Part II: Grammars

Submission: On Canvas, under Assignments
Deadline: Week 5, Thurs 10th Feb 2022

Notes on answering this question:

- **Pay attention to what kind of grammar I ask for** (*regular*, or *context free*, or some other type).
- For instance: if I ask for a **regular** grammar and you give me some other kind of grammar, then you may lose marks!
- In particular, if your regular grammar contains a rule of the form $S \longrightarrow ab\gamma$ or $S \longrightarrow D$ or $S \longrightarrow \varepsilon D$, then this is not a regular grammar and your answer contains an error.
- Always clearly specify the start symbol.
- The alphabet (set of tokens) of a language cannot contain ε. ε is the empty string, that is, an absence of tokens. If your answer contains something of the form T={ε,…} (where T is supposed to be a set of tokens for your language) — then your answer is probably wrong and you're probably losing marks.

Some of you have not met set notation before, so here's a quick tutorial:

- The language determined by the regex /a*/ is $\{ a^n \mid n \in \mathbb{N}\}$ or equivalently $\{ a^n \mid n \in \{0,1,2,\ldots\} \}$ or equivalently $\{ a^n \mid n \geq 0 \}$.
- The language determined by the regex /(a|b)?/ is { a, b, ε }.
- The language determined by the regex /a+b+/ is $\{ a^m b^n \mid m,n \geq 1 \}$ or equivalently $\{ a^m b^n \mid m \geq 1, n \geq 1 \}$ or equivalently $\{ a^m b^n \mid m,n \in \{1,2,3,\ldots\} \}$.
- The language determined by the English description "any nonzero digit" is { 1, 2, 3, 4, 5, 6, 7, 8, 9} or equivalently { 1, 2, … , 9} or equivalently $\{ n \mid 9 \geq n \geq 1 \}$ or equivalently $\{ n \mid 0 < n \leq 9 \}$.

Now on to the questions:

1. Write the language determined by the regex /a*b*/

   $L = \{a^m b^n \mid m \geq 0 , n \geq 0\}$

2. Write a **regular** grammar to generate the language determined by the regex /a*b*/

   S :: aS | aT | ε
   T :: b | bT

3. Write the language determined by the regex /`(ab)*`/

   $L = \{(ab)^n \mid n \geq 0\}$

4. Write a regular grammar to generate the language determined by the regex /`(ab)*`/

   S :: aB | ε
   B :: bS | b

5. Write the language determined by /`Whiske?y`/. The alphabet is {W,h,i,s,k,e,y} (W is a terminal symbol here).

   $L = \{ \text{Whiske}^n\text{y} \mid n \in \{0,1\} \}$

6. Write a regular grammar to generate the language matched by /`Whiske?y`/

   S ::= WhiskTy
   T ::= e | ε

7. Write a regular grammar to generate decimal numbers; the relevant regex is /`[1-9][0-9]*(\.[0-9]*[1-9])?`/.
   You may find it useful to use notation resembling D ::= 0 | 1 | … | 9 to denote an set of ten production rules.

   S ::= 1T | 2T |…..| 9T
   T ::= 0T | 1T |….| 9T | U
   U ::= .V | ε
   V ::= 0V | 1V |….| 9V| W
   W ::= 1 | … | 9

8. Give a **context free** grammar for the language $L = \{ a^n b^n \mid n \in \mathbb{N}\}$.

   S ::= aSb | ab | ε

9. Give a context free grammar for the set of properly bracketed sentences over alphabet { `0` , `(` , `)` }. So `0` and `((0))` are fine, and `00` and `0)` are not fine.

   S ::= 0 | (0(S)) | (S) | ε

10. Write a context free grammar to generate possibly bracketed expressions of arithmetic with + and * and single digit numbers.
So `0` and `(0)` and `0+1+2` and `(0+9)` are fine, and `(0+)1` and `12` are not fine.

