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 porque ninguno de los jugadores con su sola actuación puede determinar el resultado (interdependencia estratégica).

Formalmente, se podría definir como una clase de problemas de búsqueda con los componentes siguientes:

• El **estado inicial**, que incluye la posición del tablero e identifica al jugador que mueve.
• Una **función sucesor**, que devuelve una lista de pares (*movimiento, estado*), indicando un movimiento legal y el estado que resulta.
• Un **test terminal**, que determina cuándo se termina el juego. A los estados donde el juego se ha terminado se les llaman **estados terminales**.
• Una **función utilidad** (también llamada función objetivo o función de rentabilidad), que da un valor numérico a los estados terminales. En el ajedrez, el resultado es un triunfo, pérdida, o empate, con valores +1, -1 o 0. Algunos juegos tienen una variedad más amplia de resultados posibles: las rentabilidades en el *backgammon* se extienden desde +192 a -192

Clasificación de los juegos

• Juegos de suma nula: La ganancia o pérdida de un jugador se equilibra con la ganancia o pérdida del otro.
• Juegos de suma no nula: Se intenta maximizar el beneficio sin que importe si el resto de jugadores gana o pierde.
• Información perfecta: Los jugadores conocen todo lo que ha pasado desde el principio del juego. No hay que confundirlos con los de información completa que requiere que cada jugador conozca las estrategias y recompensas del resto, pero no necesariamente las acciones (dilema del prisionero).
• Información imperfecta
• Número de jugadores
Bipersonales: solo juegan dos jugadores
Multijugador: Pueden ser de dos tipos: cooperativos (se ayudan entre ellos); no cooperativos (se intenta maximizar el beneficio y hacer que el adversario gane lo mínimo posible).

[El estado inicial y los movimientos legales para cada jugador definen el árbol de juegos.
• Número de jugadores: No es lo mismo un juego bipersonal que un juego n-personal (con n>= 3) donde puede haber pagos colaterales debido a alianzas entre jugadores.]

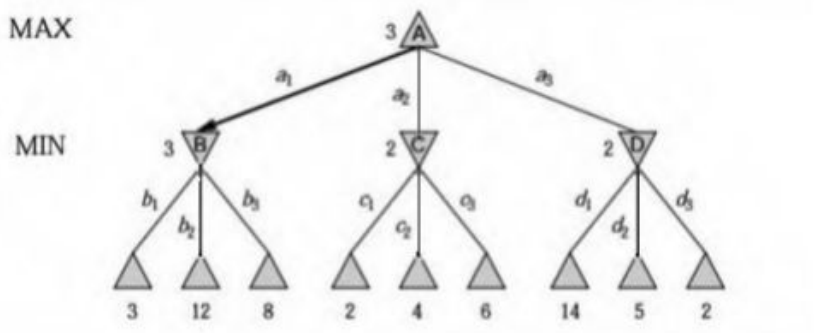
2- Describir en líneas generales el algoritmo minimax.

Minimax es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario bipersonales, de suma nula y con información perfecta. Minimax es un algoritmo recursivo, cuyo funcionamiento puede resumirse en cómo elegir el mejor movimiento para ti mismo suponiendo que tu contrincante escogerá el peor para ti.

Consideraremos juegos con dos jugadores, que llamaremos MAX y MIN. El algoritmo minimax busca la estrategia óptima, desde el punto de vista de un jugador, suponiendo que ambos juegan de forma imbatible.

Considerando un árbol de juegos, la estrategia óptima puede determinarse examinando el valor minimax de cada nodo. Este valor es la heurística de la búsqueda (Valor-Minimax) de estar en el estado correspondiente asumiendo que ambos jugadores juegan óptimamente desde allí hasta el final del juego. El valor minimax de un nodo terminal es solamente su utilidad. Además hay que tener en cuenta que MAX preferirá moverse a un estado de valor máximo, mientras que MIN prefiere un estado de valor mínimo. Entonces tenemos:

$$\text{VALOR-MINIMAX}(n) = \begin{cases} \text{UTILIDAD}(n) & \text{si } n \text{ es un estado terminal} \\ \max_{s \in \text{Sucesores}(n)} \text{VALOR-MINIMAX}(s) & \text{si } n \text{ es un estado MAX} \\ \min_{s \in \text{Sucesores}(n)} \text{VALOR-MINIMAX}(s) & \text{si } n \text{ es un estado MIN} \end{cases}$$



Por ejemplo, en el árbol de arriba, el algoritmo primero va hacia abajo a los tres nodos izquierdos, y utiliza la función heurística para descubrir que sus valores son 3, 12 y 8 respectivamente. Entonces toma el mínimo de estos valores, 3, y lo devuelve como el valor del nodo B. Un proceso similar

devuelve hacia arriba el valor de 2 para C y 2 para D. Finalmente, tomamos el máximo de 3, 2 y 2 para conseguir el valor de 3 para el nodo de raíz.

