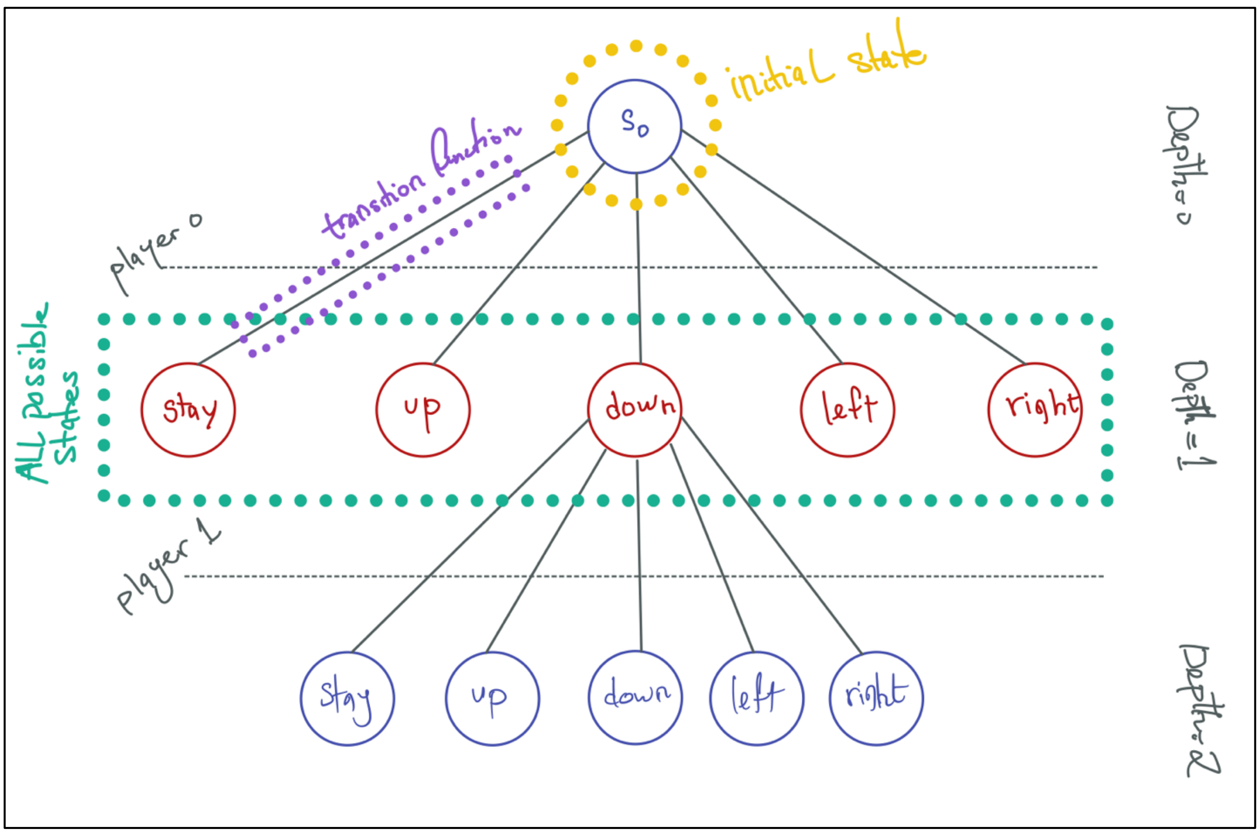
Q1 Describe the possible states, initial state, transition function of the KTH fishing derby game.

- **Possible states S** -> A list with all possible states that the game could be in. For an example: one state could be green boat at position X1 with hook at position Y1, red boat at position X2 with hook at position Y2
- **Initial game state s0** -> A state in S which describes the very first state the game begins in when it has been started
- **Transition function** -> A function that takes actions bound by rules and returns or calculates a new state. For an example: an equation that calculates the upcoming state [player new scores, new hook and fish locations, any fish has been caught, is game over] given the choice of movement the player picked, and locations of the fish based on their movement sequence.



