valores no factibles.** Considérese las 4 reglas del ejemplo. El motor de inferencia concluirá:
 1. Las 2 primeras reglas implican que $A \neq C$, ya que $A = C$ siempre conduce a conclusiones inconsistentes. → el valor $A = C$ deberá ser eliminado de la lista de valores factibles de A . Como A es binario → $A = F$.
 2. Las 3 primeras reglas implican que $A = F$ y $B = C$. → el valor $B = F$ deberá ser eliminado de la lista de valores factibles de B .
 3. Las primeras 4 reglas implican que $A = C$, $A = F$, $B = C$ y $B = F$. Por tanto, los valores $\{C, F\}$ son eliminados de las listas de valores de A y B → las listas de valores factibles de los objetos están vacías → incoherentes.

Coherencia de Reglas

- Es suficiente **realizar la comprobación de la coherencia de las reglas una vez**, tras ser introducida cada regla.
- Además, **todos los valores no factibles pueden ser eliminados** de sus correspondientes listas al ser detectados.
- El **conjunto de reglas** que forman el conocimiento debe ser **coherente**;
 - en otro caso, el sistema podrá obtener conclusiones erróneas.
→ **antes de añadir una regla a la base de conocimiento**, hay que **comprobar la consistencia** con el resto.
- Si la **regla** fuese **coherente** con el resto de reglas, se **añadiría a la base de conocimiento**;
 - en caso contrario, se devolvería al experto humano para su corrección.

Ejemplo

Coherencia de reglas

- Supongamos los cuatro objetos: $A \in \{0, 1\}$, $B \in \{0, 1\}$, $C \in \{0, 1, 2\}$ y $D \in \{0, 1\}$ y 5 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 $A = 1$.
- Supongamos que se desea añadir las tres últimas reglas a una base de conocimiento que contiene las dos primeras reglas.
 - ➔ Las Reglas 1 y 3 son incoherentes (tienen la misma premisa pero conclusiones contradictorias).
 - ➔ La Regla 3 debe ser rechazada y el experto humano informado de la razón del rechazo.
 - El experto humano corregirá posibles reglas incorrectas. La Regla 4 entrará en la base de conocimiento, puesto que es coherente con las Reglas 1 y 2. La Regla 5 es incoherente con la Regla 4. Por ello, la coherencia de ambas reglas debe ser comprobada antes de pasar a formar parte de la base de conocimiento.

Coherencia de hechos

- Los **datos/evidencias** suministrados por los usuarios deben ser también **conherentes entre sí y con el conjunto de reglas** de la base de datos.
- El sistema **no** debe **aceptar hechos que contradigan el conjunto de reglas y/o** el conjunto de **hechos** existente en cada instante del proceso.
- Ejemplo: con una BC que contenga las dos primeras reglas del ejemplo, 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 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 valor de C consistente con la BC.
- Antes de conocer los valores de los objetos, existe una solución factible.
- Ejemplo: $A = 0$, $B = 0$, $C = 0$ y $D = 1$ (estos hechos no contradicen la BC).
→ Por ello, la incoherencia surge de que los hechos y las reglas sean inconsistentes.

Estrategias para la coherencia de los hechos

1. **Eliminar 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 **coherentes con las reglas y con el conocimiento previo**.
 - Sea la base de conocimiento del ejemplo y supongamos que al sistema experto se le ha dado la información $A = 0$ y $C = 1$; \rightarrow el sistema debe saber que $B \neq 0$. \rightarrow este valor debe ser eliminado de la lista de posibles valores del objeto B .

Estrategias para la coherencia de los hechos

2. El motor de inferencia debe **comprobar que los hechos conocidos no contradicen el conjunto de reglas.**
 - En la situación anterior, 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 → el usuario podrá dar evidencias contradictorias tales como *Pago = autorizado* y *NIP = incorrecto* en el ejemplo del CA. 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 objeto, **mostrar y aceptar sólo sus valores factibles.**

Estrategias para la coherencia de los hechos

5. **Actualizar continuamente la base de conocimiento** (cuando 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 **cuando una simple unidad de información llega al sistema.**
 - Dar **varias unidades** de información a la vez **puede conducir a incoherencias** en la base de datos. Por ejemplo, dado $A = 0$, no se puede dar la información combinada $B = 0$ y $C = 1$.
 - **El orden** de la información **afecta** a los posibles valores futuros de los objetos que conducen a compatibilidad, (tras dar $A = 0$ se puede dar $B = 0$ ó $C = 1$, pero imponen restricciones diferentes a los posibles futuros valores de los restantes objetos).
 - La **actualización continua** de la base de conocimiento es **muy importante** (**no actualizar implica** la **posibilidad** de que **evidencias contradictorias** puedan convivir **en la base de conocimiento**).
 - **→ el conocimiento debe ser actualizado inmediatamente tras la incorporación de cada hecho.**
 - Así, la eliminación automática de valores no factibles y la actualización continua del conocimiento aseguran la coherencia de la BC.

