



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Sistemas expertos

Ángel Sarabia

¿Qué es la Inteligencia Artificial? (1)

Es la rama de la Ciencia que, gracias al desarrollo de los computadores, pretende emular el complejo y poco conocido proceso de razonamiento de los seres humanos, así como su manifestación y transmisión.

¿Qué es la Inteligencia Artificial? (2)

Es la capacidad de un ser vivo o de una máquina de

- ordenar informaciones, extensas observaciones y experiencias,
- descubrir interrelaciones, y
- de esta forma abstraer cosas y ligarlas entre sí.

Alexander Spori, "Spori-Computerbuch", 1971

¿Qué es la Inteligencia Artificial? (3)

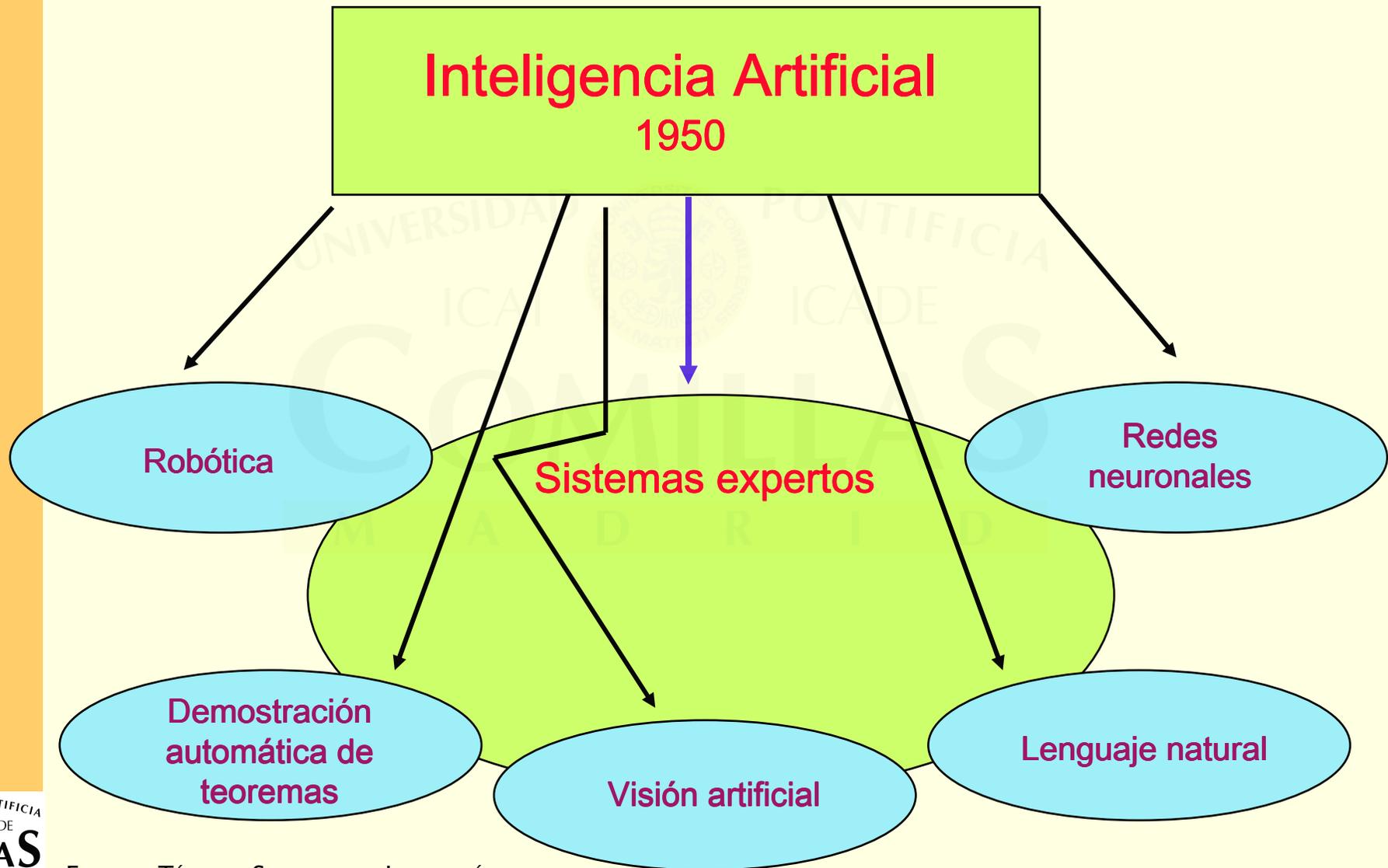
Su herramienta de trabajo es el computador.

Toda computación requiere de una representación de entidades y de un proceso para manipularla.

Sus objetivos son:

- a) Como parte de la Ingeniería Informática, resolver problemas reales, actuando como un conjunto de ideas acerca de cómo representar y utilizar el conocimiento desarrollando sistemas informáticos.
- b) Como parte de la Ciencia, buscar la explicación de diversas clases de inteligencia a través de la representación del conocimiento y de la aplicación que se da a éste en los sistemas informáticos desarrollados.

Campos de aplicación de la I.A.



PARTE I

Los sistemas expertos:

- **qué son**
- **cuando y para qué se utilizan**
- **estructura de los sistemas expertos**
- **ventajas e inconvenientes**

Una persona se dice que es experto en un problema cuando tiene conocimiento profundo y especializado sobre dicho problema.

A este tipo de conocimiento se le llama conocimiento sobre el dominio, y la palabra dominio se usa para enfatizar que ese conocimiento pertenece a un problema específico y concreto.

¿Qué son los Sistemas Expertos?

Son una expresión de los sistemas basados en el conocimiento que constituyen el campo de estudio de la llamada Inteligencia Artificial.

Es un tipo de software que aporta soluciones “*inteligentes*” a problemas, emulando el proceso lógico que un experto humano utiliza en la resolución de problemas.

Un procedimiento de solución de problemas basado en el conocimiento comprende las siguientes capacidades:

- Uso de normas o estructuras que contengan conocimientos y experiencias de expertos especializados.
- Deducción lógica de conclusiones.
- Capacidad para interpretar datos imprecisos y/o ambiguos.
- Manipulación de conocimientos afectados por valores de probabilidad y/o posibilidad.

La función de un SE es la de aportar soluciones a los problemas como si fuera un ser humano, es decir, capaz de mostrar soluciones inteligentes.

HISTORIA DE LOS SISTEMAS EXPERTOS (1)

P. Harmon y D. King, Expert Systems, New York, 1985

Los primeros balbuceos de la Inteligencia Artificial (AI) surgen antes de la II Guerra Mundial con la formulación de diferentes lógicas formales, algunas multivaloradas, y las aportaciones de la Psicología cognoscitiva.

Los acontecimientos precursores directos de la AI aparecen entre los años 1945 y 1954:

- Avances tecnológicos : aparición de ordenadores desarrollados

- Publicaciones fundamentales y estudios teóricos:

 - Simon y Newell desarrollan una teoría de la lógica “ad hoc”

 - H. Simon, “Administrative Behavior”

 - N. Wiener, “Cybernetics” (la teoría de la realimentación)

 - A. Turing, “Computing Machinery and Intelligence”

y conferencias en Macy.

HISTORIA DE LOS SISTEMAS EXPERTOS (2)

P. Harmon y D. King, Expert Systems, New York, 1985

Vienen a continuación los denominados años formativos (1955-60), en los que realmente se produce el inicio de la investigación sobre IA

Las circunstancias que propician dicha investigación

- Avances tecnológicos y de desarrollo de software:
 - incremento en la potencia de los ordenadores
 - Information Processing Language I (IPL-I)
 - LISP, lenguaje para SE, 1958, J. McCarthy
- Publicaciones y reuniones :
 - Seminario de Verano de Darmouth sobre IA (1956)
 - General Problem Solver (GPS), 1957
 - Estudios sobre la psicología del procesamiento de información

HISTORIA DE LOS SISTEMAS EXPERTOS (3)

P. Harmon y D. King, Expert Systems, New York, 1985

Los años de desarrollo y redirección van de 1961 a 1970.

Las circunstancias que lo propician se deben a:

- Avances tecnológicos y de desarrollo de software que abordan la resolución general de problemas:

- LISP (avanzado)
- Algorímic (Métodos heurísticos)
- Robótica
- DENDRAL (de Feigenbaum y Lederberg, U. de Stanford, 1965), el primer SE.
- Programas de ajedrez

- Publicaciones y reuniones :

- A. Newel y H. Simon, "Human Problem Solving"

HISTORIA DE LOS SISTEMAS EXPERTOS (4)

P. Harmon y D. King, Expert Systems, New York, 1985

Los años de especialización y éxitos transcurren de 1971 a 1980

Las circunstancias que lo propician se deben a:

- Avances teóricos, tecnológicos y de desarrollo de software que permiten el descubrimiento de sistemas basados en el conocimiento:

- PROLOG (1972)
- MYCIN (Stanford, 1972)
- TIRESIAS y EMYCIN (Stanford)
- HEARSAY II (Carnegie-Mellon)
- GUIDON (Stanford)
- MACSYMAT (MIT)
- XCON (1979, DEC)
- La ingeniería del conocimiento

- Reconocimientos internacionales :

- H. Simon recibe el Premio Nobel

HISTORIA DE LOS SISTEMAS EXPERTOS (5)

P. Harmon y D. King, Expert Systems, New York, 1985

La carrera de las aplicaciones, la competencia internacional y la entrada en juego de las empresas comerciales se desarrolla a lo largo de los años ochenta.

Las circunstancias que las propician se deben a:

- Avances teóricos y tecnológicos :

- Lógica borrosa
- PROSPECTOR (SRI)
- INTELLECT (AIC)
- Aparición de la microelectrónica y su aplicación a los computadores
- El proyecto japonés de la quinta generación.

- Reconocimientos de la utilidad práctica cara a los problemas ordinarios de empresas y organizaciones de todo tipo.

- Publicaciones:

E. Feigenbaum y P. McCorduck, "La quinta generación"

HISTORIA DE LOS SISTEMAS EXPERTOS (6)

Algunos SE clásicos

MYCIN SE para diagnóstico médico (1972)

TEIRESIAS (1973)

XCON SE para configuración de ordenadores (1979)

DELTA (1983) SE para reparación de locomotoras diesel y eléctricas.

ALDO en DISCO (1984) SE para reparación de calderas hidrostáticas usadas para desinfección de bacterias

LOS LENGUAJES DE DESARROLLO (1)

LISP (List - processing)

- Es un lenguaje funcional que ofrece la posibilidad de realizar definiciones recursivas de funciones. Realiza la unión de procedimientos de forma dinámica, en plena ejecución, y gestiona también de forma dinámica y automática la memoria

- Sus estructuras más importantes son

- listas,
- átomos y
- funciones básicas.

- Tiene muchos dialectos: el COMMONLISP se está imponiendo como estándar.

- Muy extendido en USA

LOS LENGUAJES DE DESARROLLO (2)

PROLOG (Programación Lógica)

- Está basado en la lógica de predicados. Estos predicados aparecen en tres formas distintas:

- como hechos,
- como reglas y
- como preguntas

- Sus mecanismos más importantes son

- recursividad,
- instanciación
- verificación y
- unificación,

- Muy extendido en Europa

Componentes de un sistema experto

- La base de conocimientos
- La base de hechos o datos
- El mecanismo o motor de inferencia
- La interfaz sistema-usuario
 - El submódulo de comunicaciones
 - El submódulo de explicación
- El módulo de adquisición de conocimiento

La base de conocimientos

Contiene el conocimiento y la experiencia de los expertos en un dominio determinado, convenientemente codificado, estructurado y formalizado para el posterior uso del mismo.

El conocimiento se puede representar mediante listas, descripción de los objetos relacionados con el problema en estudio, cálculo de predicados, redes semánticas y las relaciones o reglas de producción entre ellos, así como los procedimientos de aplicación que son importantes para la solución del problema. Todo ello en función del tipo de problema a resolver.

La base de hechos o datos

Es la memoria de trabajo, una memoria temporal auxiliar que almacena los datos del usuario, los datos iniciales del problema y los resultados intermedios obtenidos a lo largo del proceso de consulta y resolución.

A través de ella se puede saber no sólo el estado actual del sistema, sino también como se llegó a él.

Durante una consulta al S.E., el usuario introduce la información que se tiene del problema actual en la base de hechos y el sistema empareja esta información con el conocimiento disponible en la base de conocimiento para deducir nuevos hechos. Por ello, es conveniente que esta información se maneje con bases de datos relacionales.

El motor o mecanismo de inferencia (1)

Es la unidad lógica con la que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas que está configurado imitando el procedimiento humano de los expertos para solucionar problemas.

Funciones del motor de inferencia (2)

1. Determinar las acciones que tendrán lugar, el orden en que lo harán y cómo lo harán entre las diferentes partes del Sistema Experto.
2. Determinar cómo y cuándo se procesarán las reglas y, dado el caso, también la elección de qué reglas deberán procesarse.
3. Controlar el diálogo con el usuario.

La interfaz de usuario (1)

Gobierna el diálogo entre el sistema y el usuario, facilitando a este la posibilidad de consultas en un lenguaje lo más *natural* posible.

