

# CT332: TRÍ TUỆ NHÂN TẠO

## BÀI THỰC HÀNH SỐ 5

### ÔN TẬP

#### Phần 1: Tìm kiếm

Một robot cần tìm đường đi ra khỏi mê cung với vị trí bắt đầu và kết thúc của mê cung sẽ được cho trước. Mê cung được biểu diễn bằng một ma trận kích thước  $m \times n$ , trong đó giá trị tại mỗi ô là 0 hoặc 1 với giá trị 0 là tường của mê cung, không thể đi được; vị trí biểu diễn bởi số 1 là vị trí có thể đi được trong mê cung. Tại một vị trí bất kỳ trong mê cung, robot có thể di chuyển lên, xuống, sang trái, sang phải sao cho các vị trí này là hợp lệ (các ô này phải có giá trị là 1). Cần xác định đường đi của robot từ vị trí bắt đầu đến điểm cuối trong mê cung.

Để giải quyết bài toán trên, cần cung cấp dữ liệu đầu vào bao gồm: thông tin mê cung, trạng thái đầu, trạng thái cuối. Dữ liệu này có thể truyền trực tiếp thông qua khai báo như sau (hoặc có thể đọc từ file txt):

```
16 const int COLS = 10;
17 const int ROWS = 9;
18
19
20 // Truong hop me cung co 5 dong va 5 cot
21
22 const int maze[][COLS] = { {1, 1, 0, 1, 1},
23                             { 0, 1, 0, 1, 0},
24                             { 1, 1, 1, 1, 0},
25                             { 0, 1, 0, 1, 0},
26                             { 1, 0, 1, 1, 1}};
```

Mỗi trạng thái của mê cung sẽ được khai báo theo cấu trúc State:

```
80 struct State {
81     int robot_x;           // Toa do x cua con robot
82     int robot_y;           // Toa do y cua con robot
83 };
```

Robot chỉ có thể di chuyển lên, xuống, trái, phải sang các ô có giá trị là 1, vì vậy cần kiểm tra tính hợp lệ của các ô này thông qua hàm `check_position` được cho bên dưới:

```
102 // Dinh nghia ham kiem tra toa do x,y co hop le khong
103 bool check_position(int x, int y) {
104     if(x < 0 || y < 0)
105         return false;
106     if(x >= ROWS || y >= COLS)
107         return false;
108     if(maze[x][y] == 0) // o khong di duoc
109         return false;
110     return true;
111 }
```

Hàm di chuyển lên của robot được cho như sau:

```

115 bool up(State current_state, State *new_state) {
116
117     int x = current_state.robot_x-1;
118     int y = current_state.robot_y ;
119
120     if(check_position(x,y)) {
121         new_state->robot_x = x;
122         new_state->robot_y = y;
123         return true;
124     } else {
125         return false;
126     }
127 }

```

Hàm call\_operator sẽ lần lượt gọi các phép toán theo thứ tự di chuyển lên, xuống, trái, phải:

```

176 bool call_operator(State s, State *out, int option) {
177     switch (option) {
178         case 1:
179             return up(s, out);
180         case 2:
181             return down(s, out);
182         case 3:
183             return left(s, out);
184         case 4:
185             return right(s, out);
186         default:
187             return false;
188     }
189 }

```

Cấu trúc một node được cho như sau:

```

200 typedef struct Node{
201     State state;
202     struct Node* parent;
203     int no_function; //hành động sinh ra trạng thái hiện tại
204 } Node;

```

Chuỗi các hành động lần lượt là:

```

29 // Hàng chuỗi để in ra tên các hành động
30 const char* action[] = {"Trạng thái bắt đầu",
31                          "UP",
32                          "DOWN",
33                          "LEFT",
34                          "RIGHT"};

```

**Câu hỏi:** Dựa vào bài thực hành số 1, hãy viết các hàm cần thiết để thực hiện các công việc sau (test case được cho bên dưới):

- Cài đặt hàm DOWN, LEFT, RIGHT và in các trạng thái con có thể có của một trạng thái hiện tại
- Sử dụng **giải thuật BFS**, in thứ tự duyệt các trạng thái, tổng số trạng thái đã duyệt và chuỗi các hành động cần thực hiện để tìm ra trạng thái cuối

Kiểm tra với test case sau:

```

1 1 0 1 1
0 1 0 1 0
1 1 1 1 0
0 1 0 1 0
1 0 1 1 1

```

Trạng thái đầu: (0, 0)

Trạng thái cuối: (4, 4), (0,4), (1,4)

**Lưu ý: các câu 1a, 1b lưu vào các file riêng để tiện khi chạy chương trình**

## **Phần 2: Prolog**

Cho cơ sở tri thức gồm các phát biểu như sau:

- a. Nếu X là họ hàng của Y và Y là họ hàng của Z thì X biết Z
- b. Nếu X1 là dì của Y1 thì X1 là họ hàng của Y1
- c. Nếu X2 là cô của Y2 thì X2 là họ hàng của Y2
- d. Lan là dì của Hồng
- e. Hồng là dì của Mai
- f. Hồng là cô của Phương

## **Câu hỏi:**

Sử dụng Prolog biểu diễn tri thức đã cho, và thực hiện các truy vấn sau:

- 1. Những người nào quen biết nhau?
- 2. Giả sử Oanh là chị của Lan, vậy có thể kết luận Oanh biết Hồng không?