

## **Preguntas parcial 2- IA**

### **1.- Componentes de un juego.**

Un juego es cualquier situación de decisión con varios agentes, (jugadores) gobernada por un conjunto de reglas y con un resultado bien definido, caracterizada por que ninguno de los jugadores con su sola actuación puede determinar el resultado (interdependencia estratégica).

- **Características de los juegos:**

Las características principales de los juegos son: El número de jugadores, la información del juego (juegos de información perfecta y juegos de información imperfecta), la existencia de movimientos de azar, el orden de actuación de los jugadores, la existencia de pagos colaterales y si son juegos de suma nula o no.

- **Objetivos:** Conjunto de metas a conseguir por cada jugador dentro del juego.
- **Reglas:** Definen la viabilidad de las actuaciones y son el marco en el que los jugadores realizan sus acciones.
- **Estrategias:** Serie de actuaciones premeditadas encaminadas hacia un fin determinado.
- **Estado Inicial:** Situación en la que comienza el juego y determina las posibles estrategias de los jugadores.
- **Jugadores:** Se les denomina Min y Max donde Max inicia la partida.
- **Función sucesor:** Determina, para cada actuación legal, el estado al que conduce.
- **Función utilidad:** Determina los resultados de la partida en nodos terminales. Los resultados favorables a Max serán positivos y los favorables a Min, negativos. En caso de empate el resultado será 0.
- **Función terminal:** Determina si un nodo es o no terminal.

## **2.- Qué es el factor de ramificación y cómo afecta a la complejidad de un juego? Describe en líneas generales el algoritmo minimax y el de la poda alfa-beta**

El Factor de ramificación es el número de nodos hijos en cada nodo. Si este valor no es uniforme, se puede calcular el factor de ramificación medio. Cuanto más mayor sea el factor de ramificación las posibilidades a la hora de explorar un nodo serán mayores lo que cause un aumento en la complejidad.

- **Algoritmo MiniMax**

Es un algoritmo usado para tomar decisiones en juegos bipersonales competitivos de suma nula con información perfecta.

- Cada nodo hoja se etiqueta con 1,-1,0, si el ganador es Max, Min o empate respectivamente.
- El objetivo es encontrar un conjunto de movimientos legales que de como ganador a Max.
- Se propagan los valores de las jugadas terminales de las hojas hasta la raíz, una jugada ganadora para Max es un subárbol en el que todos los nodos terminales resultan ganadores.
- Incluso un juego simple es demasiado complejo para dibujar el árbol de juego entero.

El algoritmo calcula la acción que conduce al mejor resultado para el jugador. En el caso de Max conduce al mayor valor posible y en el caso de Min al menor. Para ello genera un árbol de juego con todos los estados posibles que resultan de las distintas acciones de los jugadores hasta obtener los nodos terminales, calcula los valores de dichos nodos terminales utilizando la función de utilidad y los de los nodos no terminales basándose en los valores de sus hijos.

El algoritmo se trata de una exploración en profundidad en la que se debe explorar todo el árbol por lo que el tiempo necesario para calcular el árbol de un caso complejo puede ser inabarcable.

Para solucionar esto surgió el algoritmo de Poda ALFA-BETA con el que conseguimos eliminar gran parte del árbol, afectando enormemente al tiempo de computo sin afectar al resultado final.

- **Poda ALFA-BETA**

Esta búsqueda alfa-beta va actualizando el valor de los parámetros según se recorre el árbol. El método realizará la poda de las ramas restantes cuando el valor actual que se está examinando sea peor que el valor actual de  $\alpha$  o  $\beta$  para MAX o MIN, respectivamente. El procedimiento es el siguiente:

- Los dos parámetros alfa y beta describen los límites sobre los valores que aparecen a lo largo del camino:
  - $\alpha$  = El valor de la mejor opción (el más alto) que se ha encontrado hasta el momento en cualquier punto del camino, para MAX.
  - $\beta$  = El valor de la mejor opción (el más bajo) que se ha encontrado hasta el momento en cualquier punto del camino, para MIN.
- Al principio del proceso de búsqueda alfa toma el valor  $-\infty$ , mientras que beta toma el valor  $+\infty$
- Desde un nodo MAX, iremos actualizando el valor de alfa a medida que vayamos explorando cada arco que cuelgue de ese nodo.
- Dicha actualización consiste en comparar el valor obtenido por cada arco con el valor actual alfa. Si el valor obtenido es mayor, éste será el nuevo valor de alfa. Por tanto, alfa almacena el mejor valor que hemos encontrado desde el nodo en que estamos. Solo nos interesará proseguir la búsqueda desde ese nodo si podemos mejorar alfa.
- En caso de que estemos en un nodo MIN, iremos actualizando el valor de beta a medida que vayamos explorando cada arco que cuelgue de ese nodo. Dicha actualización consiste en comparar el valor obtenido por cada arco con el valor actual beta. Si el valor obtenido es menor, éste será el nuevo valor de beta.
- Por tanto, beta almacena el peor valor (para MAX) encontrado por MIN hasta el momento. Solo interesará que la búsqueda prosiga si MIN pudiera reducir beta,
- El intervalo (alfa, beta) contiene los valores que se puede seguir consiguiendo en el proceso de búsqueda.
- Cuando alfa crece tanto que sobrepasa a beta, o cuando beta disminuye tanto que se hace menor que alfa, se puede hacer una poda y no seguir explorando el subárbol que cuelga desde el nodo en que nos encontramos, prosiguiendo la búsqueda en profundidad desde su nodo padre.

