

ID 914855947

Student: Hsuan-Chih, Chen

CS311 programming assignment 2

Simple Grammar Analyzer

I choose python to do this Grammar project. I also use Json to encode the rules of grammar. I put all rules in the dictionary data type such as `{"S": ["aTb"], "T": ["aTb", "#"]}`. By this way, I easily can map the rules and use them when I process my program.

The hardest part in this assignment is the parsing the string. In the assignment paper, it is very clear and helpful to lead us setting conditions. Even if the steps and conditions are very clear, this part still took me a lot of time to enable my parsing function to work without bugs. I had some problems and difficulties when programming my parsing function:

Firstly, in python, we have to delete character of the input string one by one for parsing. My way is split the input string into characters and put them orderly in a list such as `"aa#bbb"` \rightarrow `['a','a','#','b','b']`, so I can use every character in the original string. By using this way, I can accomplish the functionality of crossing off the first character of the string.

Secondly, for the stack, there is a `"pop()"` function in the python. Initially, I misunderstood and thought it's supposed to pop the first element in the list. By spending 1 hour printing the stack content in every loop, I finally found the problem that the `"pop()"` function pops the last element in the list of stack. Following the `pop()` way, I also have to change my original way of pushing elements into the stack. The order of the elements is supposed to be reversed (`aTb` \rightarrow `bTa`), and then we append each element to the last position (`#` \rightarrow `#bTa`). By this way, we can make sure the top element of the stack we will pop is the latest and right element that we want to compare with the next input character.

Thirdly, for the conditions setting, there are mainly five conditions we have to check and process the next step. It is similar with that which state we are in this machine and what's the next. I followed the steps of the assignment paper drew the graph and processed it. In fact, it is not easy for me to come up with the correct algorithm. I asked to my classmates for help and discuss with them how to do it correctly. Eventually, my parsing function did work. The sketch is that we check the string is empty (if empty, stop looping) and stack is not empty first in the loop. Then we check the top element of stack is variable or terminal, and then do either way's processes. Finally, if input string become empty (jump out of the loop) and the stack is also empty, we accepts the string. Otherwise, we rejects the string. Although it just few lines of my code, it actually spend me much time to make it work without errors.

Finally, there are some restrictions of rules on the grammar in this analyzer. They are mentioned in the assignment paper.

1. All rules in the grammar must have a terminal as the first symbol on the right hand side of the rule. For example, in my program, we cannot have the rules like $S \rightarrow Tab$ or $S \rightarrow T$. We are supposed to have $S \rightarrow aTb$ or $T \rightarrow \#$. By following the assignment paper, if now we get a variable, we have to find the rule having the first character same as the current first character of the input string. If the first character of the right hand side of the rule is a variable, it is absolutely return False and the program is crashed. Therefore, the rules of the grammar we put in are supposed to have a terminal as the first symbol on the right hand side.
2. No variable in the grammar can have two rules with the same terminal as the first symbol on the right hand side. For example, we cannot have the rules like $S \rightarrow aTb \mid aTbb$ or $T \rightarrow aTb \mid a$. The reason is that if the rules has the same terminal as the first symbol on the right hand side, the program cannot recognize which rules it should choose now. If the current input character is 'a', there are two options "aTb" | "aTbb". They are ambiguous for the program so it cannot have a right result.

Testing result:

```
>>> ===== RESTART =====
```

```
>>>
```

file name

1. $\{a^n \# b^n \mid n > 0\}$
2. $\{w \# w^R \mid w \text{ is element of } \{0,1\}^*\}$
3. $\{a^i \# b^j \# c^k \mid i = j \text{ and } i, j, k > 0\}$
4. $\{1^i \# 0^j \mid i^2 = j \text{ and } i, j > 0\}$

which file do you want to open?: 1

1. $\{a^n \# b^n \mid n > 0\}$

Please type in a string: aaaaa#bbbbbb

aaaaa#bbbbbb

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: a#b

a#b

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: aa#bb

aa#bb

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: bb#aa

bb#aa

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: aaa#bbb#ccc

aaa#bbb#ccc

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: ccc

ccc

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: aaaa#bbbb#

aaaa#bbbb#

file name

1. $\{a^n \# b^n \mid n > 0\}$

2. $\{w \# w^R \mid w \text{ is element of } \{0,1\}^*\}$

3. $\{a^i \# b^j \# c^k \mid i = j \text{ and } i, j, k > 0\}$

4. $\{1^i \# 0^j \mid i^2 = j \text{ and } i, j > 0\}$

which file do you want to open?: 2

2. $\{w \# w^R \mid w \text{ is element of } \{0,1\}^*\}$

Please type in a string: 101

101

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: 10101#10101

10101#10101

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: 11#11

11#11

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: 0000#0000

0000#0000

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: 101011#101011

101011#101011

REJECT

file name

1. $\{a^n \# b^n \mid n > 0\}$

2. $\{w \# w^R \mid w \text{ is element of } \{0,1\}^*\}$

3. $\{a^i \# b^j \# c^k \mid i = j \text{ and } i, j, k > 0\}$

4. $\{1^i \# 0^j \mid i^2 = j \text{ and } i, j > 0\}$

which file do you want to open?: 3

3. $\{a^i \# b^j \# c^k \mid i = j \text{ and } i, j, k > 0\}$

Please type in a string: aaa#bbb#ccc#

aaa#bbb#ccc#

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: aaaa#bbbb#cccc#

aaaa#bbbb#cccc#

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: a#b#c#

a#b#c#

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: a#b#cc#

a#b#cc#

ACCEPT

The string in the grammar.

continue (Y/N): y

Please type in a string: aa#bbbb#c

aa#bbbb#c

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: aaa#bbbbbb#cc#

aaa#bbbbbb#cc#

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: ###

###

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: aaaaa#b#cccc#

aaaaa#b#cccc#

REJECT

Please type in a string: bbb#aaa#cccc#

bbb#aaa#cccc#

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: cccc#aaaa#bbbb#

cccc#aaaa#bbbb#

REJECT

file name

1. $\{a^n \# b^n \mid n > 0\}$
2. $\{w \# w^R \mid w \text{ is element of } \{0,1\}^*\}$
3. $\{a^i \# b^j \# c^k \mid i = j \text{ and } i, j, k > 0\}$
4. $\{1^i \# 0^j \mid i^2 = j \text{ and } i, j > 0\}$

which file do you want to open?: 4

4. $\{1^i \# 0^j \mid i^2 = j \text{ and } i, j > 0\}$

Please type in a string: 11#0000

11#0000

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: 111#000000

111#000000

ACCEPT

The string in the grammar.

continue (Y/N): Y

Please type in a string: #0

#0

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: 111#000

111#000

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: 000#111111

000#111111

REJECT

The string is not in the grammar

continue (Y/N): Y

Please type in a string: 000000#

000000#

REJECT