

INTRODUCCION A LA INTELIGENCIA ARTIFICIAL

MÓDULO 7- SISTEMAS EXPERTOS

Referencias: Principles of Artificial Intelligence - Nils J. Nilsson

Bases de datos y Sistemas Expertos – R. Frost ...

Sistemas Expertos- Dieter Nebendahl

Sistemas Expertos y Representación del Conocimiento - Carnota, Teszkiewicz

Programming Expert System in OPS5 –Brownston, Farrell, Kant and Martin

Estos sistemas tienen trascendencia porque fueron las primeras aplicaciones de IA que lograron éxitos prácticos y comerciales. Se los puede describir como sistemas basados en conocimiento, que emulan a un experto humano, en la resolución de un problema *significativo*, en un dominio *específico*.

En la figura 1.2 se detallan los elementos involucrados en el desarrollo el uso y el mantenimiento de un sistema basado en conocimiento.

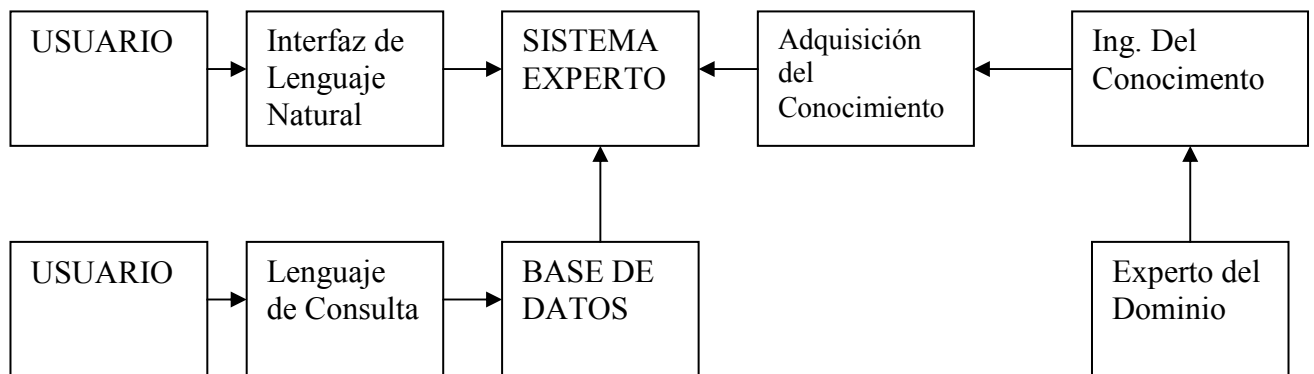


Figura 7.1 Elementos Involucrados en el desarrollo de un Sistema Experto

1.3.1 PRIMEROS DESARROLLOS

1- DENDRAL –1966

Identifica componentes químicos a partir de datos de espectrómetro de masas

Lisp, Plan - Generación - Test

2- MACSYMA –1968

Asistente en el cálculo simbólico

Lisp.

3- MYCIN –1975

Diagnostico de infecciones bacterianas y meningitis / recomendación de terapias

Lisp, Encadenamiento hacia atrás

4- PROSPECTOR –1975

Evaluación geológica de yacimientos minerales.
Lisp. En 1980 descubrió un yacimiento

5- R1 (XCON) – 1978

Configura sistemas de VAX-11/780
OPS5, 800 reglas

6- DELTA –1981

Diagnostica fallas en locomotoras diesel
Lisp / Forth 1200 reglas
REQUISITOS BASICOS

Aplicación de sistemas basados en el conocimiento según sectores

Sector	Banca Seguros	Industria	Comercio Servicios	Encargos Estatales
Aplicación	<ul style="list-style-type: none"> ● Observación de tendencias 	<ul style="list-style-type: none"> ● Control de procesos ● Gobierno de procesos ● Aviso de estados de excepción 	<ul style="list-style-type: none"> ● Observación de tendencias 	<ul style="list-style-type: none"> ● Control de centrales nucleares o de grandes redes (agua, gas)
Diseño		<ul style="list-style-type: none"> ● Configuración ● Instalaciones fabriles ● Diseño de productos 	<ul style="list-style-type: none"> ● Requisitos de Productos 	<ul style="list-style-type: none"> ● de redes de distribución (Correos, energía)
Diagnostico	<ul style="list-style-type: none"> ● Concesión de créditos ● Comprobación de hipotecas ● Análisis de siniestros 	<ul style="list-style-type: none"> ● Motivo de fallo ● Mantenimiento 	<ul style="list-style-type: none"> ● Concesión de créditos ● Cálculo de riesgos 	<ul style="list-style-type: none"> ● Diagnóstico médico (Hospitales) ■ Diagnóstico Técnico ■ (Economía energética)
Planificación	<ul style="list-style-type: none"> ● Análisis de riesgos ● Gestión de valores ■ Planificación de inversiones 	<ul style="list-style-type: none"> ● Funciones lógicas de proyectos ● Proyectos 	<ul style="list-style-type: none"> ■ Análisis de riesgos ■ Análisis de mercado 	<ul style="list-style-type: none"> ● Planificación de inversiones ● Planificación de emergencias ● Planificación de la distribución
Asesoramiento	<ul style="list-style-type: none"> ● Asesoramiento de clientes 	<ul style="list-style-type: none"> ■ Asesoramiento de clientes 	<ul style="list-style-type: none"> ● Asesoramiento de clientes ● Servicios especiales 	<ul style="list-style-type: none"> ● Asesoramiento de clientes
Formación	<ul style="list-style-type: none"> ● Formación de colaboradores ● Formación del servicio exterior 	<ul style="list-style-type: none"> ● Formación de colaboradores 	<ul style="list-style-type: none"> ● Formación de colaboradores ● Formación del servicio exterior 	<ul style="list-style-type: none"> ● Formación interna en cuestiones jurídicas

Tabla 2.1: Campos de aplicación de los Sistemas Expertos

Para usar la técnica de SE el problema a resolver debe cumplir ciertos requisitos que podemos sintetizar en que el desarrollo debe ser POSIBLE, JUSTIFICADO y APROPIADO.

