

# ***SISTEMAS EXPERTOS***

## **GRUPO 01**

- DEFINICION DE SISTEMAS EXPERTOS
  - MARCO HISTORICO
- APLICACIONES DE SISTEMAS EXPERTOS
- CONSIDERACIONES DE LENGUAJE DE PROGRAMACION
  - REPRESENTACION DEL CONOCIMIENTO

# ***DEFINICION DE UN SE***

## **Qué es un sistemas experto?**

Los sistemas expertos son programas que reproducen el proceso intelectual de un experto humano en un campo particular, Estos sistemas permiten la creación de máquinas que razonan como el hombre, restringiéndose a un espacio de conocimientos limitado.

Los sistemas expertos tienen la ventaja frente a otro tipos de programas de Inteligencia Artificial, de proporcionar gran flexibilidad a la hora de incorporar nuevos conocimientos.

### **Descripción Del Esquema**

Para realizar un sistema experto integran dos personas el Experto del Dominio (profesional X) y un Ingeniero de Conocimiento (programador),

## Descripción Del Esquema

Para realizar un sistema experto integran dos personas el Experto del Dominio (profesional X) y un Ingeniero de Conocimiento (programador)

La Base de conocimiento nos halla la base datos y estas esta compuestas por lenguajes de predicado, esta es uno de los componentes que contiene el conocimiento del experto o también llamado base de datos

Existen dos tipos de base de conocimiento:

El procedural ;

Se usa en los lenguajes. estructurados como son Pascal, C, Visual Basic etc.

El declarativo;

Esta basado en hechos que vienen a ser acciones que se dan dentro del problema se ulitizan los lenguajes Prolog y Lisp.

## El Motor de Inferencia

Su función es administrar , como, cuando, y las reglas de producción que se aplicaran para la solución de un determinado problema

## La Interface

Parte que permite la comunicación con el usuario, en forma bidireccional(ambos lados).

## DEMONIO

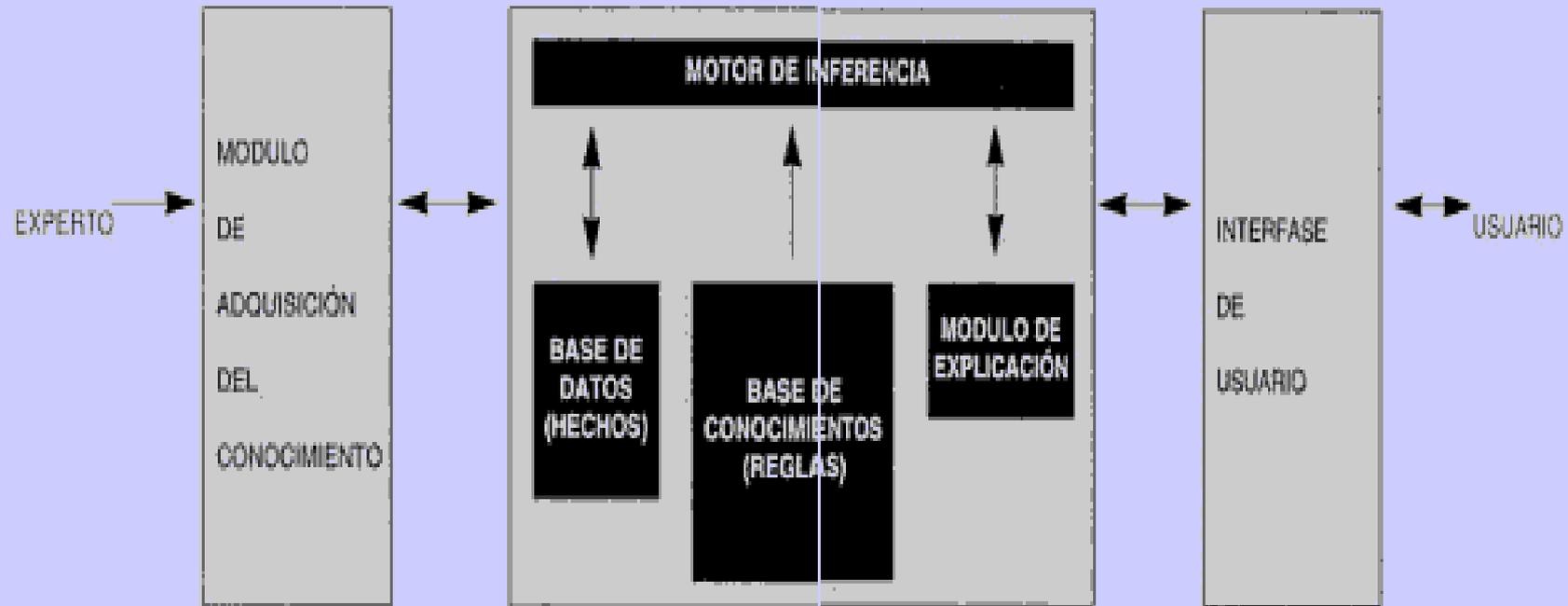
Es la parte principal de la estructura de control el cual va seguir un encadenamiento hacia atrás y hacia delante y esta a su vez está compuesta de dos campos específicos PROCEDIMIENTOS ESPECIALES son los pasos a seguir compuestas por reglas, normas de producción, ELEMENTOS DE METACONOCIMIENTO compuestas por redes neuronales, por que está e la capacidad de aprender, entender y responder a la pregunta realizada por un usuario.

