

## Informed Search

(A\* Search & Heuristics)

### \* The One Queue

⇒ All these search algorithms are the same except for fringe strategies.

↳ Conceptually, all fringes are priority queue

{Collection of nodes with  
attached priorities}

### \* Search Heuristics

⇒ A heuristic is a function that estimates how close a state is to a goal.

↳ Designed for a particular search problem.

### \* Greedy Search

⇒ Expand the node that looks closest.

⇒ Worst case: like a badly guided DFS.

### \* A\* Search

⇒ Let  $g(n) \Rightarrow$  Path Cost

$h(n) \Rightarrow$  heuristic cost (distance of goal)



⇒ A\* Search orders by the sum:

$$f(n) = g(n) + h(n)$$

⇒ Admissible (optimistic) heuristics slow down bad plans but never overweights true cost.

⇒ Inadmissible (pessimistic) heuristics breaks optimality by trapping good plans on the fringe.

⇒ A heuristic  $h$  is admissible if

$$0 \leq h(n) \leq h^*(n)$$

where  $h^*(n)$  is the true cost to a nearest goal.

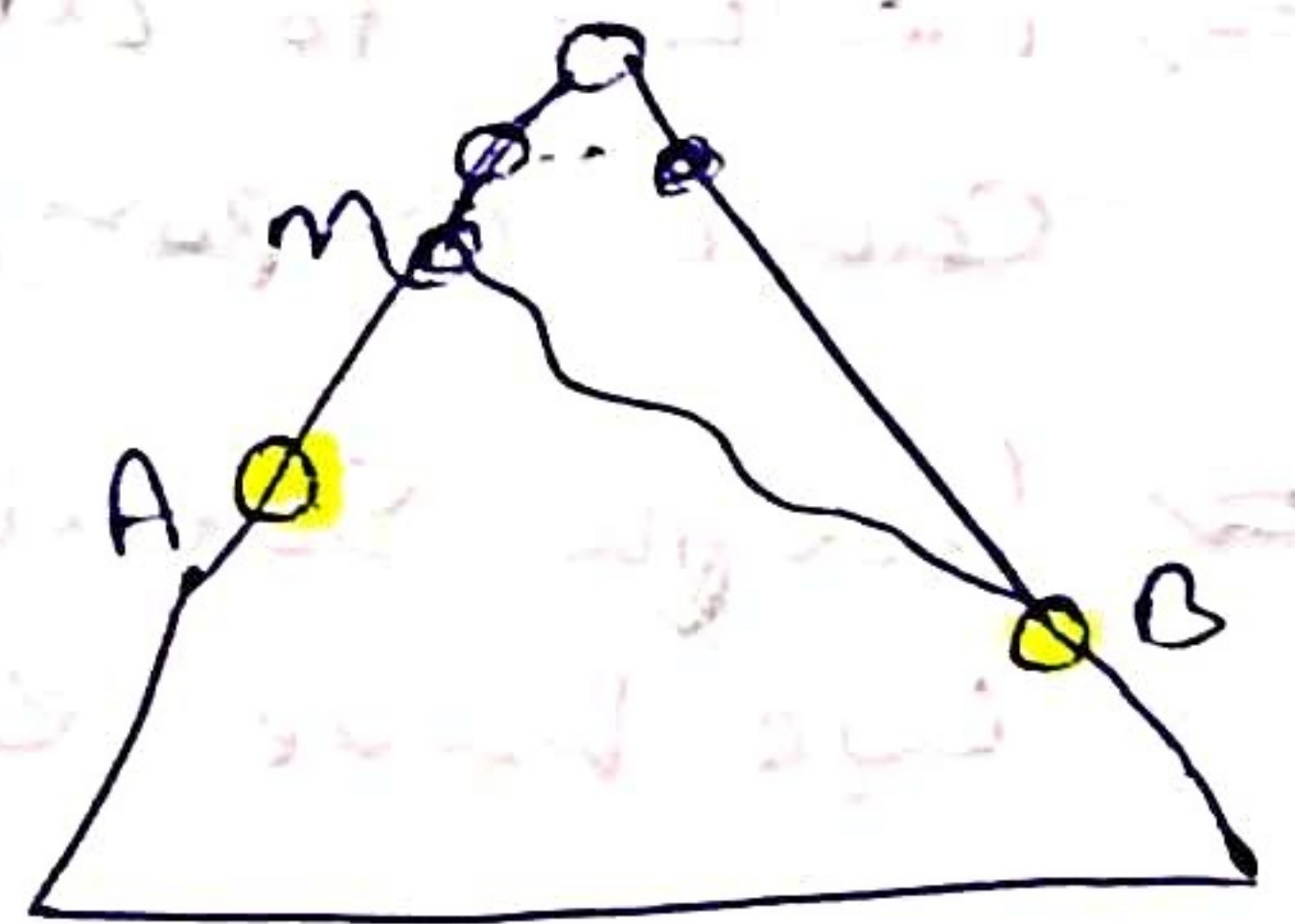
⇒ If heuristic is admissible then A\* Search is optimal.

