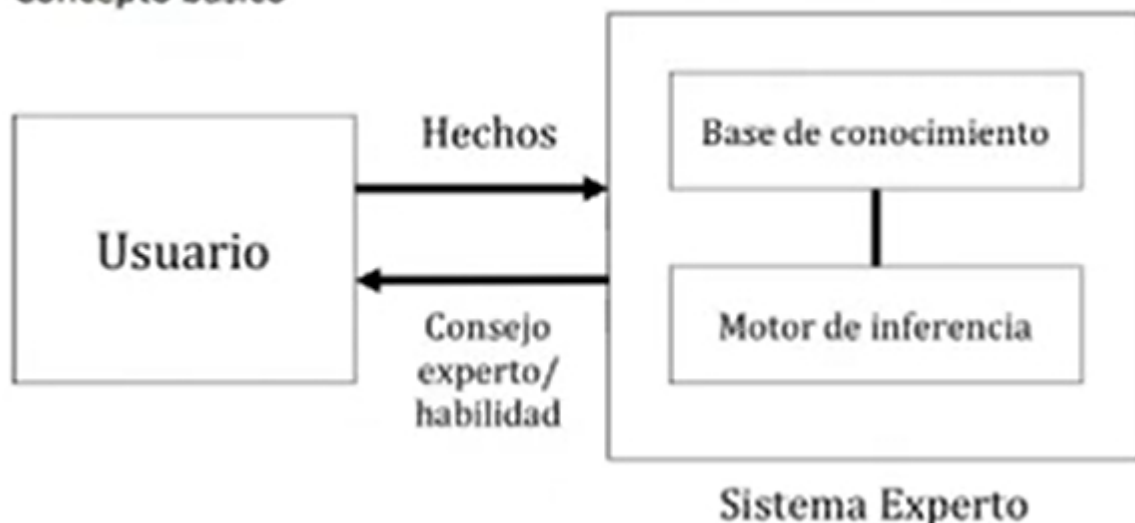


Un sistema experto (SE) es:

"Un programa de computadora inteligente que utiliza el conocimiento y procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles que requieren experiencia humana significativa para sus soluciones."

*Edward Feigenbaum*

### Concepto básico

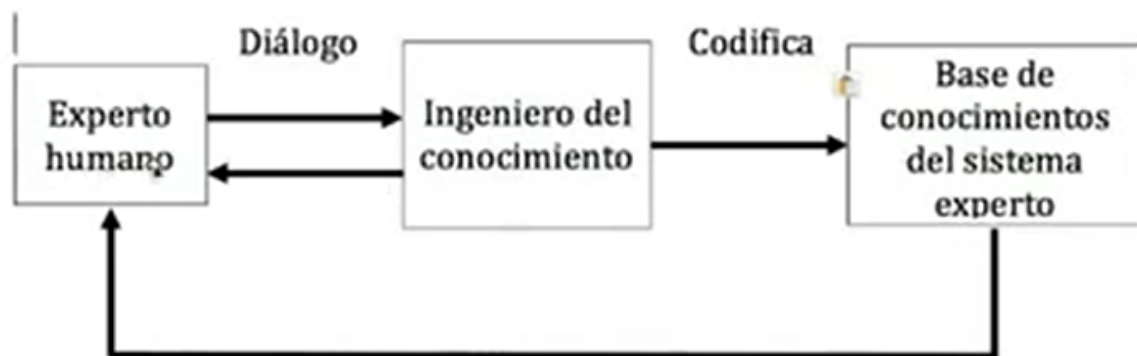


El conocimiento de un experto es específico de un **dominio del problema** (medicina, finanzas, ciencia o ingeniería). El conocimiento experto para resolver un problema específico es llamado **dominio del conocimiento** del experto.

### Ventajas de los Sistemas Expertos

- Incrementa la disponibilidad
- Reduce costos
- Reduce riesgos
- Duración
- Experiencia múltiple
- Incrementa la confiabilidad
- Explicación
- Estable, no-emocional y una respuesta completa todo el tiempo
- Tutor inteligente
- Base de datos inteligente

El proceso para construir un Sistema Experto es llamado **Ingeniería del conocimiento**, la cual adquiere el conocimiento de un humano experto (u otra fuente) y se codifica en el SE.



La expresión **sistema basado en el conocimiento** es un mejor término para las aplicaciones de tecnologías basadas en el conocimiento, las cuales podrían ser usadas para la creación de SE o de sistemas basados en el conocimiento.

La **habilidad de explicación** es una parte integral de los SE sofisticados. De hecho, desarrolla la habilidad de explicación podría permitir al usuario explorar razonamientos hipotéticos o traducir el lenguaje natural en forma de reglas.

Algunos SE permiten al sistema aprender reglas por ejemplo, mediante **reglas de inducción**.

Una limitación práctica de muchos SE es la falta de **conocimiento causal**. Esto es, el SE no tiene que entender las causas y efectos en un sistema. Es más fácil programar un SE con **conocimiento superficial** basado en conocimiento empírico y heurístico que basado en **conocimiento profundo** de las estructuras básicas, funciones y comportamientos de los objetos.

Un tipo de conocimiento superficial es el **conocimiento heurístico** (del griego que significa "descubrir"). La heurística no garantiza el éxito de la misma manera que un algoritmo garantiza la solución a un problema. En lugar de eso, la heurística utiliza reglas empíricas o conocimiento empírico obtenido de la experiencia que puede lograr la solución pero no es garantía de que funcione.

El problema de transferir el conocimiento humano a un SE es tan grande que es llamado el **cuello de botella de la adquisición del conocimiento**.

