

Sistemas Expertos



Alejandro Madruga

SISTEMAS EXPERTOS

Alejandro Madruga

**Texto © 2013 Alejandro Madruga González
Todos los derechos reservados**

Índice

1.- Sistemas de producción

- Introducción a los sistemas expertos (SE)
- Sistema de producción
- Representación a través de reglas

2.-Estrategia de búsqueda

- Definición de búsqueda de solución
- Encadenamiento hacia adelante y encadenamiento hacia atrás
- Búsqueda primero en profundidad vs primero a lo ancho
- Régimen de control: irrevocable o por tentativa
- Estrategia de control heurísticas

3.-Representación del conocimiento

- Redes semánticas
- Marcos (Frames)
- Escenas y guiones

4.-Razonamiento con incertidumbre

- Factor de certeza
- Probabilidad y teorema de Bayes
- Escenas y guiones

5.-Sistemas expertos. Aplicaciones y ejemplos

- Características de los sistemas expertos (SE)
- Paradigma de sistemas expertos: Dendral y Mycin
- Ejemplo de sistema experto. Mycin
- Ejemplo de aplicación en Javascript

1. Sistemas de producción.

Introducción a los Sistemas Expertos (SE).

Durante años la actividad de la Inteligencia Artificial estuvo dedicada a las investigaciones teóricas y al desarrollo de experimentos a través de programas que demostrarían "actitudes inteligentes", con estos programas se pretendía que la máquina jugara ajedrez, demostraría teoremas matemáticos, etc.

No fue hasta los años 70 que surgió un nuevo paradigma en la Inteligencia Artificial "los Sistemas Expertos", cuya función es desarrollar trabajos similares a los que desarrollaría un especialista en un área determinada, la idea no es sustituir a los expertos, sino que estos sistemas sirvan de apoyo a los especialistas en un "dominio" de aplicación específico.

Estos sistemas expertos son en lo esencial sistemas de computación basados en conocimientos cuyos componentes representan un enfoque cualitativo de la programación. Muchas personas pueden creer que un Sistema Experto (SE) es un sistema compuesto por subsistemas y a su vez estos por otros subsistemas hasta llegar a los programas, y que los Sistemas Expertos se miden por la cantidad de programas que contienen. Sin embargo la cantidad no es lo que prima en los SE, si no la calidad del mismo, esta calidad está dada por la separación de las reglas que describen el problema (Base de Conocimientos), del programa de control que es quien selecciona las reglas adecuadas (Motor de inferencias).

Podemos decir que un Sistema Experto es una Base de Conocimientos (BC), una Base de Hechos (BH) y un Motor (o Máquina) de Inferencias (MI). Por otra parte estos sistemas no se miden por la cantidad de instrucciones o programas sino por la cantidad de reglas que hay contenida en su Base de Conocimientos.

Para desarrollar los sistemas expertos primero es necesario abordar un área de interés, dentro de esta área se seleccionan a los expertos, que son los especialistas capaces de resolver los problemas en dicha área. Por ejemplo el área de interés de las empresas de proyectos, son precisamente los proyectos y un especialista podría ser un arquitecto, un ingeniero civil, etc. Ahora bien, casi siempre estos especialistas, son expertos en un dominio específico y es sobre este dominio, donde poseen su mayor experiencia (Dominio de Experticidad), por ejemplo un Ing. civil especializado en cimientos.

Una vez seleccionado al experto o a los expertos y estos estén de acuerdo en dar sus conocimientos, comienza a jugar su papel el "Ingeniero de Conocimientos", que es el encargado de extraerle los conocimientos al experto y darle una representación adecuada, ya sea en forma de reglas u otro tipo de representación, conformando así la base de conocimientos del sistema experto.

Formas de representación de los conocimientos:

- Reglas de producción
- Redes semánticas
- Marcos (Frames).

La forma de representación más usada es por reglas de producción, también llamadas reglas de inferencias. Casi todos los sistemas expertos están basados en este tipo de representación, ahora nos ocuparemos de los sistemas basados en reglas.

Las reglas de producción son del tipo:

SI <premisas> ENTONCES <conclusión> (SI A ENTONCES B).

Donde tanto las premisas como la conclusión, no son más que una cadena de hechos conectados por "Y" o por "O", de forma general sería:

SI <hecho1> Y/O <hecho2> Y/O.....<hechoN> ENTONCES <hecho1> Y/O<hechoN>.

Los hechos son afirmaciones que sirven para representar conceptos, datos, objetos, etc. Y el conjunto de hechos que describen el problema es la base de hechos.

Ejemplo de hechos:

Juan es un estudiante

Juan tiene 8 años

el perro es blanco

a María le gusta el cine

Pedro prefiere la película

la edad de Luis es de 25 años

Pedro tiene un salario de 200 pesos

Una regla es una combinación de hechos que permite representar conocimientos y sacar inferencias de los mismos.

Ejemplo de reglas:

R1: SI <Juan es un estudiante> Y <Juan tiene 8 años> Entonces <Juan estudia en la primaria>.

R2: SI <el perro es blanco> Y <el perro se llama Dinky> ENTONCES <el perro es de Juan>.

R3: SI <a María le gusta la película> Y <Pedro prefiere la pelota> ENTONCES <hacen falta 2 televisores>.

Observe como partiendo de hechos conocidos que describen algún conocimiento se pueden inferir nuevos hechos (nuevos conocimientos), por otra parte la regla #2 (R2), no tiene porque ser totalmente cierta, existe la posibilidad de que el perro sea de Juan, quizás se puede afirmar, si fuéramos a cuantificar esa posibilidad, que el perro pertenece a Juan con una certeza de un 80%, y por último la regla #3 (R3) es dependiente del contexto, ya que aquí se supone que ambos viven juntos y que los programas de TV coinciden.

La Base de Conocimientos (BC).-

Son el conjunto de reglas que permiten representar los conocimientos del dominio de experto donde cada regla aisladamente tiene significado propio. Normalmente los conocimientos son de tipo declarativo por lo cual la BC casi siempre es una descripción de los conocimientos del experto, por lo tanto requiere de algún mecanismo que obtenga las inferencias adecuadas para resolver el problema, alguien que seleccione las reglas y las vaya ejecutando, ese alguien es el motor de inferencias.

El Motor de Inferencias (MI) es un programa de control cuya función es seleccionar las reglas posibles a satisfacer el problema, para ello se vale de ciertas estrategias de control sistemáticas o de estrategias heurísticas.

Estrategias de control sistemático:

- Encadenamiento hacia adelante o hacia atrás.
- Búsqueda en profundidad o a lo ancho.
- Régimen de control irrevocable o por tentativa.

Estas estrategias son de forma sistemática las cuales deben llevar a la solución del problema. Podemos decir que el control sistemático es un programa de control hecho de forma "algorítmica" que aplican una heurística de propósito general cuya función es una exploración exhaustiva y metódica de la base de conocimientos.

Estos mecanismos de control son muy generales y a veces resultan inefficientes ya que siguen una secuencia de búsqueda demasiado rígida, para resolver esto se idearon las estrategias de control heurísticas.

Las estrategias de control heurísticas son programas de control que utilizan una heurística más específica y su función es una selección más restringida orientada por las necesidades del problema. Estas estrategias actúan sobre el control sistemático y en ciertos casos toma el control y dirige la búsqueda hacia ciertos criterios rompiendo así el control sistemático, una vez logrado su objetivo le devuelve el control nuevamente al sistemático.

Estrategias de control heurísticas:

- Orden de las reglas.
- Mayor credibilidad en las reglas.
- Menor número de cláusulas no instanciadas.
- Mayor número de conclusiones en las reglas.

Podemos decir que un sistema experto, es un sistema informático que utiliza técnicas apropiadas para la representación de conocimientos y la manipulación de este, de forma tal que exhiba el comportamiento de un avezado especialista en un determinado dominio del saber.

Sistema de producción

1. Sistema de producción.
2. Base de datos global o base de hechos: depende de la aplicación.
3. Conjunto de reglas de producción o base de conocimientos: del tipo A --> B.
4. Sistema de control o maquina de inferencia: que reglas son aplicada y determina el fin.

Ejemplo:

Base datos global: H1, H2, H3.

Conjunto de reglas:

H1 ---> H5 (Si Precondición entonces Acción).

H2 ---> H6

H5 ---> H7

H7 ---> H3

Sistema de control:

- Escoge que reglas pueden ser aplicadas.
- Suspende el proceso cuando la Base de datos global satisface una condición de terminación.

Observen: como las reglas cazan (unifican) con la precondición y se pasa a la acción la cual se agrega a la base de datos global (base de hechos).

Un sistema de producción proporciona una estructura que facilita la descripción y la ejecución de un proceso de búsqueda y consiste de:

- Un conjunto de facilidades para la definición de reglas.
- Mecanismos para acceder a una o más bases de conocimientos y datos.
- Una estrategia de control que especifica el orden en el que las reglas son procesadas y la forma de resolver los conflictos que pueden aparecer cuando varias reglas coinciden simultáneamente.
- Un mecanismo que se encarga de ir aplicando las reglas.

Selección de las reglas aplicables por el sistema de control:

1. La selección se hace con absoluta arbitrariedad, sin tener en cuenta información alguna sobre el problema (control sistemático).
2. Seleccionar en cada ocasión la regla correcta, haciendo uso de su conocimiento del problema (estrategia heurística).

Procedimiento Básico

1. DATOS (hechos iniciales en la memoria de trabajo).
2. Ciclo: hasta que DATOS satisfagan la condición de terminación.
3. Selecciona alguna REGLA en el conjunto de reglas que pueden ser aplicadas a DATOS.
4. Aplicar la REGLA a DATOS (actualizar la memoria de trabajo agregando o retirando hechos).

Sistemas de producción descomponibles:

Ejemplo:

Base de hechos (DB global):

Estado inicial: C, B, Z

Estado final: M, M, ..., M (se quiere que en el estado final solo hayan M)

Base de conocimientos (Conjunto de reglas)

R1: C ---> D, L

R2: C ---> B, M

R3: B ---> M, M

R4: Z ---> B, B, M

Se quiere partiendo del estado inicial, C, B, Z llegar a un estado final M, ..., M

Debe existir un mecanismo de control (maquina de inferencia) capaz de seleccionar las reglas adecuadas.

1) Se selecciona C de la Base de hechos y se aplica la R2

R2: C ---> B, M Y ahora se selecciona B de las conclusiones de la regla y se aplica R3

R3: B ---> M, M Y nos queda M, M, M (3 M)

2) Se selecciona B de la Base de hechos y se aplica la R3

R3: B ---> M, M Y nos queda M, M (2 M)

3) Se selecciona Z de la base de hechos y se aplica la R4

R4: Z ---> B, B, M Ahora se selecciona la primera B de las conclusiones y se aplica R3

R3: B ---> M, M Se selecciona la segunda B de las conclusiones y se aplica nuevamente R3

R3: B ---> M, M Y nos queda M, M, M, M, N (5 M)

Al agrupan las M en los tres pasos nos queda como resultado final en la base de hechos, 10 M

Representación a través de reglas:

Si H1 y/o H2 y/o H3... entonces H4 y/o H5...