## CARACTERISTICAS DE LOS SE

Los SE pueden considerarse como intermediarios entre el experto humano, que transmite su conocimiento al sistema, y el usuario que lo utiliza para resolver un problema con la eficacia del especialista

- El usuario puede aprender observando el comportamiento del sistema.
- Mecanismos que forman ese conocimiento difícil de expresar que permite que los expertos humanos sean eficaces calculando lo menos posible.
- La característica fundamental de un sistema experto es que separa los conocimientos almacenados (base de conocimiento) del programa que los controla (motor de inferencia). Los datos propios de un determinado problema se almacenan en una base de datos aparte (base de hechos).
- Una característica adicional deseable, y a veces fundamental, es que el sistema sea capaz de justificar su propia línea de razonamiento de forma inteligible por el usuario.

# ARQUITECTURA DE UN SE



# ***REGLAS DE UN SE***

El dominio de conocimiento representado se divide, pues, en pequeñas fracciones de conocimiento o reglas SI . . .

ENTONCES . . . Cada regla constará de una parte denominada condición y de una parte denominada acción

SI el termómetro marca 39°

Y el termómetro funciona correctamente

ENTONCES el paciente tiene fiebre

Una característica muy importante es que la base de conocimientos es independiente del mecanismo de inferencia que se utiliza para resolver los problemas.

## *... CONTINUACION*

No es necesario reprogramar todo el sistema experto.

Las reglas suelen almacenarse en alguna secuencia jerárquica lógica metarreglas (reglas sobre otras reglas) y su función es indicar bajo qué condiciones deben considerarse unas reglas en vez de otras.

SI hay reglas que usan materias baratas

Y hay reglas que usan materias caras

ENTONCES usar antes las primeras que las segundas

La base de datos o base de hechos es una parte de la memoria del ordenador que se utiliza para almacenar los datos recibidos inicialmente para la resolución de un problema.

# ***MOTOR DE INFERENCIA***

El motor de inferencias es un programa que controla el proceso de razonamiento que seguirá el sistema experto. El sistema sigue los siguientes pasos:

1. Evaluar las condiciones de todas las reglas respecto a la base de datos
2. Si no se puede aplicar ninguna regla, se termina sin éxito; en caso contrario se elige cualquiera de las reglas aplicables y se ejecuta su parte acción
3. Si se llega al objetivo, se ha resuelto el problema; en caso contrario, se vuelve al paso 1

A este enfoque se le llama también guiado por datos, porque es el estado de la base de datos el que identifica las reglas que se pueden aplicar

# *Guiado por Objetivos*

Al encadenamiento regresivo se le suele llamar guiado por objetivos, ya que, el sistema comenzará por el objetivo (parte acción de las reglas) y operará retrocediendo para ver cómo se deduce ese objetivo partiendo de los datos.