A pesar de las limitantes, los SE tienen gran éxito tratando problemas del mundo real que convencionalmente han sido imposibles de resolver, especialmente aquellos que tratan con incertidumbres o información incompleta.



---

#### Características de un SE

- Alto rendimiento
- Tiempo de respuesta adecuado
- Buena confiabilidad
- Entendible

- 
- Flexibilidad
  - Enlista todas las razones a favor y en contra de una hipótesis particular
  - Lista todas las hipótesis que pueden explicar la evidencia observada
  - Explica todas las consecuencias de un hipótesis
  - Da un pronóstico o predicción de qué ocurrirá si la hipótesis es verdadera.
  - Justifica las preguntas que el programa hace al usuario para obtener mayor información
  - Justifica el conocimiento del programa

4

1-2-3-4-1
1-2-4-3-1
1-3-2-4-1
1-3-4-2-1
1-4-2-3-1
1-4-3-2-1

Una red neuronal puede resolver el caso para 10 ciudades tan rápido como para 30 ciudades en comparación con una computadora a la que le tomaría mucho más tiempo. Sin embargo las redes neuronales no siempre entregan la respuesta óptima.

Representación del conocimiento

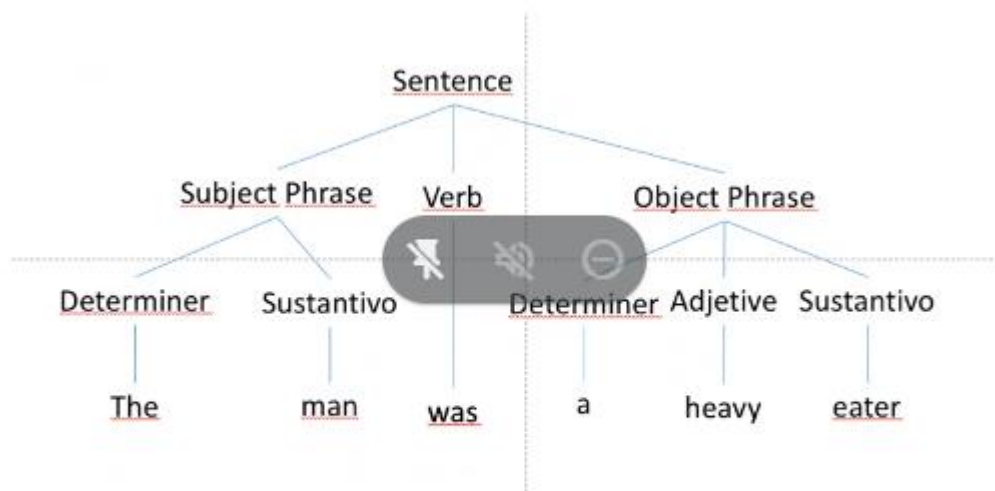
Producción de reglas

Una notación formal para definir reglas de producción es la Forma Backus-Naur BFN. Esta notación es un metalenguaje para definir la sintaxis de un lenguaje.

Una notación formal para definir reglas de producción es la Forma Backus-Naur BFN. Esta notación es un metalenguaje para definir la sintaxis de un lenguaje.

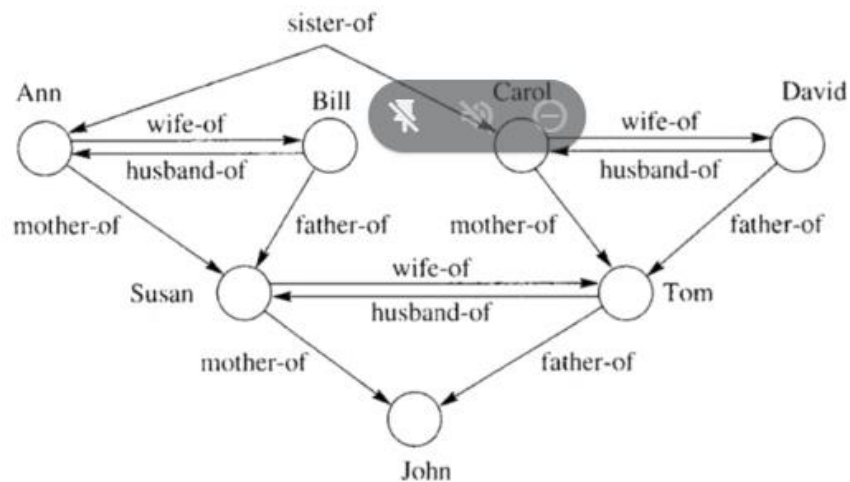
```
<sentence>::=<subject><verb><end-mark>
<subject>::=I|You|We
<verb>::=left|came
<end-mark>::=.,|?|!
```

Una gramática es un conjunto completo de reglas de producción que definen un lenguaje sin ambigüedades.



<sentence>::=<subject phrase><verb><object phrase>  
 <subject phrase>::=<determiner><noun>  
 <determiner>::=the  
 <noun>::=man

## Redes semánticas



(b) A Semantic Net



## Representación del conocimiento (Segunda parte)

La terna objeto-atributo-valor (OAV), puede ser usada para caracterizar todo el conocimiento en una red semántica.

Objeto	Atributo	Valor
manzana	color	rojo
manzana	tipo	gala
manzana	cantidad	100
uvas	color	rojo
uvas	tipo	sin-semilla
uvas	cantidad	500

## Desventajas de las redes semánticas

Desventajas de las redes semánticas

Exploración combinatoria de la busca de nodos.