Ejemplos:

Si H1 y H2 entonces H3 y H4.

Si H3 o H4 entonces H5 y H6 o H7.

Si H2 y H7 o H6 entonces H9.

En prolog las reglas son del tipo.

Si H1 y H2 y H3,... entonces Hc

O sea

Hc si H1 y H2 y H3... (Hc :- H1, H2, H3, ...).

Donde los hechos son predicados que deben cumplirse.

R1: Si X>3 y Y=5 entonces Z =3 y A<5.

En caso que X no sea mayor que 3 la regla falla y se buscara otra regla. En esta caso debe cumplirse ambas condiciones (para la X y la Y)..

Si la regla se cumple se ejecuta la acción para Z y A.

R2: Si X<5 o Y >2 entonces Z = 7

Aquí si la primera condición falla (X<5) se pasa la segunda condición. Basta con que se cumpla una de las dos condiciones.

Los hechos son afirmaciones que se refieren a los conceptos (clases y objetos) y a las relaciones (atributos) entre ellos.

Ejemplos de representación de los hechos en diferentes aplicaciones de la IA.

MYCIN

el organismo es un bacteroride

<sujeto> <relación> <concepto>

la coloración del organismo es gram+

<atributo> <sujeto> <valor>

DENDRAL

resonancia espectral con numero atómico igual a numero

<sujeto> <atributo> <valor>

M1 (shell)

Mejor color = tinto

<expresión> < = > <valor>

PROLOG

Gusta (maria, tenis)

<relación> (<concepto>, <concepto>)

EXSYS (shell)

El costo de la computadora es el factor de mayor importancia

<calificador (terminado en verbo)> <valor>

Ejemplos de reglas:

Regla 1: Si tiene espina dorsal

Y tiene cola horizontal

Y tiene un agujero para respirar

Y tiene sangre caliente.

Y los pequeños son alimentandos con leche materna

Y los hijos nacen directamente vivos

Entonces es un cetáceo.

Regla 2: Si es un cetáceo

Y mide mas de 25 pies

Entonces es una ballena

Regla 3: Si es un cetáceo

Y mide aproximadamente 6 pies

Y tiene la aleta tope vertical

Y tiene una nariz roma-pequeña

Y vive cerca de las costas

Entonces es una marsopa

Regla 4: Si es un cetáceo

Y mide aproximadamente 6 pies

Y tiene la aleta tope vertical

Y tiene una nariz en forma de pico

Y vive en el mar

Entonces es un delfín

Regla 5: Si tiene espina dorsal

Y tiene una aleta vertical

Y respira a través de branquias

Entonces es un pez.

Regla 6: Si es un pez

Y tiene la aleta tope triangular

Y tiene la boca debajo de la cabeza

Entonces es un tiburón

Observen que definieron las reglas para los cetáceos y a partir de esta se definió al delfín, la ballena, etc. Igual sucedió para los peces.

2. Estrategia de búsqueda.

Definición de búsqueda de solución.

Introducción:

Uno de los métodos de búsqueda de soluciones para problemas poco estructurados (no existe un algoritmo bien definido), son los espacios de estado

Ejemplo: Imaginemos que estamos en un parque (un Zoológico) que tiene varias entradas y varias salidas y que existen diferentes lugares donde se exhibe o se vende algo (estado).

Un algoritmo lo que haría sería dado una entrada encontrar un camino que me lleve a una salida (sin explorar las demás posibilidades).

Ejemplo de algoritmo, sea el camino: Entrada 1, estado 1, estado 4, Salida 2.

En cambio a través de una exploración no algorítmica yo podría visitar todos los lugares (estados).

Definiciones:

Problemas numéricos: Se busca una representación numérica del problema. Casi siempre aplicando un método de aproximaciones sucesivas.

Características de los algoritmos:

- Finitud: que termine.
- Definibilidad: orden de ejecución y representación de instrucciones.
- Determinismo: que siempre haya solución.

Los problemas numéricos son siempre estructurados, donde los datos y las soluciones están bien definidos, así como los pasos a seguir para llegar a las últimas.

Y pueden ser:

- Algoritmos exactos (solución exacta).
- Algoritmos aproximados (solución basada en iteraciones).

Problemas lógicos: El hallazgo de la solución de un problema mediante un “espacio” de estado y los métodos de búsqueda en este espacio de solución.

Espacio de estado: se trata de definir el problema mediante estados posibles.

Estado: Estructura de datos que representan una fotografía de un momento dado en una etapa de solución.

Es necesario hacer cambios en el espacio de estado para que pase a otro estado (movimiento), el cual también sufrirá nuevas transformaciones. Por lo cual hay que definir una serie de operadores.

Operador: convierte un estado en otro.

Pero hace falta definir el conjunto de acciones para llevar a cabo las transformaciones entre estados.

No todos los estados son admisibles a partir del estado inicial. Debe existir un conjunto de transferencias que lleve al estado final:

$E_0 \sim E_1 \sim E_2 \sim \dots \sim E_n$

Al conjunto de transferencias también se le llaman reglas del tipo:

Antecedente \rightarrow Acción.

Que sirven para realizar todos los estados.

Heurística: es una estrategia que limita drásticamente la búsqueda de soluciones de un problema.

Búsqueda exhaustiva: exploración (normalmente en forma de árbol) de todos los caminos posibles.

Estrategias de control sistemático

Como ya se vio la MI tiene asociada una estrategia de control para poder manejar la BC y esa estrategia de control debe conducir a la solución del problema.

Existen dos tipos de estrategia de control:

- Estrategia de control sistemática.
- Estrategia de control heurística.

Estrategias de tipo sistemático:

1. Encadenamiento hacia delante.
2. Encadenamiento hacia atrás.
3. Búsqueda en profundidad.
4. Búsqueda a lo ancho.
5. Régimen de control irrevocable.
6. Régimen de control por tentativa.

La filosofía de estas metodológicas sistemáticas en los sistemas expertos se fundamenta en la necesidad de obtener una movilidad global durante la invocación de las reglas en el ciclo de evaluación y ejecución.

Según el sentido de enlace entre las reglas:

- Encadenamiento hacia delante (forward-chaining).
- Encadenamiento hacia atrás (backward-chaining).

Según la dirección en la topografía que desarrollan las reglas:

- Búsqueda en profundidad.
- Búsqueda a lo ancho.

Según la estrategia de unificación (pattern-matching).

- Régimen de control irrevocable.
- Régimen de control por tentativas.

Encadenamiento hacia delante y encadenamiento hacia atrás.

Como ya se vio un espacio de estados se puede interpretar como una imagen de la evolución incremental de la base de hechos (la traza de los hechos en la memoria de trabajo), tanto de hechos establecidos como de hechos a establecer, en función de la aplicación de las reglas desencadenables.

En el encadenamiento hacia delante y hacia atrás:

El sentido de la trayectoria en el espacio de estado queda definido por el modo de funcionamiento del motor de inferencias.

Encadenamiento hacia delante:

Los hechos de la Base de datos global (base de hechos), son considerados como disparadores de las reglas, son hechos establecidos. La exploración de las reglas en estas condiciones corresponde a un razonamiento, desde los hechos hacia los objetivos.

Encadenamiento hacia atrás:

Los hechos de la base de hechos son considerados como hechos a establecer (sub-objetivos). La exploración de las reglas corresponde a un razonamiento desde los objetivos a los hechos.

Ejemplo de encadenamiento hacia delante:

Reglas:

R1: si A entonces B.

R2: si B entonces C.

R3: si C entonces Z

Hechos:

H1: A (dato de partida)

H3: Z (objetivo a alcanzar)

Regla	Memoria de trabajo	Observación;
....	(A) (X)	Estado inicial
1 de trabajo (MT)	(A, B) (X)	Se aplica la R1 y la conclusión (B) se incluye en la memoria

- | | | |
|---|------------------|---|
| 2 | (A, B, C) (X) | Se aplica la R2 y se incrementa la MT. |
| 3 | (A, b, C, Z) (Z) | Se aplica la R3 y se alcanzo el objetivo. |

Donde el espacio de estados fue el siguiente:

A \rightarrow B \rightarrow C \rightarrow Z

Para el encadenamiento hacia atrás el espacio de estado se movería al revés:

A \leftarrow B \leftarrow C \leftarrow Z

Búsqueda primero en profundidad vs primero a lo ancho.

Como ya sabemos dado un estado inicial, se van generando estados a través de la aplicación de un conjunto de reglas, las cuales al aplicarse originan un nuevo estado, así sucesivamente hasta alcanzar el estado final.

Habrá en consecuencia un estado inicial y un estado final que serán definidos como:

1. Una base de hechos en la cual aparecerá un hecho que satisface ciertas características.
2. Una base de hechos en la cual habrán desaparecido todos los hechos a establecer.
3. Una base de hechos tal que ninguna regla puede aportar una nueva modificación.

Los puntos 1 y 3 son comúnmente adoptados por los motores de encadenamiento hacia delante, mientras el 2 es adoptado por motores con encadenamiento hacia atrás.

Ejemplo de búsqueda primero en profundidad

Tomemos el ejemplo de las tinas, donde se parte de dos tinas y se quiere lograr que la de 4 galones tenga 2 galones

Vamos a definir algunas reglas de forma general:

Restricciones

1. “Si una tina se llena hasta el tope, no llenar la otra”.
2. “Si una tina se acaba de llenar hasta el tope, no se puede botar”.

Orden de las reglas:

- 1ro. Aplicar reglas para el criterio de solución.
- 2do. Aplicar reglas para llenar las tinas.
- 3ro. Aplicar reglas para vaciar de una para otra.
- 4to. Aplicar reglas para botar el contenido de una de ella.

Criterio de solución

R1: Si <la tina de 4> <tiene> <dos galones> entonces SOLUCION.

R2: Si <la tina de 3> <tiene> <dos galones> y <la tina de 4> <esta vacia> entonces <vaciar el contenido> <para> <la tina de 4>

R3: Si <la tina de 3> <tiene> <dos galones> y <la tina de 4> <no> <esta vacia> entonces <botar el contenido de><la tina de 4>

La búsqueda en profundidad de acuerdo a las condiciones impuestas seria:

<0,0> COMIENZO (lleno una de las tinas) (orden de las reglas)

<0,4> (vacío la de 4 para la de 3) (por estar llena la de 4)

<1,3> (lleno la de 3) (ya que ninguna de las dos están llenas hasta el tope)

<3,3> (vacío la de 3 para la de 4) (por estar llena la de 3 y orden de las reglas)

<2,4> (aplicando criterio de solución regla 3)

<2,0> (aplicando criterio de solución regla 2)

<0,2> SOLUCION (aplicando criterio de solución regla 1)

Régimen de control: irrevocable o por tentativa.

Los regímenes de la estrategia de control:

Las estrategias de control sistemáticas de los sistemas de producción, se completan indicando el modo en que son seleccionadas las reglas de inferencia.

Régimen de control irrevocable

Según el cual, una regla a la vez que se desencadena es aplicada irrevocablemente, sin que sea posible una reconsideración posterior sobre la selección realizada.