Por tanto podemos ver cómo el algoritmo minimax realiza una exploración primero en profundidad completa del árbol de juegos.

3- Paradigmas de Aprendizaje Automático.

El aprendizaje al igual que la inteligencia es un concepto difícil de definir porque involucra a un amplio abanico de procesos y mecanismos, sin embargo es una capacidad esencial de la inteligencia humana. De ahí su importancia para la IA, por razones teóricas o científicas (manifestación de inteligencia) y por razones de eficacia técnica, especialmente cuando existen patrones o conocimiento difíciles de describir o descubrir. En el campo del **aprendizaje automático** encontramos, principalmente, los siguientes **paradigmas de aprendizaje**:

- **Aprendizaje memorístico**: los datos se almacenan sin tratar de comprenderlos o de inferirlos a partir de otros ya conocidos y de forma repetitiva. Por ejemplo, si un sistema recibe, ante una determinada acción en cierta circunstancia, una respuesta desventajosa, puede 'memorizar' que esa acción da malos resultados y, si esos malos resultados se repiten en el tiempo, concluir (aprender) que esa no es una acción adecuada para dicha situación.
- **Aprendizaje deductivo**: es la obtención de nuevo conocimiento a partir de conocimientos que ya se poseen. También es el razonamiento artificial (dar una explicación a un evento observado a partir de conocimientos previos y crear así nuevo conocimiento).
- **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**: es el paradigma más ampliamente estudiado dentro del aprendizaje. 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 (ejemplos) 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.

Tipos de aprendizaje según su conocimiento:

- **Supervisado.** Se dispone de un profesor/supervisor que proporciona una salida deseable para cada entrada percibida, ya sea una clase o un valor a aproximar (clasificación vs regresión).
- **No supervisado.** No se dispone de una salida deseada en cada entrada, sino que se busca agrupar/clasificar los datos en función de ciertas características (medida de distancia).

- **Refuerzo.** Se aprende (sin supervisor) a partir de la información obtenida al realizar procesos de ensayo-error en los que se obtienen "señales" de beneficio/coste.

4- Describir el problema del ruido y el del sobreajuste en aprendizaje automático.

Uno de los problemas a los que se enfrentan los sistemas de aprendizaje, y que provocan el sobreajuste, es cuando los ejemplos de entrenamiento contienen ruido:

• Valores de atributos erróneos, subjetivos

• Clasificación equivocada

• Valores desconocidos

En general, existen dos métodos para manejar ruido (basados en la condición de terminación):

• Pruning (o pre-pruning): Cambiar el criterio de paro del árbol de decisión para "podar" ramas.

• Post-pruning: "Podar" ramas una vez construido el árbol.

Estas operaciones de «poda» y de aumento de la tolerancia al ruido nos ayudarían a minimizar el riesgo de sobreajuste.

El sobreajuste es el efecto que se produce al sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado. El algoritmo de aprendizaje debe alcanzar un estado en el que será capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas a las acaecidas durante el entrenamiento. Sin embargo, cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación con la función objetivo.

5- ¿Qué problemas plantea el cálculo de predicados en la resolución de problemas de IA?