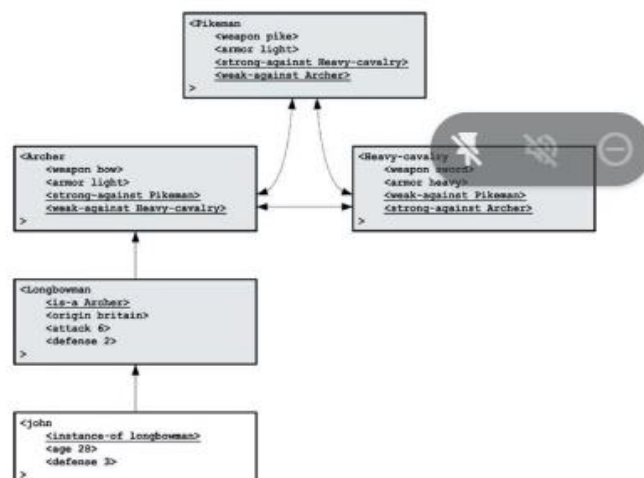
Son heurísticamente inadecuadas

La interpretación de las aristas y nodos. Un nodo llamado silla puede representar una silla en específico, la clase de todas las sillas, el concepto silla, etc.

No pueden definir el conocimiento de manera lógica.

La terna objeto-atributo-valor (OAV), puede ser usada para caracterizar todo el conocimiento en una red semántica.

esquema es el **script**, que es esencialmente una secuencia de frames ordenada respecto al tiempo. Un frame es básicamente un grupo de espacios y campos de relleno que definen el estereotipo de un objeto.



Los sistemas expertos basados en frames son útiles para representar el conocimiento causal porque su información está organizada por causa y efecto.

Los campos de relleno pueden ser valores o un rango de valores.

Mientras los espacios pueden contener procedimientos. Hay tres tipos de procedimientos adjuntos, el if-needed procedimiento que se ejecuta cuando el valor en un campo de relleno es requerido pero no tiene un valor inicial o valor por

Métodos de inferencia:

Deducción: Razonamiento lógico en el cual las conclusiones debe ir seguidas de sus premisas.

Inducción: Inferencia de un caso específico a un caso general.

Intuición: Teoría no probada. La respuesta solo aparece, posiblemente por un patrón de reconocimiento inconsciente.

Heurística: Reglas empíricas basadas en la experiencia.

Generación y prueba: Prueba y error.

Abducción: Razonamiento hacia atrás a partir de una conclusión verdadera a las premisas que podrían haber causado la conclusión.

Por defecto: En ausencia de un conocimiento específico, se asume un conocimiento general o común por defecto. 1

No-monotónica: El conocimiento previo puede estar incorrecto cuando es obtenida nueva evidencia.

Analogía: Inferir una conclusión basada en similitudes con otra situación.



## Lógica deductiva y silogismos

Un argumento lógico es un grupo de sentencias las cuales son justificadas en base a las sentencias previas en el razonamiento en cadena.

Premisa: Cualquiera que pueda programar es inteligente.

Premisa: Juan puede programar.

Conclusión: Por lo tanto, Juan es inteligente.

Los silogismos son útiles porque pueden expresarse en términos de reglas  
SI...ENTONCES.

SI cualquiera que pueda programar es inteligente Y Juan puede programar  
ENTONCES Juan es inteligente

## Reglas de inferencia

Law of Inference	Schemata	
1. Law of Detachment	$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$	
2. Law of the Contrapositive	$\begin{array}{l} p \rightarrow q \\ \hline \therefore \neg q \rightarrow \neg p \end{array}$	
3. Law of Modus Tollens	$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \therefore \neg p \end{array}$	
4. Chain Rule (Law of the Syllogism)	$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$	
5. Law of Disjunctive Inference	$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$	$\begin{array}{l} p \vee q \\ \neg q \\ \hline \therefore p \end{array}$
6. Law of the Double Negation	$\begin{array}{l} \neg(\neg p) \\ \hline \therefore p \end{array}$	
7. De Morgan's Law	$\begin{array}{l} \neg(p \wedge q) \\ \hline \therefore \neg p \vee \neg q \end{array}$	$\begin{array}{l} \neg(p \vee q) \\ \hline \therefore \neg p \wedge \neg q \end{array}$
8. Law of Simplification	$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$	$\begin{array}{l} \neg(p \vee q) \\ \hline \therefore q \end{array}$
9. Law of Conjunction	$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$	

## Modus ponens

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$



Dado que el modus ponens es un caso especial de un silogismo lógico.

En la lógica las mayúsculas representan constantes, mientras que las minúsculas variables. Esta convención es opuesta a PROLOG.

## Limitaciones de la lógica proposicional

Todos los hombres son mortales  
Sócrates es un hombre  
Por lo tanto, Sócrates es mortal



p=Todos los hombres son mortales  
q=Sócrates es un hombre  
r=Sócrates es mortal

p  
q  
∴r



## Lógica de predicados (Lógica de primer orden)

$(\forall x) H(x)$   
 $\therefore H(\text{Socrates})$



donde la proposición  $H(x)$  dice que  $x$  es un humano.

$(\forall x) (H(x) \rightarrow M(x))$   
 $H(s)$   
 $H(s) \rightarrow M(s)$   
 $M(s)$

## Resolución

La resolución es la regla de inferencia primaria en PROLOG.

```
hombre(socrates). %hecho
mortal(X):-hombre(X). %regla
```

```
?- mortal(socrates). %|
```

```
mortal(X)/ mortal(socrates)
X/socrates
```

## Razonamiento causal

Los frames y las redes semánticas son dos útiles modelos del razonamiento causal.

