Computational Intelligence in Games Summer 2019
Sanaz Mostaghim, Alexander Dockhorn, Christian Wustrau

2. Exercise Sheet

Exercise 1 Evolutionary Stable State

In which of the following payoff matrices is C an Evolutionary Stable Strategy. Why?

$$(A) = \begin{array}{c c} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{bmatrix} 6,6 & 0,5 \\ 5,0 & 1,1 \end{bmatrix} \end{array} \quad (B) = \begin{array}{c c} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{bmatrix} 5,5 & 2,5 \\ 5,2 & 1,1 \end{bmatrix} \end{array} \quad (C) = \begin{array}{c c} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{bmatrix} 0,0 & 2,1 \\ 1,2 & 1,1 \end{bmatrix} \end{array} \quad (D) = \begin{array}{c c} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{bmatrix} 1,1 & 0,1 \\ 1,0 & 1,1 \end{bmatrix} \end{array}$$

Exercise 2 Agent-Environment Interface, State Encoding

Consider the Pac-Man game in a small environment as in the picture. An agent can win and end the game, if it collects the cherry. The agent loses the game if it collides with the ghost. The game consists of the elements:

1) **Agent:** Pac Man
2) **Environment:** ghosts, cherries, walls
3) **Actions:** left, right, up, down (neutral)

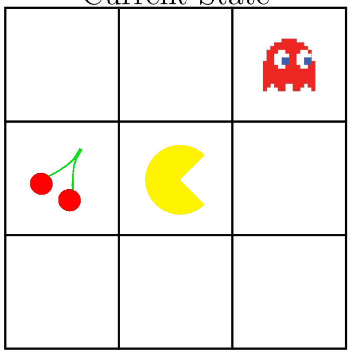
a) What is an appropriate reward for an agent, which is trying to learn how to win the game? Give an example.

b) Consider the following state encodings. What is the size of each encodings state space?

- **Encoding a)** Each state is represented as a 3x3 matrix. The content of each cell is encoded as integer number (0=empty cell, 1=agent, 2=ghost, 3=cherry, 4=wall)
- **Encoding b)** Each state is represented as a 3-tuple of 3 coordinates. The first coordinate describes the position of the agent, the second the position of the closest cherry, and the third the position of the closest ghost.
- **Encoding c)** Each state is represented as a 2-tuple of 2 relative coordinates. The first coordinate describes the relative position of the closest cherry in comparison to the agent, the second coordinate describes the relative position of the closest ghost in comparison to the agent.

c) What are the advantages and disadvantages of each state encoding.

Current State



Encoding a)

$$\begin{pmatrix} 0,0,2 \\ 3,1,0 \\ 0,0,0 \end{pmatrix}$$

Encoding b)

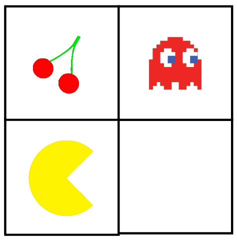
$[(1,1),(0,1),(2,2)]$

Encoding c)

$[(-1,0),(1,1)]$

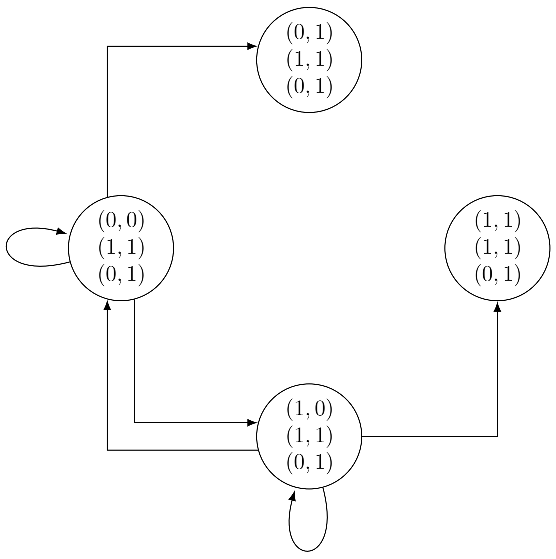
Exercise 3 Markov Decision Processes (MDP)

a) Complete the table with probabilities for the finite MDP of the Pac-Man example using the state representation (Pac-Man's pos., ghost's pos., item's pos.) and a map of size 2x2. Assume that the ghost is moving in each direction with the same probability. The game only ends if you collide with the ghost or collect the cherry.



s	a	s'	$p(s' \mid s, a)$
(0,0); (1,1); (0,1)	Right	(1,0); (0,1); (0,1)	
(0,0); (1,0); (0,1)	Up	(0,1); (1,1); (0,1)	
(0,0); (1,0); (0,1)	Up	(0,1); (0,0); (0,1)	
(0,0); (1,0); (0,1)	Up	(0,1); (1,0); (0,1)	

b) Complete the transition graph to the right assuming that the ghost is not moving. Why are some transitions missing? (Don't forget to add the action nodes to the edges)



c) How would the graph change if the ghost would be able to move?

