MLND SmartCab Project Report

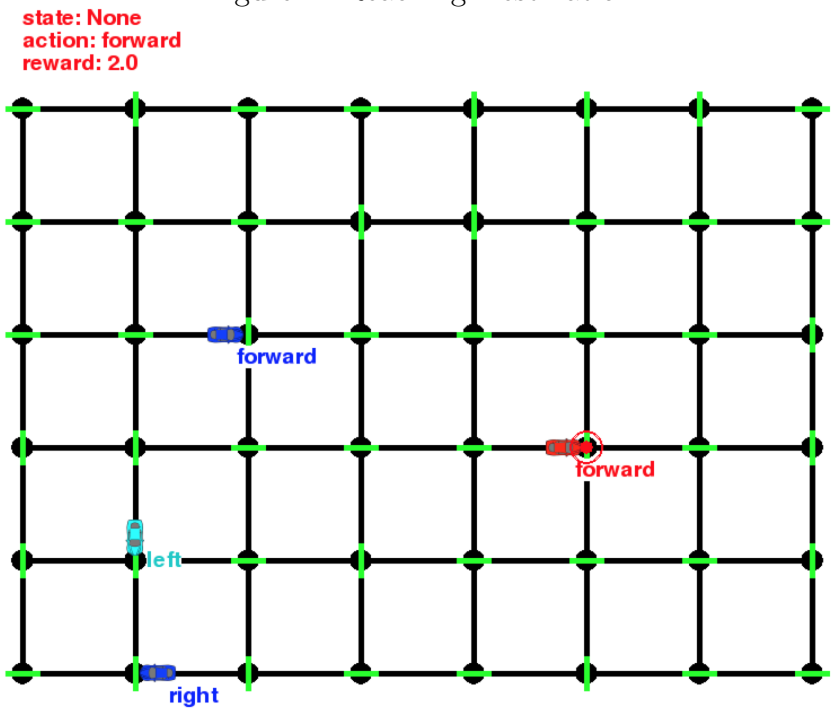
Juanyan Li

November 7, 2016

1 Implement a Basic Driving Agent

The cabs are all moving in random directions, including not taking no action. However, they are still restricted by the enforced traffic rules (no forward and left turn on red light). In this fashion, althoug not likely, the primary agent (red cab) is still possible to reach the destination by chance (red spot, see Fig 1).

Figure 1: Reaching Destination

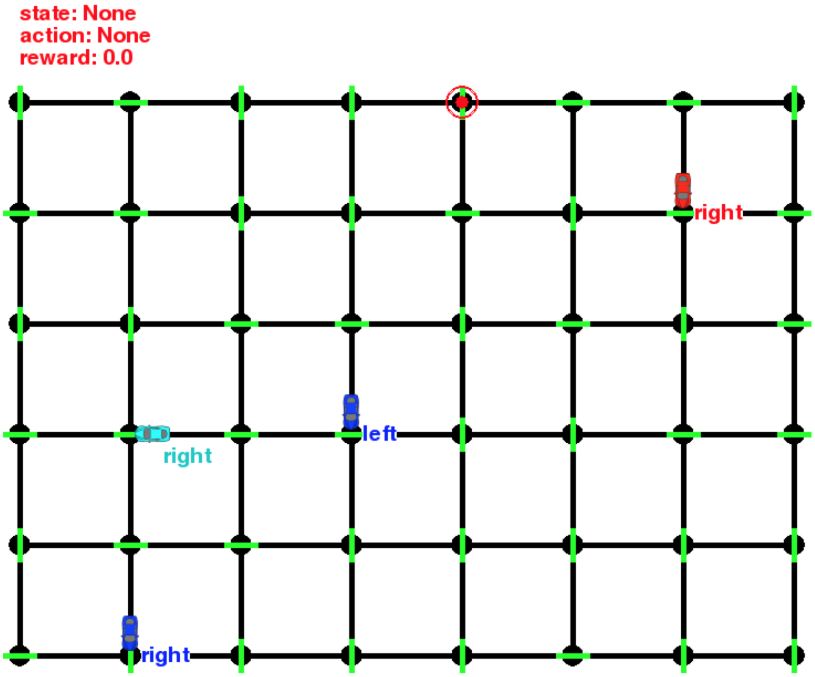


One interesting thing that I notice is the reward mechanism. There are 4 cases in total:

(i) **Action is `None`**

Agent receives no reward when no action is taken at current time step (see Fig 2).