- 2) SI la batería está buena  
ENTONCES hay electricidad.
- 3) SI hay electricidad  
Y las bujías están buenas  
ENTONCES las bujías encenderán
- 4) SI las bujías encienden  
Y hay gasolina  
ENTONCES el motor encenderá
- 5) SI el motor enciende  
Y hay buenas llantas  
ENTONCES el carro se moverá

## Actividad 2

Elija un esquema de representación de conocimiento y realice una representación parcial del conocimiento correspondiente al producto integrador.

Reglas si entonces, red semántica, no darle todas las aves

Si tiene esto, si tiene esto, si tiene esto, entonces es un avebicho

En un sistema ideal, la probabilidad debería ser la misma. Por lo cual el análisis es mucho más sencillo.

La fórmula fundamental de la probabilidad es

$$P=W/N$$

|  
donde W es el número ganados y N es el número de los posibles eventos.

$$P(1)=1/6$$

De igual forma,

$$P(2)=1/6, \dots$$

La probabilidad de perder es de

$$Q=(N-W)/N=1-P$$

## Espacio de muestras

El resultado de una prueba es un punto de muestra y el conjunto de todos los posibles puntos de muestra definen un espacio de muestras.

Un evento es un subconjunto del espacio de muestras.



## Árbol de eventos

### teoría de la probabilidad

axioma 1:  $0 \leq P(E) \leq 1$

axioma 2:  $\sum_i P(E_i) = 1$

Corolario:  $P(E) + P(E') = 1$

axioma 3:  $P(E_1 \cup E_2) = P(E_1) + P(E_2)$  para eventos mutuamente excluyentes.

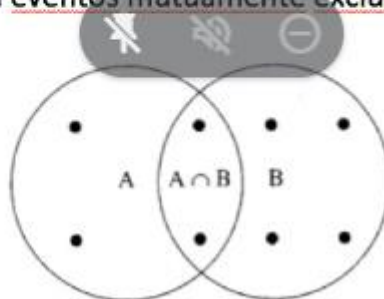
### Probabilidad condicional

#### Ley multiplicativa

#### Ley multiplicativa

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \text{ para } P(B) \neq 0.$$

$P(A \cap B) = P(A)P(B)$  para eventos mutuamente excluyentes.





## Teorema de Bayes

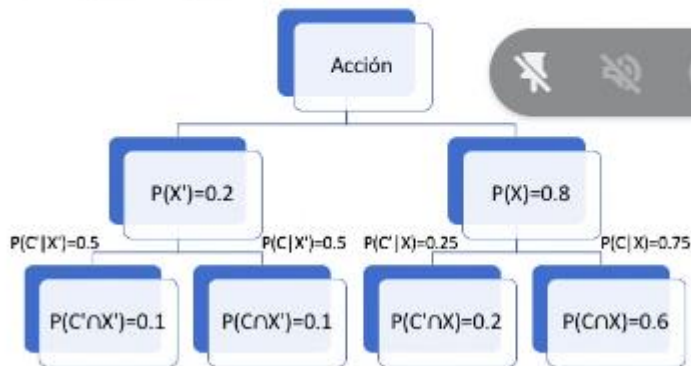
Sea  $\{A_1, A_2, \dots, A_i, \dots, A_n\}$  un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (o). Sea  $B$  un suceso cualquiera del que se conocen las probabilidades condicionales  $P(B|A_i)$ . Entonces, la probabilidad  $P(A_i|B)$  viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$$

donde:

- $P(A_i)$  son las probabilidades a priori,
- $P(B|A_i)$  es la probabilidad de  $B$  en la hipótesis  $A_i$ ,
- $P(A_i|B)$  son las probabilidades a posteriori.

	Brand X	Not Brand X	Total of Rows
Crash C	0.6	0.1	0.7
No Crash C'	0.2	0.1	0.3
Total of Columns	0.8	0.2	1.0



Posteriori  $P(H_i|E) = \frac{P(E \cap H_i)}{\sum_j P(E \cap H_j)}$

$$P(X'|C') = \frac{0.1}{0.1+0.2} = \frac{1}{3} \quad P(X'|C) = \frac{0.1}{0.1+0.6} = \frac{1}{7}$$

$$P(X|C') = \frac{0.2}{0.2+0.1} = \frac{2}{3}$$

$$P(X|C) = \frac{0.6}{0.6+0.1} = \frac{6}{7}$$

Razonamiento hipotético e inducción por retroceso.

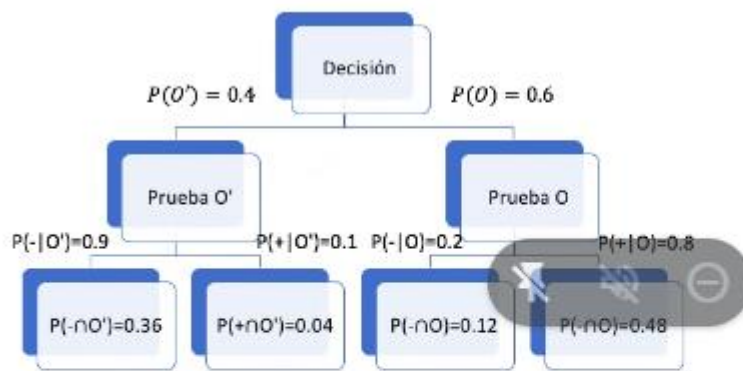
El teorema de Bayes es usado para análisis de árboles de decisión en negocios y ciencias sociales. Este mismo método es usado en PROSPECTOR para encontrar sitios favorables de exploración minera.

