

# Sistemas Expertos Basados en Reglas

Prof. José Manuel Gutiérrez

Dpto. de Matemática Aplicada. Universidad de Cantabria

## 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 las variables y el conjunto de reglas que definen el problema, y el motor de inferencia obtiene las conclusiones aplicando la lógica clásica a estas reglas. Por regla se entiende una proposición 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”.

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 1 muestra estos objetos y sus posibles valores. La Figura 1 muestra siete reglas que gobiernan la estrategia que el CA debe seguir cuando un usuario intenta sacar dinero de su cuenta.

## 2 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

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}

Table 1: Objetos y posibles valores para el ejemplo del cajero automático.

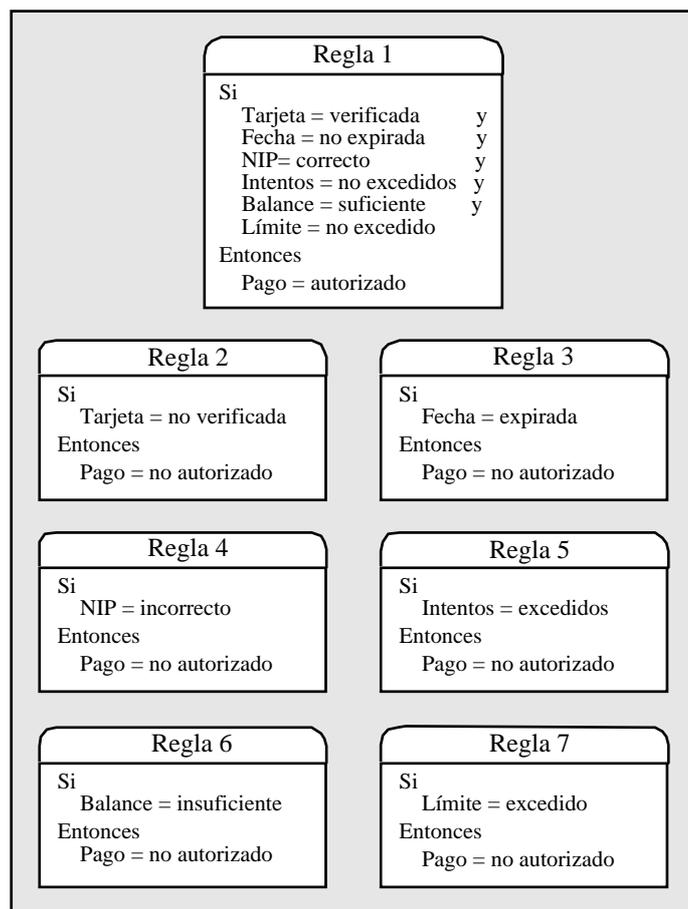


Figure 1: Ejemplos de reglas para sacar dinero de un cajero automático.

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.

Para obtener conclusiones, los expertos utilizan diferentes tipos de reglas y estrategias de inferencia y control (véase, por ejemplo, Durkin [2]). En el resto de esta sección se discuten las reglas de inferencia

- Modus Ponens,
- Modus Tollens,

y las estrategias de inferencia

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

que son utilizadas por el motor de inferencia para obtener conclusiones simples y compuestas.

## 2.1 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”. 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.

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 la regla Modus Tollens concluye que “ $A$  es falso”.

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).

## 2.2 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.

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.

La Figura 2 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 3, 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.

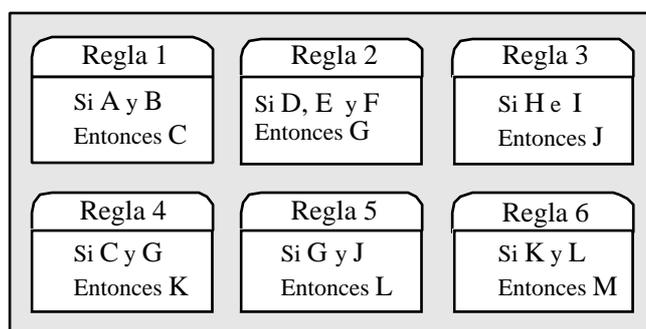


Figure 2: Un ejemplo de un conjunto de seis reglas relacionando 13 objetos.