Los requisitos más importantes que debe satisfacer son:

1. El aprendizaje de su manejo debe ser rápido.
2. Debe evitarse, en lo posible, la entrada de datos erróneos.
3. Los resultados deben presentarse al usuario en forma clara.
4. Las preguntas y explicaciones deben ser comprensibles.

Para ello, este módulo incorpora dos submódulos:

- el módulo de comunicaciones
- el módulo de explicaciones

Algunas incorporan herramientas de desarrollo de interfaces gráficas.

La interfaz de usuario (2): el módulo de comunicaciones

Permite al sistema experto interactuar con otros sistemas.

En particular este módulo es necesario cuando el sistema experto es utilizado para el control de sistemas automatizados que son gobernados por medio de ordenadores.

La interfaz de usuario (3): el módulo de explicación

- Ayuda al ingeniero de conocimiento a refinar el motor de inferencia y al experto a verificar la coherencia de la base de conocimiento.
- Explica al usuario el proceso aplicado en la resolución de un problema determinado.

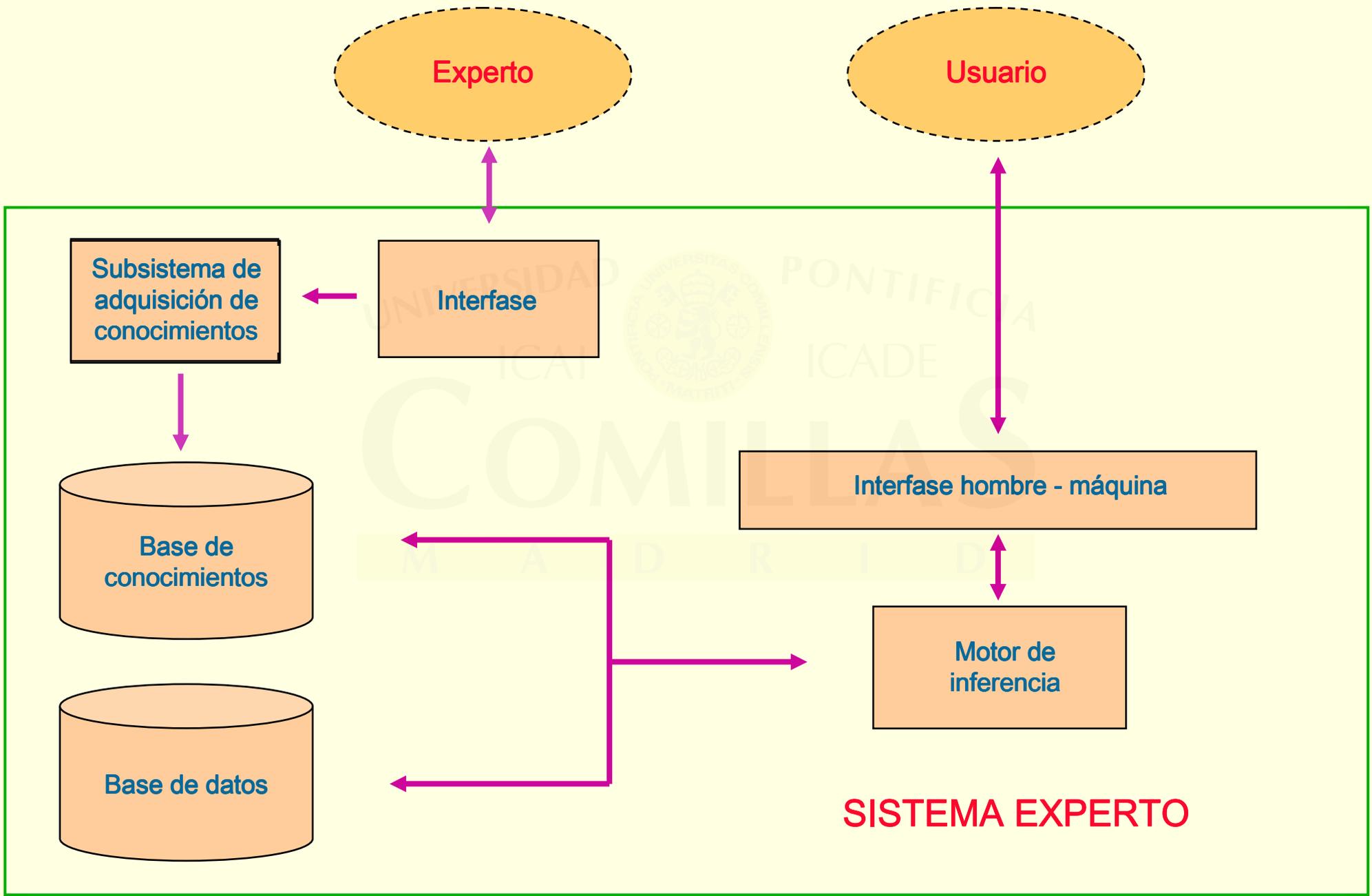
El módulo de adquisición de conocimiento

Es la herramienta que permite al ingeniero de conocimiento y/o al experto centrarse en la estructuración, implementación y actualización del conocimiento, sin tener que dedicar excesivo tiempo a la programación.

Ha de cumplir los siguientes requisitos:

1. El conocimiento (reglas, hechos, relaciones, etc.,) debe poder introducirse de la forma más sencilla posible.
2. Posibilidad de una representación clara de todas las informaciones contenidas en la base de conocimiento.
3. Comprobación automática de la sintaxis del código en el que esté representado el conocimiento.
4. Posibilidad de acceso permanente al lenguaje de programación

Arquitectura de un sistema experto



TIPOS de SISTEMAS EXPERTOS

DETERMINISTAS

PROBABILÍSTICOS

Medidas para tratar la incertidumbre y la imprecisión:

- **Factores de certeza**
- **Probabilidad**
- **Lógica borrosa**

Un SISTEMA EXPERTO PROBABILÍSTICO

Un SE para diagnóstico médico:

- Un conjunto de enfermedades: E_1, E_2, \dots, E_n
- Un conjunto de síntomas asociados: S_1, S_2, \dots, S_n

Supuesto que un paciente presenta un subconjunto S de síntomas, ¿qué enfermedad es más probable que tenga?.

Para responder a esta cuestión, se han de calcular todas las probabilidades condicionadas $P(E_i / S), i=1, 2, \dots, n$

Eso produce una tabla del tipo

Enfermedad	$P(E_i / S)$
1	0.2
2	0.1
3	0.8 más probable
4	0.4
5	0 menos probable
6	0.7

PARTE II

La representación del conocimiento y los mecanismos de razonamiento

Estructuras básicas de representación del conocimiento

- Reglas de producción
- Lógica proposicional
- Redes semánticas
- Marcos
- Objetos
- Representaciones múltiples

Las *reglas* de producción (1)

Las conclusiones del sistema se producen mediante la aplicación de *reglas* sobre los objetos y hechos presentes.

Es la forma más habitual e intuitiva de representación del conocimiento. Consiste en una estructura condicional que relaciona lógicamente la información contenida en la parte llamada *premisa* o *antecedente* con otra información contenida en la parte llamada *conclusión* ó *consecuencia* (o *consecuente*), que, a veces, se traduce en una *acción*.

La *premisa* y la *conclusión* hacen referencia a hechos mencionados en la base de conocimientos

Las *reglas* de producción (2)

Una regla se presenta bajo la forma

SI *premisa/s* **ENTONCES** *conclusión y/o acción*

Ejemplo:

SI el termómetro marca 38,2° C

Y

el termómetro funciona correctamente

ENTONCES

el paciente tiene fiebre

Cuando se aplica una regla sobre unos hechos cualesquiera se dice que la regla se *dispara*. El disparo de una regla provoca la inserción de un nuevo *hecho* en la base de hechos.

Las *reglas* de producción (3)

En muchos sistemas expertos, especialmente los dedicados a diagnóstico, se pueden considerar dos tipos de reglas de producción:

- REGLAS DE DIAGNÓSTICO
- REGLAS DE ACTUACIÓN

Ejemplo:

Base de hechos: El tiempo de ocupación de un obrero es 0.893

Regla de diagnóstico:

Si el tiempo de ocupación del obrero > 0.85
ENTONCES
estado del obrero: ***SOBRECARGADO***

Regla de actuación:

Si estado del obrero es ***SOBRECARGADO***
ENTONCES

- relevar al obrero, o
- parar actividad del obrero durante 30 minutos

Las *reglas* de producción (4)

Su uso facilita la creación y modificación de la base de conocimiento.

Permiten introducir coeficientes de verosimilitud para ponderar el conocimiento y/o la fiabilidad de los expertos, coeficientes que se van propagando durante el proceso de razonamiento según fórmulas preestablecidas.

Cuanto más reglas se consideren más potente es el sistema, aunque el incremento del número de reglas aumenta, a su vez, la dificultad de verificar la coherencia de la base de conocimiento.

Hay sistemas que trabajan con reglas en las que hay incertidumbre sobre el grado de certeza de algunos hechos y sobre el campo de validez de las propias reglas.

Las *reglas* de producción (5)

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

Por tanto, un conjunto coherente de reglas no tiene por qué producir conclusiones no contradictorias para todos los posibles conjuntos de valores de los objetos: es suficiente que exista un conjunto de valores que conduzcan a conclusiones no contradictorias.

Aquellos valores que pueden producir conclusiones inconsistentes, se llaman **valores no factibles**. El subsistema de control de coherencia que todo S.E. debe llevar ha de eliminar automáticamente el valor de todo objeto que conduzca a una inconsistencia. Por ejemplo, rechazándolo si el usuario lo da como dato.

Tabla de verdad de la implicación lógica

Modus Ponens

$$P \Rightarrow Q \equiv \bar{P} \cup Q$$

P	Q	$P \Rightarrow Q$
0	0	1
0	1	1
1	0	0
1	1	1

Modus Tollens

$$P \Rightarrow Q \equiv \bar{Q} \Rightarrow \bar{P}$$

Q	P	$\bar{Q} \Rightarrow \bar{P}$
0	0	1
0	1	1
1	0	1
1	1	0

Nótese que si Q es igual a 1, no se puede afirmar nada respecto a P , mientras que si P es 1, entonces Q es 1.

La lógica proposicional

Es similar a las reglas de producción.

Separa los elementos que componen la base de conocimiento de aquellos que controlan la operación del sistema.

Las redes semánticas

Representan el conocimiento en formas de redes o grafos.

Cada nodo de la red representa un concepto u objeto, y los arcos que los interconectan describen relaciones entre ellos.

MARCOS (*frames*)

Son estructuras de datos donde se almacena información concreta de un cierto concepto e información relacional para completar la definición del concepto.

Relacionados con las redes semánticas.

Asocian propiedades a los nodos que representan los objetos o conceptos.

Las propiedades son descritas en términos de atributos y valores asociados.

OBJETOS

Utilizados en entornos de programación orientada a objetos.

Actúan como entidades independientes, y de esta forma el control del sistema se consigue mediante el envío de mensajes del sistema a los objetos y entre ellos mismos.

Representaciones Múltiples

Son de uso frecuente y combinan dos ó más de las formas anteriores de representación a fin de conseguir una modelización lo más adecuada posible del problema.

Estructuras complejas de representación del conocimiento

- Representación declarativa

- Incluye las reglas, el cálculo de predicados y las redes semánticas.
- El conocimiento queda estructurado en fragmentos independientes entre sí y que, por tanto, son fácilmente modificables.
- Estos módulos de conocimiento se combinan mediante un mecanismo general de razonamiento y deducción.

- Representación procedimental

- Refleja explícitamente las relaciones entre los fragmentos de conocimiento.
- Estas relaciones son difíciles de modificar.
- Facilitan el uso del metaconocimiento

- Representación mixta

- Una combinación de las precedentes.
- Usa objetos y marcos.

Los mecanismos o modos de razonamiento

- Encadenamiento hacia delante

El motor de inferencia parte de los hechos para llegar a los resultados, es decir, selecciona las reglas que verifiquen las condiciones de las premisas.

- Encadenamiento hacia atrás

Se parte de los resultados y se trata de volver a los hechos para comprobar si encajan o no con el problema planteado.

- Encadenamiento mixto

Combina los dos anteriores.

- Algoritmos de búsqueda heurística

Aplicables en los caso en los que la estructura de la B.C. permita construir una estructura de árbol. En ese caso, el proceso de inferencia se convierte en un proceso de búsqueda en un árbol.

- Herencia

Es el método de inferencia usado en entornos orientados a objetos. Un objeto hijo hereda propiedades y hechos de sus padres. Así, la asignación de nuevas propiedades a un objeto se realiza a través de las relaciones entre ese objeto y el resto.

Encadenamiento de Reglas (I)

Es una de las estrategias de inferencia más utilizadas para que un S.E. obtenga conclusiones compuestas.

Puede utilizarse cuando las conclusiones de ciertas reglas coinciden con las premisas de otras.