Al inicio PROSPECTOR debe decidir las probabilidades de encontrar petróleo

$$P(O) = P(O') = 0.5$$

Al asignar el mismo peso entre posibles resultados resulta imparcial. Sin embargo, PROSPECTOR considera mejor la siguiente probabilidad

$$P(O) = 0.6 \quad P(O') = 0.4$$



[] Ejercicio de clase, calcule las probabilidades faltantes.

La ley de adición de las probabilidades nos dice

$$p(+) = P(+ \cap O) + P(+ \cap O') = 0.48 + 0.04 = 0.52$$

$$p(-) = P(- \cap O) + P(- \cap O') = 0.12 + 0.36 = 0.48$$

a reunión

Para calcular el costo desde un inicio, debemos comenzar en las hojas del árbol. En teoría de probabilidad a esto se le conoce como inducción por retroceso. Esto es, con la finalidad de conseguir el pago esperado o la meta que queremos, debemos razonar hacia atrás para encontrar las causas que nos llevarán a la meta.

El pago esperado en el nodo C

$$846,153 = (10,000,000)(12/13) - (10,000,000)(1/13)$$

El pago esperado en el nodo B

$$500,000 = (10,000,000)(1/4) - (10,000,000)(3/4)$$

Pago esperado al nodo A

$$416,000 = (846,153)(0.52) - (50,000)(0.48)$$

La progresión del sistema a través de una secuencia de **estados** es llamado **proceso estocástico** si es probabilístico.

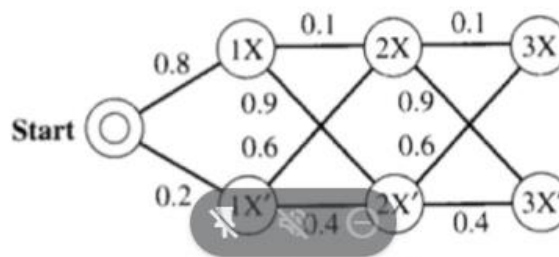
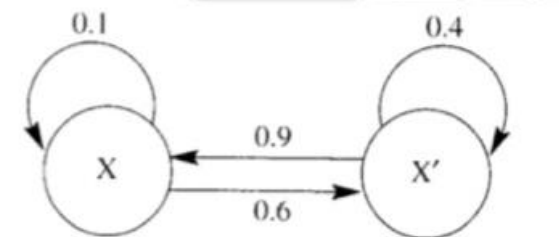
Matriz de transición

Considere dos estados  $S_1$  y  $S_2$ , la matriz de transición está dada

Presente	$S_1$		

Asuma que el 10 por ciento de toda la gente que usa la marca X comprara nuevamente la marca X. También, el 60 por ciento de la gente que no usa la marca X comprará la marca X. Considere la matriz de transición

$$T = \begin{matrix} & \begin{matrix} X & X' \end{matrix} \\ \begin{matrix} X \\ X' \end{matrix} & \begin{pmatrix} 0.1 & 0.9 \\ 0.6 & 0.4 \end{pmatrix} \end{matrix}$$



(b) Lattice Diagram of States

Suponga que el 80 por ciento usan la marca X.

La probabilidad del sistema de estar en cierto estado puede representarse por un vector de estados

donde  $P_1 + P_2 + \dots + P_n = 1$ .

Para nuestro ejemplo donde el 80 por ciento de la gente le pertenece a la marca X, entonces el vector de estados es

$$S_1 = [0.8 \ 0.2]$$

Para calcular el siguiente estado

$$S_2 = [0.8 \ 0.2] \begin{bmatrix} 0.1 & 0.9 \\ 0.6 & 0.4 \end{bmatrix} = [ \quad \quad ]$$

entonces el vector de estados es

$$S_1 = [0.8 \ 0.2]$$

Para calcular el siguiente estado

$$S_2 = S_1 T$$
$$S_2 = [0.8 \ 0.2] \begin{bmatrix} 0.1 & 0.9 \\ 0.6 & 0.4 \end{bmatrix} = [0.2 \ 0.8]$$

$$S_3 = S_2 T = [0.5 \ 0.5]$$

$$S_4 = S_3 T = [0.35 \ 0.65]$$

$$S_5 = S_4 T = [0.425 \ 0.575]$$

$$S_6 = S_5 T = [0.3875 \ 0.6125]$$

$$S_7 = S_6 T = [0.3875 \ 0.6125]$$

T = la matriz segunda de S1

Un proceso en cadena de Márkov es definido como

- 1) Un número finito de estados posibles
- 2) El proceso puede estar en uno y solo un estado a la vez.
- 3) El proceso se mueve sucesivamente de un estado a otro a través del tiempo.
- 4) La probabilidad de un movimiento depende solo del estado inmediato anterior.

Dado un conjunto de estados {A,B,C,D,E,F,G,H}, si el siguiente estado después de H es I, entonces la probabilidad condicional es

$$P(I|H) = P(I|H \cap G \cap F \cap E \cap D \cap C \cap B \cap A).$$

Ejercicio de clase,

Dada la matriz de transición para cambiar de marcas de discos duros en un año

		Next Year		
		X	X	Z
This Year	X	.5	.5	0
	Y	.25	.5	.25
	X	0	.5	.5

a) Asuma que inicialmente el 50% de la población usa la marca X, 25% usa Y, y 25% usa Z ¿Qué porcentaje usará cada marca en los siguientes uno, dos y tres años?

b) Determine el vector de estado estable.



## Lógica y SE

Un SE es un programa que guía al usuario en la solución de algún problema el cuál normalmente requiere intervención de un humano experto en el campo.

## Diagnóstico



Diagnóstico significa tratar de encontrar la causa de algún mal funcionamiento, por ejemplo, la causa de una enfermedad.

## Control

En aplicaciones de control lo principal es evitar que un sistema, como por ejemplo algún proceso industrial, entre a un estado anormal.