Este se usa cuando la estrategia de control posee suficiente conocimiento para seleccionar irrevocablemente la regla apropiada a cada descripción de estado.

Régimen de control por tentativas.

La selección y disparo de una regla implica también, memorizar el punto desencadenamiento con vista a retornar en caso necesario y aplicar otra regla que originalmente se dejó en espera.

Se usa cuando la aplicación de una regla no apropiada puede impedir o retrasar sustancialmente la terminación satisfactoria del proceso. Esta estrategia permite probar una regla, y si después descubre que no era apropiada, volver atrás y probar otra en su lugar.

Metaconocimiento.

El metaconocimiento tiene una función similar a la estrategia heurística, lo que el metaconocimiento está en la BC. Con esta solución se persigue que una MI pueda manejar diferentes heurísticas y puede estar formado por metareglas.

Metareglas: Conocimiento a cerca de sus conocimientos y no es mas que reglas que predicen sobre la aplicación de los hechos.

Este conocimiento se representa mediante un conjunto de metareglas, que contienen conocimientos sobre como razona y cuanto sabe.

Ejemplo:

Si “la infección es un absceso pélvico” y “hay reglas que mencionan en sus premisas a las bacterias” y “hay reglas que mencionan en sus premisas a los bacilos” entonces “Recurrir a las reglas que tratan a las bacterias antes que las que tratan con los bacilos”.

En general:

Si <condicion (regla y/o hecho)> entonces <accion (metaconocimiento)>

Si se cumple H y existe R1 y existe R2 entonces utilizar R1 antes que R2.

El uso de metaconocimiento puede desactivar la acción sistemática del motor de inferencia y es una manera de codificar el conocimiento de forma tal que pueda alterar o producir cambios en el proceso de búsqueda y evaluación de la MI.

También se pueden usar “modelos de reglas o semánticas” que son descripciones abstractas de los subconjuntos de reglas que tienen características comunes. Estos modelos se relacionan con el concepto de marco (frames)

Sea un sistema de control por tentativa, primero en profundidad y de los objetivos a los hechos.

Sean las reglas del tipo Conclusión si se cumplen las premisas (Prolog)

R1: C si A y B

R2: A si X y M.

R3: A si Y.

R4: X si I y J .

R5: Y si J.

R6: Y si K.

R7: B si M.

R8: B si L y M

H1: I.

H2: K.

H3: L.

H3: M.

Estos últimos son los hechos.

Se quiere partiendo del estado objetivo C llegar al estado (hecho) M:

<u>Regla</u>	<u>Base de hechos</u>	<u>Observaciones</u>
—	(C) (M)	Estado inicial
R1 condición B queda pendiente	(C, A) (B, M)	Se aplica la R1 y la condición A pasa a ser el objetivo a demostrar, la
R2	(C, A, X) (B, M)	Se aplica la R2 y X es el nuevo objetivo.
R4 la condición J fracasa.	(C, A,) (B, M)	Se aplica la R4, es la que caza con la X. Se ejecuta la condición I pero Se retorna al Objetivo A
R3	(C, A, Y) (B, M)	Se aplica la R3 y Y es el nuevo objetivo.
R5	(C, A, Y) (B, M)	Se aplica la R5 pero falla la condición J.
R6 cumple la condición A de R1	(C, A, Y, K) (B, M)	Se aplica la R6 y se cumple K. Por lo que se cumple Y y a su vez se
R1 unifique con B.	(C, A, Y, K, B) (M)	Se regresa a R1 a ejecutar la condición B. Se busca una regla que
R7	C, A, Y , K , B, M) (M)	Se aplica R7 y se alcanza el hecho M. Se llego a la solución

Estrategias de control heurísticas

Resolución de conflictos

Se aplica sobre un conjunto de reglas que pueden ser desencadenables (reglas que cazan con las condiciones pero todavía no han pasado a la base de hechos. Con esto se evita su almacenamiento en la base de hechos.

1. Orden de las reglas en la BC.
2. Reglas con mayor coeficiente de credibilidad
3. Reglas con menor número de cláusulas a instanciar.
4. Reglas con mayor número de conclusiones.
5. Reglas menos complejas.
6. Descartar las reglas ya utilizadas.
7. Reglas con mayor numero de condiciones

La heurística muchas veces puede provocar que el problema no tenga solución (producto de las podas del árbol), sin embargo las sistemáticas siempre hayan la solución.

Lo correcto es usar un método heurística para ciertos casos y que luego siga trabajando el control sistemático. O sea la heurística puede interrumpir cuantas veces sea necesaria la acción sistemática de la MI.

A medida que aparecen reglas de control heurística hay que hacerle cambios a la motor de inferencia (MI) lo cual es muy engorroso, para resolver esto se puede.

1. MI con estrategia de control parametrizada.
2. Diseñar el SE en función de la aplicación.
3. Utilizar metaconocimiento.

Ejemplo de máquina de inferencia (MI)

Veamos un ejemplo basado sobre reglas de producción del tipo SI P ENTONCES C donde P (premisas) son los hechos y C sería la conclusión la cual también es un hecho.

La MI tendrá las siguientes estrategias sistemáticas:

- Búsqueda en profundidad.
- Encadenamiento hacia adelante (de los datos al objetivo).
- Control por tentativa.

La estrategia heurística sería por el orden de las reglas:

Sean los hechos de partida A y H (donde A puede ser el hecho <pepe trabaja en computación> y H otro hecho cualquiera), y sea X el objetivo a alcanzar (el cual también se representa en forma de hecho por ejemplo el objetivo podría ser tratar de probar que <pepe no sabe nada de computación>).

Tenemos las siguientes reglas.-

- R1: si A entonces C.
- R2: si A entonces Z.
- R3: si V y Z entonces X.
- R4: si C entonces D.
- R5: si W y Q entonces V.
- R6: si H entonces F.
- R7: si D y F entonces X.

<u>Regla</u>	<u>Base de hechos</u>	<u>Observación</u>
-	(A,H) (X)	Estado inicial.

R1	(A,H,C) (X)	Se aplica la R1 y la conclusión se almacena en la MT
R2	(A,H,C,Z) (X)	Se aplica la R2 e incrementa la MT.
R3	(A,H,C,Z) (X)	Falla, va a buscar otra regla no se incrementa la MT.
R4	(A,H,C,Z,D) (X)	Se aplica la R4.
R5	(A,H,C,Z,D) (X)	Falla.
R6	(A,H,C,Z,D,F) (X)	Se aplica la R6.
R7	(A,H,C,Z,D,F,X) (X)	Éxito, se alcanzó el objetivo X.

Observar como en la memoria de trabajo se van almacenando los hechos que se satisfagan.

3. Representación del conocimiento.

Un sistema experto debe contemplar:

1. Conocimiento asersional (declarativo)
2. Conocimiento operativo (táctico y estratégico).

El asersional es la forma de representar los hechos (premisas).

Ejemplo: <el leon> <es un> <mamifero>

Donde cada expresión representa un hecho, los hechos se pueden combinar y formar reglas (táctica).

El operativo tiene carácter normativo o procedimental .

Ejemplo: si A entonces B. (sistemas de producción).

Redes semánticas

Objetos estructurados

La lógica aunque constituye una buena representación del conocimiento, no aporta mucho cuando tenemos que describir la estructura compleja del mundo y tenemos que escoger un diseño de implantación. Para ello es muy útil agrupar las propiedades de los objetos del mundo en unidades de “descripción”. Esto permite al sistema focalizar su atención en un objeto completo, sin considerar el resto de hechos que conoce, lo cual es importante para evitar la explosión combinatoria, y no solo los objetos son unidades con estructura, sino también los acontecimientos y las secuencias típicas de acontecimientos o escenarios.

Se trata pues de agrupar varias formulas lógicas en estructuras más amplias: objetos estructurados tales como “units”, “frames”, “scripts” y “redes semánticas”.

Los objetos estructurados o esquemas, son organizaciones agrupadas de experiencias típicas adquiridas sobre el mundo y que suponemos operativas en ocasiones futuras. Los marcos (frames) son a menudo utilizados para describir colecciones de propiedades o atributos de un objeto dado, por ejemplo, un árbol. Los guiones (scripts) se usan para describir secuencias comunes de sucesos, como los que ocurren cuando uno va a un restaurante.

Cuando un sistema necesita información accede al objeto estructurado apropiado obteniendo de una vez todos los hechos relevantes sobre él. Así, ciertas operaciones que de otra manera hubieran sido realizadas por aplicaciones explícitas de reglas, son realizadas automáticamente por mecanismos dependientes de la estructura de la representación. Estas representaciones estructuradas del conocimiento, al hacer explícitas informaciones adicionales facilitan el tratamiento de los problemas de búsqueda subyacente.

Una buena representación estructurada para un dominio particular ha de resolver los siguientes problemas.

1. Poder representar todos los tipos de conocimientos del dominio, considerando muchas veces sus expresiones equivalentes en el cálculo de predicados.

2. Poder manipular los objetos estructurados o las redes semánticas de manera que se puedan derivar nuevas estructuras correspondientes a nuevos conocimientos.
3. Poder incorporar a estas estructuras información adicional, que pueda ser usada para aumentar la eficiencia de los mecanismos inferenciales dirigiendo la búsqueda.
4. Facilitar la incorporación de nueva información, directamente por un operador, o bien, como objetivo más difícil de lograr, que sea el programa mismo quien controle la adquisición de conocimiento.

Representaciones procedimentales

Una de las reacciones contra la lógica fue el énfasis en los mecanismos de uso del conocimiento; en contraposición con la lógica, que pone énfasis en el aspecto declarativo o estático del conocimiento. Esta reacción dio lugar a nuevos lenguajes como PLANER, CONNIE, FUZZY, etc. Los cuales unificaban alguno de los mecanismos de resolución de problemas en el marco de un lenguaje de programación.

Las representaciones declarativas ponían énfasis en las estructuras del conocimiento almacenado. En las procedimentales la mayoría de los conocimientos son representados en forma de procedimientos, el conocimiento queda inmerso en el programa. Al comparar ambas representaciones, se ve que las premisas son más económicas y más modulares, en cambio las segundas son más potentes para dar cuenta de las excepciones y aspectos no formales de la teoría. Sin embargo en muchos sistemas de la IA hay un cambio frecuente de representación, de declarativa a procedural y viceversa, lo que indica que parte de la distensión depende más del diseñador que del sistema mismo.

Redes semánticas

Apelan a una representación del conocimiento basado en grafos conceptuales cuyo nodo son objetos o nombres de relación o propiedades y cuyos arcos son relaciones entre objetos, entre conceptos y objetos o entre conceptos, por ejemplo, si un nodo representa el concepto "adjudicación de obra" pueden haber otros nodos representativos de obra, adjudicador, importe, etc. Con cada uno de los cuales hay una relación cuya etiqueta es el nombre del concepto, en general esta es una técnica que permite representar predicados de más de dos plazas, siendo el nombre del arco quien relaciona predicados y términos, el nombre de la plaza. Al propio tiempo las relaciones entre conceptos reflejan jerarquías de clasificación, por ejemplo: "una adjudicación" es una clase de acto administrativo.

