

# TEORÍA DE SIN

## TEMA 1: INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

### ➤ Concepto de Inteligencia Artificial

- Ciencia que intenta crear programas para máquinas que imiten el comportamiento y la comprensión humano.
- Conjunto de técnicas aplicadas a una máquina para desarrollar la capacidad de éste de percibir, razonar y actuar.
- La IA tiene como objetivo el desarrollo de programas o aplicaciones capaces de realizar operaciones propias del ser humano. Generalmente, las personas pueden decir lo que piensan pero no saben explicar cómo piensan.

### ➔ Usos y utilidades

- Control de sistemas.
- Diagnósticos.
- Reconocimiento de escritura, habla o patrones.
- Forma parte de la:
- Economía.
- Medicina.
- Milicia.
- Aplicaciones software y juegos. ➤ Dos perspectivas

1. IA como ciencia **natural**: Basándose desde un punto de vista teórico analizando los fundamentos en los que se basa para su desarrollo ( análisis ). / ESTUDIO
2. IA como ciencia **artificial**: Basándose desde un punto de vista práctico plantea como deben de elaborarse los programas informáticos que sean capaces de resolver un problema. / IMPLEMENTACIÓN.

### ➤ IA desde una perspectiva sintética

- **Objetivo**: Diseñar un programa sobre una máquina específica desarrollado a partir del conocimiento que supuestamente usaba el operador humano que realizaba esa tarea.

- **El conocimiento deberá ser:** Identificado, modelado, representado y utilizado en la inferencia.

- **Metodología del trabajo:** Propia de las ingenierías.

1. Planteamiento del problema.
2. Estudio del problema para elaborar una especificación formal.
3. Análisis y diseño de una solución.
4. Desarrollo y validación de la solución, siendo típico el desarrollo de prototipos.

➤ **Tipos de tareas sintéticas en IA**

1. **Tareas en dominios formales:** Buscan la solución de problemas juguetes.

➔ **Propiedades:**

1. No hay imprecisión en el conocimiento.
2. Pocos elementos en el dominio del problema.
3. La tarea a desarrollar puede ser descrita de forma completa e inequívoca.
4. Los problemas representan micromundos formales.
5. Resultados difíciles de extrapolar a otras situaciones reales.
6. Métodos para encontrar y establecer soluciones a un problema pueden ser valiosos para abordar problemas del mundo real.
7. Problemas se resuelven mediante la búsqueda en un espacio de estados ( Búsqueda ciega o heurística ).

2. **Tareas en dominios técnicos:** Tareas genéricas válidas en muchas aplicaciones propias de cada dominio del problema/aplicación.

➔ **Propiedades**

1. Requieren un conocimiento limitado.
2. Dan lugar a la Ingeniería del conocimiento.
3. Se realiza un proceso de obtención del conocimiento a partir de un experto ( análisis ).
4. Busca una representación del conocimiento obtenido, separándolo en la medida de lo posible de los mecanismos de aplicación del mismo ( Representación ).
5. Se seleccionan las técnicas para su implementación y se desarrolla un prototipo mediante herramientas de ayuda ( diseño ).

6. Buscan procedimientos de implementación, evaluación y refinamiento de los prototipos ( ingeniería/implementación ).
7. Lenguajes y entornos de programación que facilite el desarrollo ( elaboración ).

### **3. Tareas genuinas del comportamiento humano**

- ➔ Conllevan un comportamiento artificial: Busca la síntesis de las funciones que usan la neurofisiología y la psicología para describir el comportamiento humano.
- ➔ El objetivo de la IA es comprender y duplicar las funciones que caracterizan el comportamiento humano.

#### **➔ Propiedades**

1. En el ser vivo se desarrollan de forma aparentemente sencilla.
2. Enormemente complejas a la hora de intentar implementarlas.
3. Necesidad de un uso masivo de conocimiento, representado mediante lenguaje natural.
4. Se cree que el conocimiento depende de la estructura que lo mantiene.
5. Hipótesis: Estas tareas son computacionalmente posibles.

#### **➤ Clasificación**

1. Tareas perceptoras : A través de un conjunto de sensores se representa físicamente las configuraciones espacio-temporales del medio. Se realiza un procesamiento, cuyo objetivo es identificar al medio de acuerdo con un modelo de representación interna para su entendimiento. ( CAPTACIÓN ).
2. Tareas motoras : Inciden en la realización/manipulación de la tarea ( REALIZA ).
3. Tareas de decisión : Están entre las tareas perceptoras y motoras, incluyendo los procesos cognoscitivos asociados al pensamiento. ( traducen/mandan la tarea ).

