MDPs

Exercise 1

(a) **Environment:** Opponent's chess positions (Chess position: a 'instant picture' of the own pieces at particular turn)
States: Player's own pieces' chess position (dim: *num_ of_ diff_pieces* x *num_ of_ legal_piece_pos*, discrete)
Actions: legal moves for all the pieces ≈ 80, discrete
Reward: $+1$ for winning, -1 for losing, 0 for for draw. No intermediate rewards

(b) **Environment:** 3D continuous space with coordinates of the object to pick up
States: Arm position in 8D space (continuous)
Actions: move the arm in one of the 8 dimensions (continuous) - back and forth, up and down, right and left, rotation
Reward: big positive reward for picking the object, e.g. 100, small negative rewards for each move, e.g. -1 to enforce efficiency.
The same for the placing procedure...

(c) **Environment:** 3D continuous space with possibly changing dynamics
States: position of each of the 4 rotors in 3D (continuous)
Actions: thrust power in each of the thrusters (4D, continuous)
Reward: Continuous reward for the time being on the same point on space

(d) Dialogue agent. Goal: accomplish a task the user asks for: book flight, provide information etc. in a dialogue setting. This is operating directly on utterance level, not on closed set of possible dialogue acts (templates) like most of the actual dialogue agents. The actual task can be accomplished after gathering enough knowledge from the user - turning unstructured text to structured information based on a knowledge graph, and reasoning om it.
Environment: User utterance
States: Discrete information - each user utterance produces new structured input - a edge or vertex on the knowledge graph. It's given from the environment and not agent's task to covert the utterance to structured input. Dimensions depend on the size of the knowledge graph
Actions: Pick a word from vocabulary - 40k (New states are encountered not after each action, but after building an utterance, when the agent hits the EOS token.)
Reward: Small negative for each dialogue turn, big positive reward if user marks task accomplished, big negative reward if human agent has to step in.

Exercise 2

1. In the case of MDPs each action is taken given some policy and each action can lead the agent to end in different subsequent state. From these states we can take another action and encounter new rewards, depending on our policy, so we wish to know the future rewards too. The expected return and thus the value of the action-state pair depends also on the policy. In bandit learning we don't maintain states. We end up in the same state after each action, we are not interested in the expectation of sequence of rewards but rather in expectation of the reward after taking this action.

   In bandit setting:

$$p(s', r|s, a) = p(r|a), \text{ because } s' = s, |S| = 1$$
$$\implies r(s, a) = \textstyle\sum_r r \sum_{s'} p(s', r|s, a) = \sum_r p(r|a)r = \mathbb{E}(R_t|A_t = a)$$

$$q(a, s) = \mathbb{E}(R_t + \gamma R_{t+1} + ...|S_t = s, A_t = a) \overset{bandits}{=}$$
$$= \mathbb{E}(R_t + \gamma R_{t+1} + ...|A_t = a) + ... \overset{linearity of cond.exp.}{=}$$
$$= \mathbb{E}(R_t|A_t = a) + \mathbb{E}(\gamma R_{t+1}|A_t = a) + ... \overset{bandits}{=}$$
$$= \mathbb{E}(R_t|A_t = a) + \gamma\mathbb{E}(R_t|A_t = a) + ...$$
$$\implies \text{considering future rewards doesn't provide us with no new information}$$

2. Given:

$$\textstyle\sum_a \pi(a|s) = 1 \implies \mathbb{E}(G_t|S_t = s, A_t = a)\pi(a|s) = \mathbb{E}(G_t|S_t = s)$$
$$\implies q_\pi(s, a)\pi(a|s) = v_\pi(s)$$

3. From the Bellman equation:

$$v_\pi(s) = \textstyle\sum_a \pi(a|s)(\underbrace{\sum_r \sum_{s'} p(s', r|s, a)r}_{r(s,a)} + \underbrace{\sum_{s'} \sum_r p(s', r|s, a)v_\pi(s')}_{p(s'|s,a)}) =$$
$$= \textstyle\sum_a \pi(a|s)(\sum_{s'} p(s'|s, a)r(s, a, s') + \sum_{s'} p(s'|s, a)v_\pi(s')) =$$
$$= \textstyle\sum_a \pi(a|s)(\sum_{s'} p(s'|s, a)(r(s, a, s') + v_\pi(s')))$$