

# Artificial Intelligence for Big Data Systems

## Assignment 2: Knowledge-based Problem Solving

### Python code

### Nikolay Shivarov

За да пазя фактите ще използвам речници, като ще имат същите имена като функциите и предикатите в теорията. В началото създавам двумерен масив, който ползвам за да напълня речниците isCell и тези за съседствата и от там нататък изобщо няма да го използвам. В речника isCell влизат всички клетки със стойност True(не са стени) и само на тях добавям съседите им в речниците за съседства. Във всички речници освен тези за съседствата само ключовете са ни нужни и затова слагам стойност True. В другия случай слагам key сегашната клетка и value съответния съсед. Ръчно слагам стойностите в :

```
isFinal = {(rows-1, cols-1):True}
```

```
isStarting = {(0,0):True}
```

```
path = {((0,0), ""):True }
```

За решение на проблема създавам празен списък, който пази новооткритите пътища и пускам цикъл по досегашните пътищата и добавям новите пътища в списъка след това добавям целия списък към речника path и това се повтаря докато finalPath е празен. Ще разгледаме само случая с лявата посока. Тук p[0] е клетката, до която имаме път, заместваме isCell(leftNeighbour(p[0])) и ако имаме такъв факт то можем да добавим (leftNeighbour[p[0]], p[1] + "L") към фактите, които имаме. От изпълнението на програмата получаваме два пътя: {((4, 5), 'DRRRRRDDD'): True, ((4, 5), 'DRRRRDRDD'): True}

В глобалния речник bPath ще пазя вече намерените пътища. За решение на проблема ще използвам рекурсия. Функцията има аргументи клетка и субституциите от p до pi, ако сме извършили повече от 11 субституции спираме функцията няма да гледаме пътищата с дължина над 12. Отново ще разгледаме само случая с левия съсед. Ако левия съсед не е клетка не правим нищо. Ако е левия съсед е стартовата клетка, тогава pi = "", следователно извършваме всички субституции и записваме новия път в bPath. В третия случай добавяме още една субституция и викаме функцията за левия съсед. Накрая викаме функцията с аргументи финалната клетка и празно множество от субституции.