

## Índice

Clase del 15/08/2022 .....	4
Clase del 17/08/2022 .....	4
Introducción a los sistemas expertos .....	4
Inteligencia Artificial (IA) .....	4
Ventajas de los Sistemas Expertos .....	5
Conceptos generales .....	6
Características de un SE .....	7
Clase del 22/08/2022 .....	8
Desarrollo de la tecnología en sistemas expertos.....	8
Elementos de un sistema experto.....	9
Programación imperativa.....	10
Programación funcional .....	10
Programación declarativa.....	11
Programación Orientada a Objetos .....	11
Clase del 24/08/2022 .....	13
Programación no declarativa.....	13
Programación basada en inducción.....	13
Representación del conocimiento.....	14
Producción de reglas .....	14
Redes semánticas.....	15
Clase del 29/08/2022 .....	16
Clase del 31/08/2022 .....	16
Terna objeto-atributo-valor .....	16
Desventajas de las redes semánticas.....	16
Esquemas .....	16
Frames .....	17
Lógica y conjuntos.....	18
Lógica proposicional .....	19
Lógica de predicados (Lógica de primer orden) .....	19
Clase del 07/09/2022 .....	21
Métodos de inferencia.....	21

Grafos y árboles .....	21
Problemas de espacio de estados.....	24
Árboles AND-OR .....	25
Clase del 12/09/2022 .....	27
Métodos de inferencia.....	27
Lógica deductiva y silogismos.....	28
Reglas de inferencia .....	28
Limitaciones de la lógica proposicional .....	30
Lógica de predicados (Lógica de primer orden) .....	30
Razonamiento causal y superficial .....	31
Razonamiento superficial .....	31
Razonamiento causal.....	32
Clase del 14/09/2022 .....	33
Razonamiento bajo incertidumbre .....	33
Errores e inducción .....	34
Probabilidad clásica .....	35
Clase del 19/09/2022 .....	36
Espacio de muestras.....	36
Árbol de eventos.....	36
Teoría de la probabilidad .....	36
Probabilidad condicional.....	37
Teorema de Bayes.....	37
Razonamiento hipotético e inducción por retroceso .....	38
Clase del 21/09/2022 .....	40
Razonamiento temporal y cadenas de Márkov .....	40
Matriz de transición .....	40
Clase del 26/09/2022 .....	44
Lógica y SE.....	44
Diagnóstico .....	44
Control.....	44
Planeación .....	44
Características de los sistemas expertos.....	44
Clase del 28/09/2022 .....	50

Razonamiento inexacto .....	50
Incertidumbre en reglas .....	50
Compatibilidad en reglas .....	50
Ejemplo de contradicción de reglas .....	50
Ejemplo de subsunción de reglas .....	50
Conflicto de resolución.....	51
Clase del 03/10/2022 .....	52
Razonamiento aproximado.....	52
Tipos de conjuntos difusos .....	56
Tipo 1 de conjuntos difusos .....	56
Tipo 2 de conjuntos difusos .....	56
Operaciones de conjuntos difusos.....	57
Igualdad de conjuntos .....	57
Complemento de conjunto.....	57
Pertenencia de conjuntos .....	57
Unión de conjuntos.....	57
Intersección de conjuntos .....	58
Producto de conjuntos .....	58
Conjunto potencia.....	58
Suma probabilística.....	59
Suma acotadao.....	59
Producto acotado .....	59
Diferencia acotada.....	59
Concentración.....	59
Dilatación.....	60
Intensificación .....	60
Normalización .....	61

Clase del 15/08/2022

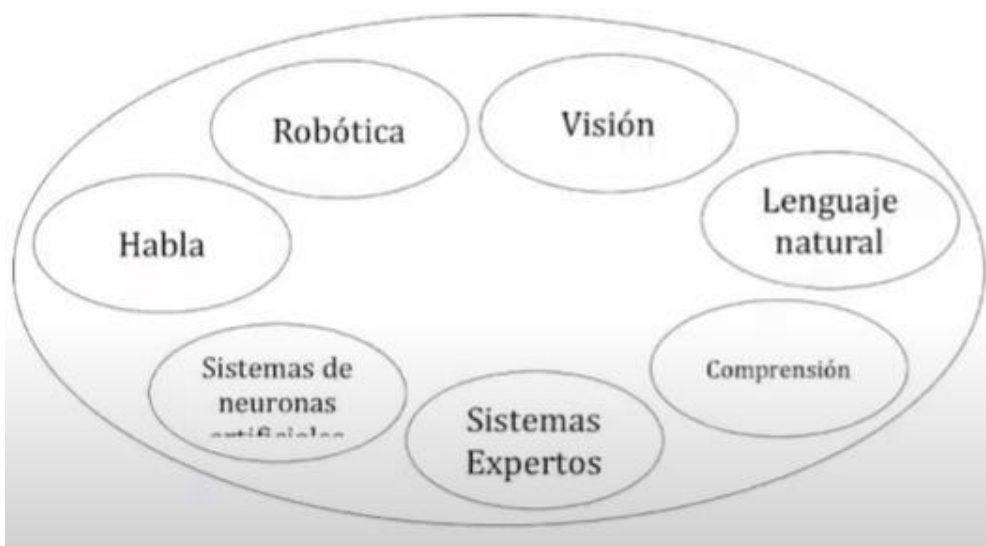
Explicación del curso

Clase del 17/08/2022

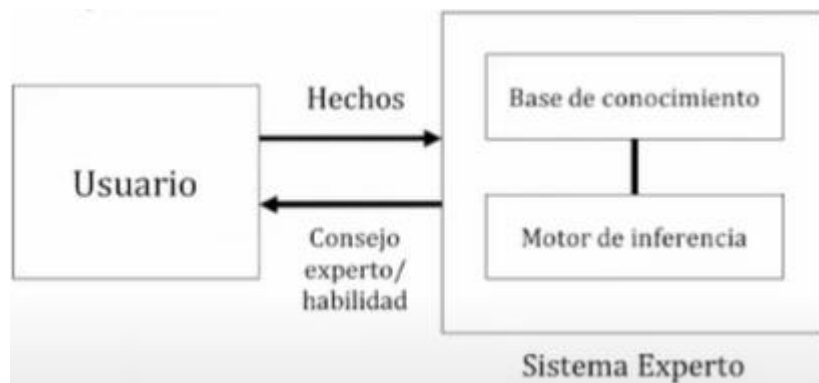
Introducción a los sistemas expertos

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.

Inteligencia Artificial (IA)



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.



En el dominio del conocimiento de lo que se sabe al respecto, el sistema experto razona o hace inferencias de la misma manera en que un experto humano inferiría la solución de un problema. Esto es, dado algunos hechos, una conclusión consecuente es inferida.

Por ejemplo, si tu espos@ no te ha hablado en un mes, tú quizás infieras que él o ella no tiene nada relevante que decir.

### 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.

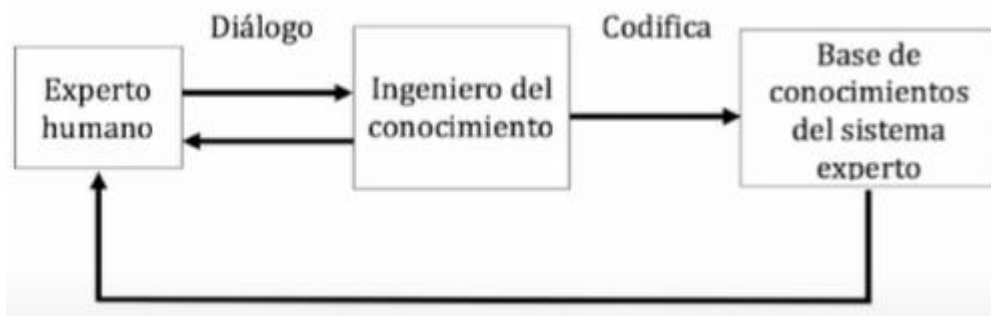
- Base de datos inteligente.

### Conceptos generales

El conocimiento de un Sistema Experto puede ser representado de diversas formas, Un método común de representación del conocimiento es en forma de Si...ENTONCES tipo-de-reglas.

Si la luz es roja ENTONCES detente

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 una 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.

Clase del 22/08/2022

### Desarrollo de la tecnología en sistemas expertos

En 1957 aparece el Solucionador General de Problemas, creado por Newell y Simon.

Uno de los más significativos resultados fue que la cognición podría ser expresada en reglas de producción SI-ENTONCES.

Si el carro no arranca Y el medidor de gasolina marca vacío ENTONCES llena el tanque de gas

En los 70's aparecen los primeros sistemas expertos.

DENDRAL (1965-1975): Interpreta la estructura molecular para identificar los componentes químicos.

MYCIN (1972): Diagnóstico de enfermedades infecciosas de la sangre.

PROSPECTOR (1974): Prospección minera.

XCON/R1 (1978): Asistía a los pedidos de los sistemas de computadores, seleccionando los componentes del sistema de acuerdo a los requerimientos del cliente.

DIPMETER (1980): Auxiliar en el análisis de los datos recolectados durante la exploración petrolera.



En los 80's aparece el software para desarrollo de sistemas expertos



Automated Reasoning Tool (ART) por Inference Corp.

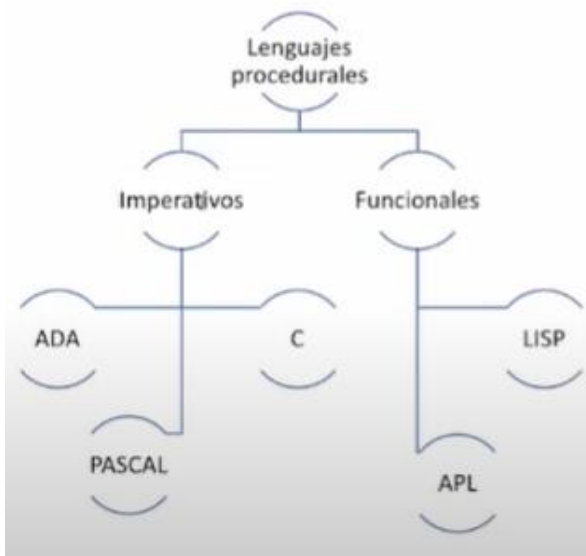
Knowledge Engineering Tool (KEE) por IntelliCorp.