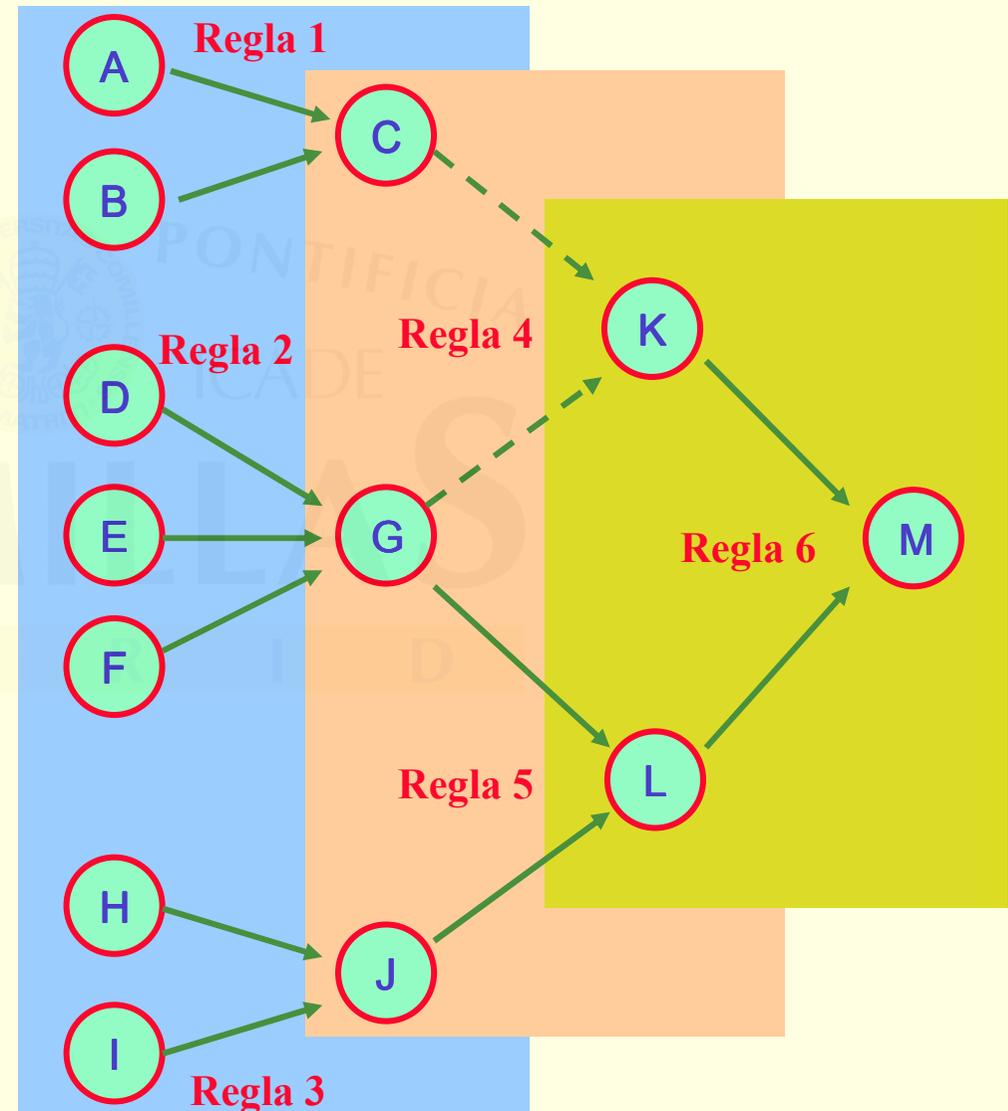
Se utilizan en problemas en los que algunos hechos (por ejemplo, síntomas de un enfermo) se dan por conocidos y se buscan algunas conclusiones (por ejemplo, la enfermedad que puede tener).

Cuando las reglas se encadenan, los hechos pueden utilizarse para dar lugar a nuevos hechos, repitiéndose el proceso hasta que no puedan obtenerse más conclusiones.

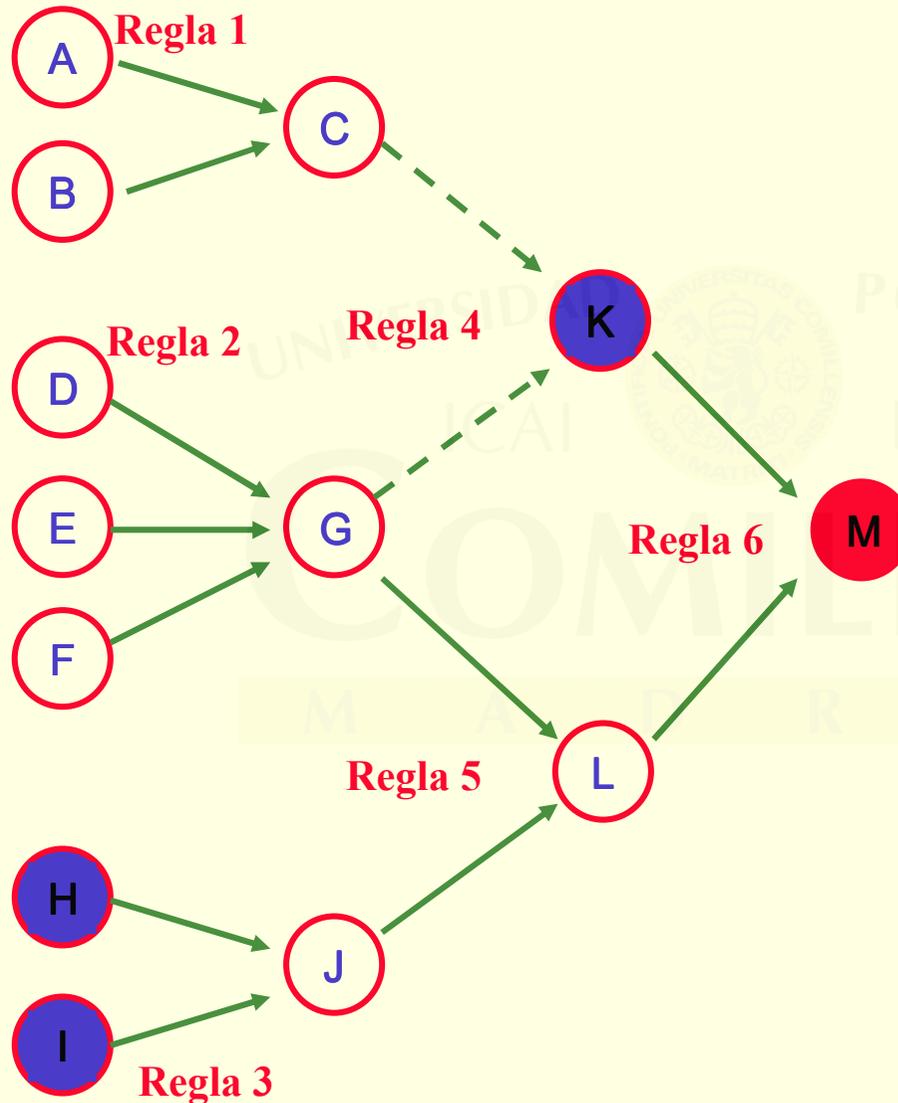
A modo de ejemplo vamos a considerar un S.E. cuyo motor de inferencia usa las dos reglas de inferencia hacia delante (Modus Ponens) y hacia atrás (Modus Tollens) y tiene 13 objetos relacionados mediante 6 reglas. Las reglas y la conexión entre los objetos se muestran en la diapositiva siguiente.

Encadenamiento de Reglas (II). Ejemplo

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



Encadenamiento de Reglas (III). Ejemplo



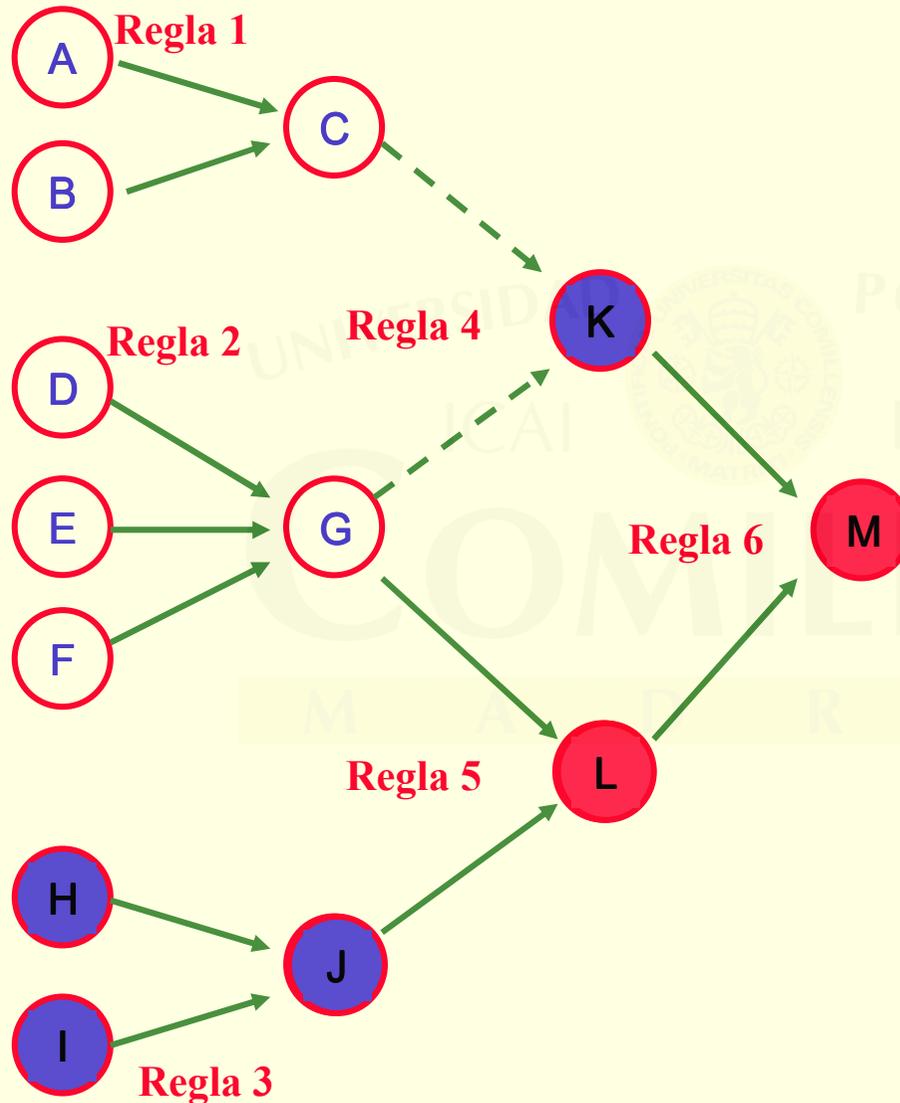
Supóngase que se dan los hechos siguientes:

H: cierto **I: cierto**

K: cierto **M: falso**

circunstancia representada en el gráfico poniendo fondo rojo a los hechos falsos y azul a los verdaderos.

Encadenamiento de Reglas (IV). Ejemplo



La regla de inferencia del Modus Ponens aplicada a la regla 3 da como resultado

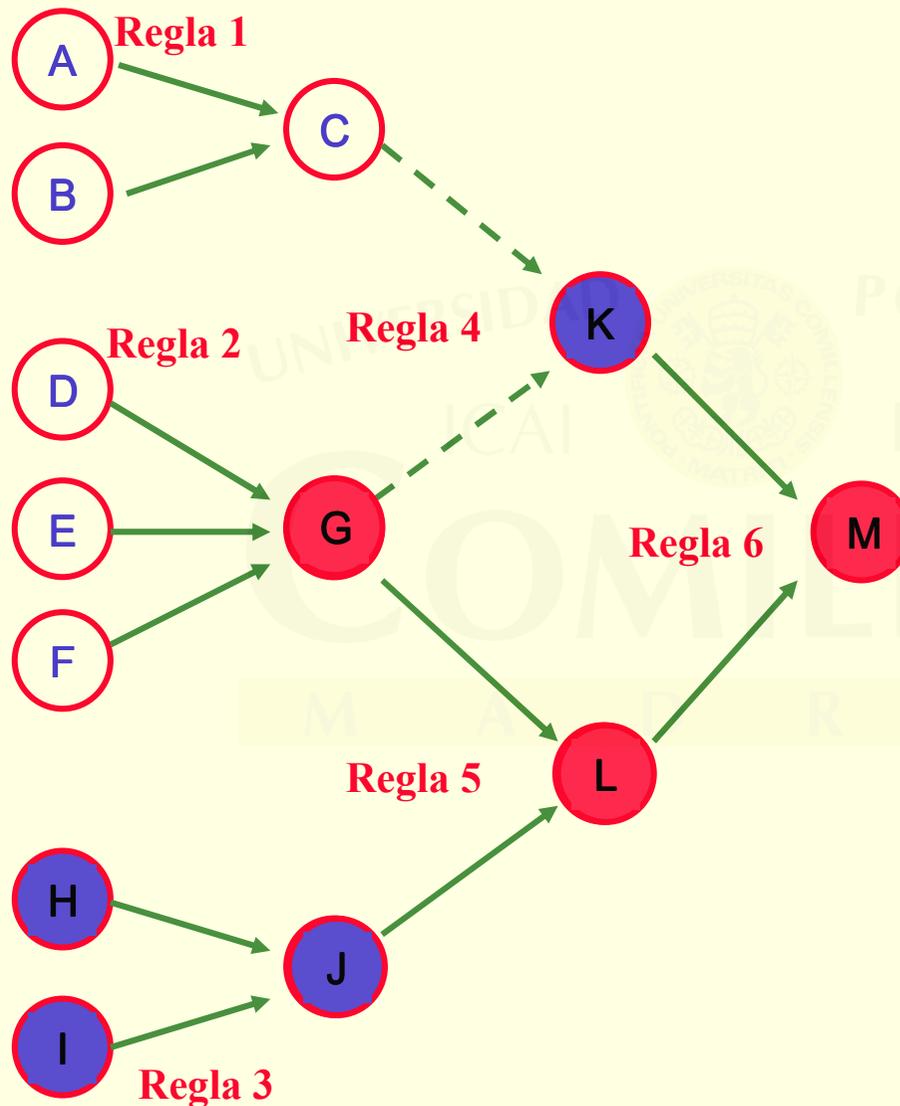
J: cierto

La regla de inferencia del Modus Tollens aplicada a la regla 6 da como resultado

K: falso ó L: falso

Como **K: cierto**, **L** será **falso**.

