

# **RESUMEN TEORÍA INTELIGENCIA ARTIFICIAL:**

## **ÍNDICE DE CONTENIDOS:**

### **BLOQUE 1: FUNDAMENTOS DE LA INTELIGENCIA ARTIFICIAL**

#### **TEMA 1: INTRODUCCIÓN A LA IA**

- 1.1 ¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?**
- 1.2 ¿POR QUÉ NO IMITAR UN CEREBRO HUMANO?**
- 1.3 HISTORIA DE LA INTELIGENCIA ARTIFICIAL**

#### **TEMA 2: AGENTES INTELIGENTES Y ENTORNOS**

- 2.1 AGENTES Y ENTORNOS**
- 2.2 FUNCIÓN Y PROGRAMA DEL AGENTE**
- 2.3 RACIONALIDAD, OMNISCENCIA, APRENDIZAJE Y AUTONOMÍA**
- 2.4 REAS**
- 2.5 TIPOS DE ENTORNOS**
- 2.6 TIPOS DE AGENTES**

### **BLOQUE 2: RESOLUCIÓN DE PROBLEMAS POR BÚSQUEDA**

#### **TEMA 3: BÚSQUEDA NO INFORMADA**

- 3.1 AGENTES REACTIVOS Y PLANIFICADORES**
- 3.2 PROBLEMAS DE BÚSQUEDA Y ESPACIO DE ESTADOS**
- 3.3 GRAFO DEL ESPACIO DE ESTADOS**
- 3.4 ÁRBOLES Y ESTRATEGIAS DE BÚSQUEDA**
- 3.5 BÚSQUEDA PRIMERO EN PROFUNDIDAD (BPP/DFS)**
- 3.6 BÚSQUEDA PRIMERO EN ANCHURA (BPA/BFS)**
- 3.7 PROFUNDIDAD ITERATIVA**
- 3.8 BÚSQUEDA DE COSTE UNIFORME (BCU/UCS)**
- 3.9 RESUMEN DE PROPIEDADES**

#### **TEMA 4: BÚSQUEDA INFORMADA**

- 4.1 HEURÍSTICA DE BÚSQUEDA**
- 4.2 BÚSQUEDA VORAZ PRIMERO EL MEJOR (GREEDY)**
- 4.3 BÚSQUEDA A\***
- 4.4 OPTIMALIDAD DE A\***
- 4.5 OPTIMALIDAD DE A\* ( $f(n) \leq f(A)$ )**
- 4.6 OPTIMALIDAD DE A\* ( $f(A) \leq f(B)$ )**
- 4.7 CREACIÓN DE HEURÍSTICAS ADMISIBLES**
- 4.8 CONSISTENCIA DE LAS HEURÍSTICAS**
- 4.9 VARIANTES DE A\***

## **BLOQUE 3: LÓGICA**

### **TEMA 5: LÓGICA PROPOSICIONAL**

- 5.1 CONOCIMIENTO**
- 5.2 LÓGICA E IMPLICACIÓN**
- 5.3 INFERENCIA DE PRUEBAS**
- 5.4 LÓGICA PROPOSICIONAL**
- 5.5 TABLAS DE VERDAD**
- 5.6 TAREAS DE RAZONAMIENTO**
- 5.7 SATISFACIBILIDAD E IMPLICACIÓN**
- 5.8 FORMA NORMAL CONJUNTIVA (CNF)**
- 5.9 SOLUCIONADORES SAT EFICIENTES**
- 5.10 PLANIFICACIÓN COMO SATISFACIBILIDAD**
- 5.11 ENCADENAMIENTO HACIA DELANTE / ATRÁS Y RESOLUCIÓN**

### **TEMA 6: LÓGICA DE PRIMER ORDEN**

- 6.1 ESPECTROS DE REPRESENTACIONES**
- 6.2 PODER EXPRESIVO Y LENGUAJE NATURAL**
- 6.3 LÓGICA PROPOSICIONAL VS DE PRIMER ORDEN**
- 6.4 MUNDO POSIBLE**
- 6.5 SINTAXIS Y SEMÁNTICA**
- 6.6 INFERENCIA EN LÓGICA DE PRIMER ORDEN**

## **BLOQUE 4: INCERTIDUMBRE**

### **TEMA 7: PROBABILIDAD**

- 7.1 INCERTIDUMBRE**
- 7.2 LEYES BÁSICAS DE PROBABILIDAD**
- 7.3 VARIABLES ALEATORIAS Y DISTRIBUCIONES**
- 7.4 PROBABILIDAD CONDICIONAL Y NORMALIZACIÓN**
- 7.5 REGLAS DE PROBABILIDAD**
- 7.6 INFERENCIA PROBABILÍSTICA**
- 7.7 INFERENCIA POR ENUMERACIÓN**
- 7.8 REGLA DE BAYES E INDEPENDENCIA**

### **TEMA 8: REDES BAYESIANAS**

- 8.1 INTRODUCCIÓN**
- 8.2 SINTAXIS Y SEMÁNTICA**
- 8.3 INDEPENDENCIA CONDICIONAL EN REDES BAYESIANAS**
- 8.4 ESTRUCTURA DE LAS REDES BAYESIANAS**
- 8.5 INFERENCIA POR ENUMERACIÓN EN REDES BAYESIANAS**
- 8.6 ELIMINACIÓN DE VARIABLES**
- 8.7 FACTORES**
- 8.8 IMPORTANCIA DEL ORDEN**
- 8.9 COMPLEJIDAD COMPUTACIONAL Y ESPACIAL**
- 8.10 POLIÁRBOLES**

### **TEMA 9: RAZONAMIENTO PROBABILÍSTICO EN EL TIEMPO**

- 9.1 INTRODUCCIÓN**
- 9.2 TIEMPO E INCERTIDUMBRE**
- 9.3 INFERENCIA EN MODELOS TEMPORALES**
- 9.4 MODELOS OCULTOS DE MARKOV**
- 9.5 FILTROS DE KALMAN**
- 9.6 FILTROS DE PARTÍCULAS**

## **▣ BLOQUE 1: FUNDAMENTOS DE LA INTELIGENCIA ARTIFICIAL**

### **▣ TEMA 1: INTRODUCCIÓN A LA IA**

#### **▣ 1.1 ¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?**

**Inteligencia Artificial:** Ciencia basada en hacer máquinas que piensen y actúen como personas, de una forma racional (tomar decisiones para alcanzar los objetivos correctos en función del conocimiento que se posee).

**Test de Turing (1950):** Comprueba si un humano es capaz de diferenciar si las respuestas a unas preguntas propuestas proceden de una persona o han sido generadas por ordenador (procesamiento de lenguaje natural, representación del conocimiento y razonamiento automático).

#### **▣ 1.2 ¿POR QUÉ NO IMITAR UN CEREBRO HUMANO?**

**Cerebro humano vs artificial:** Son buenos para tomar decisiones racionales pero no perfectos, no son tan modulares como el software, la memoria y la simulación son claves para tomar decisiones.

**Problemas del pensamiento racional:** Dificultad para expresar conocimiento informal en términos formales, requerimiento de muchos recursos computacionales para resolver problemas reales.

#### **▣ 1.3 HISTORIA DE LA INTELIGENCIA ARTIFICIAL**

**Años 40-50:** 1943 (modelo de circuito booleano del cerebro), 1950 (CMI de Turing).

**Años 50-70:** 1950 (primeros programas de IA), 1956 (adopción del término IA), 1965 (Algoritmo de Robinson).

**Años 70-90:** 1969-1979 (sistemas basados en el conocimiento), 1979-1988 (auge de las industrias de sistemas expertos), 1988-1993 (caída de las industrias de sistemas expertos).

**Años 90:** Resurgimiento de la probabilidad, atención a la incertidumbre, aumento de la profundidad técnica, agentes y sistemas de aprendizaje, 1996 (Kasparov > Deep Blue), 1997 (Deep Blue > Kasparov).

**Años 2000:** Big Data, Big Compute, Redes Neuronales, 2011 (Watson > Ken Jennings y Brad Rutter), 2016 (AlphaGo > Lee Sedol), 2022 (50000 millones en inversión).

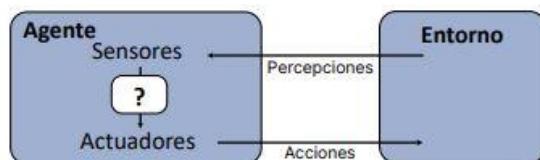
## TEMA 2: AGENTES INTELIGENTES Y ENTORNOS

### 2.1 AGENTES Y ENTORNOS

**Agente racional:** Aquel que actúa de forma que se logre el mejor resultado posible, o el esperado en caso de que haya incertidumbre. Elige aquella acción que maximiza el valor esperado de la medida de rendimiento dada la secuencia de percepciones hasta la fecha y el conocimiento previo del entorno.

