Homework 01 Review for Group 1

Task 01

Overall, your answers are mostly technically correct, but could be more formal and mathematically precise.

State description

- Chess piece position should be part of state description.

Actions

- Works, but depends on the board configuration. In an MDP, the set of actions should be independent of the state.

Reward

- This works! You could scale reward for defeating/losing a piece proportional to its value. Also, you could reward additional board configurations like checking your opponent.

Policy

- You are giving the global objective. The policy represents the probability distribution for all states.

Task 02

Again, your answers are mostly technically correct, but could be more formal and mathematically precise.

State description

- You are giving each states' components. However, there are more than 8 states as there are hundreds of possible coordinates for the lander to be on. Instead, the set of states contains all possible combinations of the lander's position, velocities, etc. .

Actions

- Could be more precise; can you fire all engines at once?

Reward

- Missing

Policy

- If you define the set of actions formally, you will have it easier to define the policy's output space/distribution.

Task 03

Your description of the reward function is correct, but the state transition function and environment dynamics work differently than you suggest. The state transition function does not contain all possible actions. It projects the current state and action on the future state. E.g. in chess, the current board configuration and move result in a different board configuration in the next turn. The future state is not known beforehand if the opponent's counter move cannot be predicted. Therefore, the state transition function is generally random, but it still exists (as a random variable).

Well Done :)

Homework 01 Review for Group 5

Task 01

- Nice description of the game itself!
- MDP description could have been a little more fine-grained i.e.
  - e.g. 8x8 chess board where each position can either be empty or one of the black or white chess pieces
  - You could describe what your transition probability function would do if the agent chooses to perform an illegal action
- What do you mean with “All movements of pieces for each s ∈ S”? → The set of all actions isn’t state dependent
- Reward function is nicely explained, except the last sentence is a bit confusing
  - reward dynamics usually do not take future states into consideration as r is only dependent on s and a at time step t
- Discounting factor isn’t part of the formal MDP description but it also doesn’t hurt to leave it there
- Correct functional description of a greedy policy, but formally it is a probability distribution over all possible actions given a state s

Task 02

- 2.1 Is a nice and easy to understand description of the entire system
- Just having the x,y coordinates as the state description is too simplistic
  - e.g. how would the lander check whether it has ground contact with one of its legs
  - velocity in both directions (x,y) and the rotation speed would also be relevant, otherwise the lander could not receive a reward for zero speed
- Otherwise the same critique points from task 01 hold

Task 03

- Correct and good description of policy evaluation
- Second screenshot displays pseudocode for policy evaluation and not iteration → missing description of how the policy is actually improved
- Explanations and examples for environment dynamics are missing. You could, for example, explain it using the Gridworld environment
- Also missing: “Discuss: Are the environment dynamics generally known and can practically be used to solve a problem with RL?“

Well Done :)

Homework 01 Review for Group 46

Task 01

General Feedback:

- formal definition of MDP is nice to have but not necessary
- Nice idea with the correction function that checks for the correctness of the move / the following state
- generally, really good description of the relevant components

Set of States:

- extensive description of the set of states (and initial state) with mathematical description - well done!
- both cases explained (all possible vs allowed board positions) which is nice

Set of Actions:

- theoretically, there are not 8x8 possibilities to pick a piece as only at max 16 chess pieces are available per side (black or white), but each piece can potentially be on every of the 64 fields → overall move set would be smaller
- apart from that again, nice and extensive description

Reward Dynamics

- you could have considered other game relevant board positions (e.g. setting or being set check)
- rewards could be given for taking or losing a piece depending on the piece's worth

Task 02

- Where do the speed limits come from? → The MDP description doesn't necessarily need them. Variables keeping track of the total speed in x & y as well as the rotational should suffice
- Wouldn't the total amount of surrounding area around (0.0) need to add up to 256 as the image is 256x256
- Mountains seem to be restricted to the lower part of the screen, thus living in a subspace of what you defined
- Overall, nice idea to have a example calculation for the set of states but kind of arbitrary values were chosen
- The remaining parts are nicely described - good job!

Task 03

- Nice explanation and examples for the reward function!
- Nice explanation and examples for the state transition function!
- in uncontrolled cases the environment dynamics are usually not known (e.g. robot interacting with the real world)

Well Done :)