Encadenamiento de Reglas (V). Ejemplo



La regla de inferencia del Modus Tollens aplicada a la regla 5 concluye que

G: falso ó J: falso

Como J: cierto , G será falso.

Encadenamiento de Reglas Orientado a un Objetivo (I)

Este mecanismo de inferencia requiere del usuario la selección de un nodo objetivo, que se supone centra la atención del problema, y se buscan los hechos que lo hacen posible.

Después, el algoritmo navega entre las reglas en busca de una conclusión sobre el nodo objetivo seleccionado.

Si no obtiene ninguna conclusión con la información inicialmente disponible, el algoritmo “pregunta” al usuario en busca de nueva información pertinente sobre los objetos relevantes en orden a obtener la información necesaria sobre el nodo objetivo

Supóngase que en el S.E que estamos utilizando como ejemplo el nodo objetivo es M y se dan los hechos siguientes:

D: cierto

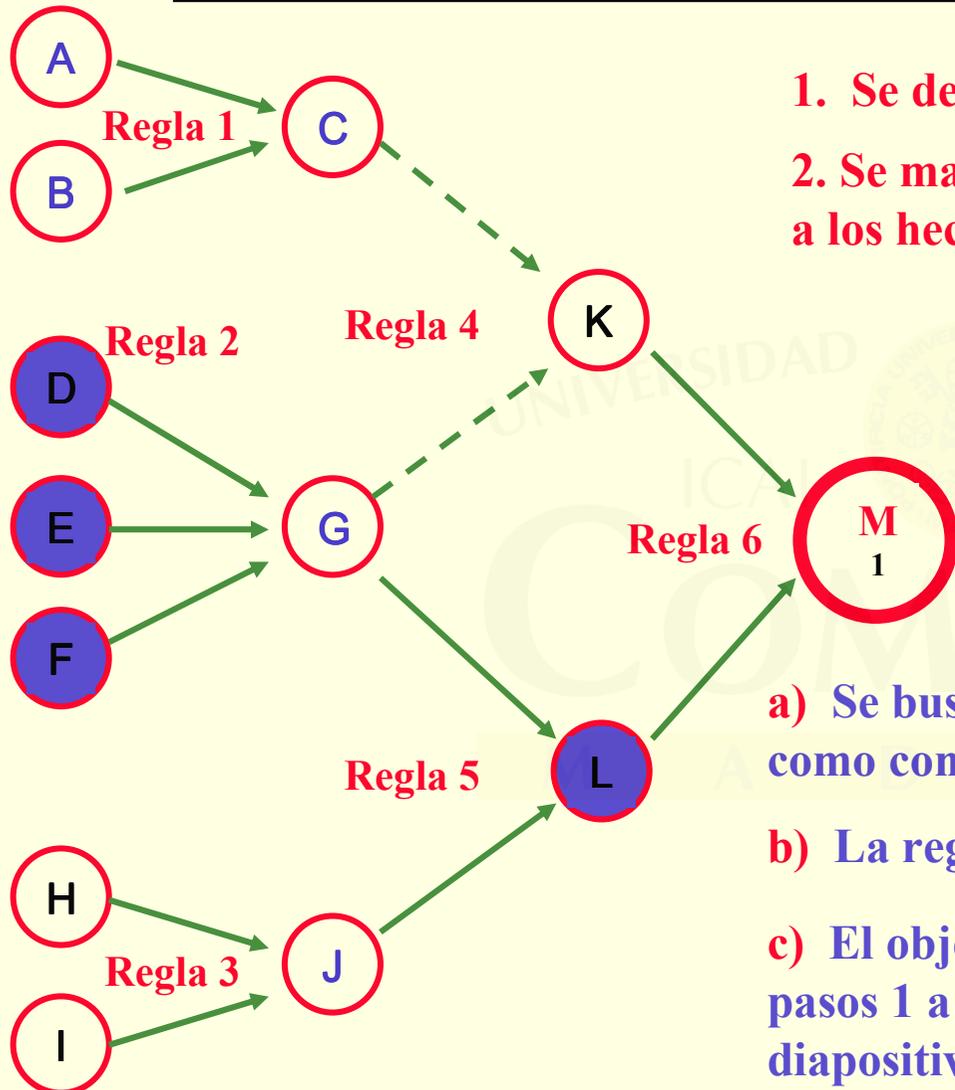
E: cierto

F: cierto

L: cierto

¿qué conclusión puede extraerse respecto al nodo objetivo M? . La situación se representa como indica el siguiente esquema.

Encadenamiento de Reglas Orientado a un Objetivo (II)



1. Se designa el nodo objeto M como *objetivo en curso*.
2. Se marca este objeto y se añade a objetos asociados a los hechos

Objetos marcados

$\{D, E, F, L, M\}$

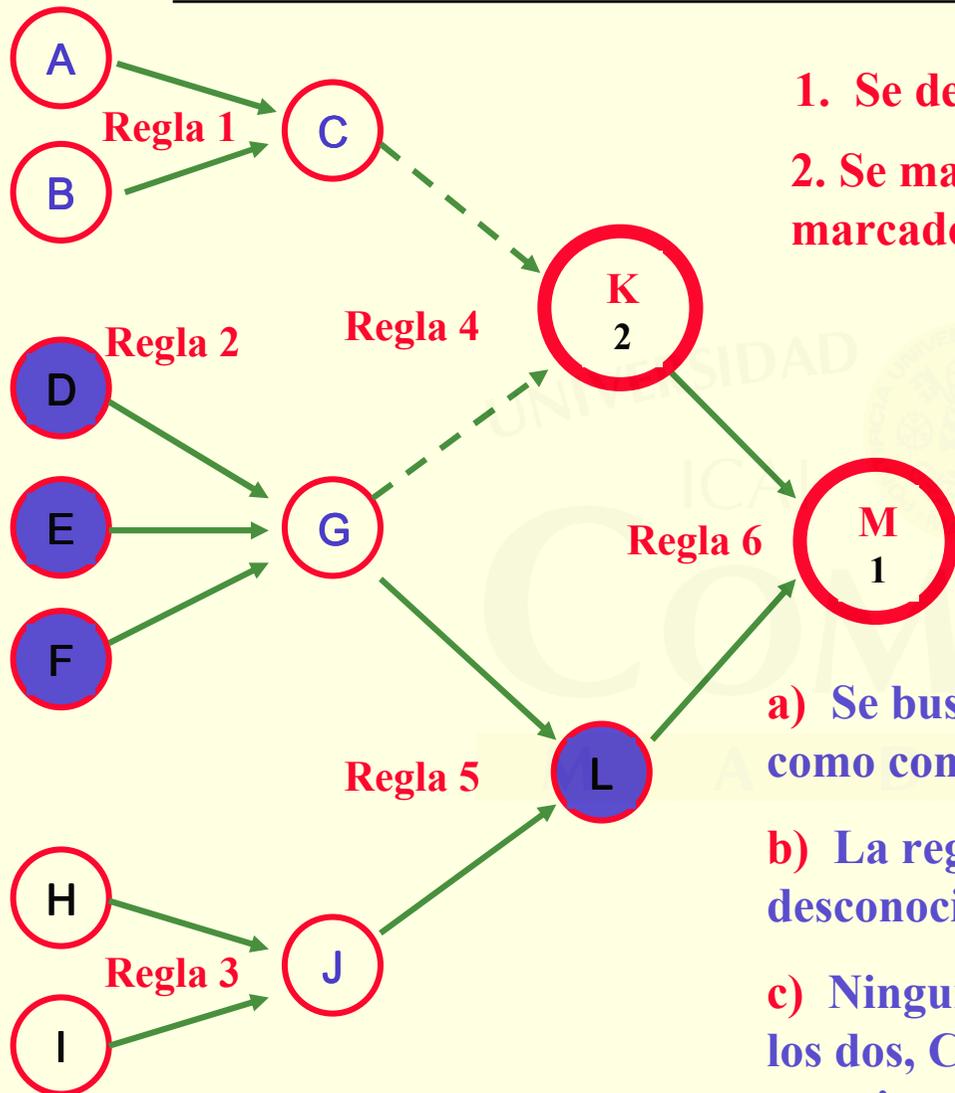
3. Objetivos previos = $\{\emptyset\}$

4. Reglas activas

$\{1, 2, 3, 4, 5, 6\}$

- a) Se busca una regla que incluya al objetivo en curso M como consecuente.
- b) La regla 6 no puede concluir, ya que K es desconocido.
- c) El objeto K no está marcado. Entonces se reiteran los pasos 1 a 4 anteriores, como se indica en la siguiente diapositiva.

Encadenamiento de Reglas Orientado a un Objetivo (III)



1. Se designa el nodo objeto K como *objetivo en curso*.

2. Se marca este objeto y se añade a objetos ya marcados

Objetos marcados

$\{D, E, F, L, M, K\}$

3. Objetivos previos = $\{M\}$

4. Reglas activas

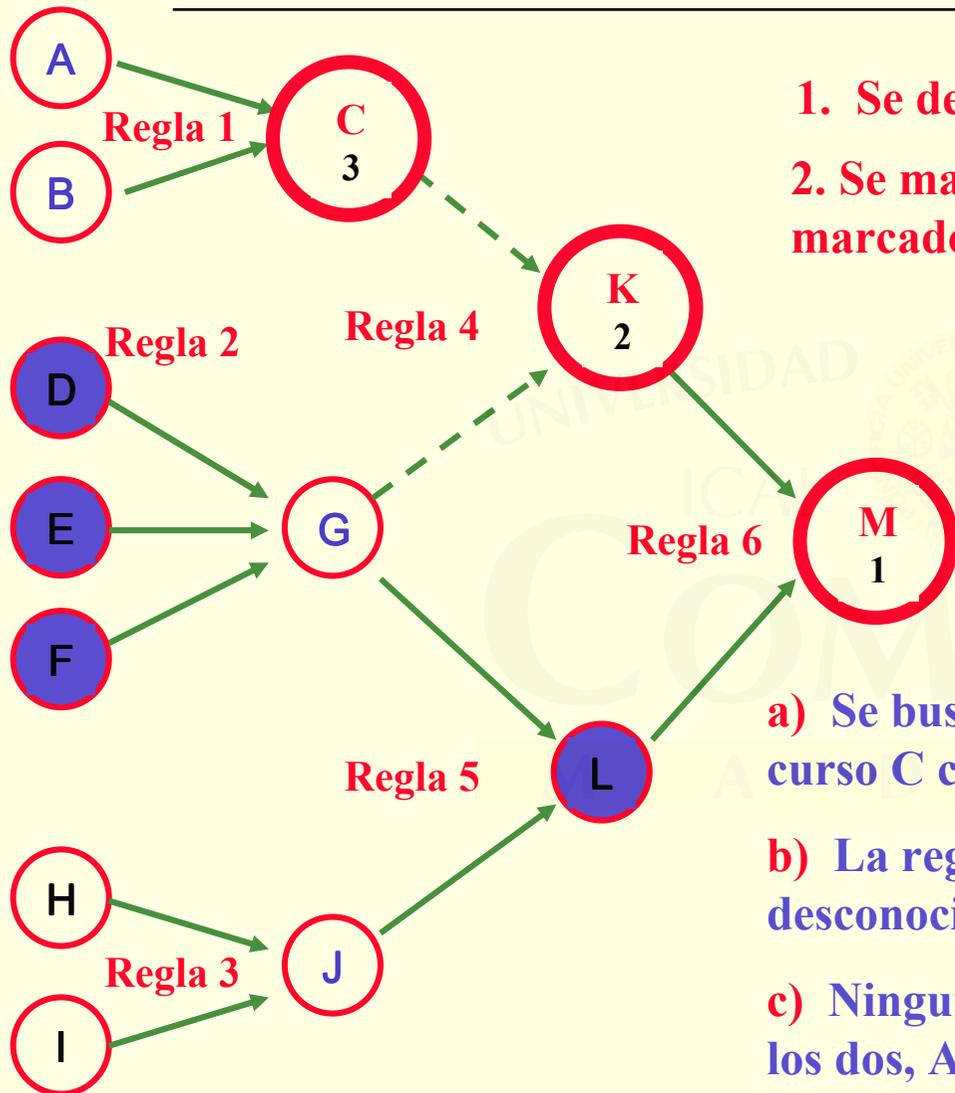
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla que incluya al objetivo en curso K como consecuente, pero no afecte al anterior M.

b) La regla 4 no puede concluir, ya que C y G son desconocidos.

c) Ninguno de los dos está marcado. Se selecciona uno de los dos, C por ejemplo, y se reiteran los pasos 1 a 4 anteriores, como se indica en la siguiente diapositiva.