**Características:** No son omniscientes ni clarividentes ni cometen errores, pero sí son autónomos y pueden explorar y aprender.

**Agentes de la IA:** Poseen grandes recursos computacionales y entornos que requieren una toma de decisiones no trivial. Perciben su entorno mediante sensores y actúa sobre él a través de actuadores.



### 2.2 FUNCIÓN Y PROGRAMA DEL AGENTE

**Función del agente ( $f: P^* \rightarrow A$ ):** Mapea de historiales de percepción a acciones, es decir, el historial completo de respuestas reales del agente ante una secuencia de percepciones.

**Programa del agente ( $f = \text{Agente}(\text{agente}, \text{máquina})$ ):** La función  $f$  depende tanto de la máquina  $M$  como del agente  $I$ , ya que las máquinas reales tienen velocidad y memoria limitadas, lo que introduce retardos.

### 2.3 RACIONALIDAD, OMNISCENCIA, APRENDIZAJE Y AUTONOMÍA

**Medida de rendimiento:** Evalúa las secuencias del entorno.

**Agente omnisciente:** Conoce el resultado de su acción y actúa de acuerdo con él. Inicialmente esto no es posible, por lo que debería explorar un medio inicialmente desconocido y aprender de lo que está percibiendo.

**Agente carente de autonomía:** Se apoya más en el conocimiento inicial del diseñador que en sus propias percepciones.

**Agente autónomo:** Aprende a determinar cómo tiene que compensar el conocimiento incompleto o parcial inicial.

### 2.4 REAS

**REAS (Medida de Rendimiento, Entorno, Actuadores, Sensores):** Define el entorno de trabajo.

**Pacman:** Parcialmente observable, multiagente, determinista, dinámico, discreto y de física conocida.

**Parchís:** Totalmente observable, multiagente, estocástico, estático, discreto y de física conocida.

**Diagnóstico:** Parcialmente observable, mono agente, estocástico, dinámico, continuo y de física conocida.

**Taxi:** Parcialmente observable, multiagente, estocástico, dinámico, continuo y de física conocida.

### 2.5 TIPOS DE ENTORNOS

**Totalmente observable:** Los sensores proporcionan acceso al estado completo del entorno en cada momento.

**Parcialmente observable:** El agente necesita memoria.

**Determinista:** El siguiente estado está totalmente determinado por el estado actual y la acción ejecutada.

**Estocástico:** El siguiente estado no está determinado por el estado actual y la acción ejecutada, por lo que puede tener que prepararse para contingencias.

**Multiagente:** El agente puede tener que comportarse de forma aleatoria evitando la previsibilidad.

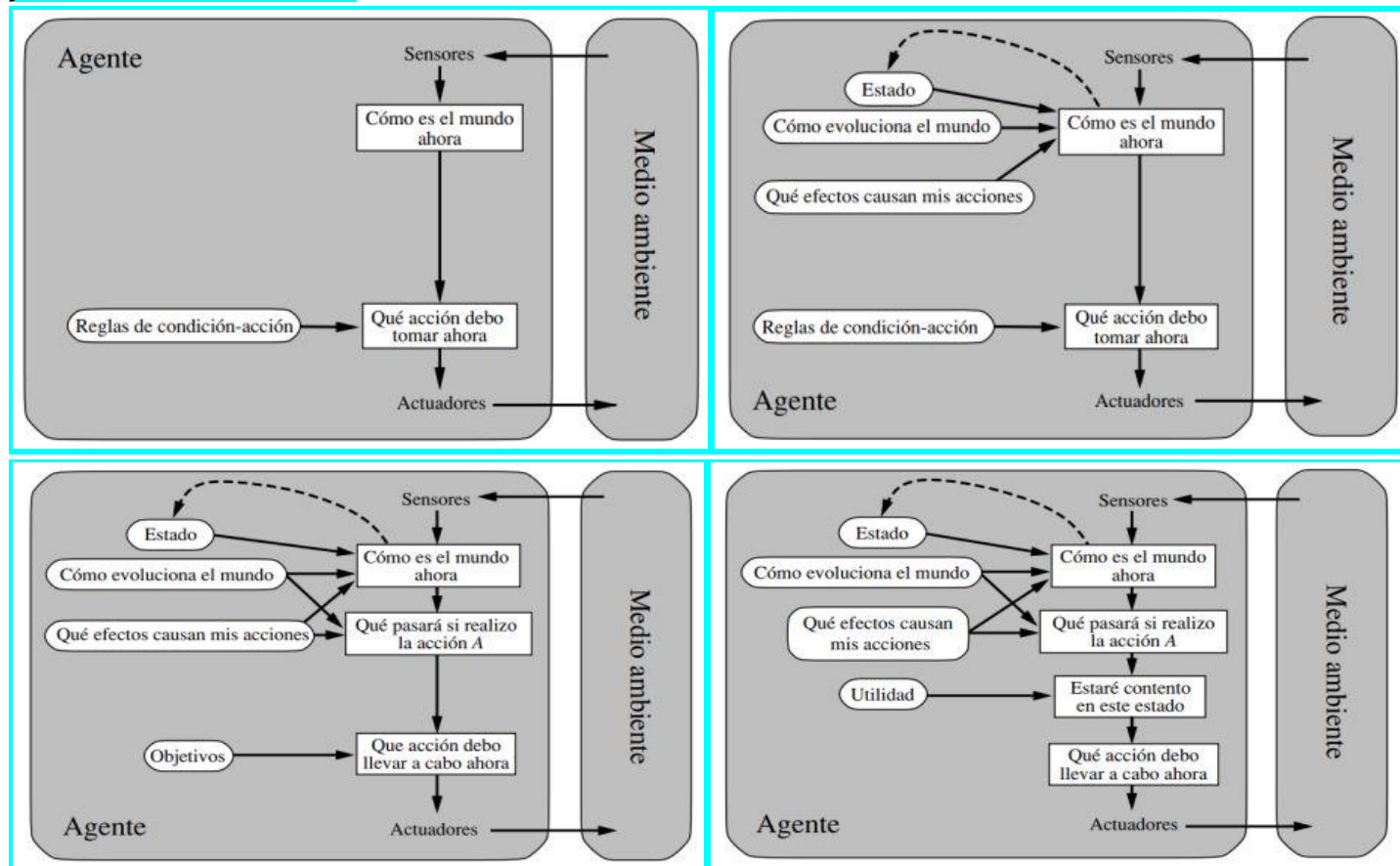
**Dinámico:** El entorno puede cambiar cuando el agente está deliberando.

**Estático:** El entorno no puede cambiar cuando el agente está deliberando, por lo que el agente tiene tiempo para tomar una decisión racional.

**Discreto/Continuo:** Se distingue según el estado del entorno, manejo del tiempo, y percepciones y acciones del agente.

**Física desconocida:** Necesidad de exploración.

## 2.6 TIPOS DE AGENTES



**Agentes reactivos simples:** Funcionan si se puede tomar la decisión correcta sobre la base de la percepción actual, y si y sólo si el entorno es totalmente observable (eliminar sensor de posición de un aspirador).

**Agentes reactivos basados en modelos:** Mantienen un estado interno que depende de la historia percibida y refleja alguno de los aspectos no observables del estado actual (coche autónomo al llegar a un cruce).

**Agentes basados en objetivos:** El programa se combina con información de los resultados de las acciones posibles para tomar las decisiones que permitan alcanzar el objetivo (taxista al llegar a un cruce).

**Agentes basados en la utilidad:** Proyectan un estado o secuencia de estados en un número real que representa un nivel de felicidad.

## BLOQUE 2: RESOLUCIÓN DE PROBLEMAS POR BÚSQUEDA

### TEMA 3: BÚSQUEDA NO INFORMADA

#### 3.1 AGENTES REACTIVOS Y PLANIFICADORES

**Agentes reactivos simples y basados en modelos:** Eligen la acción basándose en la percepción actual y puede que en alguna de las anteriores, pueden tener memoria o un modelo del estado actual del mundo, no consideran las consecuencias futuras de sus acciones y consideran cómo es el mundo ahora.

