

PREGUNTAS IA -> DESARROLLO

1. ¿En qué consiste la propiedad de ser pro-activo de un agente inteligente?

R1: Consiste en que no se debe simplemente actuar en respuesta a su entorno, es decir, establece sus propias metas y determina si puede llegar a conseguirlo.

R2: El agente debe ser capaz de cumplir sus propias metas y ser capaz de controlar sus propios objetivos a pesar de cambios en el entorno.

2. ¿Requiere percibir un agente deliberativo mientras construye un plan? ¿Y cuando lo ejecuta?

Si, aunque ya tiene una representación interna del mundo, percibe y manipula la estructura interna para usarla como razonamiento. El mundo no es estático. Lo ejecuta cuando elige la acción deseada tras el razonamiento.

3. ¿Qué característica esencial aporta el algoritmo de enfriamiento simulado frente al resto de métodos de escalada? ¿Qué le aporta? ¿Cuál es la principal dificultad de aplicar dicho algoritmo en un caso concreto?

La ventaja que nos da el algoritmo de enfriamiento simulado es que nos permite escapar de los máximos locales y permite encontrar máximos globales de forma más efectiva que un algoritmo no probabilístico. Además es eficiente y fácil de implementar.

Los problemas que nos podemos encontrar se generan a la hora de seleccionar la temperatura inicial o el método de actualización de la temperatura y, además, una vez generado el enfriamiento simulado las primeras iteraciones permiten generar estados aleatorios pero que a medida que repite iteraciones da paso a un algoritmo de escala tradicional

4. ¿Son la búsqueda primero en anchura, búsqueda primero en profundidad, y la búsqueda de coste uniforme casos especiales de la búsqueda primero el mejor? Razona la respuesta

Si, ya que la búsqueda primero en anchura es un caso especial de la búsqueda de coste uniforme cuando los costos de las aristas son positivos e idénticos. A su vez, la búsqueda de coste uniforme es una variante del algoritmo búsqueda primero el mejor.

Búsqueda en anchura: se toma el nodo más superficial de la lista de abiertos y lo introduce en la lista de cerrados, expandiéndolo y cada hijo que no esté ni en abierto ni en cerrado se introduce en la cola de abiertos, se repite el procedimiento hasta llegar al objetivo buscado. Encuentra la solución si existen, es óptimo si todos los operadores tienen el mismo coste, es eficiente si el objetivo está cerca pero consume mucha memoria.

Búsqueda en profundidad: igual que la búsqueda en anchura pero se usaría una pila (se explora el último en entrar)

Búsqueda costo uniforme: la diferencia entre este y el de anchura es que la cola de se ordena en función del costo del camino al nodo, escogiendo el de menor costo.

5. El algoritmo A* utiliza una lista de ABIERTOS y una lista de CERRADOS. Describe el propósito de cada una de esas listas.

La gestión de nodos para el algoritmo A* es la siguiente:

1. Primero un nodo se introduce en abiertos únicamente si se comprueba que no estaba previamente en cerrados ni abiertos, por lo que si hay un nuevo nodo que ya existe en abiertos es necesario comprobar si el nuevo camino es mejor que el nodo anterior.
2. Si un nodo estaba ya en abiertos quiere decir que se ha llegado al mismo nodo por un camino distinto por lo que se comprueba el coste del camino hasta el padre del primero y del segundo (Se quedará como padre del nodo el que tenga el camino con menor coste)
3. Si un nodo estaba ya en cerrados tenemos que comprobar que el camino del nodo desde el actual al inicial de ese nodo es menor que el del anterior, en caso de que sea mejor también es necesario comprobar recursivamente todos los descendientes de dicho nodo para ver si el nuevo camino mejora o empeora el coste

6. ¿Qué problemas plantea el uso del cálculo de predicados para resolver problemas de inteligencia artificial?

El problema que tiene el cálculo de predicados es que es semidecidible y además en los casos en que la refutación mediante resolución termina, el procedimiento es NP-duro

7. Describe en qué consiste el aprendizaje inductivo

El aprendizaje inductivo se basa en el planteamiento de diferentes hipótesis, donde las hipótesis planteadas estarán bien generalizada si se puede predecir con ellas ejemplos que no se conoce, en cuanto tenemos varias hipótesis consistentes, es decir, que satisfacen todos los datos, tenemos que aplicar la navaja de Ockham, donde elegiremos la hipótesis más simple y consistente con los datos

8. Concepto de agente. Concepto de agente inteligente, describe todas sus propiedades. Interés y uso de los agentes inteligentes en inteligencia artificial.