Para que sea Posible debe existir un Experto humano, distintos expertos puedan articular sus métodos y sus soluciones concuerdan. Además no hay componentes físicos en la actividad del experto (test del teléfono), demoran minutos u horas en resolver el problema, tienen un conocimiento preciso y

bien estructurado, no requiere investigación básica, el problema no es demasiado difícil, y los expertos pueden entrenar a novicios

Para que sea Justificado la pericia rescatada por el SE debe ser escasa, se perdería de otra manera, es necesaria en distintos lugares, es necesaria en horarios muy amplios, es necesaria en ambientes hostiles, se requiere que las recomendaciones sean consistentes en el espacio y el tiempo.

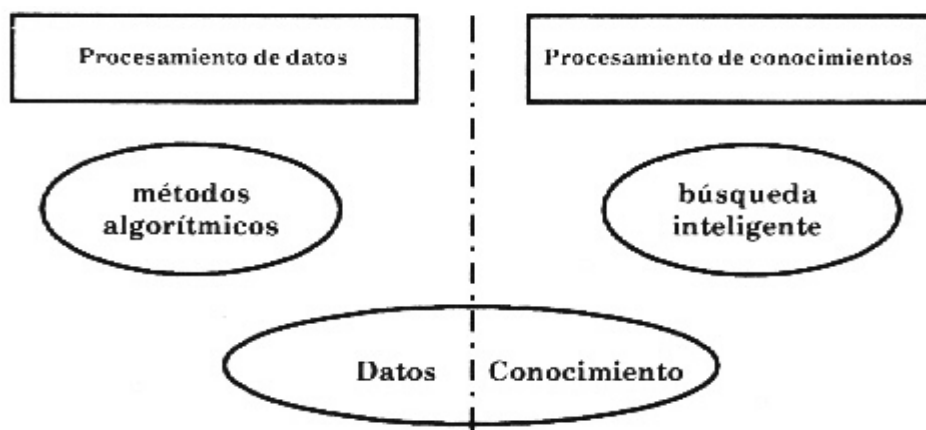
Para que sea Apropiado las operaciones requeridas para la solución del problema debe incluir operaciones simbólicas y no cálculos puros, debe usar reglas heurísticas y no algoritmos, de alcance e importancia práctica manejables pero no demasiado simple. En otras palabras los Sistemas Expertos son útiles cuando por la complejidad del problema, su comportamiento dinámico o la explosión combinatoria no resulta posible o rentable una solución convencional mediante procesamiento de datos.

La lección mas importante aprendida de los primeros fracasos de la IA es que:

“La Inteligencia requiere de Conocimiento”

Para hacer un programa inteligente, dótelo de mucho conocimiento de alta calidad y específico sobre un área restringida.

En la figura 1.3 se puede ver esquemáticamente la diferencia entre el procesamiento de datos y el procesamiento de conocimientos



1.3 Transición del procesamiento de datos al procesamiento de conocimientos

1.3.2- ESTRUCTURA BASICA

La mayoría de los sistemas de IA presenta una separación clara entre componentes comunes de un programa de computadora como son los datos, las operaciones y el control.. En un Sistema Experto (Fig 1.4) estos componentes adoptan el nombre de Base de Datos, Reglas y Máquina de Inferencias.

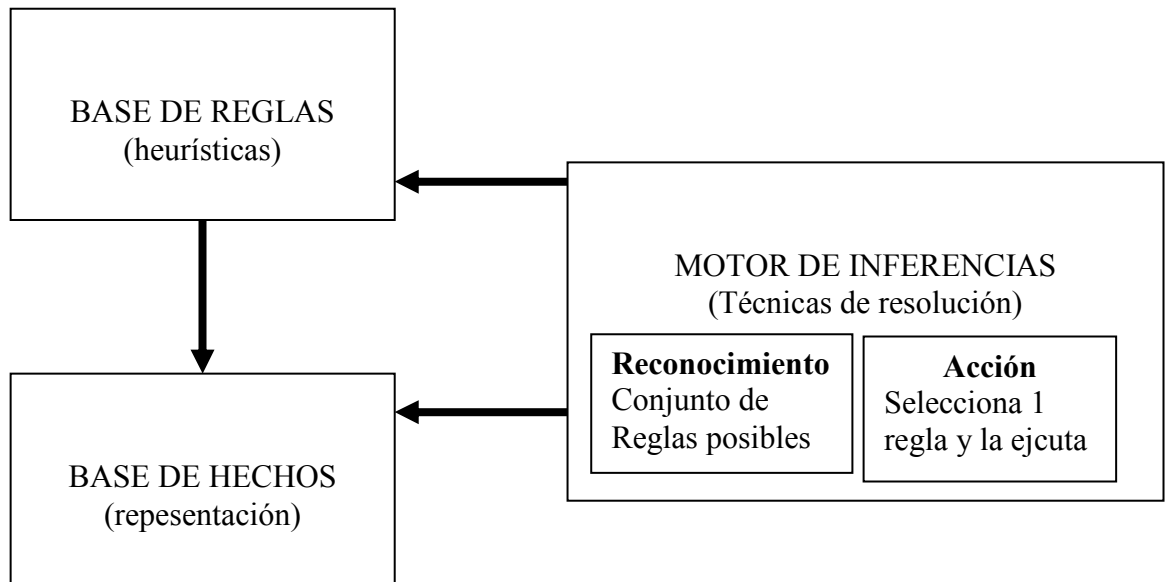


Figura 1.4 Componentes de un Sistema Experto

La base de hechos, o base de datos real es el lugar donde se representa el mundo o dominio, su tamaño y forma dependerá del problema que se esté representando, puede ser una pequeña matriz o un gran archivo indexado (listas, registros, arreglos, parámetros, etc), en la memoria de datos se representa conocimiento de dos tipos: Hechos (características conocidas del Problema) y Metas (características deseables y resultados a obtener, objetivos).

En la base de reglas, están las reglas de producción que indican como manipular los hechos, y transformarlos o producir nuevos hechos hasta encontrar la condición de terminación. Cada regla consta de dos partes, un antecedente y un consecuente, el antecedente es la pre condición que debe cumplir la base de hechos para que la regla sea aplicable, el consecuente es el resultado o acción que produce la aplicación de la regla, esta siempre se manifiesta modificando la base de hechos.