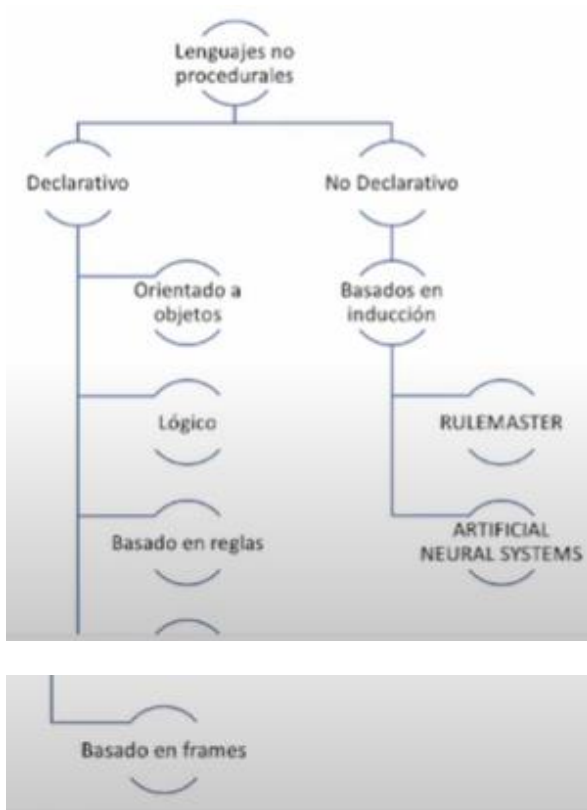
Rulemaster por Radian Corp.

CLIPS por NASA.

### Elementos de un sistema experto

- Interfaz de usuario: Mecanismo por el cual el usuario y el sistema experto se comunican.
- Componente de explicación: Explica el razonamiento del sistema al usuario.
- Memoria de trabajo: Base de datos global con los hechos usados por las reglas.
- Motor de inferencia: Realiza inferencias con base a las reglas satisfechas por los hechos u objetos, prioriza las reglas satisfechas y ejecuta la regla con la mayor prioridad.
- Agenda: Lista de reglas priorizadas creadas por el motor de inferencia.
- Componente de adquisición del conocimiento: Forma automática en la que el usuario ingresa conocimiento al sistema en lugar de tener a un ingeniero del conocimiento codificando el nuevo conocimiento.





### Programación imperativa

Lenguajes como FORTRAN, Ada, Pascal, Modula-2, COBOL Y BASIC tienen como característica que las sentencias son imperativas. Los lenguajes imperativos ofrecen un gran soporte a variables, operaciones de asignación y repetición.

Desde el punto de vista de la IA, los lenguajes imperativos no son eficientes para la manipulación simbólica. Sin embargo, lenguajes como C y Ada han sido usados como base para escribir el shell del SE.

### Programación funcional

La idea fundamental de la programación funcional es combinar simples funciones para obtener funciones más poderosas.

- Objetos de datos: con los que opera el lenguaje.
- Funciones primitivas: para operar los objetos de datos.
- Formas funcionales: Para sintetizar nuevas funciones a partir de otras funciones.

- Operaciones de las aplicaciones: sobre funciones que regresan un valor.
- Procedimientos de nomenclatura: para identificar nuevas funciones.

Los paradigmas no procedurales no dependen de que el programador de los detalles exactos de cómo un problema es resuelto. El énfasis está en especificar el objetivo a alcanzar y dejar al sistema determinar cómo hacerlo.

### Programación declarativa

El paradigma declarativo separa la meta de los métodos usados para alcanzar la meta.

### Programación Orientada a Objetos

Una instancia de una clase es un objeto de datos que puede ser manipulado. El término instanciar ha sido llevado a sistemas expertos, en el cual denota un hecho que ha coincidido con un patrón. De igual forma, una regla se dice instanciada si se ha satisfecho.

Muchas herramientas de sistemas expertos como ART y KEE permiten herencia dado que es una poderosa herramienta para construir un gran grupo de hechos.

### Programación lógica

Uno de las primeras aplicaciones de A en computadoras fue para probar teorema con el programa Logic Theorist de Newell y Simon. Su sucesor, el General Problem Solver (GPS) era capaz de resolver más que teoremas matemáticos, resolvía cualquier problema lógico. Por ejemplo:

- Ajedrez.
- Las torres de Hanoi.
- Los misioneros y los caníbales.
- Criptogramas.



En 1972 el lenguaje PROLOG fue creado por Kowalski, Colmerauer y Roussel para implementar programas lógicos por retroceso usando cláusulas de Horn (sentencias de la forma "conclusión si condiciones").

$P1 \& P2 \rightarrow C1$

$C1 \leftarrow P1 \& P2$

Ejemplo de cláusulas de Horn

alguien es mortal

si alguien es un hombre

Sócrates es un hombre

si (en todos los casos)

La eficiencia es afectada por la manera cómo Prolog busca en su base de datos. Hay programas que se ejecutan correctamente si las sub-metas, hechos y reglas son ingresadas de cierta forma, pero se produce un ciclo infinito o un error de ejecución si el orden cambia.

Sentencia1

Sentencia2

Sentencia3

...

## Clase del 24/08/2022

### Programación no declarativa

#### Programación basada en inducción

En este paradigma el programa aprende, por ejemplo. Herramientas de sistemas expertos como Ist-class y KDS ofrecen aprendizaje por inducción.

#### Sistemas de neuronas artificiales (SNA)

##### El problema del vendedor viajero

El problema consiste en calcular la ruta más corta a través de una lista de ciudades dadas.

Ciudades	Rutas
1	1
2	1-2-1
3	1-2-3-1 1-3-2-1
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.

`<sentence>::=<subject><verb><end-mark>`

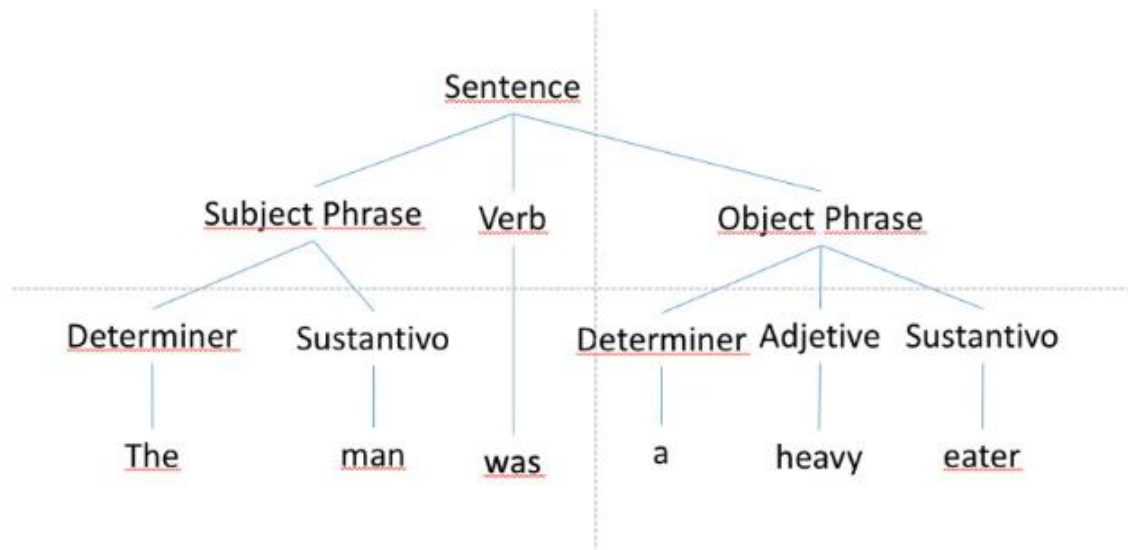
`<subject>::=I|You|We`

`<verb>::=left|came`

`<end-mark>::=.|?!|`

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

Un árbol de análisis sintáctico es una representación gráfica de una oración descompuesta en todos sus terminales y no-terminales.



`<sentence>::=<subject phrase><verb><object phrase>`

`<subject phrase>::=<determiner><noun>`

`<determiner>::=the`

`<noun>::=man`

`<verb>::=was`

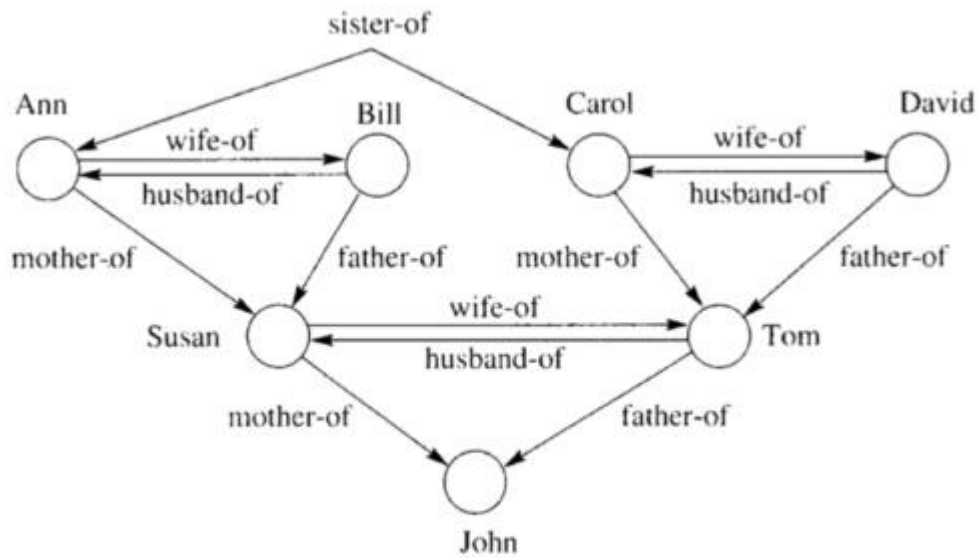
<object phrase>::=<determiner><adjective><noun>

<determiner>::=a

<adjective>::=heavy

<noun>::=eater

### Redes semánticas



(b) A Semantic Net

Hay dos tipos de aristas comúnmente usadas como IS-A y A-KIND-OF, escritos como ISA y AKO.

### Clase del 29/08/2022

No hubo clases

### Clase del 31/08/2022

#### Terna objeto-atributo-valor

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

- 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.