### 3.- ¿Que problemas plantea el cálculo de predicados en la resolución de problemas de IA?

- **Problemas Semánticos**

- **Es difícil expresar todo en fórmulas lógicas.**
- **Razonamiento temporal:** La lógica de predicados admite muy mal el manejo del tiempo. El tiempo supone en cierta medida el cambio del mundo, lo que implica que cosas que eran verdaderas pueden volverse falsas y viceversa.
- **Razonamiento acerca de predicados:** No se pueden usar un predicado como termino de otro predicado.
- **Información incompleta o imprecisa (vaguedad, probabilidad):** La lógica de predicados es una lógica aristotélica (solo admite 0 y 1).
- **Excepciones:** En la lógica de predicados cuando se dice que algo es verdad no existen excepciones, para contemplar excepciones se utiliza la circunscripción lógica para rodear el entorno donde la excepción no existe.

- **Problemas Computacionales**

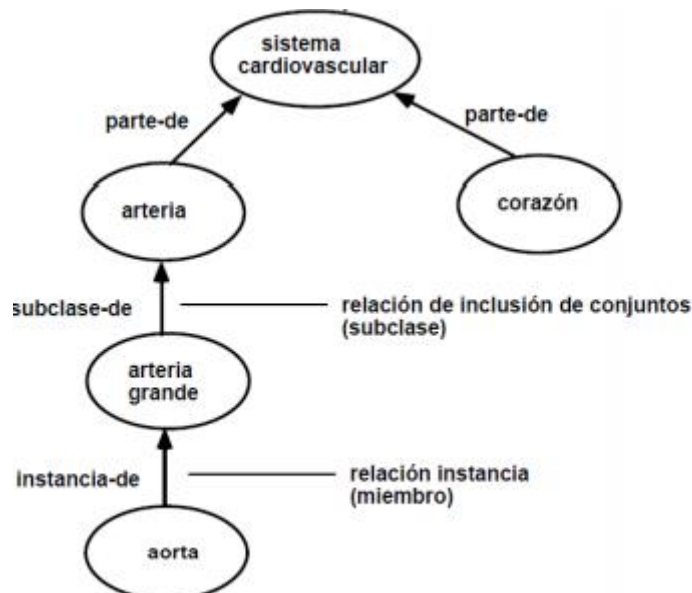
- **Consistencia:** Lo que se demuestra como verdadero lo es realmente. Refutación por solución.
- **Complejidad:** Si algo es verdadero siempre podrá demostrarse y si no se puede demostrar no se puede decir que sea falso, solo que no se tienen axiomas suficientes para probar su veracidad.
- **Complejidad Computacional:** La lógica de predicado es exponencial, lo que quiere decir que si queremos demostrar un conjunto de 100 axiomas la complejidad será de  $X^{100}$  ya que tiene que hacer todas las combinaciones posibles.

#### 4.- Modelos de conocimiento heredable ¿Qué tipo de conocimiento organizan las redes semánticas? Describir en líneas generales el concepto de “frame”.

Los modelos de conocimiento heredable se basan en la jerarquización del conocimiento de forma que se puedan inferir características o atributos de ciertos elementos a partir de otras ya conocidas. Se utilizan para tareas de reconocimiento ya que la información contenida en un marco pueden hacer que se activen otros marcos conectados con los primeros, dando lugar así a una red de activación que permite predecir y explicar la información que deriva de marcos activados inicialmente, al conocimiento que surge de este mecanismo se le suele denominar herencia o reconocimiento descendiente porque se determinan propiedades desconocidas de ciertos elementos a partir de otras ya conocidas que las encapsulan (cada clase hereda de la superclase).