#### **➤ Perspectiva Histórica. ¿Qué significa ser inteligente?**

- Se proponen 3 niveles tales que:

1. Nivel 1. Teoría computacional : Planteamiento del problema en lenguaje natural y un posible esquema de solución en términos del conocimiento humano.
2. Nivel 2. Representación y algoritmo : Elección de un lenguaje de representación para los espacios de entrada-salida y de un algoritmo que haga efectivas las transformaciones que enlazan ambas.

3. Nivel 3. Implementación: Proceso que nos lleva desde el algoritmo hasta los procesadores físicos. Incluye la selección del lenguaje de programación y la codificación.

- Agentes inteligentes

➔ **Agente**: Cualquier cosa capaz de percibir su entorno con la ayuda de sensores y actuar mediante actuadores.

➔ **Agente racional**: Es una entidad que tiene la capacidad de percibir su entorno, siendo capaz de tomar acciones por su propia cuenta de manera racional con el fin de obtener el mejor resultado posible.

➔ Podemos encontrar varios tipos de agentes:

- Agentes que aprenden.
- Agentes basados en objetivos.
- Agentes basados en utilidad.

## TEMA 2-3. RESOLUCIÓN DE PROBLEMAS CON TÉCNICAS DE BÚSQUEDA

➔ **Condiciones necesarias:**

- Representación de los estados (  $S$  ).
- Estados iniciales (  $S_0$  ).
- Representación de acciones (  $A: s \rightarrow S'$  );
- Estado/Función objetivo (  $s^* / O: s \rightarrow T \mid F$  )
- Función de coste (  $F: \{a_1, \dots, a_N\} \rightarrow R$  )
- Función mejor padre (  $P: s \rightarrow s'$  )
- Objetivo: Encontrar un camino  $\{ a_1, \dots, a_N \}$  que conecte el estado inicial con el estado final.

➔ **Búsqueda sobre árboles**

1. Frontera =  $\{ S_0 \}$
2. Mientras que la frontera no esté vacía
  - 1) actual = extraer\_estado (Frontera);
  - 2) Si  $O(\text{actual}) \rightarrow$  Devolver camino;

3) Para cada posible acción  $a_i$  desde actual

➔ Añadir estado  $a_i$  (actual) en frontera;

3. Devolver fallo;

### ➔ Métodos de búsqueda en amplitud/anchura

➤ Característica de implementación: Frontera es una cola FIFO.

➤ Se selecciona el estado que antes se introdujo en la frontera.

➤ Se selecciona uno de los estados correspondiente al camino de menor longitud.

➤ **Ventajas:** Método completo y óptimo.

➤ **Desventajas:**

1. Gran complejidad temporal y espacial  $O(n^p)$ .

2. Consume mucha memoria.

➤ Ramas infinitas: No le afectan por ser completo y óptimo.

### ➔ Métodos de búsqueda en profundidad:

➤ **Característica implementación:** Frontera es una pila.

➤ **Ventajas:**

1. Complejidad espacial reducida  $O(n \cdot p)$ ,

2. Permite ahorrar memoria.

➤ **Desventajas:**

1. No es óptimo.

2. No es complejo, pudiendo perderse por ramas infinitas.

3. Gran complejidad temporal

### ➔ Búsqueda de coste uniforme

➤ Comienza por el nodo raíz y continúa visitando el siguiente nodo que tiene menor coste total desde la raíz.

➤ Los nodos son visitados de esta manera hasta que el nodo destino es alcanzado.

➤

### ➔ Búsqueda en profundidad limitado

➤ **Característica implementación:** Tiene un límite de profundidad a partir del cual no se generan más hijos.

➤ **Consideraciones**

1. Si  $l < d$  el objetivo está fuera del límite de profundidad.
2. Si  $l > d$  estamos cogiendo un límite superior al  $n^o$  de niveles.

➤ **Dos tipos de fallo**

1. Sin solución.
2. Sin solución dentro del límite de profundidad.

➤ **Ventajas:**

1. No se pierde por ramas infinitas.
2. Complejidad en el espacio  $O(b \cdot l)$ .

➤ **Desventajas**

1. No es óptimo si el nodo meta puede no ser de profundidad mínima.
2. Es común que los límites buenos de profundidad solo se puedan establecer una vez resuelto el problema.
3. No es completo si la solución se encuentra por debajo del límite.