#### Esquemas

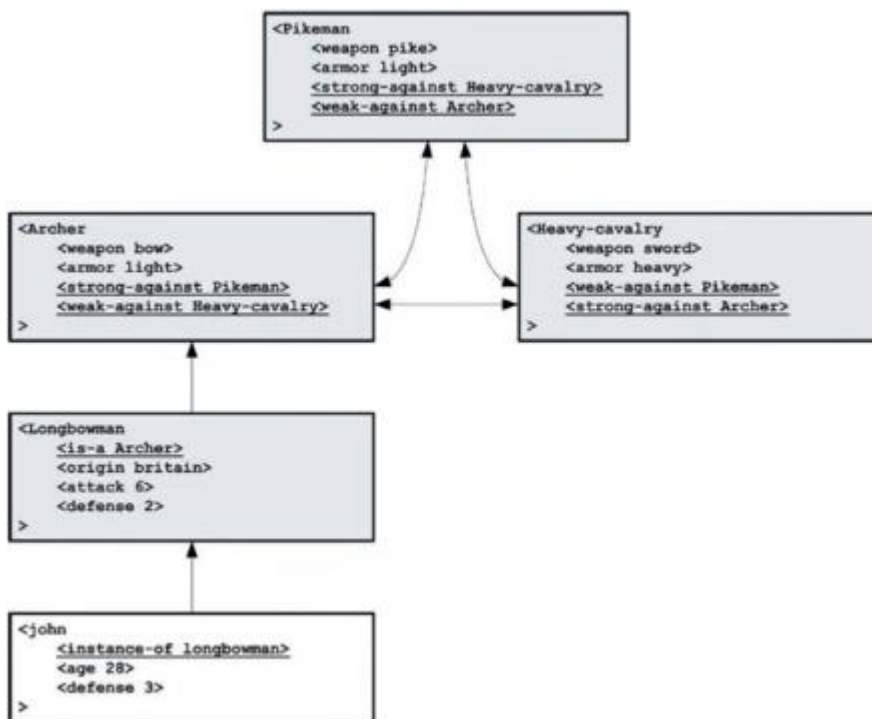
En IA, el término esquema es usado para describir una estructura más compleja de conocimiento que la red semántica. El término esquema viene de la psicología, en la cual hace referencia a la organización continua del conocimiento o respuestas de una creatura debido a un estímulo.



Un esquema conceptual es una abstracción en la cual específicos objetivos son clasificados por sus propiedades generales, un esquema es como una estructura de datos en la cual cada nodo contiene registros.

## Frames

Un tipo de esquema que ha sido usado en aplicaciones de IA es el frame. Otro tipo de 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 procedimiento 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 defecto. Los valores por defecto representan el conocimiento por sentido común; el

if-added es un procedimiento que es ejecutado cuando un valor es añadido; y el if-removal que corre cuando un valor es eliminado de un espacio.

Los campos de relleno también pueden contener relaciones, como especializaciones de espacios.

El mayor inconveniente que aparece en los sistemas de frames es que permiten la alteración o cancelación de espacios.

### Lógica y conjuntos

Los silogismos son maneras de representar el conocimiento, otra manera es mediante diagrama de Venn.

Premisa: Todos los hombres son mortales.

Premisa: Socrates es un hombre.

Conclusión: Socrates es mortal.

En prolog quedaría de la siguiente manera:

hombre(socrates).

mortal(X):-hombre(X).

?-mortal(X)

X = socrates

- mortal(Alguien/Quien/socrates):-hombre(Quien/socrates).
- hombre(socrates/Quien). %true

## Lógica proposicional

La lógica proposicional es una lógica simbólica para la manipulación de proposiciones. En particular, trata con la manipulación de variables lógicas las cuales representan proposiciones.

Una sentencia para la cual se puede determinar su valor de verdad, la llamamos declaración o proposición.

Una sentencia para la cual no se puede determinar su valor de verdad son llamadas sentencias abiertas.

“Todo en esta diapositiva es una mentira” (paradojas)

La sentencia “Las espinacas son deliciosas” también es una sentencia abierta.

Una condicional es análoga a las reglas de producción.

Si está lloviendo entonces carga un paraguas.

$p \rightarrow q$

El mayor problema de la lógica proposicional es que solo trabaja con sentencias completas.

## Lógica de predicados (Lógica de primer orden)

La lógica proposicional es un subconjunto de la lógica de predicados. La lógica de predicados se interesa en la estructura interna de las sentencias. En particular, de los cuantificadores “todo”, “alguno” y “ninguno”.

Si X es humano

Entonces X es mortal

El problema con la lógica de predicados es que el cuantificador como “mayoría” no pueden ser expresados, otra limitante es expresar cosas que, en ocasiones, pero no siempre son verdad.

“La mayoría de la clase obtuvo 10 en su primer actividad”.

The screenshot displays the SWISH web interface (swish.swi-prolog.org) with a Prolog program loaded in the editor. The program defines several predicates: `incremento`, `decremento`, `suma`, and `multiplica`. The `suma` predicate is defined recursively, and the `multiplica` predicate uses the `suma` predicate to calculate the product of two numbers. The program also includes some test cases and a goal to be executed.

```
1 incremento(0,1).
2 incremento(1,2).
3 incremento(2,3).
4 incremento(3,4).
5 incremento(4,5).
6 incremento(5,6).
7 incremento(6,7).
8 incremento(7,8).
9 incremento(8,9).
10 incremento(9,10).
11
12 decremento(X,Y):-incremento(Y,X).
13
14 #suma
15 suma(NUMERO,0,NUMERO).
16 suma(NUMERO1,NUMERO2,RESULTADO):-
17   incremento(NUMERO1,SIGUIENTE1),
18   decremento(NUMERO2,SIGUIENTE2),
19   suma(SIGUIENTE1,SIGUIENTE2,RESULTADO).
20 #multiplicacion
21 multiplica(NUMERO,1,NUMERO).
22 multiplica(OPERANDO1,OPERANDO2,RESULTADO):-
23   decremento(OPERANDO2,SIGUIENTE2),
24   multiplica(OPERANDO1,SIGUIENTE2,VARIABLE),
25   suma(OPERANDO1,VARIABLE,RESULTADO).
26
27 #X=Y+2
28
29 #2X=Y+5
30
31 #X=Y+2+2
32
33 #2X=Y+2+5
34
35 #X=32
```

The execution window shows the goal `suma(Y,2,X),multiplica(2,X,X2),suma(Y,5,X2).` with the following variable bindings:

- X = 3,
- X2 = 6,
- Y = 1

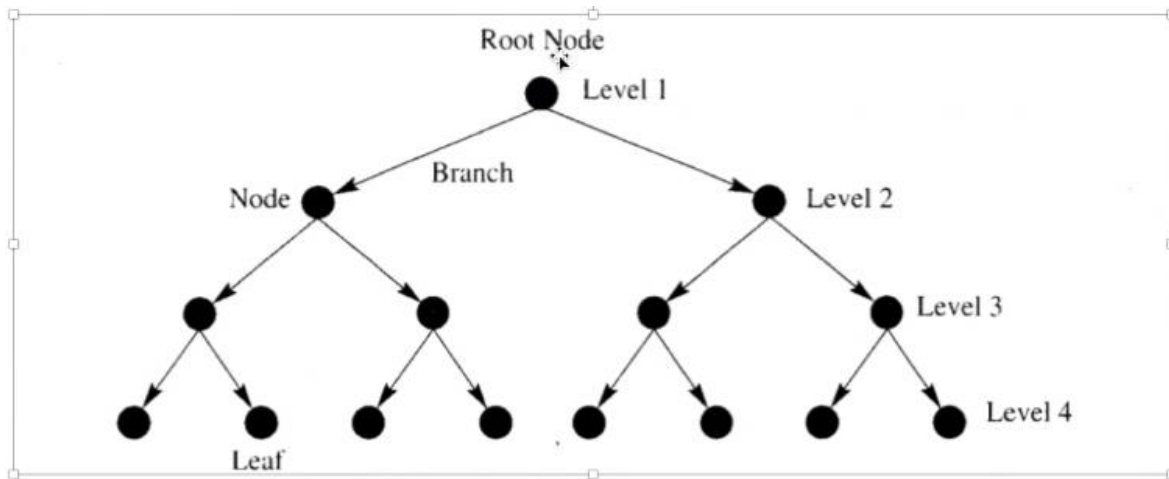
The execution window also includes a "Next" button and a "Stop" button. The "Next" button is highlighted, indicating the next step in the execution process.

Clase del 07/09/2022

Métodos de inferencia

Grafos y árboles

Un árbol es una estructura de datos jerárquica conformado por nodos, los cuáles almacenan información o conocimiento, y aristas, las cuáles conectan los nodos.



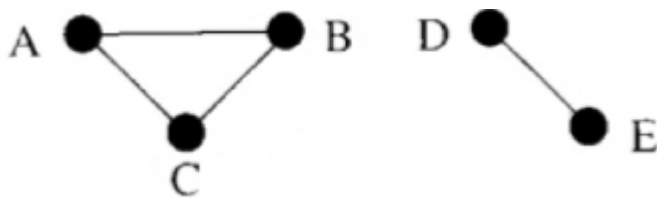
Los árboles son casos especiales de los grafos. Un grafo puede tener cero o más aristas entre sus nodos. Existen grafos ponderados en los cuáles las aristas tienen un peso.

Ejemplo una red neuronal artificial

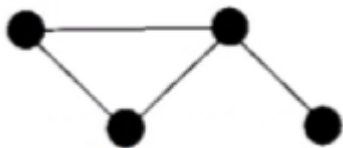
Una RNA es un ejemplo de un grafo con ciclos, debido a que durante el entrenamiento existe la retroalimentación para ajustar los pesos.

