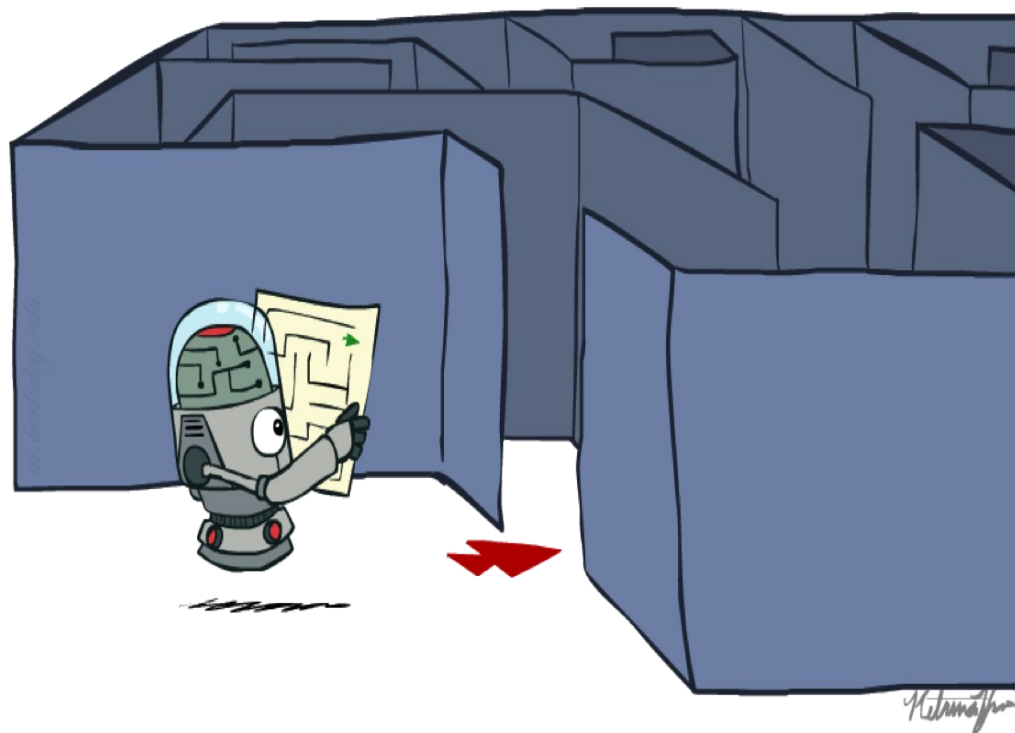


Técnicas de Inteligencia Artificial

Tema 3. Espacio de estados y búsqueda



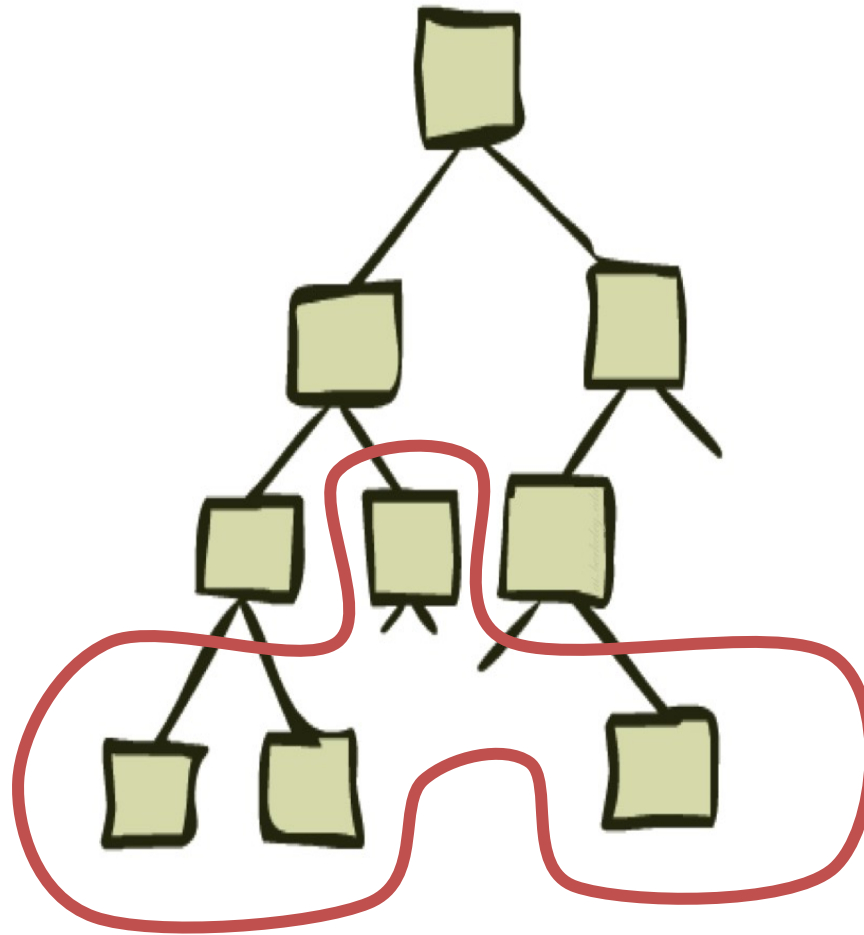
Aitziber Atutxa

[transparencias de Koldo Gojenola y Ekaitz Jauregi adaptadas de Berkeley: Dan Klein, Pieter Abbeel]

Indice

- 3.1 Problemas y espacio de estados
- 3.2 Métodos de búsqueda no informados (Uninformed Search Methods):
 - Búsqueda en profundidad (Depth-First Search)
 - Búsqueda en anchura (Breadth-First Search)
 - Búsqueda de coste uniforme (Uniform-Cost Search)
- 3.3 Métodos de búsqueda informados:
 - ❖ Heurísticos
 - ❖ Búsqueda voraz (Greedy Search)
 - ❖ Búsqueda A* (A* or A star search)
- 3.4 Búsqueda adversarial
 - ❖ Minimax
 - ❖ Poda alfa-beta
 - ❖ Expectimax

