1. Búsqueda

1.1. Búsqueda ciega

1.2. Búsqueda informada

- Una heurística h es una función $h : V \to [0, +\infty)$ donde V son los nodos. h estima la distancia a la meta y se normalmente se obtiene por relajación del problema.
  - h se dice monótona $\iff \forall n, n',\ h(n) \leq h(n') + \Gamma(n, n')$ (desigualdad triangular)
  - h se dice admisible $\iff \forall n,\ h(n) \leq h^*(n)$ donde $h^*(n)$ es el coste real óptimo de n a la meta.
    - h monótona $\implies h$ admisible
- A^* (A-estrella): ordenar la lista de abiertos por valor de $f = g + h$ ascendente.
  - A^* sin eliminación de estados repetidos (= búsqueda en árbol) y h admisible es completa y óptima.
  - A^* con eliminación de estados repetidos (= búsqueda en grafo) y h monótona es completa y óptima.

1.3. Búsqueda entre adversarios

1.3.1. Clasificación de problemas de búsqueda (= juegos)

- Suma cero, suma constante o suma variable. *Suma* se refiere a sumar los valores de la utilidad desde el punto de vista de min o de max.
  - Asignar los valores `perder = -1, empatar = 0, ganar = 1` en el ajedrez da un juego de suma cero ya que si uno pierde, el otro gana y por tanto sus valores de utilidad suman 0. Ocurre lo mismo si los dos empatan.
  - Asignar los valores `perder = 0, empatar = 1, ganar = 2` en el ajedrez da una juego de suma constante ya que si una pierde y el otro gana la suma de las utilidades desde ambos puntos de vista es 2. Ocurre lo mismo si los dos empatan ($1 + 1 = 2$).
  - Los juegos de suma variable no son susceptibles de ser atacados por búsqueda entre adversarios.

1.3.2. Minimax

- Con **poda** $\alpha - \beta$:
  - En nodos `min` se actualiza $\beta = \text{mín}(\beta, \alpha_i$ de los hijos)
  - En nodos `max` se actualiza $\alpha = \text{máx}(\alpha, \beta_i$ de los hijos)

2. Lógica de predicados

2.1. Ejercicios

2.1.1. Hoja 2, 2018: ejercicio 2

1. Dos nodos son hermanos si, siendo distintos, tienen el mismo padre.

$$\forall x, y[(\neg I(x, y) \wedge I(padreDe(x), padreDe(y))) \iff H(x, y)]$$

2. Un camino entre dos nodos es una secuencia de uno o varios enlaces entre dichos nodos.

$$\forall x, y, c[C(c, x, y) \iff (I(c, enlace(x, y)) \vee \exists z, m, n(\neg I(m, n) \wedge C(m, x, z) \wedge C(n, z, y)]$$

2.1.2. Parcial 1, 2014-2015: ejercicio 3

1. Ejemplo
2. Se puede diseñar una máquina de Turing para computar la solución de cualquier problema que pueda ser resuelto mediante la aplicación de un algoritmo sobre unos datos de entrada.

3. Una máquina de Turing universal puede simular la acción de cualquier máquina de Turing sobre los datos almacenados en su cinta

$$\forall u[Universal(u) \implies \forall t, d(comp(t, d) = comp(u, descr(t_2, d))]$$