➔ **Búsqueda primero en profundidad iterativa**

- **Característica implementación:** Aplica una búsqueda en profundidad limitada incrementando el límite si no se encuentra la solución.

➤ **Ventaja:** Es óptima.

➤ **Desventaja:** Tarda mucho.

➔ **Búsqueda en profundidad con retroceso**

- **Característica de implementación:** Genera sólo 1 hijo por nodo y por iteración.
- Cuanto mejor sea el criterio para limitar el número de estados considerados, más eficiente será el proceso de búsqueda.

➤ **Ventaja:** Ahorra más memoria que la búsqueda en profundidad.

➤ **Desventaja:** Se puede perder por ramas infinitas. ➔ **Búsqueda bidireccional**

- **Característica de implementación:** Aplica 2 búsquedas desde estado inicial y estado final.

➤ **Ventaja:** Tarda menos.

➤ **Desventaja:** Necesita conocer el estado final.

➤ **Ramas infinitas:** No le afectan si una búsqueda es en anchura.

En las búsquedas comentadas anteriormente, depende del tipo de problema, pues existen problemas como las 8 reinas que no tienen ramas infinitas.

### ➔ **Métodos de búsqueda heurísticos ➤ Búsqueda primero el mejor**

- Se seleccionará primero el nodo que se cree más cercano del objetivo.
- Trata de expandir el nodo más cercano al objetivo, alegando que probablemente conduzca rápidamente a una solución.
- De este modo, evalúa los nodos utilizando solamente la función heurística:  $f(n) = h(n)$ .

### ➔ **Búsqueda A\***

- Es una búsqueda completa debido a que siempre que exista una solución, dará como resultado el camino más corto al nodo objetivo.
- Es óptimo en búsqueda en árboles siempre y cuando se elija una heurística admisible  $h(n)$ , es decir,  $h(n)$  nunca sobrestime el coste de alcanzar el objetivo.
- En la búsqueda en grafos  $h(n)$  debe ser consistente, es decir, para cualquier nodo  $n$  y sus hijos el coste estimado de alcanzar el objetivo desde  $n$  es menor al de alcanzar el hijo más el coste de alcanzar el objetivo desde éste.
- Se calculan los costes estimados  $f(n)$  de cada nodo en Abierto, se deberá seleccionar un nodo de la cola Abiertos para pasarlo a cerrado.
- En caso de existir varios nodos con el mismo valor de  $f(n)$ , se deberá pasar a cerrado el que llevase más tiempo en la cola ( FIFO ).
- La búsqueda finaliza una vez se pasa a Cerrado el nodo objetivo, determinando así el camino más corto desde el punto inicial al objetivo.
- Es una variante del método Primero el mejor, empleando una función heurística:  $f(n) = g(n) + h(n)$ , donde:
  - $f(n)$  = Coste más barato estimado de la solución a través de  $n$ .
  - $g(n)$  = Coste real del mejor camino encontrado en un determinado momento desde la raíz hasta  $n$ .

- $h(n)$  = Es el valor heurístico del nodo  $n$ . Presenta una estimación del coste del camino óptimo desde  $n$  al objetivo.

### ➔ **Búsqueda por adversarios: Minimax**

- Es un método de búsqueda en el que 2 jugadores Max ( jugador que busca el valor más alto, suele asociarse a la propia persona ) y Min ( jugador que busca el valor más bajo ).
- Consiste en determinar al jugador que le toque jugar escoger el valor más alto o más bajo de los nodos hoja y subirlo al nodo padre. ➔ **Búsqueda por adversarios: Poda alfa-beta**
- Es un método de búsqueda similar al minimax, pero cada nodo que no sea hoja llevará asociado unos límites que heredarán del nodo padre y estos del nodo raíz, siendo por defecto  $[-\infty, \infty]$ .
- Donde  $[\alpha = -\infty, \beta = \infty]$ . Cada vez que estemos en un nodo Max editamos el valor de  $\alpha$ , y cada vez que estemos en un nodo Min editamos el valor de  $\beta$ .
- Permite ahorrar mucho más tiempo que minimax, pues una vez encuentre una rama no rentable de realizar la búsqueda, aplicará una poda ignorando la búsqueda de dicha rama.
- Su eficacia depende del orden en que se examinan los sucesores.

