Middle East Technical University
Department of Computer Engineering

CENG 111

Fall 2014

Take Home Exam 4

REGULATIONS

Due date: 10 January 2014, 23:59, Saturday *(Not subject to postpone)*

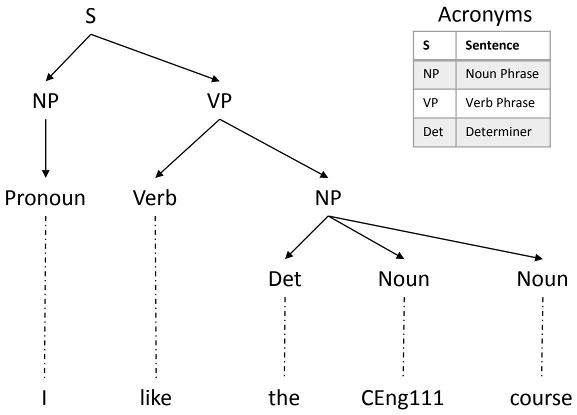
Submission: Electronically. You should put your solution code into a file called `the4.py` and submit it through the COW web system. Resubmission is allowed (till the last moment of the due date); the last one replaces the previous ones.

Team: There is **no** teaming up. The take home exam has to be done/turned in individually.

Cheating: **This is an exam:** All involved parties (source(s) and receiver(s)) get zero, and will be subject to disciplinary action.

INTRODUCTION

In this THE, you will be dealing with a very limited version of parsing natural language using trees. Although there are several mechanisms for analyzing the grammar of a language, trees are very straightforward to use for this task. For example, a subset of English can be described using a tree of the form given in the following figure.



Acronyms

S	Sentence
NP	Noun Phrase
VP	Verb Phrase
Det	Determiner

PROBLEM

Given a tree for the grammar of a language (not necessarily English) and a dictionary for the words in the language, your task is to check whether a given sentence is valid for that language. The tree will be given to you in the following form:

```
[Nodename, Branch1, Branch2, ..., BranchN]
```

where `Nodename` is a `str` data, and each `Branch-`*i* is itself a tree (i.e., a list). If a node does not have any branches, it will be given as follows:

```
[Nodename]
```

For the example tree given in Introduction, the tree should be as follows:

```
["S", ["NP", ["Pronoun"]], ["VP", ["Verb"], ["NP", ["Det"], ["Noun"], ["Noun"]]]]
```

The dictionary for the language, which actually corresponds to the possible values of the leaf nodes in the tree, will be given as a `dict` data type in Python:

```
{LeafNode-1:[value11, ..., value1K], ..., LeafNode-M:[valueM1, ..., valueML]}
```

where each `value-`*i* is a `str` data. For the example tree given in Introduction, the dictionary should be specified as follows:

```
{"Pronoun":["I", "He", "She"], "Det":["the", "a", "an"], "Verb":["like", "love", "attend"],\
     "Noun":["CEng111", "CEng100", "morning", "evening", "lecture", "breeze", "course"]}
```

SPECIFICATIONS

- Your solution should be accessible from a function conforming to the following (you can use different parameter names):

  ```
  is_valid(Tree, Dictionary, Sentence)
  ```

  which should return `True` if `Sentence` can be parsed by `Tree` given `Dictionary`.
- The given sentence will not include any punctuation, redundant spaces or characters. We will give you only a sequence of words separated by a single space. The given sentence might contain words that are not in the dictionary, in which case, the returned value of the function should be `False`.
- It is guaranteed that for a word category (e.g., `Pronoun`, `Noun`, etc. in the above example), there will be at least one word given in the dictionary.
- Your processing of the words in the dictionary and the names of the nodes in the tree should be case **insensitive**.
- Your processing should not check the tense or the plural/singular form of the verb.
- Leaf-nodes may share the same name (as in the example given above). In such cases, in the dictionary, there will be only one description of the word category for that node.
- You can use recursion or iteration freely. You can only use built-in functionalities of Python, and you are not allowed to import any libraries.

EXAMPLE RUN

```
>>> T = ["S", ["NP", ["Pronoun"]], ["VP", ["Verb"], ["NP", ["Det"], ["Noun"], ["Noun"]]]]
>>> D = {"Pronoun":["I", "He", "She"], "Det":["the", "a", "an"], \
"Verb":["like", "love", "attend"], \
"Noun":["CEng111", "CEng100", "morning", "evening", "lecture", "breeze", "course"]}
>>> is_valid(T, D, "I like the CEng111 course")
True
>>> is_valid(T, D, "I like the morning breeze")
True
>>> is_valid(T, D, "I the morning breeze like")
False
>>> is_valid(T, D, "I run every day")
False
```

EVALUATION

- We will not test your code with erroneous inputs or grammars. However, there might be words in the sentence that are not given in the dictionary.
- You will lose points if your code prints values to the screen, or if it does not conform to any of the specifications given above.
- All submissions will be tested under strictly equal conditions on multiple data sets on Linux-running lab machines using Python version 2.7.6.