

Bài tập thực hành Tuần 2

Môn Trí tuệ nhân tạo

Bài tập thực hành của tuần này là bài tập 3.9 (câu a & b) trang 119 trong cuốn Artificial Intelligence.

*“3.9 The **missionaries and cannibals** problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side, without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in **AI** because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968).*

***a.** Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.*

***b.** Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?”*

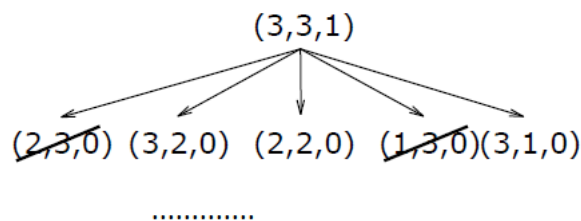
Hướng dẫn

Gọi (a, b, k) với $0 \leq a, b \leq 3$, trong đó a là số người, b là số con quỉ ở bên bờ tả ngạn vào các thời điểm mà thuyền ở bờ này hoặc bờ kia, $k=1$ nếu thuyền ở bờ tả ngạn và $k = 0$ nếu thuyền ở bờ hữu ngạn. Khi đó không gian trạng thái của bài toán được xác định như sau:

- Trạng thái ban đầu là $(3, 3, 1)$.
- Thuyền chở qua sông 1 người, hoặc 1 con quỉ, hoặc 2 người, hoặc 2 con quỉ, hoặc 1 người và 1 con quỉ.
- Trạng thái kết thúc là $(0, 0, 0)$.

Nên

- a) Vẽ sơ đồ liệt kê tất cả các khả năng có thể để tìm lời giải của bài toán (search tree).



- b) Sử dụng một thuật toán tìm kiếm để đưa ra lời giải tối ưu cho bài toán.

- Xây dựng Node:

```

class Node:
    goal_state=[0,0,0]
    num_of_instances=0
    def __init__(self,state,parent,action,depth):
        self.parent=parent
        self.state=state
        self.action=action
        self.depth=depth

        Node.num_of_instances+=1

    def __str__(self):
        return str(self.state)

    def goal_test(self):
        if self.state == self.goal_state:
            return True
        return False

    def is_valid(self):
        missionaries = self.state[0]
        cannibals = self.state[1]
        boat = self.state[2]
        if missionaries < 0 or missionaries > 3:
            return False
        if cannibals < 0 or cannibals > 3:
            return False
        if boat > 1 or boat < 0:
            return False
        return True

    def is_killed(self):
        missionaries = self.state[0]
        cannibals = self.state[1]
        if missionaries < cannibals and missionaries > 0:
            return True
        # Check for the other side
        if missionaries > cannibals and missionaries < 3 :
            return True

    def generate_child(self):
        children=[]
        depth = self.depth + 1
        op = -1 # Subtract
        boat_move = "from left shore to right"
        if self.state[2] == 0:
            op = 1 # Add
            boat_move = "from right shore to left"

        for x in range(3):
            for y in range(3):
                # by_move = "Move %s missionaries and %s cannibals %s" % (x, y, boat_move)
                new_state=self.state.copy()
                new_state[0],new_state[1],new_state[2]=new_state[0]+ op * x, new_state[1]+ op * y, new_state[2] + op * 1
                action=[x,y,op]
                new_node=Node(new_state, self, action, depth)
                if x + y >= 1 and x + y <= 2 :
                    children.append(new_node)
        return children

    def find_solution(self):
        solution = []
        solution.append(self.action)
        path = self
        while path.parent != None:
            path = path.parent
            solution.append(path.action)
        solution = solution[::-1]
        solution.reverse()
        return solution

```

- Xây dựng graph từ câu a
- Sử dụng 1 thuật toán tìm kiếm (BFS, DFS, UCS)