**Agentes planificadores objetivos y de utilidad:** Se hacen suposiciones, sus decisiones se basan en las hipotéticas consecuencias de sus acciones, deben tener un modelo de cómo evoluciona el mundo en respuesta a sus acciones, deben formular un objetivo y consideran cómo sería el mundo.

#### 3.2 PROBLEMAS DE BÚSQUEDA Y ESPACIO DE ESTADOS

**Problemas de búsqueda:** Está formado por un espacio de estados (estados alcanzables), una función sucesora con acciones y costes, un estado inicial y un test objetivo.

**Solución:** Secuencia de acciones que transforma el estado inicial en un estado objetivo.

**Estado del mundo:** Incluye hasta el último detalle del entorno.

**Estado de búsqueda:** Mantiene solo los detalles necesarios para planificar (abstracción).

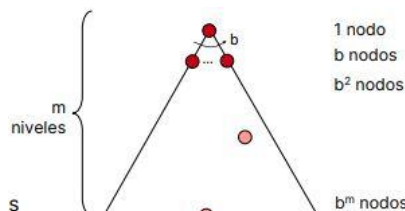
#### 3.3 GRAFO DEL ESPACIO DE ESTADOS

**Grafo del espacio de estados:** Representación matemática de un problema de búsqueda compuesto por **nod**s (abstracciones de las configuraciones del mundo), **arcos** (resultados de las acciones) y **tests / nodos objetivo**.

#### 3.4 ÁRBOLES Y ESTRATEGIAS DE BÚSQUEDA

**Árbol de búsqueda:** Se suponen todos los planes y resultados, el estado inicial es el nodo raíz, los hijos corresponden a sucesores, los nodos muestran estados correspondientes a planes que alcanzan esos estados, y para la mayoría de problemas es inviable construir el árbol completo.

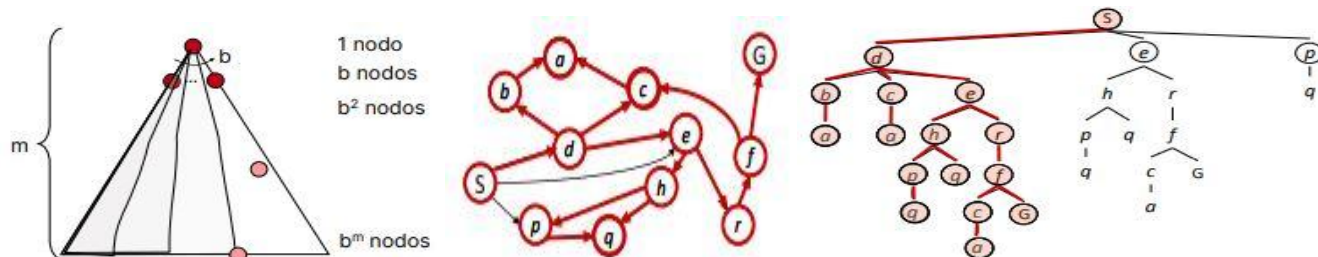
**Objetivos:** Expandir los planes potenciales (nodos del árbol), mantener una frontera de planes parciales en estudio, y tratar de expandir tan pocos nodos como sea posible.



**Propiedades de las estrategias de búsqueda:** **Completo** (solución existente), **óptimo** (camino de menor coste), **complejidad temporal o espacial**.

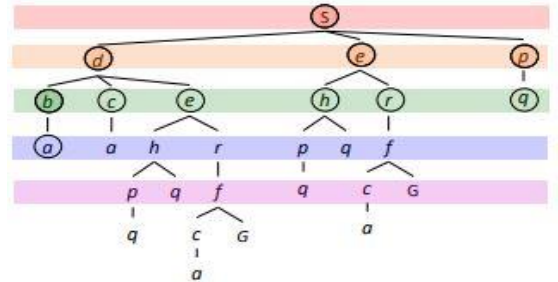
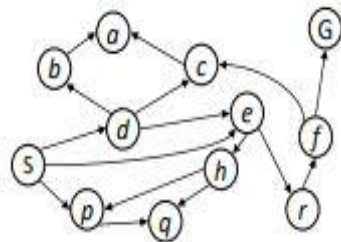
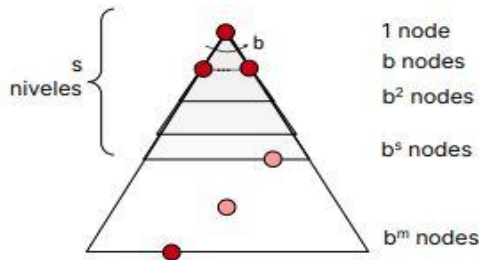
**Dibujo del árbol:** **b** (factor de ramificación, máximo número de sucesores de un nodo), **m** (máxima profundidad, longitud máxima de un camino), **número de nodos** ( $1 + b^1 + b^2 + \dots + b^m = O(b^m)$ ).

#### 3.5 BÚSQUEDA PRIMERO EN PROFUNDIDAD (BPP/DFS)



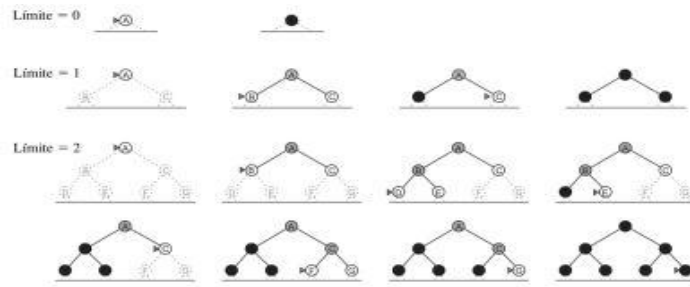
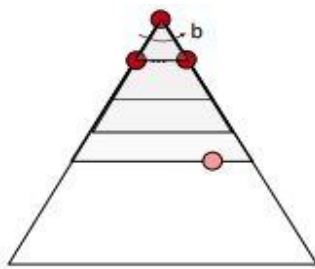
**Búsqueda primero en profundidad (BPP/DFS):** **Expansión** (el subárbol de la izquierda, es posible que se tenga que procesar todo el árbol, si m es finito tarda  $O(b^m)$ ), **espacio de frontera** (al solo tener hermanos en el camino al raíz,  $O(b \cdot m)$ ), es **completo** si m es finito y evitamos los ciclos, y no es **óptimo** ya que no llega a la solución más a la izquierda, independientemente de la profundidad o el coste.

### 3.6 BÚSQUEDA PRIMERO EN ANCHURA (BPA/BFS)



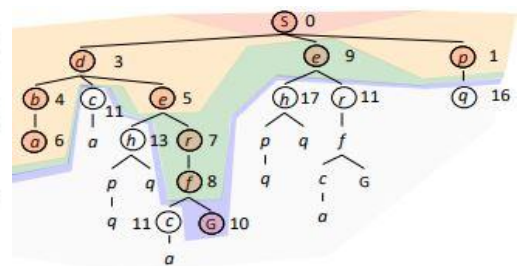
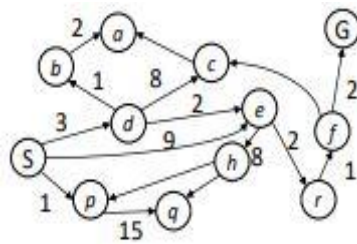
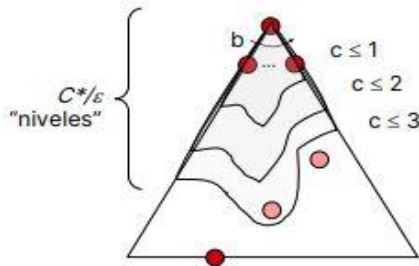
**Búsqueda primero en anchura (BPA/BFS):** **Expansión** (nodos por encima de la solución superficial, si  $s$  es la profundidad de la solución la búsqueda tarda  $O(b^s)$ ), espacio de frontera (sólo contiene el último nivel, por lo que  $O(b^s)$ ), es **completo** si  $s$  es finita, y es **óptimo** sólo si todos los costes son 1.

### 3.7 PROFUNDIDAD ITERATIVA



**Profundidad iterativa:** Busca obtener la ventaja de espacio de BPP con las ventajas tiempo de BPA. Suele utilizarse cuando hay un espacio grande de búsqueda y no se conoce la profundidad de la solución.