Ejemplo de grafos

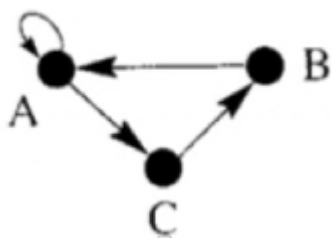
a) Un grafo no conectado



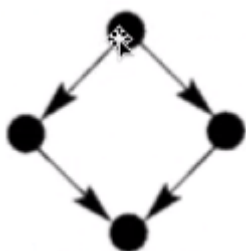
b) Un grafo conectado



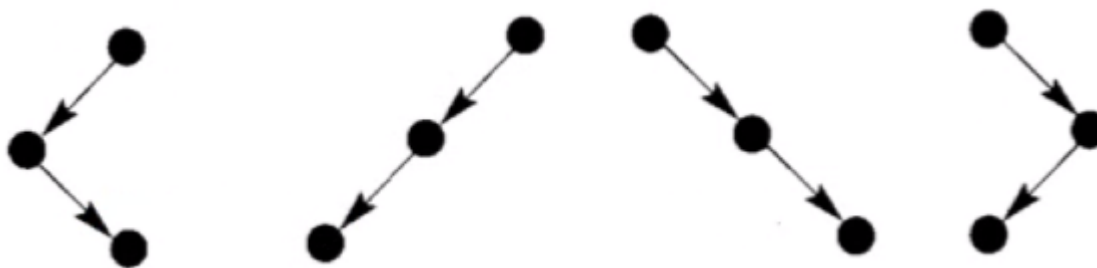
c) Un digrafo con auto-lazo y un circuito



d) Un grafo reticulado



e) Árboles binarios degenerados



Los árboles de decisión son una representación esquemática del conocimiento y un método de razonamiento sobre su conocimiento.

Por ejemplo, para clasificar animales



Un árbol binario de decisión puede tener  $2^n$  respuestas para  $n$  preguntas.



¿Entonces el juego de 20 preguntas cuántas posibles respuestas tiene?

$$2^{20} = 1,048,576 \text{ respuestas}$$

Otra herramienta útil de los árboles de decisión es que pueden construirse para auto-aprender. Si una suposición es incorrecta, un procedimiento puede preguntar al usuario por una nueva pregunta de clasificación y la respuesta. En consecuencia, dinámicamente se crearían nodos y aristas.

Los árboles de decisión pueden traducirse fácilmente a reglas de producción.

Ejemplo

SI PREGUNTA:= "¿ES GRANDE?" Y RESPUESTA = NO"

ENTONCES PREGUNTA:= "¿HACE UN CHILLIDO?"

SI PREGUNTA:="¿ES GRANDE?" Y RESPUESTA = SI"

ENTONCES PREGUNTA:="¿TIENE UN CUELLO LARGO?"

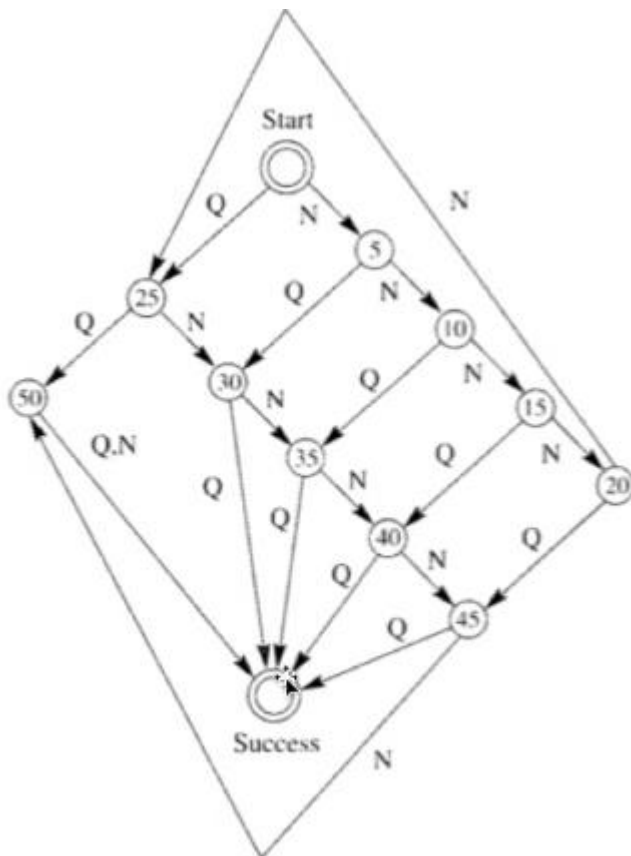
Sin embargo, los árboles de decisión son buenas herramientas de clasificación, pero no pueden manejar variables, como se espera de un sistema experto. Los sistemas expertos son herramientas de propósito general más allá de simples clasificadores.

### Problemas de espacio de estados

Un método útil de describir el comportamiento de un objeto es mediante un grafo de espacio de estados.

Un estado es una colección de características que son usadas para definir el estado de un objeto.

Ejemplo, considere una máquina de sodas.



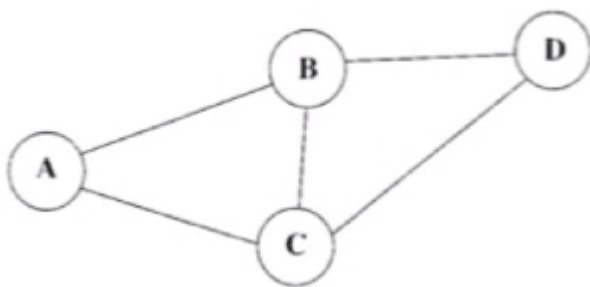
Este diagrama es llamado también máquinas de estado finito.



Los diagramas de estados son útiles para describir la solución a un problema. En este tipo de aplicaciones el espacio de estados se vuelve un problema de estados, en el cual algunos estados corresponden a etapas intermedias del problema a resolver y otros estados corresponden a respuestas.

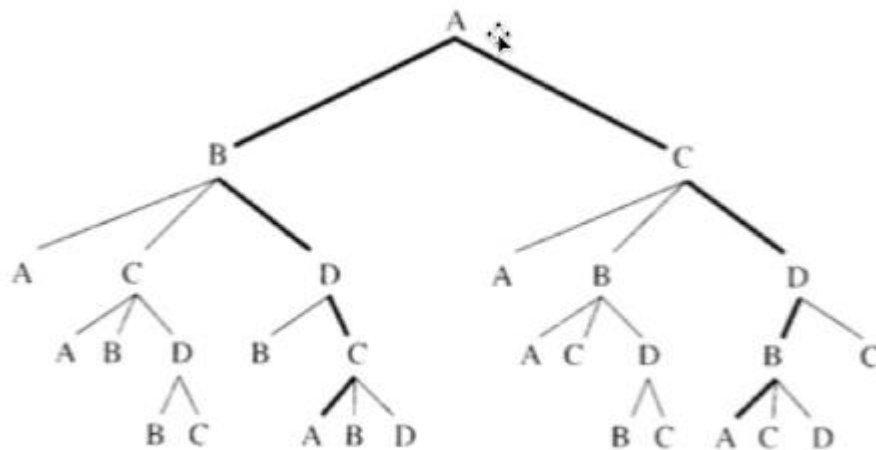
En un problema de estados hay múltiples estados de éxito correspondiente a soluciones posibles. Encontrar una solución implica encontrar un camino válido desde el inicio del problema hasta el éxito o respuesta.

El problema del vendedor viajero.



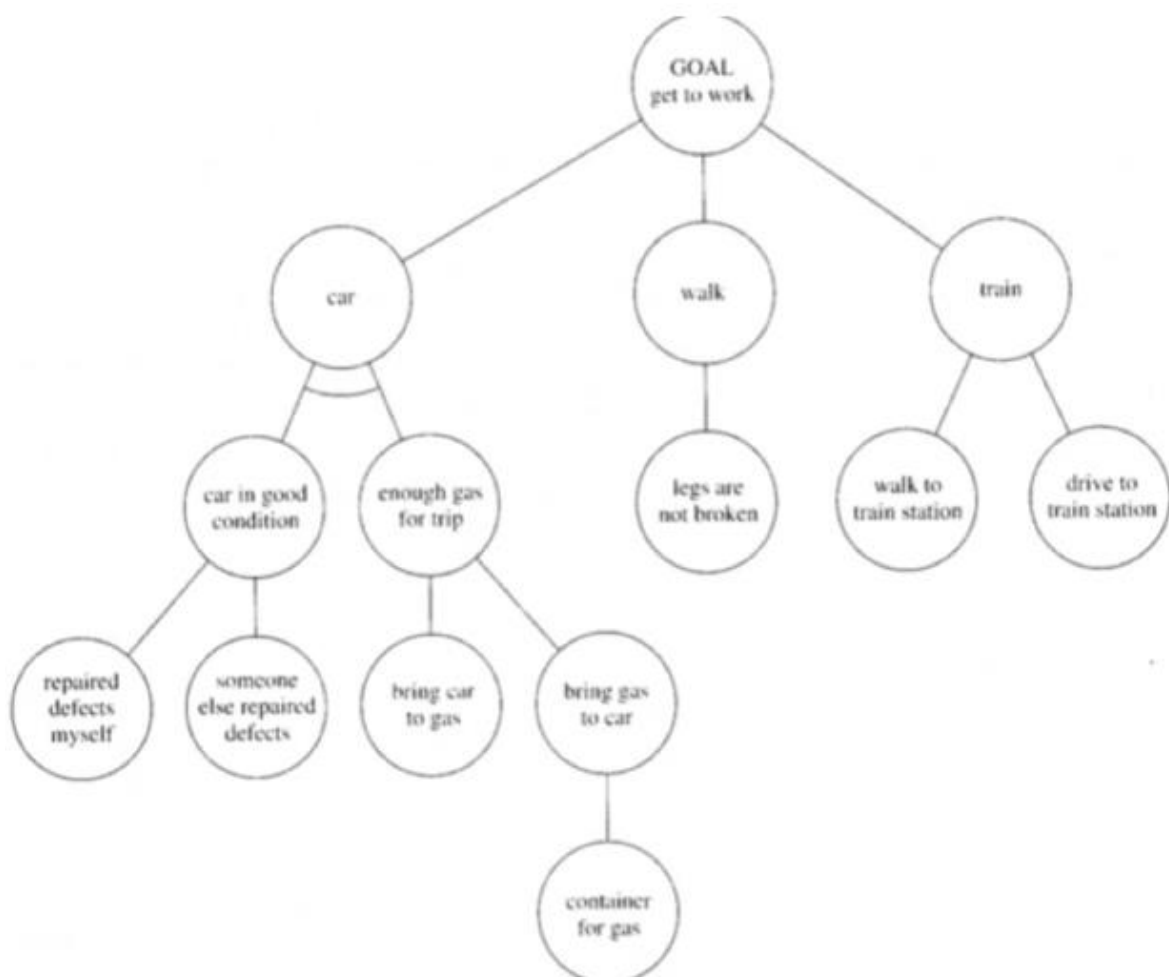
Ningún nodo puede ser visitado dos veces.

