Chapter 1 solutions **Hanna Gábor**

1. *Self-Play. Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself, with both sides learning. What do you think would happen in this case? Would it learn a different policy for selecting moves?*

   Yes, I expect it to learn a different policy: an optimal policy against an optimal policy. (Apart from the exploratory moves.)

   This can be shown by induction. For each state that necessarily leads to the end of the game, the greedy action will be the one that leads to a victory (if there exists such an action) and thus the value of this state after enough time steps will be really close to 1, if there exists a winning action and really close to 0 otherwise. Suppose that the values of states leading to the end of the game in maximum k steps are all really close to 0 or 1 reflecting who wins the game if both players play perfectly. For a state that leads to the end of the game in maximum $k+1$ steps, all the actions lead to states leading to an end in maximum k steps, hence after enough time steps, the greedy action in this state will be optimal and the value will be close to 0 or 1.

2. *Symmetries. Many tic-tac-toe positions appear different but are really the same because of symmetries. How might we amend the learning process described above to take advantage of this? In what ways would this change improve the learning process?*

   We could represent the states that are symmetricly equivalent as one state instead of many states. This would make the learning faster.

   Now think again. Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?

   No. If the opponent makes different choices in two symmetrically equivalent states, then we might have a dfferent winning chance in those, so we should not think of them as the same state.