Second Mid-Term

CS 599: Autonomous Cyber-Physical Systems

Instructor: Jyotirmoy V. Deshmukh

Assigned: April 29, 2019. Due (by email): **May 8, 2019**

Instructions:

1. **In this exam we will use a unique single digit number that we will generate from your USC student ID. Add all digits of your USC ID. Then add all the digits of the result. Keep doing this till you have a single positive integer in $[1, 9]$ – we will call this your key. For example, if my USC ID is 7895, then my key is 2**.
2. Please feel free to refer to any material that you deem fit.
3. Discussions among fellow students are generally not encouraged. If you do wish to ask questions, please do it on Slack or through email where Nicole or I will answer. Please adhere to the academic integrity policy.
4. We prefer that you turn in a pdf by email before class to both Nicole and me. You can also hand in a printed/handwritten copy in class.

Problem 1. [20 points] Consider a 2D linear dynamical system as given below. Here, k is your key.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -10 & k \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{1}$$

Given a set of initial states I and a set of unsafe states F, recall that a strict barrier certificate for a system of the form $\dot{\mathbf{x}} = f(\mathbf{x})$ is defined using a function $B(\mathbf{x})$ which has the following properties:

1. $\forall \mathbf{x} \in I$: $B(\mathbf{x}) \leq 0$,
2. $\forall \mathbf{x} \in F$: $B(\mathbf{x}) > 0$,
3. $\forall \mathbf{x}$ s.t. $B(\mathbf{x}) = 0$, $\frac{\partial B}{\partial \mathbf{x}} f(\mathbf{x}) < 0$

Let I be defined as: $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$. Let F be defined as $x_2 > 10$ or $x_2 < -10$. The value of x_1 is unconstrained for the set F. Your task is to find a barrier certificate that proves safety for the above system.

Hints. If it helps, you can use Matlab to plot the zero *level set* of your barrier function (i.e. the function $B(\mathbf{x}) = 0$). You can also use the plots to prove the first two conditions of the barrier certificate. If you are taking the Matlab route, feel free to explore functions such as `lyap`, `fimplicit`, `plot`. These can help you with finding the barrier certificate and plotting. *Just to be clear, you don't have to use Matlab at all, but it is only if you find it useful, and don't like doing pen-and-paper symbol manipulations.*

Also, for this problem, an axis-aligned ellipse is enough to serve as a barrier certificate. (I.e. a quadratic function of the form $ax_1^2 + bx_2^2 - c$, for some positive a, b, c.)

Extra Credit. [10 points] Can you find a barrier certificate that proves system safety for every value of k?

Problem 2. [30 points] In this problem, we are going to do the value iteration algorithm for reinforcement learning. First, consider the following Markov Decision Process that describes a little girl who likes mangoes. Assume that the girl is under a mango tree. She can have one of two states: q_0 is the state indicating she is on the ground, q_1 indicates that she is on the tree. The set of actions for the girl are as follows: s is an action indicating that the girl is going to stay where she is, c is an action indicating that the girl is climbing the tree. Each transition of the MDP is a tuple (q, a, p, q', r), where q, q' are the from and to states respectively, a is the action, p is the probability for this transition, and r is the reward. Let $P(q, a, q')$ denote the probability of going from q to q' with the action a, and $R(q, a, q')$ denote the reward for this transition. Let $S = \{q_0, q_1\}$, and $A = \{s, c\}$. The MDP has the following transitions:

When the girl is on the ground:

1. The girl is on the ground, decides to stay on the ground, and with 0.1 probability, a mango lands at her feet. The "reward" for eating the mango is 10. I.e., $\tau_1 = (q_0, s, 0.1, q_0, 10)$.

2. The girl is on the ground, decides to stay on the ground, but spots a snake in the grass with 0.9 probability, gets scared, and climbs the tree instead $\tau_2 = (q_0, s, 0.9, q_1, -2)$.

3. The girl decides to climb the tree. Each student gets a different little girl here. The probability that she succeeds in climbing the tree is $k/10$, where k is your key. There is effort associated with climbing the tree, this manifests as a negative reward. $\tau_3 = (q_0, c, 0.1k, q_1, -2)$.

4. The girl decides to climb the tree. But poor thing fails in doing so, and actually falls. The reward reflects this fall. $\tau_4 = (q_0, c, 1 - 0.1k, q_0, -10)$.

When the girl is on top of the tree:

1. The girl decides to stay on the tree. With probability $1 - 0.1k$, she finds a mango and eats it. The reward reflects her happiness. $\tau_5 = (q_1, s, 1 - 0.1k, q_1, 20)$.

2. The girl decides to stay on the tree, but with $0.1k$ probability, she slips and falls. Ouch! $\tau_6 = (q_1, s, 0.1k, q_0, -10)$.

3. The girl decides to climb down from the tree. With $0.1k$ probability, she discovers a giant bunch of mangoes (yum!) and changes her mind about climbing down: $\tau_7 = (q_1, c, 0.1k, q_1, 25)$.