Dependiendo del algoritmo de búsqueda, la exploración de los caminos para encontrar el correcto puede requerir una considerable cantidad de retroceso.



### Árboles AND-OR

Un tipo de árbol que es útil representando problemas de encadenamiento hacia atrás son los árboles AND-OR.



Clase del 12/09/2022

### 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.

Autoepistémica: Auto-conocimiento.

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

A = Hay energía

B = La computadora encenderá

**$A \rightarrow B$**

**A**

**$\therefore B$**

A manera de silogismo quedaría expresado como

Toda computadora con energía encenderá

Esta computadora tiene energía

---

Esta computadora encenderá

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

Modus ponens

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

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.

$$p: \neg p_1, p_2, \dots, p_n.$$

La meta,  $p$ , se satisface si todas las submetas son satisfechas. Escrito en forma de reglas de producción podemos decir

$$C1 \wedge C2 \wedge \dots \wedge Cn \rightarrow A$$

### 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.

$p$   
 $q$   
 $\therefore r$

Si Sócrates es un hombre, entonces Sócrates es mortal.

Sócrates es un hombre.

Por lo tanto, Sócrates es mortal.

$p$  = Sócrates es un hombre.

$q$  = Sócrates es mortal

$p \rightarrow q$   
 $p$   
 $\therefore q$

### 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). %verdad

mortal(X)/mortal(socrates)

X/socrates

hombre(X)/hombre(socrates)

hombre(socrates)/hombre(socrates)

true

### Razonamiento causal y superficial

#### Razonamiento superficial

1) Si un carro tiene

una buena batería

buenas bujías

gasolina

buenas llantas

ENTONCES el carro puede moverse

Si el usuario solicita la explicación el sistema puede listar sus elementos condicionales.

una buena batería

buenas bujías

gasolina

buenas llantas

### 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

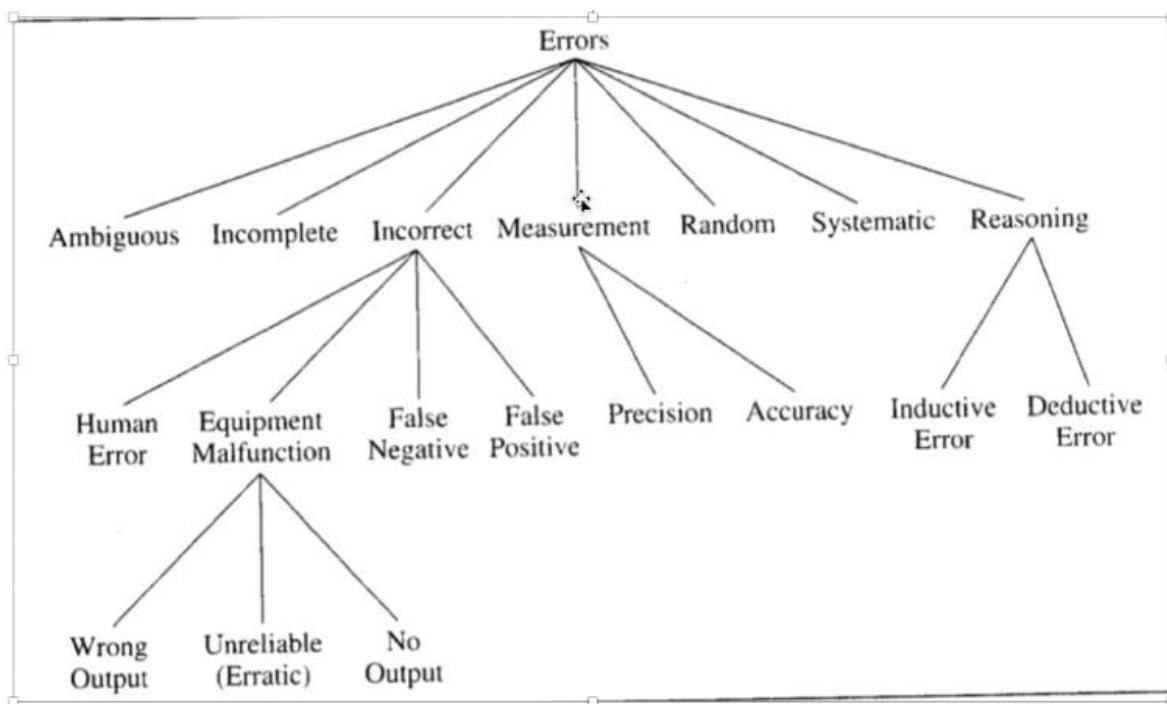
ENTONCES el carro se moverá.



Clase del 14/09/2022

## Razonamiento bajo incertidumbre

Incetidumbre: Falta de información adecuada para tomar una decisión.



Ejemplo	Error	Razón
Cierra la válvula.	Ambiguo.	¿Cuál válvula?
Gira la válvula-1.	Incompletitud.	¿Hacia dónde?
Gira la válvula-1 hacia la izquierda.	Incorrecto.	Hacia la derecha.
La válvula-1 no está atascada.	Falso negativo.	La válvula está atascada.
La válvula-1 está atascada.	Falso positivo.	La válvula no está atascada.
Gira la válvula-1 a 5.	Imprecisión.	Lo correcto es a 5.4.
Gira la válvula-1 a 5.4.	Inexactitud.	Lo correcto es a 9.2.

Gira la válvula-1 a 5.4, a 6 o a 0.	No confiable.	Error del equipo.
La válvula-1 está en 5.4, 5.5 o 5.1.	Aleatorio.	Fluctuación estadística.
La válvula-1 está en 7.5.	Error sistemático.	Falta de calibración.
La válvula-1 no está atascada porque nunca ha estado atascada antes.	Inducción inválida.	Válvula está atascada.
La salida de la válvula-1 es normal y por lo tanto la válvula-1 está en buenas condiciones.	Deducción inválida.	Válvula atascada en la posición de apertura.

### Errores e inducción

Mientras que la deducción va de lo general a lo específico, como es

Todos los hombres son mortales

Sócrates es un hombre

para deducir una conclusión específica

Sócrates es mortal

La inducción intenta generalizar a partir de lo específico a lo general, por ejemplo

Mi disco nunca ha fallado

Mi disco nunca fallará

No podemos confiar el argumento inductivo, pero en ocasiones puede resultar en un fuerte argumento.

La alarma de incendio se enciende

Hay fuego

y aún más peligro si

La alarma de fuego se enciende

Huele a humo

Hay fuego

Sin embargo, aun siendo un fuerte argumento, no podemos probar que hay fuego.

### Probabilidad clásica

Resultados de lanzar un dado 1,000,000 veces.

A priori:  $P(1) = 1/6 = 0.166\dots$

	1	2	3	4	5	6
Probabilidad	0.155	0.159	0.164	0.169	0.174	0.179

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

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

## Clase del 19/09/2022

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

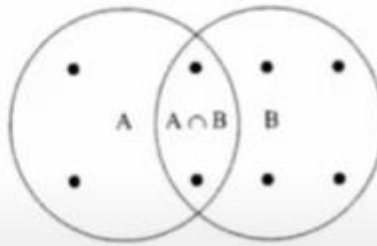
axioma 3:  $P(E_1 \cup E_2) = P(E_1) + P(E_2)$  para eventos mutuamente excluyentes.

## Probabilidad condicional

### Ley multiplicativa

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

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



## 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:

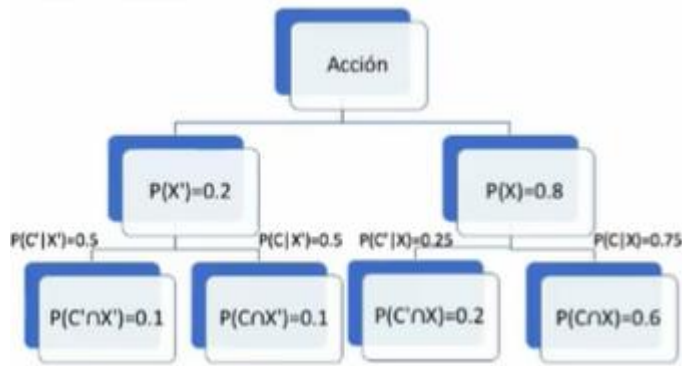
- $P(A_i)$  son las probabilidades a priori,
- $P(B|A_i)$  es la probabilidad de B en la hipótesis  $A_i$ ,
- $P(A_i|B)$  son las probabilidades a posteriori.

	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

$$P(C|X) = \frac{P(C \cap X)}{P(X)} = \frac{0.6}{0.8} = 0.75$$

$$P(C|X') = \frac{P(C \cap X')}{P(X')} = \frac{0.1}{0.2} = 0.50$$

Ejemplo del Teorema de Bayes para razonamiento bajo incertidumbre



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

$$P(X'|C') = \frac{0.1}{0.1+0.2} = \frac{1}{3} \quad P(X'|C) = \frac{0.1}{0.1+0.6} = \frac{1}{7} \quad P(X|C') = \frac{0.2}{0.2+0.1} = \frac{2}{3} \quad P(X|C) = \frac{0.6}{0.6+0.1} = \frac{6}{7}$$

### 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.

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

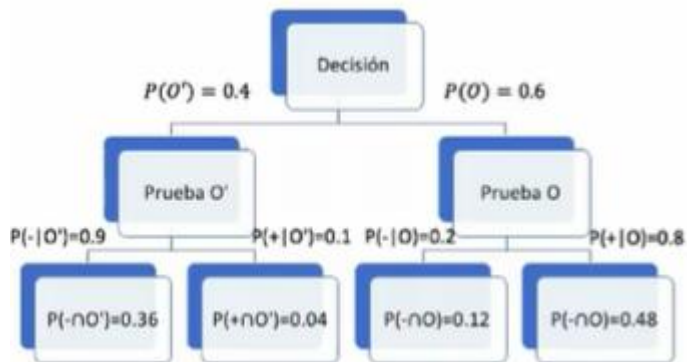
Una manera de encontrar oro y minerales es el estudio sísmico, en el cual, dinamita o maquinaria crea impulsos de sonido que viajan a través de la tierra. Las ondas de sonido son detectadas por micrófonos en distintas ubicaciones. Al observar el tiempo de llegada y la distorsión de la onda, es posible determinar estructuras geológicas y la posibilidad de petróleo y minerales. Desafortunadamente esta prueba no es 100 por ciento precisa. El sonido de las ondas puede ser afectado por algunos tipos de estructuras geológicas causando que se reporte la presencia de petróleo donde no lo hay (falso positivo), o bien, quizá reporte la ausencia de

petróleo donde lo hay (falso negativo). Asuma que en las siguientes probabilidades son resultados de los estudios sísmicos.