Por otro lado, las llamadas redes asociativas están formadas por nodos que representan un concepto enlazados entre sí de forma que se representan relaciones de inclusión, pertenencia o causalidad o bien a categorías gramaticales, Se conocen:

- **Redes semánticas:** Son redes destinadas a representar o a comprender el lenguaje natural.
  - **Ejemplo:** “El corazón es parte del sistema cardiovascular”, “Las arterias son parte del sistema cardiovascular”, “Las arterias grandes son arterias”, “La aorta es una arteria”



- **Redes de clasificación:** Es una clasificación de objetos o conceptos con sus características propias.

- **Redes casuales:** Son las que llevan asociadas una relación de influencia o causa entre los nodos que las forman.

Un frame es una estructura de datos para representar una situación estereotipada. Cada marco se caracteriza por un conjunto de slots que se asocian en general a atributos, y que en conjunto sirven para identificar los marcos.

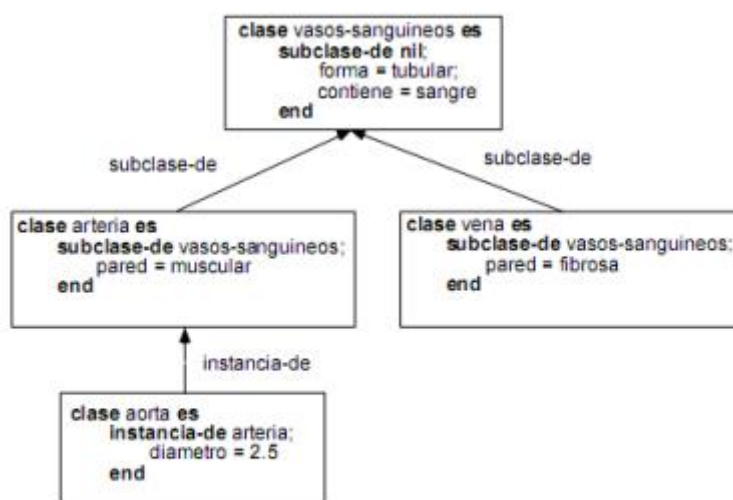
Los frames están especialmente concebidos para tareas de reconocimiento: La información recibida hace que se activen unos marcos y ésta a su vez provoca la activación de otros marcos conectados con los anteriores, dando lugar así a una red de activación, cuyo objetivo es predecir y explicar la información que se va a encontrar en esa situación. A esto se le denomina reconocimiento descendente.

Los frames proporcionan un formalismo para agrupar explícitamente todo el conocimiento concerniente a las propiedades de objetos individuales o clases de objetos, estos solo pueden tener una superclase (herencia simple).

Hay dos tipos de frame:

- **Frame de clase:** Representan conocimiento de clases de objetos
- **Frame instancia:** Representan conocimiento de objetos individuales

Ejemplo:



## 5.- Estructura y componentes de un sistema experto

- **Sistema Basado en el Conocimiento (SBC):** Un Sistema Basado en el Conocimiento es aquel Programa (en general sistema hardware-software) que emplea masivamente conocimiento para resolver un problema dentro de un dominio determinado. Un SBC necesita 3 componentes básicos:
  - Una **Base de Conocimiento** que contenga el conocimiento necesario sobre el dominio del problema a resolver.
  - Un **Motor de Inferencia** que permite razonar sobre el conocimiento de la Base de Conocimiento y los datos proporcionados por un usuario.
  - Una **Interfaz de Usuario**.
- **Sistema Experto (SE):** Los Sistemas Expertos son aquellos capaces de comportarse como un humano (resolviendo problemas, siendo consultado y justificando su razonamiento) en un determinado dominio de la actividad.
- **Sistema Experto Basado en Reglas (SEBR):** Los SEBR se pueden definir como un SBC donde el conocimiento se incluye en forma de reglas y hechos, siendo estos los Sistemas Expertos más clásicos y comunes.

Aunque estas reglas y hecho pueden implementarse, por ejemplo, mediante el cálculo de predicados, no obstante, es más “adecuado” emplear reglas de producción para las reglas.

El proceso de construcción de un SEBR es el siguiente:

1. Se extrae el conocimiento experto.
2. Se modela y se adquiere el conocimiento utilizando el lenguaje adecuado.
3. Se crea la Base de Conocimiento con el conocimiento adquirido.

En cuanto a la extracción de conocimiento en SEBR:

Los métodos para extraer conocimiento se estudian en el área de aprendizaje automático dentro de la IA (extracción de reglas o aprendizaje de reglas). Además, normalmente, la percepción del mundo por parte de un agente no es perfecta. Del mismo modo, es posible que una regla no sea aplicable siempre (aunque si en un gran número de casos). Este hecho no permite que la regla sea admitida en un sistema de cálculo proposicional o de predicados, dado que daría lugar a sistemas inconscientes, por lo que se hace necesario establecer mecanismos para tratar con incertidumbre.

