

Curso Sistemas Expertos

Basados en Reglas

Introducción

- Los sistemas expertos basados en reglas
 - Constituyen la más sencilla de las metodologías utilizadas en sistemas expertos.
 - Trabajan mediante la aplicación de reglas, comparación de resultados y aplicación de nuevas reglas basadas en una situación modificada.

Introducción

- Son una herramienta eficiente para tratar problemas de tipo determinista.
- Los sistemas expertos basados en reglas se basan en dos conceptos fundamentales
 - La base de conocimiento
 - Contiene las variables y el conjunto de reglas que definen el problema
 - Motor de inferencia
 - Obtiene las conclusiones aplicando la lógica clásica a estas reglas.

Introducción

- Por regla se entiende una proposición lógica que relaciona dos o más objetos incluye dos partes:
 - La premisa
 - 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”.

Base de Conocimiento

- En los sistemas expertos basados en reglas existen dos elementos importantes:
 - La base de conocimiento y Los datos.
 - Los datos, elemento de tipo dinámico, están formados por la evidencia o los hechos conocidos en una situación particular.
 - Dinámicos, quiere decir que pueden cambiar de una aplicación a otra, no son de naturaleza permanente y además se almacenan en la memoria de trabajo.

Base de Conocimiento

- Las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas.
- La base de conocimiento almacena el conocimiento como:
 - Conjunto de reglas y objetos;
 - Estas reglas hacen relación de los objetos.
 - La información que se almacena en esta base de conocimientos es de tipo estática;
 - Significa que no cambia de una aplicación a otra.

Base de Conocimiento

- Ejemplo: Supóngase que se tiene un conjunto de *objetos* y, por simplicidad, que cada objeto puede tener uno y solo uno de un conjunto de posibles valores.
 - Regla 1 :Si nota >9,entonces calificación =sobresaliente.
 - Regla 2 :Si puesto <20 o nota >7, entonces Admitir =si y Notificar = si.

Base de Conocimiento

- La premisa de la Regla 1 consta de una única afirmación objeto-valor,
 - Si nota >9
- Mientras que la Regla 2 constan de dos afirmaciones objeto-valor conectadas por un operador lógico.
 - Si puesto <20 o nota >7
- La conclusión de la regla, es la expresión lógica tras la palabra clave entonces

Definición de una 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*

Reglas

- 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 .

Reglas

- Las expresiones lógicas en ambas, premisa y conclusión de la Regla 1, Son simples,
- Las expresiones lógicas de las premisas y la conclusión de la Regla 2 es compuesta.
- 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 Regla 2 es compuesta.

Ejemplo 1 : Cajero Automático

- 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 cuanto 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.

Ejemplo 1 : Cajero Automático

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 }

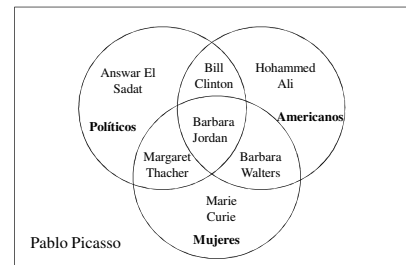
Ejemplo 1 : Cajero Automático

Regla1 Si Tarjeta =verificada y Fecha =no expirada y NIP=correcto y Intentos =no excedidos y Balance =suficiente y Límite =no excedido Entonces Pago =autorizado	Regla2 Si Tarjeta =no verificada Entonces Pago =autorizado	Regla3 Si Fecha =expirada Entonces Pago =autorizado
Regla4 Si NIP =incorrecto Entonces Pago =autorizado	Regla5 Si Intentos =excedidos Entonces Pago =autorizado	
Regla6 Si Balance =insuficiente Entonces Pago =autorizado	Regla7 Si Límite =excedido Entonces Pago =autorizado	

Ejemplo 2 : Gente famosa

- Se dispone de una base de datos consistente en individuos. Que contiene cuatro atributos: nombre, sexo, nacionalidad y profesión. la base de datos muestra sólo si una persona es americana, política y/o si es mujer. Cada uno de estos atributos es binario. En este caso, la base de datos puede contener, como mucho, $2^3=8$ conjuntos disjuntos. Estos conjuntos se muestran en la siguiente figura.

