

# 1.- El concepto de Juego. Elementos y clasificación de los juegos.

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).

Un juego puede definirse formalmente 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.

El estado inicial y los movimientos legales para cada jugador definen el árbol de juegos.

- Numero de jugadores: No es lo mismo un juego bipersonal que un juego n-personal (con  $n \geq 3$ ) donde puede haber pagos colaterales debido a alianzas entre jugadores.

## 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
- Numero 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).

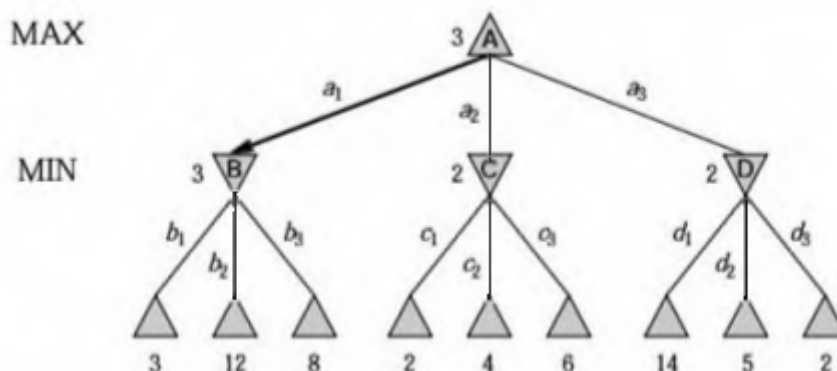
## 2.- El algoritmo Minimax. Componentes y funcionamiento.

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. El funcionamiento de minimax 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 e valor mínimo. Entonces tenemos:

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



El minimax calcula la decisión minimax del estado actual. Usa un cálculo simple recurrente de los valores minimax de cada estado sucesor, directamente implementando las ecuaciones de la definición. La recursión avanza hacia las hojas del árbol, y entonces los valores minimax retroceden por el árbol cuando la recursión se va deshaciendo. 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 como el algoritmo minimax realiza una exploración primero en profundidad completa del árbol de juegos.

### 3.- Modelos de representación de conocimiento.

El conocimiento es la representación simbólica de aspectos de un cierto dominio o universo de discurso. El conocimiento debe ser sorprendente ya que la medida de la sorpresa será la cantidad de entropía que se merma.

Existen 3 niveles de representación del conocimiento:

#### Modelos

Un modelo de representación del conocimiento es un formalismo simbólico empleado para expresar el conocimiento, un modelo mental que se caracteriza por la semántica con que representa ciertos aspectos del dominio o universo con el que un sistema software pueda trabajar. Entre sus características está aquello que puede representar (representación icónica o representación descriptiva), la granularidad de la representación (nivel de detalle), el tipo de descripciones (intensionales o extensionales), la representación del metaconocimiento (conocimiento sobre el propio conocimiento, independiente del mismo) y la jerarquización de objetos y herencia. Así, dentro de los modelos de representación descriptivos encontramos modelos inspirados por la psicología como:

1.- Modelos de conducta: Reglas de Producción o reglas IF\_\_THEN

Si A entonces B (A: condición; B: Acción)

2.- Modelos de razonamiento: Lógica (s).

Lenguaje formal para representar conocimiento y formular inferencia.

3.- Modelos de memoria: Almacenaje y representación.

3.1 Modelos simples: Ficheros y Bases de datos.

3.2 Modelos de conocimiento heredable. Almacenamiento jerarquizado:

Redes Semánticas,

Frames, guiones,

### **Lenguajes de representación**

Un lenguaje formal con símbolos, sintaxis (reglas para contruir cadenas válidas, fbf) y semánticas (relación entre las fbf de la lógica y el conocimiento que se quiere representar). Estos lenguajes se ayudan de las reglas de inferencia permiten obtener un nuevo conocimiento a partir del conocimiento previo.

-Logica proposicional y de primer orden (lógica de predicados) un sistema basado en el conocimiento es aquel que usa masivamente el conocimiento para resolver un problema dentro de un dominio determinado.

### **Lenguajes de programación**

Dependiendo del formalismo de la representación se pueden emplear distintos lenguajes de programación:

- Por ejemplo si usamos la lógica de predicados podemos usar **Prolog**
- Si por el contrario usamos listas para la representación de reglas tipos if-then podríamos usar **Lips**

## **4.- Modelos de Conocimiento heredable. Herencia.**

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.

En las redes asociativas cada nodo representa un concepto (o una proposición) y los enlaces corresponden a relaciones (inclusión, pertenencia, causalidad) o a categorías gramaticales (verbo principal, sujeto, objeto, complementos, etc.). entre ellas, se conocen:

- redes semánticas: son las destinadas a representar o a comprender el lenguaje natural.
- redes de clasificación: es exactamente lo que su nombre indica, una clasificación de objetos o conceptos con sus características propias (herencia y demás).
- redes causales: son las que llevan asociadas, junto a sus nodos que representan variables, una relación de influencia, representada por los enlaces.

