Grundlagen der Künstlichen Intelligenz

Exam Bonus Exercise 1: Part 1 – Search — 1st December 2017

Markus Koschi (last updated 30th November 2017)

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 modeled 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 a search algorithm to find the shortest ladder between a start word and a goal word. All words in between must exist in the file `wordList.txt` provided in Moodle in the folder `example_solution`.

Submission information

You may implement this in Java, Python 3, or MATLAB (R2016a). 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 3

If you implement this in Python 3, 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 (I will use MATLAB R2016a):

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

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. As an example, see `output_1.txt` in the provided folder `example_solution`. If no solution exists, your output file should be empty.

Marking

Your solution counts as a pass only 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 (i.e. your `ladders.jar / ladders.py / ladders.m` file) with a word pair and obtain a correct solution in the right format.
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 **Saturday 23rd December at 23:59**. I will mark all the solutions using a shell script. You can either pass (if all four requirements listed above are met) or fail (unfortunately, I cannot manually check for minor errors). Thus, it is very important to follow the instructions exactly! The results will be returned with automated feedback via Moodle.

[1] When I run your code, I will not run any other program apart from background processes. I am running a machine with 20GB RAM and i7 Core with 2.6GHz