4. The girl decides to climb down from the tree. She fails and falls! Ouch again! $\tau_8 = (q_1, c, 1 - 0.1k, q_0, -10)$.

Value iteration is a dynamic programming algorithm to compute the optimal policy of a given MDP, i.e. in every state it tells which action to take. In the above example, the controller is the little girl's brain, and we are going to help the girl figure out the best action to take in each state. The pseduo-code for value iteration is given below:

For each $q \in \{q_0, q_1\}$, $V_0(q) = 0$;
run := $true$;
while run **do**
  $i := i + 1$;
  foreach $q \in S$ **do**
    $$V_{i+1}(q) = \max_a \sum_{q'} P(q, a, q')(R(q, a, q') + \gamma V_i(q')) \;;$$
  end
  run := $C(V_{i-1}, V_i)$;
end

The above algorithm terminates based on the value of the Boolean flag run, which itself depends on some condition on the values of the value function between consecutive iterations. For this problem, this condition is not important. Suppose the value function after the algorithm terminates is V^*. The optimal policy is then defined as $\pi(q) = \arg\max_a \sum_{q'} P(q, a, q')(R(q, a, q') + \gamma V^*(q'))$. For this example, let $\gamma = 0.9$.

Your task is to compute three iterations of the value iteration algorithm. Also compute the optimal policy after the final iteration.

Problem 3. [25 points] Consider an asynchronous process with two tasks:

$$\begin{array}{llll} T_x: & (y > 0) & \rightarrow & x := (x+1) \bmod 3; \\ & & & y := 0 \\ & & & \\ T_y: & (x \leq 2) & \rightarrow & y := (y+1) \bmod 3; \\ & & & x := (x+1) \bmod 3; \end{array}$$

Yes, it's very similar to the one from the first exam, where I made a mistake in the guards!

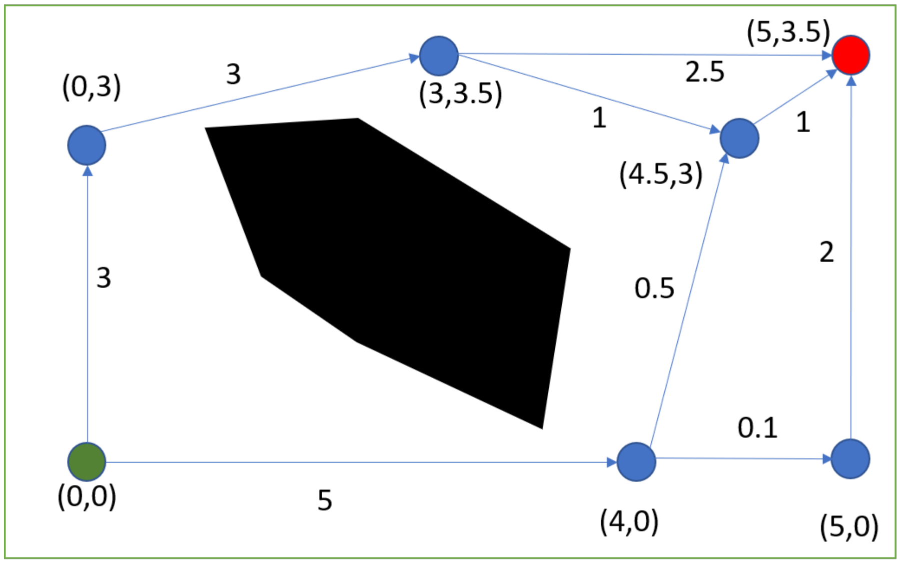

Figure 1: Probabilistic Road-Map

(a) [5 points] Starting with the initial state $x \mapsto 0, y \mapsto 0$, show the underlying transition system of this process.
(b) [20 points] Does this transition system satisfy the LTL property $\mathbf{F}(y = 2)$ under strong fairness assumptions? (I.e. under the assumption that when the guards $(y > 0)$ and $(x \leq 2)$ are infinitely often enabled, the scheduler will respectively execute T_x and T_y infinitely often).

Problem 4. [20 points]

Consider the PRM shown in Fig. 1. We want to find the shortest path from the green point to the red point. Having learned about A* search, I decide to use it because it does not require me to compute the shortest path from the start node to every node in the graph. A* does need a heuristic to estimate the cost at each node in the graph. I decided to use the heuristic where I estimate the cost to reach the goal g from any point p by the Euclidean distance between p and g. Can I do this? If you answered yes, what is the justification? If you answered no, why can I not do this?

Extra Credit [10 points] If I can use the Euclidean distance to goal heuristic, find the shortest path with A*, and compare it with the shortest path obtained using Dijkstra's algorithm. If I cannot use the Euclidean distance to goal heuristic, suggest a new heuristic that will work.

Problem 5. [5 points]
(a) What is the one thing that you will change in this course? (b) What is the one thing that you want to learn more about in this course?