Búsqueda en árbol



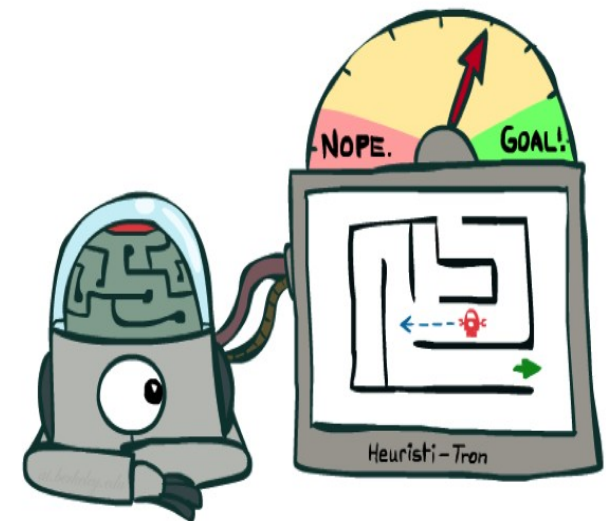
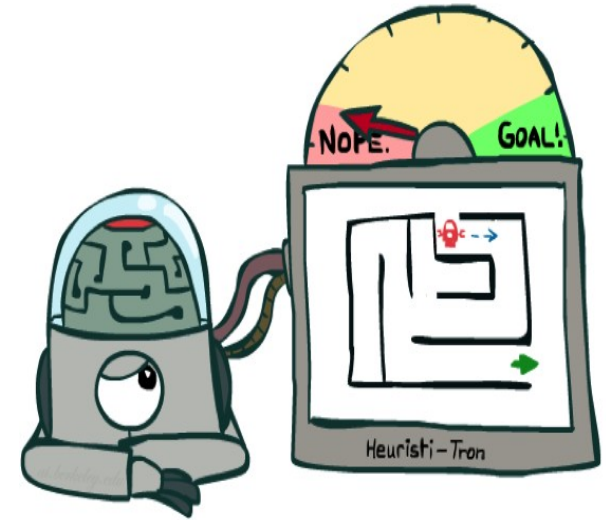
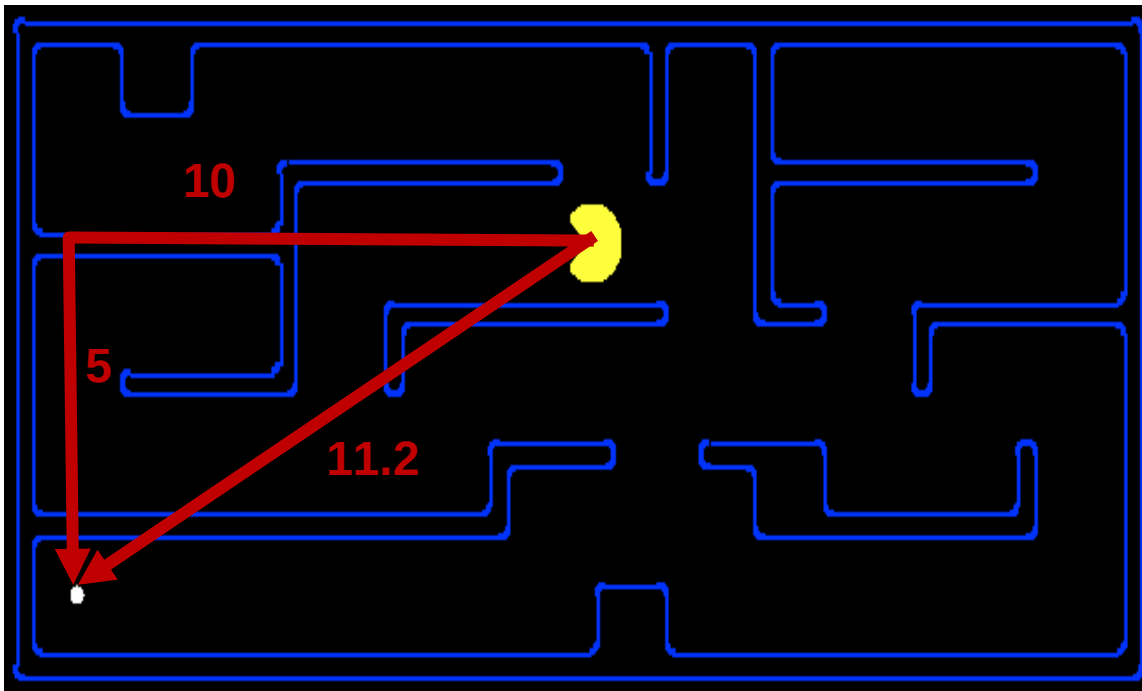
Búsqueda informada

(Informed Search)

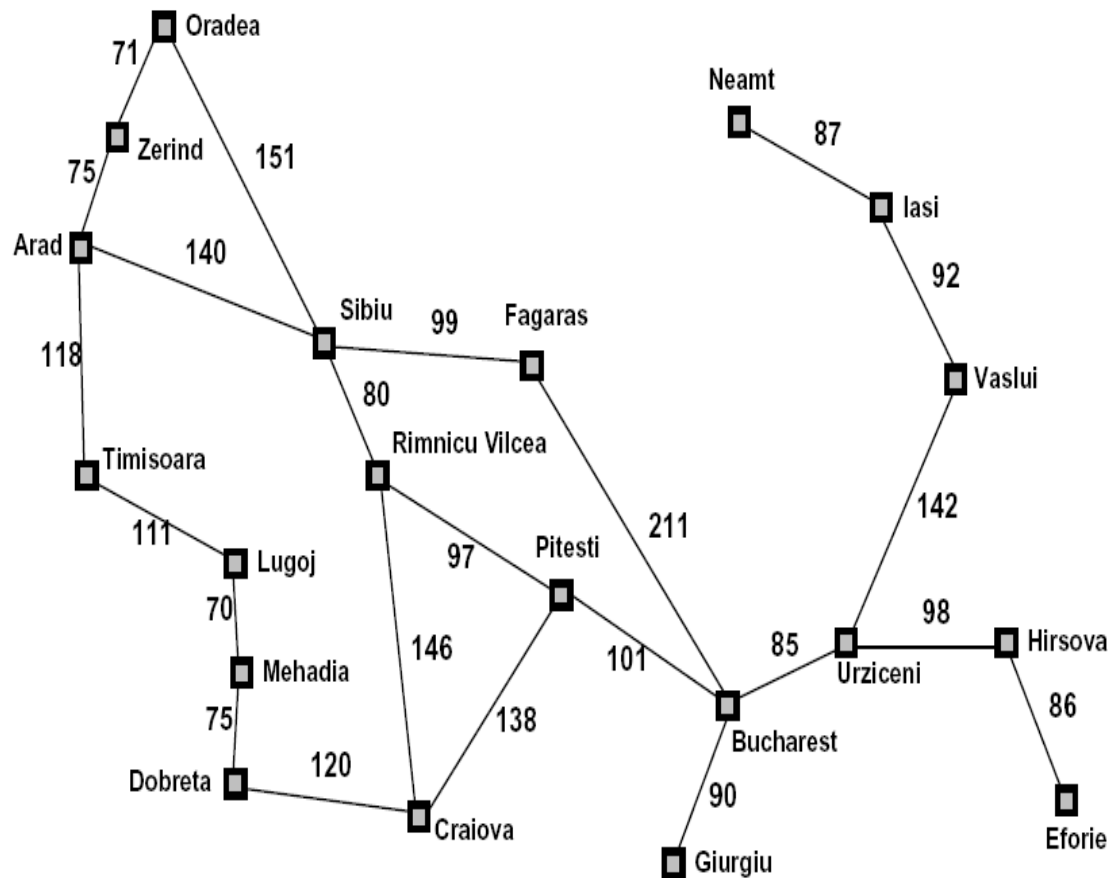


Heurísticos de búsqueda

- Un heurístico es:
 - Una función que estima el coste de alcanzar un estado objetivo
 - Diseñado para un problema de búsqueda particular
 - Ejemplos: distancia Manhattan, distancia Euclídea para buscar un camino