Ejemplo 2 : Gente famosa



Substitución de reglas

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

Substitución de reglas

Si A , entonces B y C	Si A , entonces B Si A , entonces C
Si A , entonces $B \text{ o } C$	Si A y \overline{B} , entonces C Si A y C , entonces B
Si A , entonces $\overline{B \text{ y } C}$	Si A y B , entonces \overline{C} Si A y C , entonces \overline{B}
Si A , entonces $\overline{B \text{ o } C}$	Si A , entonces \overline{B} Si A , entonces \overline{C}

Equivalencia de reglas

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

Motor de Inferencia

- El motor de inferencia usa datos y el conocimiento 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 como las conclusiones derivadas de ellos forman parte de los hechos o datos de que se dispone en un instante dado.

Motor de Inferencia

- 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

Motor de Inferencia

- Reglas de Inferencia
 - Modus Ponens
 - Modus Tollens
 - Resolución
- Estrategias de Inferencia
 - Encadenamiento de reglas,
 - Encadenamiento de reglas orientado a un objetivo,
 - Compilación de reglas

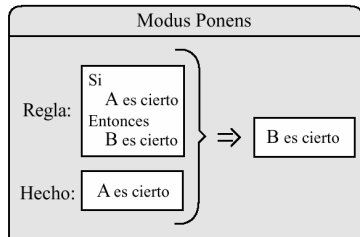
Motor de Inferencia

- Modus Ponens
 - Es 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.

Motor de Inferencia

- Modus Ponens Ejemplo:
 - Se tiene la regla, "Si A es cierto, entonces B es cierto"
 - Se sabe además que "A es cierto."
 - Entonces, 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.

Motor de Inferencia



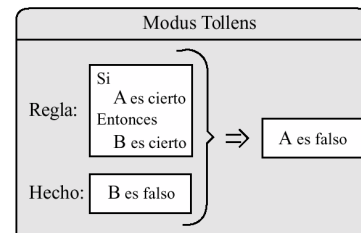
Motor de Inferencia

- Modus Tollens
 - Se utiliza también para obtener conclusiones simples.
 - En este caso se examina la conclusión
 - Si es falsa, se concluye que la premisa también es falsa.

Motor de Inferencia

- Modus Tollens Ejemplo
 - Se tiene la regla, “Si A es cierto, entonces B es cierto”
 - Se sabe que “B es falso.”
 - Entonces, utilizando la regla Modus Ponens no se puede obtener ninguna conclusión,
 - pero, usando, 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.

Motor de Inferencia



Motor de Inferencia

- Modus Ponens y Modus Tollens
 - La regla Modus Ponens se mueve hacia adelante, es decir, de la premisa a la conclusión de una regla.
 - 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 de la premisa para concluir,
 - mientras que la regla Modus Tollens necesita información sobre los objetos de la conclusión.

Motor de Inferencia

- Modus Ponens y Modus Tollens
 - 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

Ejemplo 3: Motor de Inferencia

- 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,

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

Ejemplo 3: Motor de Inferencia

- 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 \neg B, entonces \neg A”

Encadenamiento de Reglas

- Es una de las 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.

Encadenamiento de Reglas

- 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 dando 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.

Ejemplo 4: Encadenamiento de Reglas

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

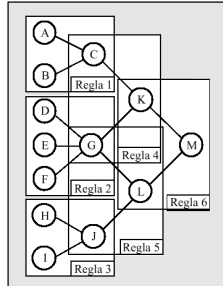
Ejemplo 4: Encadenamiento de Reglas

- Supóngase que se dan los hechos H = cierto, I = cierto, K = cierto y M = falso.
- Supóngase, en primer lugar, que el motor de inferencia usa las dos reglas de inferencia:
 - Modus Ponens y Modus Tollens.
 - La Regla 3 concluye que J = cierto (Modus Ponens).
 - 2. La Regla 6 concluye (Modus Tollens) que K = falso o L = falso, pero, puesto que K = cierto, deberá ser L = falso.
 - 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.