El motor de inferencias es encargado de determinar que reglas son aplicables, y seleccionar dentro de estas cual aplicar en un determinado momento, para iniciar un nuevo ciclo, que se llama de reconocimiento y acción, este ciclo termina cuando la condición de terminación es alcanzada.

Las reglas se comunican solo a través de la base de hechos, una regla no puede invocar a otra regla.

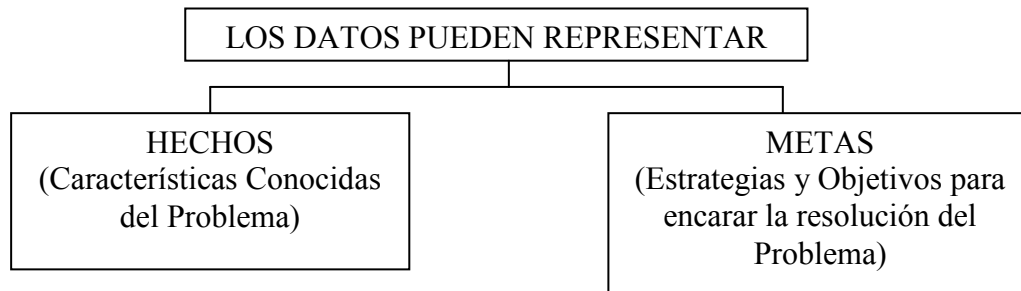
Los sistemas expertos se construyen en forma incremental, su arquitectura permite la modificación de cada una de sus partes en forma independiente.

Mientras que en la programación convencional, el programa es una secuencia ordenada de unidades básicas (las instrucciones), en el modelo de producciones el programa es un conjunto no ordenado de unidades básicas (las producciones o reglas).

MEMORIA DE DATOS

En la Memoria de Datos se almacena el conocimiento obtenido en el proceso de resolución del problema.

Este conocimiento se representa mediante estructuras simbólicas de variada complejidad (listas , registros , arreglos, parámetros, etc.). En la memoria de datos se representa conocimiento de 2 tipos:



PRODUCCIONES (REGLAS)

Una Producción consiste en una estructura de representación de conocimiento.

Esta compuesta de 2 partes:

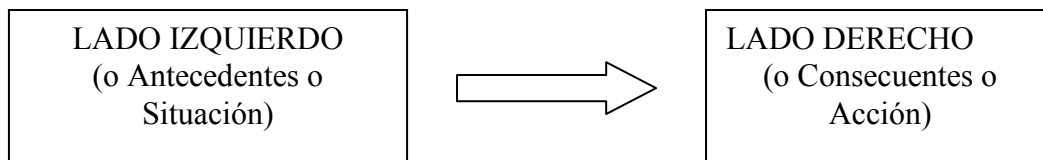
Lado Izquierdo:

Combinación Booleana de Cláusulas.

Una cláusula establece una restricción sobre el valor de un atributo u objeto. (i.e color=verde, precio > 1000, etc.)

Lado Derecho:

Consiste en un procedimiento.



Ejemplo de una producción de un Sistema Experto en diagnóstico sobre el Expert System Environment (IBM).

```
IF      sintoma-inicial-tor      is `Suben y Bajan Solos`
AND    tandem-parado            is `no`
AND    manipulador-ok          is `si`
AND    llave-manual-auto-tor    is `automatico`
AND    problema-sigue-en-manual is `no`
AND    offset-out-modulo-ia     > 10
AND    offset-out-modulo-agc    is in interval > 10 : <= 30
THEM   falla                   is `AGC DUDOSO`
AND    terapia                   is `Ajustar el 1p del IA .....`
```

MAQUINAS DE INFERENCIA “BASICAS”

MAQUINAS DE
INFERENCIA
“BASICAS”

“Encadenamiento
Hacia Adelante”

Analizan las
reglas desde la
CONDICION

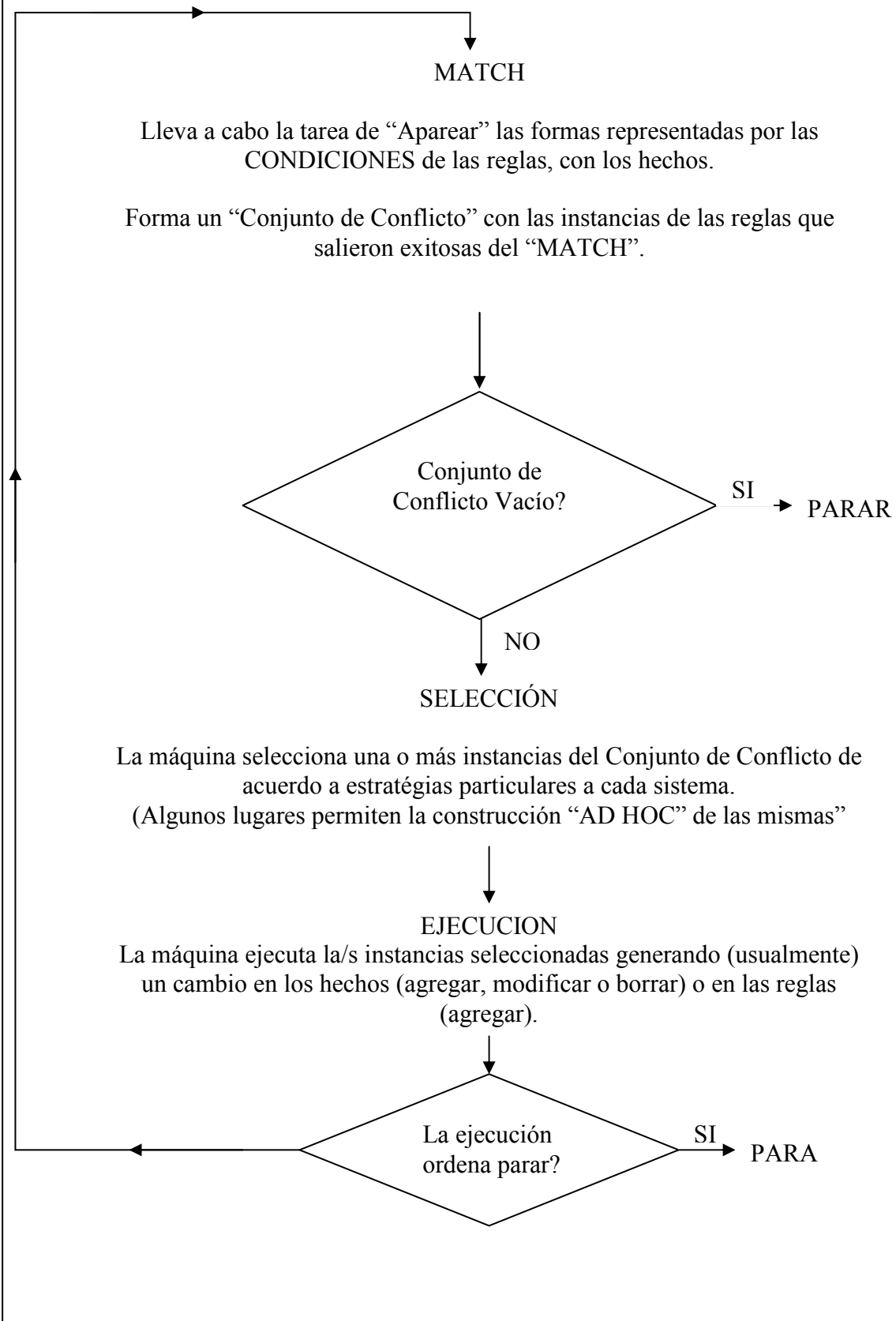
“Encadenamiento
Hacia Atrás”