1. Obtener las reglas relevantes, buscando la expresión E en la parte acción
2. Si no se encuentran reglas para aplicar, entonces no se tienen datos suficientes para resolver el problema; se termina sin éxito o se piden al usuario más datos
3. Si hay reglas para aplicar, se elige una y se verifica su parte condición C con respecto a la base de datos.
4. Si C es verdadera en la base de datos, se establece la veracidad de la expresión E y se resuelve el problema
5. Si C es falsa, se descarta la regla en curso y se selecciona otra regla
6. Si C es desconocida en la base de datos (es decir, no es verdadera ni falsa), se le considera como subobjetivo y se vuelve al paso 1 (C será ahora la expresión E)

# *Mas sobre características de los SE*

La mayoría de los sistemas expertos contienen un módulo de explicación, diseñado para aclarar al usuario la línea de razonamiento seguida en el proceso de inferencia. Si el usuario pregunta al sistema cómo ha alcanzado una conclusión, éste le presentará la secuencia completa de reglas usada. Esta posibilidad de explicación es especialmente valiosa cuando se tiene la necesidad de tomar decisiones importantes amparándose en el consejo del sistema experto

El módulo de adquisición del conocimiento permite que se puedan añadir, eliminar o modificar elementos de conocimiento (en la mayoría de los casos reglas) en el sistema experto.

# Catalogación de Sistemas Expertos

- **ANALÍTICOS O INTERPRETATIVOS:**

- **Diagnósticos.**
- **Entendimiento.**

- **SINTÉTICOS O GENERATIVOS:**

- **Resolución de problemas.**
  - **Diseño.**
  - **Tratamiento.**

# *Base de conocimiento*

Los conocimientos que se suelen almacenar son de cuatro tipos: Conocimiento objetivo, que describe la situación real del sistema; Conocimiento de sucesos, relacionado con los sucesos que ocurren en el tiempo de ejecución; Conocimiento del funcionamiento del sistema, de cómo se hacen las cosas, y Metaconocimiento, que relaciona los anteriores

Existen varias formas de representar el conocimiento:

Marcos («Frames»): son estructuras de datos donde se almacenan información concreta de un cierto concepto e información relacional para completar la definición del concepto

- Redes semánticas: son representaciones gráficas del conocimiento, mediante nodos, que representan objetos o conjuntos de objetos, y arcos, que relacionan dichos objetos.

# *Base de conocimiento y base de Hechos*

- Reglas: es la forma más extendida de representación del conocimiento. Representan la forma de razonar. Tienen la forma IF <condicion> THEN <accion/conclusion>. Es importante el orden de ejecución de las reglas, por ello se les suele dotar de prioridades. Una regla que activa otra se denomina metaregla, y se suelen usar para desarrollar progresivamente el conocimiento del experto.

## Base de hechos

Representan el conocimiento del estado del sistema en un cierto instante. Usualmente se representa en una base de datos, y su información está directamente enlazada con la base de conocimientos.

# ***MECANISMOS DE INFERENCIA***

Se pueden diferenciar dos mecanismos de inferencia:

- Encadenamiento hacia adelante: se extraen conclusiones a partir del cumplimiento de las condiciones de ciertas reglas
- Encadenamiento hacia atrás: se suponen ciertas las conclusiones de una regla y, como consecuencia, se van disparando aquellas reglas que provocarían la regla original.

# *INTERFACE HOMBRE-MAQUINA*

Existen cuatro alternativas para la implementación de un sistema experto:

" Lenguajes de alto nivel (HLL): son lenguajes de propósito general (C, Fortran, Basic,...). Tienen como ventajas su eficiencia, su familiaridad y la portabilidad a cualquier entorno, pero el inconveniente de no estar preparados para una programación basada en símbolos.

