

PRACTICAL No. 4

Name : Shantanu Mane

Roll No. : E-63

Topic: Parsing

Platform: Windows or Linux

Language to be used: Python or Java (based on the companies targeted for placement)

Aim: (A) Write a program to validate a natural language sentence. Design a natural language grammar, compute and input the LL (1) table. Validate if the given sentence is valid or not based on the grammar.

Input: NLP grammar and LL (1) parsing table (from file)

Implementation: String parsing rules

Output: Each step-in string parsing and whether the input string is valid or invalid.

CODE :

```
import pandas as pd

def util(ll1):
    tab = pd.DataFrame(
        ll1,
        columns=[
            "championship",
            "ball",
            "toss",
            "is",
            "want",
            "won",
            "Played",
            "me",
            "I",
            "you",
            "India",
            "Australia",
            "Steve",
            "John",
            "the",
            "a",
            "an",
        ],
    )
    tab["Nonterm"] = ["S", "NP", "VP", "N", "V", "P", "PN", "D"]
    tab.set_index("Nonterm", inplace=True)
    return tab

def validator(input):
```

```
from beautifultable import BeautifulTable
```

[illegible]

```

        "Australia",
        "Steve",
        "John",
        "",
        "",
        "",
    ],
    ["", "", "", "", "", "", "", "", "", "", "", "", "", "", "the",
"a", "an"],
]
tab = util(ll1)
table = BeautifulTable()
table.column_headers = ["Buffer", "Stack"]
buffer = input.split(" ")
buffer.reverse()
stack = ["S"]
table.append_row([buffer.copy(), stack.copy()])
while buffer != [] and stack != []:
    index = stack.pop(0)
    key = buffer[-1]
    if key not in tab.columns:
        print("Invalid input")
        table.append_row([buffer.copy(), stack.copy()])
        print(table)
        return

    rule = tab.loc[index][key].split(" ")

    if "" in rule:
        print("Invalid input")
        table.append_row([buffer.copy(), stack.copy()])
        print(table)
        return

    stack = rule + stack
    table.append_row([buffer.copy(), stack.copy()])

    if key in rule:
        buffer.remove(key)
        stack.remove(key)
        table.append_row([buffer.copy(), stack.copy()])
print(table)
print("Valid input")

input = "India won the championship"
validator(input)
input = "championship India won"
validator(input)

```

OUTPUT:

Buffer	Stack
['championship', 'the', 'won', 'India']	['S']

+-----+-----+		
['championship', 'the', 'won', 'India']	['NP', 'VP']	
+-----+-----+		
['championship', 'the', 'won', 'India']	['PN', 'VP']	
+-----+-----+		
['championship', 'the', 'won', 'India']	['India', 'VP']	
+-----+-----+		
['championship', 'the', 'won']	['VP']	
+-----+-----+		
['championship', 'the', 'won']	['V', 'NP']	
+-----+-----+		
['championship', 'the', 'won']	['won', 'NP']	
+-----+-----+		
['championship', 'the']	['NP']	
+-----+-----+		
['championship', 'the']	['D', 'N']	
+-----+-----+		
['championship', 'the']	['the', 'N']	
+-----+-----+		
['championship']	['N']	
+-----+-----+		
['championship']	['championship']	
+-----+-----+		
[]	[]	
+-----+-----+		

Valid input

Invalid input

+-----+-----+		
Buffer	Stack	
+-----+-----+		
['won', 'India', 'championship']	['S']	
+-----+-----+		
['won', 'India', 'championship']	[]	
+-----+-----+		

(B) Use Virtual Lab on LL1 parser to validate the string and verify your string validation using simulation.

Link for Virtual Lab: http://vlabs.iitb.ac.in/vlabs-dev/vlab_bootcamp/bootcamp/system_deligators/labs/exp2/index.php

Output: Validation from Virtual lab simulator

Details:

PART A:

- Construct and consider a natural language grammar that can validate an English sentence.
- Solve the NLP grammar by hand for LL(1) parser and create parsing table
- Input the above parsing table and grammar using a file.
- Write program for performing string validation

PART B:

- Go to Virtual lab: Go through all the tabs, paste screen shots for all steps (including tests), validate your string parsing with the simulator (screen shot expected).