Proof

Assumptions

~~Assumptions~~

- \* A is optimal goal node
- \* B is suboptimal goal node
- \*  $h$  is admissible



Claim

→ A Exits the fringe before B

⇒ Imagine B is on the fringe

⇒ Some ancestor  $n$  of A is on the fringe, too

•  $f(n) \leq f(A)$  {As  $h$  is admissible}

•  $f(A) < f(B)$  {As B is suboptimal}

•  $f(n) < f(B) \Rightarrow$  so  $n$  expands before

→ If  $n=A$  then A expands before B



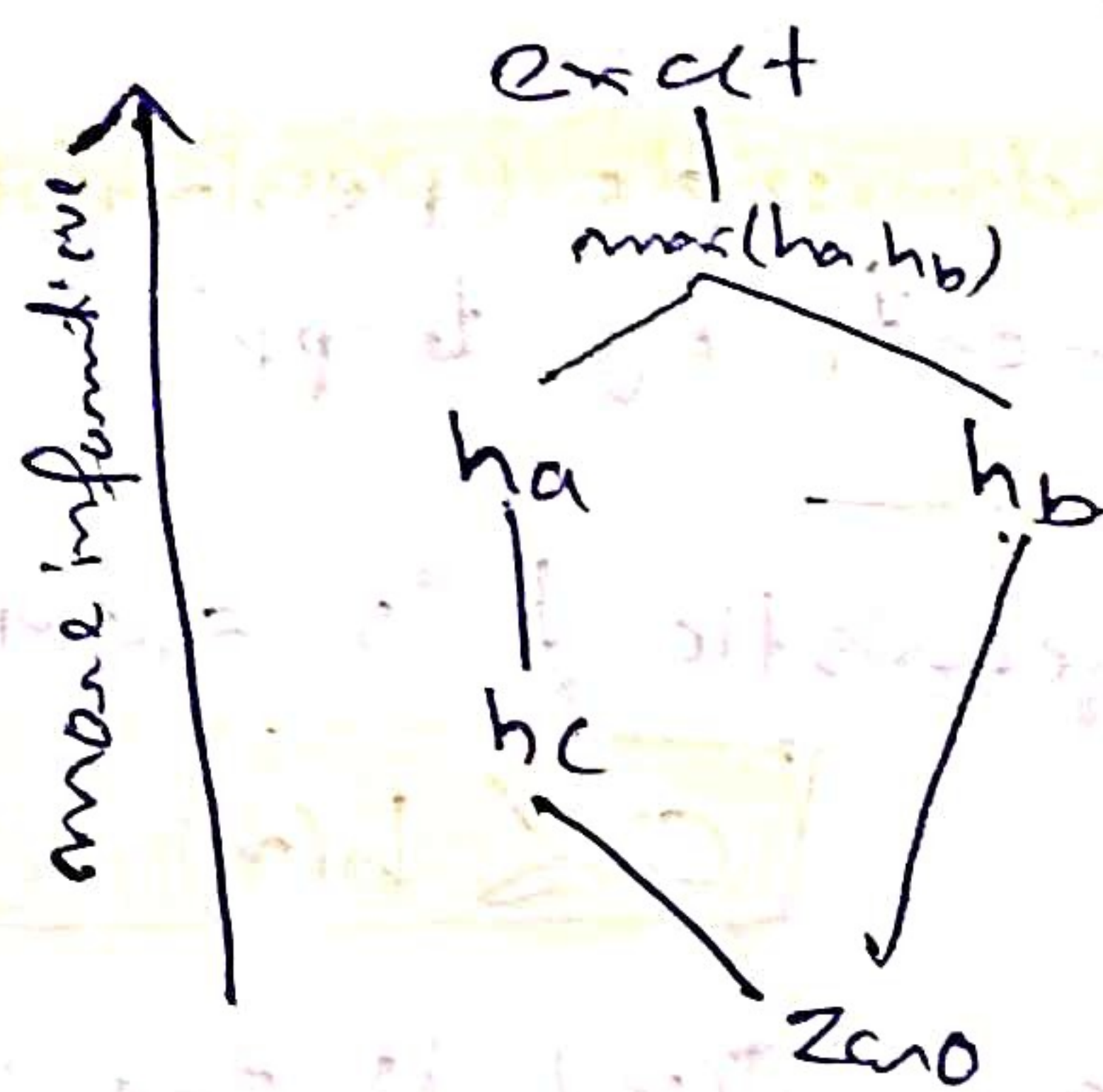
⇒ Most of the work in solving hard Search Problem optimally is coming up with admissible heuristics.

⇒ Relaxed-problem heuristic

### ★ Dominance

$$\forall n \ h_a(n) > h_b(n)$$

⇒ Max of admissible heuristics is admissible



### ★ Graph Search

Tree Search ⇒ Extra Work!

⇒ Failure to detect repeated states can cause exponentially more work.

⇒ Graph Search will keep track of list of nodes we have expanded and not expand them again.

⇒ How to implement:

- \* Tree Search + Set of expanded states ("Closed Set")
- \* Expand the Search tree node-by-node but
- \* Before expanding a node, check to make sure its state has never been expanded before.
- \* If not now skip it, if new add it to closed set.



⇒ For a graph  $A^*$  search to return optimal path, heuristic need to be consistent.

→ {A stronger condition than admissible heuristics}

⇒ Consistent heuristic:

$$h(A) - h(C) \leq \text{cost}(A \text{ to } C)$$

⇒ Consequence of consistency:

↳ The f value along a path never decreases

$$h(A) \leq \text{cost}(A \text{ to } C) + h(C)$$

$$f(A) = g(A) + h(A) \leq g(A) + \underbrace{\text{cost}(A \text{ to } C) + h(C)}_{f(C)}$$

$$f(A) \leq f(C)$$