" Lenguajes simbólicos: fundamentalmente LISP y PROLOG. Son los lenguajes de alto nivel más utilizados por su adaptación a la lógica de la base de conocimientos representada mediante símbolos, y por su mecanismo de extraer conclusiones. Su eficiencia aumenta en estaciones de trabajo diseñadas para dichos lenguajes.



" Herramientas de desarrollo ("shells"): son programas preparados para el desarrollo de sistemas expertos pues incorporan el motor de inferencia, independiente de las bases de hechos y de conocimiento.

" Entornos de desarrollo ("environments"): añaden a los "shells" el uso de un entorno gráfico (ventanas, iconos,...) que facilita el desarrollo del sistema experto.

# Razonamiento Basado en Casos. (Case-Based Reasoning)

Los sistemas de razonamiento basado en casos («Case-Based Reasoning») pertenecen al área de la inteligencia artificial y se utilizan para la resolución de problemas. Se basan en problemas similares ocurridos en el pasado, que se denominan casos

La diferencia más importante entre ambos sistemas estriba en la forma de resolver los problemas: en los sistemas expertos se utiliza el razonamiento deductivo a partir de los datos conocidos, mientras que en los sistemas de razonamiento basado en casos se utiliza el razonamiento analógico, es decir, a partir del problema se usa el conocimiento de casos históricos.

# ***UTILIZACION DE LOS SISTEMAS EXPERTOS***

- El entorno del problema es rico en conocimiento del mismo y pobre en experiencias previas, fundamentalmente por su inviabilidad.
- No es fácil comparar unos problemas con otros.
- La experiencia se puede implementar en forma de reglas.
- No se puede aprender del pasado.

Los sistemas de razonamiento basado en casos se utilizan entonces cuando:

- Los problemas se representan mejor en forma de conocimiento del medio que en forma de reglas.
- Es más fácil extraer conocimiento mediante casos ya ocurridos que mediante reglas. Así se hace menos costosa la adquisición del conocimiento.
- Es necesario incorporar creatividad y sentido común para resolver el problema.
- Se repiten los problemas cada cierto tiempo.

# MODELO BUCHANAN

*Etapa 1: Familiarizarse con el Problema y el Dominio.*

*Abarca desde la lectura de libros o artículos, las entrevistas o charlas con las personas familiarizadas con el tema y la búsqueda de un experto que esté dispuesto a colaborar en la construcción del sistema; como así también la definición de cuales son la funciones o tareas más idóneas para ser realizadas por el sistema experto.*

*Etapa 2: Delimitar el Sistema.*

*Significa que por medio de entrevistas con el experto, con el objetivo de identificar y caracterizar el problema informalmente. El experto de campo y el ingeniero de conocimiento definen el alcance del sistema experto, es decir, que problemas va a resolver concretamente el sistema experto.*

# ***MODELO BUCHANAN***

## ***Etapa 3: Obtener la Estructura de Inferencia del Sistema Experto.***

Con el problema adecuadamente definido el ingeniero de conocimiento empieza a determinar los principales conceptos del dominio que se requieren para realizar cada una de las tareas que va a resolver el sistema. Este trata de entender que conceptos son relevantes e importantes solicitándole al experto que explique y justifique los razonamientos que utiliza para resolver los problemas.

El ingeniero de conocimiento debe prestar atención al experto de campo para encontrar la estructura básica que el experto utiliza para resolver el problema

El ingeniero de conocimiento además debe reconocer las estrategias básicas que usa el experto cuando desarrolla su tarea, que hechos establece primero, que tipos de preguntas realiza primero

# .... CONTINUACION

## ***Etapa 4: Definir el Sistema Experto Prototipo.***

El ingeniero de conocimiento debe formalizar el conocimiento obtenido del experto. Esta tarea implica definir que arquitectura permitirá una mejor organización del conocimiento.

