

UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INDUSTRIAL  
SISTEMAS EXPERTOS E INTELIGENCIA ARTIFICIAL



INVESTIGACIÓN:

REDES SEMANTICAS:

ESQUEMAS, MARCOS, LOGICA PROPOSICIONAL Y CUANTIFICADORES

CATEDRÁTICO:

ING. CRUZ ANTONIO GALDAMEZ RIVERA

ESTUDIANTE:

MEJÍA AGUILAR, CARLOS ALBERTO  
PÉREZ CHACÓN, MARÍA ELICIA

SOYAPANGO, 30 DE JUNIO DEL 2004

## ÍNDICE

<i>Introducción</i> .....	3
<i>Objetivos</i> .....	4
<i>Sistemas Expertos</i> .....	5
<i>Componentes de un Sistema Experto</i> .....	6
Representación de la Base de Conocimiento en un Sistema Experto.....	11
Lógica Preposicional.....	11
<i>Redes Semántica</i> .....	15
<i>Frames (Marcos o Prototipos</i> .....	21)
Esquemas de representación de conocimiento formalizados.....	23
<i>Cuantificadores</i> .....	29
<i>Conclusión</i> .....	34

## Introducción

El proceso de conocer o aprender se basa en la adquisición o representación de la realidad, ya sea de un objeto o un proceso que represente dicho ente.

La representación del conocimiento es una de las tareas más complejas con las que se ha enfrentado el hombre desde sus orígenes, y muchos problemas no han tenido solución por la compleja característica que el hombre por naturaleza tiene de describir la realidad de forma simbólica.

Existen múltiples formas de representar el conocimiento en un sistema, y cada una de ellas presenta alguna ventaja sobre el resto.

Las formas más importantes de representación del conocimiento son lógica proposicional, redes semánticas, marcos o prototipos, esquemas, cuantificadores.

Para una mejor comprensión de estas formas se detallan a continuación.

## Objetivos:

### *General:*

*Conocer las formas mas importantes de representación del conocimiento en un Sistema Experto.*

### *Específicos :*

- *Conocer que es un Sistema Experto.*
- *Áreas de aplicación de un Sistema Experto.*
- *Componentes de un Sistema Experto.*
- *Aplicaciones de un Sistema Experto.*
- *Representación de la Base del Conocimiento de un Sistema Experto.*

## Sistemas Expertos

### ¿Qué son?

Los Sistemas Expertos, rama de la Inteligencia Artificial, son sistemas informáticos que **simulan** el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción de un **experto humano** en cualquier rama de la ciencia.

Estas características le permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, aprender de la experiencia y los datos existentes, comunicarse con expertos humanos, explicar el por qué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior.

### ¿Por qué utilizar un Sistema Experto?

1. Con la ayuda de un Sistema Experto, **personas con poca experiencia** pueden resolver problemas que requieren un "conocimiento formal especializado".
2. Los Sistemas Expertos pueden obtener conclusiones y resolver problemas de forma **más rápida** que los expertos humanos.
3. Los Sistemas Expertos razonan pero en base a un conocimiento adquirido y **no tienen sitio para la subjetividad**.
4. Se ha comprobado que los Sistemas Expertos tienen al menos, **la misma competencia** que un especialista humano.
5. El uso de Sistemas Expertos es especialmente recomendado en las siguientes situaciones:
  - Cuando los expertos humanos en una determinada materia son escasos.
  - En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.
  - Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

**APLICACIONES:** Medicina, Economía, Psicología, Derecho y prácticamente todas las ramas del conocimiento.

## Componentes de un Sistema Experto

Una característica decisiva de los Sistemas Expertos es la separación entre conocimiento (reglas, hechos) por un lado y su procesamiento por el otro. A ello se añade una interfase de usuario y un componente explicativo.

A continuación os muestro una breve descripción de cada uno de los componentes

1. **La Base de Conocimientos** de un Sistema Experto contiene el conocimiento de los hechos y de las experiencias de los expertos en un dominio determinado.
2. **El Mecanismo de Inferencia** de un Sistema Experto puede simular la estrategia de solución de un experto.
3. **El Componente Explicativo** explica al usuario la estrategia de solución encontrada y el porqué de las decisiones tomadas.
4. **La Interfase de Usuario** sirve para que éste pueda realizar una consulta en un lenguaje lo más natural posible.
5. **El Componente de Adquisición** ofrece ayuda a la estructuración e implementación del conocimiento en la base de conocimientos.

### **La base de conocimientos**

La Base de conocimientos contiene todos los hechos, las reglas y los procedimientos del dominio de aplicación que son importantes para la solución del problema.

Para entender la base de conocimientos lo explicare con un ejemplo "marítimo", creo que así todos seréis capaces de entenderlo.

Los hechos son del tipo: El velero "Tiburón" tiene una longitud de 6m. La representación de este conocimiento puede realizarse orientándola, por ejemplo, según objetos. Los objetos de una base de conocimientos pueden ser entonces: barco, barco a motor, barco a vela. Estos objetos están relacionados de tal forma que un barco a vela tiene todas las cualidades de un barco, y además todas las cualidades específicas de un barco a vela. Todas las cualidades de un barco, por ejemplo: Desplazamiento sobre el agua, vienen descritas con el "barco". A través de la relación formulada, el barco a vela "hereda" estas cualidades, de forma que sólo hará falta describir sus cualidades particulares.

Este tipo de programación se define como programación orientada a objetos y se utiliza con frecuencia en el desarrollo de los Sistemas Expertos. Puede darse el caso de que determinados procesos y funciones deban subordinarse a unos objetos en particular, por ejemplo la velocidad como función de la fuerza y la dirección del viento. La velocidad se determinará en función de los datos particulares.

Cómo se lleva a cabo la clasificación en grupos de las características y de los procedimientos alrededor de un objeto con las técnicas de programación, y cómo deben ser las relaciones entre los objetos pueden variar mucho de aplicación a aplicación.

Junto a estos objetos, la base de conocimientos dispone de reglas. Estas reglas se representan en forma de:

### **Si *premisas* Entonces *Conclusión y/o Acción***

En la zona de las premisas se solicitan vinculaciones lógicas referentes a las cualidades de los objetos.

En la zona de la conclusión se añaden nuevos hechos y cualidades a la base de conocimientos y/o se ejecutan acciones. Esto se define a menudo como programación orientada a reglas.

Como podréis deducir se nos plantean muchas preguntas para realizar la labor, como por ejemplo:

- ¿Qué objetos serán definidos ?
- ¿Cómo son las relaciones entre los objetos?
- ¿Cómo se formularán y procesarán las reglas?
- ¿La base de conocimientos hace totalmente referencia a la solución del problema?
- ¿La base de conocimientos es consistente?