Q2 **Describe the terminal states of the KTH fishing derby game.**

- A state from S is a terminal state s_t when $\mu(p,s_t)=[]$. This can occur when there are no possible, legal movements for the current player available and the game ends.
- In our algorithm: a terminal state is when the end of the observations has been reached [fish movement sequence], the maximum depth allowed has been reached, or the elapsed time allowed for the turn has been reached

Q3 **Why is v(A,s)=Score(Green boat)−Score(Red boat) a good heuristic function for the KTH fishing derby (knowing that A plays the green boat and B plays the red boat)?**

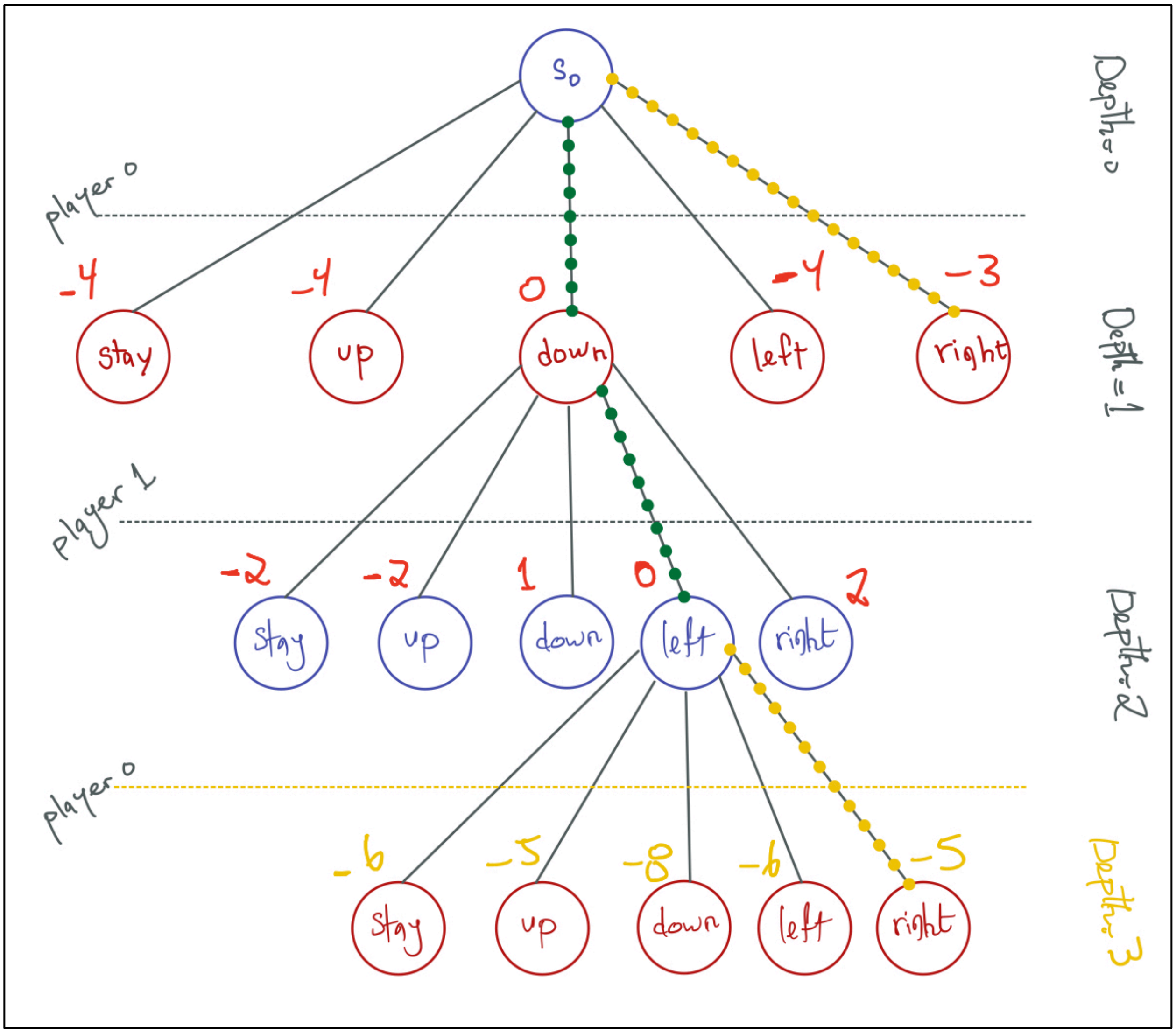
- It is very simple to compute
- Represents a very good utility for the boat agents, as the main indicator of good performance is a higher score in the game
- It fulfills the zero-sum game rule. which means $\gamma(A,s)+\gamma(B,s) = 0$ or equivalently $\gamma(A,s) = -\gamma(B,s)$

Q4 **When does v best approximate the utility function, and why?**

- A heuristic best approximates the utility when it can choose the best solution given that there exists several solutions, within a reasonable amount of time
- Some heuristics can't give a reasonable solution unless we have gone down the game tree very deep, which could be a resource hog [computation power, and time]
- Best Heuristic is the one that can give a reasonable solution without having to go deep down the game tree

Q5 **Can you provide an example of a state s where v(A,s)>0 and B wins in the following turn? (Hint: recall that fish from different types yield different scores).**