1. que el formalismo usado es el apropiado para reflejar los conceptos y el proceso de inferencia del experto.
2. que las características particulares de construcción del lenguaje capturen exactamente los aspectos estructurales más importantes de los conceptos usados por el experto.
3. que la estructura del control del lenguaje al activar las reglas refleje la estrategia usada por el experto.
4. que las reglas reflejen asociaciones y métodos que:
  - a) son los usados por el experto.
  - b) son modelos aceptables de dichos métodos.

# ***APLICACIONES***

- \* Auditoría: Análisis de la materialidad y del riesgo, evaluación del control interno, planificación de la auditoría, evaluación de la evidencia, análisis de cuentas concretas, formación de opinión, emisión del informe, auditoría interna, auditoría informática, etc.
- \* Contabilidad de costes y de gestión: Cálculo y asignación de costes, asignación de recursos escasos, control y análisis de desviaciones, planificación y control de gestión, diseño de sistemas de información de gestión, etc.

# ***APLICACIONES***

\* Contabilidad financiera: regulación legal, normas y principios contables, recuperación y revisión analítica de registros contables, diseño de sistemas contables, imputación contable, consolidación de estados contables, etc.

\* Análisis de estados financieros: Análisis patrimonial, financiero y económico de los estados contables, salud financiera de la empresa, cálculo e interpretación de ratios, cálculo y análisis de tendencias, etc.

# *Lenguajes de programación de sistemas expertos*

Los dos lenguajes más importantes en el desarrollo de Sistemas Expertos son:

- ◆ **LISP** (List-Processing)
- ◆ **PROLOG** (**PRO**gramación **LOG**ica)
- ◆ **RITA, ROSIE, y ROOS.**

# *LISP (List-Processing)*

Algunos datos historicos:

- ◆ Fue diseñado a finales de los años 50 por **McCarthy** .
- ◆ Desarrollado para el procesamiento de listas.
- ◆ A lo largo de los últimos años se han desarrollado muchos dialectos, por ejemplo *MACLISP*, *COMMONLISP*, *INTERLISP*, *ZETALISP*, donde el **COMMONLISP** se está imponiendo cada vez más como estándar.

# *LISP (List-Processing)*

En LISP se dan los siguientes conceptos característicos:

## ◆ Listas y Átomos

◆ La estructura más importante es la lista. Los átomos pueden subordinarse a cualidades.

◆ ( **B C** ) lista de átomos **B y C**.

◆ También está permitida una lista vacía , "( )" ó " NIL" , que significa lo mismo.

# *LISP (List-Processing)*

- ◆ Los átomos son números , cadenas de caracteres o símbolos. Un símbolo puede tener varios valores, al igual que una variable en otros lenguajes de programación, como por ejemplo un número, o también puede ser el nombre de una función, o incluso ambos.
- ◆ Además a un símbolo pueden subordinarse cualidades, que además del valor del símbolo, contienen información adicional. Estas cualidades también reciben el nombre de atributos.

# *LISP (List-Processing)*

## ◆ La Función

Cada función LISP y cada programa LISP tiene estructura de lista. Los programas no pueden distinguirse sintácticamente de los datos. LISP ofrece sus propias funciones básicas.

- ◆ Forma de Trabajo: LISP es un lenguaje funcional. Ofrece la posibilidad de realizar definiciones recursivas de funciones. La unión de procedimientos se realiza de forma dinámica, es decir en plena ejecución, y no como en otros lenguajes de programación. El sistema realiza automáticamente una gestión dinámica de memoria.