Las respuestas a estas preguntas son el trabajo del Ingeniero del conocimiento junto con la colaboración de los expertos.

### **El mecanismo de inferencia**

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

Una conclusión se produce mediante aplicación de las reglas sobre los hechos presentes.

Ejemplo.

Una Regla es: Si p y q entonces r

p y q son justo aquellos hechos que se mencionan en la cláusula "si" de la regla, es decir, las condiciones para la aplicabilidad de la regla. Aplicar la regla es: deducir de los hechos p y q el hecho r.

En un Sistema Experto existirá un hecho sólo cuando esté contenido en la base de conocimientos.

Los hechos que constan en la cláusula "si" se llaman *premisas*, y el contenido en la cláusula "entonces" se llama *conclusión*. Cuando se aplica una regla sobre algunos hechos cualesquiera se dice que se *dispara*. El disparo de una regla provoca la inserción del nuevo hecho en la base de conocimientos.

Las funciones del mecanismo de inferencia son:

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

La decisión sobre los mecanismos de procesamiento de reglas, es decir, qué estrategias de búsqueda se implementarán, es de vital importancia para la efectividad del sistema en su conjunto.

Ante problemas o clases de problemas distintos se estructuran, como es lógico, diferentes mecanismos de inferencia. El mecanismo de inferencia debe de estar "adaptado" al problema a solucionar. Una imposición de dinero exige, bajo ciertas circunstancias, una estrategia distinta de procesamiento del conocimiento que un diagnóstico de fallos de máquina.

### **El componente explicativo**

Las soluciones descubiertas por los expertos deber poder ser repetibles tanto por el ingeniero del conocimiento en la fase de comprobación así como por el usuario. La exactitud de los resultados sólo podrá ser controlada, naturalmente, por los expertos.

Siempre es deseable que durante el trabajo de desarrollo del sistema se conozca el grado de progreso en el procesamiento del problema. Como os he dicho en anterioridad nos pueden surgir unas preguntas como las siguientes:

- ¿Qué preguntas se plantean y por qué?
- ¿Cómo ha llegado el sistema a soluciones intermedias?
- ¿Qué cualidades tienen los distintos objetos?

A pesar de insistir sobre la importancia del componente explicativo es muy difícil y hasta ahora no se han conseguido cumplir todos los requisitos de un buen componente explicativo. Muchos representan el progreso de la consulta al sistema de forma gráfica. Además los componentes explicativos intentan justificar su función rastreando hacia atrás el camino de la solución. Aunque encontrar la forma de representar finalmente en un texto lo suficientemente inteligible las relaciones encontradas depara las mayores dificultades. Los componentes explicativos pueden ser suficientes para el ingeniero del conocimiento, ya que está muy familiarizado con el entorno del procesamiento de datos, y a veces bastan también para el experto; pero para el usuario, que a menudo desconoce las sutilezas del procesamiento de datos, los componentes explicativos existentes son todavía poco satisfactorios.

## La interfase de usuario

En este componente como todos bien sabéis es la forma en la que el sistema se nos presentará ante el usuario. Como en los anteriores nos surgen dudas y preguntas como por ejemplo:

- ¿Cómo debe responder el usuario a las preguntas planteadas?
- ¿Cómo saldrán las respuestas del sistema a las preguntas que se le planteen?
- ¿Qué informaciones se representarán de forma gráfica?

Los requisitos o características de la interfase que presentaremos al usuario voy a resumirlas en cuatro, que a mi opinión son las más importantes y las más a tener en cuenta al desarrollar el sistema:

1. **El aprendizaje del manejo debe ser rápido.** El usuario no debe dedicar mucho tiempo al manejo del sistema, debe ser intuitivo, fácil en su manejo. No debemos olvidar que nuestro sistema simula al comportamiento de un experto. Debe sernos cómodo y relativamente sencillo en cuanto al manejo.
2. **Debe evitarse en lo posible la entrada de datos errónea.** Ejemplo: Poneros en la situación de que nuestro sistema a un médico. Cuando nosotros acudimos a un médico, le contamos y detallamos nuestros síntomas y el con sus preguntas junto con nuestras respuestas nos diagnostica nuestra "enfermedad". Imaginaros que acudimos a un medico y le decimos que nos duele una pierna en lugar de un brazo, el diagnostico será inútil. El ejemplo es muy exagerado pero demuestra la importancia en la correcta introducción de los datos al sistema.
3. **Los resultados deben presentarse en una forma clara para el usuario.** Vuelvo al ejemplo del médico. Si nuestro médico nos diagnostica un medicamento pero en nuestra receta no nos escribe cada cuantas horas hemos de tomarlo por ejemplo, por muy bueno que sea el medicamento, la solución a nuestro problema será ineficiente por completo. Por eso se insiste en que los resultados debe ser claros y concisos.
4. **Las preguntas y explicaciones deben ser comprensibles.**

\*\* Con estas cuatro reglas crearemos nuestro interfase con grandes posibilidades de éxito\*\*

## El componente de adquisición

Un buen componente de adquisición ayudará considerablemente la labor del Ingeniero del Conocimiento. Este puede concentrarse principalmente en la estructuración del conocimiento sin tener que dedicar tanto tiempo en la actividad de programación. Como hice en el campo de la interfase, daremos unas reglas o requisitos para la realización de nuestro componente de adquisición.

### Requisitos o características del componente de adquisición:

1. El conocimiento, es decir, las reglas, los hechos, las relaciones entre los hechos, etc., debe poder introducirse de la forma más sencilla posible.
2. Posibilidades de representación clara de todas las informaciones contenidas en una base de conocimientos.
3. Comprobación automática de la sintaxis.
4. Posibilidad constante de acceso al lenguaje de programación.

Como se pone en práctica cada uno de los requisitos dependerá del lenguaje de programación elegido y del hardware que tengamos. El experto deberá estar algo familiarizado con el componente de adquisición para poder realizar modificaciones por sí sólo.

# Representación de la Base de Conocimiento en un Sistema Experto

## 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.

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

Hoy es Viernes  
Ayer llovió  
Hace frío

La lógica proposicional, permite la asignación de un valor verdadero o falso para la sentencia completa, no tiene facilidad par analizar las palabras individuales que componen la sentencia. Por este motivo, la representación de las sentencias del ejemplo, como proposiciones, sería:

hoy\_es\_Viernes  
ayer\_llovió  
hace\_frío

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

hoy\_es\_Viernes y hace\_frío.

A la proposición anterior dada como ejemplo, se la denomina fórmula bien formada (well-formed formula, wff). Una fórmula bien formada puede ser una proposición simple o compuesta que tiene sentido completo y cuyo valor de veracidad, puede ser determinado. La lógica proposicional proporciona un mecanismo para asignar valores de veracidad a la proposición compuesta, basado en los valores de veracidad de las proposiciones simples y en la naturaleza de los conectores lógicos involucrados.