## TEMA 4. REPRESENTACIÓN DEL CONOCIMIENTO CON LÓGICA DE PREDICADOS

### ➤ **Conectivas**



	Símbolo	Nombre	Lectura
<b>Singular</b>	$\neg p$	Negación	No p
<b>Binarias</b>	$p \vee q$	Disyunción	p o q
	$p \wedge q$	Conjunción	p y q
	$p \rightarrow q$	Condicional	Si p entonces q p solo si q p es condición suficiente para q q si p q cuando p
	$p \leftrightarrow q$	Bicondicional	p si y solo si q
	$p \oplus q$	Disyunción Exclusiva	

### ➤ Propiedades deducidas

Idempotente	$P \wedge P \equiv P$	$P \vee P \equiv P$
Ley de acotación	$P \vee \top \equiv \top$	$P \wedge \emptyset \equiv \emptyset$
Ley de absorción	$P \vee (P \wedge Q) \equiv P$	$P \wedge (P \vee Q) \equiv P$
Ley de involución	$\neg(\neg P) \equiv P$	
Se cumple que	$\neg \emptyset \equiv \top$	$\neg \top \equiv \emptyset$
Ley de Morgan	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$	$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
Distributiva	$A \wedge (B \vee C)$	$(A \wedge B) \vee (A \wedge C)$
Bicondicional	$A \leftrightarrow B$	$(A \rightarrow B) \wedge (B \rightarrow A)$

### ➤ Cuantificadores

➤ Aparecen junto a las variables, teniendo asociados un ámbito:

- ❖ **Cuantificador Universal  $\forall$** : Puede interpretarse como "Para todo".  $\forall ( ) \rightarrow ( )$
- ❖ **Cuantificador Existencial  $\exists$** : Se interpreta como "Existe al menos un".  $\exists ( ) \wedge ( )$

### ➤ Equivalencias

$$\begin{aligned}\forall x p(x) &\equiv \neg(\exists x \neg p(x)) \\ \exists x p(x) &\equiv \neg(\forall x \neg p(x)) \\ \neg(\forall x p(x)) &\equiv \exists x \neg p(x)\end{aligned}$$

➤ **Predicados:**

- No pueden cuantificarse ni usarse como variables.
- Sólo retornan Verdadero o falso.
- Estructura: Propiedad/relación (variables).
- Es la formalización de una propiedad o relación.

➤ **Funciones**

- Semántica: Nombre ( $A_1, A_2, \dots, A_n$ )
- Formato: Similar al de los predicados, pero presentan valores (True/false).
- Retornan un valor de cualquier tipo.

➤ **Lógica de clausulas**

- Clausula: Fórmula lógica constituida por una disyunción de literales, en que cada literal  $L_i$  es una fórmula atómica negada (literal negativo) o no negada (literal positivo) :  $L_1 \vee \dots \vee L_n$ .
- Orden en que aparecen no es relevante y se puede representar la misma clausula como un conjunto de literales:  $\{L_1 \vee \dots \vee L_n\}$ .
- Si el conjunto está vacío, la clausula se denomina clausula vacía y se simboliza por  $\{\}$ .

➤ **Sintaxis de lógica Clausal**

- Un átomo es una variable proposicional:  $p, q, \dots$
- Un literal es un átomo o la negación de un átomo:  $L, L_1, \dots$
- Una clausula es un conjunto finito de literales:  $C, C_1, \dots$
- La clausula vacía es el conjunto vacío de literales
- Conjuntos finitos de clausulas:  $S, S_1, \dots$

➤ **Algoritmo de Unificación**

- Unificación: Proceso que consiste en encontrar una asignación de variables que haga idénticas a las fórmulas que se desea unificar.
- Su resultado se expresa como un conjunto de pares sustitución/variable para cada una de sus variables asignadas.