NODO_ARCO_NODO

Cada nodo representa algún concepto o objeto y el arco la relación entre conceptos, objetos o ambos.

Una representación de este tipo constituye un marco en el que pueden resolverse problemas de identificación de elementos mediante procedimientos diseñados expresamente para que incorporen "teoría" sobre cada tipo de arco, para obtener las distintas respuestas. Estos procedimientos constituyen realmente el motor inferencial del sistema. El sistema de adquisición de conocimientos consiste en la introducción de nuevos arcos de alguno de los tipos incorporados (entendido) por el motor inferencial.

Ejemplo: Pedro trabaja en el departamento de producción.

trabaja en

PEDRO trabaja en DPTO. DE PRODUCCION

Esto es una unidad funcional en una red semántica.

Pedro es un miembro de una clase de los supervisores

es un

PEDRO es un SUPERVISOR

“es un” se utiliza para designar miembros de alguna clase de objetos ($x \in A$).

“tiene” indica que un concepto es parte de otro.

“es” indica que un concepto es atributo de otro.

Pedro trabaja en el departamento de producción situado en el edificio 1.

Pedro es joven.

Pedro tiene los ojos azules.

Puede ser que en la empresa existan dos Pedro con diferentes funciones.

Ejemplo: Pedro 2 trabaja para Pedro 1 desde septiembre de 1983 hasta junio del 2003.

Representación de un suceso organizado alrededor del verbo.

Para sistemas individuales se han desarrollados conjuntos consistentes e integrados de relaciones verbales. Se denominan “relaciones caso” desarrollada por Fillmore.

Se trata e definir la estructura superficial de las frases mediante un conjunto pequeño y cerrado de relaciones “casos”, que existen dentro de la estructura profunda de la frase, entre sustantivos (o frases substantivadas) y verbos. Un conjunto típico de relaciones puede incluir estas.

1. Agente: iniciador de la acción identificada por el verbo.
2. Objeto: el sustantivo afectado por la acción estado identificado por el verbo.
3. Localización: localización de la acción o estado identificado por el verbo,
4. Dativo: persona afectada por la acción o estado identificado por el verbo.

El suceso específico de trabajar para alguien se ha etiquetado como <trabaja> y toma su identidad, como suceso del nodo “prototipo trabaja” mediante el arco acto.

Los sustantivos de los objetos que participan en el suceso son Pedro 1, Pedro 2, septiembre – 1983 y junio – 2003.

Las estructuras prototipo son muy parecidas a las representaciones “marco”(frames).

Marcos (frames).

Las personas poseen una cantidad de esquemas conceptuales que representan emplazamientos, situaciones y papeles específicos que nos inclinan hacia ciertos detalles en ausencias de conocimientos más precisos. Estos esquemas conceptuales se llaman marcos (frames) los cuales se utilizan como

modelos internos del mundo, o representaciones epistemologías sistemáticas. Los cuales se pueden usar en una amplia variedad de dominios, como los contextos: visual, de resolución de problemas y semánticos.

Los marcos son una estructura del conocimiento que describe un objeto específico y contiene múltiples ranuras (slots) las cuales contienen hechos, características o especificaciones acerca del objeto. Creados en 1975 por Minsky, un marco es similar a una lista de propiedades con valor. La diferencia principal es que los frames pueden encadenarse a otras estructuras.

Los marcos incorporan la representación de entidades complejas cuyas propiedades se incluyen en ranuras como en las redes semánticas pero también incorporan punteros a otros marcos y procedimientos por defecto para obtener conceptos cuando se especifican.

En los marcos los elementos se presentan convencionalmente en la descripción de un objeto o de un suceso. Se agrupan de modo que se puedan procesar o accederse a ellos como una unidad.

Veamos el concepto general supervisor.

Nombre: Supervisor

Especialización en: Empleado.

Nombre: Unidad (primer apellido, nombre)

Edad: Unidad (años)

Dirección: Dirección

Departamento: Rango (producción, administración)

Salario: Salario

Fecha de comienzo: Unidad (mes, año)

Hasta: Unidad (mes, año) (valor por defecto: ahora)

El esqueleto de un marco para el concepto supervisor.

El nombre del marco “especialización en” indica que un marco que en cada instante, representa un concepto que soporta un subconjunto de relaciones con los nombres de los conceptos que actúan como valores de sus campos de información. El campo “especialización en” es, de esta manera utilizado para construir una jerarquía hereditaria similar a la formada por el enlace “es un” en las redes semánticas. Por lo tanto, empleado sería un marco de mayor nivel.

Ejemplo de marcos (frame):

Frame: silla

Pertenece a: muebles

Subconjuntos posibles: pupitre, silla de tijera.

Material: madera, metal, plástico

Cantidad de patas (base): 1, 2, 3, 4.

Otro ejemplo:

Frame: mi silla favorita.

Pertenece a: silla

Subconjuntos posibles: ninguno

Material: madera

Cantidad de patas: 4

Donde cada slots puede ser a su vez un nuevo frame.

A los frames se les puede incluir procedimientos. Por ejemplo, para el caso de la silla se puede incluir el slot envejecimiento.

Frame: silla

.....

Vejez: calcular el tiempo entre ahora y cuando fue construida

Hay tres tipos diferentes de contenido para los campos de información:

1. El contenido es una constante.
2. Es el nombre de otro marco: Dirección, Salario.
3. Las etiquetas: Unidad, rango, etc.

Durante el procesamiento con los sistemas con marco es útil, a menudo, poder imponer restricciones sobre los tipos de objetos que pueden ser valores de un ítem de información. Las etiquetas "unidad" y "rango" nos proporcionan las restricciones estándar: "unidad" especifica que es preciso dar ciertos tipos de objetos, "rango" especifica el conjunto de objetos del cual debemos seleccionar uno. Esta información se puede utilizar por ejemplo, para dirigir una pregunta de un programa que introduce datos en una BD de "empleados".

Finalmente están los valores por defecto que se dan entre paréntesis y que se ligan a los valores de unidad y rango. Las etiquetas de marco "unidad", "rango" y "defecto" son denominados "rasgos" del ítem de información,

Demonios

Juan cogió su cartera y compro los lápices con un peso nuevo.

Aunque no se dice es evidente que el peso salió de la cartera. Aquí se programa un demonio que se despierte al mencionar una cartera y busca la referente a pesos, concluyendo, si la encuentra, que el peso salió de la cartera.

Se tiende a programar demonios para las acciones (verbos) tales como empujar, tirar, deslizar.

Los demonios son una forma de inferencia la cual en lugar de ser llevado a cabo sistemáticamente como las reglas, son sacados "ad hoc" para los propósitos de un problema particular.

Ejemplos de marcos:

Frame: mamífero

Es un : animal

Tiene: pelos

Da: leche.

Frame: carnívoro

Es un: mamífero.

Come: carne.

Frame: Leopardo

Es un: carnívoro.

Color: leonado.

Manchas: oscuras.

Frame: Zumo

Es un: leopardo

Edad: (Procedimiento E)

Observen que el Frame es el nombre del animal, de la especie, etc. Y las ranuras nos permiten acceder a las características de del frame. Para el leopardo se puede a través de la información contenidas en las ranuras saber que es un carnívoro y que tiene color leonado.

La ranura "es un" me permite acceder al frame carnívoro y a su vez a las características de los carnívoros (herencia), por tanto yo puedo saber que el leopardo es un mamífero y que come carne (lo hereda del frame: carnívoro). Lo mismo sucede para el frame mamífero el cual le pasa sus propiedades al frame: carnívoro.

El frame Zumo, que es el nombre del leopardo. La ranura edad, actuara como un demonio en el sentido que cuando se invoque llamará a un procedimiento y este buscará la fecha de nacimiento de Zumo y buscará en la maquina la fecha actual y le restará la anterior y se la asignará a la ranura edad.

Escenas y guiones

Otra de las formas de representación dentro de la línea de los esquemas semánticos son los

guiones.

Sean

E: implicaciones políticas de los acontecimientos

A: implicaciones políticas que involucran actores.

Podemos preguntarnos.

1. ¿Es creíble E? ¿Puede ocurrir o haber ocurrido E?
2. Si Y en caso de E . ¿Que sucedería?
3. Si Y en caso de E ¿Qué haría A?
4. Cuándo E. ¿Qué habría hecho A?
5. ¿Que causo E o que pretende conseguir E?

Oraciones del tipo:

1. A hace X
2. X causa Y
3. A desea Y

Para esto se crean categorías conceptuales generales. Estas categorías se deben combinar para especificar acontecimientos genéricos que a su vez combinan episodios.

Episodio: una sucesión temporal de acontecimientos genéricos potenciales, que permite múltiples ramificaciones.

Ejemplo:

Si una nación A es atacada por una nación B, la primera o puede convertirse en satélite de la segunda o resistir exitosamente y rechazar el ataque.

Los episodios se integran por medio de un guión principal que presenta la ideología global.

Escenarios:

B ataca a A

Causas de B

1. B quiere los recursos de A
2. B quiere dominar al mundo y comenzó atacando a A.

Posibilidades de A;

1. A tiene potencial para resistir.
2. A tiene aliados que lo apoyan.

3. A tiene conflicto internos con una fuerte oposición dentro.

Posibles resultados.

1. B se apodera de los recursos de A.
2. B se apodera de A y se prepara para atacar otro país.
3. B es derrotado por A.
4. B es derrotado por los aliados de A.
5. B se apodera de A y la oposición toma el poder como aliado de B.

Y así cada resultado traerá a su vez nuevos episodios con nuevos escenarios.

A la pregunta: Si Y en caso de E. ¿Qué sucederá?

Se puede contestar.

Si 2 entonces 7 a menos que 4

O sea un guión seria:

Si B quiere dominar al mundo entonces se apoderara primero de A a menos que los otros países ayuden a A a derrotar a B.

Guión: Se trata de interpretar el mundo personal en términos de un conjunto de categorías y expectativas que son peculiares hasta cierto punto y que comparte con las con las expectativas de los demás una semejanza en su estructura básica.

Ejemplo:

Guión

Tema1

Plan 1

Atomo1

Atomo2

.....

Plan 2

.....

Tema2

.....

Atomos:

P: propósitos.

A: acciones

E: estados.

El concepto construir implica un actor que ejecute la construcción empleando medios adecuados, para ello se utiliza el átomo A.

Condiciones entre parejas de átomos.

PE: proposito-estado.

AE: accion-estado. El átomo A esta ligado casualmente con el átomo E.

PA: proposito-accion: El actor del átomo A es el agente (para la acción A) del actor del átomo P.

“Una acción se lleva a cabo a fin de obtener una meta deseada por el patrocinador de la acción”.

La intención en la acción se interpreta en términos de propósitos: El agente no solo ha de creer que A conduce a S, sino que además, debe hacer a para lograr S, esto es desea S y debe ser su razón para hacer A.

El término de propósito aquí se confunde con causa.

Las moléculas mas elaboradas del tipo P-A-E se llaman planes y surgen cuando hay dudas si el actor puede o no alcanzar directamente su propósito.

Ejemplo: A hace X, X causa Y, A desea Y.