### 3.8 BÚSQUEDA DE COSTE UNIFORME (BCU/UCS)



**Búsqueda de coste uniforme (BCU/UCS):** **Expansión** (nodos con menor coste que la solución más barata, la profundidad efectiva es coste de solución / coste de acción =  $C^*/\epsilon$ , por lo que tarda  $O(b^{C^*/\epsilon})$ ), **espacio de frontera** (sólo contiene el último nivel, por lo que  $O(b^{C^*/\epsilon})$ ), es **completo** si el coste es finito, y es **óptimo**.

### 3.9 RESUMEN DE PROPIEDADES

Criterio	Primero en profundidad	Primero en anchura	Profundidad iterativa	Coste uniforme
¿Completa?	No	Sí	Sí	Sí
Tiempo	$O(b^m)$	$O(b^{d+1})$	$O(b^m)$	$O(b^d)$
Espacio	$O(b^m)$	$O(b^{d+1})$	$O(b^m)$	$O(b^d)$
¿Óptima?	No	Sí	Sí	Sí



## TEMA 4: BÚSQUEDA INFORMADA

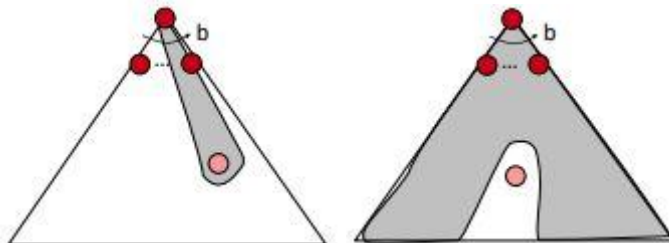
### 4.1 HEURÍSTICA DE BÚSQUEDA

**Heurística:** Función que estima lo cerca que está un cierto estado del objetivo siguiendo el camino de menor coste.

**Heurística admisible (optimista):** Ralentiza los planes malos, pero nunca sobreestima los costes reales. Una heurística es admisible si  $0 \leq h(n) \leq h^*(n)$ , donde  $h^*(n)$  es el coste verdadero del camino desde  $n$  hasta un objetivo.

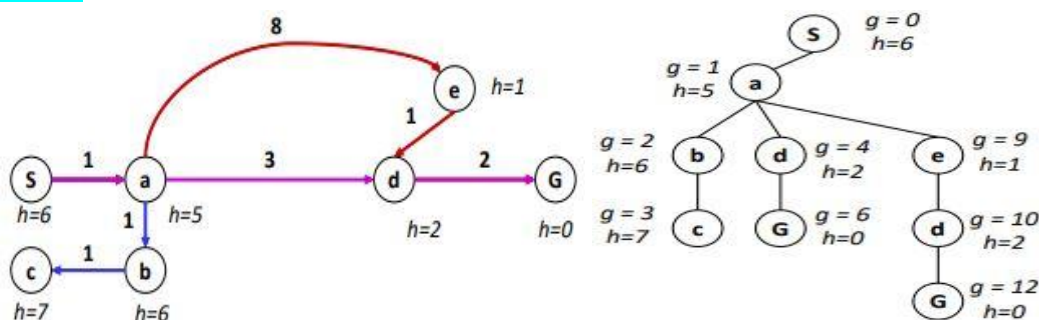
**Heurística inadmisble (pesimista):** Rompe la optimalidad al dejar atrapados los buenos planes en la frontera.

### 4.2 BÚSQUEDA VORAZ PRIMERO EL MEJOR (GREEDY)



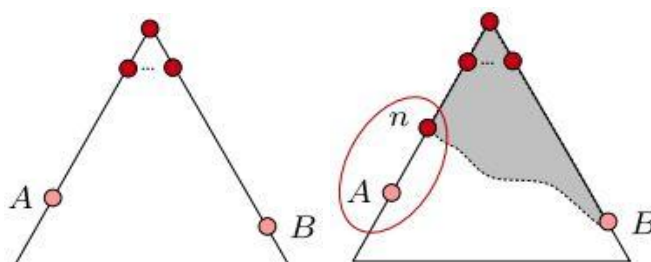
**Búsqueda voraz primero el mejor (greedy):** Estrategia (expandir el nodo que parece que está más cerca del objetivo), complejidad temporal y espacial ( $O(b^m)$  en el peor caso), heurística (estimación de la distancia al objetivo más cercano para cada estado), no es óptima ni completa.

### 4.3 BÚSQUEDA A\*



**A\* (BCU + voraz):** La búsqueda por coste uniforme ordena por coste del camino o coste hacia atrás ( $g(n)$ ), mientras que la búsqueda voraz ordena por proximidad al objetivo o coste hacia delante ( $h(n)$ ). Por lo tanto, A\* ordena por el sumatorio de ambas ( $f(n) = g(n) + h(n)$ ).

### 4.4 OPTIMALIDAD DE A\*



**Optimalidad de la búsqueda en un árbol A\*:** Asumiendo que A es un nodo objetivo óptimo, B es un nodo objetivo subóptimo y  $h$  es admisible, se puede demostrar que A saldrá de la frontera antes que B.

**Prueba de optimalidad:** Se imagina que B está en la frontera y que algún antecesor  $n$  de A también está en la frontera, por lo que  $n$  será expandido antes que B ( $f(n) \leq f(A) \leq f(B)$ ) y la búsqueda en A\* es óptima.



#### 4.5 OPTIMALIDAD DE A\* ( $f(n) \leq f(A)$ )

$f(n) = g(n) + h(n) = (\text{coste camino a } n) + (\text{est. coste de } n \text{ a } A)$

$f(A) = g(A) + h(A) = (\text{coste camino a } A) + (\text{est. coste de } A \text{ a } A)$

**Heurística:**  $h(A) = (\text{est. coste de } A \text{ a } A) = 0$

$f(n) = g(n) + h(n) = (\text{coste camino a } n) + (\text{est. coste de } n \text{ a } A)$

$f(A) = g(A) = (\text{coste camino a } A)$

$(\text{coste camino a } n) + (\text{est. coste de } n \text{ a } A) \leq (\text{coste camino a } A)$

$g(n) + h(n) \leq g(A)$

$f(n) \leq f(A)$

#### 4.6 OPTIMALIDAD DE A\* ( $f(A) \leq f(B)$ )

$f(A) = g(A) + h(A) = (\text{coste camino a } A) + (\text{est. coste de } A \text{ a } A)$

$f(B) = g(B) + h(B) = (\text{coste camino a } B) + (\text{est. coste de } B \text{ a } B)$

**Heurística:**  $h(A) = h(B) = 0$

$f(A) = g(A) = (\text{coste camino a } A)$

$f(B) = g(B) = (\text{coste camino a } B)$

$(\text{coste camino a } A) < (\text{coste camino a } B)$

$g(A) < g(B)$

$f(A) < f(B)$

#### 4.7 CREACIÓN DE HEURÍSTICAS ADMISIBLES

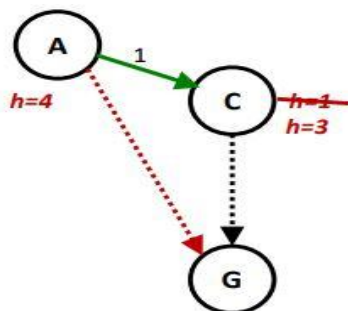
**Heurísticas admisibles (teorema 1):** El problema de P2 es una versión relajada de P1 si  $A2(s) \supseteq A1(s)$  para cada  $s$ , donde  $A$  son las acciones disponibles y  $s$  los estados.

**Heurísticas admisibles (teorema 2):**  $h2^*(s) \leq h1^*(s)$  para cada  $s$ , por lo que  $h2^*(s)$  es una heurística admisible para P1.

**Dominancia de heurísticas:**  $h1 \geq h2$  si  $\forall n \ h1(n) \geq h2(n)$  siempre que ambas heurísticas sean admisibles.

**No dominancia de heurísticas:** Se crea una nueva heurística calculando el máximo de ambas ( $h(n) = \max(h1(n), h2(n))$ ).

#### 4.8 CONSISTENCIA DE LAS HEURÍSTICAS



**Admisibilidad:** Coste  $\leq$  coste real al objetivo ( $h(A) \leq h^*(A)$ ).

**Consistencia:** Coste heurístico del "arco"  $\leq$  coste real del arco ( $h(A) - h(C) \leq c(A,C)$  o  $h(A) \leq c(A,C) + h(C)$ ).

**Consecuencias:** El valor  $f$  nunca decrece a lo largo de un camino ( $h(A) \leq c(A,C) + h(C) \Rightarrow g(A) + h(A) \leq g(A) + c(A,C) + h(C)$ ).

