Lab 6 –Syntax Analysis

SEG2106 – Software Construction

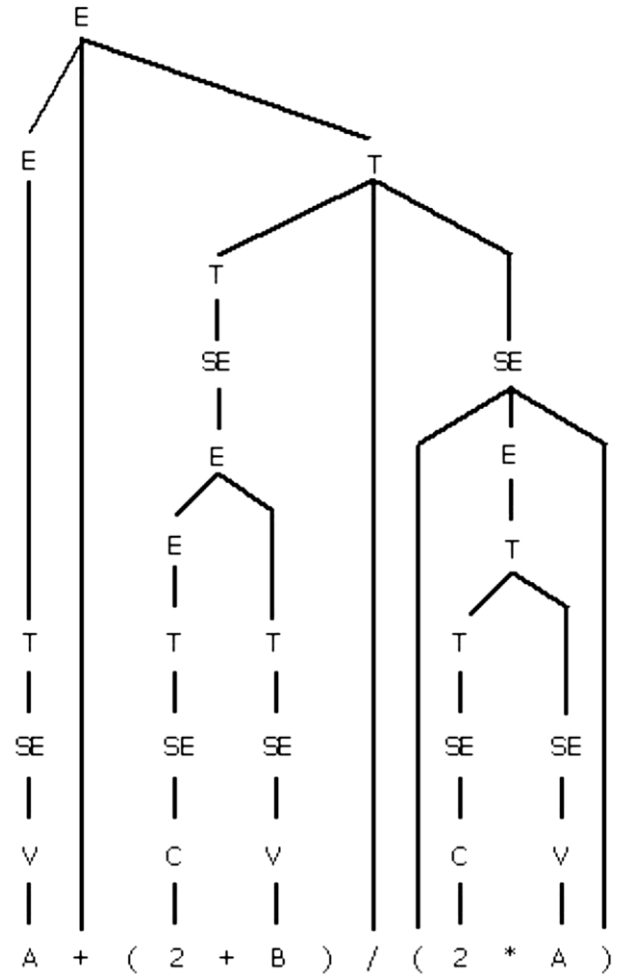
Exercise 1: Construction of a syntax tree (25 points)

We consider the following grammar:

```
E → T | E - T | E + T
T → SE | T * SE | T / SE
SE → V | C | ( E )
V → A | B
C → 0 | 1 | 2 | 3
```

Write down a syntax tree for the following expression:

```
A + ( 2 + B ) / ( 2 * A )
```

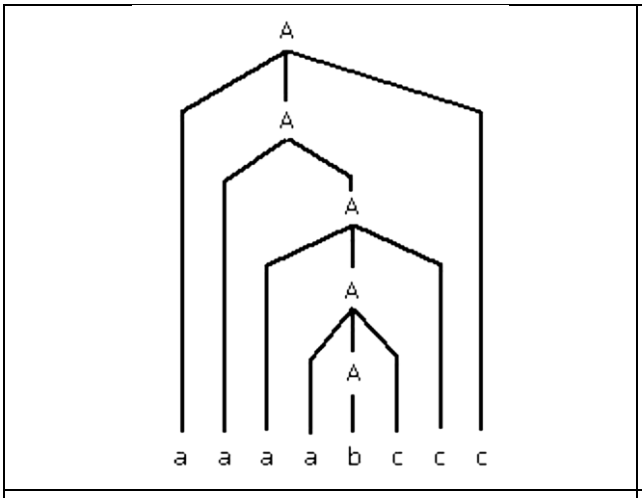


Exercise 2: Ambiguity (25 points)

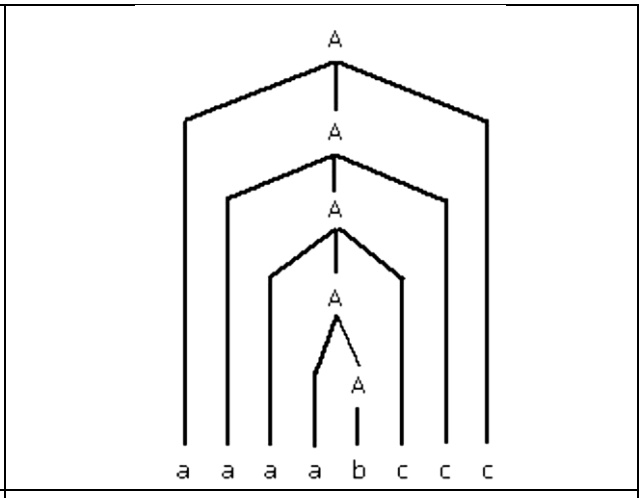
Given the following Grammar (**A** is the starting symbol):

A → a A c | a A | b

Is the latter grammar ambiguous? - Explain in a few words why it is - or is not.



Tree representing string **aaaabccc**



Another tree representing string **aaaabccc**

As it can be seen above, the grammar provided for this exercise is ambiguous. More than one tree can be constructed to represent the same substring.

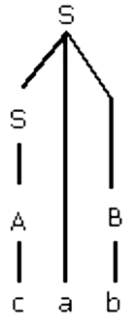
Exercise 3: Generation of sentences (25 points)

We consider the following grammar:

S → S a B | A
A → d A | c
B → b | A

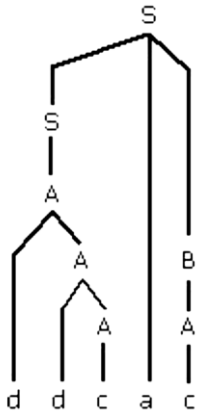
Which of the following sentences can be generated by this grammar?

1. cab



2. c

3. cabb → cannot be generated
4. caab → cannot be generated
5. ccaabb → cannot be generated
6. ccaab → cannot be generated
7. ddcac



Exercise 4: Design a grammar (25 points)

Develop a grammar for the language that consists of all strings formed from the characters a and b that have the same number of a's and b's. For example, the strings aabb, baba, and aabbababba are part of this language, but not the strings abb, bba, and aabbabb.

Explain why your grammar "does the job".

One possible solution:

S → aSb | bSa | SS | ε