Ejemplo 4: Encadenamiento de Reglas

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.

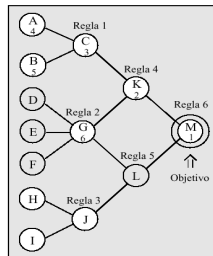


Encadenamiento de Reglas Orientado a un Objetivo (EROO)

- Este algoritmo requiere del usuario seleccionar,
 - 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.

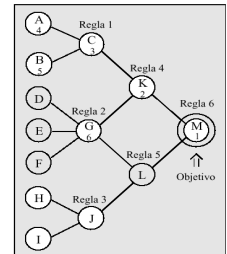
Ejemplo 5: EROO

- Considérense las seis reglas del ejemplo 4
 - Si se selecciona el nodo M como nodo objetivo y que se sabe que los objetos D; E; F y L son *verdad*.
 - Estos nodos están sombreados
 - Las etapas del algoritmo de encadenamiento de reglas orientado a un objetivo se ilustran en dicha figura donde el número en el interior de un nodo indica el orden en el que se visita cada nodo.



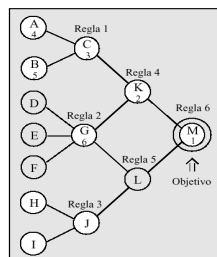
Ejemplo 5: EROO

- Se asigna el valor *Verdadero* 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 Objetos Marcados = { D; E; F; L; M }.
 - Objetos Previos = {}.
 - Las seis reglas están activas. Por tanto, se tiene Reglas Activas = {1;2;3;4;5;6}.



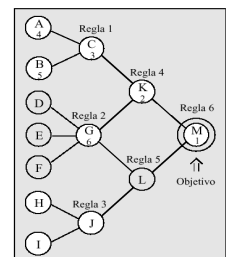
Ejemplo 5: EROO

- Se busca una regla que incluya M.
- La Regla 6 no puede concluir puesto que el valor del objeto K es desconocido.
- El objeto K no está marcado. Entonces
 - Objetos Previos = {M}
 - Se elige el objeto K como objetivo en curso.
 - El objeto K está marcado. Por tanto se tiene, Objetos Marcados = {D;E;F;L;M; K}.



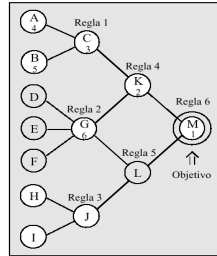
Ejemplo 5: EROO

- 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.



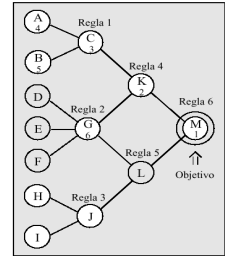
Ejemplo 5: EROO

- Los objetos C y G no están marcados. Entonces
 - Objetivos Previos = {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 Objetos Marcados = {D; E; F; L; M; K; C}.



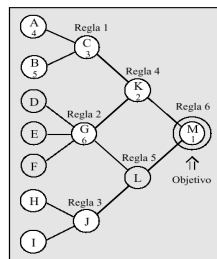
Ejemplo 5: EROO

- Se busca una regla activa que incluya el objetivo en curso C pero no los objetos previos {M; K}. Se encuentra: **Regla 1**.
- La **Regla 1** no puede concluir ya que se desconocen los valores de A y B.
- A y B no están marcados. Entonces
 - Objetivos Previos = {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, Objetos Marcados = {D; E; F; L; M; K; C; A}.



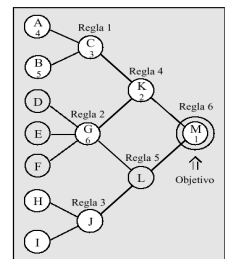
Ejemplo 5: EROO

- Se busca una regla activa que incluya A pero no {M; K; C}. No se encuentra ninguna.
- Puesto que A es diferente de M, se pregunta al usuario por el valor del objeto A. Supóngase que A es *cierto*, entonces se hace A = *cierto*.
- El objetivo en curso A no coincide con M. Por tanto, C se designa como objetivo en curso y se elimina de la lista Objetivos Previos. Por ello, Objetivos Previos = {M; K}.