**Búsqueda en árbol:** A\* es óptimo si la heurística es admisible, el BCU es un caso especial ( $h = 0$ ).

**Búsqueda en grafos:** A\* es óptima si la heurística es consistente, el BCU es óptimo ( $h = 0$ ).

#### 4.9 VARIANTES DE A\*

**IDA\* (Iterative Deepening A\*):** Funciona como profundidad iterativa, pero usa un límite  $f$  en vez de un límite de profundidad.

**RBFS (Recursive best-first search):** Usa un límite  $f$ , donde  $f$  es el valor del mejor camino alternativo desde cualquier antecesor del nodo actual.

**SMA\*:** Usa toda la memoria disponible para la cola descartando los peores nodos cuando se llena, pero recuerda su valor en el padre.

## BLOQUE 3: LÓGICA

### TEMA 5: LÓGICA PROPOSICIONAL

#### 5.1 CONOCIMIENTO

**Base de conocimiento (KB):** Conjunto de sentencias que representan afirmaciones sobre el mundo en un lenguaje formal.

**Enfoque declarativo para crear un agente:** Decirle lo que necesita saber o pedirle que aprenda el conocimiento, para después preguntarse qué hacer obteniendo las respuestas de la base de conocimiento.

**Algoritmo de inferencia:** Base de conocimiento (hechos específicos del dominio) y motor de inferencia (código genérico).

#### 5.2 LÓGICA E IMPLICACIÓN

**Lógica:** Indica las sentencias que están permitidas, cuáles son los mundos posibles, y qué frases son verdad en cada mundo. Se dice que el modelo  $m$  'satisface  $\alpha$ ' o 'es un modelo de  $\alpha$ ', donde  $M(\alpha)$  representa todos los modelos de  $\alpha$ .

**Implicación:**  $\alpha \models \beta$  significa ' $\alpha$  implica  $\beta$ ' o ' $\beta$  se sigue de  $\alpha$ ' si y sólo si en todos los mundos donde  $\alpha$  es verdad,  $\beta$  también es verdad. Los mundos de  $\alpha$  son un subconjunto de los modelos de  $\beta$  ( $M(\alpha) \subseteq M(\beta)$ ).

#### 5.3 INFERENCIA DE PRUEBAS

**Prueba:** Demostración de implicación entre  $\alpha$  y  $\beta$ .

**Tipos de algoritmos:** Sólido o sound (deriva solo sentencias implicadas) y completo (deriva cualquier sentencia que esté implicada).

**Comprobación de modelos:** Para cada mundo posible, si  $\alpha$  es verdad, se asegura que  $\beta$  también lo sea.

**Demostración de teoremas:** Busca una secuencia de pasos de prueba que lleven de  $\alpha$  a  $\beta$ .

#### 5.4 LÓGICA PROPOSICIONAL

Dado un conjunto de símbolos de proposición  $\{X_1, X_2, \dots, X_n\}$ , se tienen las siguientes sentencias complejas:

$\neg$  (negación / no): Si  $\alpha$  es una sentencia, entonces  $\neg \alpha$  es una sentencia.

$\wedge$  (conjunción / y): Si  $\alpha$  y  $\beta$  son sentencias, entonces  $\alpha \wedge \beta$  es una sentencia.

$\vee$  (disyunción / o): Si  $\alpha$  y  $\beta$  son sentencias, entonces  $\alpha \vee \beta$  es una sentencia.

$\Rightarrow$  (implicación / implica que): Si  $\alpha$  y  $\beta$  son sentencias, entonces  $\alpha \Rightarrow \beta$  es una sentencia.

$\Leftrightarrow$  (bicondicional / si y sólo si): Si  $\alpha$  y  $\beta$  son sentencias, entonces  $\alpha \Leftrightarrow \beta$  es una sentencia.

Sea  $m$  un modelo que asigna verdadero o falso a  $\{X_1, X_2, \dots, X_n\}$ , se tiene que:

Si  $\alpha$  es un símbolo, entonces su valor de verdad se da en  $m$ .

$\neg \alpha$  es verdad en  $m$  si  $\alpha$  es falso en  $m$ .

$\alpha \wedge \beta$  es verdad en  $m$  si  $\alpha$  es verdad en  $m$  y  $\beta$  es verdad en  $m$ .

$\alpha \vee \beta$  es verdad en  $m$  si  $\alpha$  es verdad en  $m$  o  $\beta$  es verdad en  $m$ .

$\alpha \Rightarrow \beta$  es verdad en  $m$  si  $\alpha$  es falso en  $m$  o  $\beta$  es verdad en  $m$ .

$\alpha \Leftrightarrow \beta$  es verdad en  $m$  si  $\alpha \Rightarrow \beta$  es verdad en  $m$  y  $\beta \Rightarrow \alpha$  es verdad en  $m$ .

#### 5.5 TABLAS DE VERDAD

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
Falso	Falso	Verdadero	Falso	Falso	Verdadero	Verdadero
Falso	Verdadero	Verdadero	Falso	Verdadero	Verdadero	Falso
Verdadero	Falso	Falso	Falso	Verdadero	Falso	Falso
Verdadero	Verdadero	Falso	Verdadero	Verdadero	Verdadero	Verdadero

#### 5.6 TAREAS DE RAZONAMIENTO

Dadas una KB inicial más una secuencia de percepciones y acciones: La localización con un mapa y percepción local determina dónde estoy; el mapeado con un sensor de localización determina cuál es el mapa; el SLAM determina dónde estoy y cuál es el mapa; y la planificación determina qué secuencias de acciones están garantizadas que van a servir para alcanzar el objetivo.

## } 5.7 SATISFACIBILIDAD E IMPLICACIÓN

**Sentencia válida:** Sentencia que es verdad en todos los mundos.

**Sentencia satisfacible:** Sentencia que es verdad en al menos un mundo, cuyo problema se denomina SAT.

**Comprobación de implicación:**  $\alpha \models \beta$  ( $\alpha$  es la KB y  $\beta$  es lo que queremos probar),  $\alpha \Rightarrow \beta$  (es verdadero en todos los mundos),  $\neg(\alpha \Rightarrow \beta)$  (es falso en todos los mundos),  $\alpha \wedge \neg\beta$  (es falso en todos los mundos, es decir, insatisfacible).

## } 5.8 FORMA NORMAL CONJUNTIVA (CNF)

**Forma normal conjuntiva (CNF):** Cada sentencia puede expresarse como una conjunción ( $\wedge$ ) de cláusulas, donde cada cláusula es una disyunción  $\vee$  de literales y cada literal es un símbolo, negado o no.

## } 5.9 SOLUCIONADORES SAT EFICIENTES

**DPLL (Davis-Putnam-Logemann-Loveland):** Núcleo de los solucionadores modernos, que utiliza la enumeración recursiva en profundidad en todos los modelos con algunos extras, nombrados a continuación.

**Terminación anticipada (Early Termination):** Para si se satisfacen todas las cláusulas o si cualquier cláusula se vuelve falsa.

**Literales puros:** Si todas las apariciones de un símbolo en cláusulas aún no satisfechas tienen el mismo signo, entonces se le da ese valor al símbolo.

**Cláusulas unitarias:** Si una cláusula se queda con un único literal, se establece el valor del símbolo para satisfacer la cláusula.

## } 5.10 PLANIFICACIÓN COMO SATISFACIBILIDAD

**Caso observable y determinista:** El problema de planificación puede resolverse si y sólo si existe alguna asignación satisfactoria, donde la solución se obtiene a partir de los valores de verdad de las variables de acción.

**Caso  $T = [1, \infty)$ :** Se inicializa la KB con las físicas para  $T$  pasos temporales, para así afirmar que el objetivo es verdadero en el momento  $T$  (es importante leer las variables de acción de la solución SAT).

**Estimación del estado:** Implica hacer un seguimiento de lo que es cierto en el momento, dada una historia de observaciones y acciones.

## } 5.11 ENCADENAMIENTO HACIA DELANTE / ATRÁS Y RESOLUCIÓN

**Encadenamiento hacia delante (FC):** Aplica el Modus Ponens para generar nuevos hechos y requiere que la KB solo contenga cláusulas definidas, ya sean uno o varios símbolos.

**Propiedades:** Es sólido y completo para KB formadas por cláusulas definidas, se ejecuta en tiempo lineal y se utiliza en sistemas reales para actualizar bases de datos.