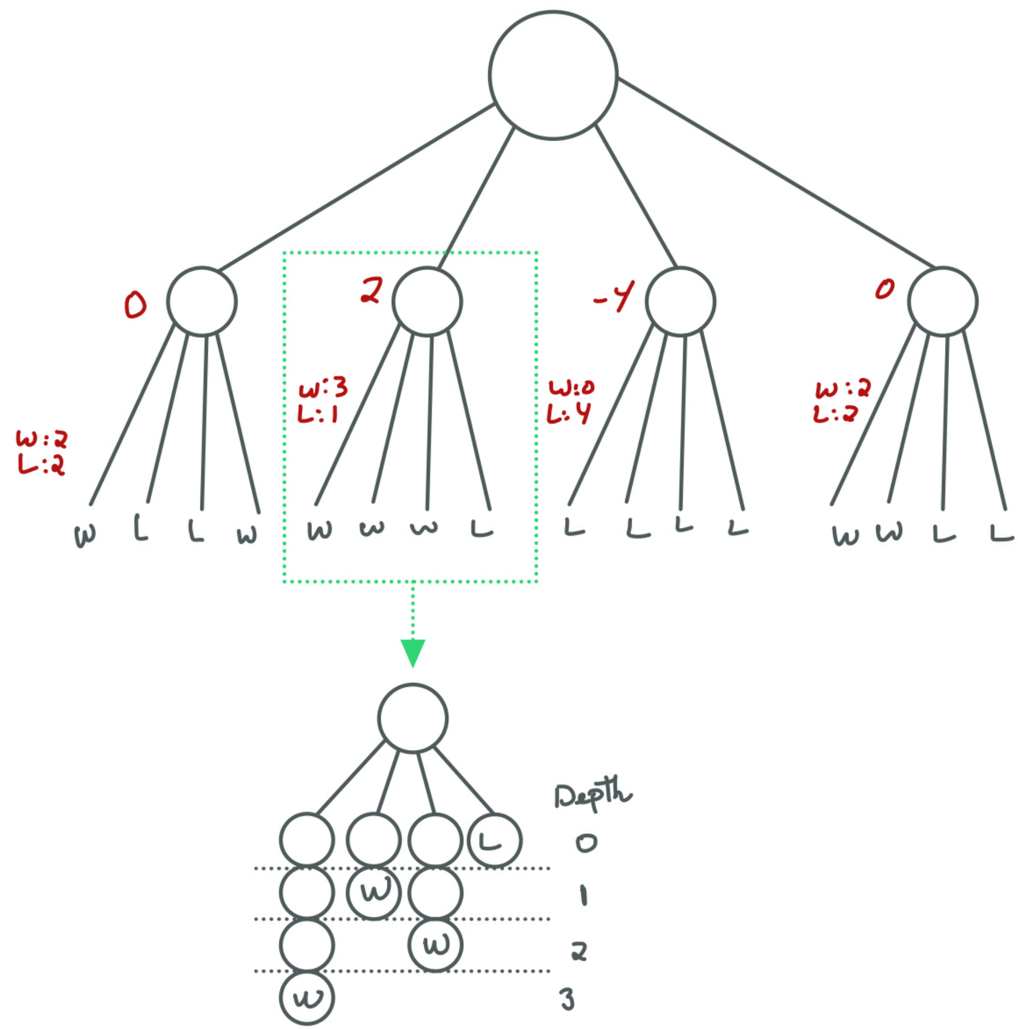
- First example: if my heuristic is the difference between scores, then it could be due to not calculating deep enough into the game tree to figure out that the current movement where v(A,s)>0 is not a good one because the next movement would cause it to be v(A,s)<0



- Second example: a bad heuristic where it doesn't properly approximate the utility function. e.g., a heuristic that calculates the distance difference to fish for the two opponents without keeping the score of fish in mind

Q6 **Will η suffer from the same problem (referred to in Q5) as the evaluation function v? If so, can you provide with an example? (Hint: note how such a heuristic could be problematic in the game of chess, as shown in Figure 2.2).**

- Yes. for an example a $\eta(A,s)=|w(s,A)|-|l(s,A)|$ = 5 - 1 = 4. in this example $\eta(A,s) > 0$, however, the lose state could be reached literally the next movement compared to the other 5 wins, which may require more than one movement. η does not consider how likely those wins or loses may occur.



- consider for instance $\eta(A,s) = 10$ for a given state s. This just means that there are 10 more reachable terminal states that result in a victory for player A than for player B, however it tells nothing of the distribution of such states, and how much B can do to counteract those moves