Ejemplo Coherencia de hechos (Agentes secretos)

- Tenemos 4 agentes secretos, Alberto, Luisa, Carmen y Tomás, cada uno en un país del conjunto: 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.
- No se sabe quién ha enviado cada mensaje, pero se sabe que Tomás miente (¿es un agente doble?) y que los demás agentes dicen la verdad.
- Pregunta: ¿**quién está en cada país?**

Ejemplo Coherencia de hechos (Agentes secretos)

- Se diseña un sistema experto para resolver este problema.
- Se tienen 4 objetos: Alberto, Luisa, Carmen y Tomás.
 - Cada objeto puede tomar uno de 4 valores: Egipto, Francia, Japón o España.
- Tomás es el único que miente, → se considera que un telegrama suyo es siempre falso. → 2 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.

Ejemplo Coherencia de hechos (Agentes secretos)

- Con estas reglas se intentará ahora averiguar el valor que toma el objeto Tomás:

1. Tomás está posiblemente en Egipto. En ese caso se obtienen las conclusiones:

- 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.

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 (el problema es que el conjunto de reglas anterior no contiene esta información) → Egipto es un valor imposible para el objeto Tomás, (Tomás no puede estar en Egipto).

2. Tomás está posiblemente en Japón. En ese caso se obtienen las conclusiones:

- Luisa está en España, por la Regla 2.
- Alberto está en Francia, por la Regla 4.
- Carmen está en Egipto, por la Regla 6.

No hay contradicción → Japón es un valor posible para el objeto Tomás.

Ejemplo Coherencia de hechos (Agentes secretos)

- Con las reglas anteriores **el motor de inferencia no puede concluir en qué país está cada agente** (las reglas *no contienen la información "sólo un agente puede estar en cada país."*)
- Se obtiene un **conjunto de reglas adicionales** sobre esta información.
- Cada país puede estar ocupado por exactamente un agente, → Dado un agente en un país dado, se necesitan 3 reglas para garantizar que ninguno de los restantes está en ese país. Dado que se tienen 4 agentes, resultan 12 reglas (3 reglas × 4 agentes). Sin embargo, si se utiliza la regla de inferencia **Modus Tollens**, sólo son necesarias seis reglas, pues las restantes resultan redundantes. Ej: para Egipto:
 - 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.

Ejemplo Coherencia de hechos (Agentes secretos)

- Ojo! Existen un conjunto de seis reglas equivalentes a las anteriores.
- Ejemplo: la regla
 - Regla 14a: Si Tomás está en Egipto, entonces Carmen no está en Egipto, es equivalente a (Modus Tollens) la Regla 14.
 - ➔ Se necesitan sólo seis reglas por país.
- Los conjuntos de seis reglas para cada uno de los restantes países se generan de forma similar.
 - ➔ Se tienen un total de 24 reglas adicionales que representan el hecho de que exactamente un agente puede estar en cada país.

Explicación de conclusiones

- Como comentáramos en el tema 1, **las conclusiones no bastan** para satisfacer al usuario de un sistema experto.
- Normalmente, **los usuarios esperan** que el sistema proporcione 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 concluido) forman la base del mecanismo de explicación**, 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** → conoce de qué regla procede cada conclusión.
- El sistema puede **dar al usuario la lista de hechos concluidos junto con las reglas que se han utilizado** para obtenerlos.

Ejemplo explicación de conclusiones

- Sean las reglas que vemos a la derecha.
- Supongamos A,B,D,E, F,H, y I ciertos y los restantes objetos con valores desconocidos.
- Aplicando el algoritmo de encadenamiento y examinando las reglas ejecutadas, el sistema experto puede suministrar la explicación:

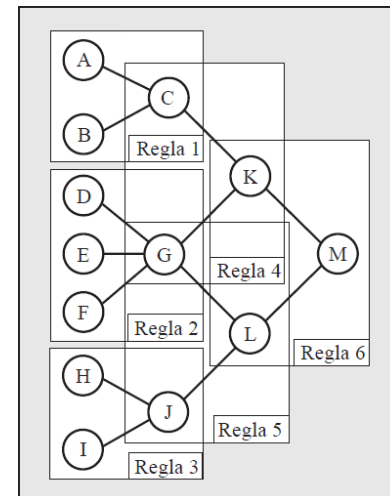
1. **Hechos dados:**

- A = cierto, B = cierto, D = cierto, E = cierto, F = cierto, H = cierto, I = cierto.