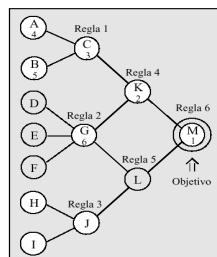
Ejemplo 5: EROO

- Se busca una regla activa que incluya 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
 - Objetivos Previos = {M; K; C}.
 - Se elige como objetivo en curso el único objeto no marcado, B.
 - Se marca el objeto B. Por ello, Objetos Marcados = {D; E; F; L; M; K; C; A; B}.



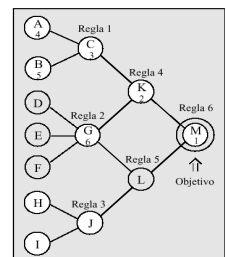
Ejemplo 5: EROO

- Se busca una regla activa que incluya B pero no los previos {M; K; C}. Como no hay ninguna, se va a la Etapa 5.
- Puesto que B no coincide con M, se pregunta el valor de B. Supóngase que B es *cierto*, entonces B = *cierto*.
- Como B no coincide con M, se designa a C como objetivo en curso y se elimina de Objetivos Previos {M; K}.
- Se busca una regla activa que incluya C mas no {M; K}. Se encuentra la Regla 1.



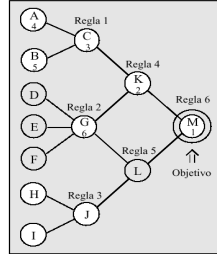
Ejemplo 5: EROO

- Puesto que A = *cierto* y B = *cierto*, entonces C = *cierto* por la Regla 1.
- El objetivo C no coincide con el inicial M. Entonces, K será objetivo en curso y se elimina de Objetivos Previos = {M}.
- Se busca una regla activa que incluya K pero no en los objetivos anteriores {M}. Se encuentra la Regla 4.
- La Regla 4 no puede concluir ya que el valor del objetivo G es desconocido.



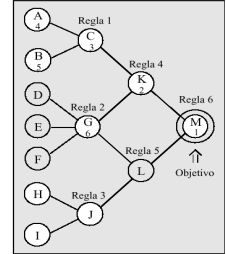
Ejemplo 5: EROO

- El objeto G no está marcado.
Entonces
 - Objetivos Previos = {M; K}.
 - El único objeto no marcado G se elige como objetivo en curso.
 - Se marca el objeto G. Por ello, Objetos Marcados = {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.



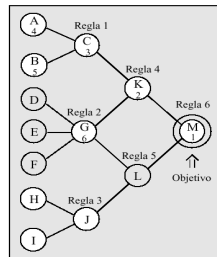
Ejemplo 5: EROO

- Puesto que D = cierto, E = cierto y F = cierto, entonces G = cierto por la Regla 2. Ahora se va a la Etapa 6.
- El objetivo G no coincide con M. Entonces, se designa el objetivo previo K como objetivo en curso y se elimina de Objetos Previos = {M}.
- Se busca una regla activa que incluya K pero no los anteriores (M). Se encuentra la Regla 4.
- Puesto que C = cierto y G = cierto, entonces K = cierto por la Regla 4.



Ejemplo 5: EROO

- K no coincide con M. Entonces, se designa el objetivo previo M como objetivo en curso y se elimina de Objetos Previos = {}.
- Se busca una regla activa que incluya 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 M = cierto.



Encadenamiento de Reglas Orientado a un Objetivo

- 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 éstos sean posibles.

Control de la Coherencia

- El objetivo del control de la coherencia consiste en:
 - Ayudar al usuario a no dar hechos inconsistentes, por ejemplo, dándole al usuario las restricciones que debe satisfacer la información demandada.
 - Evitar que entre en la base de conocimiento cualquier tipo de conocimiento inconsistente o contradictorio.
- Debe hacerse controlando la coherencia de:
 - Las reglas y
 - Los hechos.

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

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.

Coherencia de hechos

- 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.