1.1 Chess

You are tasked with creating an AI for the game of chess. To solve the problemusing Reinforcement Learning, you have to frame the game of chess as a Markov Decision Process (MDP). Describe both the game of chess formally as a MDP,also formalize the respective policy.

Ans>>

In chess, an MDP can be defined as follows:

State space S: The state space of chess is defined as all possible board configurations that can occur during a game.

Action space A: The action space of chess is defined as all possible moves that can be made by a player in a given state.

Transition function T: The transition function of chess is defined as the probability of moving from one state to another state given an action.

Reward function R: The reward function of chess is defined as the value assigned to each state-action pair. In chess, a positive reward can be assigned for winning the game, while a negative reward can be assigned for losing the game.

Discount factor γ: The discount factor of chess is defined as the factor by which future rewards are discounted.

1.2 LunarLander

Check out the LunarLander environment on OpenAI Gym: Check out this Link!.Describe the environment as a MDP, include a description how the policy is formalized.

ANS>>>

The Lunar Lander environment can be formalized as a Markov Decision Process (MDP)as follows:

The state space consists can be of the position and velocity of the lander and its orientation3.

The action space can be consists of four discrete actions: do nothing, fire left orientation engine, fire main engine, fire right orientation engine.

The reward function can be defined as follows: +100 points for landing on the landing pad, -100 points for crashing, and -0.3 points for each frame that passes3.

1.3 Model Based RL: Accessing Environment Dynamics

ANS>>>

Policy Iteration:

In policy iteration, we start by choosing an arbitrary policy and then we iteratively evaluate and improve the policy until convergence.

Policy evaluation:

Policy Evaluation computes the value functions for a policy using different equations.

The reward function can be defined as the expected reward that the agent achieves after completing an action 'a' while being in a state 's'.

The state transition function can be defined as the probability of reaching some future state 's`', given the agent performs some action 'a' while currently being in a state 's'.

Example 1 :- Chess Example

Example 2 :- Lunar Lander Example

The environment dynamics are not generally known due to the environment being a dynamical system with infinite states. Even if we assume that there is a single agent in a somewhat stable and small environment, there is the possibility of the agent's actions being non-deterministic and slippery. This means that the transition of the agent from one state to another will not be smooth and there will be a probability that the agent takes an erratic action that was not intended.

The classic exploration-exploitation dilemma also persists even in a simple environment with a single agent. This means that the agent has to find a balance between exploring and exploiting the states of the environment so that it doesn't get stuck in a state that provides just a local small reward and not a global big reward. The goal of the agent should be to minimize over fitting and explore all possible states and sequence of actions while keeping in mind the cost of performing each action and the cost of each sequence of actions