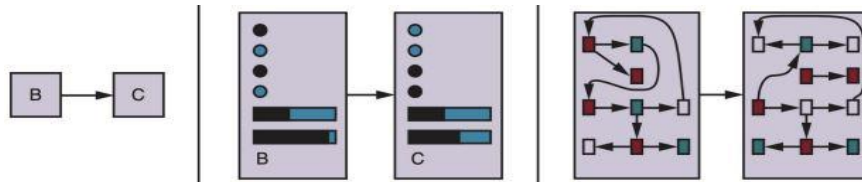
**Encadenamiento hacia atrás (BC):** Inverso al encadenamiento hacia delante.

**Propiedades:** Es sólido y completo para KB de cláusulas definidas, y puede ejecutarse en tiempo lineal si se maneja inteligentemente la indexación y el almacenamiento de la caché.

**Resolución:** Cada cláusula CNF puede escribirse como conjunción de símbolos  $\Rightarrow$  disyunción de símbolos, ésta regla toma dos cláusulas e infiere una de ellas resolviendo símbolos complementarios en lados opuestos de  $\Rightarrow$ . Es completo para la lógica proposicional y utiliza tiempo exponencial en el peor de los casos.

## TEMAS 6: LÓGICA DE PRIMER ORDEN

### 6.1 ESPECTROS DE REPRESENTACIONES



**Espectro atómico:** Cada estado del mundo es indivisible, no tiene estructura interna (problemas de búsqueda).

**Espectro factorizado:** Cada estado se divide en un conjunto de variables o atributos (planificación, lógica proposicional, redes Bayesianas y redes neuronales).

**Espectro estructurado:** Los objetos y sus distintas relaciones se describen explícitamente (lógica de primer orden, bases de datos, programas lógicos y programas probabilísticos).

### 6.2 PODER EXPRESIVO Y LENGUAJE NATURAL

**Objetivo:** Adoptar las bases de la lógica proposicional, independientemente del contexto y sin ambigüedades, construyendo sobre ellas una lógica más expresiva tomando ideas del lenguaje natural y evitando sus defectos.

**Elementos del lenguaje natural:** **Objetos** (sustantivos y grupos nominales), **relaciones unarias o n-arias** (verbos), **funciones** (un valor para cada entrada).

### 6.3 LÓGICA PROPOSICIONAL VS DE PRIMER ORDEN

**Lógica proposicional:** Se suponen hechos que se dan (o no) en el mundo, y cada hecho puede estar en estado verdadero o falso, donde cada modelo asigna un estado a cada símbolo de proposición.

**Lógica de primer orden:** Se supone que el mundo está formado por objetos con ciertas relaciones entre ellos que pueden o no mantenerse, donde los modelos son más complicados que en la lógica proposicional.

**Diferencias:** Distinto compromiso ontológico (diferencia en lo que asumen sobre la naturaleza de la realidad).

### 6.4 MUNDO POSIBLE

**Elementos:** Conjunto no vacío de objetos; un conjunto de tuplas de  $k$  objetos para cada predicado/relación  $k$ -ario en el lenguaje; un mapeo de tuplas de  $k$  objetos a  $k$  objetos para cada función  $k$ -aria en el lenguaje; y un objeto particular como las funciones constantes 0-arias para cada símbolo constante.

### 6.5 SINTAXIS Y SEMÁNTICA

**Términos:** Símbolo constante ( $A$ ,  $B$ , Juan), símbolo de función ( $BFF(Juan)$ ), variable lógica ( $x$ ).

**Símbolo de predicado ( $Knows(A, BFF(B))$ ):** Es verdadero si y sólo si los objetos a los que se refieren los términos están en la relación a la que se refiere el predicado),

**Igualdad entre términos ( $BFF(BFF(BFF(B)))=B$ ):** Es verdadero si y sólo si los términos se refieren a los mismos objetos.

**$\forall x Knows(x, BFF(x))$ :** Es verdadero en el mundo  $w$  si y sólo si es verdadero en todas las extensiones de  $w$  donde  $x$  se refiere a un objeto en  $w$ .

**$\exists x Knows(x, BFF(x))$ :** Es verdadero en el mundo  $w$  si y sólo si es verdadero en alguna extensión de  $w$  donde  $x$  se refiere a un objeto en  $w$ .

### 6.6 INFERENCIA EN LÓGICA DE PRIMER ORDEN

**Instanciación universal (UI):** Podemos inferir cualquier sentencia obtenida sustituyendo un término base por la variable ( $\forall x King(x) \wedge Greedy(x) \Rightarrow Evil(x) = King(John) \wedge Greedy(John) \Rightarrow Evil(John) = \dots$ ).

**Instanciación existencial:** La variable se sustituye por un nuevo símbolo constante

( $\exists x Crown(x) \wedge OnHead(x, John) = Crown(C1) \wedge OnHead(C1, John)$ ).

**Aplicación de reglas de inferencia a sentencias de primer orden:** La regla general es una versión elevada del Modus Ponens ( $KB = Person(Socrates), \forall x Person(x) \Rightarrow Mortal(x) = Mortal(Socrates)$ ).

**Unificación:** Encontrar sustituciones que hacen que expresiones lógicas diferentes parezcan idénticas ( $UNIFY(Knows(John, x), Knows(y, Mother(y))) = \{y/John, x/Mother(John)\}$ ).

## **▣ BLOQUE 4: INCERTIDUMBRE**

### **▣ TEMA 7: PROBABILIDAD**

#### **▣ 7.1 INCERTIDUMBRE**

**Problemas:** Observabilidad parcial; sensores con ruido; inmensa complejidad de modelización y predicción del tráfico o de la cola de seguridad; desconocimiento de la dinámica del mundo.

La teoría de la probabilidad es necesaria para tomar decisiones racionales y maximizar la utilidad esperada:

$$a^* = \underset{a}{\operatorname{argmax}} \sum_s P(s | a) U(s)$$

#### **▣ 7.2 LEYES BÁSICAS DE PROBABILIDAD**

**Leyes básicas de la probabilidad discreta:** Se empieza con un conjunto  $\Omega$  de mundos posibles y con un modelo de probabilidad que asigna un número  $P(\omega)$  a cada mundo  $\omega$ , que debe satisfacer  $0 \leq P(\omega) \leq 1$  y  $\sum_{\omega \in \Omega} P(\omega) = 1$  (la probabilidad de un suceso es la suma de las probabilidades de sus mundos).

**Leyes básicas de la probabilidad continua:** Un suceso/evento es cualquier subconjunto de  $\Omega$ , la probabilidad de un suceso es la suma de probabilidades sobre sus mundos.

#### **▣ 7.3 VARIABLES ALEATORIAS Y DISTRIBUCIONES**

**Variable aleatoria:** Es una función determinista de  $\omega$  cuyo rango es el conjunto de todos los posibles valores.

**Distribución de probabilidad:** Da la probabilidad para cada valor  $x$  en su rango, teniendo en cuenta  $P(X = x) = \sum_{\omega: X(\omega)=x} P(\omega)$ , siendo  $P(x)$  una abreviación cuando no es ambiguo y  $P(X)$  toda la distribución.

**Distribución de probabilidad continua:** Se tiene que, de las distribuciones marginales no puede sacarse la conjunta, pero, partiendo de la conjunta, sí pueden sacarse las marginales.

**Distribución marginal:** Son subtablas que eliminan variables, y en las que se puede realizar la marginalización, lo que permite colapsar una dimensión sumando  $P(X = x) = \sum_y P(X = x, Y = y)$ .

#### **▣ 7.4 PROBABILIDAD CONDICIONAL Y NORMALIZACIÓN**

**Probabilidad condicional (probabilidad de a dado b):**  $P(a|b) = \frac{P(a,b)}{P(b)}$ .

**Normalización:** Tiene como objetivo hacer que todas las probabilidades sumen 1, que consiste en multiplicar todas las entradas por  $\alpha = 1/(\text{suma sobre las entradas})$ .

#### **▣ 7.5 REGLAS DE PROBABILIDAD**

**Regla del producto:**  $P(a|b)P(b) = P(a,b) \Leftrightarrow P(a|b) = \frac{P(a,b)}{P(b)}$ .

**Regla de la cadena:**  $P(x_1, x_2, x_3) = P(x_3 | x_1, x_2) P(x_1, x_2) = P(x_3 | x_1, x_2) P(x_2 | x_1) P(x_1)$ , por lo que se tiene que  $P(x_1, x_2, \dots, x_n) = \prod_i P(x_i | x_1, \dots, x_{i-1})$ .