2. **Conclusiones y explicaciones:**

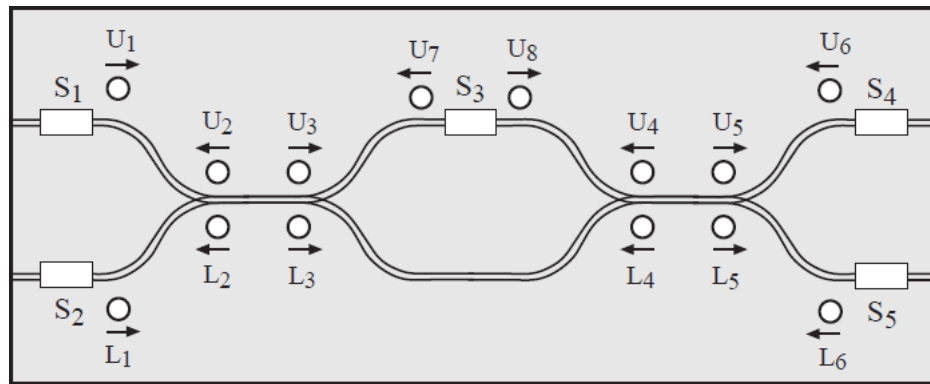
- C = cierto, basada en la Regla 1.
- G = cierto, basada en la Regla 2.
- J = cierto, basada en la Regla 3.
- K = cierto, basada en la Regla 4.
- L = cierto, basada en la Regla 5.
- M = cierto, basada en la Regla 6.

Regla 1 Si A y B Entonces C	Regla 2 Si D, E y F Entonces G	Regla 3 Si H e I Entonces J
Regla 4 Si C y G Entonces K	Regla 5 Si G y J Entonces L	Regla 6 Si K y L Entonces M



Ejemplo de Aplicación

- Los sistemas de control de tráfico actuales son necesariamente complejos.
- Se ilustra cómo diseñar un sistema experto basado en reglas para resolver un problema de control de tráfico muy simple.
- Se 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, U_1, \dots, U_8 , y seis en la parte inferior, L_1, \dots, L_6 .

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

Ejemplo de Aplicación

- El objetivo es diseñar un **conjunto de reglas que eviten la colisión de los trenes.**
 1. Si la señal de tráfico $U1$ está verde, entonces permitirse la salida de un tren en la vía $S1$ y no la salida de los trenes de la vía $S2 \rightarrow L1$ tiene que estar en rojo. Lo mismo para las vías $S4$ y $S5$. Esto da 2 primeras reglas. Si el motor de inferencia utiliza Modus Tollens, estas reglas garantizan que cuando las señales de la parte baja de las vías estén en verde, las señales de sus partes superiores estén en rojo, implicando las dos reglas:
 - Regla 1a: Si $L1 = \text{verde}$, entonces $U1 = \text{rojo}$.
 - Regla 2a: Si $L6 = \text{verde}$, entonces $U6 = \text{rojo}$.
 2. Si la vía $S1$ está ocupada \rightarrow la señal $U2$ debe estar en rojo para evitar que un tren entre en la vía ocupada. Similar para las demás vías, dando lugar a 6 reglas adicionales.

Ejemplo de Aplicación

3. Si ambas señales $U3$ y $L3$ están en rojo \rightarrow ningún tren puede salir de la vía $S1$. La misma condición vale para las señales $U5$ y $L5$.
 - Regla 9: Si ($U3 = \text{rojo}$ y $L3 = \text{rojo}$) o ($U5 = \text{rojo}$ y $L5 = \text{rojo}$), entonces $U1 = \text{rojo}$.
 - De forma análoga 5 reglas asociadas a las otras cinco vías
4. Para evitar la colisión de los trenes procedentes de las vías $S1 - S2$ y $S4 - S5$, son necesarias las reglas siguientes:
 - Regla 15: Si $U3 = \text{verde}$, entonces $U4 = \text{rojo}$
 - Regla 16: Si $L3 = \text{verde}$, entonces $L4 = \text{rojo}$.
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 reglas del tipo: $U2 = \text{verde} \rightarrow L2 = \text{rojo}$.
6. Finalmente, para evitar la colisión de un tren de la vía $S3$ con un tren de las otras cuatro vías, se imponen reglas del tipo:
 - $U1 = \text{verde}$ o $L1 = \text{verde} \rightarrow U7 = \text{rojo}$

Ejemplo de Aplicación

- 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 de abajo y el conjunto de reglas de la derecha.

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

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

Introduciendo Incertidumbre

- Los sistemas basados en reglas descritos pueden aplicarse a situaciones deterministas.
- Sin embargo, hay **muchos casos prácticos** que **implican incertidumbre**.
- Por ejemplo, en el ejemplo 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 la misma.
- Por ello, es **útil extender la lógica clásica para incorporar incertidumbre**.
- Esto se realiza mediante la introducción de varias medidas para tratar la incertidumbre.
 - Castillo y Álvarez describen la aplicación de estas medidas para mejorar los sistemas expertos basados en reglas.
 - Johnson y Keravnou describen algunos prototipos de sistemas expertos basados en lógicas inciertas.
- En el tema siguiente veremos los **sistemas expertos basados en probabilidad**, que *extienden a los sistemas basados en reglas incorporando la incertidumbre*.

Bibliografía

- Enrique Castillo, José Manuel Gutiérrez, Ali S. Hadi.
Sistemas Expertos y Modelos de Redes Probabilísticas
<http://personales.unican.es/gutierjm/papers/BookCGH.pdf>