Encadenamiento de Reglas Orientado a un Objetivo (IV)



1. Se designa el nodo objeto C como *objetivo en curso*.

2. Se marca este objeto y se añade a objetos ya marcados

Objetos marcados

$\{D, E, F, L, M, K, C\}$

3. Objetivos previos = $\{M, K\}$

4. Reglas activas

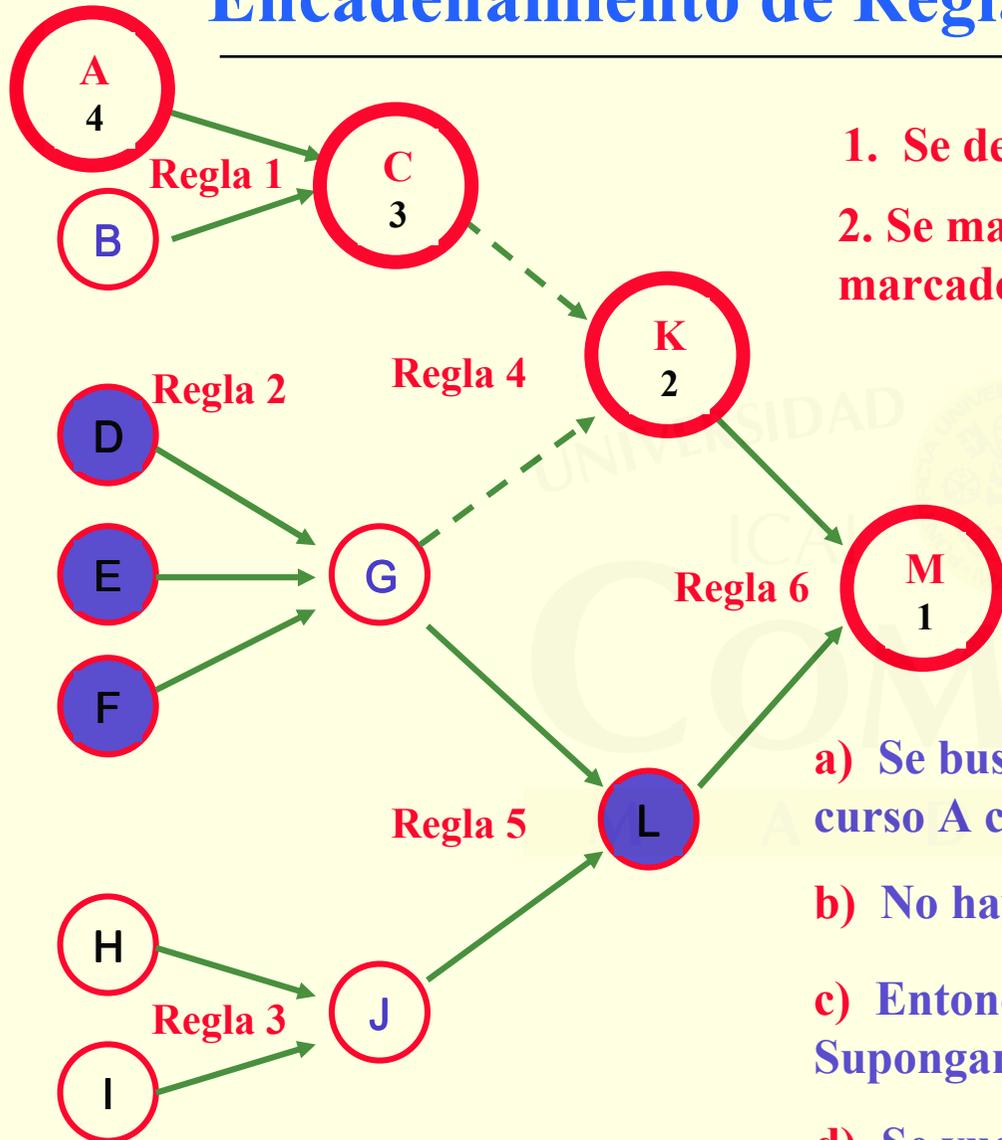
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objetivo en curso C como consecuente, pero no afecte a los previos.

b) La regla 1 no puede concluir, ya que A y B son desconocidos.

c) Ninguno de los dos está marcado. Se selecciona uno de los dos, A por ejemplo, y se reiteran los pasos 1 a 4 anteriores, como se indica en la siguiente diapositiva.

Encadenamiento de Reglas Orientado a un Objetivo (V)



1. Se designa el nodo objeto A como *objetivo en curso*.

2. Se marca este objeto y se añade a objetos ya marcados

Objetos marcados

$\{D, E, F, L, M, K, C, A\}$

3. Objetivos previos = $\{M, K, C\}$

4. Reglas activas

$\{1, 2, 3, 4, 5, 6\}$

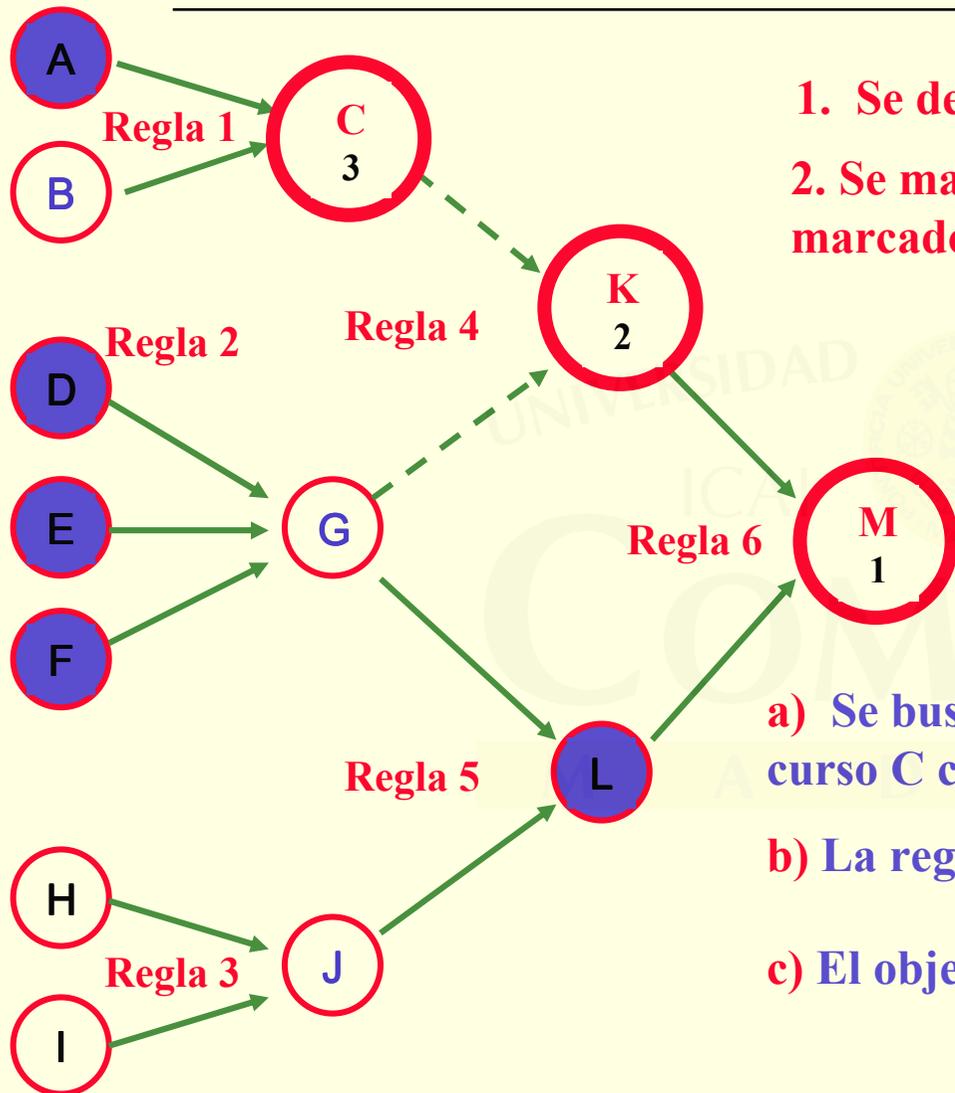
a) Se busca una regla activa que incluya al objetivo en curso A como consecuente, pero no afecte a los previos.

b) No hay ninguna regla con esa característica.

c) Entonces se pregunta al usuario por el valor de A. Supongamos que toma el valor *cierto*.

d) Se vuelve hacia atrás en el árbol, es decir, se vuelve a tomar C como objetivo en curso.

Encadenamiento de Reglas Orientado a un Objetivo (VI)



1. Se designa el nodo objeto C como *objetivo en curso*.

2. Se marca este objeto y se añade a objetos ya marcados

Objetos marcados

$\{D, E, F, L, M, K, C, A\}$

3. Objetivos previos = $\{M, K\}$

4. Reglas activas

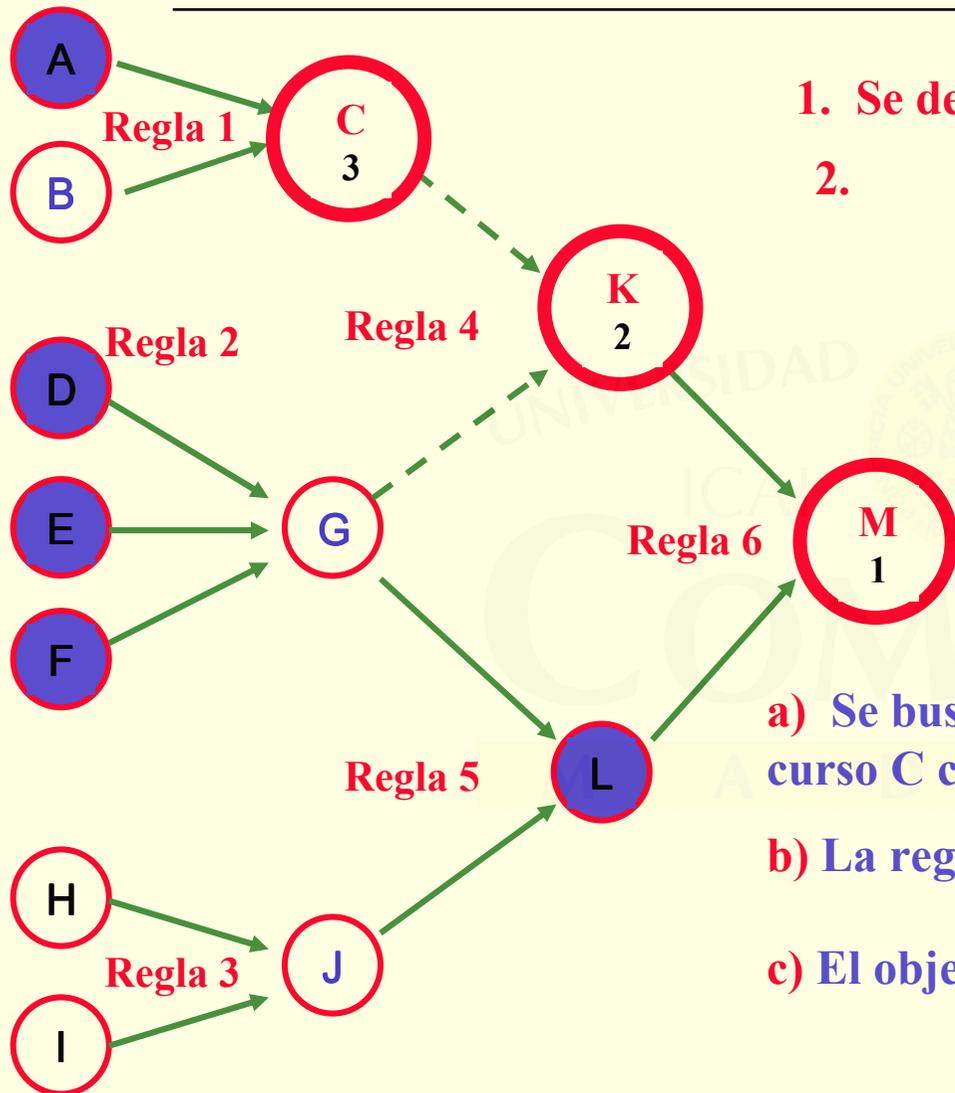
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objetivo en curso C como consecuente, pero no afecte a los previos.

b) La regla 1 no puede concluir, ya que B es desconocido.

c) El objeto B no está marcado.

Encadenamiento de Reglas Orientado a un Objetivo (VII)



1. Se designa el nodo objeto C como *objetivo en curso*.

2.

