Design Lab (CS69202)
Spring 2023
Assignment 1
Lex and Yacc Basics : Parentheses and Sentences

Task 1 (10 marks) :

Given an input string consisting of '(' and ')', write a Python code using PLY so that the input contains matching parenthesis. Also print the token list.

Example : Input : () () (()) ((())) (()())
Output : Valid Matching. Note: The spaces are to be ignored.
Input : (()))()()))(((()
Output : Invalid Matching. Note: The spaces are to be ignored.

Task 2 (20 marks):

Given an input string consisting of words in the English alphabet, write a Python code using PLY so that the input is correct as per the Grammatical Rules. You have to create PLY rules accordingly. Add the parts of speech associated with each word for the **valid** sentences only. Consider only lowercase alphabets for the inputs. Also print the token list.
In addition to it, create your own custom rules so that it can detect grammatically correct sentences accurately. Keep input sentences of size at least 2 words and a maximum of 8 words else consider sentences as invalid.

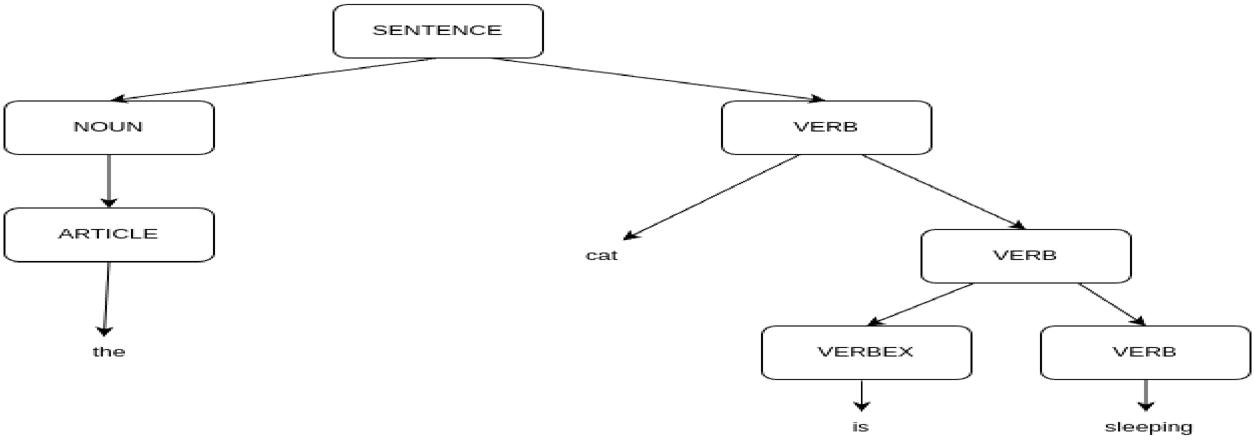
Example : Input : the cat is sleeping.
Output : Valid Statement.
Article Noun Verb Verb

Input : the is the beauty.
Output : Invalid Statement.

Associated Grammar Rules in the example:

SENTENCE -> NOUN VERB
NOUN -> ARTICLE | Noun
ARTICLE -> a | an | the | epsilon
VERB -> Noun VERB| VERBEX VERB| Verb
VERBEX -> is | am | was | were | are

Noun and Verb are the associated Nouns and Verbs of the sentence.
An example of the valid input grammar tree is :



Bonus (5 marks):

- You can add more rules than the given ones in order to validate a larger number of grammatically correct sentences.

Update (For Task 2):

To consider the regex for the terminals in your grammar (i.e. verb, noun, adjective, article etc), use the Brown corpus from the nltk package (I hope you have the idea of nltk package as done in Computing Lab last semester).

From the Brown corpus take a minimum of 10 words or maximum words possible per POS tag as follows :

1. AT -article (Words of POS type AT will be stored in a terminal Article)
2. NN + NNS + NP + NPS - Noun (Words of POS type NN, NNS, NP, NPS will be stored in a terminal Noun (at most 10 words combined)).
3. JJ - Adjective (Words of POS type JJ will be stored in a terminal Adjective)
4. VB, - Verb (Words of POS type VB will be stored in a terminal Verb)

In addition, to handle inputs containing terms like like 'is sleeping', 'was talking', create a rule (along with the above 4 points)

'verbc -> verbca verbcb'

where verbca = {is, am, was, are, were} ,
and verbcb = {sleeping, talking, crying, laughing, feeding, eating, bathing. grumbling, loitering, watching}.

You NEED not have to enforce any strict restrictions on POS grammar rule usage.
Also, you NEED not worry about the test cases. The test cases will be dealt with.

Input format is a single line from the console. Example : 1) The ship sails. 2)The cat is sleeping etc.
Write a readme file mentioning about the rules and regexes implemented in Task 2 only.

Also,you can write some more rules as said earlier (bonus) for precision.

INSTRUCTIONS

1. Write **Python** code using **PLY** to extract the above fields. Your program should show all the possible query fields a user can ask for (from the above list items).
2. You must think correctly about what kind of **errors** can come in the process and try to handle them. Use the PLY package in python. PLY ref: https://www.dabeaz.com/ply/
3. You must **NOT** use any other parsing tools apart from **PLY** (ex: **Beautiful Soup** is a **strict no** or **any other framework**) . Should anyone not adhere to this instruction, they will be awarded **ZERO** marks.
4. Your code should address the objectives using **PLY.** Anyone found addressing the objective with no such use of **PLY** will be awarded **ZERO** marks.
5. Not adhering to these instructions can **incur** a penalty (worst case being **0 marks**).
6. You can write a readme file to provide any particular instructions related to program execution steps, input format, or anything that you might think is useful for the evaluator while evaluating the assignment.
7. **Plagiarism** in any form is **not** allowed. Students found **copying/sharing** code will be awarded **0 marks**. You may discuss ideas, share your logic etc but you must not share/copy code at all costs.

DELIVERABLES

Submit all the python files in a folder named in the format: <Roll No.>_DesLab_A1. Compress this folder to zip format, creating a compressed file <Roll No.>_DesLab_A1.zip .Upload this compressed file to moodle.