$$P(+|O) = 0.8 \quad P(-|O) = 0.2 \text{ (falso negativo)}$$

$$P(+|O') = 0.1 \text{ (falso positivo)} \quad P(-|O') = 0.9$$

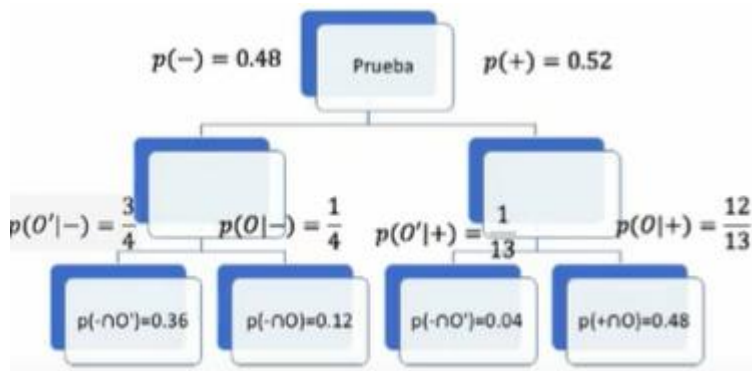
Entonces nuestro árbol inicial quedaría como



La ley de adición de las probabilidades nos dice

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

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



## Clase del 21/09/2022

## Razonamiento temporal y cadenas de Márkov

Razonar acerca de eventos que dependen del tiempo es llamado razonamiento temporal, una manera de lograrlo es con probabilidades.

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

		Futuro	
		$S_1$	$S_2$
Presente	$S_1$	$P_{11}$	$P_{12}$
	$S_2$	$P_{21}$	$P_{22}$

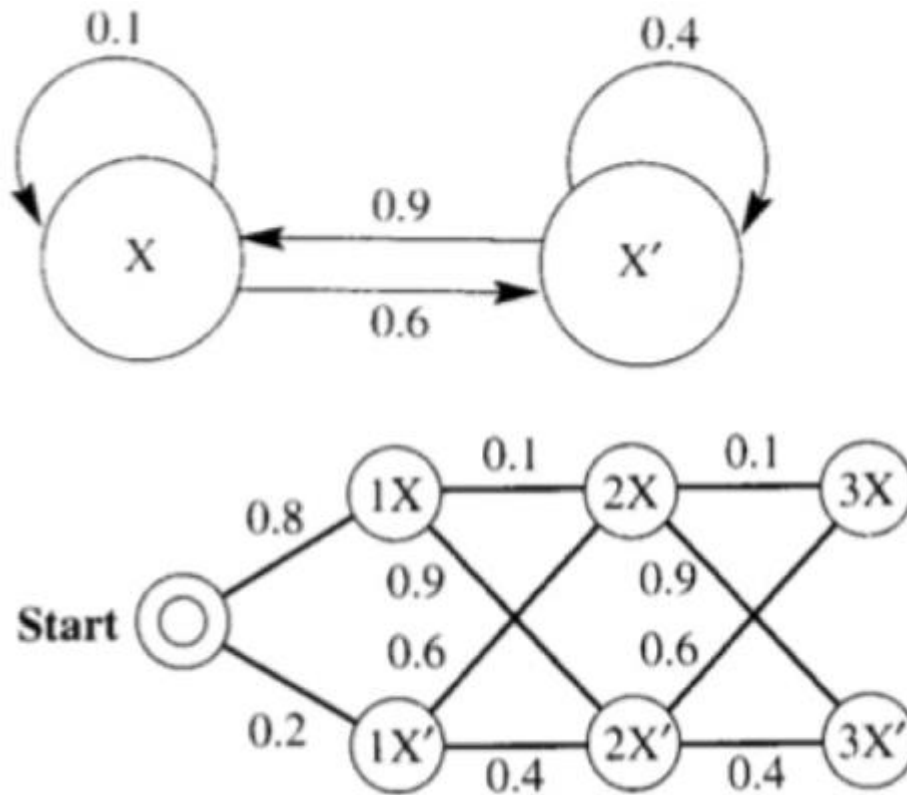
donde el elemento  $P_{mn}$  es la probabilidad de transición del estado  $m$  al estado  $n$ .

## Ejemplo

Asuma que el 10 por ciento de toda la gente que usa la marca  $X$  comprará 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.

		$X$	$X'$
$T=X$	$X$	0.1	0.9
	$X'$	0.6	0.4





(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.

$$S = [P_1 P_2 \dots P_n]$$

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

$$S_2 = S_1 T$$

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

$$S_3 = S_2T = [0.5 \ 0.05]$$

$$S_4 = S_3T = [0.35 \ 0.65]$$

$$S_5 = S_4T = [0.425 \ 0.575]$$

$$S_6 = S_5T = [0.3875 \ 0.6125]$$

$$S_7 = S_6T = [0.40625 \ 0.59375]$$

$$S_8 = S_7T = [0.396875 \ 0.603125]$$

$$S = ST = [0.4 \ 0.06]$$

En el cual llegamos a un vector de estado estable, esto es,

$$S = ST$$

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

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

El ejemplo de la marca X es un proceso en cadena de Márkov y la matriz de estado estable puede ser encontrada directamente

$$[X \quad Y] \begin{bmatrix} 0.1 & 0.9 \\ 0.6 & 0.4 \end{bmatrix} = [X \quad Y]$$

$$\begin{aligned} 0.1X + 0.6Y &= X \\ 0.9X + 0.4Y &= Y \end{aligned}$$

## Clase del 26/09/2022

### 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 intermedio 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.
- Pueden contener reglas sujetas a incertidumbre.
- El sistema cuenta con una interfaz de usuario.
- Tiene la capacidad de explicar cómo o por qué se alcanzó alguna conclusión.
- Soporta la adquisición de nuevo conocimiento.

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 staphylococcus

En cláusulas definidas

identidad\_del\_organismo(staphylococcus):-

mancha\_del\_organismo(gram\_positiva),

morfologia\_del\_organismo(coccus),

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

El (0.7) dado en la conclusión de la regla de MYCIN, es un ejemplo de incertidumbre de la regla.

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

- Si Y es un componente necesario para X e Y presenta un mal funcionamiento entonces X presenta también un mal funcionamiento;
- Si X exhibe un síntoma de fallo Z entonces 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).

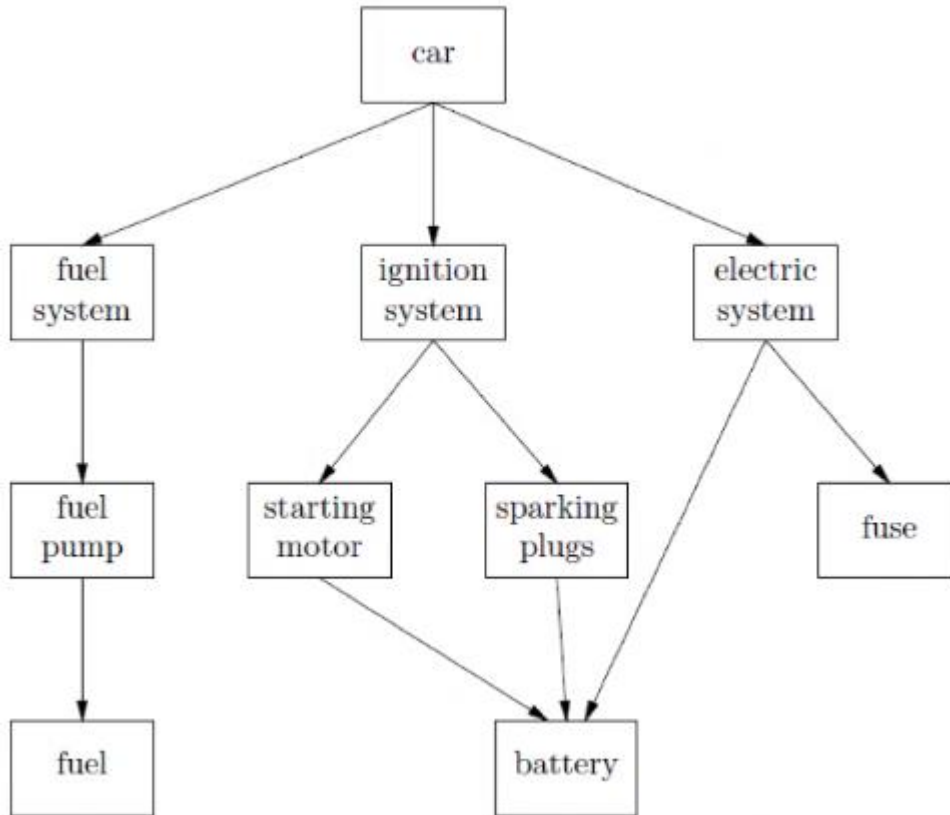
Para la segunda cláusula necesitamos el predicado indirecto/1.

- X tiene una falla indirecta si existe un componente el cual es necesario para X y que presenta un mal funcionamiento.

malFuncionamiento(X):-sintoma(Y, X), not(indirecto(X)).

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

Ejercicio escriba las cláusulas necesita/2 donde se especifica cuestiones como "necesita(carro, sistema\_de\_encendido)".



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).
- 2) 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).

Base de conocimiento

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

malFuncionamiento(X):-sintoma(Y,X),not(indirecto(X)).

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

necesita(carro,sistema\_de\_encendido).

necesita(carro,sistema\_de\_combustible).  
necesita(carro,sistema\_electrico).  
necesita(sistema\_de\_encendido,motor\_de\_arranque).  
necesita(sistema\_de\_encendido,bujias).  
necesita(sistema\_electrico,fusible).  
necesita(sistema\_electrico,bateria).  
necesita(sistema\_de\_combustible,bomba\_de\_combustible).  
necesita(bujias,bateria).  
necesita(motor\_de\_arranque,bateria).  
necesita(bomba\_de\_combustible,gasolina).