Una acción se lleva a cabo a fin de obtener una meta deseada por el patrocinador de la acción.

P-A-S: Hay que interpretar el lazo causal en términos de propósitos.

Los pasos intermedios hacia la meta deben estar especificados en el plan.

Los temas están compuestos por planes interdependientes de dos actores, y es fundamentalmente por referencia a ellos como damos sentido a los mundos sociales y personales que experimenta los miembros de diversas culturas.

Propósitos anónimos de dos actores (planes).

1. Uno o ambos pueden tener un papel en el plan del otro, y en este contexto hay al menos tres modos comúnmente reconocidos de papeles.
 - a) Uno de los dos o ambos pueden actuar como agente del otro, para el plan en su totalidad o para partes del él, y la delegación puede ser temporal o establecida.
 - b) Uno o ambos, pueden estar involucrados en la meta del otro.
 - c) Uno o ambos, pueden ser parte interesada con relación a la ejecución del plan del otro.
2. Uno o ambos pueden tener una actividad valorativa positiva o negativa hacia el plan (en todo o en parte) del otro, y hacia su propio papel en el mismo, si lo tiene.
3. Uno o ambos autores pueden tener la capacidad de facilitar o interferir las acciones del otro, ya

sea en puntos particulares o en todo el plan.

Cada una de estas dimensiones (papel, actitud y capacidad de facilitación) es independiente lógicamente de las demás y puesto que cada una de ellas puede ser instanciada recíprocamente y no es una dirección y como cada una de ellas se puede aplicar a distintas partes del plan del otro en vez de a la totalidad, es obvio que las posibilidades estructurales son muy grandes.

4. Razonamiento con incertidumbre

Factores de certeza

Conocimiento incierto o impreciso

Como ya hemos señalado para el proceso deductivo seria:

Si hay una absoluta credibilidad A y existe la regla $A \rightarrow B$ entonces hay absoluta credibilidad en B.

Y si A es un conjunto de cláusulas del tipo A_1, A_2, \dots, A_n . Pero puede ser que B no sea totalmente cierta.

Cuando el conocimiento no es confiable

A sugiere B con un factor de confianza

H y B sugieren C con otro factor de confianza.

Formas de resolver esto.

1) Aplicando probabilidades.

$P(B/A)$: probabilidad de que se presente B a partir de A (encadenamiento hacia delante).

2) Aplicando conjuntos difusos.

Y tendremos:

$A \rightarrow B \text{ cf}$ (dado que B no está totalmente cierta se da un cf: coeficiente de certeza)

También puede suceder que A no sea totalmente creíble.

$Acf' \rightarrow Bcf$

O sea

$A_1cf_1 \text{ y } A_2cf_2 \text{ y } \dots \rightarrow Bcf$

Cuando el conocimiento no es confiable

A sugiere B con un factor de confianza

H y B sugieren C con otro factor de confianza.

Formas de resolver esto.

1.-Aplicando probabilidades.

$P(B/A)$: probabilidad de que se presente B a partir de A (encadenamiento hacia delante).

2.-Aplicando conjuntos difusos.

El problema de totalizar el valor de la incertidumbre implica el calculo sobre un conjunto de coeficientes relacionados por los operadores lógicos AND, OR y \rightarrow

A tal efecto se consideran los siguientes conceptos:

1. Determinar el valor de la premisa de una regla por efecto de la conjunción de las cláusulas contenida en la misma (AND).
2. Determinar la propagación del valor de la premisa de una regla sobre la conclusión de la misma (\rightarrow).
3. Determinar el valor de la disyunción por efecto de la combinación de varias reglas con la misma conclusión (OR).

VP: valor de la premisa de una regla (cf1 AND cf2 AND cf3 ...)

También: $VP = cf1 * cf2 * cf3 \dots = \min(cf1, cf2, cf3, \dots)$.

VR: valor de la regla por la propagación (VP \rightarrow cf)

También: $VR = VP * cf = \min(VP, cf)$

VC: valor de la conclusión por la combinación de las VR (VR1 OR VR2)

También: $VC = VR1 + VR2 - VR1 * VR2 = \max(VR1, VR2)$.

Ejemplo de cálculo de los factores de certeza

R1: Si A1cf11 y B1cf12 entonces C1cf1

R2: Si A2cf21 y B2cf22 entonces C2cf2

Para hallar el valor de la premisa

R1: $VP1 = \min\{cf11, cf12\}$

R2: $VP2 = \min\{cf21, cf22\}$

Para hallar el valor de la regla (propagación).

R1: $VR1 = VP1 * cf1$

$$R2: VR2 = Vp2 * cf2$$

Para hallar el valor de las conclusiones (combinación)

$$VC = VR1 + VR2 - VR1 * VR2$$

Hay que hallar un valor mínimo (umbral) tal que $VC > Vmin$

Para ello se halla el valor máximo que puede tener ese umbral dándole a todos los Cf de las premisas el valor de uno y se tendrá.

$$R1: Si A1(1) y B1(1) entonces C1cf1$$

$$R2: Si A2(1) y B2(1) entonces C2cf2$$

$$VP1 = 1$$

$$VP2 = 1$$

Por tanto.

$$VR1 = 1 * cf1 = cf1$$

$$VR2 = 1 * cf2 = cf2$$

Donde

$$VC = cf1 + cf2 - cf1 * cf2 = Vi = Vmax$$

Y se tendrá $Vmin < Vmax$

Para cada conclusión del mismo tipo se obtiene:

$$Vmin < Vi$$

Ejemplo de conocimiento incierto:

H1: Te sugiero ir al cine.

H2: Te sugiero quedarte en casa.

H3: Tienes dinero.

H4: Tienes tiempo suficiente

H5: Está lloviendo.

H6: La película esta buena.

H7: La película es de acción.

H8: La película es un drama.

H9: La película es de amor.

H10: La película es una comedia.

R1: Si H10 entonces H6 80%

R2: Si H8 entonces H6 60%

R2: Si H7 entonces H6 70%

R3: Si H6 entonces H1 60%

R4: Si H6 y H3 entonces H1 80%

R5: Si H6 y H3 y H4 entonces H1 95%.

El valor de certeza se da en % entre [0 , 100]

Probabilidades y teorema de Bayes

Introducción a las probabilidades.

Sucesos:

Un suceso puede ser casi cualquier cosa. Llamaremos H al suceso consistente en que cierta hipótesis es verdadera, y E al suceso consistente en la presencia una determinada evidencia que puede ir a favor o en contra de la hipótesis.

Probabilidades:

$P(H)$ es la probabilidad de que H sea cierta.

$P(E)$ es la probabilidad de que E ocurra.

Las probabilidades son números comprendidos entre 0 y 1.

Si la probabilidad es 1 el suceso siempre ocurre.

Si es 0 el suceso no tiene ninguna posibilidad de ocurrir.

$P(\text{no } H) = 1 - P(H)$ es la probabilidad de que H sea falsa.

Dos sucesos son independientes si la probabilidad de que ocurran ambos conjuntamente es igual al producto de las probabilidades de que ocurra cada uno de ellos separadamente.

$P(E1 \text{ y } E2)$ es la probabilidad de que ocurra E1 y E2 conjuntamente. E1 y E2 son independientes si y solo si $P(E1 \text{ y } E2) = P(E1) \cdot P(E2)$.

Si no son dependientes: $P(H \text{ y } E) = P(H / E) * P(E)$ (1)

$P(H / E)$ es la probabilidad de que ocurra H , supuesto que E ha ocurrido ya.

Si H y E son independientes $P(H / E) = P(H)$.

En general $P(H / E) = P(H \text{ y } E) / P(E)$ (2)

Análogamente $P(E / H) = P(E \text{ y } H) / P(H)$

De donde $P(H / E) = P(E : H) * P(H) / P(E)$ Sustituyendo por (1)

O lo que es lo mismo $P(H : E) = P(E : H) * P(H) / P(E)$ (Teorema de Bayes)

Teorema de Bayes

Si <nieva> entonces <hace frío>

Si E entonces H .

Condición	Probabilidad
Nieve y frío	NF
Nieve y no frío	NFN
No nieve y frío	NNF
No nieve y no frío	NNFN

Donde $NF + NFN + NNF + NNFN = 1$.

Analizando la probabilidad

$$P(\text{nieve}) = NF + NFN$$

$$P(\text{frío}) = NF + NNF$$

Queremos saber que probabilidad existe que haya frío cuando cae nieve.

$$P(\text{frío} / \text{nieve}) = P(\text{frío y nieve}) / P(\text{nieve}) = NF / (NF + NFN)$$

De igual forma que probabilidad existe que caiga nieve cuando hay frío.

$$P(\text{nieve} / \text{frío}) = NF / (NF + NNF)$$

Combinando estas igualdades

$$NF / (NF + NFN) = NF / (NF + NNF) * (NF + NNF) / (NF + NFN)$$

$$P(\text{frío/nieve}) = P(\text{nieve/frío}) = P(\text{frío}) / P(\text{nieve}) \text{ (El teorema de Bayes)}$$

Ejemplo de probabilidades

Como ya sabemos:

$P(A)$ es la probabilidad de A.

$P(A \text{ y } B)$ es la probabilidad conjunta de A y B.

$P(A|B)$ es la probabilidad condicional de A, dado B.

Dada la siguiente tabla:

	1. Mañana lloverá	2. Mañana no lloverá
1.-Lluvioso	0.6	0.4
2.-Seco	0.45	0.55

También sabemos que la probabilidad condicional $P(A / B)$ se puede interpretar como Si B entonces A. Sería la probabilidad de que mañana hiciera un tiempo particular dado el tiempo que hace hoy, por ejemplo: Hay una probabilidad 0.6 de que llueva mañana, si hoy llueve.

Primero necesitamos saber cuál es la probabilidad de que llueva mañana. La información de que disponemos es la siguiente.

Sean:

L: lluvioso, S: seco y M: mañana.

Donde $P(M / L)$ se interpreta como:

Si “un día cualquiera está lluvioso” entonces “la probabilidad de que llueva al día siguiente” es 0.6 (si L entonces M 0.6)

Y $P(M / S)$ como:

Si “un día cualquiera está seco” entonces “la probabilidad de que llueva al día siguiente: es 0.45. (si S entonces M 0.45).

Queremos saber si mañana lloverá. Hay que hallar la probabilidad total.

Sea $P(M)$ la probabilidad de que llueva mañana.

Dada la formula

$$P(A \text{ y } B) = P(A / B) * P(B).$$

Tendremos:

$$P(M \text{ y } L) + P(M / L) * P(L).$$

O sea la probabilidad de que llueva hoy y mañana es igual a la probabilidad de que mañana llueva dado que hoy llueve, multiplicado por la probabilidad de que llueva hoy.