## Planeación

Planeación significa intentar encontrar una secuencia de estados de transición que terminen en un estado final específico mediante una secuencia de estados intermedios dados por un estado inicial.

## Características de los sistemas expertos:



- Se dividen en motor de inferencia y base de conocimientos. La base de conocimientos contiene las reglas que describen el conocimiento general acerca de algún problema de dominio. El motor de inferencia es usado para inferir conocimiento a partir de la base de conocimiento.
- Puede contener reglas sujetas a incertidumbre.
- El sistema cuenta con una interfaz de usuario.
- |



SE = Base de conocimientos + Control + Interfaz de usuario

### Ejemplo MYCIN

SI la mancha del organismo es gram-positiva

Y la morfología del organismo es coccus

Y la conformación del crecimiento del organismo es en grumos

ENTONCES la identidad del organismo es coccus (0.7)

En cláusulas definidas

identidad\_del\_organismo(staphylococcus):-

mancha\_del\_organismo(gram\_positiva),

morfologia\_del\_organismo(coccus),

conformacion\_del\_crecimiento\_del\_organismo(grumos).

Consideremos un diagnóstico para problemas al encender un automóvil. De manera general:

- Si Y es un componente que necesita X y Y presenta un mal funcionamiento entonces X presenta un mal funcionamiento;
- Si X exhibe un síntoma Y y Y también X presenta un mal funcionamiento o existe algún otro componente que presenta mal funcionamiento el cuál es necesario para X.

malFuncionamiento(X):-necesita(X,Y),malfuncionamiento(Y).

El motor de inferencia es usado para inferir nuevo conocimiento a partir del conocimiento existente. Usando dos estrategias

- 1)Comenzar desde lo que es conocido e inferir nuevo conocimiento a partir de éste.  
(Desencadenamiento hacia adelante)

- 1)Comenzar desde la conclusión a probar y aplicar razonamiento hacia atrás hasta que la conclusión depende de lo que ya es conocido. (Desencadenamiento hacia atrás)

## Módulo de explicación

Una explicación puede ser representada como una prueba o un árbol de derivación. Para nuestro ejemplo tenemos tres tipos de metas:

- La meta vacía (representada por la constante true);
- La meta compuesta de la forma  $X \text{ e } Y$ ;
- La meta consistente de una sola variable.

Consideremos la siguiente base de conocimiento

```
abuelo(X,Z):-progenitor(X,Y),progenitor(Y,Z).  
progenitor(X,Y):-padre(X,Y).  
padre(adan,guillermo).  
padre(guillermo,catalina).
```

Ejemplo de la recolección de pruebas.

```
resuelve(padre(X,Y),prueba(padre(X,Y),verdad)):-padre(X,Y).
```

```
resuelve(abuelo(X,Z),prueba(abuelo(X,Z),Px,Py)):-  
resuelve(progenitor(X,Y),Px),resuelve(progenitor(Y,Z),Py).
```

```
resuelve(progenitor(X,Y),prueba(progenitor(X,Y),Z)):-resuelve(padre(X,Y),Z).
```

I\*



Como hemos visto, es fácil probar una meta verdadera. La situación se complica cuando intentamos resolver metas que contienen literales negativas. Dado que la negación es una falla de la regla que significa NOT X y tiene éxito si X falla, entonces no hay prueba que retornar. Una solución simple sería añadir la cláusula:

```
resuelve(padre(X,Y),prueba(padre(X,Y),falso)):-not(padre(X,Y)).
```

## Interfaz de usuario

Considere la siguiente base de conocimiento

`malFuncionamiento(X):-posibleCausa(Y,X),sintoma(Y,X).`  
`posibleCausa(ponchada,llanta).`

En el caso de diagnóstico, los síntomas se especifican durante el tiempo de ejecución. Esto es, cuando el motor de inferencia encuentra ciertos predicados no busca por la definición de los predicados en la base de conocimientos si no que pregunta al usuario por esa información.



Ejemplo:

`sintoma(X,Y):-confirma(X,Y).`

`confirma(X,Y):-`

`write("Esta la"),write(Y),tab(1),write(X),write("?"),nl,read(si).`

 INTERFAZ DE

USUARIO



---

## Razonamiento inexacto

### Incertidumbre y reglas

#### Incertidumbre en reglas:

- Reglas individuales
  - Antecedente
    - Errores
    - Probabilidad de la evidencia
    - Combinando evidencia
- Conflictos de resolución
- Incompatibilidad de reglas

$E_1 \text{ AND } E_2 \text{ AND } E_3$

$E_1 \text{ AND } E_2 \text{ OR } E_3$

$E_1 \text{ AND NOT } E_2 \text{ OR } E_3$

#### ◦ Consecuente

- Errores
- Probabilidad de la evidencia
- Conflictos de resolución
- Incompatibilidad de reglas

---

#### Compatibilidad de reglas:

- Prioridad de reglas
- Contradicción de reglas
- Subsunción de reglas
- Redundancia de reglas
- Reglas perdidas
- Fusión de datos





Ejemplo de **contradicción** de reglas

- 1) SI hay fuego ENTONCES pon agua sobre el fuego
- 2) SI hay fuego ENTONCES no pongas agua sobre el fuego

Ejemplo de subsunción de reglas

- 3) SI  $E_1$  ENTONCES H
- 4) SI  $E_1$  Y  $E_2$  ENTONCES H

Conflicto de resolución:

- Prioridad implícita de reglas
  - Especificidad de patrones

(ball elipsoidal)

(ball)

- Recientes hechos de coincidencia de patrones
- Ordenamiento de patrones
  - Lexicográficos (LEX)
  - Análisis de medios y fines (MEA)

}

- Prioridad explícita de reglas

- 
- 3) SI  $E_1$  ENTONCES H
  - 4) SI  $E_1$  Y  $E_2$  ENTONCES H

En OPS5 o CLISP, la regla 4) es más específica porque tiene dos patrones en sus antecedentes y por eso tiene una prioridad implícita más alta.