#### 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 e Y;
- La meta consistente de una sola variable.

Consideramos 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).

```

Consulta:

?- resuelve(abuelo(X,Y),Z).

```

prueba(abuelo(adan, catalina),
      prueba(progenitor(adan, guillermo),
            prueba(padre(adan, guillermo), verdad)
            ),
      prueba(progenitor(guillermo, catalina),
            prueba padre(guillermo, catalina), verdad)
      )
)

```

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).

Consulta:

?- malFuncionamiento(X).

Está la llanta ponchada?

Si

X=llanta

## Clase del 28/09/2022

### Razonamiento inexacto

#### Incertidumbre en reglas

- Reglas individuales.
  - Antecedente
    - Errores.
    - Probabilidad de la evidencia.
    - Combinado evidencia.

$$E_1 \text{ AND } E_2 \text{ AND } E_3$$
$$E_1 \text{ AND } E_2 \text{ OR } E_3$$
$$E_1 \text{ AND NOT } E_2 \text{ OR } E_3$$
  - Consecuente
    - Errores.
    - Probabilidad de la evidencia.
- Conflictos de resolución.
- Incompatibilidad de reglas.

#### Compatibilidad en reglas

- Prioridad de reglas.
- Contradicción de regla.
- 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.
  - Recientes hechos de coincidencia de patrones.
  - Ordenamiento de patrones.
    - Lexicográficos (LEX).
    - Análisis de medios y fines (MEA).
  - Ordenamiento de las reglas como son introducidas.
- Prioridad explícita de reglas.
  - 3) Si  $E_1$  entonces H.
  - 4) Si  $E_1$  y  $E_2$  entonces H.

En OPSS 0 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).

SI  $E_1$  Y  $E_2$ , ENTONCES H

SI  $E_2$  Y  $E_1$ , ENTONCES H

Las reglas anteriores son esencialmente las mismas bajo LEX. Mediante el análisis de medios y fines (MEA), el primer patrón tiene mayor prioridad.

## Clase del 03/10/2022

### 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.

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

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”.

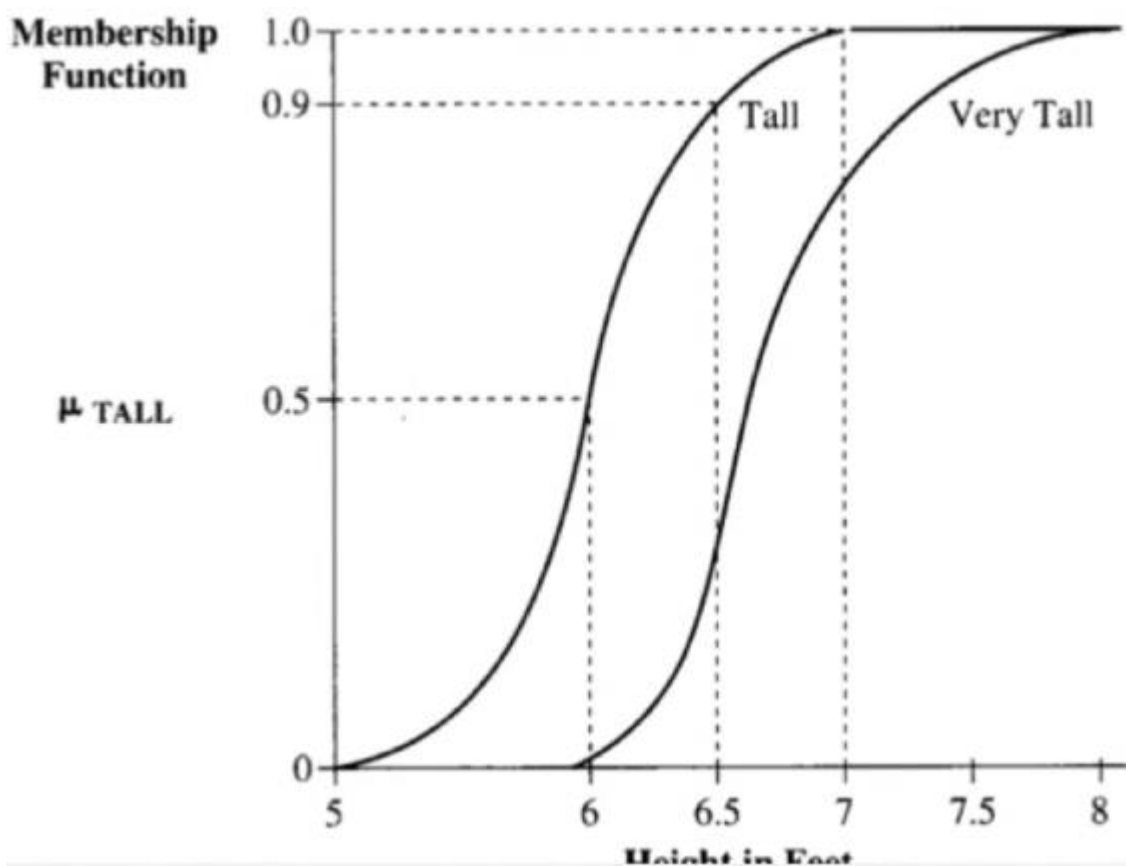
“El clima es cálido”.

“Gire la perilla un poco más”.

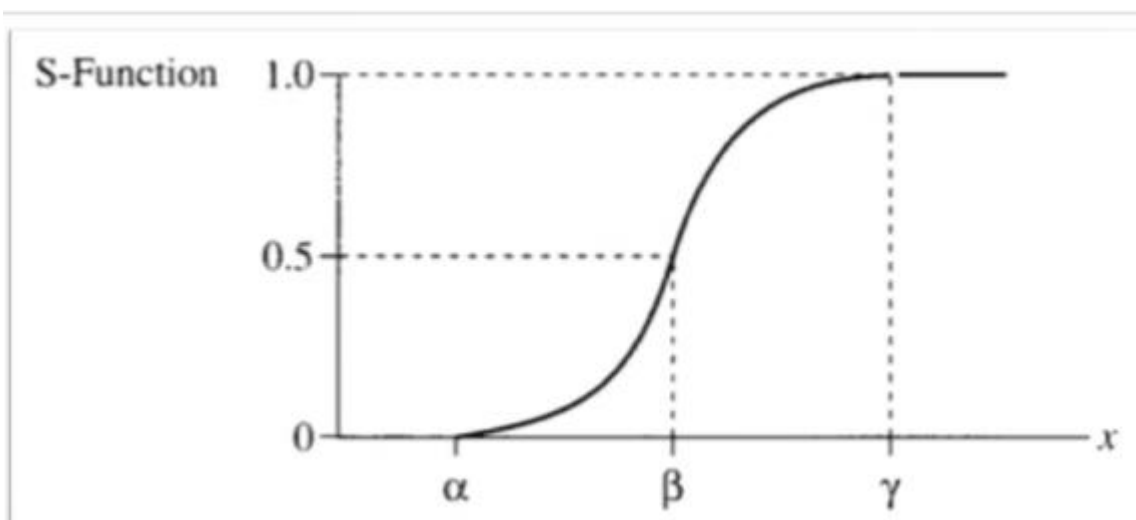
“La mayoría de los exámenes son difíciles”.

“Si la masa está muy dura, añada mucha agua”.

Una proposición difusa contiene la 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 usado con frecuencia en conjuntos difusos como función de membresía.

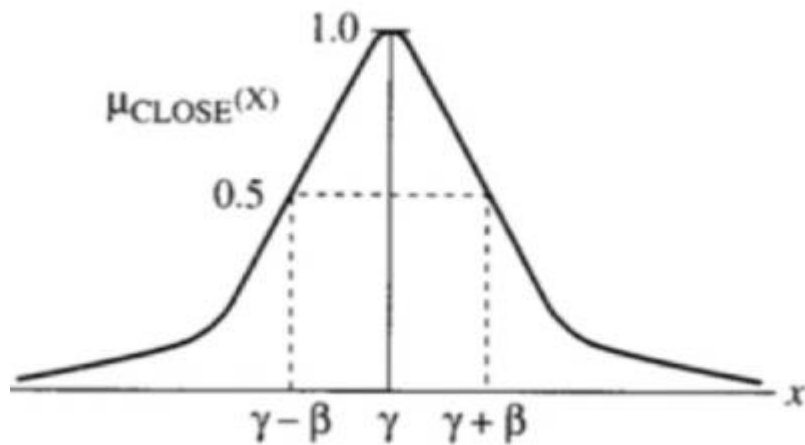


$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \text{for } x \leq \alpha \\ 2\left(\frac{x - \alpha}{\gamma - \alpha}\right)^2 & \text{for } \alpha \leq x \leq \beta \\ 1 - 2\left(\frac{x - \gamma}{\gamma - \alpha}\right)^2 & \text{for } \beta \leq x \leq \gamma \\ 1 & \text{for } x \geq \gamma \end{cases}$$

Para nuestro ejemplo, quedaría como

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

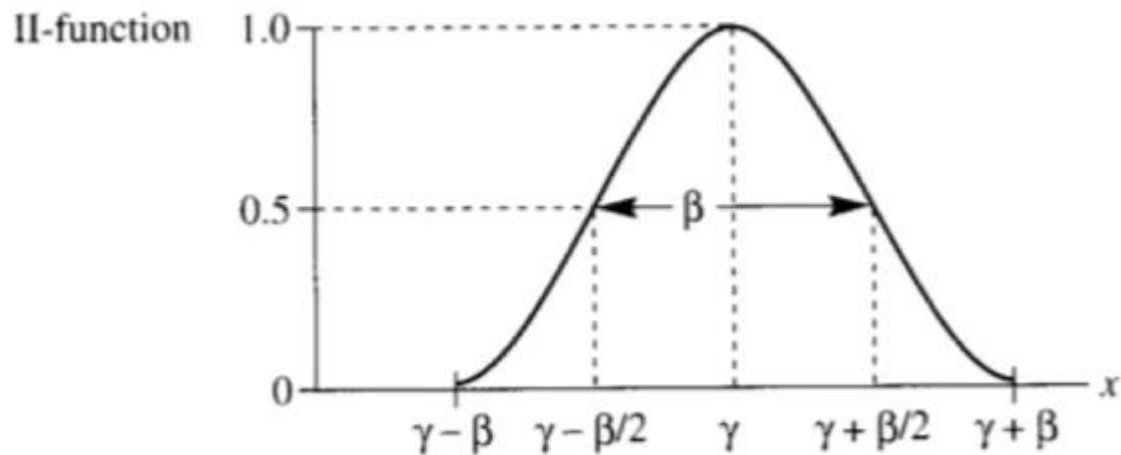
Una función de membresía para la proposición “X está cerca de Y” es



Por ejemplo, la función de membresía podría representar todos los números cercanos a un valor específico  $\gamma$ , como en “X está cerca de 6”, donde X podría ser {5.9, 6, 6.1}.

$$\mu_{\text{cerca}}(x) = \frac{1}{1 + \left(x - \frac{\gamma}{\beta}\right)^2}$$

Otra función con una curva similar es:



$$\Pi(x; \beta, \gamma) = \begin{cases} S(x; \gamma - \beta, \gamma - \beta/2, \gamma) & \text{for } x \leq \gamma \\ 1 - S(x; \gamma, \gamma + \beta/2, \gamma + \beta) & \text{for } x \geq \gamma \end{cases}$$

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 soporta el conjunto difuso. Para este subconjunto, soporta todos los elementos excepto el 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,