# *LISP (List-Processing)*

- ◆ Un componente importante de un sistema LISP es la gestión dinámica de la memoria. El sistema administrará el espacio en la memoria para las listas en constante modificación, sin que el usuario lo deba solicitar. Libera los espacios de memoria que ya no son necesarios y los pone a disposición de usos posteriores. La necesidad de este proceso se deriva de la estructura básica de LISP, las listas, que se modifican de forma dinámica e ilimitada.
- ◆ Además un sistema LISP abarca bastante más que el solo intérprete del lenguaje LISP. Consta de algunos cómodos módulos que ofrecen ayuda en el desarrollo y control del progreso en programas, como son el Editor, el File-System y el Trace. Por supuesto estos módulos sólo están en versiones de LISP que contengan la conocida interfaz gráfica IDE típica de los modernos lenguajes visuales. (IDE = entorno de desarrollo integrado).

# ***PROLOG (PROgramación LOGica)***

- ◆ Es una realización de lógica de predicados, como lenguaje de programación.
- ◆ La lógica se representa en forma de predicados. Estos predicados aparecen en tres formas distintas: como hechos, como reglas y como preguntas. La lógica formulada como hechos y reglas se define como base de conocimientos. A esta base de conocimientos se le pueden formular preguntas.

## ***RITA, ROSIE, y ROOS***

- ◆ Para el desarrollo de los sistemas expertos se utilizan lenguajes de ingeniería del conocimiento. Los lenguajes más difundidos para el desarrollo de los SE son **RITA, ROSIE, y ROOS.**

# *Representación de Conocimiento*

- ◆ En organismos biológicos se estima que el conocimiento es almacenado como estructuras complejas de neuronas interconectadas.
- ◆ En las computadoras, el conocimiento también se almacena como estructuras simbólicas, pero en forma de estados eléctricos y magnéticos.
- ◆ En forma natural, el ser humano representa el conocimiento simbólicamente: imágenes, lenguaje hablado y lenguaje escrito. Adicionalmente, ha desarrollado otros sistemas de representación del conocimiento: literal, numérico, estadístico, estocástico, lógico.

# *Representación de Conocimiento*

- ◆ La ingeniería cognoscitiva ha adaptado diversos sistemas de representación del conocimiento que, implantados en un computador, se aproximan mucho a los modelos elaborados por la psicología cognoscitiva para el cerebro humano. Entre los principales se tienen:

## **Lógica Simbólica Formal:**

- Lógica proposicional
- Lógica de predicados.
- Reglas de producción.

## **Formas Estructuradas:**

- Redes asociativas.
- Estructuras marco.
- Representación orientada a objetos.

# *Lógica Proposicional*

- ◆ La lógica proposicional es la más antigua y simple de las formas de lógica.
- ◆ Utilizando una representación primitiva del lenguaje, permite representar y manipular aserciones sobre el mundo que nos rodea.
- ◆ La lógica proposicional permite el razonamiento, a través de un mecanismo que primero evalúa sentencias simples y luego sentencias complejas, formadas mediante el uso de conectivos proposicionales, por ejemplo Y (AND), O (OR). Este mecanismo determina la veracidad de una sentencia compleja, analizando los valores de veracidad asignados a las sentencias simples que la conforman.

# *Lógica Proposicional*

Una proposición es una sentencia simple que tiene un valor asociado ya sea de verdadero (V), o falso (F). Por ejemplo:

◆ **Hoy es Miércoles, Anteayer llovió, Hace Calor**

Las proposiciones pueden combinarse para expresar conceptos más complejos. Por ejemplo:

**hoy\_es\_Viernes y hace\_frío.**

**Conectores básicos de la lógica proposicional**

NOMBRE	CONECTOR	SÍMBOLO	DENOMINACIÓN	REPRESENTACIÓN LÓGICA
Conjunción	AND	$\wedge$	Leyes Equipotenciales	$A \Rightarrow B = \sim A \vee B$
Disyunción	OR	$\vee$		$A \wedge \sim A = F$
Negación	NOT	$\sim$		$A \vee \sim A = V$
Implicación	If-Then	$\Rightarrow$	Leyes Conmutativas	$A \wedge B = B \wedge A$
Equivalencia	Igual	$=$		$A \vee B = B \vee A$
			Leyes Distributivas	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
			Leyes Asociativas	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$ $A \vee (B \vee C) = (A \vee B) \vee C$
			Leyes Absortivas	$A \wedge (A \vee B) = A$ $A \vee (A \wedge B) = A$
			Leyes de DeMorgan	$\sim(A \wedge B) = \sim A \vee \sim B$ $\sim(A \vee B) = \sim A \wedge \sim B$