Un agente va a ser una abstracción de un ente que tiene que resolver. Vamos a intentar resolver las componentes que rodean a este ente.

Un agente inteligente es un sistema de ordenador, situado en algún entorno y capaz de realizar acciones de forma autónoma, además de ser flexible para lograr los objetivos. Propiedades:

Situación -> EL agente está situado porque recibe entradas sensoriales

Autonomía -> El sistema es capaz de actuar sin la intervención directa de los humanos y tiene el control sobre sus propias acciones y estado interno

Flexibilidad ->

Es autónomo, ya que es capaz de tomar sus propias decisiones sin la intervención directa del humano. Capacidad rápida y simple.

Es social ya que es capaz de interactuar con otros agentes

Es reactivo ya que en función de lo que esté percibiendo tiene que responder a lo que ocurre sin la intervención humana.

Pro-activo : no debe simplemente actuar en respuesta a su entorno, establece sus propias metas y determina si puede llegar a conseguirlo.

Para la IA es de gran importancia e interés el diseño de sistemas multiagente, ya que es un sistema diseñado e implementado con varios agentes interactuando. Son interesantes para representar problemas que tienen múltiples formas de ser resueltos, múltiples perspectivas y/o múltiples entidades para resolver el problema

9. ¿Qué es la búsqueda no informada y en qué se diferencia de la búsqueda informada (o heurística)? Detalla en el algoritmo general de búsqueda en qué punto concreto se diferencia los dos tipos de búsqueda.

La búsqueda informada utiliza el conocimiento para encontrar la solución, mientras que la no informada no lo utiliza, en términos generales no se proporciona información sobre la solución, además la eficiencia de la búsqueda informada es mayor que la de la no informada, debido a que consume menos tiempo y costo ya que tiene una visión general de la solución al contrario que la no informada. En conclusión, la búsqueda informada proporciona la dirección con respecto a la solución mientras que en la búsqueda no informada no se da ninguna sugerencia con respecto a ella, esto hace que la búsqueda no informada sea más larga cuando se implementa el algoritmo.

El punto donde se diferencian los dos tipos de búsqueda tiene relación con el uso de la función heurística ya que reciben como input el estado actual del problema y generan un valor numérico en base a su proximidad al estado meta, de esta forma nuestro objetivo será minimizar dicha función.

10. ¿Por qué los métodos de escalada utilizan criterios aleatorios en algunas decisiones? Describe el uso de algún algoritmo concreto.

Debido a que estos métodos se utilizan en problemas en los que el camino hacia la solución no es relevante, por lo que en este espacio de estados cada nodo posee información heurística por lo que es fácil compararlos y buscar un máximo local viajando por los nodos hasta encontrar un máximo local o idealmente global.

Ejemplo -> Algoritmo de escalada simple

El algoritmo de escalada simple nos dice que mientras el estado actual E , no sea el objetivo se selecciona un operador A y se aplica sobre el estado E , para después compararse con la función heurística de manera que si $f(A(E)) > f(E)$ entonces pasa a ser $A(E)$ y se continúa con la búsqueda

11. ¿Qué caracteriza a los algoritmos genéticos como métodos de escalada?

El uso de criterios aleatorios, ya que el proceso consiste en generar y evaluar un conjunto de soluciones iniciales seleccionando los individuos que se cruzarán, mutación posible de los hijos o padres, esto se da hasta que algún individuo de la población cumple con las características deseadas en la solución del problema.

12. ¿Hay siempre garantía de que el algoritmo A^* encuentre la solución si ésta existe? Detalla la respuesta.

El algoritmo A^* se encuentra entre los algoritmos denominados como búsqueda primero el mejor, de manera que podemos garantizar que nuestro algoritmo A^* siempre encuentra una solución si esta existe, además garantizando que el camino para obtener la solución nombrada anteriormente será el de menor costo posible, ya que además, con ayuda de una heurística admisible (no sobreestima el valor de los nodos) podremos alcanzar este objetivo con el menor costo posible.