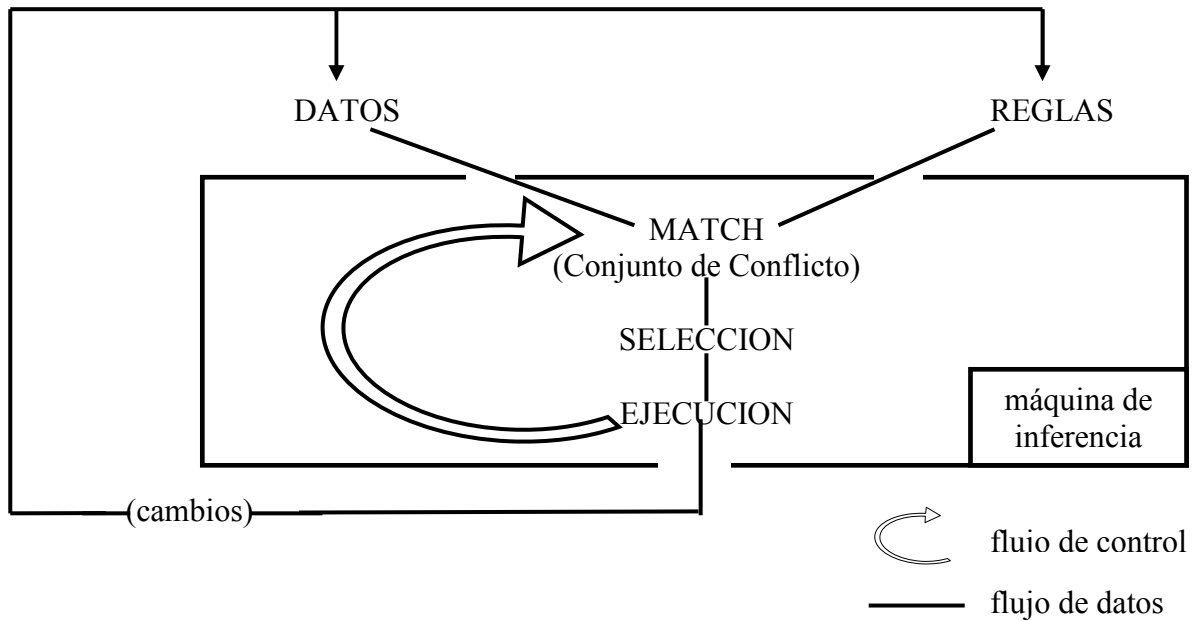
Analizan las
reglas desde la
CONCLUSION

Importante: El significado (semántica) de la regla es INDEPENDIENTE de como se la interpreta.

LA MAQUINA DE INDIFERENCIA “HACIA ADELANTE”

Implementa un ciclo llamado de Reconocimiento-acción.





Veamos como ejemplo un programa elemental compuesto de 3 producciones que determinan como tratar un cliente que no ha pagado su cuenta.

Regla1-bueno

SI no ha sido asignada una categoría de cuenta para el cliente
Y el cliente tiene buenos antecedentes de cuenta

ENTONCES asigne al cliente una categoría 1

Regla2-malo

SI no ha sido asignada una categoría de cuenta para el cliente
Y el cliente tiene malos antecedentes de cuenta

ENTONCES asigne al cliente una categoría 0

Regla3-malo-pero-antiguo

SI no ha sido asignada una categoría de cuenta para el cliente
Y el cliente tiene malos antecedentes de cuenta
Y el cliente tiene más de 10 años de antigüedad

ENTONCES asigne al cliente una categoría 1

Nota: Obsérvese que, cuando es cierta la Regla3-malo-pero-antiguo, siempre es cierta Regla2-malo.

EJEMPLO

Ciclo Reconocimiento-Acción

Match : Crea un match cuando todas las **CONDICIONES** de una regla son satisfechas por algún elemento de la Memoria de Trabajo.

Selección : Elige una regla. La que más condiciones tiene. Si queda más de una, elige al azar.

Ejecución : Modifica la Memoria de Trabajo según indica la parte procedural de la regla.