Supóngase que se dan los hechos  $H = \text{cierto}$ ,  $I = \text{cierto}$ ,  $K = \text{cierto}$  y  $M = \text{falso}$ . Supóngase, en primer lugar, que el motor de inferencia usa las dos reglas de inferencia Modus Ponens y Modus Tollens. En este caso, 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  $L = \text{falso}$ .
3. La Regla 5 concluye (Modus Tollens) que  $G = \text{falso}$  o  $J = \text{falso}$ , pero, puesto que  $J = \text{cierto}$ , deberá ser  $G = \text{falso}$ .

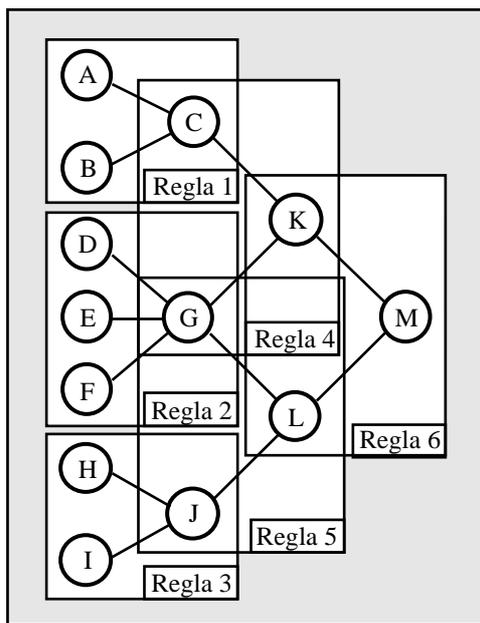


Figure 3: Una representación gráfica de las relaciones entre las seis reglas de la Figura 2.

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.

### 2.3 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 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.

Considérense las seis reglas de las Figuras 2 y 3. 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 4. Las etapas del algoritmo de encadenamiento de reglas orientado a un objetivo se ilustran en la Figura 4, donde el número en el interior de un nodo indica el orden en el que se visita cada nodo. Estas etapas son:

El algoritmo de encadenamiento de reglas orientado al objetivo marcado procedería de la forma siguiente:

- 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

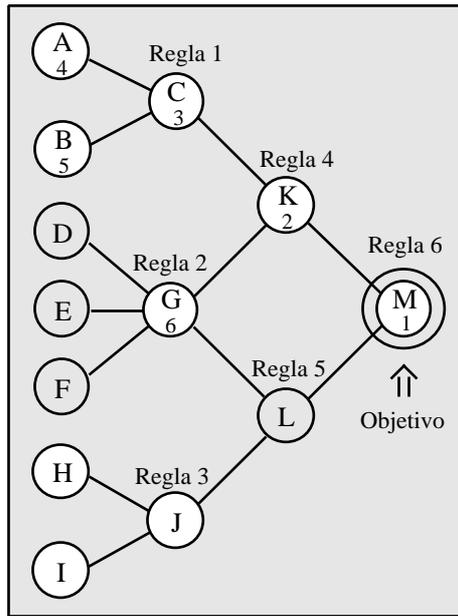


Figure 4: Un ejemplo que ilustra el algoritmo de encadenamiento de reglas orientado a un objetivo. Los nodos cuyo valor es conocido se han sombreado, el nodo objetivo se ha rodeado por una circunferencia, y el número en el interior de un nodo indica el orden en el que se visita cada nodo.

- 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 busca una regla que incluya el objetivo en curso  $M$ .
- La Regla 6 no puede concluir puesto que el valor del objeto  $K$  es desconocido.
- 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 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.
- La Regla 4 no puede concluir puesto que se desconocen los valores de los objetos  $C$  y  $G$ .