## La lógica proposicional

Permite expresar y razonar con declaraciones que son o verdaderas o falsas

e.g.,  
el Tec es mejor que el Itam  
lógica es fácil

Este tipo de declaraciones se llaman *proposiciones* y se denotan en lógica proposicional con letras mayúsculas (e.g.,  $P, Q, \dots$ )

$P$ s y  $Q$ s también se llaman *proposiciones atómicas* o *átomos*

Los átomos se pueden combinar con *conectores lógicos* (dando proposiciones compuestas)

negación:  $\sim, \neg$

conjunción:  $\&, \wedge$

disyunción:  $\vee$

implicación:  $\supset, \rightarrow$

doble implicación:  $\leftrightarrow$

e.g.,  
 $G$  = 'Lógica proposicional es fácil'  
 $D$  = 'me estoy aburriendo'

$G \wedge D$  = 'Lógica proposicional es fácil' y 'me estoy aburriendo'

Solo algunas combinaciones de átomos y conectores son permitidas: *formulas bien formadas* (*wff*)

Una *wff* en lógica proposicional es una expresión que puede ser de la siguiente forma:

1. un átomo es un *wff*
2. Si  $F$  es *wff* entonces  $\neg F$  también es
3. Si  $F$  y  $G$  son *wff* entonces:  $F \wedge G, F \vee G, F \rightarrow G$  y  $F \leftrightarrow G$  son *wff*
4. ninguna otra formula es *wff*

Prioridad:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

*wff* es solo sintaxis, no dice si la fórmula es verdadera o falsa (i.e., no dice nada de su semántica)

El significado de una fórmula proposicional se puede expresar por medio de un función:

$$w : prop \rightarrow \{true, false\}$$

La función  $w$  es una *función de interpretación* que satisface:

$F$	$G$	$\neg F$	$F \wedge G$	$F \vee G$	$F \rightarrow G$	$F \leftrightarrow G$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Si  $w$  es una interpretación que asigna a una fórmula dada, el valor de verdad *true*, Entonces  $w$  se dice ser un *modelo* de  $F$

Para  $N$  átomos hay  $2^N$  combinaciones para formar una tabla de verdad

Una fórmula se dice *válida* si es verdadera bajo cualquier interpretación (tautología)

Una fórmula es *inválida* si no es válida

Una fórmula es *insatisfascible* o *inconsistente* si es falsa bajo cualquier interpretación (contradicción) Else es *satisfascible* o *consistente*

Una fórmula es válida cuando su negación es insatisfascible y viceversa

Válido		inválido	
siempre cierto	a veces T o F	siempre falso	
satisfacible		insatisfacible	

Dos fórmulas  $F$  y  $G$  son equivalentes ( $F \equiv G$ ) si los valores de verdad de  $F$  y  $G$  son iguales bajo cualquier interpretación

Leyes de equivalencias:

$$\neg(\neg F) \equiv F$$

$$F \vee G \equiv G \vee F$$

$$F \wedge G \equiv G \wedge F$$

$$(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$$

$$(F \vee G) \vee H \equiv F \vee (G \vee H)$$

$$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$$

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$$

$$F \leftrightarrow G \equiv F \rightarrow G \wedge G \rightarrow F$$

$$F \rightarrow G \equiv \neg F \vee G$$

$$\neg(F \wedge G) \equiv \neg F \vee \neg G$$

$$\neg(F \vee G) \equiv \neg F \wedge \neg G$$

Una fórmula  $G$  se dice que es una *consecuencia lógica* de un conjunto de fórmulas

$F = \{F_1, \dots, F_n\}, N \geq 1$ , denotado por  $F \models G$  si para cada interpretación  $w$  para la cual  $w(F_1 \wedge F_2 \wedge \dots \wedge F_n) = true$ , entonces  $w(G) = true$

$$F \equiv G \text{ si } F \models G \text{ y } G \models F \text{ o } \models (F \leftrightarrow G)$$

Satisfacibilidad, valides, equivalencia y consecuencia lógica son nociones semánticas (generalmente establecidas por medio de tablas de verdad)

Para derivar consecuencias lógicas también se pueden hacer por medio de operaciones exclusivamente sintácticas (e.g., *modus ponens*, *modus tollens*).

Sistemas que tienen operaciones sintácticas se llaman *sistemas de deducción (formales)*, e.g., un sistema axiomático con un lenguaje formal (e.g., lógica proposicional) un conjunto de reglas de inferencia (las operaciones sintácticas) y un conjunto de axiomas.

## Redes Semánticas

Este método de representación del conocimiento esta basado en las relaciones entre objetos. Los nodos de una red semántica corresponden a los objetos y los arcos describen las relaciones entre los objetos. Así puede tomarse un arco con sus dos nodos como una sola unidad de conocimiento.

Pero en la red semántica no esta contenida la información sobre el procesamiento de la red. Las reglas de inferencia deben estar expresadas de forma explícita.

Una red semántica ofrece una buena visión general sobre las relaciones y dependencias de un área de conocimiento y es muy apropiada para la estructuración y verificación del conocimiento por el experto. Los enunciados de las relaciones deben ser expresados fuera de la red. Estas relaciones pueden ser uní o bidireccionales

Una red semántica consiste de un conjunto de nodos los cuales representan objetos, conceptos o situaciones y enlaces que representan las relaciones existentes entre los nodos.

**Ejemplo:** Relación semántica para representar la relación entre una silla y su propietario.

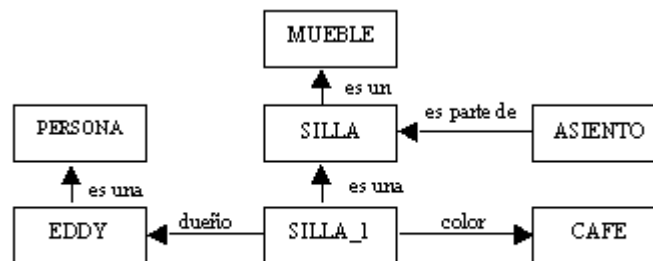


Fig. 2.6. Red semántica propietario de una silla

A las redes semánticas también se le denomina redes asociativas, debido principalmente al carácter asociativo de los enlaces.