Ejemplo: función Heurística



Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

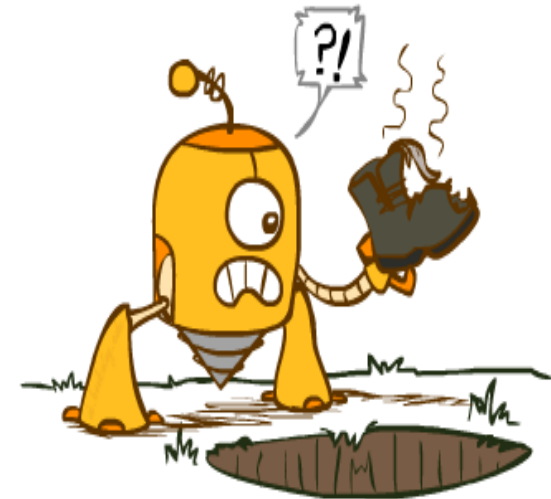
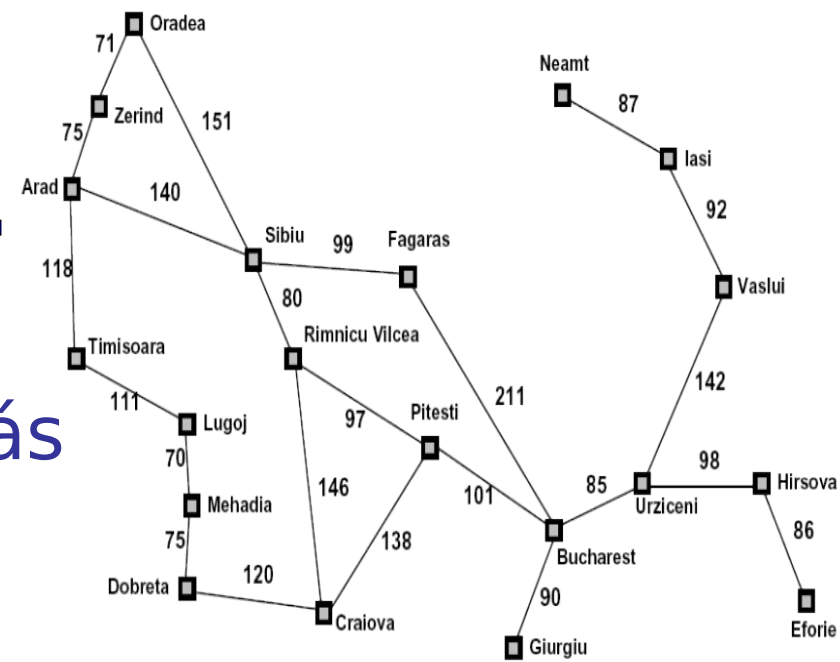
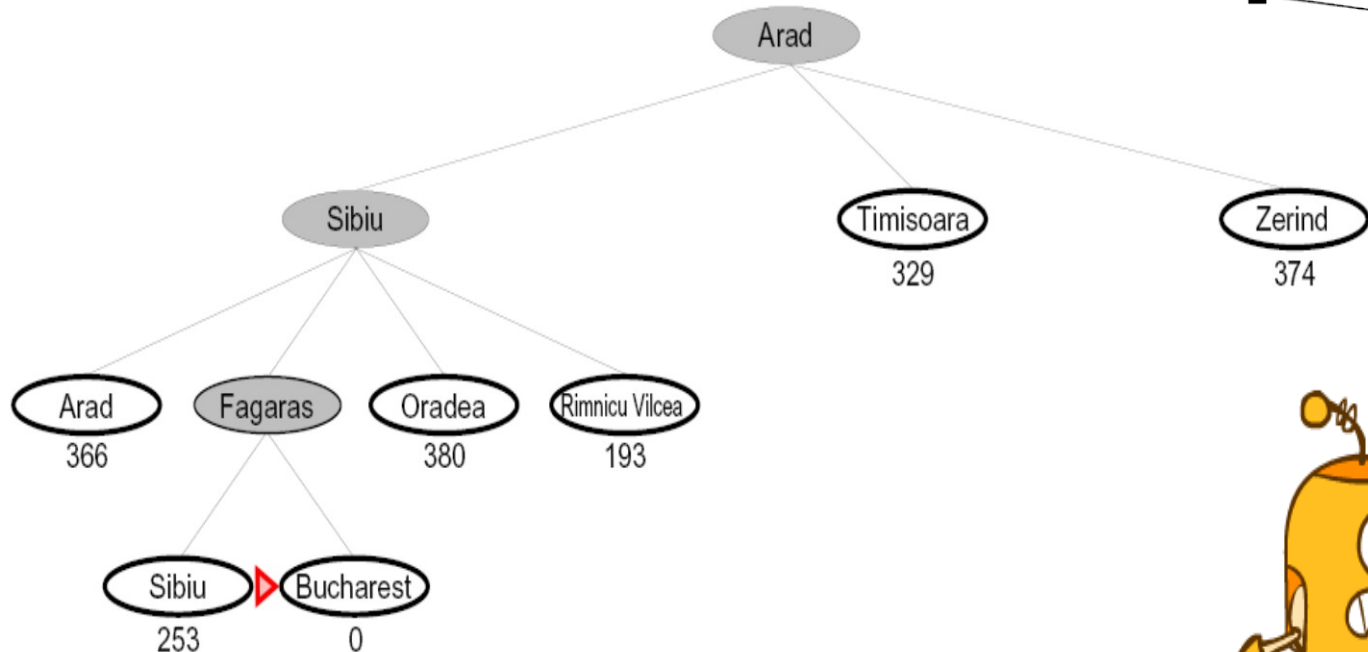
$h(x)$

Búsqueda voraz (Greedy Search)



Greedy Search

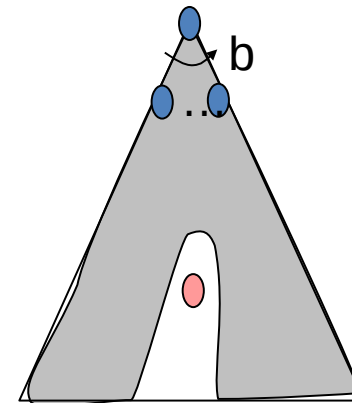
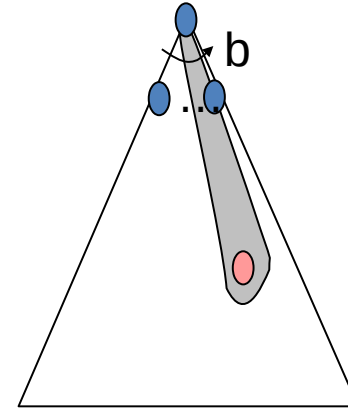
- Expandir el nodo que parece más cercano...