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

Ejercicio de clase, encuentre los siguientes conjuntos:

$$\text{Alto}_{0.1} = \{5.5, 6, 6.57, 7.5, 8\}$$

$$\text{Alto}_{0.5} = \{6, 6.57, 7.5, 8\}$$

$$\text{Alto}_{0.8} = \{6.5, 7, 7.5, 8\}$$

$$\text{Alto}_1 = \{7, 7.5, 8\}$$

### Tipos de conjuntos difusos

#### Tipo 1 de conjuntos difusos

Por ejemplo, el conjunto ALTO.  $\mu_f: x \rightarrow [0,1]$

#### Tipo 2 de conjuntos difusos

$$\mu_{\text{ALTO}}(5) = \text{MENOR AL PROMEDIO}$$

$$\mu_{\text{ALTO}}(6) = \text{PROMEDIO}$$

$$\mu_{\text{ALTO}}(7) = \text{MAYOR AL PROMEDIO}$$

Donde cada uno de estos términos representa un subconjunto de tipo 1.

$$\mu_{\text{LESS THAN AVERAGE}}(x) = 1 - S(x; 4.5, 5, 5.5)$$

$$\mu_{\text{AVERAGE}}(x) = \prod (x; 1, 5.5)$$

$$\mu_{\text{GREATER THAN AVERAGE}}(x) = S(x; 5.5, 6, 6.5)$$



## Operaciones de conjuntos difusos

## Igualdad de conjuntos

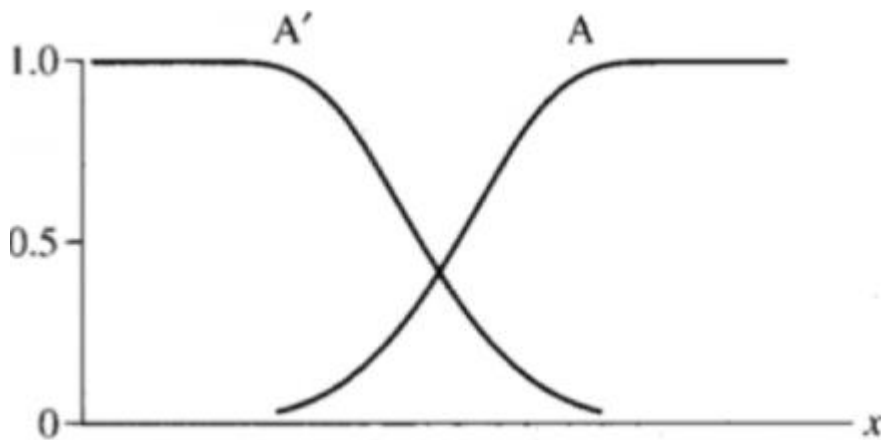
$$A=B$$

$$\mu_A(x) = \mu_B(x) \text{ para toda } x \in X$$

## Complemento de conjunto

$$A'$$

$$\mu_{A'}(x) = 1 - \mu_A(x) \text{ para toda } x \in X$$



## Pertenencia de conjuntos

$$A \subseteq B$$

El conjunto difuso A es un subconjunto de B si y solo si

$$\mu_A(x) \leq \mu_B(x) \text{ para todo } x \in X$$

A es un subconjunto propio de B si A es subconjunto de B y, además

$$\mu_A(x) \leq \mu_B(x) \text{ y } \mu_A(x) < \mu_B(x) \\ \text{para al menos una } x \in X$$

## Unión de conjuntos

$$A \cup B$$

$$\mu_{A \cup B}(x) = \vee (\mu_A(x), \mu_B(x)) \text{ para todo } x \in X$$

donde el operador de unión  $\vee$ , significa el máximo de los argumentos.

Intersección de conjuntos

$$A \cap B$$

$$\mu_{A \cap B}(x) = \wedge (\mu_A(x), \mu_B(x)) \text{ para todo } x \in X$$

donde el operador de intersección  $\wedge$ , significa el mínimo de los argumentos.

Bajo estas definiciones las leyes estándares de los conjuntos clásicos, como conmutatividad y asociatividad, entre otros, son aplicables a conjuntos difusos.

Excluyendo reglas como la de contradicción o la del tercero excluido.

Para un conjunto difuso

$$A \cup A' \neq U$$

$$A \cap A' \neq \emptyset$$

La única restricción en intersección es

$$A \cap A' = \min(\mu_A(x), \mu_{A'}(x)) \leq 0.5$$

si  $\mu(x)$  está definida en el rango cerrado  $[0, 1]$  y  $\mu_{A'}(x) = 1 - \mu_A(x)$ . Entonces la única restricción en la unión es

$$A \cup A' = \max(\mu_A(x), \mu_{A'}(x)) \geq 0.5$$

Producto de conjuntos

$$A \cdot B$$

$$\mu_{AB}(x) = \mu_A(x) \mu_B(x)$$

Conjunto potencia

$$A^N$$

$$\mu_{A^N} = (\mu_A(x))^N$$

Suma probabilística

$$A \hat{+} B$$

$$\begin{aligned}\mu_{A \hat{+} B}(x) &= \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \\ &= 1 - (1 - \mu_A(x))(1 - \mu_B(x))\end{aligned}$$

Suma acotada

$$A \oplus B$$

$$\mu_{A \oplus B}(x) = \wedge (1, (\mu_A(x) + \mu_B(x)))$$

Producto acotado

$$A \odot B$$

$$\mu_{A \odot B}(x) = \vee (0, (\mu_A(x) + \mu_B(x) - 1))$$

Diferencia acotada

$$A \mid - \mid B$$

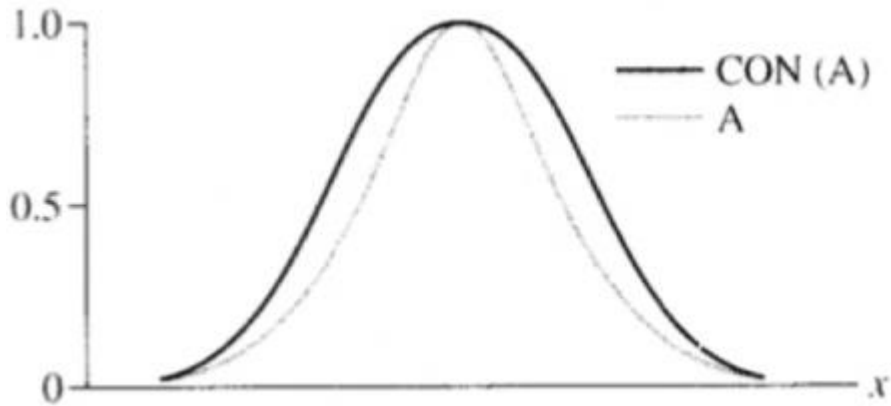
$$\mu_{A \mid - \mid B} = \vee (0, (\mu_A(x) - \mu_B(x)))$$

Concentración

CON(A)

$$\mu_{CON(A)}(x) = (\mu_A(x))^2$$

La operación CON puede ser usada como el efecto lingüístico “muy”.

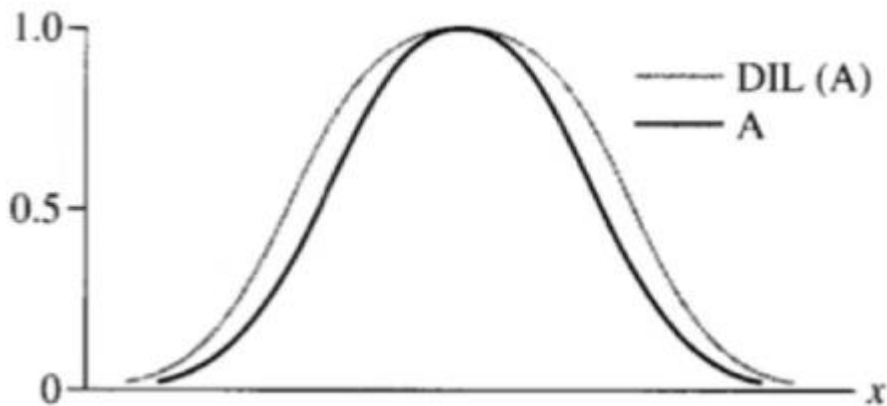


Dilatación

DIL(A)

$$\mu_{DIL(A)}(x) = (\mu_A(x))^{0.5}$$

La operación de dilatación puede ser usada como “Más o menos”.



Intensificación

INT(A)

$$\mu_{INT(A)}(x) = \begin{cases} 2(\mu_A(x))^2 & \text{for } 0 \leq \mu_A(x) \leq 0.5 \\ 1 - 2(1 - \mu_A(x))^2 & \text{for } 0.5 \leq \mu_A(x) \leq 1 \end{cases}$$



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.

Normalización

NORM(A)

$$\mu_{NORM(A)}(x) = \mu_x / \max (\mu_{A(x)})$$