### **Propiedades de las Redes Semánticas**

*Las redes Semánticas poseen la propiedad de la Herencia; en la que unos nodos heredan las propiedades o atributos de Nodos de una clase mayor.*

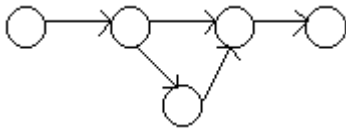
**Nota.**

*A través de la característica de la herencia, las Redes Semánticas tienen la capacidad de inferir conocimiento.*

## *Estructura de una Red Semántica*

### *Def1. Red Semántica*

- *Una Red Semántica es un conjunto de Nodos y Arcos.*
- *Una Red Semántica es una representación grafica del conocimiento*
- *Una Red Semántica es una representación grafica del conocimiento en la que existe una jerarquía de Nodos.*



### *Def. Nodo.*

*Un Nodo es identificado por un objeto*

*Nodo = objeto*

*Donde un Objeto puede ser representado por:*

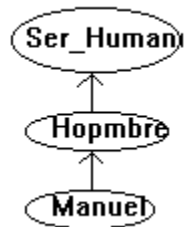
- 1. Personas*
- 2. Animales*
- 3. Eventos*
- 5. Acciones*
- 6. Conceptos*
- 7. Atributos o características que identifican a un objeto.*

**Nota.**

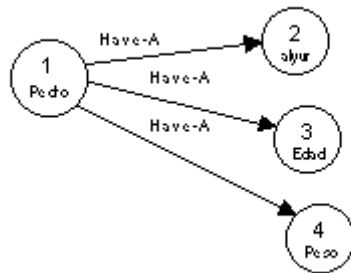
Los Nodos de un Red Semántica están unidos por arcos, los cuales indican la relación que existe entre ellos.

Existen ciertos tipos de Arcos típicos entre los cuales están.

1. ISA (Es-un) el cual usa para identificar que un cierto pertenece a una clase mayor de objeto.



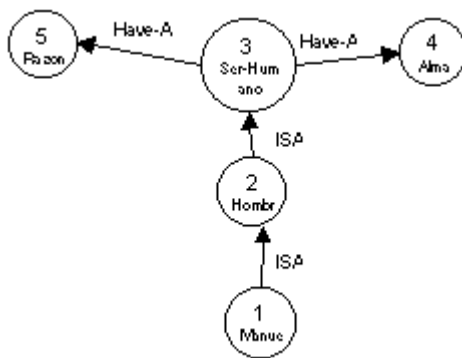
2. HAVE-a (tiene-un), este tipo de arcos se utiliza para identificar que un cierto nodo tiene o pase una cierta características o atributo o propiedad.



### *Propiedades de las Redes Semánticas*

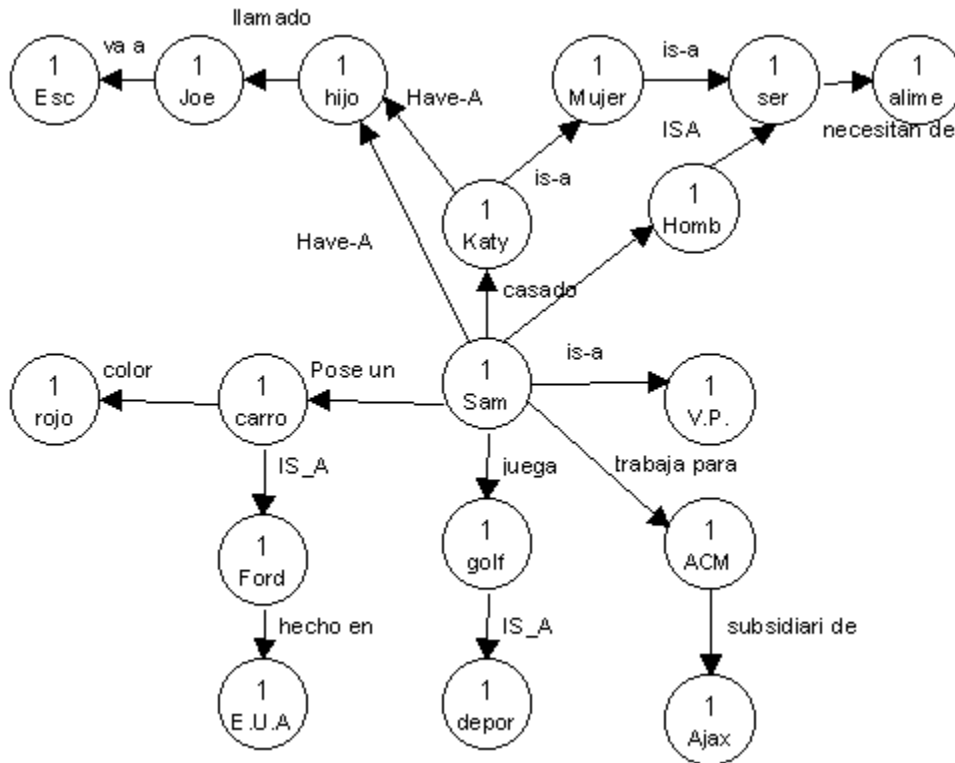
*Las redes Semánticas poseen la propiedad de la Herencia; en la que unos nodos heredaron las propiedades o atributos de Nodos de una clase mayor.*

### *Ejemplo de Redes Semánticas.*



### *Nota.*

*A través de la característica de la herencia, las Redes Semánticas tienen la capacidad de inferir conocimiento.*



Considera la siguiente Red Semántica acerca de SAM y su familia.

1. Necesita Sam Alimento

Si, por la herencia que recibe de los seres humanos.

2. Trabaja SAM para AJAX

Si por característica o por propiedades de herencia ya que CME

Pertenece AJAX.

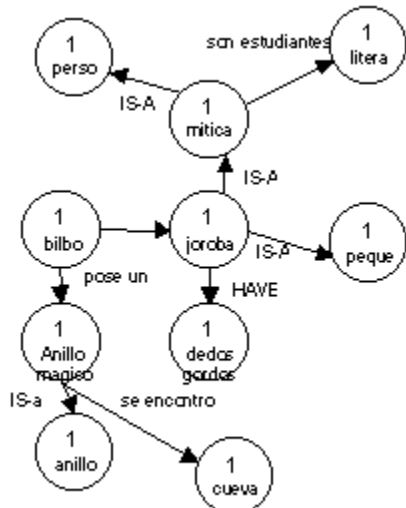
**Nota.**

En Redes Semánticas las inferencias de información o respuestas o preguntas se hacen en base a las propiedades de herencia que existen entre los Nodos.

**Ejemplo.**

Dado el siguiente conocimiento construya la correspondiente Red Semántica.

- Los Jorobados son personas pequeñas.
- Bilbo es un jorobado.
- Los jorobados tienen dedos gordos.
- Bilbo posee un anillo mágico.
- El anillo fue encontrado en una cueva.
- Los jorobados son personas míticas y las personas míticas son estudiadas por los estudiantes de literatura.



Que posee Bilbo

Un anillo mágico.

## *Frames ( Marcos o Prototipos )*

Es una forma de organizar la información referente a un objeto, donde se describen las propiedades del objeto en forma similar a un registro de una base de datos (de hecho lo es), donde se reservan los campos, para cada propiedad.

