Project 3

4 October 2016

How are computer languages implemented?

- A computer language like Python has well-defined syntax and semantics
- The Python interpreter has two big tasks:
  - Parsing the program (which is essentially a big text string) into an internal representation
  - Interpreting/executing this internal representation

Aim of Project

- We will define and implement a very simple programming language for controlling a very simple robot
  - Simple sequential programs (question 1)
  - Programs using simple functions (question 2)
  - Programs using limited iteration (question 3)

Skills developed

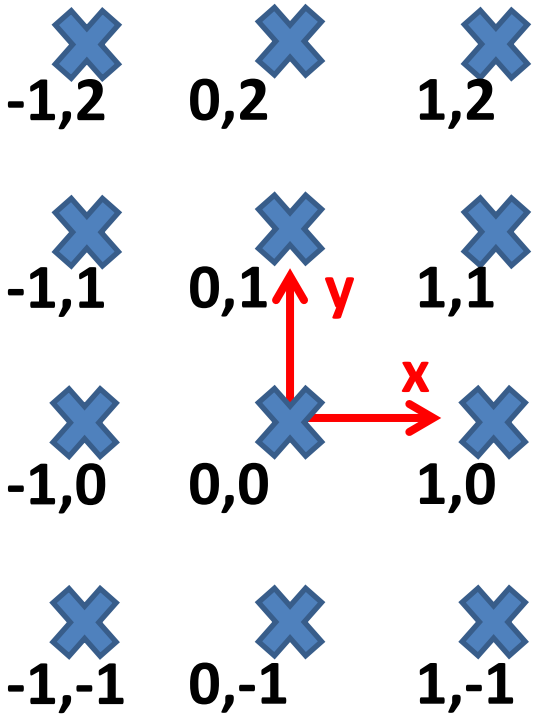
- This project will practice your skills with dictionaries
- You will learn how to write code in one language that interprets code in a different language

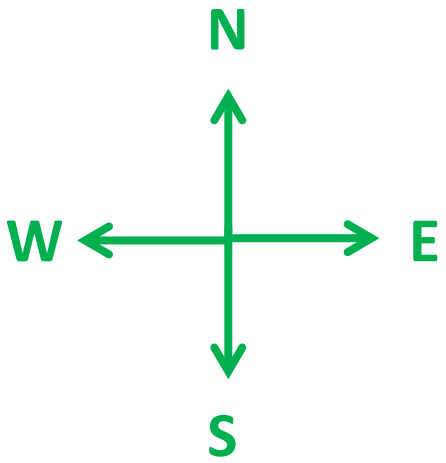
Background: Robots

- Robots can execute a given sequence of actions
- We need to predict what the state of the robot will be after executing a sequence of actions

A Very Simple Robot

The robot moves on an x,y grid

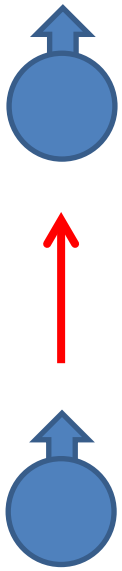




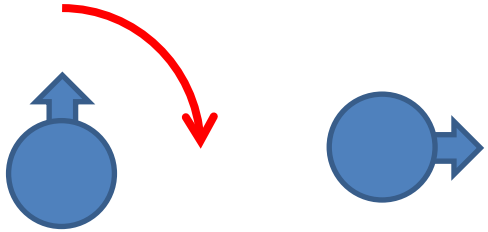
and can face in one of four directions: N, S, E, W

A Very Simple Robot

The robot can perform two actions



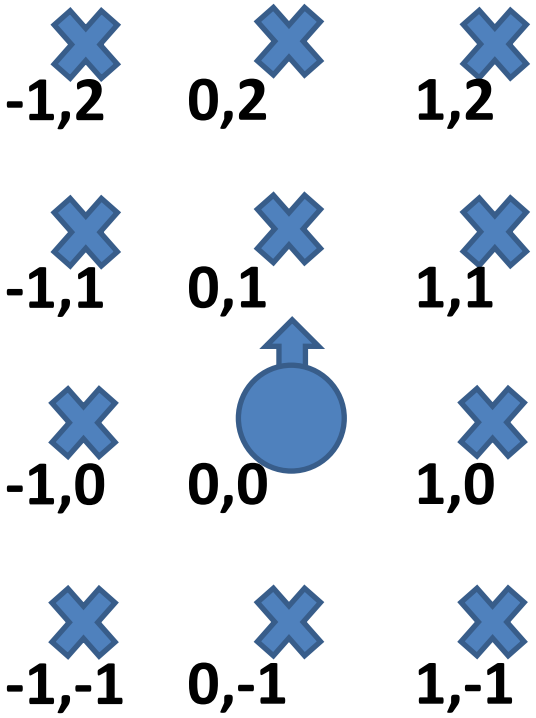
Move
(forward one step
on the grid)



Turn
(90 degrees to right
on the spot)

A Very Simple Robot

The state of the robot is described by (x, y, direction)