El cálculo de predicados presenta dos grupos de problemas:
·Los problemas semánticos.
Conciernen problemas de expresión en fórmulas lógicas, como puede ser el caso de una heurística (ordenar los predicados, reglas frecuentes), expresar el metaconocimiento, una jerarquía y herencia, igualdades con sentido común.
En razonamiento temporal y de predicados: la variable del tiempo es difícil de manejar y no se puede escribir en lógica de predicados al igual que ocurre con las heurísticas; las bases de

conocimiento, en conocimiento experto, están ordenadas por la frecuencia. Se relacionan el metaconocimiento y la heurística, tratandose como iguales los predicados.
Que haya información incompleta, es decir, que falten datos; o imprecisa: vaga, con probabilidad o desconocimiento a priori (Aplicando el principio de Laplace: Da la misma probabilidad a todo).
Excepciones (relacionadas con la herencia y la jerarquía) y monotonía (haber deducido algo bajo excepciones)
·Los problemas computacionales.
Encontrar la consistencia, solidez, que tenga la garantía de la verdad absoluta; lo que se demuestra como verdadera lo es realmente. Se suele demostrar por refutación por resolución.
Completitud, que solo se da en la refutación por resolución: si una cosa es verdadera, puede demostrarse.
La complejidad computacional, o tratabilidad del predicado, sin un algoritmo en tiempo polinomial. Haciendo que sean de tipo NP (nondeterministic polynomial time)

1.PROBLEMAS SEMÁNTICOS:

1.1.Es dificil de expresar todo en formulas lógicas: heurísticas (ordenando los predicados o reglas frecuentes), metaconocimiento, jerarquía (subconjunto) y herencia (cumple las propiedades básicas), igualdad (ligado con el razonamiento acerca del predicado) y sentido común (relacionado con las heurísticas, metaconocimiento).

1.2.Razonamiento temporal: Es un problema la variable del tiempo, se maneja muy mal.

1.3.Razonamiento acerca de predicados:Las heurísticas no se pueden escribir en logica de predicados,el tiempo tampoco. Las bases de conocimiento en conocimiento experto estan ordenadas por la frecuencia. Se relacionan el metaconocimiento y la heurística, tratandose como iguales los predicados.

Ejemplo:

Para todo conjunto x, X€{Blanco}-> X€{Bonito}. Por esta inferencia debemos clasificar nuestros predicados.

1.4.Información Incompleta (Faltan datos (ES VERDAD O ES FALSA)) y o imprecisa (vaguedad o imprecisión, probabilidad o desconocimiento a priori (Aplicando el principio de Laplace: Da la misma probabilidad a todo)). Por ejemplo hay diferentes tipos de azul por lo tanto ni es cierto ni falso.

1.5.Excepción: Relacionado con la herencia y la jerarquía.

1.6.Monotonía: Si he deducido algo bajo excepciones.

2.PROBLEMAS COMPUTACIONALES:

2.1.CONSISTENCIA: Solidez (que tenga la garantía de la verdad absoluta). Lo que se demuestra como verdadera lo es realmente. Se suele demostrar por refutación por resolución.

2.2:Completitud: (Solo se da en la refutación por resolución), si una cosa es verdadera puede demostrarse. Semidecidible.
2.3:Complejidad computacional: (tratabilidad) .No tienen un algoritmo en tiempo polinomial. Es de tipo NP (SAT 3).

6- ¿Qué tipo de conocimiento organizan las redes semánticas?

Una red semántica o esquema de representación de red, es una forma de representación de conocimiento lingüístico en la que los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles. Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.

Una red semántica consta de un número de nodos que representan objetos e información descriptiva acerca de ese objeto. Los objetos pueden ser elementos físicos tales como un libro, un carro, eventos o acciones. Un concepto puede ser la ley de Ohm, un evento puede ser una fiesta, una acción puede ser fabricar una casa o escribir un libro.

Los nodos en las redes semánticas están interconectados por enlaces (arcos) que describen las relaciones entre los varios objetos. Los arcos pueden definirse de varias formas, dependiendo de la clase de conocimiento representado.

Se define una red semántica como un grafo dirigido, cuyos nodos representan individuos; un arco está etiquetado con el nombre de la relación que éste representa; varios arcos pueden tener la misma etiqueta, mientras que cada individuo es representado por un solo nodo.

Algunos de los arcos más comunes usados para representar jerarquía son del tipo “es-un” o “tiene-un”. El “es-un” muestra una relación de clase, esto es, que un objeto pertenece a una clase grande o categorías de objetos. El enlace “tiene-un” es utilizado para identificar características o atributos de los nodos objetos. Otros arcos son utilizados para propósitos de definición. En general, un enlace designa cualquier relación utilizada para interconectar objetos.

