Part II: Grammars

Submission: On Canvas, under Assignments
Deadline: Week 5, Thurs 16th Feb 2023

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*/

The language determined is $\{a^n b^m \mid n,m >=0\}$

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

S ::= a | aS | aT | T
T ::= b | bT

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

The language determined is $\{(ab)^n \mid n>=0\}$

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

S ::= ε | ab | abS

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

The language determined is {Whiske^{n}y | 1>=n>=0}

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

S ::= Whiskey|Whisky

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 ::= A | A.BA | AB
A ::= C | CBAB
C ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ε
B ::= 0 | ε

8. Give a **context free** grammar for the language L = { a^n b^n | n ∈ ℕ}.

S ::= ε | aSb

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 | (S)

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.

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

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

S ::= a | b | aSa | bSb | 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 ::= S1S1S | 0S | 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".

S ::= 0 | 4 | 8 | A00 | A04 | A08 | A12 | A16 | … | A96
A ::= 0 A | 1 A | 2 A | 3 A | 4 A | 5 A | 6 A | 7 A | 8 A | 9 A | AA | 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

S → 0A | 3A | 6A | 9A | 1B | 4B | 7B | 2C | 5C | 8C
A → 0A | 3A | 6A | 9A | 1B | 4B | 7B | 2C | 5C | 8C | ε
B → 0B | 3B | 6B | 9B | 1C | 4C | 7C | 2A | 5A | 8A
C → 0C | 3C | 6C | 9C | 1A | 4A | 7A | 2B | 5B | 8B

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).

Context Free Grammar

2. Sentence ⟶ Subject Verb Object. (Here Sentence, Subject, Verb, and Object are non-terminal symbols.)

Context Free Grammar

3. $X \longrightarrow Xa$.

Left Recursive because X is the non-terminal.

4. $X \longrightarrow XaX$.

Context Free Grammar

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

The object language generated by this can be {a, aa, aaa, aaaa, …} as object language is the set of strings of terminals.

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

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

S ::= ab | ba | abS | baS | ε

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

S ::= aba | abSa

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

S ::= a | b | aSa | bSb | ε

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

S ::= S'bS'bS'
S' ::= aS' | ε

5. $\{a^n b^m | 0 \leq n \leq m \leq 2n\}$

S ::= abS | baS | abbS | bbaS | babS | ε

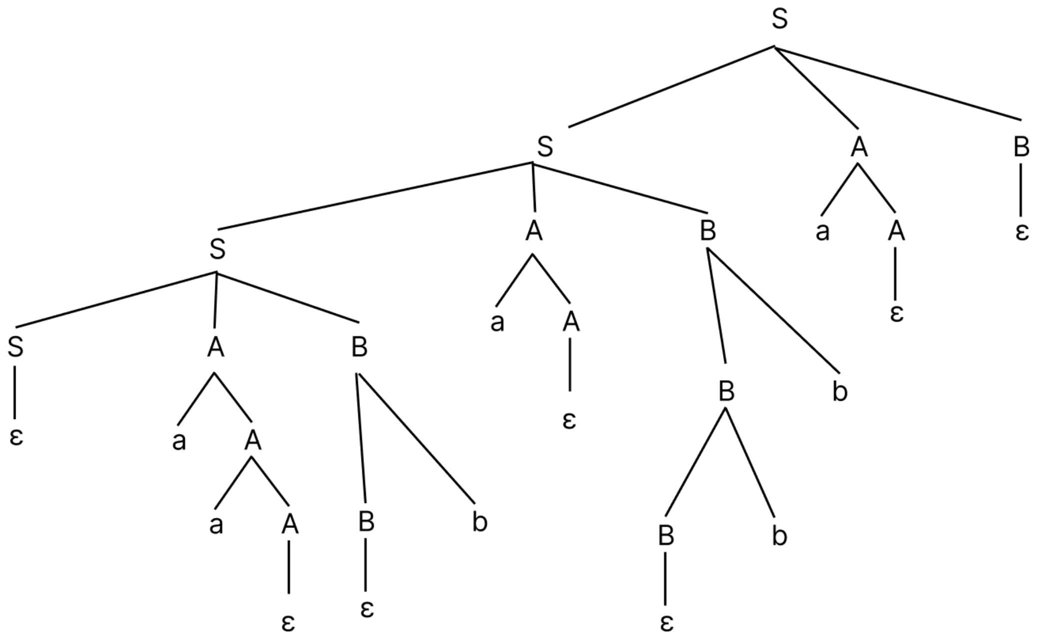
19. Consider the grammar $G=(\{S,A,B\},\{a,b\},P,S)$ with productions

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

1. Give a leftmost derivation for aababba.

S => SAB
=> SABAB
=> SABABAB
=> εABABAB
=> aABABAB
=> aaABABAB
=> aaεBABAB
=> aaBbABAB
=> aaεbABAB
=> aabaABAB
=> aabaεBAB
=> aabaBbAB
=> aabaBbbAB
=> aabaεbbAB
=> aababbaAB
=> aababbaεB
=> aababbaε
=> 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

The above production rule is ambiguous because when we try to generate string “aababaaa”, this can be produced by two methods.