Objetos marcados

$\{D, E, F, L, M, K, C, A\}$

3. Objetivos previos = $\{M, K\}$

4. Reglas activas

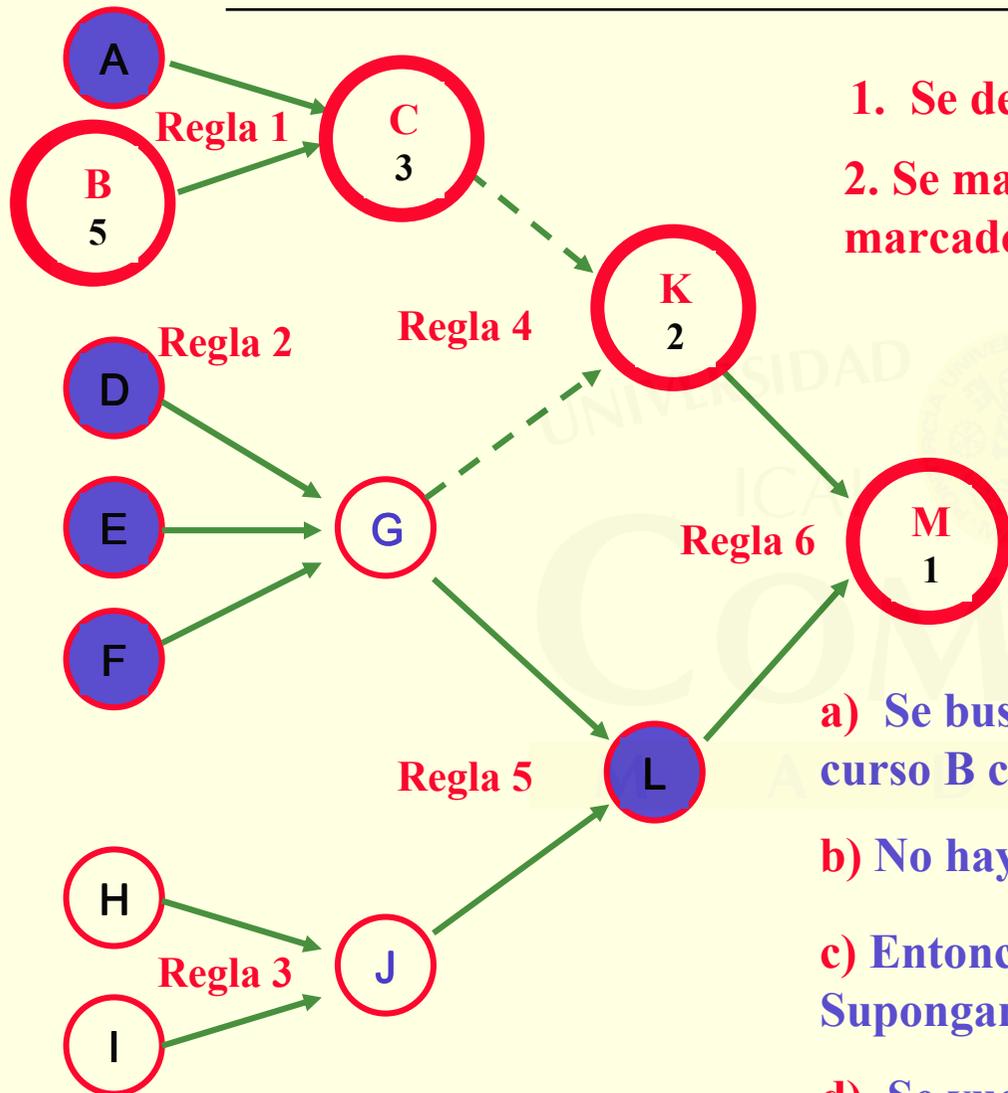
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objetivo en curso C como consecuente, pero no afecte a los previos.

b) La regla 1 no puede concluir, ya que B es desconocido.

c) El objeto B no está marcado.

Encadenamiento de Reglas Orientado a un Objetivo (VIII)



1. Se designa el nodo objeto B como *objetivo en curso*.

2. Se marca este objeto y se añade a objetos ya marcados
Objetos marcados

$\{D, E, F, L, M, K, C, A, B\}$

3. Objetivos previos = $\{M, K, C\}$

4. Reglas activas

$\{1, 2, 3, 4, 5, 6\}$

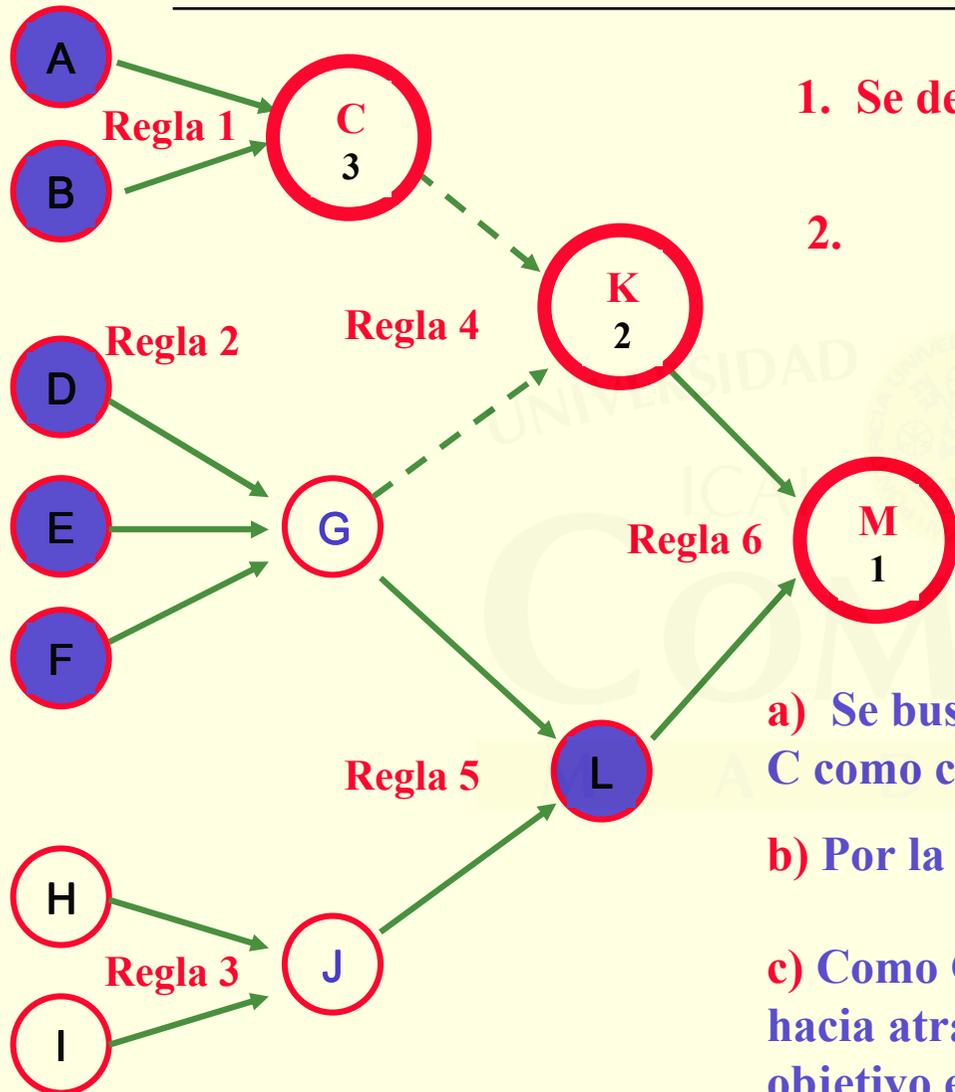
a) Se busca una regla activa que incluya al objetivo en curso B como consecuente, pero no afecte a los previos.

b) No hay ninguna regla con esa característica.

c) Entonces se pregunta al usuario por el valor de B. Supongamos que toma el valor *cierto*.

d) Se vuelve hacia atrás en el árbol, es decir, se vuelve a tomar C como objetivo en curso.

Encadenamiento de Reglas Orientado a un Objetivo (IX)



1. Se designa el nodo objeto C como *objetivo en curso*.

2. *Objetos marcados*

$\{D, E, F, L, M, K, C, A, B\}$

3. Objetivos previos = $\{M, K\}$

4. Reglas activas

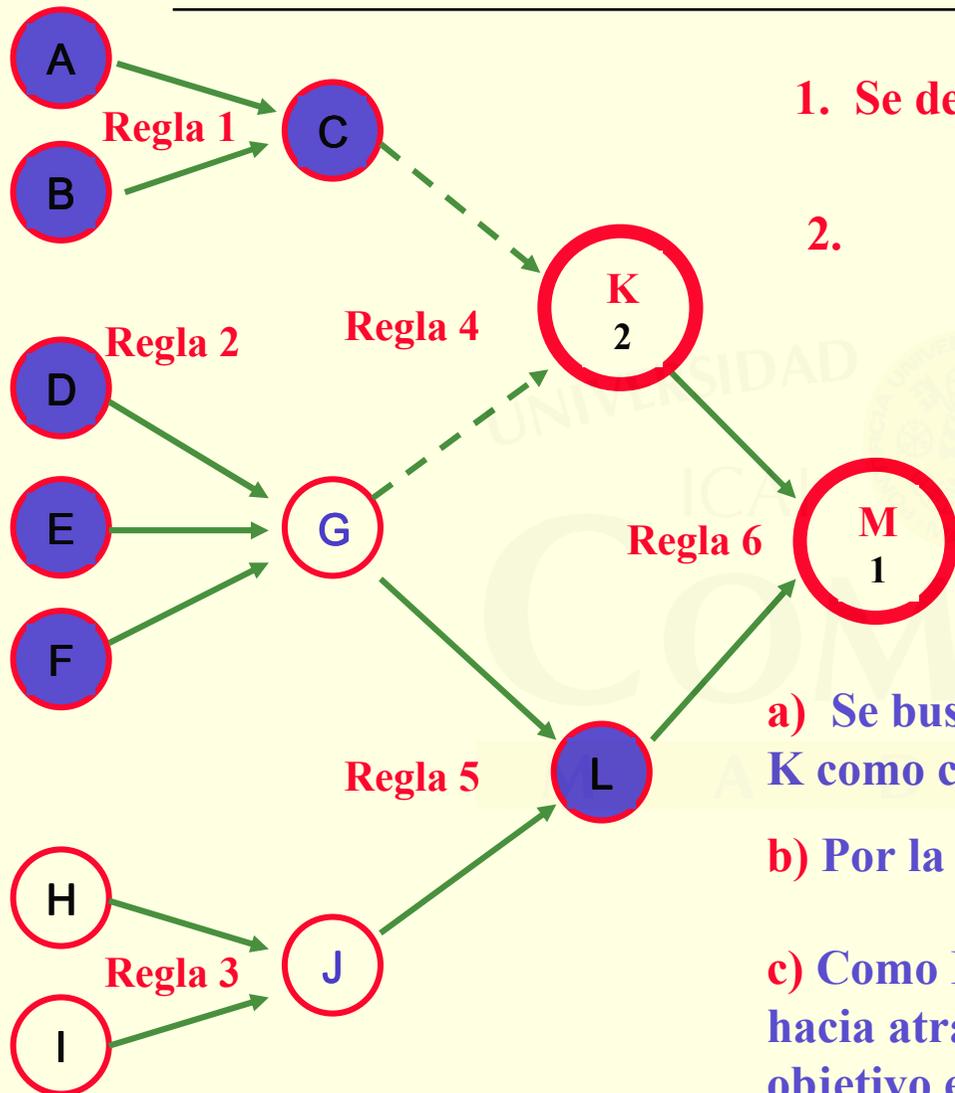
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objeto en curso C como consecuente, pero no afecte a los previos.

b) Por la regla 1 concluimos que C es *cierto*.

c) Como C no coincide con el objetivo inicial, se vuelve hacia atrás en el árbol, es decir, se vuelve a tomar K como objetivo en curso.