➤ **Normalización a Forma Clausal o Conjuntiva FNC**

- Para aplicar el método de demostración por refutación es necesario que todas las fórmulas lógicas involucradas en el problema se puedan re-escribir en forma de clausulas semánticamente equivalentes.
- Esta normalización se puede llevar a cabo mediante un algoritmo de transformación que consta de los siguientes pasos:

1. Eliminar condicionales  $(A \rightarrow B) = (\neg A \vee B)$
2. Anidar negaciones ( $\neg$ )
  - a.  $\neg(\neg A) = A$
  - b.  $\neg(A \wedge B) = (\neg A \vee \neg B)$
  - c.  $\neg(A \vee B) = (\neg A \wedge \neg B)$
  - d.  $\neg(x) a(x) \equiv (\forall x) \neg a(x)$
  - e.  $\neg(\forall x) b(x) \equiv (\exists x) \neg b(x)$
3. Se estandarizan los nombres de variables colocándole nombres distintos a variables asociadas con cuantificadores distintos.
4. Mover todos los cuantificadores a la izquierda sin cambiar su orden relativo (clausula en forma normal prenex)
5. Eliminar cuantificadores existenciales, mediante skolemización, consistente en reemplazar las variables existenciales por:
  - a. Una constante  $c$  si el cuantificador existencial asociado no se encuentra dentro del alcance de ningún cuantificador universal
  - b. Una función  $f(x)$  si el cuantificador existencial se encuentra dentro del alcance del cuantificador universal correspondiente a  $x$
  - c. Una función  $f(x, y)$  si el cuantificador existencial se encuentra dentro del alcance de los cuantificadores de  $x$  e  $y$ , etc.
6. Eliminar cuantificadores universales
7. Convertir expresiones en una conjunción de disyunciones (usar asociativa y distributiva):  $(A \vee (B \wedge C)) = ((A \vee B) \wedge (A \vee C))$
8. Separar las conjunciones, generando una cláusula por cada conjunción.
9. Estandarizar variables.

## ➤ Principio de Resolución

- Resolución: Técnica utilizada para probar teoremas en lógica, utilizando refutación para comprobar una sentencia determinada.
- Refutación: Se basa en negar la sentencia original para intentar crear una contradicción, demostrando de este modo que la sentencia original es verdadera. Se aplica a sentencias escritas en forma de clausula.

## TEMA 5. REPRESENTACIÓN DEL CONOCIMIENTO CON REGLAS

### ➔ Reglas

- ➔ Los sistemas basados en reglas analizan datos y preguntan por nueva información hasta alcanzar una conclusión.
- ➔ Los sistemas expertos actuales utilizan reglas, marcos, lógica y otros conceptos de los lenguajes ontológicos.
- ➔ **Componentes básicos**
- ➔ Motor de inferencia : Coordina la información procedente del resto de módulos.
- ➔ Base de conocimiento : Contiene las reglas utilizadas para representar el conocimiento disponible en un determinado dominio.
- ➔ Base de afirmaciones : Contiene los hechos/afirmaciones conocidos inicialmente y aquellos que se van creando en el proceso de inferencia.
- ➔ Interfaz de usuario : A través del cual se puede solicitar u ofrecer información al usuario.
- ➔ Una regla constará de 2 partes:
- ➔ Antecedente : Figuran cláusulas o condiciones que se deberá cumplir para que la regla sea ejecutada.
- ➔ Consecuente : Conclusiones a desarrollar cuando la regla sea ejecutada.
- ➔ **Tipos de encadenamiento**
- ➔ El motor de inferencia será el encargado de llevar a cabo el proceso de establecer las conclusiones a partir de la base de conocimientos, pudiendo realizarlo de 2 formas:
  1. Encadenamiento hacia delante : Se basa en ejecutar aquellas reglas cuyo antecedente sea cierto a partir de la información que hay en el sistema. Por otra parte, la introducción de nueva información a partir del consecuente de cada regla ejecutada permitirá la ejecución de otras reglas.
  2. Encadenamiento hacia atrás : Consiste en seleccionar aquellas reglas cuyo consecuente permita demostrar cierta condición, si ésta no puede ser demostrada a partir de la base de afirmaciones.

A su vez, las condiciones que estén en los antecedentes de las reglas seleccionadas pasarán a convertirse en nuevos sub-objetivos a demostrar, entrando en un proceso recursivo que finaliza cuando se encuentra información buscada en la base de afirmaciones.

## ➔ Tipos de dependencia

1. Dependencia reversible : Si la premisa se cumple el consecuente se activa, y si la premisa se retracta el consecuente se retracta.
2. Dependencia irreversible : Si la premisa se cumple el consecuente se activa, pero si la premisa se retracta no tiene por qué hacerlo el consecuente.