A ::= 0|1|2|3|4|5|6|7|8|9
S ::= (S) | S+S | S*S| A

11. Give a grammar for palindromes over the alphabet { `a` , `b` }.

S ::= aSa | bSb | a | b | ε

12. A parity-sequence is a sequence consisting of 0s and 1s that has an even number of ones. Give a grammar for parity-sequences.

S ::= 0S | 1A0S | ε
A ::= 0A | ε

13. Write a grammar for the set of numbers divisible by 4 (base 10; so the alphabet is `[0-9]`). You might like to read this page on divisibility testing, first. You may use dots notation to represent a sequence, as in for example the sequence Z3,Z6,Z9,...,Z999 to mean "Z followed by some number divisible by three and strictly between 0 and 1000".

A1 ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
A2 ::= 12 | 16 | 20 |…| 96
A3 ::= 4 | 8

S ::= AA2 | A3 | ε
A ::= A1A | ε

14. Write a grammar for the set of numbers divisible by 3 (base 10; so the alphabet is `[0-9]`). So for example `0`, `003`, and `120` should be in your language, and `1`, `2`, and `5` should not

D_0 = 0 | 3 | 6 | 9
D_1 = 1 | 4 | 7
D_2 = 2 | 5 | 8

S_0 ::= D_0S_0 | D_1S_2 | D_2S_1 | D_0
S1 ::= D_0S_1 | D_1S_0 | D_2S_2 | D_1
S2 ::= D_0S_2 | D_1D_1 | D_2S_0 | D_2

15. *(Unmarked)* Write a grammar for the set of numbers divisible by 11 (base 10; so the alphabet is `[0-9]`).

More on Grammars ….

16. State which of the following production rules are *left-regular*, *right-regular*, *left-recursive*, *right-recursive*, *context-free* (or more than one, or none, of these):

1. Thsi ⟶ This (a rewrite auto-applied by Microsoft Word).
   None of these
2. Sentence ⟶ Subject Verb Object. (Here Sentence, Subject, Verb, and Object are non-terminal symbols.)
   Context free
3. X ⟶ Xa. Left recursive
4. X ⟶ XaX. Context free

17. What is the object language generated by X⟶Xa (see Lecture 2)? Explain your answer.

The language generated by this grammar will be a set of string of terminals. 'a' is the terminal.

X -> Xa -> Xaa -> Xaaa -> ……

18. Construct context-free grammars that generate the following languages:

1. $(ab|ba)^*$,

   S -> abS | baS | ε

2. $\{(ab)^n a^n | n \geq 1\}$,

   S -> abSa | aba

3. $\{w \in \{a,b\}^* | \text{ w is a palindrome}\}$

   S ::= aSa | bSb | a | b | ε

4. $\{w \in \{a,b\}^* | \text{ w contains exactly two bs and any number of as }\}$