Si además esta estructura es usada en una red semántica, los campos no redefinidos de un nodo hijo son heredados del nodo padre, de esta manera es muy fácil y natural manejar las excepciones

### **Prototipos o Marcos**

Esta representación surgió inicialmente de la idea de representar el conocimiento en forma de "estructuras" que son accedidas al encontrar un problema, y fue orientada al problema de reconocimiento en visión [Minsky 75].

Minsky define a un frame (marco, prototipo) como una "estructura de datos para representar una situación estereotípica".

Podemos pensar de dichas estructuras como una especie de "record" que tiene una serie de campos (slots) a los que se pueden asignar una serie de valores (fillers). Ciertos campos tienen un valor fijo, y otros se les asignan valores dependiendo de la situación particular (campos o nodos terminales)

La idea es que al encontrarnos en una nueva situación buscamos el marco más cercano y llenamos los nodos terminales con los atributos de dicho problema:

### ***Frames***

*Un Frames es una estructura de datos que contiene todo el conocimiento acerca de un objeto.*

*Los Frames son organizados en forma jerárquica y esta característica es utilizada como medio para responder a Queris (Preguntas).*

*Los Frames surgieron de la pregunta orientada a objetos. Cada frames describe un objeto en particular.*

*Por lo tanto cada frames me representa una estructura de conocimiento en la cual se esta describiendo a un objeto, un evento o una situación ó cualesquier otro elemento.*

*En los frames se describe el conocimiento acerca de un objeto utilizado SLOTS (Ranuras).*

## **Tipos de SLOTS en los Frames**

### **1. Hay SLOt para describir Conocimiento Declarativo como lo es:**

- Peso de un Objeto.*
- La Altura*
- La forma*
- El color*
- Hobbies (leer, nadar, trotar)*

**2. Hay Slots para describir conocimiento Procedural** (*el cual se refiere a pequeños procesos que me identifica a ciertas funciones. (Ejem. Velocidad, aceleración, trabajo, calculo de energía, etc.).*)

### **3. Que identifican a Reglas.**

*Ejem. If maquina caliente THEN prender ventilador.*

### **4. Un Slot puede indicar la conexión con otro FRAME**

**5. UN Slot puede indicar la conexión con otros frames de representación de conocimiento como lo son con las Redes Semánticas.**

## **FRAME**

*Es un bloque grande de conocimiento acerca de un objeto.*

*FRAME de un Auto.*

- Clase; transportación*
- Nombre del fabricante; Audi*
- Origen del fabricante*

*Nota.*

*Una de las características de los frames es que un frame puede identificar a toda una familia de objetos.*

## Esquemas de representación de conocimiento formalizados

### *Esquemas de representación de conocimiento.*

Las representaciones de conocimiento se pueden clasificar en dos grandes clases: *Declarativa* y *Procedimental*. Los esquemas declarativos hacen énfasis en la representación de conocimiento como una acumulación de hechos estáticos junto con una información limitada que describe como se va a emplear el conocimiento. Los esquemas procedimentales enfatizan la representación del conocimiento en forma de reglas dinámicas que describen procedimientos para usar el conocimiento, con poca memoria y directamente como hechos.

Los responsables de los primeros esquemas de representación formalizados fueron Quillian (1968) y Papero & Wodmansee (1971). Los elementos básicos que invariablemente encontramos en todos los esquemas de redes, comúnmente denominados *redes semánticas*, son:

- Estructuras de datos en nodos, que representan conceptos, unidas por arcos que representan las relaciones entre los conceptos.
- Un conjunto de procedimientos de inferencia que operan sobre las estructuras de datos.

Básicamente, podemos distinguir tres tipos tradicionales de redes semánticas:

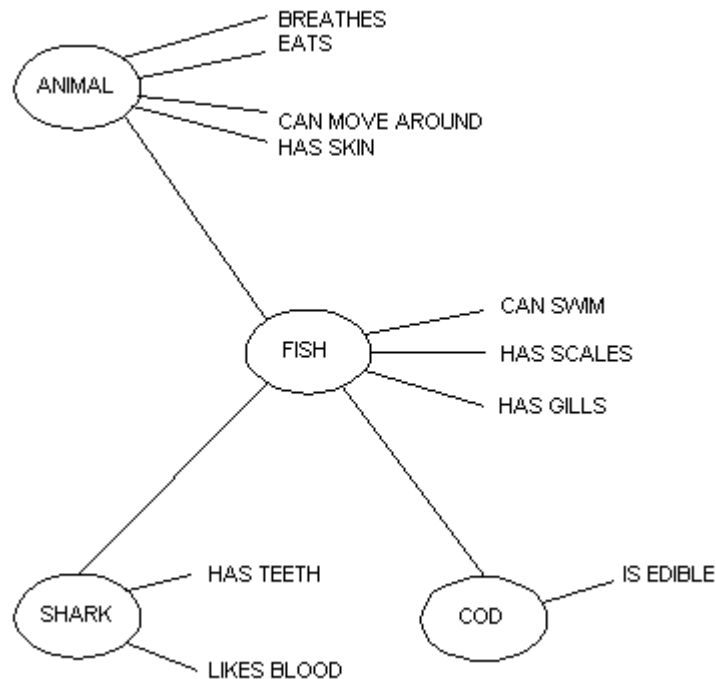
- **Redes IS-A**, en las que los enlaces entre nodos están etiquetados.
- **Grafos conceptuales**: en los que existen dos tipos de nodos: de conceptos y de relaciones.
- **Redes de marcos (*frames*)**: en los que los puntos de unión de los enlaces son parte de la etiqueta del nodo.

En general, cuando se habla de redes semánticas se suele hacer referencia a uno de estos esquemas, normalmente a las redes IS-A o a los esquemas basados en marcos, que comparten ciertas características fundamentales, mientras que los grafos conceptuales presentan diferencias importantes. De entre estas características compartidas destacamos la *herencia por defecto* (*default inheritance*). En una red semántica, los conceptos (estructuras, clases, marcos, dependiendo del esquema concreto) están organizados en una red en la que existe un nodo superior (*top*: T) al que se le asigna uno o varios nodos hijos, que a su vez tienen otros conceptos hijos y así sucesivamente hasta que se alcanza el final (*bottom*), cuyos nodos pueden ser o bien conceptos, o bien instancias.

De los tres tipos de redes semánticas, el esquema basado en marcos es el que permite una mayor flexibilidad, y el que ha recibido mayor atención por parte de los investigadores de ciencia cognitiva y lingüística. Los grafos conceptuales también siguen recibiendo en la actualidad mucha atención por parte de algunos investigadores, aunque cuentan con muchos detractores.