## ➔ ¿Qué ejecutar primero?

➔ Existen distintas estrategias:

1. Ordenación de reglas : Se colocan en primer lugar las reglas que se desea examinar antes.
2. Control de agendas : El conjunto de reglas listas para ser ejecutadas pueden reunirse en una agenda en la que se ordenan según su valor de prioridad.
3. Criterio de actualidad : Consiste en ejecutar primero aquellas reglas cuyo antecedente se apoya en información más reciente.
4. Criterio de especificidad : Se prefieren para su ejecución las reglas más específicas.
5. Mundo cerrado : Consiste en, dado un hecho que no podemos demostrar si es cierto o no, se considera como cierto su negación.

## TEMA 6. REPRESENTACIÓN DEL CONOCIMIENTO CON REDES ASOCIATIVAS

### ➔ Redes asociativas

➔ Una red está formada por un conjunto de nodos unidos entre sí por enlaces de diferentes tipos.

➔ En las redes asociativas, cada nodo representa un concepto y los enlaces generalmente representan los tipos de relaciones de los nodos.

### ➔ Dependencia conceptual de Schank

➔ Tiene como objetivo representar conceptos mediante la construcción de oraciones haciendo uso de categorías conceptuales:

- PP ( objetos del mundo real ).
- ACT ( acciones del mundo real ).

- PA ( atributos de objetos) • AA ( atributos de acciones ).
- T ( tiempos ).
- LOC ( localizaciones/posiciones ).

➔ Se representan las relaciones a partir de las categorías:

- p ( pasado)
- ningún modificador (presente).
- f (futuro) • c ( condicional)
- / (negación).
- ? (pregunta)
- ts = x ( relación que comienza en el instante x ).
- tf = x ( relación termina en el instante x ).

➔ Las principales acciones primitivas que considera son:

- ATRANS: Transferencia de una relación como posición, propiedad o control.
- PTRANS: Cambio de la posición física del objeto.
- MTRANS: Transmitir una idea, decir, contar o comunicar.
- PROPEL: Aplicar fuerza física a un objeto.
- MOVE: Mover una parte del cuerpo.
- GRASP: Acto para coger un objeto.
- INGEST: Ingerir algo.
- SPEAK: Producir sonidos con la boca.
- EXPEL: Expulsión desde el cuerpo del objeto al exterior.
- DO: Hacer, posibilitar, favorecer o impedir.
- ATTEND: Atender.
- SPEAK: Producir un sonido
- MBUILD: Deducir o inferir una idea mental.

➔ **Representación gráfica con Sowa**

➔ Permiten expresar relaciones globales entre frases, existiendo nodos que representan a dichas frases, proposiciones párrafos, ... ➔ Máquina de inferencia

- **Especialización:** Obtener una propuesta particular de una propuesta general por especialización de algunos elementos.

- **Generalización:** Contrario a especialización.

- **Unión y simplificación:** Similar al principio de resolución en lógica.

➔ Permite expresarse en forma lineal, de tal forma que los nodos aparecen entre corchetes y las etiquetas de los arcos entre paréntesis.

- AGT: Agente de Entidad que realiza la acción.

- ATR: Atributo de una entidad.

- OBJ: Entidad afectada por la acción.

- RCP: Receptor de la acción.

- ORIGEN: Origen de la acción.

- DEST: Destino de la acción

- TEMP: Tiempo de la acción.

- PAST: Pasado

- FUTURE: Futuro

- NEC: Negación de la acción ( no se realiza ).

- LOC: Lugar de la acción.

- REASON: Motivo de la acción.

- INST: Instrumento con el cual se realiza la acción.

- POS: Posesión de una entidad.

➔ **Redes bayesianas**

➔ Consiste en fijar en la red el valor de las variables conocidas y calcular la probabilidad de las variables cuyo valor se desconoce.

➔ Para ello utilizamos la probabilidad condicionada, permitiendo obtener la probabilidad para cada nodo.

➔ Dada una red, debemos determinar si 2 nodos son independientes, de tal forma que para solventarlo hemos de fijarnos en nodo sumidero y nodo no sumidero.

➔ Dos nodos son independientes si en el camino que une a dichos nodos encontramos:

1. Nodo sumidero : Si no conocemos su valor ni el de sus hijos, podemos afirmar que son independientes.

2. Nodo no sumidero: Si conocemos su valor, son independientes.

➔ Para el cálculo de probabilidades tenemos las siguientes fórmulas:

- $P(X|Y) = P(X,Y) / P(Y)$
- $P(X,Y)$  son independientes:  $P(X,Y) = P(X) * P(Y)$
- $P(X,Y)$  no son independientes:  $P(X,Y) = P(X/Y) * P(Y)$ .