- ¿Qué puede fallar?

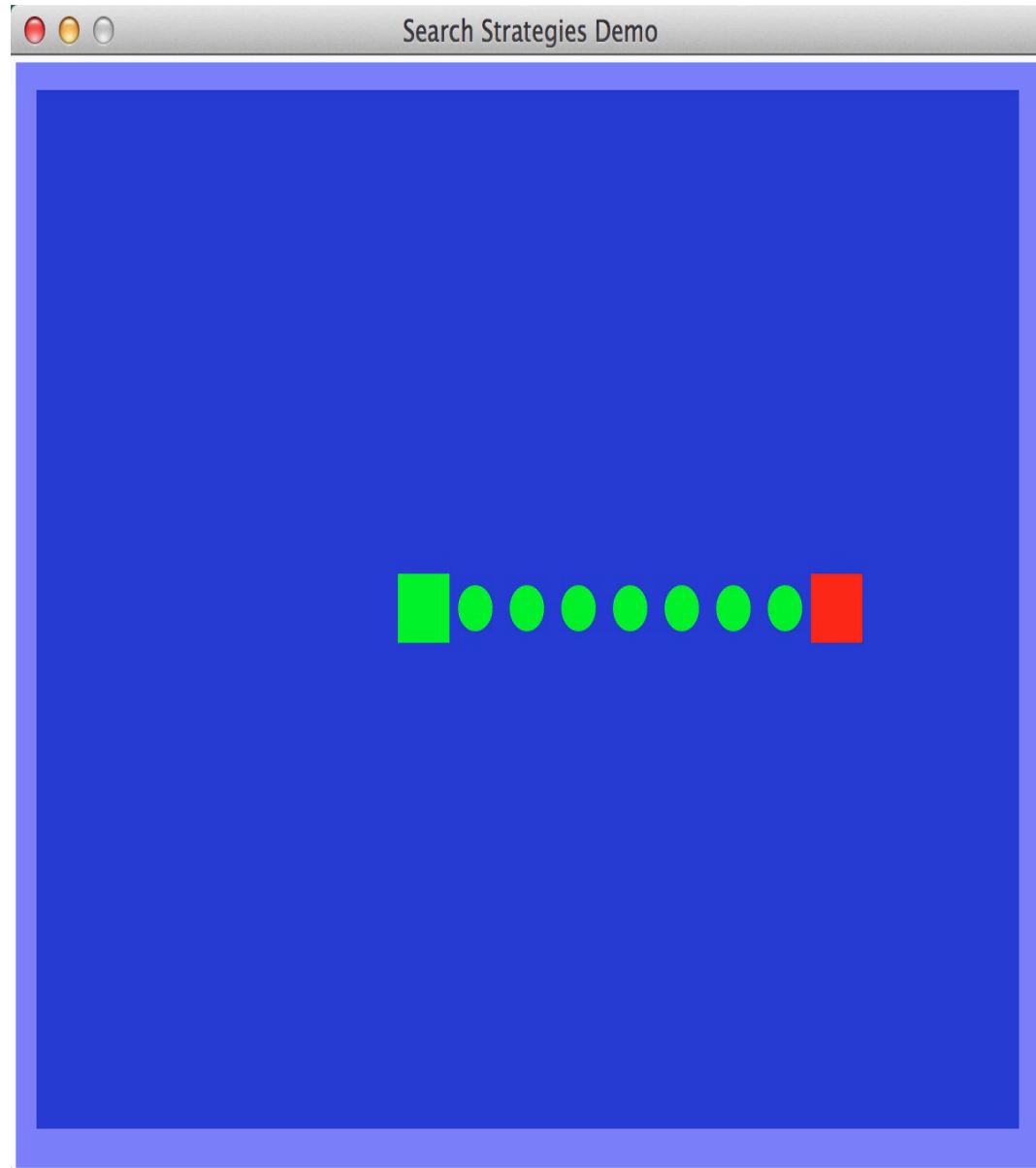
Greedy Search

- Estrategia: expandir el nodo que pensamos que está más cerca de un estado objetivo
 - Heurístico: estimación de la distancia al objetivo más cercano desde cada estado
- Un caso frecuente:
 - Greedy Best-first nos lleva directamente a un objetivo (incorrecto)
- Caso peor: como DFS mal guiado



[Demo: contours greedy empty (L3D1)]
[Demo: contours greedy pacman small
maze (L3D4)]

Video of Demo Contours Greedy (Empty)



Video of Demo Contours Greedy (Pacman Small Maze)

