CSCI415 Compiler Design Fall2022

Nile University
جامعة النيل

Name : .....

ID : .....

Quiz 3

1) Explain in English the language described by the grammar:

- Terminal symbols = { a , b }

$S \rightarrow aSbS$

$S \rightarrow bSaS$

$S \rightarrow \varepsilon$

(hint : generate some strings by the grammar to check the language)

Answer : generate strings with equal numbers of ‘a’ and ‘b’

- Terminal symbols = { a , b }

$S \rightarrow aSbb$

$S \rightarrow abb$

(hint : generate some strings by the grammar to check the language)

Answer : generate strings with number of ‘b’ double number of ’a’

Terminal symbols = { a , b }

$S \rightarrow aSb$

$S \rightarrow SS$

$S \rightarrow \varepsilon$

(hint : generate some strings by the grammar to check the language)

Answer : generate strings that start with ‘a’ , number of ‘a’ equal number of ‘b’ and ‘b’ can’t precedes ‘a’

Terminal symbols = { (,) }

$S \rightarrow A$

$A \rightarrow (A)$

$A \rightarrow A A$

$A \rightarrow ( )$

(hint : generate some strings by the grammar to check the language)

Answer : generate strings with balanced brackets whenever a bracket opened it’s closed

2) For the following grammar :

Terminal symbols = { 0 , 1 }

S → A1B

A → 0A | ε

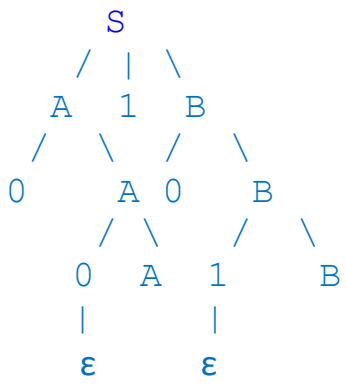
B → 0B | 1B | ε

The Input : **00101**

a- Construct left most derivation *(3 points)*

b- From parse tree of the input *(3 points)*

S => A1B => 0A1B => 00A1B => 001B => 0010B => 00101B => 00101



For the following grammar : *(6 points)*

Terminal symbols = { 0 , 1 }

S → A

A → 0A1A

A→ 0A

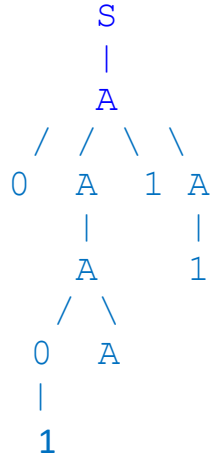
A → 1

The Input : **00111**

a- Construct left most derivation *(3 points)*

b- From parse tree of the input *(3 points)*

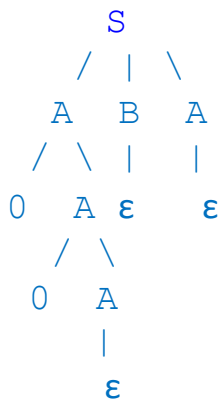
S => A => 0A1A => 00A1A => 0011A => 00111



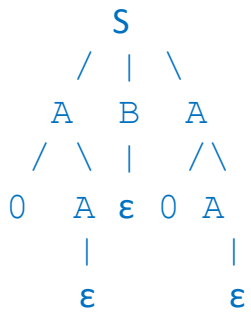
For the following grammar :
Terminal symbols = { 0 , 1 }

S → ABA
A → 0A | ε
B → 1B | ε
The Input : **00**
a- Construct left most derivation
b- From parse tree of the input

S => ABA => 0A => 00



another tree as it ambiguous grammar



For the following grammar : (6 points)

Terminal symbols = { 0 , 1 }

S → AB |00B

A → 0 | 0A | ε

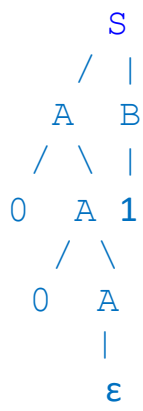
B → 1

The Input : **001**

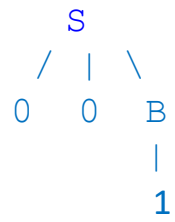
a- Construct left most derivation

b- From parse tree of the input

S => AB => 0AB => 00B => 001



S => 00B => 001



Additional Question :

What CFG for Boolean expressions !(not) , &(and) , |(or)

Answer :

not have the highest precedence then and lastly or

so

Terminal symbols = { ! , or , and , true, false }

E → E or E' | E'

E' → E' and F | F

F → ! E | true | false