Sin duda el tipo de red semántica por excelencia es el de redes IS-A; de hecho, muchas veces se menciona este tipo como sinónimo de red semántica, y los restantes tipos también incorporan este tipo de enlaces o arcos (links). Una red IS-A es una jerarquía taxonómica cuya espina dorsal está constituida por un sistema de enlaces de herencia entre los objetos o conceptos de representación, conocidos como nodos. Las redes IS-A son el resultado de la observación de que gran parte del conocimiento humano se basa en la adscripción de un subconjunto de elementos como parte de otro más general. Las taxonomías clásicas naturales son un buen ejemplo: un perro es un cánido, un cánido es un mamífero, un mamífero es un animal.

La representación mediante la lógica de predicados es suficiente para expresar las relaciones entre los elementos, pero la estructuración jerárquica facilita que la adscripción de propiedades a una determinada categoría se reduzca a aquellas que son específicas a la misma, heredando aquellas propiedades de las categorías superiores de la jerarquía, tradicionalmente de una forma monotónica, es decir sin excepciones. El siguiente ejemplo de red IS-A (Figura 16), tomado de (Brachman 1983: 31) ejemplifica una red semántica típica con herencia de propiedades.



**Figura 16:** Ejemplo de red IS-A

El concepto de herencia es fundamental para entender el funcionamiento de las redes semánticas. Siguiendo a Shastri (1988), definimos la herencia como el sistema de razonamiento que lleva a un

agente a deducir propiedades de un concepto basándose en las propiedades de conceptos más altos en la jerarquía.

Como Brachman (1983) recuerda, los nodos de las estructuras IS-A se han usado para representar muchas cosas, pero la división más importante es la interpretación *genérica* o *específica* de los nodos, es decir, si éstos representan un sólo individuo o varios. Los nodos situados en lo más bajo de la jerarquía y que denotan individuos concretos o instancias, son llamados *tokens*, mientras que los nodos superiores, que denotan clases de individuos son considerados *types*.

Puesto que en una misma jerarquía podemos obtener nodos de ambos tipos, se debe hacer explícita una distinción de los tipos de enlaces. Por un lado existen enlaces que conectan categorías (genéricas) con otras categorías, y por otro, enlaces entre categorías e individuos. Las primeras pueden expresar relaciones como por ejemplo: *subconjunto/ súper conjunto, generalización/ especificación, AKO*, es decir, a kind of, contenido conceptual, restricción de valores y tipo característico del conjunto. Las relaciones genérico/individuales también pueden ser de varios tipos: *pertenencia al conjunto, predicación, contenido conceptual y abstracción*.

Las redes IS-A son muy flexibles, pero los investigadores de IA han puesto de manifiesto algunos problemas y desventajas importantes, entre los que cabe destacar los siguientes:

1. La elección de los nodos y arcos es crucial en la fase de análisis. Una vez se ha decidido una estructura determinada, es muy complicado cambiarla.
2. Dificultad para expresar cuantificación. Por ejemplo en expresiones tales como "algunos pájaros vuelan" o "todos los pájaros pían".
3. Del mismo modo, las redes semánticas presentan grandes dificultades para representar la dimensión intencional. Por ejemplo en proposiciones tales como "Pedro cree que Ana sabe conducir".

Esto llevó a idear otros esquemas de representación con una estructura más compleja que simples nodos y arcos, que fuesen capaces de dar cabida a éstas y otras situaciones. Concretamente, John Sowa (1984) propuso los grafos conceptuales, aunque el esquema de representación basado en marcos (*frames*) es el que ha tenido mayor aceptación.

Además de las redes ISA y los grafos conceptuales existe otro esquema de representación del conocimiento muy usado en los últimos años: los esquemas de marcos. El primer proponente de la noción de marco (*frame*) fue el psicólogo cognitivo Marvin Minsky, quien resume su esencia de este modo:

Here is the essence of frame theory: When one encounters a new situation (or makes a substantial change in one's view of a problem), one selects from memory a structure called a *frame*. This is a remembered framework to be adapted to fit reality by changing details as necessary. Minsky (1975: 211)

La fundamentación psicológica de los marcos es parecida a la de los scripts de Schank (Schank et al. 1977): cuando nos enfrentamos con una situación determinada, intentamos ajustarla a otra

parecida de la que ya tenemos experiencia previa y esperamos que aparezcan un número de elementos comunes y se sucedan algunas situaciones. Por ejemplo, si entramos en una habitación de hotel, esperamos encontrar una cama, un armario, un baño, etc. Nuestra mente reconocerá las instancias específicas de la nueva habitación y los acomodará al estereotipo que ya poseemos. La base de la teoría la conforman, por tanto, las situaciones estereotipadas.

Informalmente, un marco es una estructura de datos compleja que representa una situación estereotipada, por ejemplo hacer una visita a un enfermo o acudir a una fiesta de cumpleaños. Cada marco posee un número de casillas (*slots*) donde se almacena la información respecto a su uso y a lo que se espera que ocurra a continuación. Al igual que las redes semánticas, podemos concebir un marco como una red de *nodos* y relaciones entre nodos (*arcos*). Una base de conocimiento basada en marcos es una colección de marcos organizados jerárquicamente, según un número de criterios estrictos y otros principios más o menos imprecisos tales como el de similitud entre marcos. A nivel práctico, podemos considerar los marcos como una red semántica con un número de posibilidades mucho mayor, entre las que destacan especialmente, la capacidad de activación de procesos (*triggering*) y de *herencia no-monotónica* mediante *sobre control (overriding)*, en la que un nodo hijo hereda todos los *slots* de su padre a menos que se especifique lo contrario.

Como es evidente, la representación de conocimiento basada en marcos debe mucho a las redes semánticas. Sin embargo, los esquemas de representación basados en marcos insisten en una organización jerárquica de éstos, mientras que las redes semánticas no requieren tal organización. La estructura de nodos de los marcos es también mucho más rica que la de las redes semánticas, conteniendo sistemas de activación de procesos (*triggering*). Esto ocurre cuando en lugar de llenar un slot con un valor determinado, se indica un procedimiento que será el encargado de devolver un determinado valor.

Podemos resumir las principales características de los marcos en las siguientes (Minsky 1975; Winograd 1975):

- **Precisión (*Explicitness*):** los objetos, las relaciones entre objetos y sus propiedades se describen de forma precisa; en ausencia de evidencia contraria se usan valores por omisión.
- **Activación dinámica de procesos (*Triggering*):** es posible adjuntar procedimientos a un marco o alguno de sus componentes de forma que se llamen y ejecuten automáticamente tras la comprobación de cambio de alguna propiedad o valor (p. ej. IF-NEEDED, IF-ADDED).
- **Herencia por defecto no-monotónica:** los marcos están conceptualmente relacionados, permitiendo que los atributos de los objetos sean heredados de otros objetos predecesores en la jerarquía.
- **Modularidad:** la base de conocimiento está organizada en componentes claramente diferenciados.

