SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Prof. Jana Koehler, Dr. Álvaro Torralba

Exercise Sheet 2.
Solution

Exercise 6: A* and Hill Climbing. (3.5 Points)

Consider the state space from Figure 1.

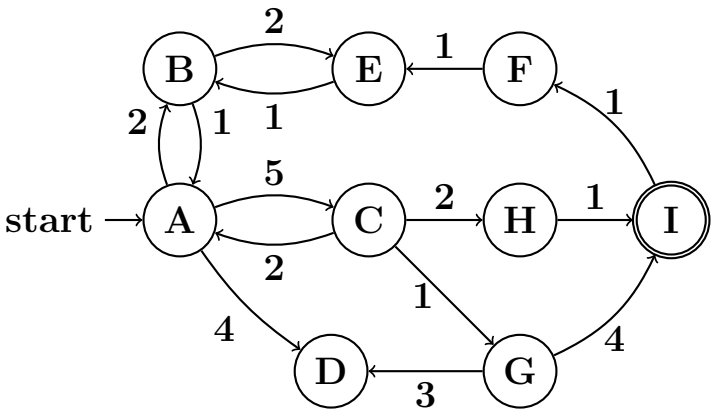


Figure 1: State space used in Exercise 6.

(i) Run A^* search on this problem. As a heuristic estimate for a state s, use the minimal number of edges that are needed to reach a goal state from s (or ∞ if s is not solvable, e.g., h(C)=2). Draw the search graph and **annotate each node with the g and h value as well as the order of expansion**. Draw duplicate nodes as well, and mark them accordingly by crossing them out. If the choice of the next state to be expanded is not unique, expand the lexicographically smallest state first. Give the solution found by A^* search. Is this solution optimal? Justify your answer.

(ii) Run the hill climbing algorithm, as stated in the lecture, on this problem. **Use the heuristic function from part (i)**. For each state, provide all applicable actions and the states reachable using these actions. **Annotate states with their heuristic value. Specify which node is expanded in each iteration of the algorithm.** If the choice of the next state to be expanded is not unique, expand the lexicographically smallest state first. Does the algorithm find a solution? If yes, what is it and is it optimal?

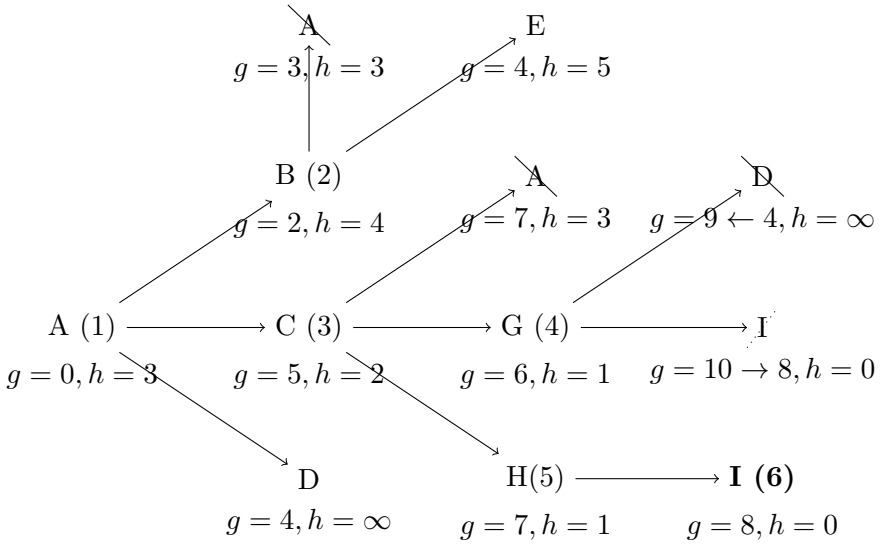


Figure 2: Solution to Exercise 6 (i). The algorithm stops when the goal node I is selected for expansion.

(iii) Could hill-climbing stop in a local minimum without finding a solution? If yes, give an example heuristic $h : \{A, B, ..., I\} \rightarrow \mathbb{N}_0^+ \cup \{\infty\}$ for the state space depicted in Figure 1 that leads hill-climbing into a local minimum, and explain what happens. If no, please explain why.

(Solution)

(i) The solution is A, C, H, I. A^* is guaranteed to return an optimal solution if the heuristic that is being used is consistent. Thus, it remains to show that the described heuristic h is consistent, i.e., that for every transition $s \xrightarrow{a} s'$, it holds that $h(s) - h(s') \leq c(a)$. By definition of our heuristic the following holds: $\forall s, s' \in S \setminus \{D\} : s \xrightarrow{a} s' \implies |h(s) - h(s')| = 1$. That means that the difference in heuristic values between any state s and any successor state s' of s is maximally 1, except for state D. In the special case of state D being the successor state, $h(s) - h(s') = -\infty$, which is less than the cost of any action (since D is a dead end it does not have any successors). Otherwise the minimal cost for any transaction is 1, so the heuristic is consistent.

(ii) Hill climbing returns the solution A, C, G, I, which is not optimal. The expansion order is as follows:

- State A
  $A \to B$: $h(B) = 4$
  $A \to C$: $h(C) = 2 \Leftarrow$ expand
  $A \to D$: $h(D) = \infty$
- State C
  $C \to A$: $h(A) = 3$
  $C \to G$: $h(G) = 1 \Leftarrow$ expand
  $C \to H$: $h(H) = 1$
- State G
  $G \to D$: $h(D) = \infty$
  $G \to I$: $h(I) = 0 \Leftarrow$ expand
- State I
  $I \to F$: $h(F) = 6$
- I is a local minimum, search stops.

