Result and Findings

Correct-by-synthesis reinforcement learning with temporal logic constraints

Karan Muvvala

Results: For this course project, my primary aim was to reproduce the results illustrated Example 1. in the original paper and verify that the strategies learned by the system robot operating in an adversarial environment with specification ϕ comprising of only safety assumptions ϕ^a on the system specification learns only the most optimal strategy μ^* i.e the system robot learns only the most optimal way to perform the given task - in our case the task is to always be in a diagonally opposite position to the environment robot as in equation 5. This optimal policy satisfies the temporal specification as per ϕ while also maximizing the rewards ensuring both qualitative (with respect to temporal specification) and quantitative (with respect to the underlying reward function) behaviors. This is evident by the plot in Fig 2. which shows a scatter plot of V (state values) of each state in $\hat{G}$ against the desired reward function R which in our case is a sum of discounted reward function as in (3) with $\gamma = 0.9$. For a 6 x 6 grid world, I can reach a diagonally opposite cell in 5 steps or less. Thus the learned optimal policy μ^* converges to the state values V as given by $\frac{1}{1-\gamma}\gamma^k$ for $k \in 1, .., 5$. The task used in Example 1. in my report is the same as the task in the original paper. Thus, the state values V for each state $s \in \hat{G}$ eventually converges to desired optimal values.

Example 2. is an extension of example 1 in which we include static obstacles in a 7 x 7 grid world to demonstrate that the strategy learned satisfies not only the temporal constraints of always not colliding with each other, but also satisfies constraints such as always not collide with static obstacles. Even in this case, we observe that the strategy learned is optimal with respect to the underlying reward function.

Findings: An interesting observation to note is that the number of total states $|S| \in \hat{G}$ in the original paper Table I is slightly lower than that in Table I in my report. I speculate that this could be due to the fact that the authors of the original paper manually remove some states that will eventually lead to collisions. Another interesting note is that the set of permissive strategies μ_p computed using Slugs are for turn-based games while the maximin_Q learning algorithm that the authors cite in the original paper is for stochastic concurrent games (*both players take actions simultaneously*). You can find the minimax_Q (a variant of maximin_Q) learning implementation in "learning_reward_function/Players/minimaxq_player.py". I did proceed with the minimax_Q algorithm but encountered some compatibility issues as the strategies computed by the Slugs toolbox are for turn-based game and the players usually collided with each other during the learning routine when learning as per the mimimax_Q learning algorithm thus failing to satisfy temporal constraints in Example 1. As the authors of the main paper also cite that there do exist convergence proof of Q-learning algorithm for alternating Markov games, it indeed gave me the same results as stated in Algorithm 1 of my report and as elaborated in the Examples section in my report.