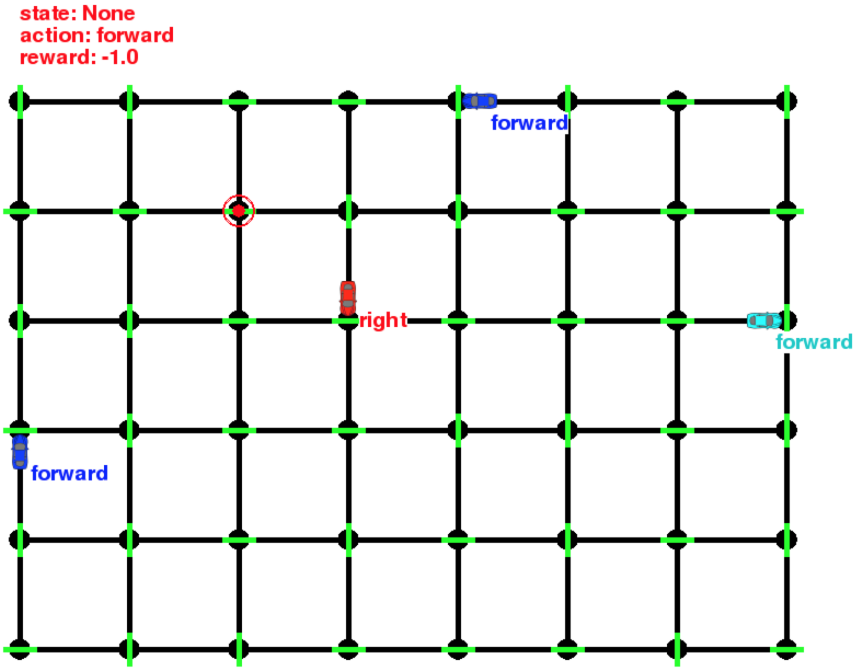
Figure 2: No Action



(ii) **Action is prohibited by traffic rule**

Agent receives -1.0 reward when proposed action violated the traffic rule at current time step (see Fig 3).

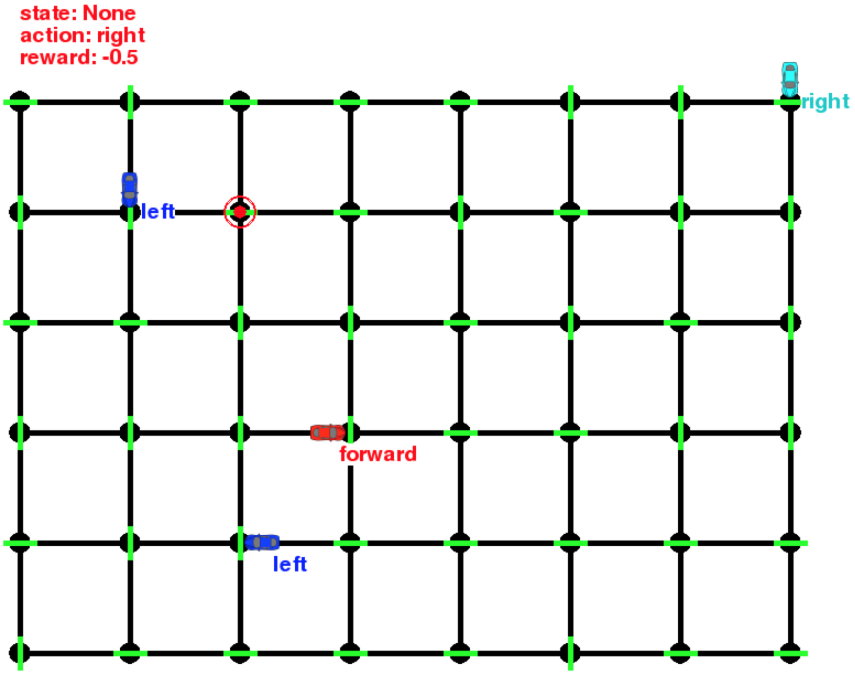
Figure 3: Violating Traffic Rules



(iii) **Action is moving away from destination**

Agent receives -0.5 reward when proposed action is heading away from destination (see Fig 4).