El potencial de estas características puede verse, por ejemplo, en los mecanismos de razonamiento que son capaces de llevar a cabo: el razonamiento en un esquema basado en marcos se lleva a cabo mediante dos mecanismos básicos: el reconocimiento (*recognition* o

paterna-matching) y la herencia. En el entorno de los marcos el proceso de reconocimiento de patrones se centra en encontrar el lugar más apropiado para un nuevo marco dentro de la jerarquía de marcos. Esto requiere que el mecanismo de reconocimiento sea capaz de recibir información sobre la situación existente (en forma de marco) y lleve a cabo una búsqueda del marco más adecuado de entre todos contenidos en la base de conocimiento.

Desde la formulación teórica de Minsky han existido diversas implementaciones de este sistema representacional, entre los que cabe destacar los ya considerados clásicos KRL, OWL y, sobre todo, KL-ONE (Brachman & Schmolze 1985). Este último es sin duda el más relevante, en cuanto que a partir de él han surgido numerosos sistemas de representación del conocimiento que se están desarrollando o usando hoy en día. De hecho, se suele hacer referencia a los lenguajes de programación /representación basados en marcos mediante el calificativo "al estilo de KL-ONE". KL-ONE es el producto de muchos años de investigación de algunos de los que se pueden considerar como padres de la representación del conocimiento, entre los que destacan Ronald Brachman, Rusty Bobrow, Hector Leves qué, Jim Schmolze o William Woods.

Como es habitual en las implementaciones de marcos, una taxonomía de KL-ONE comprende conceptos, con un número de propiedades atribuidas y una serie de relaciones entre conceptos. Más concretamente, un nodo (concepto) de KL-ONE consiste en un conjunto de *roles* (generalizaciones de las nociones de atributo, parte, constituyente, rasgo, etc.) y un conjunto de condiciones estructurales que expresan relaciones entre esos roles. Los conceptos están enlazados a conceptos más generales en una relación especial llamada SUPERC. El concepto más general en la taxonomía es denominado *superconcepto*, y *subsume* (es más general que) todos los demás.

De lo que no cabe ninguna duda es que el sistema de representación basado en marcos es el que ha contado con mayor aceptación ya que es enormemente flexible e impone un número muy reducido de restricciones en lo que respecta a su implementación. Las modernas ontologías de conceptos (véase § 5.3) se basan en mayor o menor medida en un modelo de marcos, aunque, en general, no emplean sistemas de activación de procesos por tratarse de representaciones con un marcado carácter estático, enfocadas a surtir de información a otras aplicaciones que contienen los procedimientos.

Cabe preguntarse si en realidad es necesario emplear un complejo sistema de representación del conocimiento para la gestión terminológica. En la práctica, sólo es indispensable un sistema que permita la creación de jerarquías conceptuales, la asignación de atributos y características a los conceptos y la explicitación de relaciones entre conceptos, garantizando la coherencia interna del conjunto de conceptos: por ejemplo, que no existan conceptos repetidos, se anule la posibilidad de crear estructuras circulares, además, por supuesto, de permitir la asignación de términos a estos conceptos.

Después de estudiar los esquemas de representación más destacados de los que podemos servirnos para la representación del conocimiento especializado, la opción adoptada en ese sentido ha sido bastante práctica: pensamos que sería adecuado contar con un sistema de representación conceptual compatible con sistemas de representación más complejos con los que

acabamos de describir, pero que no presentara una complejidad inherente que, en vez de facilitar, dificultara el trabajo terminográfico.

Lo ideal sería contar con un sistema de representación conceptual que impusiese muy pocas restricciones en cuanto al modo de trabajo básico del terminólogo moderno, facilitándole la tarea en cuanto a organización conceptual se refiere y, al mismo tiempo, estuviese basado en esquema conceptual válido y aceptado como sistema de representación del conocimiento. Evidentemente, un sistema así abre un número de posibilidades que van más allá del trabajo terminográfico, permitiendo su integración con sistemas computacionales de Inteligencia Artificial en general, sistemas expertos, Traducción Automática, etc.

Además, debemos pensar que un sistema de estas características nos ofrece un valor añadido considerable puesto que, aunque no saquemos partido de características procedimentales avanzadas, tales como el reconocimiento de patrones, sí que nos puede ser muy útil la herencia de valores por defecto. Ésta, además de ahorrar trabajo y tiempo, nos ayuda a validar una estructura conceptual determinada: si, por ejemplo, un número considerable de valores heredados han de ser sobrecontrolados (negados), entonces puede ser que nuestra jerarquía conceptual sea errónea.

Una herramienta de este tipo permitiría al terminógrafo comprobar que la estructura conceptual del ámbito de especialidad cuya terminología ha de sistematizar se ajusta a sus necesidades y es lo suficientemente completa y coherente. Por supuesto que la adopción de tal sistema va a suponer una serie de restricciones e imposiciones metodológicas, ya que, por ejemplo, el sistema no nos va a permitir introducir un término sin antes especificar el concepto al que se debe asignar y para introducir un concepto va a ser necesario especificar el lugar que éste ocupa en la jerarquía (de

qué concepto es hijo). Sin embargo, pensamos que las ventajas que se obtienen compensan con creces este trabajo adicional ya que este tipo de sistema nos va a imponer un método de trabajo estricto que va redundar en una base terminológica altamente estructurada y con todas las garantías en cuanto a coherencia interna.

## Cuantificadores

### *Lógica de predicados: cuantificación*

El cuantificador universal  $\forall x$  se utiliza para aseverar que una fórmula es verdadera para todos los valores de la variable asociada.

El cuantificador existencial  $\exists x$  se emplea para aseverar que por lo menos existe alguna asignación para  $x$ , que hará que la fórmula asociada sea

### *Cuantificadores:*

Para los casos en los que no se puede especificar el número de caracteres que incluye una coincidencia, las expresiones regulares admiten el concepto de cuantificadores.

Con los cuantificadores se puede especificar el número de veces que debe aparecer un elemento dado de una expresión regular para que se cumpla una coincidencia.

### *La tabla siguiente muestra los distintos cuantificadores y su significado:*

Carácter Descripción

\* Coincide cero o más veces con el carácter o subexpresión anterior. Por ejemplo,  $zo^*$  coincide con  $z$  y con  $zoo$ .  $*$  equivale a  $\{0,\}$ .

+ Coincide una o más veces con el carácter o subexpresión anterior. Por ejemplo,  $zo^+$  coincide con  $zo$  y  $zoo$ , pero no con  $z$ .  $+$  equivale a  $\{1,\}$ .

? Coincide una vez o ninguna con el carácter o subexpresión anterior. Por ejemplo,  $da(do)?$  coincide con  $da$  en  $da$  o en  $dao$ . ? equivale a  $\{0,1\}$

$\{n\}$   $n$  es un entero no negativo.