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

The above production rule is unambiguous because when we try to generate string “ababbb”, this can be produced by only one method.

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.)

S ::= -AB | AB
B ::= +AB
B ::= -AB
B ::= ε
A ::= 1 | 2 | 3| 4 | 5 | 6 | 7 | 8 | 9 | AD
D ::= 0 | 1 | 2 | 3| 4 | 5 | 6 | 7 | 8 | 9

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.

Using the Grammar

<exp> ::= ⟨digit⟩ | ⟨digit⟩+⟨exp⟩ | ⟨exp⟩∗⟨digit⟩

⟨digit⟩ ::= 2|3|4

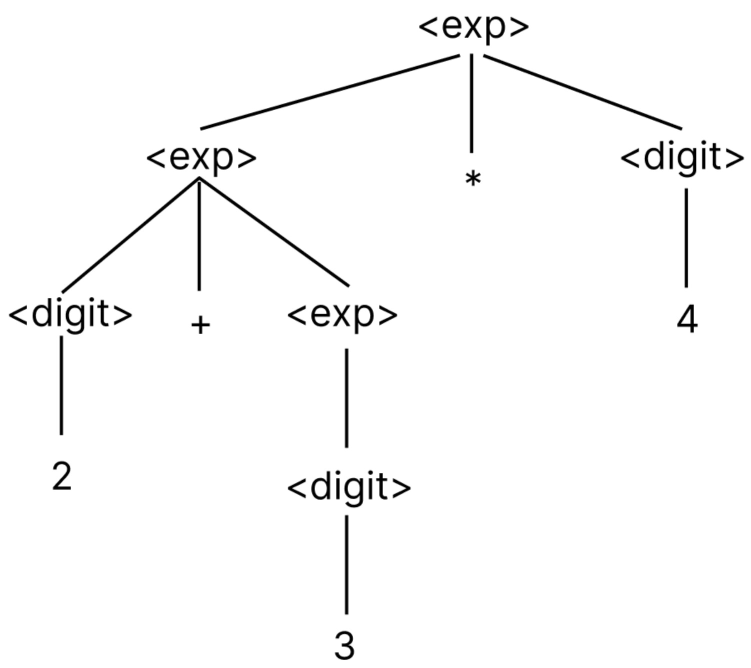


*Figure 1:Figure for 2 + 3 * 4*

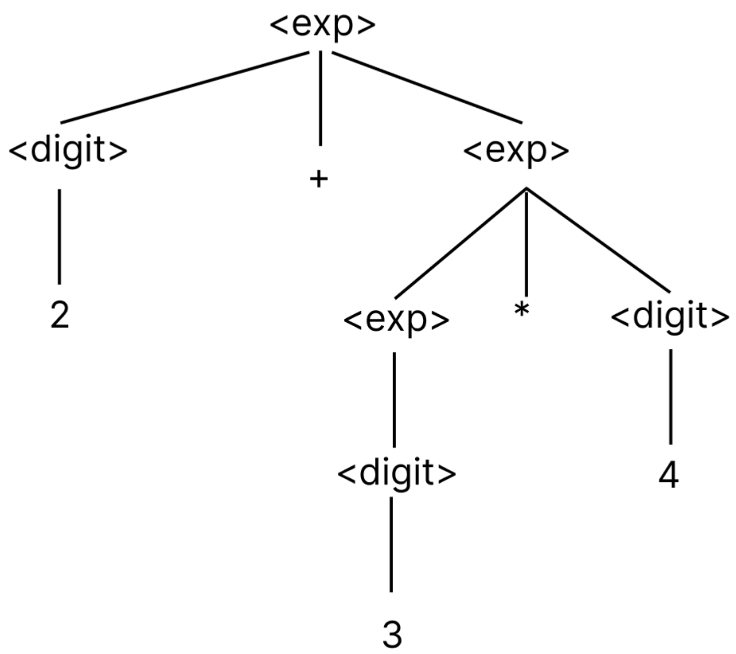


*Figure 2: For 2 + 3 * 4*