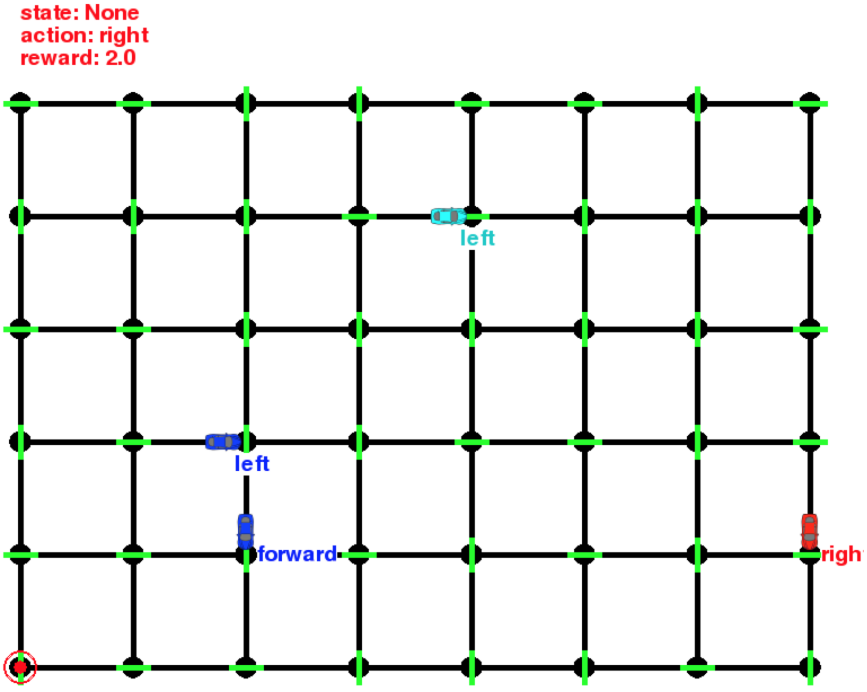
Figure 4: Heading Towards Wrong Direction



(iv) **Action is moving towards destination**

Agent receives 2.0 reward when proposed action is heading towards destination (see Fig 5).

Figure 5: Heading Towards Right Direction



2 Inform the Agent

1) **Define states**

The states for the learning agent are defined in a way that it would provide useful guidance for what action(s) to be taken under traffic conditions. The `sense()` function in class `Environment` give four outputs: traffic light, oncoming traffics, traffics from the left and right. The outputs can be seen as contraints that limit the available actions the agent can take at the current step. The following table describes all the possible situations:

	Red	Green
Oncoming	{none,right}	{forward,right}; {none, left}
Left	{forward}; {none,left,right}	{none, right}
Right	{none,forward,left,right}	{none,right}

Take red light for example, when the agent is facing a red light, oncoming traffic can have actions: None and right. Either one would affect the agent's decision, hence they can be grouped together. Similarly, traffic from the right can have actions: None, forward, left, right, but none of them would influence the agent in our setting. However, traffics from the left are different: if it is moving forward, then the agent cannot take a right turn; otherwise the agent would have the same behavior as the previous two cases. Same analysis can be applied in the green light's case.

In conclusion, a total of four states can be defined according to the avaible actions the learning agent can take:

(i) 'NR': agent can take no action or turn right. This happens when it is red light and no forward traffic from the left;

(ii) 'N': agent can only stay where it is. This happens when it is red light and there is forward traffic from the left;

(iii) 'NFR': agent can stay, go forward or take a right turn. This happens under green light with oncoming traffic turning right or moving forward;

(iv) 'NFLR': agent can take any valid actions. This happens under green light and there is no oncoming traffic moving forward or taking right turn.

2) **State space**

Given the state space defined in 1), the cab will strictly obey the traffic rules at each intersection. Instead of choosing all random actions, the cab will now randomly choose from a set of available actions permitted by the current status. It is possible that the cab can reach the destination after a number of trials.

3) **Update state**

The learning agent will update its state based on the current inputs sensed from the environment. Below shows some possible states appearing during the simulation.