**Equivalencias en lógica proposicional**

# *Lógica de predicados*

- ◆ La lógica de predicados está basada en la idea de las sentencias realmente expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales cualidades, relaciones o atributos, se denominan predicados. Los objetos se conocen como argumentos o términos del predicado.

# *Lógica de predicados*

- ◆ Al igual que las proposiciones, los predicados tienen un valor de veracidad, pero a diferencia de las proposiciones, su valor de veracidad, depende de sus términos. Es decir, un predicado puede ser verdadero para un conjunto de términos, pero falso para otro.

Por ejemplo, el siguiente predicado es verdadero:

- ◆ **color (yerba, verde)**

El mismo predicado, pero con diferentes argumentos, puede no ser verdadero:

- ◆ **color (yerba, azul)            o            color (cielo, verde)**

# *Lógica de predicados*

- ◆ Cada uno de los argumentos en los ejemplos de predicados dados anteriormente, representan a un objeto específico. Tales argumentos se denominan constantes. Sin embargo, en la lógica de predicados se pueden tener argumentos que en determinado momento pueden ser desconocidos. Estos son los argumentos tipo variable.
- ◆ En el ejemplo: color (yerba, X), la variable X, puede tomar el valor de verde, haciendo que el predicado sea verdadero; o puede tomar el valor de azul, dando lugar a que el predicado sea falso.

# *Reglas de producción*

Los sistemas basados en reglas son los más comúnmente utilizados. Su simplicidad y similitud con el razonamiento humano, han contribuido para su popularidad en diferentes dominios. Las reglas son un importante paradigma de representación del conocimiento.

Las reglas representan el conocimiento utilizando un formato SI-ENTONCES (IF-THEN), es decir tienen 2 partes:

La parte SI (IF), es el antecedente, premisa, condición o situación; y

La parte ENTONCES (THEN), es el consecuente, conclusión, acción o respuesta.

# *Reglas de producción*

Las reglas pueden ser utilizadas para expresar un amplio rango de asociaciones, por ejemplo:

SI está manejando un vehículo Y se aproxima una ambulancia, ENTONCES baje la velocidad Y hágase a un lado para permitir el paso de la ambulancia.

# ***El Proceso de Razonamiento***

El proceso de razonamiento en un sistema basado en reglas es una progresión desde un conjunto inicial de afirmaciones y reglas hacia una solución, respuesta o conclusión. Como se llega a obtener el resultado, sin embargo, puede variar significativamente:

- ◆ Se puede partir considerando todos los datos conocidos y luego ir progresivamente avanzando hacia la solución. Este proceso se lo denomina *guiado por los datos* o de **encadenamiento progresivo** (*forward chaining*).
- ◆ Se puede seleccionar una posible solución y tratar de probar su validez buscando evidencia que la apoye. Este proceso se denomina *guiado por el objetivo* o de **encadenamiento regresivo** (*backward chaining*).

# ***Razonamiento Progresivo***

En el caso del razonamiento progresivo, se empieza a partir de un conjunto de datos colectados a través de observación y se evoluciona hacia una conclusión. Se chequea cada una de las reglas para ver si los datos observados satisfacen las premisas de alguna de las reglas. Si una regla es satisfecha, es ejecutada derivando nuevos hechos que pueden ser utilizados por otras reglas para derivar hechos adicionales. Este proceso de chequear reglas para ver si pueden ser satisfechas se denomina *interpretación de reglas*.

La interpretación de reglas es realizada por una máquina de inferencia en un sistema basado en conocimiento. La interpretación de reglas, o inferencia, en el razonamiento progresivo involucra la repetición de los pasos que se indican en la siguiente figura.

