Grundlagen der künstlichen Intelligenz

Exam Bonus Programming Exercise 1 — 4th November 2016

Problem 1.1: Ladders

Ladders is a word game where a player tries to get from one word, the "start word" to another, the "goal word" in as few steps as possible. In each step, the player must either add one letter to the word from the previous step, or take away one letter, and then rearrange the letters to make a new word.

Example: **croissant** to **baritone**

croissant

(- C)

arsonist

(- S)

aroints

(+ E)

notaries

(+ B)

baritones

(- S)

baritone

This can be modelled as a search problem, where each node represents a word and the children (actions) of this node are the words that can follow in the next step. You should implement Breadth First Search or Iterative Deepening Search to find the shortest ladder between a start word and a goal word. All words in between must be found in the file `wordList.txt` provided in the folder `example_solution`.

Submission information

You may implement this in Java, Python or MATLAB. Your submission should consist of the following files at least:

1. `output_1.txt` – the answer to **croissant** – **baritone**
2. `output_2.txt` – the answer to **crumpet** – **treacle**
3. `output_3.txt` – the answer to **apple** – **pear**
4. `output_4.txt` – the answer to **lead** – **gold**
5. `readme.txt` – details on the version of Java/Python/Matlab you used as well as a description of your solution, and any packages you used.
6. `wordList.txt` – the text file that I provided you
7. `ladders.jar / ladders.py / ladders.m` – see below for details

These files should be on the root directory. In addition, you may need to precompute some data for speed. These precomputed files should also be uploaded.

Java

If you implement this in Java, you should provide a file `ladders.jar` in the root directory which takes the two words as arguments, so that I can execute it like this:

```
>> java -jar ladders.jar startword goalword
```

Python

If you implement this in Python, you should provide a file `ladders.py` in the root directory taking two words as arguments which I can execute so:

```
>> python ladders.py startword goalword
```

MATLAB

If you implement this in MATLAB, your file `ladders.m` should be a function I can execute like so in the MATLAB command line:

```
>> ladders('startword','goalword')
```

Please do not use absolute paths! Use paths relative to the working directory. The `startword` and `goalword` will be in lower case, i.e. `croissant baritone`. The output should be the file `output.txt`, which is saved in the root directory. The output file should have all the words lower case, start with `startword` and end with the `goalword`, and have each word at each step on a new line. An example is in `output.txt` in this folder. If no solution exists, your output file should be empty.

Marking

Your solution counts as a pass if:

1. the files `output_1.txt`, `output_2.txt`, `output_3.txt` and `output_4.txt` are correct, i.e., they find a ladder of shortest length.
2. I can execute your code and obtain a correct solution in the right format. I will try one of the given word pairs (i.e. one of **croissant** – **baritone**, **crumpet** – **treacle**, **apple** – **pear** or **lead** – **gold**) and one new word pair.
3. your code finds the solution to **croissant** – **baritone** in under 5 minutes[1].
4. you have not copied your solution (use of existing packages is fine as long as cited in the readme). We will use a plagiarism detection tool and any copied code will annul all bonus exercises from both the copier and the copied person!

Submission will close on **Friday 23rd December at 23:59**. I will then mark all the solutions using a shell script (so it is very important to follow the instructions exactly!) and will return them with automated feedback. If the error is minor, e.g. a formatting error or a conceptual error producing a wrong solution, you will then have a week to correct your solution and reupload if necessary.

[1] When I run your code I will not run any other program apart from background processes. I am running a Macbook Pro with 16GB RAM and i7 Core with 2.8GHz