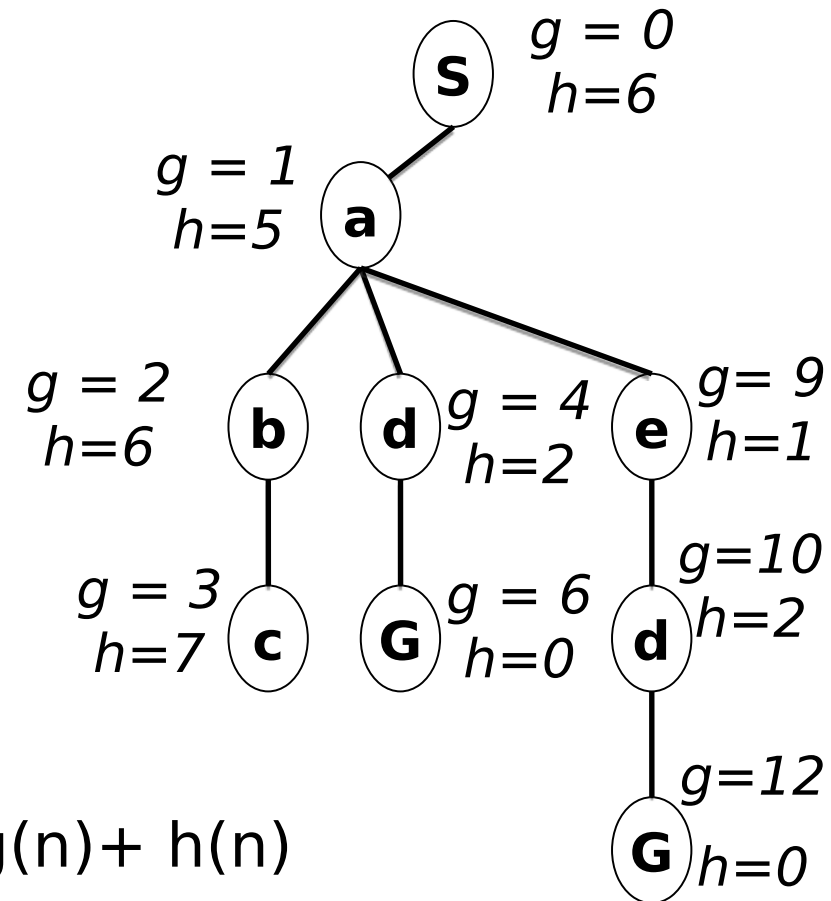
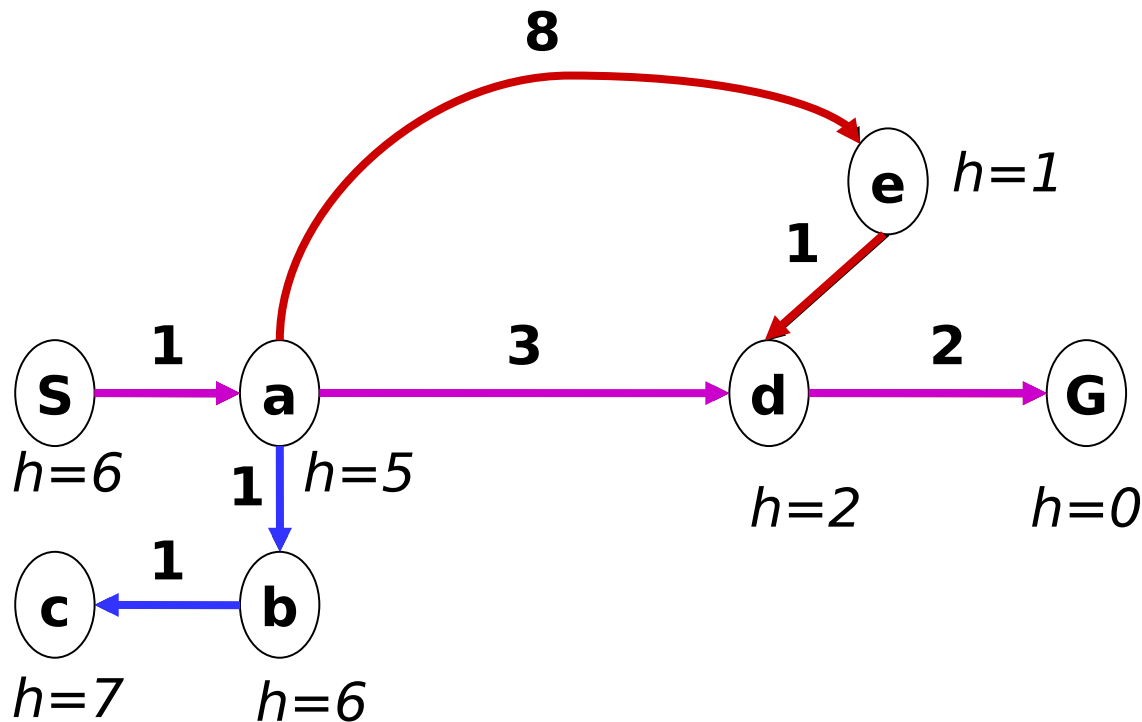


Búsqueda A* (A* Search)



Combinando UCS y Greedy

- Coste uniforme (UCS) ordena por el coste del camino, o backward cost $g(n)$
- Greedy ordena por la cercanía del objetivo, o forward cost $h(n)$

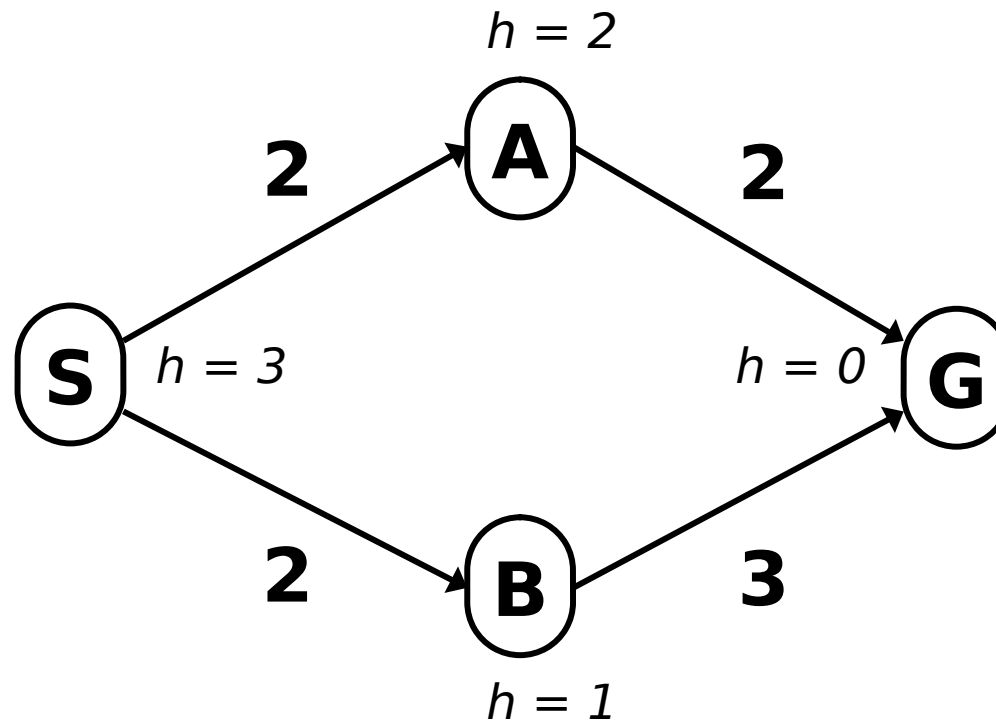


- A* Search ordena por la suma: $f(n) = g(n) + h(n)$

Ejemplo: Teg Grenager

¿Cuándo debería terminar A*?