13. ¿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 el número de descendientes por nodo que tendrá un juego, de manera que cuanto mayor sea este más complejo y tardía será la resolución del juego.

Tanto el algoritmo minimax como la poda alfa-beta son algoritmos que nos permiten buscar una solución para un juego, son algoritmos de exploración de árboles de juegos.

En el algoritmo minimax el valor $V(J)$ de un nodo J de la frontera de búsqueda es igual al de su evaluación estática, de manera que si J es un nodo MAX entonces su valor es igual al máximo de los valores de sus nodos sucesores, y si J es un nodo MIN su valor $V(J)$ es igual al mínimo de los valores de sus nodos sucesores.

En alfa-beta ocurre de manera parecida pero no igual, ya que este algoritmo consta de dos cotas donde cada nodo tendrá dos variables asociadas: ALFA representa el mejor valor encontrado hasta el momento por los nodos MAX y es una cota inferior (solo puede crecer) mientras que beta el mejor valor encontrado por los nodos MIN y es una cota superior (sólo puede descender)

14. ¿Qué tipo de conocimiento organizan las redes semánticas?

Forma de representación de conocimiento lingüístico en la que los conceptos y sus interrelaciones se representan mediante un grafo.

15. Propiedades de un agente para ser llamado inteligente

Reactivo: tiene que percibir el entorno y responder a sus cambios.

Pro-activo: además de ser reactivo, debe de cumplir sus propios objetivos.

Social: interactuar con otros agentes o humanos.

16. Tipos de agentes

Agente reactivo: Selecciona las acciones sobre la base de las percepciones actuales, es simple y tiene una arquitectura de programación más simple que un agente deliberativo. Percibe y actúa.

Percibe y actúa sin ninguna estructura de mundo y tiene módulos de comportamiento predefinidos.

Agente deliberativo: Contiene un modelo simbólico de mundo, las decisiones se toman utilizando mecanismos de razonamiento lógico. Tienen una estructura interna que consta de 3 fases: percepción, planificación y actuación, esta estructura tiene fallos ya que el entorno no es estático, tiene que detectar cambios e ir actualizando.

17. Definición de arquitectura de subsunción

Consiste en agrupar en módulos de comportamiento, cada módulo tiene una acción asociada y recibe la percepción directamente y comprueba la condición de manera que:

Si se cumple la condición -> el módulo devuelve la acción

Se separan en comportamientos por prioridad y un módulo puede subunirse a otro

18. ¿Cuáles son las estrategias de control?

Irrevocables: Donde solo mantenemos en memoria un único nodo, que incluye la descripción completa del sistema en ese momento.

Tentativas: ->

- Retroactiva (backtracking): Se mantiene el camino desde el estado inicial hasta el actual
- Búsqueda en grafos: se guardan todos los estados en memoria

19. ¿Qué diferencia hay entre una arquitectura de control retroactiva y una búsqueda de profundidad sobre grafos?

Retroactiva: solo se guarda un hijo de cada estado, con lo que el grafo es realmente una lista.

Búsqueda en profundidad sobre grafos: se guardan todos los estados o nodos generados hasta el momento.

20. ¿Por qué los métodos de escalada utilizan criterios aleatorios en algunas decisiones?

Cómo se selecciona aleatoriamente un punto inicial, elimina el riesgo de que se almacenen óptimos locales pero nunca el óptimo global.

21. ¿Qué aporta el uso de la aleatoriedad en los métodos de escalada?

Evita mínimos locales. Gracias a la aleatoriedad se puede conseguir encontrar el mínimo global realizando diferentes iteraciones de la búsqueda.

22. ¿Qué caracteriza a los algoritmos genéticos como métodos de escalada?

Son algoritmos de optimización basados en el proceso de la evolución de Darwin, hay una población, los más adecuados se reproducen y tienen descendencia (a veces con mutaciones).

No necesitan partir de un nodo o estado inicial, ya que hay toda una población y su objetivo es encontrar una solución cuyo valor definición objetivo sea óptimo

23. La mayor parte de los programas para juegos no almacenan los resultados de las búsquedas de un movimiento para el siguiente movimiento, ya que normalmente empiezan de cero cada vez que es el turno del ordenador.

