Universidad de La Sabana
Facultad de Ingenería

Felix Mohr & Sergio Amortegui
2025.2

Fundamentos de Inteligencia Artificial

- Sheet 1-

Exercise 1 (10 points) The environment (S, E, ψ) of the connect-4 game can be formally described as follows:

- S. The fields form a set of combinations of row position and column position, i.e., $F = \{1, .., 6\} \times \{1, .., 7\}$. A state is simply an assignment (function) of tile information (red = -1, yellow = 1, no tile = 0) to each field, i.e., $s : F \to \{-1, 0, 1\}$. The *set* of states $S = \{s \mid s : F \to \{-1, 0, 1\}\}$ is simply the set of such assignments.
- E. An event is the column where a tile might be placed, so $E = \{1, .., 7\}$.
- $\psi(s, e)$. Write the state s as a set of tuples[1] $\{(f_1, v_1), .., (f_n, v_n)\}$ where $f_i \in F$ are fields and $v_i \in \{-1, 0, 1\}$.

  First, let $t : S \times E \to \{0, 1, .., 6\}$ measure the number of tiles that are already located in the column in a state, and $t(s, e) = 0$ if there are no tiles yet in column e in state s.

  Next, let $p(s) = -1$ if the number of total tiles in s, i.e. $\sum_{e=1}^{7} t(s, e)$, is even (including 0); otherwise $p(s) = 1$. $p(s)$ indicates the player whose tile will be placed next.

  $\psi(s, e)$ is only defined if $t(s, e) < 6$; otherwise no more tile can be placed in column e.

  If $t(s, e) < 6$, then define the field $f_e := (e, t(s, e) + 1)$ whose value is going to change from 0 to the color of the current player, i.e., $p(s)$. Formally: $\psi(s, e) := s \setminus \{(f_e, 0)\} \cup \{(f_e, p(s))\}$; so, technically, one removes the old assignment for the field f_e (no color) and adds the new assignment with the new color (in Python one would simply overwrite a variable in a matrix).

Your task is now to complement the Python class `ConnectState` at the indicated positions (check the documentation in the class to see what the functions do) and to upload this file to gradescope.

The most difficult function in the class is `get_winner`, because here you must check the winning conditions of a player.

Hint 1: You are absolutely allowed to add further (*optional*) parameters to the any function, but these should not be *necessary* when calling from the outside.

Hint 2: Use the `show` function to visualize a board; this is helpful for debugging.

Hint 3: Maybe write yourself some unit tests in which you manually create some boards with a red/yellow winner and check that all your functions work properly.

Hint 4: It is possible to elegantly solve this task in less than 150 lines of clean and easily understandable code. By no means should you need more than 300.

[1] A function $f : X \to Y$ can be written as a set of pairs $\{(x_1, y_1), (x_2, y_2), .. \mid x_i \in X, y_i = f(x_i)\}$. For example $f(x) = x^2$ on $X = \{-2, -1, 0, 1, 2\}$ is $\{(-2, 4), (-1, 1), (0, 0), (1, 1), (2, 4)\}$

Exercise 2 (Optional Practice Exercise) This is an exercise that allows you to practice how to model an environment. You are invited to coordinate an in-person (group) meeting with your lecturer to discuss your solution.

Consider the Tic, Tac, Toe game (check reference if unknown).

1. Characterize the environment of the game (observability, determinism, number of agents, types of agents, etc.)
2. Give a formal description of
   a) the state space of the game
   b) the possible events
   c) the state transition function

   Remember that a formal description of a system is not a text. It consists of the definition of base sets (explicit or implicit), complex sets (cartesian products), and functions; text may *help* explain the formal elements.

Keep in mind that the environment is an objective description of the system. It is *not* formulated from the perspective of one of the players.

Look at the first exercise to see an example of a formalization for the connect-4 game and use this as an inspiration.