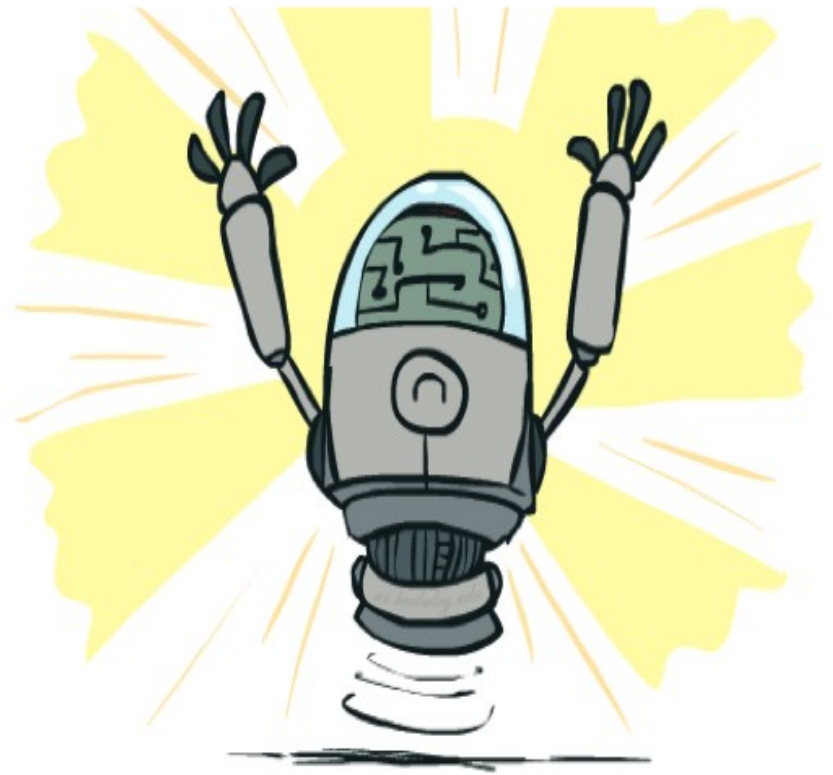
- ¿Cuándo encolamos un objetivo?



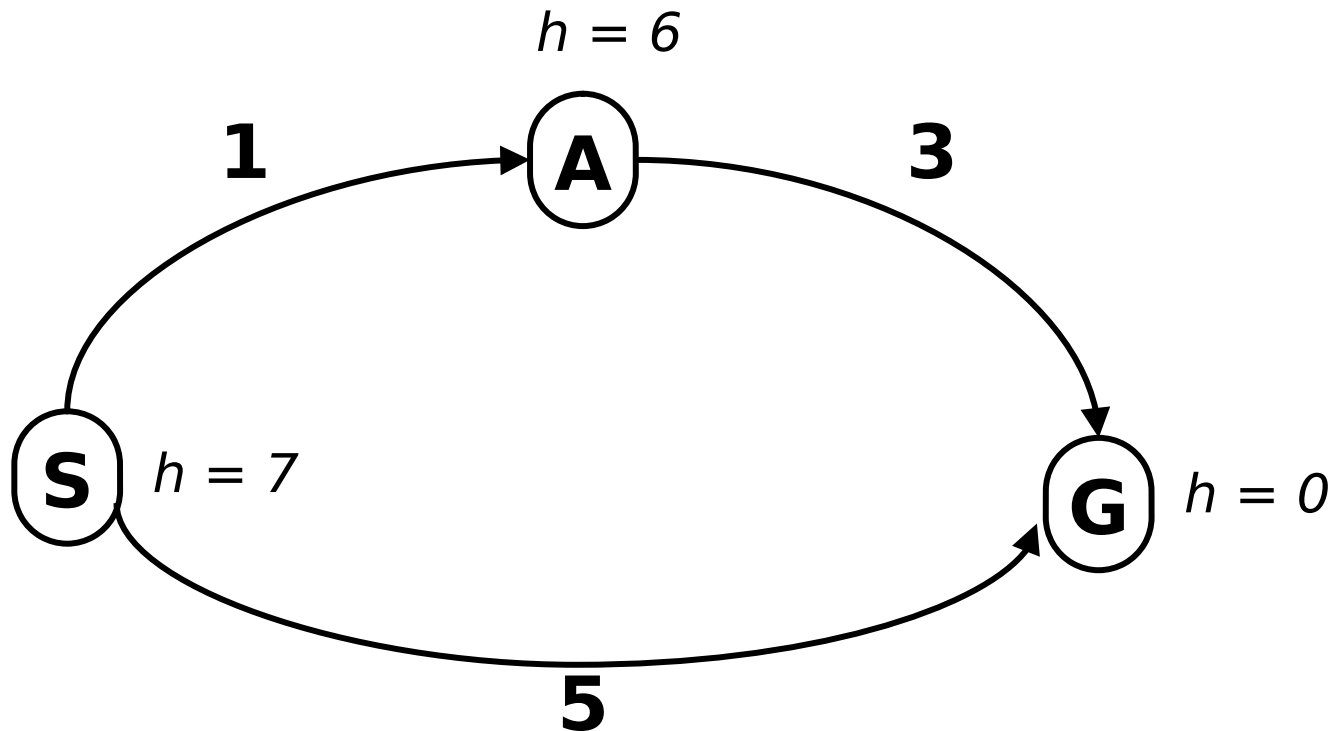
- No: solo se debe parar cuando desencolamos un objetivo

Optimalidad

- Tree search:
 - A* es óptimo si el heurístico es admisible
 - UCS es un caso especial ($h = 0$)
- La consistencia implica admisibilidad pero no al revés, aunque
- En general, la mayoría de los heurísticos admisibles tienden a ser consistentes, especialmente si vienen de problemas relajados