#### **▣ 7.6 INFERENCIA PROBABILÍSTICA**

**Inferencia probabilística:** Calcula una probabilidad deseada a partir de un modelo de probabilidad.

**Características:** Normalmente se utiliza para una variable de consulta dadas las evidencias; representan las creencias del agente dadas las pruebas; las probabilidades cambian con nuevas evidencias, lo que hace que se actualicen las creencias.

## 7.7 INFERENCIA POR ENUMERACIÓN

**Inferencia por enumeración:** Dado un modelo de probabilidad  $P(x_1, \dots, x_n)$ , se quieren particionar las variables  $x_1, \dots, x_n$  en variables de evidencia ( $E = e$ ), variables de consulta ( $Q$ ) y variables ocultas ( $H$ ), todo ello para obtener  $P(Q | e)$ .

**Características:** **Paso 1** (seleccionar las entradas coherentes con las evidencias), **paso 2** (marginalizar  $H$  del modelo para obtener la unión de la consulta y la evidencia,  $P(Q, e) = \sum_h P(Q, h, e)$ ), **paso 3** (normalizar,  $P(Q | e) = \alpha P(Q, e)$ ).

**Problemas:** Complejidad temporal  $O(d^n)$  en el peor de los casos, que puede ser exponencial en el número de variables ocultas; complejidad espacial  $O(d^n)$  para almacenar la distribución conjunta en memoria, donde  $O(d^n)$  son los puntos de datos que estiman las entradas de la distribución conjunta.

## 7.8 REGLA DE BAYES E INDEPENDENCIA

**Regla de Bayes:**  $P(a | b)P(b) = P(a, b) = P(b | a)P(a) \Leftrightarrow P(a|b) = \frac{P(b|a)P(a)}{P(b)}$ .

**Independencia:** Dos variables  $X$  e  $Y$  son (absolutamente) independientes si  $\forall x, y P(x, y) = P(x)P(y)$ , factorizando la distribución conjunta en un producto de dos distribuciones más simples, lo que se traduce en  $P(x, y) = P(x|y)P(y)$ , donde  $P(x | y) = P(x)$  y  $P(y | x) = P(y)$ .

**Independencia condicional:** Establece que  $X$  es condicionalmente independiente de  $Y$  dado  $Z$  si y sólo si  $\forall x, y, z P(x | y, z) = P(x | z)$  o si y sólo si  $\forall x, y, z P(x, y | z) = P(x | z)P(y | z)$ .



## TEMA 8: REDES BAYESIANAS

### 8.1 INTRODUCCIÓN

**Redes bayesianas:** Técnica para describir distribuciones conjuntas complejas usando distribuciones condicionales simples, además de ser un subconjunto de la clase general de modelos gráficos.

**Características:** Usan la causalidad local (independencia condicional) en un mundo compuesto por muchas variables, donde cada una interactúa localmente con otras pocas.

**Nodos:** Variables con dominios que pueden tener o no un valor asignado.

**Arcos:** Interacciones que indican la influencia directa entre variables. Formalmente hablando, la ausencia de arco codifica la independencia condicional.

**Probabilidades condicionales:** Se tienen  $P$  (variable padres) y tablas de probabilidad condicional (CPT).

### 8.2 SINTAXIS Y SEMÁNTICA

**Sintaxis:** Se tiene un conjunto de nodos (uno por cada variable  $X_i$ ), un grafo acíclico dirigido, una distribución condicional para cada nodo dadas sus variables padre en el grafo, y una tabla de probabilidad condicional (CPT), en la cual cada fila representa una distribución para el hijo dado un posible valor de sus padres, estableciendo que: **Red Bayesiana = Topología (gráfico) + Probabilidades condicionales locales.**

**Fórmula general para RBs dispersas:** Se tienen  $n$  variables con un tamaño máximo de rango  $d$  y un número máximo de padres  $k$ , cuya distribución conjunta completa es de tamaño  $O(d^n)$  y cuya red bayesiana es de tamaño  $O(n * d^k)$ , escalando linealmente con  $n$  siempre y cuando la estructura causal sea local.

**Semántica global:** Las redes bayesianas expresan distribuciones conjuntas como el producto de distribuciones condicionales en cada variable ( $P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Padres}(X_i))$ ).

### 8.3 INDEPENDENCIA CONDICIONAL EN REDES BAYESIANAS

**Independencia condicional:** Comparando la semántica global de las redes bayesianas con la identidad de la regla de la cadena ( $P(X_i | \dots, X_n) = \prod_i P(X_i | X_1, \dots, X_{i-1})$ ), se puede asumir sin perder generalidad, que

$X_1, \dots, X_n$  están ordenadas en orden topológico según el grafo, por lo que  $\text{Padres}(X_i) \subseteq X_1, \dots, X_{i-1}$ . Por lo tanto, la red de Bayes expresa independencias condicionales  $P(X_i | X_1, \dots, X_{i-1}) = P(X_i | \text{Padres}(X_i))$ , para lo que se eligen padres para el nodo  $X_i$  que lo cubran de otros predecesores que garanticen su validez.

### 8.4 ESTRUCTURA DE LAS REDES BAYESIANAS

**Estructura:** Toda variable es condicionalmente independiente de sus no descendientes dados sus padres, donde cada nodo es condicionalmente independiente de todas las demás variables dado su manto de Markov, compuesto por padres, hijos, y los padres de los hijos.

### 8.5 INFERENCIA POR ENUMERACIÓN EN REDES BAYESIANAS

**Inferencia por enumeración:** Implica calcular sumas de productos de números, aunque se puede dar el caso que el número de productos puede llegar a ser demasiado grande.

### 8.6 ELIMINACIÓN DE VARIABLES

**Idea:** Desplazar las sumas hacia el interior en la medida de lo posible, es decir,

$$P(R | j, m) = \sum_{t,a} P(R)P(t)P(a|R,t)P(j|a)P(m|a) = \alpha P(R) \sum_t P(t) \sum_a P(a|R,t) P(j|a) P(m|a).$$

**Proceso:** Hacer el cálculo de dentro hacia fuera, es decir, sumar primero sobre  $a$  y después sobre  $t$ .

**Problema:**  $P(a|R,t)$  no es un solo número, sino muchos números diferentes en función de los valores de  $R$  y  $t$ .  
**Solución:** Utilizar factores (matrices de números de varias dimensiones con operaciones apropiadas sobre ellas).

**Proceso:** Teniendo en cuenta que  $P(Q|E_1 = e_1, \dots, E_k = e_k)$ , se empieza con los factores iniciales (CPT locales instanciadas por evidencia). Mientras siga habiendo variables ocultas, se elige una de ellas ( $H_j$ ) y se elimina del producto de todos los factores que la mencionan. Y por último, se unen todos los factores restantes y se normaliza.

## 8.7 FACTORES

**Distribución conjunta ( $P(X, Y)$ ):** Compuesta por entradas  $P(x, y)$  para todo  $x, y$ , y por matrices  $|X|$  e  $|Y|$  que suman 1.

**Conjunta proyectada ( $P(x, Y)$ ):** Es una porción de la distribución conjunta compuesta por entradas  $P(x, y)$  para una  $x$  particular y todas las  $y$ , y un vector de  $|Y|$  elementos, cuyos valores suman  $P(x)$ , que establecen que el número de variables mayúsculas equivale a la dimensionalidad de la tabla.

**Condiciona único ( $P(Y | X)$ ):** Compuesta por entradas  $P(y | x)$  para una  $x$  fija y todas las  $y$ , que suman 1.

**Familia de condicionales ( $P(X | Y)$ ):** Compuesta por múltiples condicionales y entradas  $P(x | y)$  para todas las  $x, y$ , que suman  $|Y|$ .

**Producto punto a punto:** Se tiene que el nuevo factor tiene la unión de las variables de los dos factores originales, por lo tanto, cada entrada es el producto de las entradas correspondientes de los factores originales ( $P(J|A) \times P(A, J) = P(A, J)$ ).

**Aumento de factores (fórmula):**  $P(U, V) \times P(V, W) \times P(W, X) = P(U, V, W, X)$ .

**Suma de una variable:** Suma eliminando una variable desde un factor, encogiéndolo el factor a uno más pequeño, para lo que primero se proyectan los factores en cada sentido, y después se suman los productos ( $\sum_j P(A, J) = P(A, j) + P(A, \neg j) = P(A)$ ).

## 8.8 IMPORTANCIA DEL ORDEN

**Ordenando los términos como Z, A, B, C, D, se tiene que:**

$P(D) = \alpha \sum_{z,a,b,c} P(z)P(a|z)P(b|z)P(c|z)P(D|z) = \alpha \sum_z P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z)P(D|z)$ , donde el factor más grande tiene 2 variables (D,Z).