Si el día de hoy es lluvioso:

$$P(M \text{ y } L) = 0.6 * P(L).$$

Si el día de hoy es seco.

$$P(M \text{ y } S) = 0.45 * P(S).$$

Pero buscábamos $P(M)$, la probabilidad que mañana llueva. Pues bien, la probabilidad que mañana llueva es, evidentemente, igual a la probabilidad de que mañana llueva y hoy no llueva. Por tanto:

$$\begin{aligned} P(M) &= P(M / L) + P(M / S) = P(M / L) * P(L) + P(M / S) * P(S) \\ &= 0.6 * P(L) + 0.45 * P(S) \end{aligned}$$

Como sabemos, por sentido común, que o bien hoy es un día lluvioso, o bien es un día seco, por lo que $P(L) + P(S) = 1$, es decir, es una cosa u otra con probabilidad 1. Por tanto:

$$\begin{aligned} P(M) &= 0.6 * P(L) + 0.45 * (1 - P(L)) \\ &= 0.6 * P(L) + 0.45 - 0.45 * P(L) \\ &= 0.15 * P(L) + 0.45 \end{aligned}$$

Por consiguiente para responder a la pregunta ¿Cuales la probabilidad de que llueva mañana? Necesitamos saber la probabilidad de hoy sea un día lluvioso.

Esta es, desde luego, una pregunta de fácil respuesta, porque suponemos que la probabilidad de que llueva mañana es la misma probabilidad de que llueva cualquier otro día, lo que significa que es la misma probabilidad de que llueva hoy. Es decir $P(M) = P(L)$, y, por tanto:

$$P(M) = 0.15 * P(M) + 0.45$$

$$0.15 * P(M) = 0.45$$

$$P(M) = 0.45 / 0.85 = 0.53$$

Y obtendremos:

$$P(M \text{ y } L) = 0.6 * P(M). \text{ (ya que } P(M) = P(L) \text{)}$$

$$P(M \text{ y } L) = 0.6 * 0.53 = 0.32$$

$$P(M \text{ y } S) = 0.45 * (1 - P(M)) \text{ (ya que } P(L) = 1 - P(S) \text{ y } P(L) = P(M) \text{).}$$

$$P(M \text{ y } S) = 0.45 - 0.45 (0.53) = 0.21$$

5. Características de los sistemas expertos: ejemplos y aplicaciones.

Característica de los Sistemas Expertos (SE).

Característica de los SE.

- Un SE no debe demorar más de 3 horas en resolver un problema.
- Un SE debe aplicarse a cosas prácticas.
- Debe contar con la colaboración del experto.
- El conocimiento del experto no está en los libros de texto.
- El conocimiento está distribuido pertenece a varias personas.
- Los sistemas expertos tienen pocos niveles de profundidad.
- Para resolver la incompletitud de los conocimientos (información incompleta) se utilizan reglas redundantes. Ejemplo:

Si A y B y C entonces X

Si A y B entonces X

Si A entonces X

Los expertos no razonan a partir de principios (razona a partir de su experiencia), solo los inexpertos razonan a partir de principios.

Consejos a la hora de diseñar un sistema experto

- Separar el generador de inferencias de la base de conocimientos.
- Utilizar una representación tan uniforme como sea posible, siendo las reglas de producción la forma preferida.
- Mantener simple el generador de inferencias (estructura de control).
- Proporcionar algún tipo de facilidad para que el sistema pueda explicar sus conclusiones.
- Favorecer los problemas que requieran el uso de sólidos cuerpos de conocimientos asociativos empíricos sobre aquellos que se puedan resolver utilizando conocimiento causal o matemático.
- Es esencial el compromiso sobre la relación de capacidades humanas a transferir a la máquina.

Etapas de desarrollo de un sistema experto:

1. Identificación: Que se quiere que haga el SE.
2. Conceptualización: Expresar los conocimientos de manera semiformal.

3. Formalización: Diseñar las estructuras para organizar los conocimientos.
4. Implementación: Formalizar las reglas que representan conocimientos.
5. Chequeo: Validación de las reglas

Identificación: se determinan las características del problema, se describen los casos.

Conceptualización: Encontrar los conceptos que representen los conocimientos. La identificación del problema.

Formalización: Durante la formalización es importante que el ingeniero del conocimiento se familiarice con los siguientes tópicos:

- Las diferentes técnicas de representación del conocimiento y las búsquedas heurísticas usadas en los SE.
- Los shells existentes que puedan agilizar el desarrollo del proceso.
- Otros sistemas expertos que puedan resolver similares problemas y se puedan adaptar al problema.

Implantación: Hacer el prototipo: Llevarlo a la computadora y hacer las interfaces con el usuario.

Ingeniería del conocimiento.

1. Representación del conocimiento.
2. Adquisición de los conocimientos.
3. Procesamiento del conocimiento

Hasta ahora se ha hecho referencia al núcleo del sistema experto (BC, BH y MI) en los cuales se representan los conocimientos del experto en forma de reglas, los hechos o datos globales de partida y la forma de inferir esos conocimientos, pero no se ha dicho como se comunica el sistema con el usuario y con el experto.

Módulo de interacción con el usuario (interfaces con el usuario):

Facilita el diálogo con el usuario, permite hacerle preguntas al sistema e incluso obtener conocimientos análogos a los del experto. Estas interfaces podrían ser programas de entrada/salida de forma dialogada y las explicaciones pueden ser obtenidas de la memoria de trabajo donde se almacenan los pasos para llegar a la solución.

Módulo de ayuda para la adquisición de conocimientos (interfaces con el experto):

Permite al experto consultar los conocimientos almacenados en la base de conocimientos y en muchos casos dar la posibilidad de incluirle nuevos conocimientos. Su objetivo es que el experto pueda introducir directamente sus conocimientos en la máquina sin necesidad de ver al ingeniero del conocimiento.

Paradigma de Sistemas Expertos: DENDRAL Y MYCIN.

Veremos algunos ejemplos que han sido paradigmas de SE dentro de la IA.

DRENDAL: Reducción de grandes espacios de búsqueda mediante factorización (heurística).

Ayudar a los químicos a inferir la estructura de los componentes químicos a partir de los datos del espectro de masa.

Se partió de la técnica de generar y probar la cual depende de la forma en la que se desarrolla el espacio de estados y de predecir que ramas conducirán a la solución, hasta que profundidad se debe explorar una rama sin que se pierda una posible solución.

Fue el pionero en el cambio de enfoque hacia las estructuras de control dependientes del dominio.

MYCIN. El manejo de evidencia incierta.

Sistema para el diagnóstico de infecciones bacterianas.

Se basa en conocimientos indefinidos y datos imprecisos en vez de los manejos del espacio del problema.

DENDRAL

El paradigma de búsqueda en espacios de estados. El cual se basa en una búsqueda a través de una red de nodos, donde cada uno representa un posible estado del problema, donde casi siempre una búsqueda exhaustiva de todo el espacio es imposible (explosión combinatoria). Un método es controlar la búsqueda en el espacio de estado basándose en la relación que hay entre los estados.

Una técnica muy conocida es la de “generar y probar”. Comenzando por algún estado inicial, se utiliza un generador para producir un conjunto de estados descendentes, se aplica entonces a ese conjunto una serie de pruebas de validez para reducirlo a un tamaño más manejable. Estas pruebas adoptan normalmente la forma de restricciones, también es posible incorporar estas restricciones al generador reduciendo el número de estados descendentes a probar. El método “generar y probar” solo funciona, si cuando un estado se declara invalido, todos sus estados descendentes lo son.

DENDRAL, fue el primer sistema basado en conocimientos que utilizó el método de “generar y probar” para las conexiones de las estructuras de compuestos de química orgánica. El proyecto comenzó en 1965 con el objetivo de proporcionar un soporte en la computadora a los químicos, profesionales que no eran expertos en determinadas técnicas analíticas y tuvo tal éxito que inspiró a la creación del área de los sistemas expertos en la IA.

La tarea de la aplicación.

DENDRAL: es un sistema diseñado para ayudar a los químicos a determinar la estructura de algún compuesto desconocido, con especial énfasis en el uso de los datos provenientes de un espectrómetro de masas.

El programa de planificación de DENDRAL

En la práctica hizo falta un programa de planificación para producir las restricciones a la hora de generar. El ciclo completo es PLANIFICAR – GENERAR – PROBAR.

Con el planificador se evita la generación superflua y simplifica la fase de prueba.

Semejante a MYCIN, DENDRAL no pretende simular los procesos interpretativos que utilizan los químicos. No obstante si pretende complementar los métodos de estos.

Con la planificaron se infiere una lista con las estructuras necesarias(1) y con las prohibidas (2) las cuales ayudaran, mas tarde a limitar las estructuras proporcionadas por el generador .

(1) Lista correcta (2) Lista incorrecta

Ejemplo de regla:

R1: Si hay dos picos de las unidades de masa X1 y X2, de modo que

$X_1 + X_2 = M + 28$

$X_1 - 28$ es un pico alto

$X_2 - 28$ es un pico alto

Al menos, o X_1 es alto o X_2 es alto

Entonces CETONA

R2: Si se satisface las condiciones para ser una CETONA

43 es un pico alto

58 es un pico alto

$M - 43$ es un pico bajo

$M - 15$ es bajo o, posiblemente 0

Entonces METIL-CETONA.

Encadenando hacia delante si se aplica la regla R1 el PLANIFICADOR puede considerar aquellas reglas que tenga en sus condiciones el grupo CETONA.

R3: Si es CETONA y Entonces ETIL-CETONA.

O sea encadena con las reglas R2 y R3 de acuerdo con su espectro de masa y la correcta la incluye en la lista correcta y la otra en la lista incorrecta. Lo cual permite al planificador determinar la posición de un grupo funcional y registrar la información adicional para que sea utilizada más tarde por el generador.

Generador de estructura: es un programa que genera el espacio de búsqueda con las posibles estructuras químicas.

MYCIN

Este sistema opera sobre una BC bien organizada sobre las enfermedades infecciosas, donde el conocimiento es inexacto por lo que el punto central son las técnicas para expresar medidas de opinión, llamadas factores de certeza.

La estructura de control es simple y está basada en un sistema de producción.

1. Una colección de hechos.
2. Un conjunto de reglas de producción.

3. Un generador de inferencias con encadenamiento hacia delante o hacia a tras (o ambos), junto a una estructura de conocimiento que capacita a la estructura de control para decidir que reglas, de todas las posibles, debe tomar parte en el mecanismo de inferencia.
4. Un mecanismo que realice inferencias, a partir de evidencias inciertas o incompletas..

En MYCIN las agrupaciones de reglas dentro de un determinado contexto se definen en tiempo de generación. El generador de inferencias proporciona una búsqueda dirigida por objetivos o por datos.

La existencia de una estructura de control dependiente del dominio, la hace específica. Por eso se han extraído de MYCIN las partes independientes del dominio para formar el MYCIN “vacio” o “hueco”, EMYCIN. El cual se ha empleado en otros dominios tales como PUFF (cuidado respiratorio intensivo), SACON (cálculo de estructuras de ingeniería), GRAVIDA (consejos para el embarazo), CLOT (desórdenes de la sangre) y VM (manejo de la ventilación).

Hechos.

<contexto><parametro><valor>

El contexto es alguna entidad del mundo real. Ejemplo: un paciente.

Un parámetro es un atributo del contexto. Ejemplo: la edad.

El valor es una particularización del parámetro. Ejemplo: 25 años.

A cada terna se le asocia un valor de certeza (FC).