Un marco es una estructura de datos para representar una situación estereotipada, como encontrarse en un cierto tipo de sala de estar o asistir a un cumpleaños infantil.

Cada marco se caracteriza por un conjunto de campos o slots que se asocian en general a atributos, y que en conjunto sirven para identificar los marcos.

Los marcos están especialmente concebidos para tareas de reconocimiento: la información recibida hace que se activen unos marcos y esta a su vez provoca la activación de otros marcos conectados con los primeros, dando lugar así a una red de activación, cuyo objetivo es predecir y explicar la información que se va a encontrar en esa situación. Este reconocimiento suele denominarse herencia o más generalmente reconocimiento descendente.

## 5.- Paradigmas de Aprendizaje.

- Aprendizaje memorístico.
  - Almacenamiento de información
- Aprendizaje deductivo.
  - Inferencia y deducción empleando modelos Lógicos
  - Razonamiento Artificial
  - Por sí mismo no puede justificar la creación de nuevo conocimiento sin un conocimiento inicial de partida.
  - El aprendizaje deductivo de reglas aumenta la eficiencia de un sistema mediante la deducción de reglas adicionales a partir de las reglas y los hechos de un dominio, conocidos previamente. Las conclusiones que el sistema puede derivar al utilizar las nuevas reglas también se podrían derivar sin ellas; sin embargo, con estas reglas adicionales, el sistema puede realizar las derivaciones de forma más eficiente.
- 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
  - Entender y resolver una situación por su parecido con otras anteriormente resueltas.
- Aprendizaje inductivo
  - Es el paradigma más ampliamente estudiado dentro del aprendizaje.
  - Hablando en términos muy generales se trata de aprender un concepto o una clasificación a partir de ejemplos y contraejemplos.
  - 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 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.

## 6.- Aprendizaje Inductivo. Empleo de árboles de decisión en Aprendizaje Inductivo.

El aprendizaje puede tomar muchas formas, dependiendo de la naturaleza de las herramientas de desarrollo, el componente a ser mejorado, y de la realimentación disponible. Si la realimentación disponible, tanto de un profesor como del entorno, proporciona el valor correcto para los ejemplos, el problema de aprendizaje se denomina aprendizaje supervisado. La tarea, también llamada

aprendizaje inductivo consiste en aprender una función a partir de ejemplos de sus entradas y salidas. Es decir, encontrar una hipótesis consistente que verifique los ejemplos. La navajade Ockham sugiere elegir la hipótesis consistente más sencilla. La dificultad de esta tarea depende de la representación elegida.

### **Árboles de decisión como herramienta de desarrollo**

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.

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. El valor de la salida puede ser a su vez discreto o continuo; aprender una función de valores discretos se denomina **clasificación**; aprender una función continua se denomina **regresión**.

Un árbol de decisión desarrolla una secuencia de test para poder alcanzar una decisión. Cada nodo interno del árbol corresponde con un test sobre el valor de una de las propiedades, y las ramas que salen del nodo están etiquetadas con los posibles valores de dicha propiedad. Cada nodo hoja del árbol representa el valor que ha de ser devuelto si dicho nodo hoja es alcanzado. La representación en forma de árboles de decisión es muy natural para los humanos; en realidad muchos manuales que explican cómo hacer determinadas tareas (por ejemplo, reparar un coche) están escritos en su totalidad como un único árbol de decisión abarcando cientos de páginas.

### **Expresividad de lo árboles de decisión**

Los árboles de decisión pueden expresar cualquier función a partir de los atributos de entrada. De forma trivial, hay un árbol de decisión consistente para cualquier conjunto de entrenamiento con un camino asociado a cada ejemplo, pero seguramente no será bueno para generalizar nuevos ejemplos. Es preferible encontrar árboles de decisión más compactos.

### **Inducción de árboles de decisión**

Existen múltiples formas de inferir el árbol

– Aunque inferir un árbol puede parecer una tarea muy complicada existe una forma **trivial** para ello. Esta consiste en crear una ruta del árbol por cada instancia de entrenamiento. De esta manera se crea el árbol de manera muy sencilla pero tiene los inconvenientes de crea árboles excesivamente grandes y que no funcionan bien con instancias nuevas.

– La forma **óptima** de inferir el árbol sería encontrando uno (el más pequeño de todos) que fuese compatible con todas las instancias, pero computacionalmente es inviable. Para intentar llegar al mejor árbol posible (**pseudo-óptimo**) se selecciona en cada nivel el atributo que produzca la mejor división posible.

### **Elección de los atributos**

En aprendizaje de árboles de decisión, el esquema que se utiliza para seleccionar atributos está diseñado para minimizar la profundidad del árbol final. La idea es elegir el atributo que proporcione una clasificación lo más exacta posible de los ejemplos. Un atributo perfecto divide los ejemplos en conjuntos que contienen sólo ejemplos positivos o negativos.