## 6.- Paradigmas de Aprendizaje Automático.

Los principales paradigmas de aprendizaje automático son:

- **Aprendizaje Memorístico:**

Surge cuando el aprendizaje consta de asociaciones arbitrarias o cuando el sujeto lo hace de forma arbitraria. Los datos se almacenan sin tratar de comprenderlos y de forma repetitiva.

- **Aprendizaje deductivo:**

Consiste en aplicar la inferencia y deducción empleando modelos lógicos. Es decir, es la obtención de nuevo conocimiento a partir de conocimientos que ya se poseen.

- **Aprendizaje analítico:**

Basado en explicaciones. Construir una explicación para cada ejemplo en relación con un concepto dado y generalizar la explicación de modo que pueda emplearse en el futuro.

- **Aprendizaje analógico:**

Se buscan soluciones a problemas nuevos encontrando similitudes con problemas ya conocidos y adaptando sus soluciones. Se basa en la idea de que si dos situaciones son similares en algún aspecto, entonces también pueden serlo en otros.

- **Aprendizaje Inductivo:**

Se trata de aprender un concepto o una clasificación a partir de ejemplos y contraejemplos. Se formulan hipótesis mediante la búsqueda de regularidades en unos ejemplos de 'entrenamiento' observados previamente y se aceptan o rechazan dichas hipótesis con la aparición de nuevos ejemplos. El aprendizaje inductivo combinado con el analítico es uno de los métodos más potentes, aunando las ventajas de ambos (mejor precisión en la generalización y dependencia en el conjunto de entrenamiento cuando existe poco conocimiento previo)

Los aprendizajes también se pueden dividir en función del conocimiento utilizado, estos son:

- **Aprendizaje Supervisado:**

Para cada entrada se dispone de un supervisor que proporciona una salida deseada ya sea una clase o un valor a aproximar.



- **Aprendizaje no Supervisado:**

No se dispone de una salida deseada para cada entrada, sino que se busca agrupar los datos en función de ciertas características.

- **Aprendizaje por Refuerzo:**

Se aprende a partir de la información obtenida al realizar procesos de ensayo-error en los que se obtienen señales de beneficio/coste.

## **7.- Describir el problema del ruido y el del sobreajuste en aprendizaje automático.**

- **Ruido**

El ruido se puede producir porque haya demasiados datos y no se puedan clasificar, o porque dos datos iguales se encuentran clasificados en clases diferentes lo que, al no saber si el error está en la clasificación o en los datos, causa que decidir qué datos son buenos y cuales malos sea más complicado.

Para solucionar esto ponemos las dos clasificaciones iguales al primer vector e iguales al segundo vector y probamos con el conjunto de tests para así ver cual de las dos descripciones es más correcta o puede ser más correcta.

- **Sobreajuste**

Hay sobreajuste cuando se ajusta, muy bien a los conjuntos de entrenamiento, pero muy mal al test. Esto se debe a que se han aprendido también datos con errores.

Se dice que una hipótesis  $h$  se sobreajusta al conjunto de entrenamiento si existe alguna hipótesis  $h'$  tal que el error de  $h$  es menor que el de  $h'$  sobre el conjunto de entrenamiento, pero es mayor sobre la distribución completa de ejemplos del problema.

## 8.- ¿Qué son y como se construyen los árboles de decisión

Un árbol de decisión toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve una “decisión”, el valor previsto de la salida dada la entrada. Los atributos de entrada pueden ser discretos o continuos, y según como sea la salida se tratará de un árbol de clasificación (Salida discreta) o de regresión (Salida continua).

Los árboles de decisión tienen una estructura similar a los árboles convencionales, diferenciándose en sus nodos no terminales donde, en los árboles de decisión, guardan una condición sobre un atributo. Para elegir el atributo es buena idea que esté divida el conjunto de ejemplos en subconjuntos que sean o “todos positivos” o “todos negativos”

Se parte de una tabla con observaciones que son ejemplos de aprendizaje, se escoge un atributo y se pregunta sobre sus valores y como clasificarlos, seguidamente se va buscando en cada paso el que mejor clasifica y finalmente se obtiene el árbol de decisión.

- **Árbol Trivial**

Se crea una rama por cada observación. Son árboles excesivamente grandes y no funcionan bien con instancias nuevas.

- **Árbol optimo**

Es el árbol más pequeño posible, pero para ello hay que generar todos los árboles posibles y no posibles lo que lo convierte en un método inviable computacionalmente.

- **Árbol Pseudo-Optimo**

Se utiliza una heurística de selección de atributos, consiste en elegir en cada paso el atributo que mejor clasifica. Es pseudo-optimo porque quizás alternando un par de atributos se clasificaría mejor.