Encadenamiento de Reglas Orientado a un Objetivo (X)



1. Se designa el nodo objeto K como *objetivo en curso*.

2. *Objetos marcados*

$\{D, E, F, L, M, K, C, A, B\}$

3. Objetivos previos = $\{M\}$

4. Reglas activas

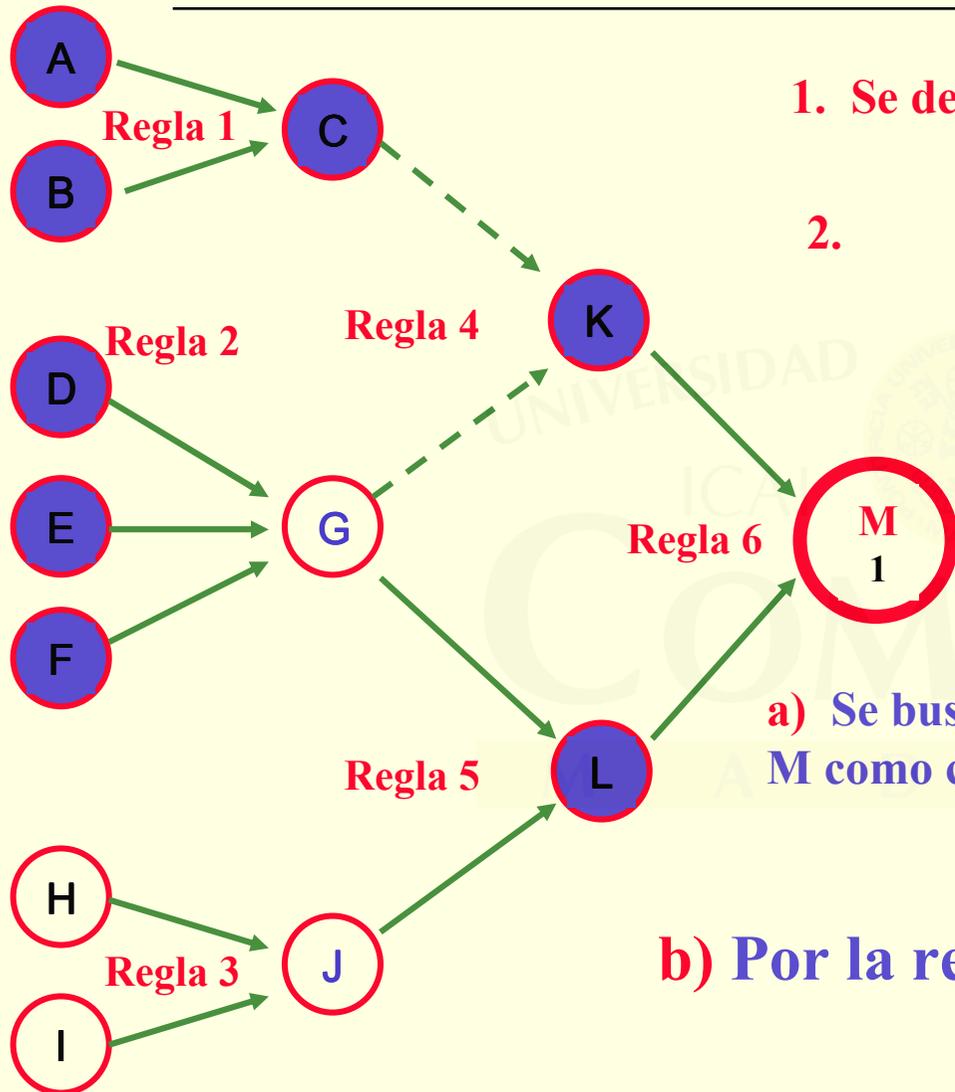
$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objeto en curso K como consecuente, pero no afecte a los previos.

b) Por la regla 4 concluimos que K es *cierto*.

c) Como K no coincide con el objetivo inicial, se vuelve hacia atrás en el árbol, es decir, se vuelve a tomar M como objetivo en curso.

Encadenamiento de Reglas Orientado a un Objetivo (XI)



1. Se designa el nodo objeto M como *objetivo en curso*.

2. *Objetos marcados*

$\{D, E, F, L, M, K, C, A, B\}$

3. Objetivos previos = $\{\emptyset\}$

4. Reglas activas

$\{1, 2, 3, 4, 5, 6\}$

a) Se busca una regla activa que incluya al objeto en curso M como consecuente.

b) Por la regla 6 concluimos que M es *cierto*.

Otro ejemplo sencillo

Supongamos un sistema cuya base de conocimiento contiene las siguientes reglas:

R1: SI *jersey* ENTONCES *bingo*

R2: SI *chaqueta* ENTONCES *cena*

R3: SI *cena* ENTONCES *teatro*

Como actúan los mecanismos de razonamiento (1)

- Encadenamiento hacia delante
(o enfoque guiado por datos o hechos)

Problema: Determinar si se da *teatro* sabiendo que se cumplen *jersey* o *chaqueta*

1º. Lo primero que hace el motor de inferencia es introducir en la B.H. *jersey* y *chaqueta*.

B.H. = { *jersey, chaqueta* }

2º. El sistema identifica las reglas aplicables: R1 y R2.

3º. Selecciona y aplica R1. Esto genera *bingo* , que se añade a la B.H.

B.H. = { *jersey, chaqueta, bingo* }

4º. Como no se ha solucionado el problema, vuelve a identificar un subconjunto de reglas aplicables, excepto la ya usada, que no cambiaría el estado de la B.H.: R2 y R3

5º. Selecciona y aplica R2. Esto genera *cena* , que se añade a la B.H.

B.H. = { *jersey, chaqueta, bingo, cena* }

6º. Como aún no se ha solucionado el problema, el sistema selecciona otro subconjunto de reglas aplicables, en este caso R3.

7º. Selecciona y aplica R3. Esto genera *teatro* , que se añade a la B.H.

B.H. = { *jersey, chaqueta, bingo, cena, teatro* }

8º. Como *teatro* está en la B.H. se ha llegado a una respuesta positiva a la pregunta propuesta.

Como actúan los mecanismos de razonamiento (2)

Encadenamiento hacia atrás
(o enfoque guiado por objetivos)

Problema: Determinar si se da *teatro* teniendo
en la base de hechos *jersey* y *chaqueta*

- 1°. B.H. = { *jersey, chaqueta* }
- 2°. El sistema identifica las reglas aplicables. En este caso sólo R3 ya que es la única que tiene *teatro* como consecuente. Ello genera *cena*.
- 3°. Como *cena* no se encuentra en la B.H., es decir, no es ni verdadero ni falso, se le considera como *subobjetivo*.
- 4°. El sistema intentará probar *cena*, identificando las reglas aplicables: R2
- 5°. Selecciona y aplica R2 y se obtiene *chaqueta*, que es verdadero en la B.H.
- 6°. Como se ha probado el subobjetivo, se prueba también teatro.

Un ejemplo sencillo de diagnóstico de averías (1)

Reglas

R1: SI el motor obtiene gasolina
Y
SI el motor gira
ENTONCES
Problema con las bujías

R2: SI NO gira el motor
Y
SI NO encienden las luces
ENTONCES
Problema con la batería

R3: SI No gira el motor
Y
SI encienden las luces
ENTONCES
Problema con el starter

R4: SI hay gasolina en el depósito
ENTONCES
El motor obtiene gasolina

Posibles problemas identificados

Problema con las bujías
Problema con la batería
Problema con el starter

Un ejemplo sencillo de diagnóstico de averías (2)

Detectando los problemas

a) Problema con las bujías: conclusión de la REGLA 1

Dos nuevas metas: el motor obtiene gasolina
el motor gira

- a1) Para probar la primera podemos usar la REGLA 4, de la que es conclusión
Una nueva meta a probar: hay gasolina en el depósito
No hay regla que concluya esta prueba, por tanto el sistema

PREGUNTA al usuario:

¿Hay gasolina en el depósito?

Supongamos que la respuesta es **SI** (la respuesta se guarda para no repetir la pregunta)

El sistema ha probado que el motor obtiene gasolina

Un ejemplo sencillo de diagnóstico de averías (3)

Detectando los problemas

a) Problema con las bujías: conclusión de la REGLA 1

Dos nuevas metas: el motor obtiene gasolina
el motor gira

a2) Para probar la segunda, como no hay regla que concluya esta prueba, el sistema

PREGUNTA al usuario:

¿Gira el motor?

Supongamos ahora que la respuesta es NO

Como no hay otras reglas que concluyan en que hay problemas con las bujías, el sistema concluye que el problema con las bujías no es la solución del problema, y pasa a considerar otra hipótesis:

Problema con la batería

Un ejemplo sencillo de diagnóstico de averías (4)

Detectando los problemas

b) Problema con batería:

El sistema ya ha registrado que el motor no gira

Tiene que probar que *no encienden las luces*

Como no hay regla que concluya esta prueba, el sistema

PREGUNTA al usuario:

¿Encienden las luces?

Supongamos ahora que la respuesta es **NO**

El sistema concluye que

la causa de la avería es un problema con la batería

Un ejemplo sencillo de diagnóstico de averías (5)

El proceso completo de interacción entre el sistema y el usuario se resume en el siguiente diálogo:

Sistema : ¿Hay gasolina en el depósito?

Usuario: SI

Sistema : ¿Gira el motor?

Usuario: NO

Sistema : ¿Encienden las luces?

Usuario: NO

Sistema : *Hay un problema con la batería*

La interfase de usuario y el módulo explicativo: El diálogo con el SE

Si el usuario desea seguir el *razonamiento* del sistema, se produce el siguiente diálogo:

Sistema : ¿Hay gasolina en el depósito?

Usuario: SI

Sistema : ¿Gira el motor?

Usuario: NO

Sistema : ¿Encienden las luces?

Usuario: ¿Por qué?

Sistema : Trato de probar que hay problema con la batería. Se ha establecido el hecho de que NO funciona el motor. Por tanto si NO encienden las luces habrá un problema con la batería. ¿Encienden las luces?

Usuario: NO

Sistema : Concluyo que *hay un problema con la batería*

Usuario: ¿Cómo?

Sistema : Esto se deduce de la Regla 2:

SI NO gira el motor Y SI NO encienden las luces

ENTONCES *hay un problema con la batería*

NO gira el motor fue dado por el usuario

NO encienden las luces fue dado por el usuario

Un sistema experto para identificar vehículos

Reglas

R1: SI el vehículo tiene dos ruedas
Y
SI utiliza motor
ENTONCES
es una motocicleta

R2: SI tiene dos ruedas
Y
SI NO utiliza motor
ENTONCES
es una bicicleta

R3: SI tiene 4 ruedas
Y
SI utiliza motor
Y
SI pesa menos de 3500 kg
ENTONCES
es un turismo

R4: SI tiene 4 ruedas
Y
SI utiliza motor
Y
SI NO pesa menos de 3500 kg
ENTONCES
es un camión

R5: SI tiene 4 ruedas
Y
SI NO utiliza motor
ENTONCES
es un carrito para transportar pequeños bultos

Vehículos identificados : motocicletas bicicletas turismos camiones carretillas

PARTE III

SISTEMAS EXPERTOS

Historia

¿Por qué y en qué situaciones utilizarlos?

Cómo se desarrollan

Sus principales campos de aplicación

¿ Por qué desarrollar un S.E. ?

El coste de formación de nuevos expertos es muy alto.

Existe el riesgo de perder los conocimientos y experiencia que poseen los expertos.

El número de auténticos expertos en un campo dado es casi siempre pequeño.

Es conveniente disponer de expertos sin restricciones de lugar y/o tiempo.

Los expertos tienen muchas veces que trabajar en ambientes hostiles.

El entorno del problema es rico en conocimiento sobre el mismo y pobre en experiencias previas, casi siempre a causa de su inviabilidad.

No es fácil comparar unos problemas con otros y no se puede aprender del pasado.

La experiencia se puede implementar en forma de reglas.

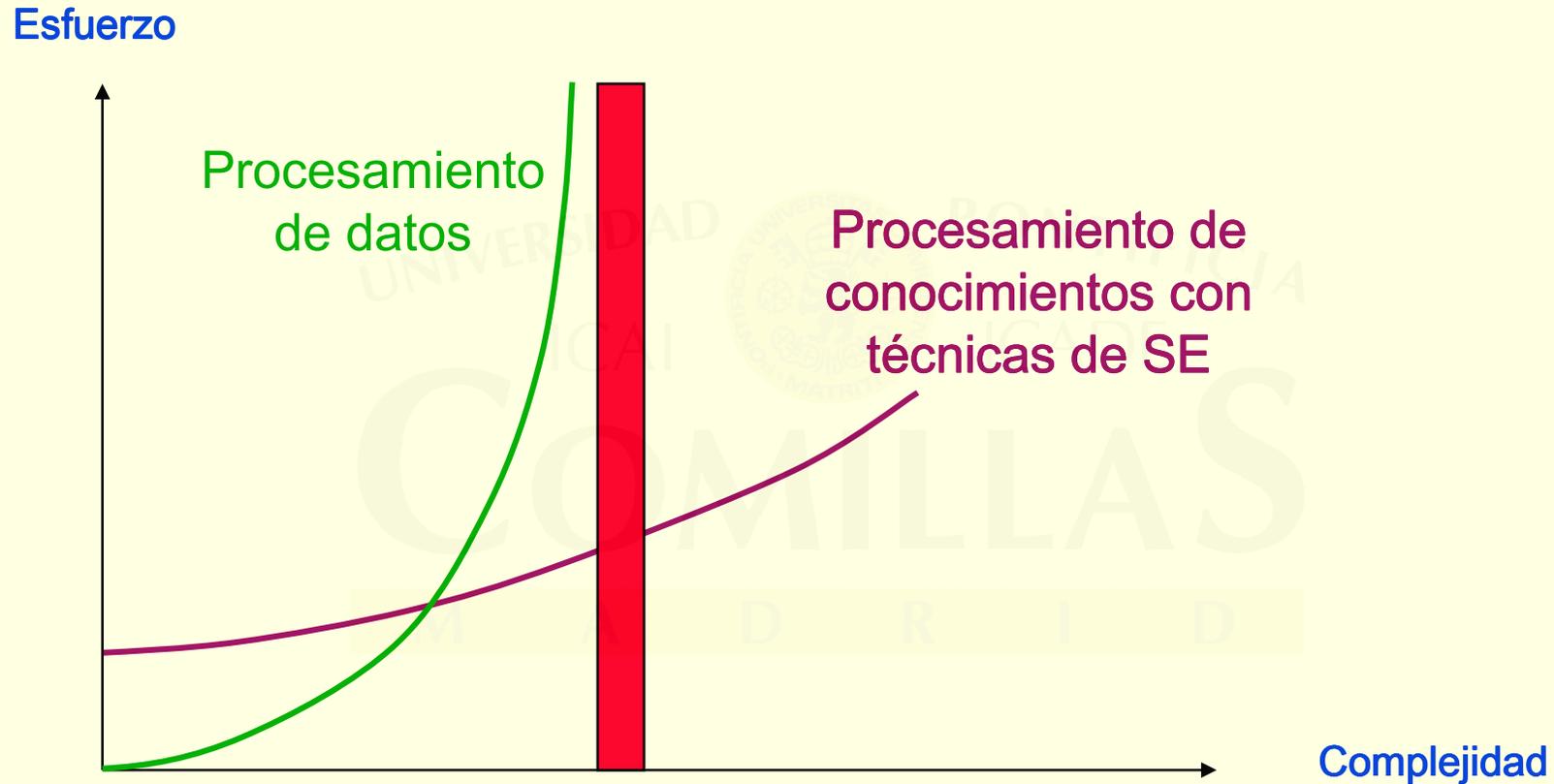
Criterios a tener en cuenta ante un problema para decidir si se desarrolla o no un S.E.