¿Por qué?

Porque la situación sobre la cual desarrollar el árbol de posibilidades suele ser completamente distinto, ya que en el turno del adversario la situación del juego puede haber cambiado totalmente y ya no ser válido todo lo calculado anteriormente.

24. ¿Por qué la estructura básica para representar un juego es un árbol y no un grafo?

Un árbol de juego es una representación explícita de todas las formas de jugar a un juego.

En un grafo se permitiría volver a un estado anterior del juego, lo cual en la mayoría de los juegos no es posible y no dejaría evaluar rápidamente y eficientemente las posibles situaciones del juego.

25. Relación entre árbol de juego y árbol Y/O

En los árboles Y/O se indica el orden de consecución de tareas a realizar para alcanzar un objetivo. Esto es parecido al orden de movimientos que se deben de dar en un juego para llegar a una situación determinada.

26. ¿Por qué en el algoritmo MINIMAX es conveniente utilizar estrategias de búsqueda retroactivas para la búsqueda parcial?

Para ahorrar espacio en memoria.

27. Definición de calculo proposicional

Es una representación de tipo general y decidible. El problema viene cuando se quiere razonar sobre conjuntos de cosas, por ejemplo grafos o jerarquías de conceptos.

28. Definición de cálculo de predicados.

Es una representación de tipo general más rica que la proposicional. Sus características son:

- Sólido -> Asegura que una conclusión inferida es cierta
- Completo -> Asegura que una inferencia producirá una conclusión verdadera
- Decidible -> asegura que la inferencia es posible (semidecidible)
- Problema -> es semidecidible y cuando la refutación mediante resolución termina, el procedimiento es NP-duro.

29. ¿Por qué el cálculo proposicional es limitado para lo que requiere la inteligencia artificial frente al cálculo de predicados?

Porque el cálculo proposicional nos permite razonar sobre conjuntos de cosas (grafos o jerarquías de conceptos), mientras que el cálculo de predicados nos permite una representación más rica.

30. Definición de aprendizaje automático

Modifica el mecanismo de decisión del agente para mejorar su comportamiento, son programas que mejoran su comportamiento con la experiencia

31. Definición de aprendizaje no supervisado

Aprende a partir de patrones de entrada para los que no se especifican los valores de sus salidas

32. Definición de aprendizaje supervisado

Aprender una función a partir de ejemplos de sus entradas y salidas.

Métodos basados en modelos -> representan el conocimiento aprendido en algún lenguaje de representación (modelo o hipótesis).

Métodos basados en instancias -> representan el conocimiento aprendido como un conjunto de prototipos descritos en el mismo lenguaje usado para representar la evidencia

33. Definición de aprendizaje por refuerzo

Es un tipo de aprendizaje automático en el que el algoritmo de aprendizaje recibe algún tipo de valoración acerca de la idoneidad de la respuesta.

Los objetivos se expresan mediante una función de recompensa que asigna valores enteros a las acciones que ejecuta el agente

34. Definición de aprendizaje inductivo (por ejemplos)

Se basa en proporcionar al agente una serie de ejemplos y que vaya aprendiendo a través de ellos. Sean X los ejemplos y $f(x)$ la clase, buscaremos una hipótesis que identifique a esta función sobre un conjunto de ejemplos en el que estoy trabajando.

Valores perdidos -> atributos de nuestros ejemplos de los cuales no poseemos información.

35. ¿Qué relación hay entre el aprendizaje inductivo y el aprendizaje supervisado?

Ambos aprenden a partir de ejemplos, pero además en el aprendizaje supervisado conocemos el resultado de esos ejemplos (están etiquetados / clasificados).

36. Explica cómo se aplica la navaja de ockham en aprendizaje en general y que el aprendizaje de árboles de decisión en particular.

En el aprendizaje en general: entre las distintas hipótesis siempre debo elegir la más simple que sea consistente con los datos.

En el aprendizaje de árboles de decisión: se usa a la hora de clasificar los ejemplos, por ejemplo si tengo 30 ejemplos positivos y 3 negativos, puedo suponer por ockham que eso es ruido o errores.