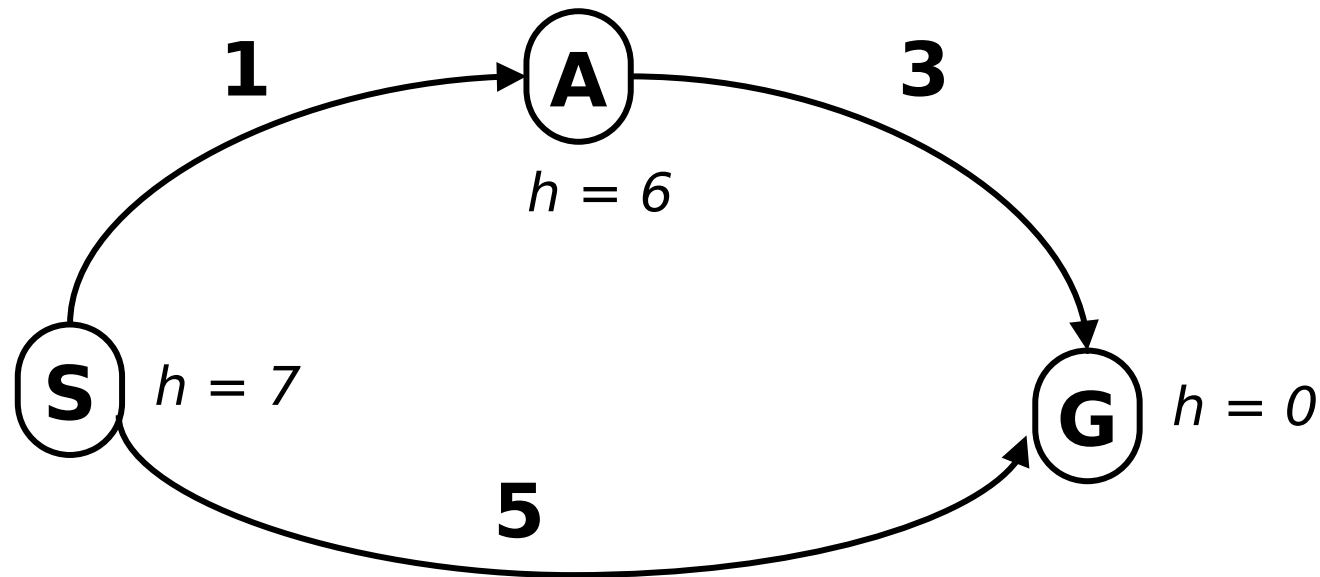


¿Es el heurístico admisible?



- ¿Qué es lo que falla?
- ¡Necesitamos que las estimaciones sean menores que los costes reales!

¿Es el heurístico Admisible?



¿Cuándo se dice que un Heurístico es admisible?

Cuando no sobre-estima el **coste acumulado**

Para todo $s \in S$: $h(s) \leq c(s)$

1) $c(s)$ coste real acumulado desde el nodo **S** hasta la meta **G**

2) $h(s)$ heurístico asociado a un nodo (también se le puede llamar coste estimado del nodo) o medida de estimación de lo bueno que es ese nodo con respecto a la meta (G).

¿Es A* Óptimo?

- El que un Heurístico sea Admisible asegura que nos va a devolver la solución óptima, pero no necesariamente la más eficiente. Por ejemplo, pongamonos en el caso extremo:

Para todo $s \in S$:

$$h(s) = 0 \text{ y } c(s) \geq 0$$

Esto lo hace admisible dado que

$$h(s) \leq c(s)$$

Pero al final convierte al A* en lo mismo que el Uniform Cost

¿La heurística es consistente?

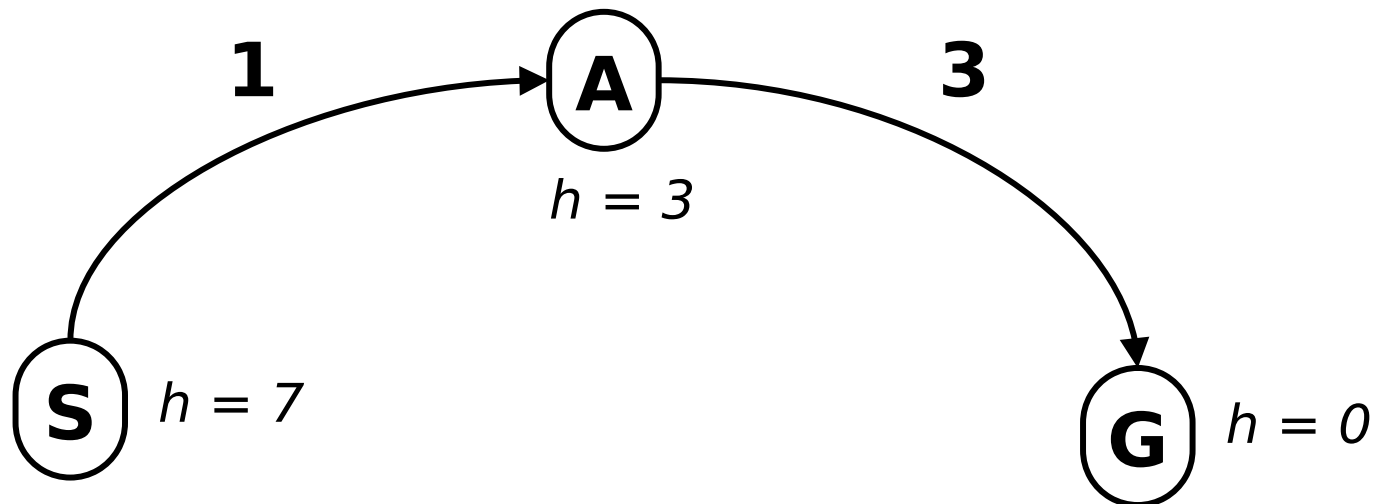
La consistencia es un concepto asociado a cada paso o transición