FC toma valores en el intervalo [-1, 1] donde -1 es la negación, 0 la ignorancia absoluta y 1 total certidumbre.

Ejemplo:

Organismo 1 – Identidad pseudomonas .8 (La identidad del organismo 1 es, con un factor de certeza 0.8, pseudomonas).

Reglas de producción.

Si premisa entonces acción (FC).

En MYCIN las reglas se definen para poder encadenar hacia delante o hacia atrás. Las reglas se utilizan repetidamente utilizando el conocimiento dependiente del dominio para decidir que reglas se deben considerar, hasta que se dé un contenido a los hechos. En algunos casos, las reglas darán contenido a los hechos mediante preguntas al médico.

Ejemplo de reglas:

Si <no se sabe si el organismo fue capaz de desarrollarse en presencia del aire (aerobicamente)>

Y <el lugar de cultivo es la sangre>

O <el laboratorio ha intentado que el organismo se desarrolle sin la presencia del aire (anaerobicamente)>

Y <el organismo fue capaz de desarrollarse anaerobicamente>

Entonces <hay evidencias de que el organismo es aerobio (.5) O anaerobio (.2).

Las premisas serán verdad si todos los hechos son verdad.

Factor de certeza.

En general:

Si < contexto-parametro-valor (FC1) > y/o < contexto-parametro-valor (FC2) y/o.....

Entonces <contexto-parametro-valor (FC) .

$\text{Min}(\text{FC1}, \text{FC2}, \dots) * \text{FC} = \text{RC}$

Si la regla no venía arrastrando ningún valor de certeza entonces RC se colocara como su factor de certeza (FC) , si ya tenía con un valor IC. Se calculara el FC actual.

$\text{FC} = \text{IC} + |\text{RC}| (1 - |\text{IC}|)$ $\text{RC}, \text{IC} > 0$

$\text{FC} = -(|\text{IC}| + |\text{RC}| (1 - |\text{IC}|))$ $\text{RC}, \text{IC} < 0$

$\text{Fc} = (\text{IC} + \text{RC}) / [1 - \min(|\text{IC}|, |\text{RC}|)]$ $\text{RC} \cdot \text{IC} < 0$

La combinación de -1 y 1 se define como 1.

Este método del factor de certeza permite que el sistema se pueda enfrentar al razonamiento de juicios.

Estructura de control dependiente del dominio

El orden de las reglas no es significativo se aplican todas las reglas relevantes hasta que una determinada consigue un objetivo con certeza, en cuyo caso el resto de las reglas se ignora. La relevancia de una regla se define por la estructura del control dependiente del dominio, la cual se organiza como un árbol de contexto donde se organizan los hechos de la BC en una jerarquía, y se utiliza para definir el flujo de la consulta. Las reglas se agrupan en conjuntos de reglas definidas por el conjunto de tipos de contexto de aplicación de las mismas (una regla siempre se refiere a un determinado tipo de contexto).

Obtención de información a partir del usuario.

Las preguntas se hacen o bien cuando fallan las reglas o bien cuando la información ha de provenir del usuario (ejemplo: datos de laboratorio). Se puede reelaborar una pregunta y permite diferentes conjunto de preguntas para usuarios con experiencia y sin ella, y se basa en la propiedad de “preguntar – primero”, el sistema siempre procura primero obtener el valor de un parámetro a partir del usuario.

El diseñador debe proporcionar un medio aceptable de comunicación con el usuario, MYCIN no fue totalmente satisfactorio en ese aspecto, lo cual se resolvió en ONCOCIN.

Facilidades de explicación

MYCIN como muchos otros SE tienen la facilidad de explicar y justificar su funcionamiento, lo cual es útil para funciones educativas. En cualquier momento el usuario puede preguntarle “como” o “porque”, tomo tal decisión. Esto es a través de reglas

Ejemplo:

Si “está nublado” entonces “va a llover”

“Si “va a llover” entonces “coger la capa”

Ahora se le puede pregunta al sistema:

Porque "coger la capa"?

Porque "está nublado" y "va a llover".

Los Se son el resultado de una serie de consultas entre los constructores del sistema y expertos en el dominio de conocimiento de interés. Este proceso es laborioso y propenso a errores. Los expertos saben normalmente, más de lo que piensan y no siempre pueden expresar fácilmente sus procesos de razonamiento.

Ejemplo de una aplicación: Termi.

TERMI: Sistema para la asistencia técnica en la selección de soluciones tecnológicas para tratamiento térmico.

Desarrollado con técnicas de Inteligencia Artificial para prestarles asistencia, técnica tanto al inversionista como al tecnólogo, a la hora de elegir los modelos de hornos más adecuados para tratamiento térmico.

Ahora veremos cómo partiendo de las ideas de DENDRAL (Sistema para resolver la estructura de los compuestos en química), se desarrolló una aplicación para elegir los modelos de hornos para el tratamiento térmico. Observen como áreas tan disímiles son abordadas con técnicas muy parecidas.

Historia de Termi

Cuando se escoge una solución tecnológica, siempre queda la duda si esa es la mejor solución o si existen otras mejores, las cuales no se tuvieron en cuenta por falta de tiempo.

La solución óptima en muchos casos es sólo un ideal, ya que existen muchas condicionales, que van desde problemas con las materiales, disponibilidad de equipos, características ambientales y topográficas; hasta los gustos de alguien. Por otra parte lo que hoy puede ser una buena solución, mañana puede dejar de serlo al variar los requerimientos y no ser aceptable. Por tanto, hacer un programa que halle una buena solución y de flexibilidad, es una tarea difícil con los requisitos de la programación convencional.

Encontrar una buena solución, significa realizar alguna búsqueda en un espacio de posibles soluciones. Es algo así como explorar un laberinto, donde las soluciones son los caminos que conducen a alguna de las salidas -pueden haber varias salidas. Lo óptimo pudiera ser el camino más corto a una de las salidas. ¿A cuál salida? Eso dependería de las reglas que se apliquen para moverse dentro del laberinto. La habilidad para explorar los caminos más prometedores y desechar los caminos poco fértiles, los que no conducen a ninguna parte, o sea, la heurística.

Uno de los métodos para recorrer el espacio de soluciones o de estados, es "generar y probar". Se genera una posible solución y luego se prueba a ver si es aceptable. El primer sistema desarrollado con estas técnicas de inteligencia artificial (como ya vimos) fue DENDRAL, el cual se utilizó para deducir las estructuras químicas correctas, en los años 70. Esta misma idea de generar y probar, fue aplicada por nosotros en TERMI: Sistema para la asistencia técnica en la selección de soluciones tecnológicas generales para tratamiento térmico, desarrollado en el lenguaje PROLOG.

Con este sistema se resuelve la selección de los modelos de hornos necesarios para ejecutar la producción y realización del balance carga-capacidad: tratamientos térmicos de piezas fundidas, forjadas o soldadas, empleando concretamente hornos de cámara de carro o solera móvil (calentado eléctricamente

o por combustible); aunque también se puede incluir otros tipos de hornos en la base de hechos. Se han considerado los siguientes tipos de tratamientos térmicos: recocido, normalizado, recocido suscrito, temple y revenido.

La tarea de escoger una sola solución, no era fácil, ya que cada solución tenía sus pros y sus contras. Es lo que se conoce como soluciones de compromiso, las cuales son mejores en unos aspectos y peores en otro. Por ello decidimos que en lugar de dar una solución única, dar un racimo de soluciones todas buenas, y que el usuario escogiera después la más racional. Se le deja la decisión al humano, la máquina sólo propone.

El sistema TERMI consta de cinco módulos: módulo de datos, módulo de entrada, módulo de solución, módulo de salida y módulo de enlace. Cada módulo es independiente del otro. El corazón del sistema es el módulo de soluciones, que es donde se generan y prueban las soluciones tecnológicas, tales como escoger los modelos de hornos, los indicadores de producción, tipos de tratamientos, turnos de trabajo, etc., y proponer una posible configuración; en otras palabras, eleva su propuesta del equipamiento con sus características y posibilidades al probador, esté último es el encargado de verificar si cumple con los requerimientos, y con las restricciones impuestas por el experto.

Un inconveniente que se presentó, era cuando TERMI no encontraba ninguna solución; o sea, ningún horno cumpla con las restricciones. En ese caso el sistema terminaba su ejecución y no decía "nada". Por lo cual se hizo necesario incluirle explicaciones para saber que habla pasado. ¿Qué fue lo que falló? Como es lógico nadie aceptaría un programa que no diese solución alguna, para algunos casos, aunque fuese cierto que la solución no existía, ya que ninguna cumplía con las exigencias; pero..., ¿Con cuáles de las exigencias no cumplía? Y eso fue lo que se hizo, incluirle explicaciones relacionadas con el espacio de Búsqueda; para en caso de algún fracaso, explicar que fue lo que Pasó, qué requerimientos no se cumplieron.

Además de generar (nosotros le llamamos selección) y probar (revisión) se le agrega otra fase, la de análisis. Antes de seleccionar alguna posible solución, se debe realizar un análisis. Todos sabemos por experiencia, que muchos problemas se pueden atacar de forma directa, o es fácil determinar su imposibilidad de abordarlos. Tomemos como ejemplo que se quieren guardar objetos en una caja, los objetos demasiado grandes, que a simple vista se ve que no caben; nadie se tomaría el trabajo de cargarlos, llevarlos hasta la caja y probar si cabe o no; ese esfuerzo inútil, uno siempre se lo ahorra, ya que a simple vista es evidente que no cabe en la caja. A través del análisis se trata de evitar la generación o selección de soluciones que de entrada se puede saber que son incorrectas con una simple Inspección; por tanto, su función es detectar esos casos y almacenar las explicaciones pertinentes. También realiza la restricción de la base de hechos o de datos: en caso que usted desee hornos rusos, haría la selección solamente sobre los hornos rusos, evitando la búsqueda innecesaria de hornos de otros países.

Otra posibilidad, es cuando los hornos no dan la capacidad de carga esperada, lo cual obligaría una revisión de las exigencias impuestas. Teniendo en cuenta esto, el sistema le propone al tecnólogo las soluciones más próximas tanto por defecto como por exceso a la capacidad de producción deseada, de esta forma nunca se va con las manos vacías.

A la hora de introducir los datos, a través del módulo de entrada, se trato de que esta fuese lo más flexible posible, ya que se pretendía que el sistema pudiese usarse lo mismo por especialistas (tecnólogos) para realizar sus proyectos tecnológicos o para realizar la asistencia técnica a inversionistas.

Los datos pueden ser suministrados a la computadora de tres formas, como un valor en sí: real, entero, etc. Como un intervalo de valores; ejemplo, 60-90, tomar todos los valores enteros comprendidos entre 60 y 90. O decir simplemente "no sé" y no dar dato alguno, en este caso el sistema probaría con los rangos de valores dados por el experto. Esto trae como consecuencia que mientras más "no sé" del usuario, menos criterio tendría el sistema sobre lo que este desea, y al final daría un conjunto de soluciones mucho mayor.