Sin embargo

- 5) SI  $E_3$  ENTONCES H

Tendrá mayor prioridad que la regla 3 si el hecho  $E_3$  fue ingresado después del  $E_1$ .

El ordenamiento de patrones se puede hacer de manera lexicográfica (LEX)



```
3 gauthier: Ruben Aramazan
4 """
5
6 garceta_pie_dorado = ['59', 'amarillo', 'negro', 'blanco']
7 garza_ganadera = ['50', 'amarillo', 'amarillo', 'blanco']
8
9 print('*****')
10 print('* Clasificador de aves endémicas de Jalisco *')
11 print('*****')
12 print('--> Ingrese los datos solicitados:')
13
14 pregunta = []
15
16 tamaño = input("\n--> Tamaño general: ")
17 ojos = input("--> Color de Ojos: ")
18 pico = input("--> Color de Pico: ")
19 plumaje = input("--> Color de Plumaje: ")
20
21 pregunta.append(tamaño)
22 pregunta.append(ojos)
23 pregunta.append(pico)
24 pregunta.append(plumaje)
```

\* Clasificador de aves endémicas de Jalisco \*

\*\*\*\*\*

--> Ingrese los datos solicitados:

--> Tamaño general: 59

--> Color de Ojos: amar

|| meet.google.com está compartiendo tu pantalla. [Dejar de compartir](#) [Ocultar](#)

---

## Razonamiento aproximado

### Conjuntos difusos y lenguaje natural

En los conjuntos difusos un objeto puede pertenecer parcialmente a un conjunto. El grado de pertenencia en un conjunto difuso es medible por una generalización de la función característica, llamada **función de membresía** o función de compatibilidad

$$\mu_{\text{lambda}}(x): x \rightarrow [0,1]$$

Un valor particular de la función membresía es llamado **grado de membresía**.

Los conjuntos difusos y los conceptos son comúnmente usados en el lenguaje natural, como:

"Juan es alto"

"El clima es **cálido**"

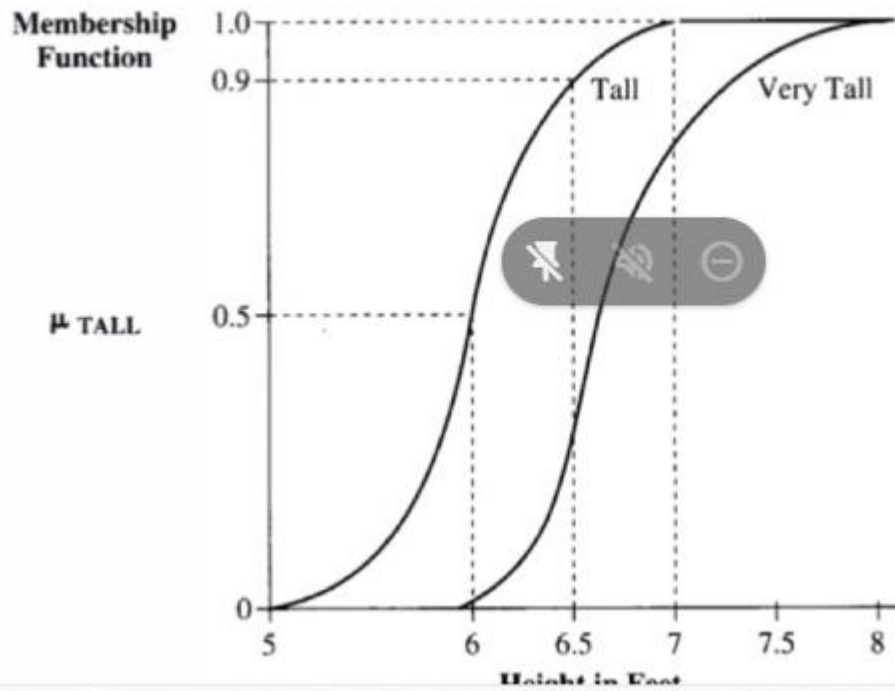
"Gire la perilla un poco más"

"La mayoría de los exámenes son difíciles"

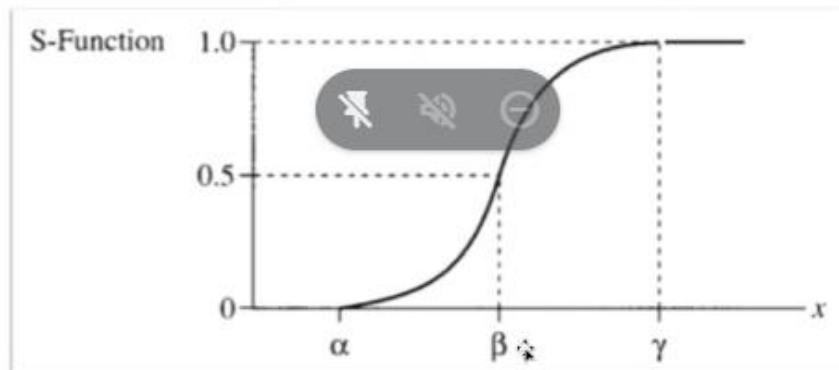
"Si la masa está muy dura, añada mucha agua"

"Si la masa está muy dura, añada mucha agua"

Una proposición difusa contiene palabra como alto, la cual es identificador del conjunto ALTO. Por ejemplo, la proposición "**Juan es alto**", puede ser verdadera en cierto grado: Un poco verdad, algo verdadera, bastante verdadera, muy verdadera y así. Un valor de verdad difuso es llamado calificador difuso. Las proposiciones pueden tener cuantificadores como La mayoría, Algunos, Usualmente, entre otros.