- 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 busca una regla activa que incluya el objetivo en curso  $C$  pero no los objetos previos  $\{M, K\}$ . Se encuentra la Regla 1.
- La Regla 1 no puede concluir puesto que se desconocen los valores de los objetos  $A$  y  $B$ .
- 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 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.
- 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  $A = cierto$ .
- 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,  $ObjetivosPrevios = \{M, K\}$ .
- Se busca una regla activa que incluya el objetivo  $C$  pero no los anteriores  $\{M, K\}$ . Se encuentra la Regla 1.
- La Regla 1 no puede concluir porque el valor del objeto  $B$  es desconocido.
- 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 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.
- 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  $B = \text{cierto}$ .
- 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,  $\text{ObjetivosPrevios} = \{M, K\}$ .
- Se busca una regla activa que incluya el objetivo en curso  $C$  pero no los anteriores  $\{M, K\}$ . Se encuentra la Regla 1.
- Puesto que  $A = \text{cierto}$  y  $B = \text{cierto}$ , entonces  $C = \text{cierto}$  por la Regla 1.
- 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,  $\text{ObjetivosPrevios} = \{M\}$ .
- Se busca una regla activa que incluya el objetivo en curso  $K$  pero no los anteriores  $\{M\}$ . Se encuentra la Regla 4.
- La Regla 4 no puede concluir puesto que el valor del objeto  $G$  es desconocido.
- El objeto  $G$  no está marcado. Entonces
  - $\text{ObjetivosPrevios} = \{M, K\}$ .
  - El único objeto no marcado  $G$  se elige como objetivo en curso.
  - Se marca el objeto  $G$ .
  - Por ello,  $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C, A, B, G\}$ .
- Se busca una regla activa que incluya el objetivo en curso  $G$  pero no los anteriores  $\{M, K\}$ . Se encuentra la Regla 2.
- Puesto que  $D = \text{cierto}$ ,  $E = \text{cierto}$  y  $F = \text{cierto}$ , entonces  $G = \text{cierto}$  por la Regla 2. Ahora se va a la 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,  $\text{ObjetivosPrevios} = \{M\}$ .
- Se busca una regla activa que incluya el objetivo en curso  $K$  pero no los anteriores  $\{M\}$ . Se encuentra la Regla 4.
- Puesto que  $C = \text{cierto}$  y  $G = \text{cierto}$ , entonces  $K = \text{cierto}$  por la Regla 4.

- 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,  $ObjetivosPrevios = \phi$ .
- Se busca una regla activa que incluya el objetivo en curso  $M$ . Se encuentra la Regla 6.
- Puesto que  $K = cierto$  y  $L = cierto$ , entonces  $M = cierto$  por la Regla 6.
- El objetivo en curso  $M$  coincide con el inicial. En consecuencia.
- El algoritmo devuelve el valor  $M = cierto$ .

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.

### 3 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.

#### 3.1 Coherencia de Reglas

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.

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

Table 2: Una tabla de verdad que muestra que las Reglas 1 y 2 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

Table 3: Una tabla de verdad que muestra que las Reglas 1–4 son incoherentes.

Es decir, es suficiente que exista un conjunto de valores que conduzcan a conclusiones no contradictorias.

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  $B = F$ .

Las Reglas 1–2 son coherentes puesto que, tal como se muestra en la Tabla 2, para  $A = F$ , no producen conclusiones. Sin embargo, 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 3.

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 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.

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.

### 3.2 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.

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.

