PAC MAN as Markov Decision Process.

In this document, we formulate Pac-Man game as a MDP.

Assumptions/Rules of PAC MAN:

- When a ghost first enters the board from the center box it picks a direction at random and moves in that direction.
- A ghost only changes directions when it reaches a wall.
- When a ghost reaches a wall, it picks randomly among the possible directions and begins moving in the randomly selected direction.
- We do not consider fruits in game for simplicity.
- We do not consider number of lives left for Pac man for simplicity
- Consider ghosts stay in edible states for some fixed(constant) time steps after Pac man eats power pellet.

MDP Formulation:

An MDP is defined by set of **{S, A, T, R}** where

S: State Space i.e. set of all possible states ‘s’.

A: Action Space i.e. set of all possible states ‘a’.

R: Reward for being in a state

T: Transition dynamics/ Probability transition Matrix.

Action Space in PAC-MAN MDP:

At any given situation action of Pac-Man is choosing the direction of its movement, so

Action Space ‘**A**’ is **{Left, Right, Up, Down}**

State Space in PAC-MAN MDP:

The main objective of state space and state ‘s’ at any given time ‘t’ is to represent all the relevant information which helps in decision making i.e. solving the MDP.

‘s_t’ represents the state vector i.e. state of system at time ‘t’.

And ‘s_t’ is defined by representing the following information.

States can be decomposed w.r.t pac man, ghosts, maze content and its used in Transition dynamics.

- Location of dots left in the Maze
- Current direction(d_p) and location (x_p,y_p) of Pac man in the maze and it’s distance in all 4 directions with respect to
    - Distance to walls, corners
    - distance to Power Pellets (each of them).
    - Distance(shortest) w.r.t each ghost
    - Distance to nearest dot.
- Number of ghosts in the center box of the maze
- Current location, direction of each of ghosts in the maze
    - Their state i.e. are they edible or not
    - Counter value for reset if ghosts are in edible state
    - Distance of each ghost from power pellets
- Number of Power pallets left in the maze.
- Indicator states to illustrate possible legal actions given the maze & Pac man’s location
- A state to indicate termination.

So the state vector can be defined as

[x_p, y_p , d_p, x_{pwr}, y_{pwr}, x_{pwl}, y_{pwl}, x_{pwu}, y_{pwu}, x_{pwd}, y_{pwd}, x_{pcr}, y_{pcr}, x_{pcl}, y_{pcl}, x_{pcu}, y_{pcu}, x_{pcd}, y_{pcd}, $x_{p\_pp\_1}$,$y_{p\_pp\_2}$,…, $x_{p\_pp\_k}$,$y_{p\_pp\_k}$, xd_{p_g1} ,yd_{p_g1}, xd_{p_g2} ,yd_{p_g2}, xd_{p_g3} ,yd_{p_g3}, xd_{p_g4} ,yd_{p_g4},x_{p_d},y_{p_d},n_g_{cb}, x_{g1}, y_{g1} ,d_{g1} ,e_{g1},x_{g2}, y_{g2} ,d_{g2}, d_{g2} ,x_{g3}, d_{g3}, e_{g3} ,y_{g3} , x_{g4}, y_{g4} , d_{g4}, e_{g4} ,n_p, I_l, I_r, I_u, I_d, Counter,Terminate, Location of dots in maze].

Where

- (x_{pwr}, y_{pwr}) indicate location of wall on the right side of pac man’s current location and similarly for left, up, down; in notation ‘p’ stands for pac man, ‘w’ stands for wall, ‘r’ for right direction, similarly for other directions.
- (x_{pcr}, y_{pcr}) indicate location of corner on the right side of pac man’s current location and similarly for left, up, down; in notation ‘p’ stands for

pac man, ‘c’ stands for corner, ‘r’ for right direction, similarly for other directions.

- ($x_{p\_pp\_1}$,$y_{p\_pp\_2}$)indicate distance of pac man from power pellet where p stands for pacman, pp for power pellet, 1 for power pellet index to indicate distance to multiple power pellets.
- (xd_{p_g1} ,yd_{p_g1}) Indicate the pac man distance w.r.t each ghost in x, y axis.
- (x_{p_d},y_{p_d}) indicate the relative position/distance for pac man w.r.t nearest dot.
- n_g_{cb} indicate the number of ghosts in the center box.
- , x_{g1}, y_{g1} ,d_{g1} ,e_{g1} indicate ghost states location, direction and whether a ghost is edible or not.
- n_p indicate number of power pellets left
- Counter indicate the number of time steps left until ghosts are edible.
- I_l, I_r, I_u, I_d indicate legal actions possible at pac man’s location.
- Terminate is a state which indicates reset when ghost eats pacman

Reward Function:

Consider score of the game which updates after every state transition or every time step, and score is the cumulative reward accumulated from the beginning of the game.

So, Δscore is the reward obtained for taking an action at state s^0 and when we transit to state s^1.

Δscore for any s^t to s^{t+1} is

$$R(s^t) = \Delta\text{score} = \begin{cases} +1 & \text{if Pac man eats dot} \\ +100 & \text{if Pac man eats a ghost} \\ +1000 & \text{if maze is cleared} \\ +0 & \text{otherwise.} \\ -100000 & \text{if ghost eats Pac man} \end{cases}$$

Transition dynamics:

Transition dynamics define the rules of environment or game.

Let's update state vector conditioned on action i.e.

- Dots location: If Pac man eats a dot then remove that dot location from state vector otherwise dots location states remain same.
- Power Pellets: If Pac man eats a power then remove that pellet's location from state vector otherwise power pellets location states remain same.(n_p,x_{p_pp},y_{p_pp})
- Ghosts States (for each of the ghost):
  - If Ghost eats Pac man: terminate game/reset.
  - Counter:
    - If Pac Man eats a power pellet Initialize/ reinitialize counter
    - Else if counter value is non-zero decrement it by one
  - Edible State:
    - If Counter value (after update) ≠ 0 then ghost is edible i.e. e_g=1
    - If counter value is 0 (after update), ghost is not edible i.e. e_g=0.
  - Location:
    - If Pac Man eats ghost, ghost location is reset to center box i.e. [(x_g, y_g) = (0,0)] and number of ghosts in center box is also updated i.e. n_g_{cb}++.
    - Else if ghost bumps into a wall chose one of possible directions randomly and update (x_g, y_g) in that direction.
    - Else if ghost is at the center it chooses a direction randomly and then update (x_g, y_g) in chosen direction.
    - Else (regular ghost state update) update (x_g, y_g) in its current direction.
- Pac Man States: Once ghost states are updated,
  - Update Pac man's current position and direction after taking an action.(x_p, y_p , d_p)
  - Update Distance w.r.t each ghost states, Power pellets of Pac man
  - Also update distance w.r.t walls, corners in all directions w.r.t Pac man's current position.
  - Update possible actions at updated pac man's position.(I_l, I_r, I_u, I_d)