Las redes semánticas se utilizan tanto como modelos de organización de la memoria humana así como de esquemas de representación de significados de oraciones en lenguaje natural.

Esta representación es ventajosa, pues no hay que hacer mecanismos inferenciales para hallar la relación entre nodos. Simplemente lo que hay que buscar es un camino entre esos dos nodos mediante un algoritmo para el procesamiento de grafos (algoritmos de recorrido).

7- Describir en líneas generales el concepto de “frame”

Un marco (frame) es una estructura de datos compleja que contiene un agregado de información usado para representar conceptos, clases de objetos o instancias individuales con características y experiencias bien conocidas. El marco describe ese objeto en gran detalle.

El detalle está dado en forma de **ranuras** las cuales describen los varios atributos y características del objeto y situación. La información almacenada en un frame se distribuye en diferentes campos, llamados aspectos o ranuras (slots). Cada aspecto tiene la información sobre un atributo del objeto que se modela, los cuales representan las propiedades del marco.

Se denomina *ejemplificación de un marco* a uno con sus ranuras llenas. Podemos ver entonces al marco como una clase de entidades y a una ejemplificación de él como una entidad particular.

A cada aspecto se le pueden asociar varios tipos de información, llamados facetas del aspecto. Algunas de las propiedades que nos gustaría ser capaces de representar y utilizar en el razonamiento incluyen:

- VALUE: Almacena el valor para el aspecto.
- PROCEDURE: Contiene un procedimiento para calcular el valor que debe ser almacenado en la faceta VALUE.
- DEMONS: Contiene procedimientos que tienen que ser ejecutados cuando cambia el valor almacenado en VALUE.
- DEFAULT: Contiene un valor inicial, por omisión o valores comúnmente usados para la faceta VALUE.
- RESTRICTIONS: Contiene un conjunto de expresiones lógicas que tienen que ser verdaderas para el valor almacenado en VALUE.
- EXPLANATION: Almacena documentación sobre el aspecto.

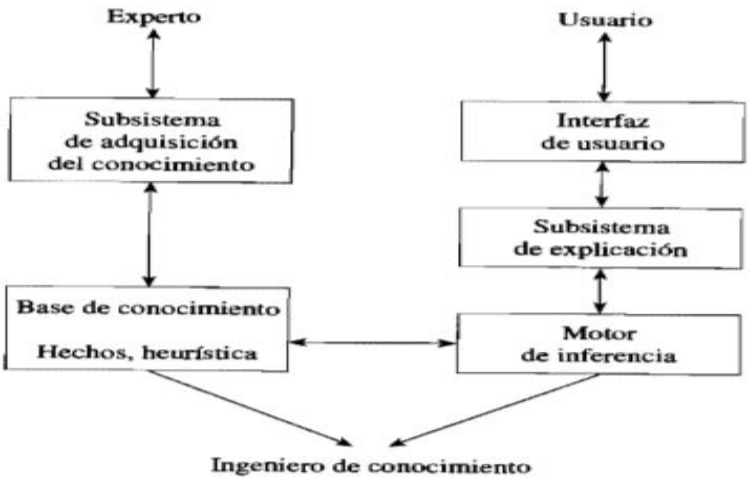
No necesariamente un aspecto incluye todas las facetas.

IMPORTANTE LA JERARQUÍA!!!
Los frames pueden ser organizados como una jerarquía, para lo cual es suficiente incluir en el frame uno o más aspectos que contengan un enlace a un frame superior en la jerarquía, de la cual ella es una instancia, Cuando esta estructura jerárquica es usada, un frame puede heredar las propiedades de otro frame, o sea, compartir información a través de la herencia.

8- Estructura y componentes de un sistema experto

Un sistema experto incorpora conocimiento sobre ámbitos específicos del conocimiento humano, tales como la medicina o los negocios. Los principales componentes del sistema son la base de conocimiento, el motor de inferencia y la base de hechos.

La base de conocimiento está compuesta por hechos y reglas (obtenidas de los expertos) del cálculo de predicados acerca del ámbito de la aplicación (aunque también podría componerse de conocimiento representado por alguna variante sintáctica del cálculo de predicados).



El motor de inferencia se compone de todos los procesos que manipulan la base de conocimiento para deducir la información perdida por el usuario; por ejemplo, resolución, encadenamiento hacia atrás o hacia delante (en algunos casos, la base de conocimiento y los mecanismos de inferencia pueden estar estrechamente relacionados).