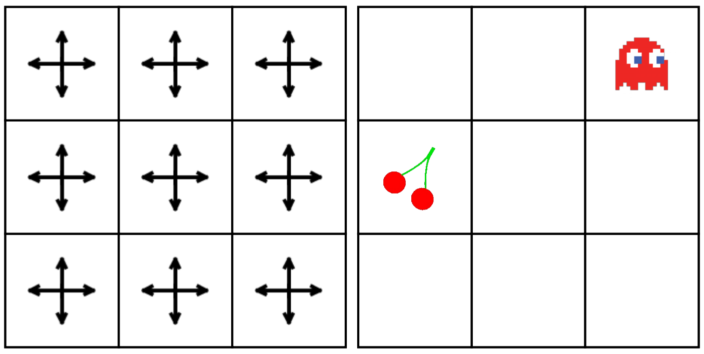
Exercise 4 Iterative Policy Evaluation

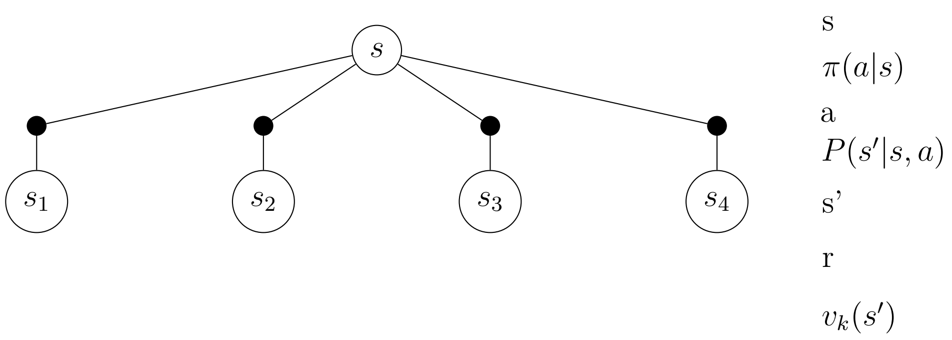
Consider the following simplified Pac-Man game:

- The game continues as long as Pac-Man did not collect the single cherry.
- Pac-Man can move in the four cardinal directions: up, down, left, right
- State transitions and rewards:
  - reaching an empty cell yields a reward of 0
  - colliding with a ghost yields a reward of -9
  - collecting the item yields a reward of +100 and ends the game
  - any other action that takes the agent off the grid, leaves the state unchanged and gives a reward of -1
- This is an undiscounted task: $\gamma = 1$
- For simplicity the ghost is not moving.

a) Complete the values for the value function for k = 1 and complete the given backup diagram for a Pac Man following the equi-probable random policy (all actions equally likely), for all states $\pi(a|s) = 1/4$

b) What would the optimal policy look like? (Either repeat the value function calculation two more times or use a computer program to calculate the necessary state values.)





The following task can be solved in pairs of two. Please make sure that your solution includes the name of both group members as a comment at the top of the file

Exercise 5 Programming Exercise - Update Rules for Pokemon EGT

Pokemon EGT is an implementation of the Hawk-dove game based on Pokemon with a structured population. All game dynamics except update rules are implemented and explained in the colaboratory.

Task

- Implement the update rule Birth-Death
- Implement the update rule Death-Birth
- Implement the update rule Imitation
- Which other variants of update rules are possible ? Try to find two other rules and explain how they work.

Joining the Google Colab project

- Introduction to Google Colab:
  `https://colab.research.google.com/notebooks/welcome.ipynb`
- Join the Google Colab project:
  `https://colab.research.google.com/drive/1NxweyKFowbrPHuOvgByLN_PWeobKOu46`

Accessing all files in Github

- Link to files on GitHub
  `https://github.com/christianwustrau/pokemon_egt`

Included files:

- *setup.py* – defines the Pokemon class, all necessary variables, init and plot functions
- *update.py* – defines the update rule for every round
- *game.py* – defines the main game functions for fighting