La función sigmoideal es una función matemática usada con frecuencia en conjuntos difusos como función de membresía.

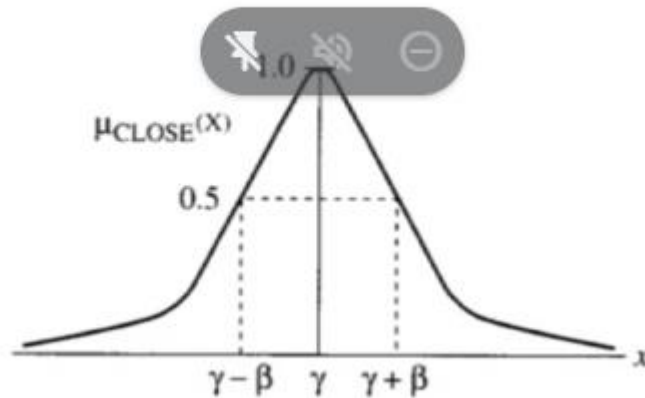


$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \text{for } x \leq \alpha \\ 1 - 2\left(\frac{x - \gamma}{\gamma - \alpha}\right)^2 & \text{for } \beta \leq x \leq \gamma \\ 1 & \text{for } x \geq \gamma \end{cases}$$

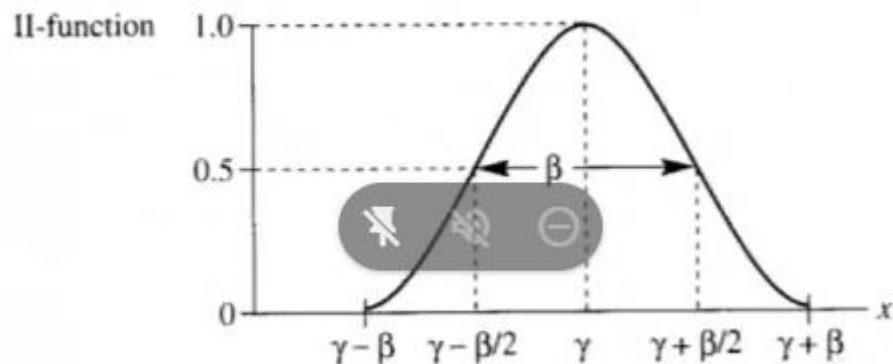
Para nuestro ejemplo, quedaría como

$$S(x; 5, 6, 7) = \begin{cases} 0 & \text{para } x \leq 5 \\ 2\left(\frac{x - 5}{7 - 5}\right)^2 & \text{para } 5 \leq x \leq 6 \\ 1 - 2\left(\frac{x - 7}{7 - 5}\right)^2 & \text{para } 6 \leq x \leq 7 \\ 1 & \text{para } x \geq 7 \end{cases}$$

Una función de membresía para la proposición "X está cerca de Y" es



Otra función con una curva similar es:





En lugar de una función continua, la función de membresía puede ser un conjunto finito de elementos. Por ejemplo, en el universo de las alturas definido como

$$U = \{5, 5.5, 6, 6.5, 7, 7.5, 8\}$$

un subconjunto difuso puede ser definido como

$$ALTOS = \left\{ \frac{0}{5}, \frac{0.125}{5.5}, \frac{0.5}{6}, \frac{0.875}{6.5}, \frac{1}{7}, \frac{1}{7.5}, \frac{1}{8} \right\}$$

El símbolo / indica para un valor de pertenencia  $\mu(x) > 0$  el conjunto de elementos soportan el conjunto difuso. Para este subconjunto, soporta todos los elementos excepto el  $\frac{0}{5}$ .

El soporte de un conjunto difuso, F, es un subconjunto del universo, X, definido como

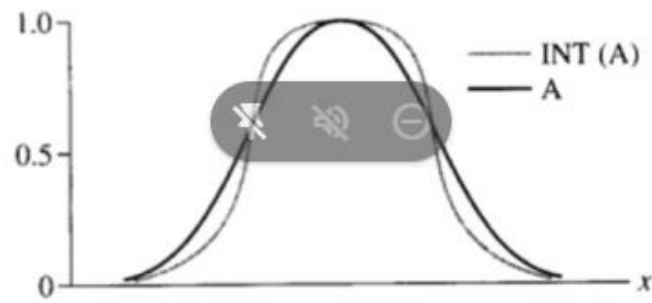
El símbolo / indica para un valor de pertenencia  $\mu(x) > 0$  el conjunto de elementos soportan el conjunto difuso. Para este subconjunto, soporta todos los elementos excepto el  $\frac{0}{5}$ .

El soporte de un conjunto difuso, F, es un subconjunto del universo, X, definido como

$$\text{soporte}(F) = \{x | x \in X \text{ y } \mu_f > 0\}$$

Un concepto relacionado con los conjuntos de soporte son los cortes alfa,

$$F_\alpha = \{x | \mu_F(x) \geq \alpha\} \text{ para } 0 < \alpha \leq 1$$



Como ejemplo, en electrónica, considere el punto de cruce como la definición del ancho de banda. La operación de intensificación amplifica la señal entre el ancho de banda, mientras reduce el "ruido" fuera del ancho de banda.