Reglas

Regla-bueno

SI categoría-cliente = no-hay,
Y antecedentes-cliente = buenos,
⇒ categoría-cliente = prioridad

Regla-malo

SI categoría-cliente = no-hay,
Y antecedentes-cliente = malos,
⇒ categoría-cliente = normal

Regla-malo-antiguo

SI categoría-cliente = no-hay,
Y antecedentes-cliente = malos,
Y antigüedad-cliente > 10,
⇒ categoría-cliente = prioridad

HECHOS

1: cliente=Juan
categoría-cliente = no-hay
antecedentes-cliente=malos
antigüedad-cliente = 22

2: cliente= Pedro
categoría-cliente = no-hay
antecedentes-clientes=malos
antigüedad-cliente = 9

HECHOS

1: cliente=Juan
categoría-cliente = **prioridad**
antecedentes-cliente=malos
antigüedad-cliente = 22

2: cliente= Pedro
categoría-cliente = no-hay
antecedentes-clientes=malos
antigüedad-cliente = 9

HECHOS

1: cliente=Juan
categoría-cliente = **prioridad**
antecedentes-cliente=malos
antigüedad-cliente = 22

2: cliente= Pedro
categoría-cliente = **normal**
antecedentes-clientes=malos
antigüedad-cliente = 9

Ciclo Reconocimiento-Acción N° 1

Match: {[Regla2-malo,1]
[Regla3-malo-pero-antiguo,1]
[Regla2-malo,2]}

Selección :[Regla3-malo-pero-antiguo,1]

Ejecución: (cliente = Juan)
categoría-cliente = prioridad

Ciclo Reconocimiento-Acción N°2

Match: { [Regla2-malo,2]}

Selección :[Regla2-malo,2]

Ejecución: (cliente = Pedro)
categoría-cliente = normal

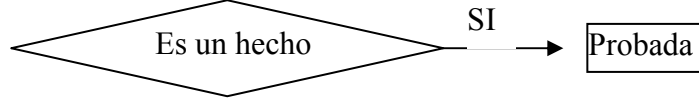
Ciclo Reconocimiento-Acción N°3

Match: { }

Selección: ninguna → para

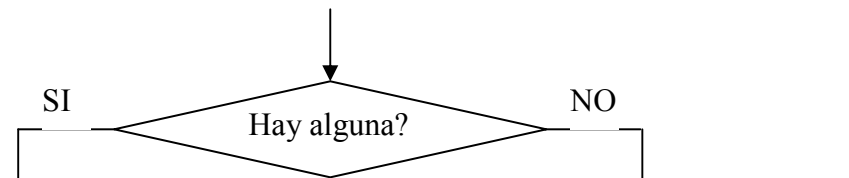
LA MAQUINA DE INDIFERENCIA “HACIA ATRAS”

SELECCIONA UNA POSIBLE SOLUCION AL PROBLEMA (HIPOTESIS)



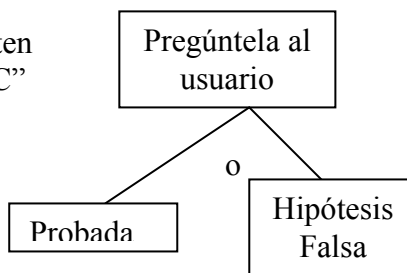
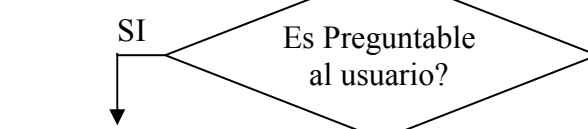
MATCH

Colecta las reglas que concluyen la hipótesis corriente a partir de “mirar” la CONCLUSION de las mismas



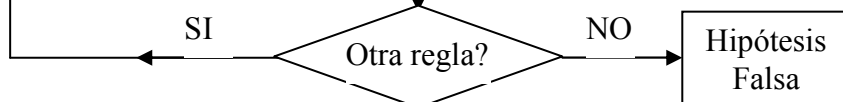
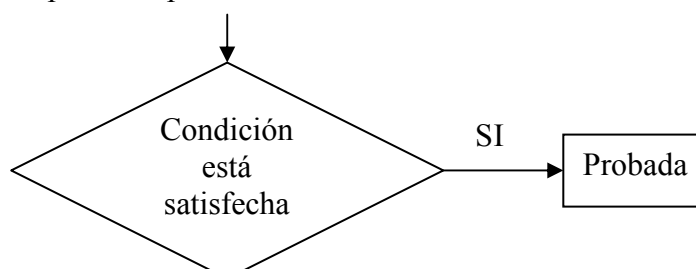
SELECCIÓN
Las ordena de acuerdo a estrategias particulares a cada sistema.
Algunos lenguajes permiten la construcción “ADHOC” de estas estrategias

EJECUCION



Para cada regla, de a una por vez y en el orden establecido realice el siguiente procedimiento hasta probar la hipótesis:

Considere a cada componente de la condición (cláusula) como una hipótesis a probar



EJEMPLO DE ENCADENAMIENTO “HACIA ATRÁS”

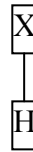
- Reglas:**
- 1) Si A y C \rightarrow E
 - 2) Si F y C \rightarrow H
 - 3) Si B y E \rightarrow H
 - 4) Si B \rightarrow C
 - 5) Si H \rightarrow X

Hechos: A y B

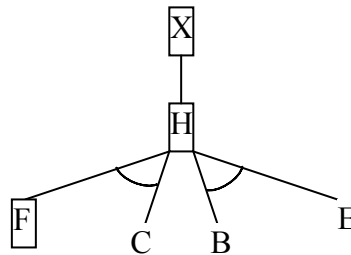
X es una solución al problema
X es una hipótesis

X?

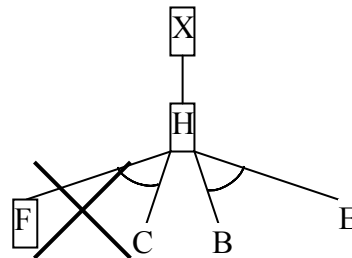
X no es un hecho,
X es solo concluida por 5)
H es nueva hipótesis.