(iii) Yes it is possible that hill-climbing stops without finding a solution. Consider the following heuristic:

State s	A	B	C	D	E	F	G	H	I
h(s)	2	2	2	2	2	2	2	2	0

Hill climbing starts at A (with $h(A) = 2$) and chooses randomly one of its child nodes (B, C, or D). Since either one of them has an equal heuristic value of 2, Hill Climbing stays at node A, which is then returned as a local minimum without finding the solution.

(/Solution)

Exercise 7: Admissible Heuristics. (2.5 Points)

Consider a variant of the Vacuum Cleaner problem from the lecture where a robot has to clean a 3×3 square room (see Figure 3). There are four possible actions: up, left, right, and down. There is no suck action since the robot automatically cleans any dirty spot it

stands on. Hence, its task is moving around the room and visit all dirty spots. Throughout the exercise, we use Manhattan distance.

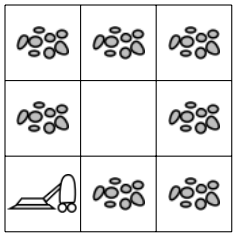


Figure 3: Illustration of the Vacuum Cleaner problem

Which of the following heuristics are admissible? Why / Why not? Give clear proof arguments. (Formal proofs are not needed.)

(i) h_1 = Number of dirty spots.

(ii) h_2 = Number of clean spots.

(iii) h_3 = Sum of the distances from the robot to all the dirty spots.

(iv) $h_4 = h_1 + h_2$.

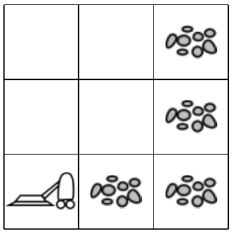
(v) h_5 = Minimum distance from the robot to any dirty spot.

Note: To prove a heuristic admissible, you need to show that it is admissible for every state of the illustrated 3×3 example. To show that a heuristic is not admissible, a counter example is sufficient.

(Solution)

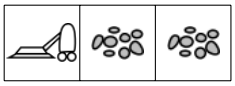
(i) h_1: Admissible, because to clean one dirty spot it takes at least one action and one action can clean at most one spot. the robot needs to reach at least one dirty spot.

(ii) h_2: Not admissible (Not even a heuristic, since $h_2(G) \neq 0$)). Consider the following counter example:



Here, $h_2 = 5$, while $h^* = 4$, which makes it not admissible.

(iii) h_3: Not admissible. Consider the following counterexample, where $h_3 = 1 + 2 = 3 > 2 = h^*$ (assume that the rest of the board has already been cleaned):



(iv) h_4: Not admissible. Consider again the counter example from h_3. It is $h_5 = h_1 + h_2 = 2 + 1 = 3$, but $h^* = 2$. (Since it incorporates h_2 which is not a heuristic, the combination is also not admissible).

(v) h_5: Admissible, because the robot needs to reach at least one dirty spot.

(/Solution)

Exercise 8: DNF. (4 Points)

Transform the following formulas to DNF. To do so, follow the steps from the lecture (Chapter 5 slide 19) specifying which steps are you applying and giving the intermediate results.

Note: Simplify the resulting formulas where possible.

(a) $(P \to R) \to (Q \leftrightarrow \neg R)$

(b) $\neg P \wedge (Q \vee \neg(\neg R \vee \neg Q))$

(c) $\neg(A \to \neg(\neg B \vee \neg C))$

(d) $(A \leftrightarrow \neg D) \vee (\neg C \wedge B)$

(Solution)

(a)
- Eliminate $\leftrightarrow$: $(P \to R) \to \big((Q \to \neg R) \land (\neg R \to Q)\big)$
- Eliminate $\to$: $\neg(\neg P \lor R) \lor \big((\neg Q \lor \neg R) \land (R \lor Q)\big)$
- Move $\neg$ inwards: $(P \land \neg R) \lor \big((\neg Q \lor \neg R) \land (R \lor Q)\big)$
- Distribute $\land$ over $\lor$: $(P \land \neg R) \lor \big((\neg Q \land R) \lor (\neg Q \land Q) \lor (\neg R \land R) \lor (\neg R \land Q)\big)$

DNF: $(P \land \neg R) \lor (\neg Q \land R) \lor (\neg Q \land Q) \lor (\neg R \land R) \lor (\neg R \land Q)$

(b)
- Move $\neg$ inwards: $\neg P \land \big(Q \lor (R \land Q)\big)$
- Distribute $\land$ over $\lor$: $(\neg P \land Q) \lor \big(\neg P \land (R \land Q)\big)$

DNF: $(\neg P \land Q) \lor (\neg P \land R \land Q)$

(c)
- Eliminate $\to$: $\neg\big(\neg A \lor \neg(\neg B \lor \neg C)\big)$
- Move $\neg$ inwards: $\big(A \land (\neg B \lor \neg C)\big)$
- Distribute $\land$ over $\lor$: $(A \land \neg B) \lor (A \land \neg C)$

DNF: $(A \land \neg B) \lor (A \land \neg C)$

(d)
- Eliminate $\leftrightarrow$: $\big((A \to \neg D) \land (\neg D \to A)\big) \lor (\neg C \land B)$
- Eliminate $\to$: $\big((\neg A \lor \neg D) \land (D \lor A)\big) \lor (\neg C \land B)$
- Distribute $\land$ over $\lor$: $\big((\neg A \land D) \lor (\neg D \land D) \lor (\neg A \land A) \lor (\neg D \land A)\big) \lor (\neg C \land B)$

DNF: $(\neg A \land D) \lor (\neg D \land D) \lor (\neg A \land A) \lor (\neg D \land A) \lor (\neg C \land B)$

(/Solution)