## TEMA 7. REPRESENTACIÓN DEL CONOCIMIENTO CON MARCOS Y GUIONES

➔ Con los marcos se pretendía introducir un método de representación del conocimiento y de razonamiento que supere las limitaciones de la lógica a la hora de abordar problemas como la visión artificial, comprensión del lenguaje,

...

➔ Mediante los guiones se quiso extender los grafos de dependencia conceptual, debido a que como se menciona anteriormente resultan insuficientes.

### ➔ **Marcos**

- Son estructuras de datos que incluyen conocimiento sobre un objeto en particular.
- Podemos definirlos como principios de POO aplicada en el campo IA.

#### ➤ Puntos de vista

- Una misma entidad puede definirse con tantos marcos como puntos de vista se tenga de ellos.
- Cada uno de los puntos de vista se corresponde con un marco, y cada marco tendrá diferentes campos según se necesite.

#### ➤ Redes taxonómicas con marcos

- Establecen jerarquías entre conceptos.
- Permiten la herencia de propiedades.
- Se utiliza un enlace ISA
- Relaciona clases con superclases.
- Relaciona instancias con clases.
- La relación es transitiva.



- En algunos casos se impiden los ciclos.

Un concepto puede heredar propiedades de sus antepasados o antecesores.

➤ Tipos de herencia :

1. Estricta: Todas las propiedades de un nodo superior son heredadas por los nodos inferiores ( lógica clásica).
2. Por defecto: Todas las propiedades son heredadas, a no ser que se indique lo contrario.

➤ Se definen propiedades de un campo:

- Valor por defecto: Se asigna el marco y se hereda.
- Multivaluada: Univaluado/multivaluado
- Restricciones: Limitan el conjunto de valores de un campo o la ocupación a la que pertenece ( instancias de profesión).
- Certeza: Credibilidad del valor asignado.
- Facetas de interfaz: Mensajes destinados a la interacción con el usuario.

➔ **Guiones**

- Un guion es una estructura de conocimiento que contiene una secuencia estereotipada de acciones.
- Puede considerarse como un tipo especial de marco:
  - Cada campo es un suceso/acción.
  - Los sucesos forman una secuencia.
- Como acción se consideran la conversación, el pensamiento, actos de voluntad y sentimientos.
- Las escenas del guion sirven para reconocer situaciones.
- Un guion se compone de los siguientes elementos:
  1. **Escenas:** Sucesos en el guion, enlazados casualmente.
  2. **Roles y objetos:** Personas y cosas que intervienen, incluyendo restricciones para indicar que personas/objetos pueden asignarse a las variables.
  3. **Cabeceras:** Nombre del guion, condiciones, instrumentos y lugares. Su misión es activar el guion en el momento oportuno.
  4. **Resultados:** Hechos ciertos al completar el guion.

➤ Cada escena se representa con un grafo de dependencia conceptual:

- **Ventajas:**

- Al igual que los grafos de dependencia conceptual GDC, representan el conocimiento sin depender de ningún idioma en particular.

- Con los guiones se permite integrar con el fin de formar una historia coherente.

- **Inconvenientes**

- Rigidez del mecanismo de representación.

- Algunas herramientas permiten definir líneas alternativas.

- Incapacidad de compartir información entre guiones.

- Incapacidad de expresar motivaciones e intenciones.

- Los marcos son estructurados, pero la lógica no.

- Los marcos poseen nuevas posibilidades de razonamiento:

- ➔ Razonamiento por defecto.

- ➔ Uso de la información incompleta o imprecisa.

- La lógica realiza más inferencias de las deseadas porque es incapaz de conocer cuáles son relevantes y cuáles no ( explosión combinatoria).

- En la vida real se trabaja con información incompleta e imprecisa.

## TEMA 8. INTRODUCCIÓN AL APRENDIZAJE AUTOMÁTICO

### ➔ Árboles de decisión

➤ Se utiliza una representación en forma de árbol:

- Los nodos que no son hojas representan un atributo o una pregunta.

- Los nodos hojas representan respuestas o clases.

➤ Para construir el árbol lo haremos a partir de un conjunto de datos.

### ➤ Clasificación

- Dado un patrón X, para obtener su clase independientemente del conjunto del aprendizaje.
- Para cada nodo sin terminar, responder la pregunta de acuerdo a los atributos de X.