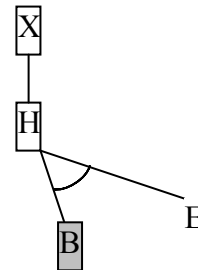
H no es un hecho
H es concluida por 2) y 3)
Supongamos orden (2 3)
Comienza por 2)
F es nueva hipótesis



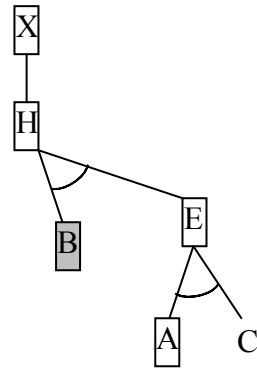
F no es un hecho
F no es concluida por ninguna regla.
F no es preguntable.
F no es cierta.
H no puede ser probada por 2)
Bactracking a 3)
B es nueva hipótesis



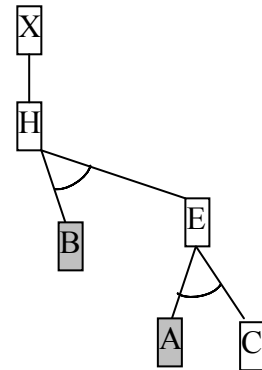
B es un hecho
B es cierta.
E es nueva hipótesis



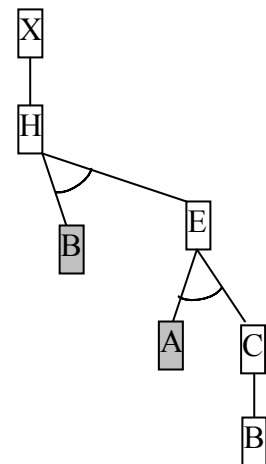
E es concluida por 1)
A es nueva hipótesis



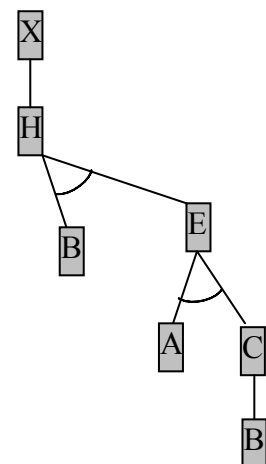
A es un hecho
A es cierta
C es nueva hipótesis



C es concluida por 4)
B es nueva hipótesis



B es un hecho
B es cierta
C es cierta
H es cierta
X es cierta



Observaciones sobre el Control en un Sistema de Producciones

En un Sistema de Producciones el Control se determina mediante las sucesivas re-evaluaciones de los datos y no por estructuras estáticas incorporadas en el lenguaje (como en el modelo procedural).

Se dice entonces que esta forma de computar es “Manejada por los Datos” (Data Driven).

Esto significa que las reglas sólo se pueden comunicar entre ellas por medio de los datos.

No hay tal cosa como una regla llamando explícitamente a otra/s. Para lograr esto es necesario que la regla “invocante” modifique los datos de tal manera que en el próximo ciclo del intérprete la regla/s invocada/s resulte seleccionada del conjunto de conflicto.

El nombre que se le asigna a una regla constituye un identificador único para ayuda del programador, y no puede ser referenciado por otra regla a la manera de un llamado a procedimiento.

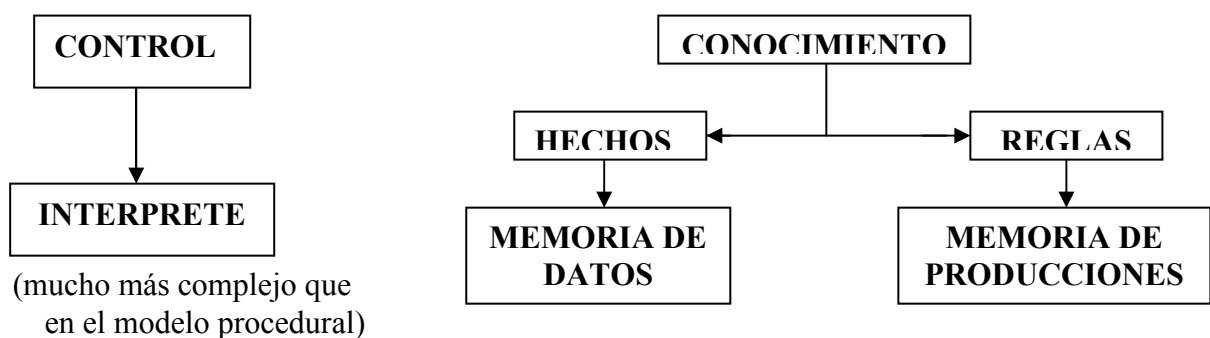
Las reglas no son evaluadas secuencialmente y no resulta siempre posible determinar mediante la inspección del código que regla va a ser ejecutada.

Volviendo al ejemplo de las reglas para determinar el trato que debe recibir un cliente que no ha pagado su cuenta recientemente, hicimos notar que si es cierta la Regla2-malo-pero-antiguo. En este caso la solución es trivial mediante un reordenamiento.

Pero es casos complejos puede no haber un orden único que sea correcto siempre.

De aquí se desprende un concepto que es central al tema de representación y utilización de conocimiento y que constituye el núcleo del poder de un Sistema de Producciones :

EN UN SISTEMA DE PRODUCCIONES EL CONTROL Y EL CONOCIMIENTO SE ENCUENTRAN SEPARADOS



CUANDO USAR ENCADENAMIENTO “HACIA ADELANTE” O “HACIA ATRÁS”

Los Sistemas que razonan “Hacia Adelante” son apropiados cuando hay un número grande de estados finales aceptables (soluciones) y un estado inicial único. Este tipo de problemas se denominan a veces como “Constructivos” (i.e. Configuración, Diseño).

Otra consideración para utilizar encadenamiento “Hacia Adelante” es el hecho de que se disponga de todos los datos necesarios para encontrar una solución al problema al inicio de la consulta. Este es el caso de los Sistemas de Monitoreo o Control, en donde el “input” esta periódicamente siendo actualizado por medio de sensores.

En estos sistemas, las reglas emplean conocimiento “Heurístico” para ir guiando la búsqueda en el espacio del problema.

En general, las reglas en estos sistemas, contienen gran cantidad de conocimiento sobre el control del proceso de resolución del problema. Esta característica les confiere gran flexibilidad a expensas de tener que representar conocimiento que no es propio del problema.

