COMP3702/7702 ARTIFICIAL INTELLIGENCE

Semester 2 2017: Tutorial 3 Solutions

Question 1

a)

We know that h_1 is an admissible heuristic. That is to say, it does not overestimate the true cost to the goal. From the question, we know that $h_1(s)$ has values between 0.1 and 2.0.

$h_2(s) = h_1(s) + 5$. We can't guarantee that $h_2(s)$ is admissible because $h_2(s)$ is larger than $h_1(s)$. Because of this, we cannot say for sure whether or not $h_2(s)$ overestimates the true cost to the goal. If however, $h_2(s)$ was smaller than or equal to $h_1(s)$ for any state s, then we could guarantee this.

$h_3(s) = 2h_1$. This heuristic will always be larger than $h_1(s)$. Thus, for the same reason as $h_2(s)$, we cannot guarantee its admissibility.

$h_4(s) = \cos(\pi h_1(s))$. We know that: $-1 \leq \cos(x) \leq 1$ and that $0.1 \leq h_1(s) \leq 2.0$. This means that there may be some state s, such that $\cos\left(\pi h_1(s)\right) > h_1(s)$. For example when $h_1(s) = 0.1$, $h_4(s) = \cos(0.1\pi) = 0.99998 > 0.1$. For this reason, we cannot guarantee the admissibility of $h_4(s)$.

$h_5(s) = h_1(s)\cos(\pi h_1(s))$. This heuristic is equivalent to: $h_1(s)\, h_4(s)$. Since the output of $h_4(s)$ is being scaled by $h_1(s)$, we can guarantee that $h_5(s) \leq h_1(s)$. Since we know that $h_1(s)$ is admissible and thus never overestimates the true cost to the goal, we can guarantee the same for $h_5(s)$.

b)

Despite $h_2(s)$ not being admissible, we can guarantee that it generates an optimal path. Consider three states s_1, s_2 and s_3 such that $h_1(s_1) < h_1(s_2) < h_1(s_3)$. Since $h_2(s) = h_1(s) + 5$, we can say that $h_2(s_1) < h_2(s_2) < h_2(s_3)$. As such, the order in which nodes are expanded from the frontier of a minimum-cost priority queue in an A* search

using $h_1(s)$ and $h_2(s)$ are the same. Since we know that using $h_1(s)$ produces an optimal path, we can guarantee that $h_2(s)$ also produces an optimal path.
A similar argument can be used to show that using $h_3(s)$ in an A* search will also result in an optimal path despite $h_3(s)$ not being an admissible heuristic.

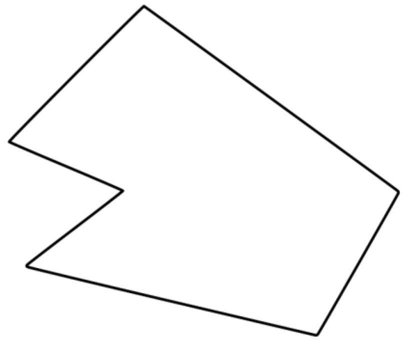
$h_4(s) = \cos\left(\pi h_1(s)\right)$. Given three states s_1, s_2 and s_3 such that $h_1(s_1) < h_1(s_2) < h_1(s_3)$, we cannot guarantee that $h_4(s_1) < h_4(s_2) < h_4(s_3)$. To check this, consider the following: $h_1(s_1) = \frac{1}{4}$, $h_1(s_2) = \frac{1}{2}$, $h_1(s_3) = \frac{3}{4}$. In this case $h_4(s_1) = \frac{\sqrt{2}}{2}$, $h_4(s_2) = 0$, $h_4(s_3) = \frac{-\sqrt{2}}{2}$. This means that the order in which nodes are expanded from the frontier of the priority queue can differ between an A* search using $h_4(s)$ and $h_1(s)$. Because $h_4(s)$ is also a non-admissible heuristic, we cannot guarantee that using $h_4(s)$ will generate an optimal path.

For the same reason as $h_4(s)$, using $h_5(s)$ in an A* search may expand nodes from the priority queue in a different order to an A* search using $h_1(s)$. But because $h_5(s)$ is an admissible heuristic, it will still generate an optimal path.