La Base de Hechos, también conocida como memoria de trabajo o base de datos global, es la que contiene los datos de partida y los criterios de parada, la misma se va actualizando durante la ejecución del sistema.

Además de los componentes anteriores los sistemas expertos necesitan interactuar con el usuario y con el experto.

Interfaz con el usuario, es la que facilita el diálogo con el usuario, permite hacerle preguntas al sistema e incluso obtener conocimientos análogos a los del experto. Las explicaciones pueden ser obtenidas de la base de hechos donde se almacenan los pasos para llegar a la solución.

Interfaz con el experto, permite al experto consultar los conocimientos almacenados en la base de conocimientos y da la posibilidad de incluir nuevos conocimientos. Su objetivo es que el experto pueda introducir directamente sus conocimientos en la máquina sin necesidad de ver al ingeniero que desarrolló el sistema.

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

El **factor de ramificación** es uno de los tres términos que definen la complejidad. Expresa el máximo número de sucesores de cualquier nodo.

La complejidad de un juego determina, junto con la completitud y la optimización, una de las formas en las que se evalúa el rendimiento del algoritmo de un juego. Puede ser complejidad en tiempo: cuánto se tarda en encontrar una solución; o en espacio: cuánta memoria se necesita para el funcionamiento de la búsqueda.

COMPLEJIDAD DEL ALGORITMO MINIMAX Y LA PODA ALFA-BETA:

El algoritmo minimax realiza una exploración primero en profundidad completa del árbol de juegos. Si la profundidad máxima del árbol es *m,* y hay *b* movimientos legales en cada punto, entonces la complejidad en tiempo del algoritmo minimax es *O(b^m).* La complejidad en espacio es *O(bm)* para un algoritmo que genere todos los sucesores a la vez, o *O(m)* para un algoritmo que genere los sucesores uno por uno. Para juegos reales, desde luego, los costos de tiempo son totalmente poco prácticos, pero este algoritmo sirve como base para el análisis matemático de juegos y para algoritmos más prácticos.

La búsqueda alfa-beta actualiza el valor de α y β según se va recorriendo el árbol y poda las ramas restantes en un nodo (es decir, termina la llamada recurrente) tan pronto como el valor del nodo actual es peor que el actual valor α y β para max o min, respectivamente. Esto sugiere que pudiera valer la pena tratar de examinar primero los sucesores que probablemente serán los mejores.

Si asumimos que esto puede hacerse, resulta que alfa-beta tiene que examinar sólo *O(b^(m/2))* nodos para escoger el mejor movimiento, en vez de *O(b^m)* para minimax. Esto significa que el factor de ramificación eficaz se hace √b en vez de *b*. De otra manera, alfa-beta puede mirar hacia delante aproximadamente dos veces más que minimax en la misma cantidad del tiempo. Si los sucesores se examinan en orden aleatorio más que primero el mejor, el número total de nodos examinados será aproximadamente *O(b^(3m/4))* para un moderado b.

10- ¿Se puede resolver de forma exacta un juego bipersonal con información perfecta? ¿De qué forma?

Si, se puede. Con un procedimiento Minimax o un algoritmo poda Alfa-Beta se puede resolver un juego bipersonal con información perfecta, siempre y cuando se tenga suficiente tiempo como para explorar los suficientes nodos del árbol de juego para hallar la "victoria" para el jugador que tiene el turno.

John Von Neumann probó el teorema fundamental del minimax, por el que se demuestra que para juegos de suma cero con información perfecta entre dos competidores existe una única solución óptima.

"Siempre existe una forma racional de actuar en juegos de dos participantes, si los intereses que los gobiernan son completamente opuestos."

11- ¿Por qué podemos decir que la poda alfa-beta es un algoritmo mejor que el procedimiento minimax para la resolución de juegos?

Porque permite descartar caminos del árbol de juego, ahorrando tiempo para poder explorar otros caminos más prometedores, mientras que el procedimiento minimax se recorre todo el árbol de juego directamente, incluso si hay un camino en el árbol que no se vaya a desarrollar porque sea muy prometedor para alguno de los jugadores. Se tiene en cuenta que ambos jugadores van a escoger la mejor opción posible.