Los Sistemas que razonan “Hacia atrás” son apropiados cuando existe un solo “Estado Final” y muchos posibles “Estados Iniciales”.

Los Problemas de Diagnóstico son apropiados para ser tratados “Hacia Atrás” pues en general hay un solo diagnóstico correcto y una gran cantidad inicial de datos “A priori” potencialmente relevantes como para entrarlos y confiar que sean suficientes para resolver el problema.

En contraste con los sistemas que encadenan hacia adelante, las reglas en un sistema que encadena hacia atrás contienen solo conocimiento sobre el dominio, ya que para controlar el proceso basta con el conocimiento de control depositado en la máquina de inferencia.

En breve: Cuando no se disponen prácticamente de datos iniciales, y estos constituyen un universo “a priori” grande, resulta conveniente “guiar” su adquisición a partir de probar hipótesis, garantizándose una coherencia en las preguntas realizadas al usuario.

CUANDO USAR SISTEMAS DE PRODUCCIONES

Para un hombre que solo posee un martillo, el mundo entero le parece un clavo.
Maschler

Formalmente cualquier proyecto de software puede ser encarado tanto con modelos procedurales como con sistemas de producciones.

Sin embargo la productividad (y a veces hasta la eficacia) del proyecto de software dependerá de cuan bien la/s herramienta/s elegidas se adapten a las características del mismo.

En general, un Sistema de Producciones es apropiado cuando:

“Es deseable que el intérprete determine el orden en que los procedimientos deben ser invocados”.
(Forgy)

Problemas adecuados para usar Sistemas de Producciones:

1) Etapa de Relevamiento y Análisis en problemas grandes y poco estructurados (pobrementemente definidos).

- Es el caso de captar el conocimiento en un Sistema Experto.
- Es el caso de captar la funcionalidad de un Sistema Complejo.

Debido a su intrínseca modularidad, las reglas parecen ser el medio más adecuado para capturar el conocimiento en un sistema que se encuentra en constante cambio.

El propósito de utilizar aquí un Sistema de Producciones es ayudar al analista a entender el problema a través de la construcción rápida de un prototipo.

Salvo que el prototipo sea suficientemente general y eficiente, habrá que recodificar para obtener la aplicación final.

Una vez que el problema es entendido desde el prototipo, puede suceder que sea reducible a algoritmos. En cuyo caso un lenguaje procedural constituye una alternativa mucho más eficiente.

2) Cuando el ámbito de resolución del problema es complejo y las respuestas a él deben ser diversas y basadas en la atención de diversos factores.

Atacar estos casos con modelos procedurales lleva a implementar estructuras de control complejas y oscuras.

En un Sistema de Producciones resulta natural distraer el control y focalizar sobre un subconjunto de reglas, en base a la particular sensibilidad del sistema a los datos del ambiente.

3) Sistemas Expertos

Los Sistemas Expertos son programas escritos para resolver problemas de la misma manera que lo resolvería un experto humano.

Al hablar de “Experto” estamos diciendo que es necesaria una gran “Pericia” para resolver el problema. Tal característica se presenta con frecuencia en dominios Medicinales, Industriales y Militares.

Razones Para programar Sistemas Expertos con Reglas:

- Los Expertos suelen sentirse “Cómodos” expresando su conocimiento en términos de reconocer una situación y actuar en consecuencia.

Las Reglas constituyen un mecanismo relativamente uniforme de representación de conocimiento sin necesidad de recurrir a construcciones complejas de programación.

- Los Sistemas Expertos se construyen incrementalmente a medida que se va consiguiendo adquirir conocimiento del experto. Es necesario entonces contar con un marco de representación y utilización de conocimiento que sea extensible y modificable.

Un Sistema Experto se dice que esta “Basado en Conocimiento” cuando el mismo se encuentra almacenado en forma explícita y separada de otros componentes del sistema.

TIPOS DE SISTEMAS EXPERTOS

CATEGORIA	PROBLEMA
INTERPRETACION	INFERIR DESCRIPCION DE UNA SITUACION A PARTIR DE DATOS DE SENSORES.
PREDICCION	INFERIR CONSECUENCIAS PROBABLES DE UNA DADA SITUACION.
DIAGNOSTICO	INFERIR CAUSA DE UNA FALLA A PARTIR DE SINTOMAS OBSERVABLES.
DISEÑO	CONFIGURAR UN OBJETO SUJETO A UN CONJUNTO DE RESTRICCIONES.
PLANEAMIENTO	DISEÑO DE ACCIONES
MONITOREO	COMPARACION DE OBSERVACIONES CON RESULTADOS ESPERADOS
CONTROL	GOBIERNO GLOBAL DEL COMPORTAMIENTO DE UN SISTEMA.

CUANDO NO USAR LOS SISTEMAS DE PRODUCCIONES

Cuando es posible encontrar algoritmos prácticos para resolver el problema.

Cuando el método de resolución del problema está caracterizado por:

- alta secuencialidad
 - control fijo (aunque complejo)
 - estructuras de datos fija sobre las cuales se ejecutan siempre las mismas acciones.
- ⇒ El modelo de Sistema de Producciones no es apropiado.

PROBLEMAS HIBRIDOS

Un gran número de problemas prácticos deben ser resueltos mediante la participación concurrente de los dos modelos computacionales. Estos problemas se denominan híbridos.

Hay 2 variedades:

- La resolución del problema es esencialmente sensible al contexto pero requiere del apoyo de rutinas externas eficientes.

i.e: Construcción de modelos de estimación de costos para planificar un proyecto de software.
Toma de Decisiones + apoyo de rutinas estadísticas, manejo de Base de Datos, etc.

- La resolución del problema es esencialmente algorítmica pero requiere en varios puntos de toma de decisiones.

i.e: Ciertos Diseños en Ingeniería.

En el primer caso es conveniente que el modelo de sistema de producciones tenga facilidad de comunicación con otros lenguajes o sistemas.

En el segundo caso es conveniente que pueda ser invocado desde otras aplicaciones.

Una solución consiste en integrar en el mismo lenguaje los dos modelos computacionales procedural + producciones.

Caso OPS83 Y KNOWLEDGE TOOL (IBM).