1. La necesidad y el valor de la solución que aporte al problema el sistema experto debe compensar los costes de su desarrollo.
2. Los expertos humanos no están siempre disponibles para resolver cuestiones relacionadas con la situación. O son muy caros de consultar cada vez que se presenta un problema.
3. El problema puede resolverse usando técnicas de tratamiento y razonamiento simbólicos y/o métodos heurísticos. Las tareas a realizar son eminentemente prácticas y no demasiado fáciles en razón de la gran cantidad de datos y relaciones a manejar.
4. El problema está bien estructurado y no requiere mucho conocimiento basado en el sentido común.
5. El problema no puede ser resuelto fácilmente usando métodos de cálculo más tradicionales.
6. Es posible encontrar expertos que cooperen y sean claros en la expresión de su conocimiento.
7. El problema es de tamaño y alcance adecuados, es decir, el número de conceptos que deben ser manejados es limitado y no requiere conocimientos de muchas áreas diferentes.

Cuando tiene sentido su aplicación los sistemas expertos ofrecen una valiosísima ayuda para

- 1. Evitar fallos en labores rutinarias complejas**
- 2. Ampliar de forma más rápida los conocimientos de los especialistas.**
- 3. Diagnosticar los fallos con mayor rapidez y conseguir tareas de planificación más completas, robustas y consistentes.**
- 4. Reducir la carga de los expertos, disminuyendo el riesgo de que tomen decisiones erróneas, y acelerar el proceso de toma de decisiones.**

RELACIÓN ENTRE EL ESFUERZO Y LA COMPLEJIDAD EN EL PROCESAMIENTO DE DATOS



El procesamiento de conocimientos permite superar la barrera de la complejidad

Comparación entre un sistema experto y un sistema clásico de procesamiento de información

Sistema experto

Representa y usa conocimiento

Base de conocimiento separada del mecanismo de procesamiento

Puede contener errores

Un módulo del sistema explica el proceso del tratamiento del conocimiento

El sistema puede operar con sólo algunas de sus reglas

Los cambios en las reglas son fáciles

Puede operar con información incompleta

La ejecución usa métodos heurísticos y lógica

Sistema clásico

Representa y usa datos

El conocimiento y su procesamiento están combinados en un programa

No admite errores en su estructura

No da explicaciones; sólo resultados

El sistema funciona como un todo

Los cambios son tediosos

Necesita información completa para operar

La ejecución se hace paso por paso

Un ejercicio

Una agencia de viajes nos propone diseñar un SE para ayudar a la gente a elegir el lugar donde pasar las vacaciones.

Discutir si esta situación pudiera ser adecuada para desarrollar un SE y, si así fuera, decir como podríamos comenzar a adquirir el conocimiento de experto necesario y que tipo de estructura lógica podría desarrollarse.

Desarrollo de un Sistema Experto

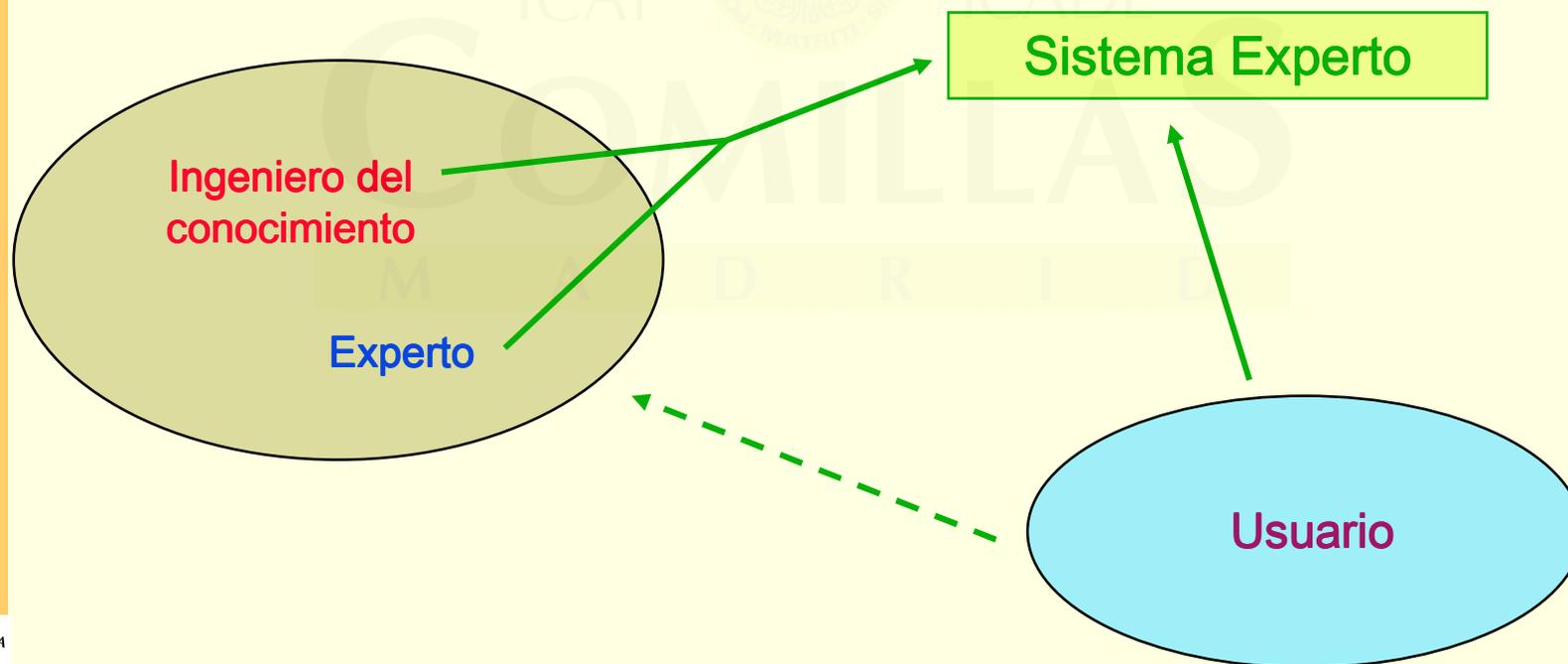
FASES

- Estudio de viabilidad técnica y económica
- Desarrollo del prototipo
- Desarrollo del sistema

Desarrollo del prototipo del Sistema Experto

El equipo

- El experto
- El ingeniero del conocimiento
- El usuario



Alternativas para el desarrollo del prototipo de un SE (1)

La implementación de un SE obliga al desarrollo de cada uno de los elementos del mismo a través de lenguajes o herramientas de desarrollo.

Hay cuatro alternativas para este objetivo:

- Lenguajes de alto nivel
- Lenguajes simbólicos
- Herramientas de desarrollo (conchas y marcos)
- Entornos de desarrollo

Alternativas para el desarrollo del prototipo de un SE (2)

- Lenguajes de alto nivel

Son lenguajes de propósito general (C++, Fortran, etc.,)

Son bien conocidos, su uso está muy generalizado, son eficientes y tienen un alto grado de portabilidad.

Presentan el inconveniente de no estar preparados para una programación basada en símbolos.

Alternativas para el desarrollo del prototipo de un SE (3)

- Lenguajes simbólicos

Son lenguajes de alto nivel, adaptados a la lógica de la base de conocimientos que está representada mediante símbolos, y por su mecanismo para extraer conclusiones.

Su eficiencia aumenta cuando son utilizados en estaciones de trabajo especialmente diseñadas para ellos.

Los más conocidos y usados son **LISP y PROLOG.**

Alternativas para el desarrollo del prototipo de un SE (4)

- Herramientas de desarrollo

Conchas (shells)

Un SE, específico para cada aplicación, que contiene una base de conocimientos vacía, el mecanismo de inferencia, independiente de las bases de hechos y de conocimiento, el componente explicativo y, a veces, la interfase de usuario.

Marcos (frames)

Son parte, no obligada, de las conchas. Contienen nombres, cualidades y valores de objetos. Son componentes explicativos que, en algún caso contienen un mecanismo de inferencia propio, distinto al de las conchas.

Alternativas para el desarrollo del prototipo de un SE (5)

- Entornos de desarrollo (environments)

Añaden a las conchas el uso de un entorno gráfico (ventanas, iconos,...) que facilita el desarrollo del SE.

Tienen la ventaja de disminuir el tiempo de desarrollo, pero suelen ser caros y poco flexibles.

Riesgos en el desarrollo de un SE

1. En muchos casos no existen aplicaciones similares que puedan servir de orientación al ingeniero de desarrollo.
2. Con frecuencia, las especificaciones y requisitos del problema están planteados con poca precisión.

Diseño y especificación requieren una pronta determinación del software a emplear y de la funcionalidad de los componentes.

Modelos funcionales de los sistemas expertos

Categoría	Tipo de problema	Uso
Interpretación	Deducir situaciones a partir de datos observados	Análisis de imágenes Inversiones financieras Reconocimiento del habla
Planificación	Desarrollar planes para alcanzar unos objetivos	Planificación militar Programación de proyectos e inversiones
Control	Mantener el funcionamiento de un sistema dentro de especificaciones dadas	Interpreta, predice y supervisa su funcionamiento Control de tráfico aéreo Monitorización de una planta de producción Estrategia militar
Predicción	Inferir posibles consecuencias a partir de una situación conocida	Previsión de tráfico Evolución de la Bolsa Predicción del tiempo
Diagnóstico	Deducir el estado de un sistema a partir de síntomas	Diagnóstico médico Diagnóstico de averías y detección de fallos Mantenimiento sintomático
Enseñanza	Recoger y mostrar el conocimiento	Aprendizaje de la experiencia ajena
Diseño	Configurar objetos bajo especificaciones dadas	Diseño de edificios, automóviles, circuitos

Otra caracterización de las aplicaciones de los SE por sectores

Sector / Aplicación	Banca Seguros	Industria	Comercio Servicios	Proyectos estatales
Diagnóstico	<ul style="list-style-type: none"> - Concesión de créditos - Comprobación de hipotecas - Análisis de siniestros 	<ul style="list-style-type: none"> - Mantenimiento - Análisis de fallos 	<ul style="list-style-type: none"> - Concesión de créditos - Cálculo de riesgos 	<ul style="list-style-type: none"> - Diagnósticos médicos en hospitales - Diagnóstico técnico
Planificación	<ul style="list-style-type: none"> - Análisis de riesgos - Planificación de Inversiones 	<ul style="list-style-type: none"> - Proyectos - Funciones lógicas de proyectos 	<ul style="list-style-type: none"> - Análisis de mercado - Análisis de riesgos 	<ul style="list-style-type: none"> - Economía energética Planificación de: <ul style="list-style-type: none"> - Inversiones - Emergencias
Control de procesos y supervisión	<ul style="list-style-type: none"> - Gestión de valores - Observación de tendencias. 	<ul style="list-style-type: none"> - Control de procesos - Gobierno de procesos - Aviso de emergencias 	<ul style="list-style-type: none"> - Observación de tendencias. 	<ul style="list-style-type: none"> - Control de centrales nucleares o grandes redes de distribución (agua, energía)
Diseño		<ul style="list-style-type: none"> - Configuración - Instalaciones fabriles - Diseño de productos 	<ul style="list-style-type: none"> - Requisitos de producción 	<ul style="list-style-type: none"> - Diseño de redes de distribución (agua, energía, correos)
Formación	<ul style="list-style-type: none"> - de colaboradores - de personal propio - del servicio exterior 	<ul style="list-style-type: none"> - de colaboradores - de personal propio 	<ul style="list-style-type: none"> - de colaboradores - de personal propio - del servicio exterior 	<ul style="list-style-type: none"> - formación interna en cuestiones jurídicas
Asesoría	<ul style="list-style-type: none"> - de clientes 	<ul style="list-style-type: none"> - de clientes 	<ul style="list-style-type: none"> - de clientes - de servicios especiales 	<ul style="list-style-type: none"> - de clientes

Un sistema experto en monitorización