### ➤ Construcción automática: ID3

- Se basa en reducir su entropía.
- Obtiene buenos árboles, pero no son óptimos.
- Es un algoritmo codicioso.
- Requiere que la gran cantidad de patrones sea muy superior a la cantidad de clases.
- Aplicaciones: Atributos discretos y un conjunto finito de clases.

### ➤ ID3: Ejecución

- Comienza a partir de un conjunto de patrones.
- Si bien hay conjuntos con patrones de diferentes clases:
  - ➔ Para cada atributo, aún elegido, clasifique los patrones del conjunto.
  - ➔ Calcule el beneficio de clasificar por medio de cada atributo, basado en la entropía.
  - ➔ Se elije el atributo que produce el mayor beneficio.
  - ➔ Se repite el procedimiento para cada subconjunto.

### ➔ **Entropía**

- Magnitud que mide la información en un flujo de datos, es decir, cuánta información nueva se proporciona.
- Es una medida de aleatoriedad de elementos en un sistema.

### ➔ **Fórmulas**

- Probabilidad de que un atributo K se establezca en un valor V en el subconjunto S:

$$P(K = v, S) = |S_k = v| / |S|$$

- Entropía del conjunto S:

$$E(P(res = v_1, S), \dots, P(res = v_n, S)) = \sum_{i=1}^n -P(res = v_i, S) * \log_2(P(res = v_i, S))$$

- Beneficio de clasificar mediante el atributo K:

$$Profit(S, K) = E(s) - \sum_{value \in k} -P(k = value, S) * E(Sv)$$

## ➔ El perceptrón y las redes neuronales

- Están inspirados en las conexiones neuronales de nuestro cerebro, en particular en las neuronas multipolares.

$$\sum_{value \in k} w_i * x_i + b \quad |f(z) \quad f(z) \text{ es la función de activación la más simple es la función del peso.}$$

- Utilidad: Sólo puede resolver funciones de hiperplano ( realizan un corte simple ).

## ➔ Neurona artificial y su funcionamiento

- Una neurona artificial es un elemento computacional que recibe unas entradas.
- Calcula la suma ponderada de las entradas y se activa si dicha suma no supera un umbral.
- El entrenamiento de una neurona artificial consiste en ajustar los pesos de la suma ponderada para acertar sobre un conjunto de datos de entrenamiento.
- Divide el espacio de entrada en 2 por un plano. A un lado del plano se activa y al otro lado del plano no se activa.

## ➔ Entrenamiento: Regla Delta

- Los datos se representan al perceptrón y los pesos se actualizan si la salida no es correcta:

$\alpha$  es la tasa de aprendizaje, y debería ser pequeña.

$$w_i = w_i + \alpha * (Y_{esperada} - Y_{obtenida}) X_i$$

$$b = b + \alpha * (Y_{esperada} - Y_{obtenida})$$

### ➤ Comentarios

- Las redes de avance se entrenan mediante un algoritmo derivado de la regla de propagación hacia atrás.
- Las redes concurrentes se entrenan mediante algoritmos evolutivos:

■ Existen 3 capas en las redes neuronales:

1. Capa entrada.
2. Capa oculta.
3. Capa salida.

➤ Aprendizaje sin supervisión o no supervisado

- Se utiliza para clasificar un conjunto de datos sin etiquetas.
- Las técnicas de agrupamiento identifican grupos de datos.
- Cada grupo debe ser homogéneo y diferente a los otros grupos.

■ Distancia: La distancia entre elementos del mismo grupo debe ser pequeña y grande entre elementos de diferentes grupos.

- Los algoritmos de agrupamiento requieren parámetros que afecten a los resueltos.
- Se produce cuando el conjunto de entrenamiento no tiene características objetivo.
- Se basa en agrupar elementos parecidos entre sí y diferenciarlos de los que son más diferentes.

Por lo general, cada grupo se caracteriza por su media y variante.

➔ **Kmeans**

1. Inicializar al atar los centros de los grupos K ( parámetro ).
  - a) Se asignan y actualizan los centros.
  - b) Cada patrón se asigna a su centro más cercano.
  - c) Se calculan los nuevos centros de acuerdo con la media de los patrones asignados.
2. Se repite el procedimiento hasta la convergencia. Al actualizar los centros, algunos patrones cambian a otro centro más cercano.