EJERCICIOS DEL MÓDULO SISTEMAS EXPERTOS

1) Sea un Sistema de Producciones constituido por las reglas y el ciclo Reconocimiento-Acción dados en el ejemplo del capítulo.

Sea de la Memoria de Trabajo contiene:

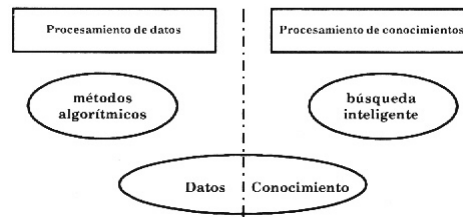
Memoria de Trabajo
1: cliente=Juan categoría-cliente = no-hay antecedentes-cliente=malos antigüedad-cliente = 22
2: cliente= Pedro categoría-cliente = no-hay antecedentes-clientes = buenos antigüedad-cliente = 9
3: cliente=Carlos categoría-cliente = no-hay antecedentes-cliente = malos antigüedad-cliente = 10

- ¿Cuáles son los apareamientos resueltos por Match?
- ¿Qué regla será seleccionada por Selección?
- Hasta que el conjunto de conflicto sea vacío, muestre a la manera de ejemplo, el estado de la Memoria de Trabajo.

2) Dadas las mismas reglas y Memoria de Trabajo de 1), cambie el ciclo Reconocimiento-Acción según, simule, a la manera del ejemplo dado, el comportamiento del Sistema de Producciones.

Ciclo Reconocimiento-Acción	
Match	: Idem anterior.
Selección	: Elige <u>todas</u> las reglas.
Ejecución	: Ejecuta las reglas seleccionadas en orden decreciente de especificidad. En caso de no decidir lo anterior, ejecuta aleatoriamente.

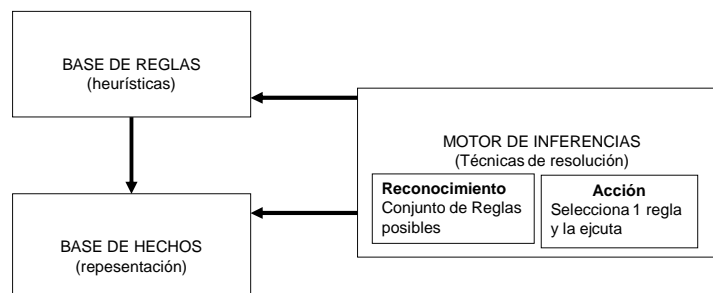
Sistemas Expertos



1.3 Transición del procesamiento de datos al procesamiento de conocimientos

Cambio de paradigma

Sistemas Expertos

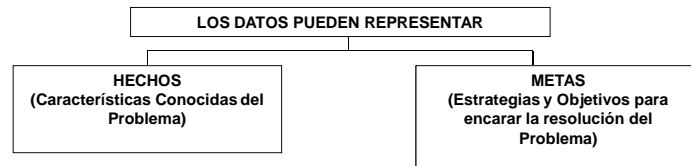


Elementos: Reglas, Hechos, Motor de inferencia

Sistemas Expertos

MEMORIA DE DATOS

En la Memoria de Datos se almacena el conocimiento obtenido en el proceso de resolución del problema. Este conocimiento se representa mediante estructuras simbólicas de variada complejidad (listas, registros, arreglos, parámetros, etc.). En la memoria de datos se representa conocimiento de 2 tipos:

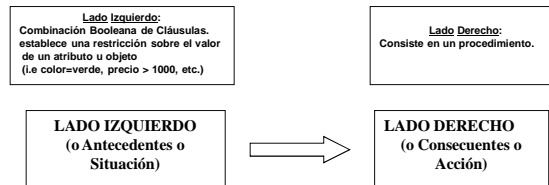


Memoria de Datos

Sistemas Expertos

PRODUCCIONES (REGLAS)

Una Producción consiste en una estructura de representación de conocimiento. Esta compuesta de 2 partes:



Memoria de Reglas

Sistemas Expertos

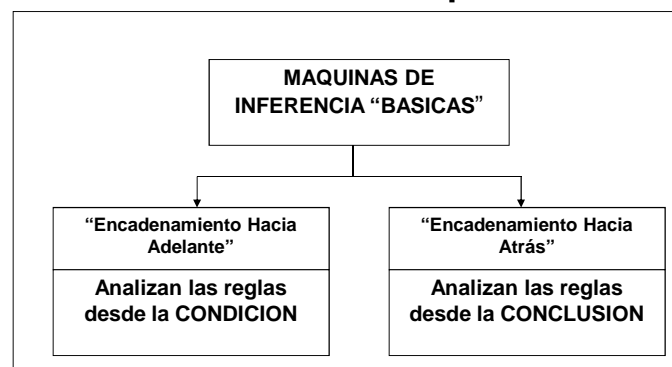
PRODUCCIONES (REGLAS)

Ejemplo de una producción de un Sistema Experto en diagnóstico sobre el Expert System Environment (IBM).

IF	sintoma-inicial-tor		is `Suben y Bajan Solos`
AND	tandem-parado		is `no`
AND	manipulador-ok		is `sí`
AND	llave-manual-auto-tor		is `automatico`
AND	problema-sigue-en-manual	is `no`	
AND	offset-out-modulo-ia		> 10
AND	offset-out-modulo-age	is in interval > 10 : <= 30	
THEM	falla		is `AGC DUDOSO`
AND	terapia		is `Ajustar el Ip del IA
.....			

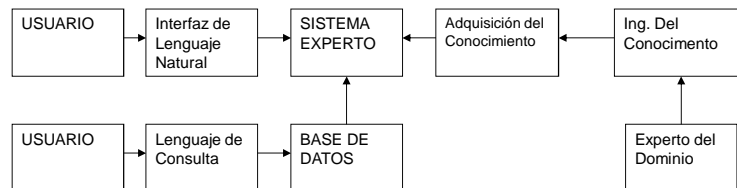
Memoria de Reglas

Sistemas Expertos



Importante: El significado (semántica) de la regla es INDEPENDIENTE de como se la interpreta.

Sistemas Expertos



Elementos Funcionales de un Sistema Experto