Paola Noun

Project Writeup

GamesCrafters sept. 2020

# Toads and Frogs

Toads and Frogs is a two-player game that was invented by British mathematician Richard Guy.

## I. The Rules.

The game is played on a single row and n column board (i.e. 1xn). At any time during the game, an empty square can be occupied by a single toad or frog.

The conventional way to start with, however, is with our toads lined up on the left most side of our strip and our frogs to the right. As seen in the figure below:

When it is your turn (whether you are a toad or a frog), you can:

- Move to an empty square that is directly after to you (meaning there is no moving backwards).
- Hop over a square containing the opposite player if the one directly after it is empty.

Some things to keep in mind:

- You cannot hop over an empty square.
- You cannot hop over a square occupied by yourself (i.e. toad if you are player R or frog if you are player L).

Following convention, the first player unable to move on his turn loses. That is our primitive loss.

## II. My Implementation.

### Position Representation:

The positions are represented by tuples, which contain two string values: (“R---L”, “R”).

The above representation stores the current board representation at index 0 and whose turn it currently is at index 1.

Generating Moves:

- The generate_moves(position) function utilizes a helper called rules(player, list).
- It returns the output of rules called with inputs:
  - Who the current player is (position[1])
  - The current state of the board as a list (list(position[0])).
- Rules:
  - This function does the bulk of the work in terms of gameplay
  - It starts by initializing an empty positions list that will be returned by the function:
    - If the list remains empty, there are no possible moves left
  - There are basically two cases to account for:
    - If the player is a toad:
      - We iterate through the list from left to right
    - If the player is a frog:
      - We iterate through the list from right to left
    - In both of the cases we must check whether:
      - We can move to the next empty space
      - We can hop over our opponent
    - All possible scenarios are added to the positions list as tuples.

Doing Moves:

The do_move(position, move) function will return the new position after the move is executed. Since move itself is basically the new position, we just set new_position to move.

Primitive Values:

It was previously mentioned that the first player unable to move loses. Therefore, if the positions list from generate_moves is empty we can return lose. Otherwise the outcome is still unknown, we are not yet at the bottom of our game tree.

In the second case, I decided to utilize the deque data structure from collections. That way the current possible moves can be stored temporarily so that generate_moves only needs to be called once on a current position. We therefore append the current

possible moves to the temp_moves deque, that way, in our solver, we can just use pop() instead of calling generate_moves again.

## III. Some Helpful Links.

Wikipedia:
https://en.wikipedia.org/wiki/Toads_and_Frogs#:~:text=The%20combinatorial%20game%20Toads%20and%20Frogs%20is%20a%20partisan%20game,Ways%20for%20your%20Mathematical%20Plays.&text=A%20one%2Dplayer%20puzzle%20version,game%20has%20also%20been%20considered.

Wolfram Alpha Demonstration:
https://demonstrations.wolfram.com/TheGameOfToadsAndFrogs/