Coincide exactamente  $n$  veces. Por ejemplo,  $o\{2\}$  no coincide con la  $o$  de  $doy$  pero coincide con las dos  $oes$  en  $cooperar$ .

$\{n,\}$   $n$  es un entero no negativo.

Coincide como mínimo  $n$  veces. Por ejemplo,  $o\{2,\}$  no coincide con la  $o$  de  $sol$  y coincide con todas las  $oes$  en  $nooooo$ .  $o\{1,\}$  equivale a  $o^+$ .  $o\{0,\}$  equivale a  $o^*$ .

$\{n,m\}$  m y n son enteros no negativos, donde  $n \leq m$ . Coincide n veces como mínimo y m como máximo.

Por ejemplo,  $o\{1,3\}$  coincide con las tres primeras oes en goooooo.  $o\{0,1\}$  equivale a  $o?$ .

Tenga en cuenta que no se puede incluir un espacio entre la coma y los números.

Como los números de capítulos son fácilmente más de nueve en un documento extenso, se necesita un sistema para tratar números de capítulo de dos o tres dígitos.

Los cuantificadores le permitirán hacerlo.

La siguiente expresión regular coincide con encabezados de capítulo que tengan cualquier número de dígitos:

```
/Chapter [1-9][0-9]*/
```

Observe que el cuantificador aparece después de la expresión del intervalo.

Por tanto, se aplica a toda la expresión del intervalo que, en este caso, especifica sólo dígitos del 0 al 9, ambos incluidos.

En este caso, no se utiliza el cuantificador + porque no es necesario que haya un dígito en la posición segunda o subsiguiente.

Tampoco se utiliza el carácter ? ya que limita los números de capítulos a dos dígitos solamente. Como mínimo debe coincidir un dígito después de Chapter y un carácter de espacio.

Si sabe que el número de capítulos está limitado a 99, puede utilizar la siguiente expresión para especificar como mínimo un dígito, aunque no más de dos.

```
/Chapter [0-9]{1,2}/
```

El inconveniente de la expresión anterior es que en los números de capítulos mayores que 99 sólo coincidirá con los dos primeros dígitos.

Otro inconveniente es que Chapter 0 coincidiría.

A continuación, se incluyen las expresiones más adecuadas para que sólo coincidan dos dígitos:

```
/Chapter [1-9][0-9]?/
```

or

```
/Chapter [1-9][0-9]{0,1}/
```

Los cuantificadores \*, + y ? se conocen como expansivos porque coinciden con la mayor cantidad de texto posible.

Sin embargo, en ocasiones es preferible obtener una coincidencia mínima.

Por ejemplo, supongamos que busca en un documento HTML la aparición del título de un capítulo encerrado en una etiqueta H1.

Este texto aparece en el documento como:

```
<H1>Chapter 1 – Introduction to Regular Expressions</H1>
```

La siguiente expresión coincide con todos los elementos comprendidos entre el símbolo menor que (<) de apertura y el símbolo mayor que (>) que cierra la etiqueta H1.

$\langle . * \rangle$

Si sólo necesita hacer coincidir la etiqueta de apertura H1, la expresión siguiente no expansiva sólo coincide con  $\langle H1 \rangle$ .

$\langle . * ? \rangle$

Si coloca ? después de un cuantificador del tipo \*, + o el signo ?, la expresión se transforma de coincidencia expansiva a no expansiva, o mínima.

**cuantificadores.**

*Las expresiones:*

Todo hombre es mortal.

Algunos hombres son sabios.

Pueden traducirse respectivamente como:

Para todo x, si x es hombre entonces x es mortal.

Existe un x, tal que x es hombre y x es sabio.

**Otros giros utilizados para la expresión "para todo x", son:**

Todo x

Cualquiera x

Cada x

que se simbolizan por "x" y se llama cuantificador universal.

**Otros giros utilizados para la expresión "Existe un x" son:**

Hay x

Existe x, tal que

A

Algunos x

Que se simbolizan por "\$x" y se llama cuantificador existencial.

*Existen tres formas de convertir una función proposicional Px en una proposición a saber:*

- Haciendo la sustitución de las variables por un término específico.
- Anteponiendo la expresión "para todo x" o cuantificador universal.
- Anteponiendo la expresión "existe al menos un x" o cuantificador existencial.

*El enunciado "existe al menos un x tal que Px" se representa como:*

$(\exists x)(Px)$

El enunciado "para todo x, Px" se representa como:

$(\forall x)(Px)$ .

Al anteponer a la función proposicional Px un cuantificador, se dice que la variable x ha pasado a ser una variable ligada.

Una proposición de la forma  $(\forall x)(Px)$  es verdadera cuando todas las sustituciones de la variable x por términos específicos del conjunto de referencia convierten a Px en enunciado verdadero.

Un enunciado de la forma  $(\exists x)(Px)$  es verdadero cuando al menos un caso de sustitución de la variable x por un término específico del conjunto de referencia, convierte a Px en un enunciado verdadero.

Las proposiciones universales pueden aparecer negadas, como en el enunciado: "No todos son mecánicos".

En este caso la simbolización será  $\neg (\forall x)(Mx)$  donde Mx es la función proposicional "x es mecánico" que toma valores dentro del conjunto de referencia formado por los hombres.

Las palabras "ningún", "ninguno", "nada", "nadie" corresponden también a enunciados universales con negaciones, pero de una manera distinta a las proposiciones anteriores.

La proposición "ninguno es mecánico" no equivale a la proposición "no todos son mecánicos" sino a la expresión "para todo x, x no es mecánico" que se simboliza  $(\forall x)(\neg Mx)$ .

Las proposiciones anteriores pueden estar negadas, como por ejemplo "no es cierto que hay fantasmas" la cual se simboliza como  $\neg (\exists x)(Fx)$  donde Fx simboliza la expresión "x es un fantasma".

Análogamente a lo que ocurre con los cuantificadores universales, las proposiciones existenciales puede tener negaciones internas como "algo no es mortal" la cual se simboliza como  $(\exists x) (\neg Mx)$  donde  $Mx$  simboliza la expresión "x es mortal".

## Ejercicios.

*Simbolizar los siguientes enunciados:*

Todo es perecedero.

Hay marcianos.

Alguien no es perfecto.

No hay cosas sólidas.

Si todo es rojo, hay algo rojo.

Nada se mueve.

No todo es perecedero.

Nada es perecedero.

Algunos números negativos no son enteros.

Algunos gobiernos no respetan la libertad.

### Conclusión :

*En* el expuesto trabajo se ha presentado la descripción de uno de los componentes de un Sistema Experto como lo es la Base del Conocimiento y sus principales representaciones.

Además conocer la estructura de un Sistema Experto.