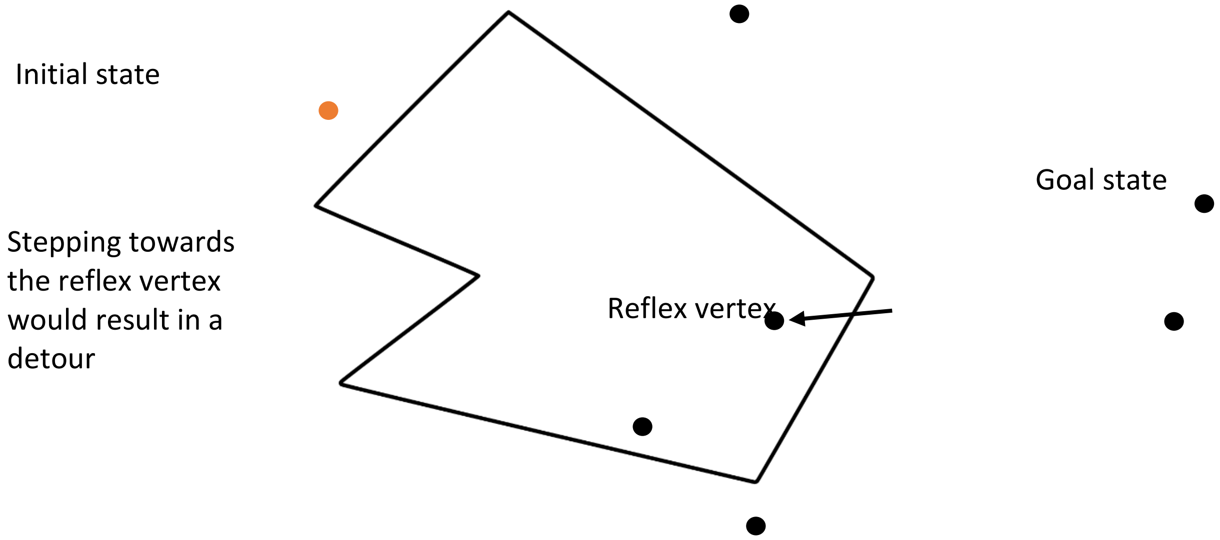
Question 2

Yes we can remove vertices and edges from G and still ensure that the shortest collision-free path between a given initial and goal point can be found.

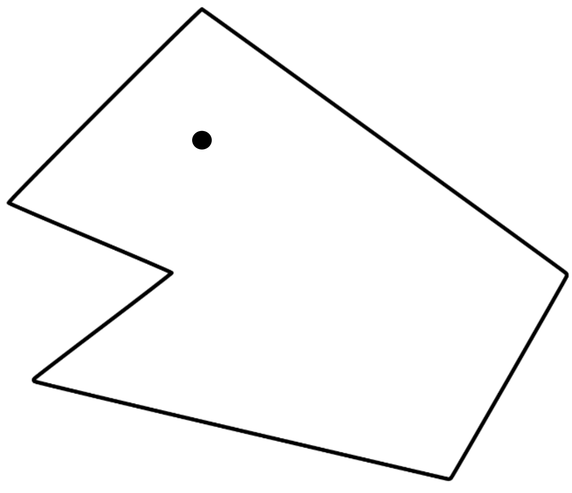
For visualisation purposes, we will consider this non-convex polygon to be an example of an obstacle in the question.

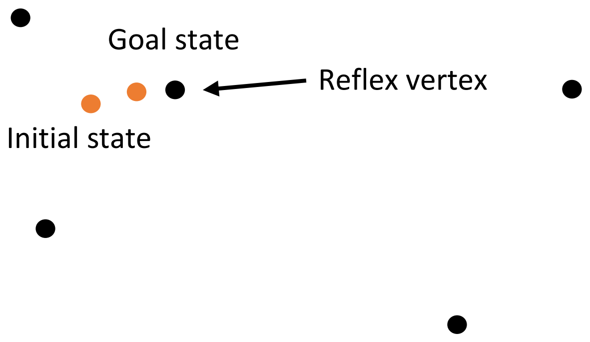


We can remove the reflex vertices and all the edges are connected to it.



If the obstacle vertex nearest to the given initial/goal states is not a reflex vertex, then a path that moves to the reflex vertex is essentially a detour and will be longer than a path that does not pass through such a vertex.





Even if the obstacle vertex nearest to the given initial/goal states was a reflex vertex, stepping to the reflex vertex would still result in a detour. One could move more directly towards the goal state without moving towards the reflex vertex.

Therefore, removing reflex vertices and edges connected to it from the visibility graph will not change the optimality of the path we can find.

Question 3

a)

The C-space for this robot is real one-dimensional space spanning 0 to 360.

b)

The configuration space is drawn in green [0, 360]

The forbidden region is drawn in red [45, 135]