**Ordenando los términos como A, B, C, D, Z, se tiene que:**

$P(D) = \alpha \sum_{a,b,c,z} P(a|z)P(b|z)P(c|z)P(D|z)P(z) = \alpha \sum_a \sum_b \sum_c \sum_z P(a|z)P(b|z)P(c|z)P(D|z)P(z)$ , donde el factor más grande tiene 4 variables (A,B,C,D), por lo que, teniendo  $n$  hojas, su factor es de tamaño  $2^n$ .

## 8.9 COMPLEJIDAD COMPUTACIONAL Y ESPACIAL

**Complejidad computacional y espacial:** Está determinada por el factor más grande, donde el orden de eliminación puede afectar en gran medida al tamaño del factor más grande.

## 8.10 POLIÁRBOLES

**Poliárbol:** Grafo dirigido sin ciclos no dirigidos, es decir, aquellos en los que no aparecen bucles cuando se quitan las flechas. La complejidad de eliminación de variables de estos grafos es lineal en el tamaño de la red si se va eliminando de hojas a raíces.

## TEMA 9: RAZONAMIENTO PROBABILÍSTICO EN EL TIEMPO

### 9.1 INTRODUCCIÓN

Un agente toma decisiones en función de su estado.

En un entorno parcialmente observable, este agente debe estimar su estado actual con la ayuda de sus sensores. Para esto necesita:

Mantener una estimación del estado o estados más probables.

**Modelo de transiciones:** A partir de la estimación de estado, indica cómo evoluciona el mundo en el próximo instante de tiempo.

**Modelo de sensor:** A partir de las percepciones, el agente pueda actualizar su estimación de estado.

**Teoría de la Probabilidad:** Cuantifica esta estimación del estado y estos modelos.

**Mundo cambiante:** Se modela usando una variable para cada aspecto del mundo en cada punto en el tiempo.

### 9.2 TIEMPO E INCERTIDUMBRE

Nuestro mundo ya no es estático, como hasta ahora, en el que una variable aleatoria tiene un valor fijo (reparar un coche, tratar a un paciente de diabetes o localizar un robot).

El mundo se modela como una secuencia de instantes de tiempo, cada uno con unas variables aleatorias:

Conjunto de variables no observables  $X_t$  y un conjunto de variables de evidencia observables  $E_t$ , donde la observación en  $t$  es  $E_t = e_t$  para algunos valores de  $e_t$ .

Debemos especificar cómo evoluciona el mundo (modelo de transición) y cómo las variables de evidencia obtienen sus valores (modelo de sensor).

**Modelo de transición:**  $P(X_t | X_{0:t-1})$ .

**Supuesto de Markov:** El estado actual depende solo de un número fijo finito de estados previos

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1}).$$

**Procesos estacionarios:** Procesos de cambio que se rigen por leyes que no cambian con el tiempo, donde  $P(R_t | R_{t-1})$  es el mismo para todo  $t$  (sólo tenemos que especificar un CDT).

**Modelo de sensor:**  $P(E_t | X_{0:t}, E_{0:t-1}) = P(E_t | X_t)$

**Supuesto de Markov del sensor:** El estado actual es suficiente para generar las actuales percepciones ( $P(X_{0:t}, E_{1:t}) = P(X_0) \prod P(X_t | X_{t-1}) P(E_t | X_t)$ ), donde  $P(X_0)$  es el estado inicial del modelo,  $P(X_t | X_{t-1})$  es el modelo de transición y  $P(E_t | X_t)$  es el modelo del sensor.

**Supuesto de Markov de primer orden:** Las variables de estado contienen toda la información necesaria para caracterizar la distribución de probabilidad para el siguiente instante de tiempo. Si no se cumple, podemos incrementar el orden de Markov y el conjunto de variables de estado.

### 9.3 INFERENCIA EN MODELOS TEMPORALES

**Filtrado:** Calcula el estado actual dadas todas las evidencias hasta el momento  $P(X_t | e_{1:t})$ . Es lo que hace un agente racional para realizar un seguimiento del estado actual para tomar decisiones racionales.

**Predicción:** Calcula la distribución sobre el estado futuro dada toda la evidencia hasta la fecha  $P(X_{t+k} | e_{1:t})$  para  $k > 0$ . Es útil para evaluar posibles cursos de acción según los resultados esperados.

**Suavizado:** Calcular la distribución sobre un estado pasado, dada toda la evidencia hasta el presente  $P(X_k | e_{1:t})$  para  $0 \leq k < t$ . El suavizado proporciona una mejor estimación del estado de la que estaba disponible en ese momento, porque incorpora más evidencia.

**Explicación más probable:** Dada una secuencia de observaciones, podríamos desear encontrar la secuencia de estados que es más probable que haya generado esas observaciones. Es decir, deseamos calcular  $\arg\max_{X_{1:t}} P(X_{1:t} | e_{1:t})$ . Los algoritmos para esta tarea son útiles en muchas aplicaciones, incluido el

reconocimiento de voz, donde el objetivo es encontrar la secuencia de palabras más probable, dada una serie de sonidos, y la reconstrucción de cadenas de bits transmitidas a través de un canal ruidoso.

**Aprendizaje:** Si aún no se conocen los modelos de transición y sensor, se puede aprender de las observaciones. La inferencia proporciona una estimación de qué transiciones ocurrieron realmente y de qué estados generaron lecturas del sensor, y estas estimaciones se pueden utilizar para actualizar los modelos.

**Filtrado y predicción:**  $P(X_{t+1} | e_{1:t+1}) = f(e_{t+1}, P(X_t | e_{1:t}))$ .

**Estimación recursiva:**

$$P(X_{t+1} | e_{1:t+1}) = P(X_t | e_{1:t}, e_{t+1}) = \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1}, e_{1:t}) = \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1}, e_{1:t}).$$

**Resultando todo esto en:**  $P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) \sum_{X_t} P(X_{t+1} | X_t) P(X_t | e_{1:t})$ .

**Predicción:** Continuar con el filtrado sin incluir evidencias:  $P(X_{t+k+1} | e_{1:t}) = \sum_{X_{t+k}} P(e_{t+k+1} | X_{t+k}) P(X_{t+k}, e_{1:t})$ .

### 9.4 MODELOS OCULTOS DE MARKOV

**Modelo oculto de Markov (HMM):** Modelo probabilístico temporal en el que el estado del proceso se describe mediante una única variable aleatoria discreta.

**Modelo de Transición  $P(X_t | X_{t-1})$ :** Matriz T de tamaño SxS donde S es el número de posibles estados

$$T_{ij} = P(X_t = j | X_{t-1} = i).$$

**Modelo del Sensor:** Necesitamos especificar, para cada estado, qué tan probable es que el estado haga que aparezca  $e_t$  ( $P(e_t | X_t = i)$ ) para cada estado i ( $f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t}$ ).

### 9.5 FILTROS DE KALMAN

Es un estimador óptimo recursivo para variables continuas y multivariable:

$$P(X_{t+\Delta} = x_{t+\Delta} | X_t = x_t, X'_t = x'_t) = N(x_t + x'_t \Delta, \sigma^2)(x_{t+\Delta}).$$

Si la distribución actual  $P(X_t | e_{1:t})$  es gaussiana y el modelo de transición  $P(X_{t+1} | x_t)$  es gaussiano lineal,

entonces la predicción de la distribución es  $P(X_{t+1} | e_{1:t}) = \int_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t}) dx_t$ .

Si la predicción  $P(X_{t+1} | e_{1:t})$  es gaussiana y el modelo de sensor  $P(e_{t+1} | X_{t+1})$  es lineal gaussiano, la

distribución actualizada es también gaussiana  $P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t})$  ( $P(X_0) = N(\mu_0, \Sigma_0)$ ).

### 9.6 FILTROS DE PARTÍCULAS

Se genera una población de partículas de la distribución  $P(X_0)$  y el siguiente ciclo de manta continua:

**Predicción** (por cada partícula, se calcula  $x_{t+1}$  a partir de  $x_t$  siguiendo  $P(X_{t+1} | x_t)$ ), **Actualización** (cada muestra se pondera por la probabilidad de que se asigne a la nueva evidencia  $P(e_{t+1} | x_{t+1})$ ), **Resampling** (se reconstruye la distribución  $P(X_{t+1})$  a partir de la población de partículas generando una nueva.