h es consistente \leftrightarrow

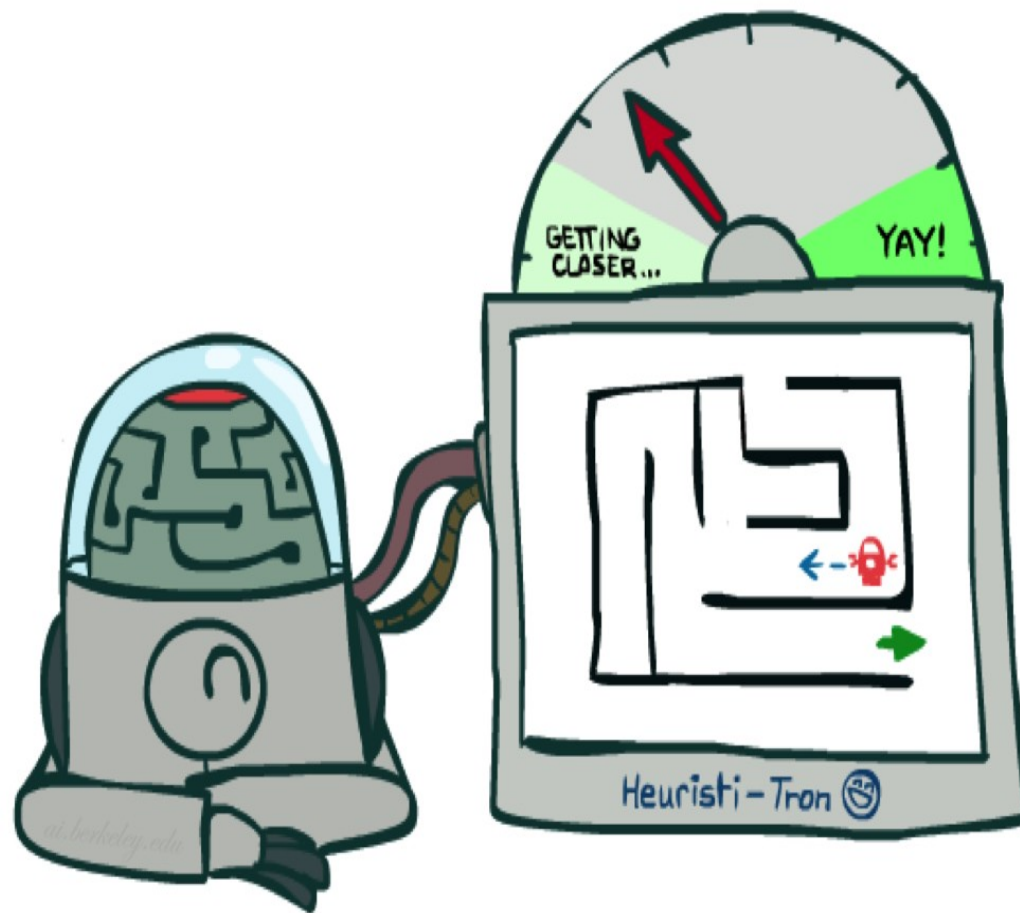
1.- $h(\text{Meta}) = 0$

2.- $h(s) - h(n) \leq c(s \rightarrow n)$

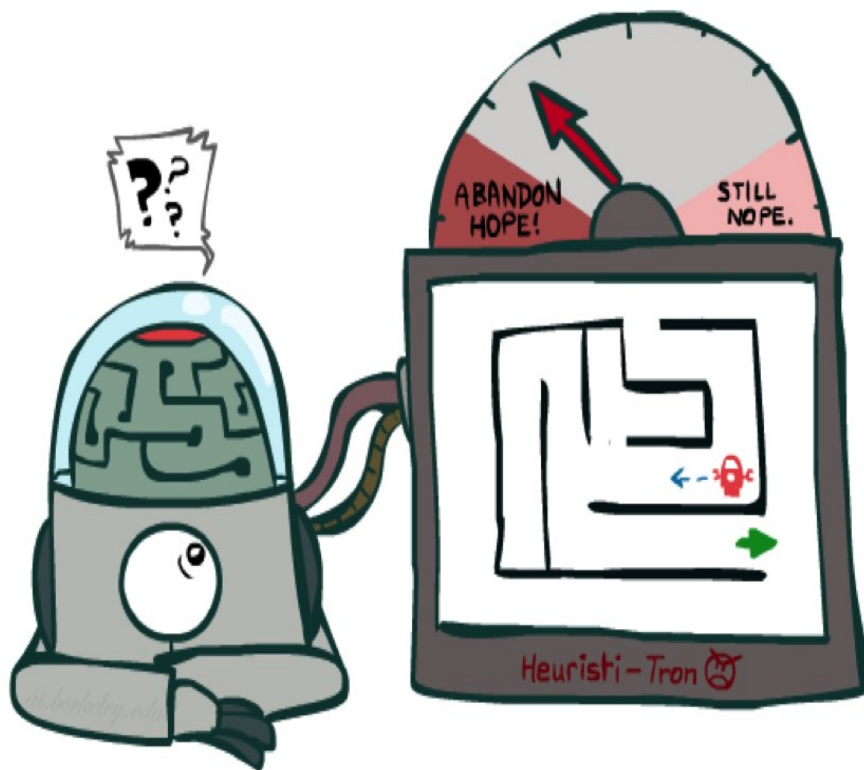
$c(s,n)$ = coste de transición de s a n



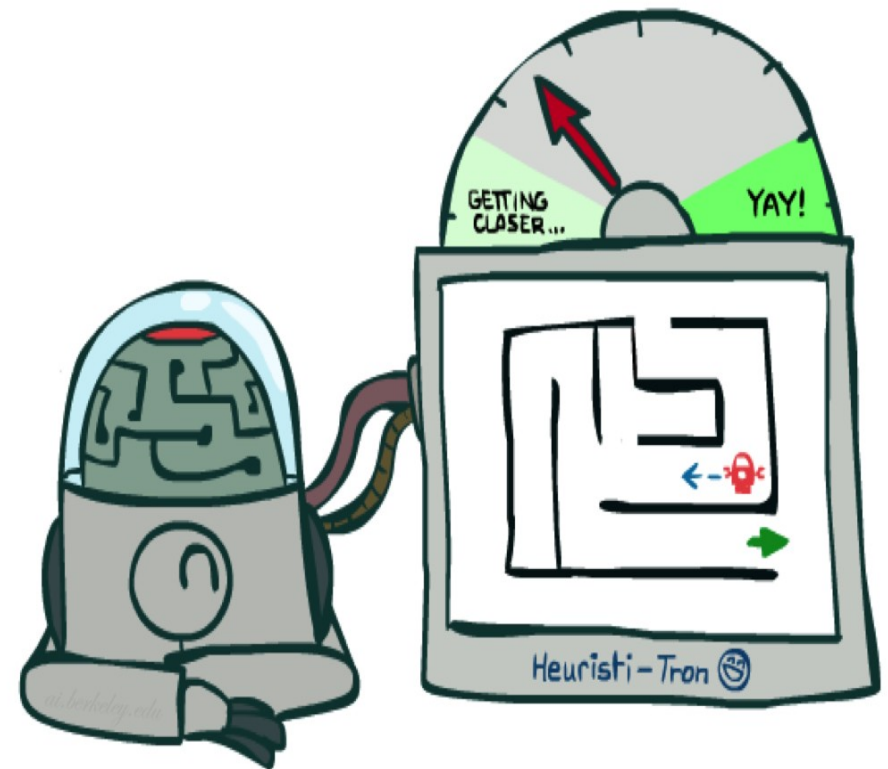
Heurísticos Admisibles



Idea: Admisibilidad



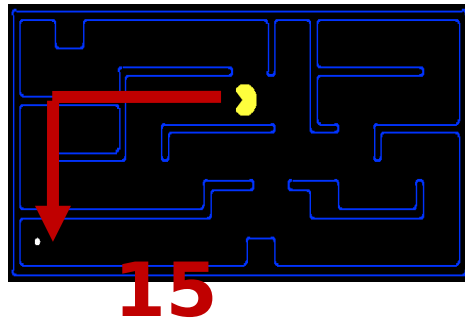
Heurísticos inadmisibles
(pesimistas) rompen la
optimalidad al *atrapar* buenos
planes en el borde (fringe)



Los heurísticos admisibles
(optimistas) ralentizan el
proceso (planes malos) pero
nunca superan el coste real

Heurísticos Admisibles

- Ejemplos:
 - Distancia de Manhattan
 - Distancia Euclídea



Optimalidad

- Tree search:
 - A* es óptimo si el heurístico es admisible
 - UCS es un caso especial ($h = 0$)
- La consistencia implica admisibilidad pero no al revés, aunque
- En general, la mayoría de los heurísticos admisibles tienden a ser consistentes, especialmente si vienen de problemas relajados