State: (0, 0, ‘N’)

State: (-1, 2, ‘W’)

Our Programming Language

- Each "program" is a list of operators corresponding to actions of the robot
- e.g. ['move', 'turn', 'move']
- We always assume that the robot starts in the initial state (0, 0, 'N')
- Our aim is to write an interpreter for our language, which can determine the final state of the robot after executing a given program

Question 1

Write a function simple_interpreter(code) that takes a program code in our language, and returns a tuple corresponding to the new state of the robot after executing the program

The initial state before executing the program is always (0, 0, 'N')

Examples

- ['move', 'turn', 'move']
  leaves the robot in the state (1, 1, 'E')
- ['move', 'turn', 'turn', 'move']
  leaves the robot in the state (0, 0, 'S')
- ['turn', 'turn', 'turn', 'move', 'move']
  leaves the robot in the state (-2, 0, 'W')

Question 2

Some sequences of operators are frequently repeated

e.g. ‘turn’, ‘turn’, ‘turn’ to turn to the left

We can define a macro, which associates a name with a sequence of operators
e.g. {'turnleft' : ['turn', 'turn', 'turn']}

We can then use these macros in our program, and our interpreter should replace the appearance of a macro by the corresponding sequence of operators

Examples

Given a dictionary of macros:

```
{'turnleft' : ['turn', 'turn', 'turn'], 'turnright' : ['turn'],
 'bigleftturn' : ['move', 'move', 'turnleft', 'move', 'move'],
 'bigrightturn' : ['move', 'move', 'turnright', 'move', 'move']}
```

- ['turnleft', 'turnright']
  leaves the robot in the state (0, 0, 'N')
- ['bigleftturn', 'bigrightturn']
  leaves the robot in the state (-4, 4, 'N')

Question 2

Write a function macro_interpreter(code, macros) that takes a program code in our language and a dictionary of macro definitions macros, and returns a tuple corresponding to the new state of the robot after executing the program

The initial state before executing the program is always (0, 0, 'N')

You can assume that the macros will not result in any infinite loops

Question 3

Finally, we want to introduce a simple form of iteration into our language: 'repeat' count body 'end'

e.g. ['repeat', 3, 'move', 'end']

Our interpreter should replace the repeat statement by <count> copies of the sequence of operators in the <body>

e.g. ['move', 'move', 'move'] which evaluates to (0, 3, 'N')

Note that the body of a repeat statement can contain any number and combination of 'move', 'turn' and 'repeat' statements

Examples

- ['turn', 'repeat', 2, 'move', 'turn', 'end']
  is equivalent to
  ['turn', 'move', 'turn', 'move', 'turn']
  and leaves the robot in the state (1, -1, 'W')
- ['repeat', 2, 'repeat', 2, 'move', 'turn', 'end', 'end'] is equivalent to
  ['move', 'turn', 'move', 'turn', 'move', 'turn', 'move', 'turn']
  and leaves the robot in the state (0, 0, 'N')

Question 3

Write a function repeat_interpreter(code) that takes a program code in our language with repeat statements, and returns a tuple corresponding to the new state of the robot after executing the program

The initial state before executing the program is always (0, 0, 'N')

Question 3

The language for Question 3 does **not** include macros

Note that an 'end' statement should be matched with the nearest 'repeat' statement

Note that it is quite tricky to implement nested repeat statements. If you can only implement an interpreter that supports a single level of repeat statements rather than nested repeat statements, then you should submit that and you are likely to get at least some marks for this question.

Academic Honesty

- We had to interview several students after Project 1 to discuss the originality of their project submission

Academic Honesty

- You are permitted to look at code on the web for ideas (although not code from other students for this project).
- If you base your code on ideas from code that you have seen, you should add a comment to your code as a citation, e.g., the URL of the web page and the idea that you used.
- Note that you should not just copy code from the web verbatim.
- You should write your own code based on the ideas that you have seen.
- If your submitted code is mostly a copy of code from the web, you may lose marks as your submission is not really your own original work.

Academic Honesty

- All assessment items (worksheets, projects, test and exam) must be **your own, individual, original work**.
- For example, you must not copy the code of other students, and you must not make your code available to others to see. Do not give other students your login id and password, do not share USB memory drives, do not post your code on public forums, or any other activity that would make your code available to others. Likewise, do not ask other students to see their code. If other students ask to see your code, please say "no", as copying (collusion or plagiarism) is considered academic misconduct, and all students involved may face penalties (both the student who copied, and the student who made their code available).
- Any code that is submitted for assessment may be automatically compared against other students' code and other code sources using sophisticated similarity checking software, and cases of potential copying may lead to a formal academic misconduct hearing.
- For further information, please see the university's Academic Honesty and Plagiarism website, or ask your lecturer.

Conclusion

- The project questions will be submitted via Grok
- The specification of the project questions and the deadline will be announced very soon in Grok
- This project is worth 10% of the final subject
- We will be marking the correctness, quality, readability and commenting of your code
- Submit early and often