## 4 Tratamiento de la Incertidumbre

En la Lógica clásica, lo único que puede deducirse de una regla (“modus ponens” y “modus tollens”) es que si su premisa es cierta, también lo será su conclusión (ver Figura 5). Por tanto, dada la regla: “Si  $A$  es cierto, entonces  $B$  es cierto” puede decirse que  $A$  implica  $B$  con probabilidad 1 (Figura 5)(a)). Sin embargo, este modelo tiene importantes limitaciones, porque son habituales las situaciones prácticas en las cuales no es válido el modelo anterior. Por ejemplo, la presencia de algunos síntomas no garantiza siempre la existencia de una enfermedad. Por tanto, sería muy útil generalizar la lógica clásica y por medio de una lógica “incierto”. En este nuevo contexto, la regla anterior puede generalizarse de la forma siguiente: “ $A$  implica  $B$  con probabilidad  $Pr(B|A)$ ”, donde  $Pr(B|A)$  es la probabilidad de  $B$  dado  $A$  (Figura 5)(b)). Además, el otro extremo de la lógica clásica, representado por la regla: “ $A$  implica  $B$  con probabilidad 0” puede ser generalizado por “ $A$  implica  $\bar{B}$  con probabilidad 1”, donde  $\bar{B}$  es denota el complementario de  $B$  (Figura 5)(c)).

Uno de los problemas relacionados con este tipo de lógica es la propagación de incertidumbre. Obsérvese que ahora cada sentencia (hecho) debe de estar acompañada de una medida de incertidumbre (probabilidad, factor de certeza, etc.) y que, cuando se combinan varios hechos, ha de darse a las conclusiones obtenidas una medida de su incertidumbre. Sin embargo, el problema principal es que aunque se conozca la medida de incertidumbre asociada con las premisas, las conclusiones pueden tener, en teoría, un número infinito de valores inciertos. Además, si se permite la posibilidad de reglas inciertas la cosa se complica aún más.

Este hecho muestra la necesidad del desarrollo de nuevos sistemas expertos, como los basados en probabilidad.

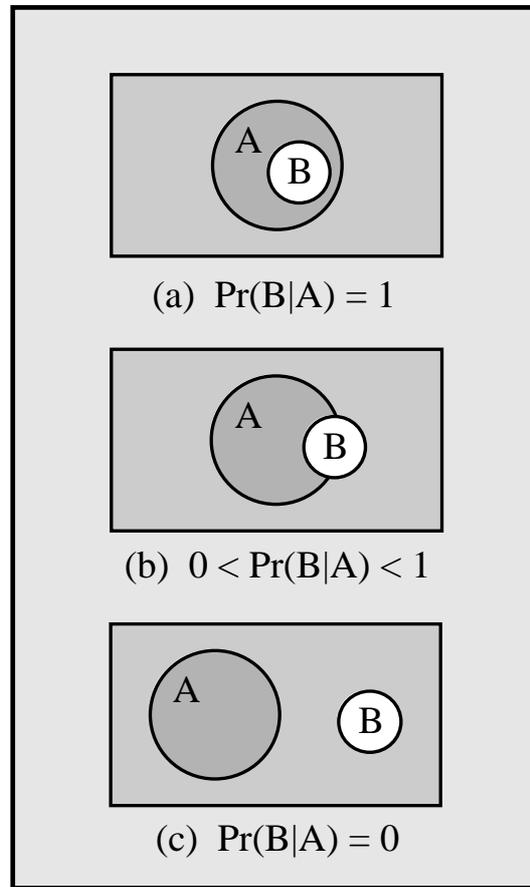


Figure 5: Ejemplos de implicaciones inciertas: (a)  $A$  implica  $B$  con  $\Pr(B|A) = 1$ , (b)  $A$  implica  $B$  con  $\Pr(B|A) = p$ , donde  $0 < p < 1$ , y (c)  $A$  implica  $B$  con  $\Pr(B|A) = 0$ .

## References

- [1] Castillo, E., Gutiérrez, J.M. y Hadi, H (1997), *Expert Systems and Probabilistic Network Models*. Springer, New York. Versión Española editada por la Academia Española de Ingeniería.
- [2] Durkin, J. (1994), *Expert Systems: Design and Development*. Maxwell Macmillan, New York.