



UNIVERSIDAD DE LOS LAGOS

INGENIERÍA CIVIL EN INFORMÁTICA

Parcial 1:

Juego de Ranitas

“IA aplicada a un juego”

Juan Carlos Barría Rival

Álvaro Marcelo Cárdenas Ritter

Felipe Ignacio Maldonado Gómez

Patricio Alejandro Trujillo Arias

16 de Noviembre del 2020

Juego de Ranitas

Es un juego basado en la tradición China que es una caja de madera con siete agujeros, existen dos grupos de 3 bolas de madera de diferente color que se ubican en cada agujero, solamente queda un agujero libre. Existe esta nueva reinterpretación que cambia la caja por siete piedras sobre un río, y a las bolas en ranas.



(a) Estado Inicial



(b) Estado Final

El juego consiste en cambiar la posición de las ranas de un lado al otro. Cada rana puede saltar a la piedra vacía si está a lado de ella (salto simple), o si hay una rana delante de ella, saltando sobre ella (salto doble). Cada grupo de ranas tiene su sentido, por lo que el salto solamente se puede ejecutar en aquella dirección.



Se Pide:

- Defina una representación abstracta del juego, considerando sus movimientos posibles y las restricciones aplicables al código.
- Escriba al menos cinco niveles en el árbol de estados de acuerdo a la representación y movimientos definidos.
- Implemente un código en Python que permita encontrar una solución al juego a través de un algoritmo de Búsqueda No Informada.
- Implemente un código en Python que permita encontrar una solución al juego a través de un algoritmo de Búsqueda Informada.
- Una vez terminados ambos códigos, realice una serie de experimentos para ver el comportamiento de cada algoritmo.

1. Representación

El espacio lo representamos como un arreglo unidimensional [0:6] donde cada una de las ranas es representada por un número en este caso específico utilizamos los valores 1 y 2 para representar las ranas dependiendo su color (1 para las verdes y 2 para las cafés) y por otro lado el espacio vacío se representa con el número 0.

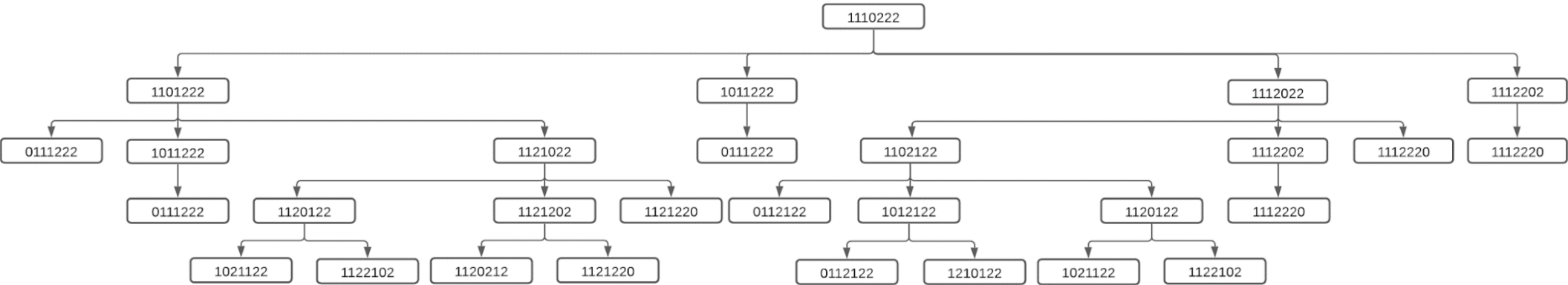
Estado inicial = [1, 1, 1, 0, 2, 2, 2]

Estado final = [2, 2, 2, 0, 1, 1, 1]

2. Movimientos y Restricciones

- Las ranas solo se pueden mover al espacio vacío.
- Las ranas tienen 2 tipos de saltos, el salto simple el cual consiste en avanzar directamente al espacio vacío que se encuentra de manera consecutiva a él, y el salto doble el cual consiste en saltar un espacio que se encuentra ocupado por otra rana.
- Las ranas de un mismo color o en este caso el mismo número solo tienen una dirección de movimiento esto nos indica que no pueden volver a la posición anterior.
- Las ranas sólo se podrán mover dentro del arreglo, esto indica que si la rana quiere moverse hacia una dirección y no le quedan espacios disponibles para ejecutar algunos de los 2 tipos de saltos no podrá ejecutar ningún movimiento.

3. Árbol de Estados



4. Tabla Comparativa

	Algoritmo Anchura	Algoritmo Beam
Cantidad de Iteraciones	135	127
Historial de estados	140	140
Niveles	16	18

Se puede inferir, a través de la tabla, que el Algoritmo de anchura realizó una cantidad de 135 iteraciones quedándole por revisar 5 estados, en cambio el Algoritmo Beam realizó 127 iteraciones faltándole revisar 13 estados. Hay que considerar que el algoritmo de anchura llegó a la solución en 16 niveles (15 movimientos), mientras que el Algoritmo Beam lo realizó en 18 niveles (17 movimientos).