Si el usuario tiene su problema bien definido, el conjunto de soluciones final serán pequeñas. Normalmente a los tecnólogos les da de dos a cuatro soluciones; Sin embargo, a alguien no conocedor de la materia puede darle hasta más de cuarenta soluciones, lo cual es lógico, ya que el sistema tiene que escoger entre mas variantes y las decisiones corren por él. Aunque este no es el objetivo, se quiere que el tecnólogo sea el que decida, da la posibilidad a un no experto de aprender con TERMI: analizando las soluciones que el sistema da para diferentes casos; lo cual permite simular en pocos minutos posibles variantes de soluciones las cuales llevarla a obtenerlas a mano.

Como método de trabajo nos dimos a la tarea de resolver un problema específico, para ello nos dedicamos a entender el dominio de aplicación o de experticia, y luego en función de los objetivos trazados utilizar las herramientas necesarias.

Objetivos del sistema.

La selección de los modelos de hornos la hace, a través de búsquedas heurísticas, sobre reglas de inferencias en PROLOG, que describen las relaciones de los hornos con los tipos de tratamientos.

La selección optimizada de los modelos de hornos, para solera móvil, necesarios para ejecutar la producción y realización del balance carga capacidad.

Este sistema se destina para el cálculo de capacidades productivas; e instalaciones para el tratamiento térmico de semiproductos de acero fundidos, forjados o soldados; y también para hierro fundido, empleando concretamente hornos de cámara de carro o solera móvil (calentado eléctricamente o por combustible); aunque también se puede incluir otros tipos de hornos en la base de hechos.

Se ha considerado la posibilidad de analizar los siguientes tipos de tratamientos, en forma general.

-RECOCIDO	-REVENIDO
-NORMALIZADO	-RECOCIDO SUBCRITICO
-TEMPLE	

Características del sistema.

Módulo de datos: Se describen las características de los hornos, tipos de tratamientos y los turnos de trabajo.

Módulo de entrada: Los datos son pedidos en forma de consulta por el sistema, el cual da explicaciones sobre la forma de teclearlos. En casi todos los casos los datos se pueden dar de 3 formas: como valores fijos, como intervalos o en caso de desconocerse "no sé".

Módulo de solución: Realiza la exploración heurística de los posibles espacios de estado. Este módulo lo primero que hace es ir a la memoria de trabajo a buscar la base de datos creada por el módulo de entrada y da la posibilidad de: verificar soluciones, calcular la solución o generación de soluciones, este último consta de 3 fases.

Fase de análisis: Su función es evitar la selección de soluciones inútiles, para ello este módulo realiza un corte sobre los datos eliminando la información superflua. Su función es verificar la validez de los datos, formar la base de hechos del problema y de existir contradicciones almacenan las explicaciones sobre las fallas encontradas en la base de soluciones.

Fase de selección: Su función es buscar en la base de hechos y generar las posibles soluciones del problema.

Fase de revisión: Le aplica restricciones a las posibles soluciones, las cuales de ser admitidas, pasan a la base de soluciones del problema, de no existir ninguna solución aceptable se dejan en la base de estado las que tengan un significado más próximo al esperado de acuerdo con las restricciones impuestas.

Modulo de salida: imprime las soluciones aceptadas en la base de solución o las soluciones más próximas almacenadas en la base de estado.

Módulo de enlace: realiza el enlace de todos los módulos.

Ejemplos de hechos.

indicador (solera, media, temple, 90).

solera_movil (temp, HORNO).

Donde horno es una estructura functorial del tipo:

HORNO = horno (TIPO_DE_ENERGIA, MODELO, LONGITUD, ANCHO, ALTURA, SUPERFICIE).

Ejemplos de reglas.

SI se selecciono el horno Y se selecciono el indicador Y se selecciono el fondo de tiempo ENTONCES se genero una posible solución.

SI se selecciono algún horno Y se borraron los hornos de la BD ENTONCES el análisis del horno fue correcto.

SI se dieron las dimensiones para seleccionar los hornos Y se incluyeron los hornos seleccionados en la BH ENTONCES se selecciono algún horno.

SI se dio el volumen de producción Y no se dio el horno O se dio el horno Y se dijo el país del horno Y fue incluido en la BS ENTONCES ir a resolver.

SI se dio el indicador en forma de intervalo Y se creó la BH para ese intervalo Y se borraron los indicadores de la BD ENTONCES análisis del indicador concluido.

SI se dio un valor como indicador Y se borraron todos los indicadores de la BD ENTONCES análisis del indicador concluido.

BD: son los datos introducidos por el usuario.

BH: son los hechos que se asumen como verdadero, puede ser cambiado por el sistema en dependencia de los datos dados.

BE: son los resultados más próximos a la solución esperada.

BS: son las soluciones propuestas por el sistema, en caso de no existir ninguna solución, se darán explicaciones sobre las causas del fracaso.

Las reglas permiten la generación y prueba de soluciones aplicando restricciones explícitas (dadas por el usuario) o implícitas (incluidas en las reglas).

Ejemplo de un programa en Javascript.

Se tienen dos tinas una de 3 gls y otra de 4 gls, ambas vacías y una pila o llave de agua, para ir llenándolas. Se quiere dejar dos galones en la tina de 4 gls, ejecutando las acciones de llenar una tina, vaciar una tina en la otra y botar el contenido de una tina.

Las reglas van a ser del tipo: Si A y B y ... Entonces X

Por ejemplo la regla 1 seria:

Si <la tina de 3 gls está vacía> y <la tina de 4 gls está vacía> entonces <llenar la tina de 3 gls> y <se imprime la acción>

Este es el programa en Javascript.

```
<HTML>
<HEAD>
<TITLE>Tinas</TITLE>
</HEAD>
<BODY>
<CENTER>Programa TINAS <CENTER>
<BR>
<script>
function tinas()
{//Base de hechos
  var x = 0
  var y = 0
  if (document.ftinas.tina[0].checked)
    {x = eval(document.ftinas.tina[0].value)
     document.write("Llenar la tina de 3 gls (3,0)" + "<br>")
    }
  else
    {y = eval(document.ftinas.tina[1].value)
     document.write("Llenar la tina de 4 gls (0,4)" + "<br>")
    }
//Motor de inferencia
for (i=1;i<=15;i++)
  { if (y == 2)
    { document.write("Solucion satisfactoria: Hay dos galones en la tina de 4" + "<br>")
      break
    }
  }
</script>
</BODY>
</HTML>
```

//Base de conocimientos

//Regla 1:

```
if (x == 0 && y == 0)
{ x = 3
  y = 0
  document.write("Llenar la tina de 3 gls (3,0)" + "<br>")
  continue
}
```

//Regla 2:

```
if (x == 0 && y == 0)
{ x = 0
  y = 4
  document.write("Llenar la tina de 4 gls (0,4)" + "<br>")
  continue
}
```

//Regla 3:

```
if (x == 3 && y == 0)
{ x = 0
  y = 3
  document.write("Vaciar la tina de 3 gls en la de 4 gls (0,3)" + "<br>")
  continue
}
```

//Regla 4:

```
if (x == 0 && y == 4)
{ x = 3
  y = 1
  document.write("Vaciar la tina de 4 gls en la de 3 gls (3,1)" + "<br>")
  continue
}
```

//Regla 5:

```
if (x == 0 && y == 3)
{ x = 3
  y = 3
  document.write("Llenar la tina de 3 gls (3,3)" + "<br>")
  continue
}
```

```

}

//Regla 6:
if (x == 3 && y == 1)
{ x = 0
  y = 1
  document.write("Botar el contenido de la tina de 3 gls (0,1)" + "<br>" )
  continue
}

//Regla 7:
if (x == 3 && y == 3)
{ x = 2
  y = 4
  document.write("Vaciar la tina de 3 gls en la de 4 gls (2,4)" + "<br>" )
  continue
}

//Regla 8:
if (x == 0 && y == 1)
{ x = 1
  y = 0
  document.write("Vaciar la tina de 4 gls en la de 3 gls (1,0)" + "<br>" )
  continue
}

//Regla 9:
if (x == 2 && y == 4)
{ x = 2
  y = 0
  document.write("Botar el contenido de la tina de 4 gls (2,0)" + "<br>" )
  continue
}

//Regla 10:
if (x == 1 && y == 0)
{ x = 1
  y = 4
  document.write("Llenar la tina de 4 gls (1,4)" + "<br>" )
  continue
}

```

```

    }

//Regla 11:
if (x == 2 && y == 0)
{
    x = 0
    y = 2
    document.write("Vaciar la tina de 3 gls en la de 4 gls (0,2)" + "<br>" )
    continue
}

//Regla 12:
if (x == 1 && y == 4)
{
    x = 3
    y = 2
    document.write("Vaciar la tina de 4 gls en la de 3 gls (3,2)" + "<br>" )
    continue
}

}

}

</script>

```

Cual de las tinas prefiere llenar?

```

<br>
<form name="ftinas">
<input type="Radio" name="tina" value="3" checked> Tina de 3 gls
<br>
<input type="Radio" name="tina" value="4"> Tina de 4 gls
<br>
<input type="Button" name="" value=" Entrar " onclick="tinas()">
<br>
</form>
</BODY>
</HTML>

```

Observen que el programa consta de una base de hechos, un motor de inferencia y base de conocimientos compuesta por 12 reglas. Noten que las reglas impares se refieren a comenzar llenado la tina de 3 gls y las reglas pares son para el llenado de la tina de 4 gls.

El motor de inferencia lo único que hace es recorrer todas las reglas y en caso de que se llegue a la

solución detiene el proceso.

La base de hechos inicializa los hechos (pone en cero las variables) y prepara las condiciones para la interfaz con el usuario.

Es bueno destacar que la base de conocimientos (BC) y la máquina de inferencias (MI) están separados lógicamente (usted puede seguir agregando reglas sin que esto afecte la ejecución del programa), pero físicamente están unidas dentro del mismo programa. En la práctica la BC y la MI deben estar separadas lógica y físicamente.

Pruebe agregar la siguiente regla y verá que no afecta para nada al programa:

//Regla 13:

```
if (x == 3 && y == 4)
{
    x = 3
    y = 0
    document.write("Botar el contenido de la tina de 4 glos (3,0)" + "<br>")
    continue
}
```

La regla 13 quiere decir que si está llena la tina de 3 y está llena la tina de 4 entonces botar el contenido de la tina de 4.

Esto es un ejemplo de cómo cualquier lenguaje por simple que parezca puede ser utilizado para resolver problemas que requieran de inteligencia.

Conclusión.

A través de este sistema se puede lograr la simulación de soluciones y realizar comparaciones con los modelos de hornos dados como resultado, así como predecir las soluciones más eficientes.

Por otra parte los datos dados en forma opcional le sirven de restricción al sistema en su búsqueda, por tanto, mientras menos datos se den, habría menos restricciones y mayor sería el rango de modelos de hornos a seleccionar.

Mediante este sistema puede llevarse a cabo la verificación de hornos propuestos o existentes en obras, o considerar posibles ampliaciones del equipamiento.

Como conclusión podemos decir, que con técnicas sencillas de inteligencia artificial, se puede lograr aplicaciones útiles, las cuales con la programación convencional, se haría engorrosa y lenta en su concepción.