**Unificación (*Matching*).**- En este paso, en las reglas en la base de conocimientos se prueban los hechos conocidos al momento para ver cuáles son las que resulten satisfechas. Para decir que una regla ha sido satisfecha, se requiere que todas las premisas o antecedentes de la regla resuelvan a verdadero.

**Resolución de Conflictos.**- Es posible que en la fase de unificación resulten satisfechas varias reglas. La resolución de conflictos involucra la selección de la regla que tenga la más alta prioridad de entre el conjunto de reglas que han sido satisfechas.

**Ejecución.**- El último paso en la interpretación de reglas es la ejecución de la regla. La ejecución puede dar lugar a uno o dos resultados posibles: nuevo hecho (o hechos) pueden ser derivados y añadidos a la base de hechos, o una nueva regla (o reglas) pueden ser añadidas al conjunto de reglas (base de conocimiento) que el sistema considera para ejecución.

# *Redes Asociativas*

- ◆ Las redes asociativas se caracterizan por representar el conocimiento en forma gráfica. Agrupan una porción de conocimiento en dos partes: objetos y relaciones entre objetos. Los objetos se denominan también nodos (elementos del conocimiento) y las relaciones entre nodos se denominan enlaces o arcos. Cada nodo y cada enlace en una red semántica, deben estar asociados con objetos descriptivos.

# *Representación mediante Plantillas*

Una *plantilla* representa un objeto o situación describiendo la colección de atributos que posee. Están formadas por un nombre y por una serie de campos de información o *ranuras* (*slots*). Cada *ranura* puede contener uno o más *enlaces* (*facets*). Cada *enlace* tiene un valor asociado. Varios *enlaces* pueden ser definidos para cada *ranura*, por ejemplo:

- ◆ *Rango*.- El conjunto de posibles valores para la ranura.
- ◆ *Valor*.- El valor de la ranura.
- ◆ *Default*.- El valor a ser asumido si no se especifica alguno.

# Representación mediante Objetos

Un objeto es definido como una colección de información representando una entidad del mundo real y una descripción de cómo debe ser manipulada esta información, esto es los *métodos*. Es decir, un objeto tiene un nombre, una caracterización de clase, varios atributos distintivos y un conjunto de operaciones. La relación entre los objetos viene definida por los *mensajes*. Cuando un objeto recibe un *mensaje* válido, responde con una acción apropiada, retornando un resultado.

NOMBRE OBJETO	Limpiador Izquierdo
UN-TIPO-DE	Limpiador
ATRIBUTOS	Hecho de metal y caucho Longitud: 14 pulgadas Color: negro y plateado Localización: inferior izquierda
OPERACIONES	Función: remover humedad de parabrisa Activado: se mueve en arco sobre el parabrisa repetidamente de izquierda a derecha Desactivado: se mueve a posición de

reposo

# ***LINKS PARA FAUSTO Y DAVID***

<http://www.monografias.com/trabajos10/exper/exper.shtml>

[www.dei.va.edu/cpy/tar17/sexpreti/historia.htm](http://www.dei.va.edu/cpy/tar17/sexpreti/historia.htm)

[www.sarder.upeu.edu.pe/~miguel+/temas/expertos.htm](http://www.sarder.upeu.edu.pe/~miguel+/temas/expertos.htm)

[www.fotunecity.com/skycod.per/ronrow/207/se/portada.htm](http://www.fotunecity.com/skycod.per/ronrow/207/se/portada.htm)

<http://www.psicologiacientifica.com/publicaciones/biblioteca/articulo>

<http://www.uco.es/~i62lapeb/Practica%202/Cuestiones.htm>

<http://www.esi2.us.es/~dco/sistemas.htm#INDICE>

**NOTA: LES PROMETI MANDARLES LOS LINKS PARA QUE ENTIENDAN LO QUE ES UN MOTOR DE INFERENCIA Y COMO FUNCIONA**