

Assignment 04 (Due: Friday, December 6, 2019)

CSCE 322

Contents

1	Instructions	1
1.1	Data File Specification	1
1.2	colsAndPlayers(Maze)	2
1.3	fewestSlides(Maze,Slides)	2
1.4	goalWall(Maze)	2
1.5	noAdvantage(Maze) (15% Extra Credit)	3
2	Compilation & Execution	3
3	Webgrader Warning	3
4	Naming Conventions	3
5	Point Allocation	4
6	External Resources	4

List of Figures

1 Instructions

This assignment will use Prolog to extract certain information about the state of a **Slippery Crossings** game.

1.1 Data File Specification

The maze will be represented as a list of lists of symbols.

```

[l,u,u,u,d,r,r,u,u,l,r].
[x,x,x,x,x,x,x,x].
[x,-,-,-,1,x,x].
[x,x,-,x,x,-,-,x].
[x,-,-,2,-,x,-,x].
[x,-,-,-,x,-,-,x].
[x,x,-,-,x,-,-,x].
[x,x,-,-,-,x,x,x].
[x,-,x,x,-,-,3,x].
[x,-,x,-,-,-,-,x].
[x,g,-,-,-,-,-,x].
[x,x,x,x,x,x,x,x].

```

1.2 colsAndPlayers(Maze)

The query `colsAndPlayers(Maze)` will be successful when the maze has an odd number of columns and an odd number of players in the maze, or an even number of columns and an even number of players in the maze.

```

maze
[x,x,x,x,x,x,x,x]
[x,-,-,-,1,x,x]
[x,x,-,x,x,-,-,x]
[x,-,-,2,-,x,-,x]
[x,-,-,-,x,-,-,x]
[x,x,-,-,x,-,-,x]
[x,x,-,-,-,x,x,x]
[x,-,x,x,-,-,3,x]
[x,-,x,-,-,-,-,x]
[x,g,-,-,-,-,-,x]
[x,x,x,x,x,x,x,x]
false

```

1.3 fewestSlides(Maze,Slides)

The query `fewestSlides(Maze,Slides)` will be successful when `Slides` is unified with the shortest valid path (fewest number of slides) from the current state of the maze to a solved maze for Player 1. There may be multiple shortest paths, and `fewestSlides(Maze,Slides)` should report them all, if `;` appears in the query. You may assume that maze that requires more than 10 slides to solve is unsolvable and that the other players will not move. You may assume that no valid path will exceed 10 slides.

```

[x,x,x,x,x,x,x,x]
[x,x,-,-,x,-,x,x]
[x,-,-,-,x,-,-,x]
[x,-,-,-,4,-,3,x]
[x,1,x,-,-,x,x,x]
[x,-,-,-,-,x,x,x]
[x,x,-,x,-,x,x,x]
[x,-,-,x,-,-,2,x]
[x,x,-,x,-,-,-,x]
[x,x,-,-,g,-,-,x]
[x,x,x,x,x,x,x,x]
moves
[d,r,d]
true

```

1.4 goalWall(Maze)

The query `goal(Maze)` will be successful when the maze contains a goal that is adjacent to a wall (by a move of up, down, left, or right).

```

[x,x,x,x,x,x,x,x,x,x]
[x,-,-,-,-,x,2,-,x,x,x]
[x,-,-,g,-,1,-,-,-,x,x]
[x,-,-,x,-,-,-,-,x,-,x]
[x,-,-,x,-,-,-,x,x,-,x]
[x,-,-,-,4,x,-,-,-,-,x]
[x,-,-,-,-,-,-,-,3,x]
[x,x,-,-,x,x,-,x,x,-,x]
[x,x,x,x,x,x,x,x,x,x]

```

1.5 noAdvantage(Maze) (15% Extra Credit)

The query `noAdvantage(Maze)` will be successful when the maze contains a player that has a shorter path to the goal than Player 1.

```

[x,x,x,x,x,x,x,x,x,x]
[x,x,x,-,-,-,-,-,-,x]
[x,-,-,g,-,x,-,-,-,-,x]
[x,-,-,-,-,x,-,x,-,-,x]
[x,-,-,-,1,-,-,x,-,x,x]
[x,x,-,x,x,-,x,-,-,-,x]
[x,x,-,-,-,-,-,-,-,2,x]
[x,-,-,-,x,x,-,-,-,-,x]
[x,-,-,-,-,-,-,-,-,-,x]
[x,-,-,x,-,-,-,-,x,-,x]
[x,x,x,x,x,x,x,x,x,x]

```

2 Compilation & Execution

Your program will be tested on `cse.unl.edu`, using `pl. a04tests.pl` will include test cases for testing your program. You can run the test cases with the commands:

```
[a04tests].  
loadHelpers.  
part01.
```

3 Webgrader Warning

Prolog programs can take some time to run. The webgrader is limiting individual processes to 5 minutes.

4 Naming Conventions

You will be submitting at least 3 `.pl` files. The filenames should be `csce322a04part01.pl`, `csce322a04part02.pl`, and `csce322a04part03.pl`

5 Point Allocation

Component	Points
<code>csce322a04part01.pl</code>	30
<code>csce322a04part02.pl</code>	40
<code>csce322a04part03.pl</code>	30
Total	100

6 External Resources

[Prolog - Wikibooks](#)

[Learn Prolog Now!](#)

[Prolog Tutorial Category:Prolog - Rosetta Code](#)