   S -> XbXbX | ε
   X -> aX | ε

5. $\{a^n b^m \mid 0 \le n \le m \le 2n\}$

   S -> aST | ε
   T -> b | bb

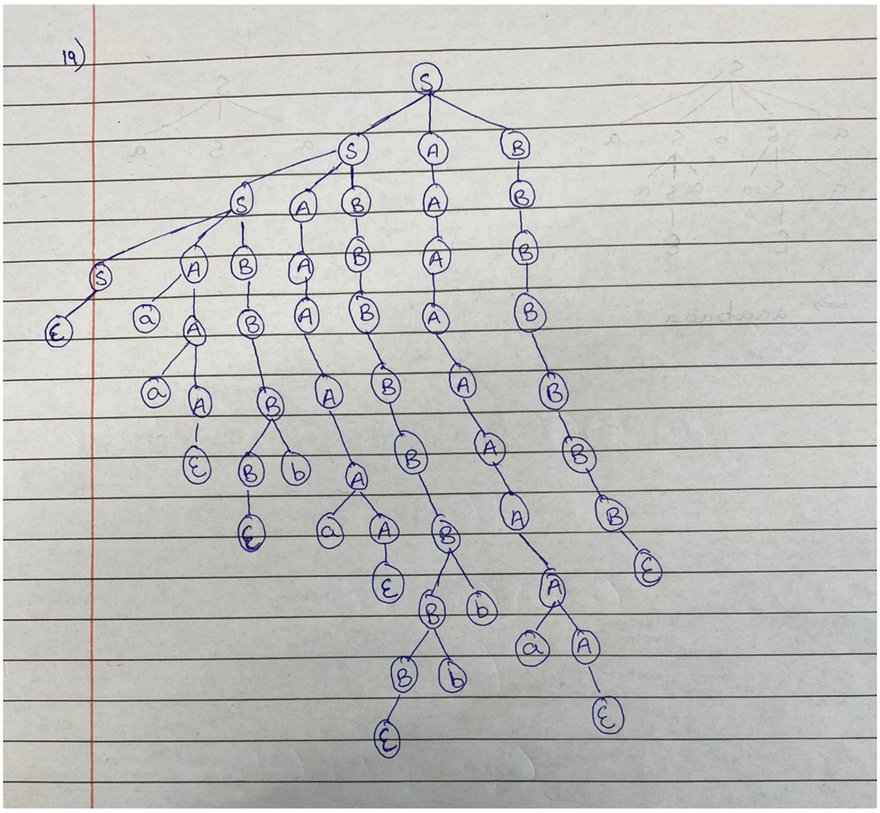
19. Consider the grammar $G=(\{S,A,B\},\{a,b\},P,S)$ with productions

$S \longrightarrow SAB \mid \varepsilon$
$A \longrightarrow aA \mid \varepsilon$
$B \longrightarrow Bb \mid \varepsilon$

1. Give a leftmost derivation for aababba.

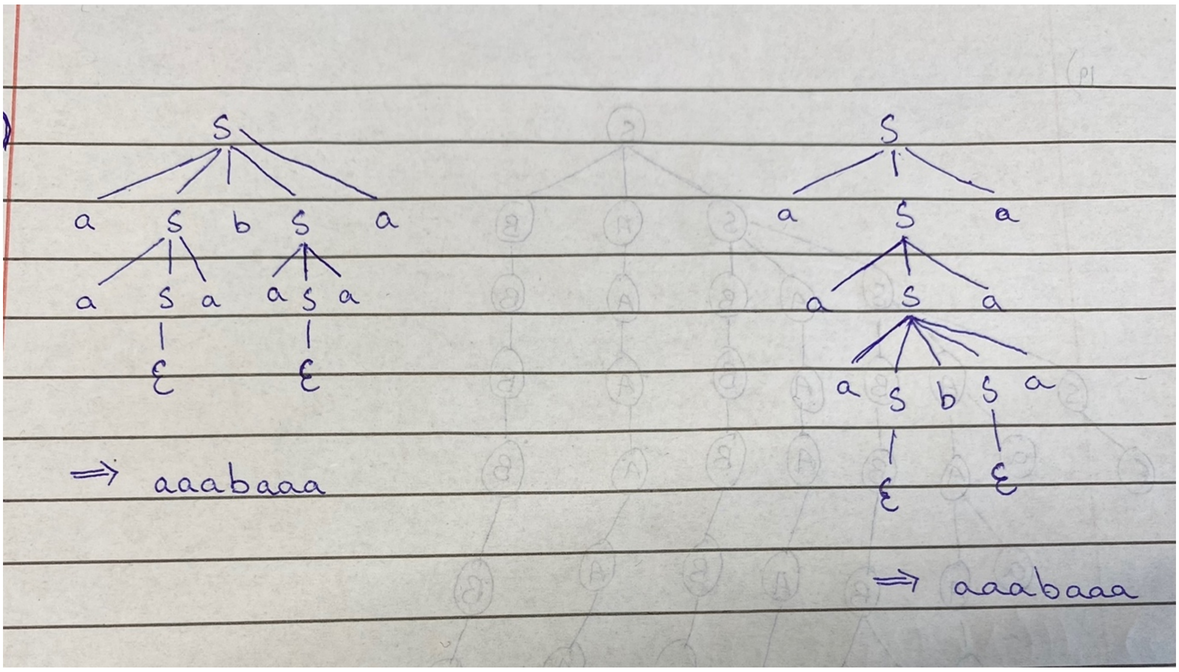
   S -> SAB
   S -> SABAB
   S -> SABABAB
   S -> ABABAB
   S -> aABABAB
   S -> aaABABAB
   S -> aaBABAB
   S -> aaBbABAB
   S -> aabABAB
   S -> aabaABAB
   S -> aabaBAB
   S -> aabaBbAB
   S -> aabaBbbAB
   S -> aababbAB
   S -> aababbaAB
   S -> aababba

2. Draw the derivation tree corresponding to your derivation.

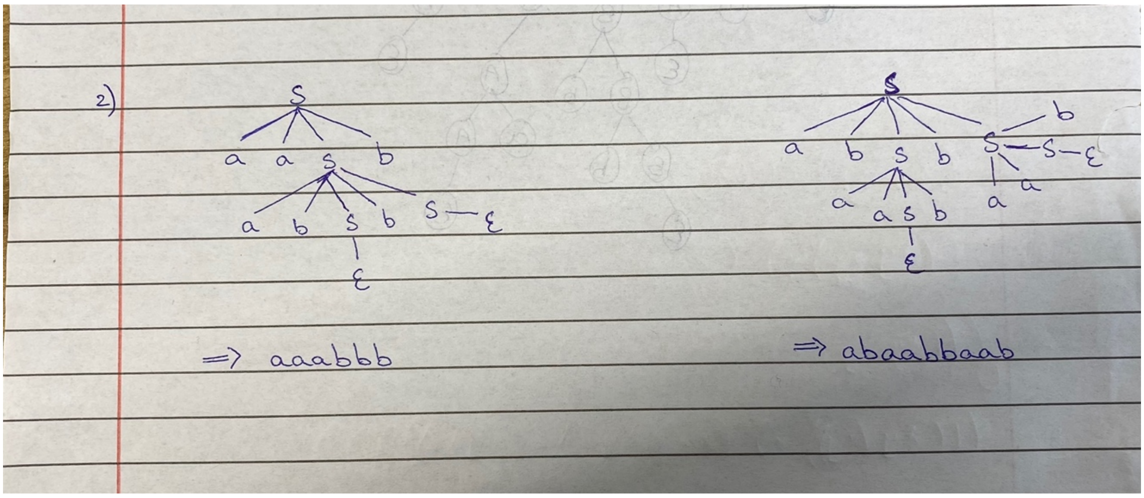


20. State with proof whether the following grammars are ambiguous or unambiguous:

1. G=({S},{a,b},P,S) with productions S ⟶ aSa | aSbSa | ε



2. G=({S},{a,b},P,S) with productions S ⟶ aaSb | abSbS | ε



21. Rewrite the following grammar to eliminate left-recursion:

exp ⟶ exp + term | exp – term | term

<exp> ::= <term><exp’>
<exp’> ::= +<term><exp’> | -<term><exp’> | ε

22. Write a grammar to parse arithmetic expressions (with + and ×) on integers (such as 0, 42, -1). This is a little harder than it first sounds because you will need to write a grammar to generate correctly formatted positive and negative numbers; a grammar that generates -01 is unacceptable. You may use dots notation to indicate obvious replication of rules, as in “$\mathrm{D}\longrightarrow 0|\cdots|9$”, without comment.
(Note that a parser is not an evaluator. This question is **not** asking you to write an evaluator. Also note that the grammar only needs to be unambiguous if the question asks you to provide an unambiguous grammar.)

23. Write two distinct parse trees for $2+3*4$ and explain in intuitive terms the significance of the two different parses to their denotation.