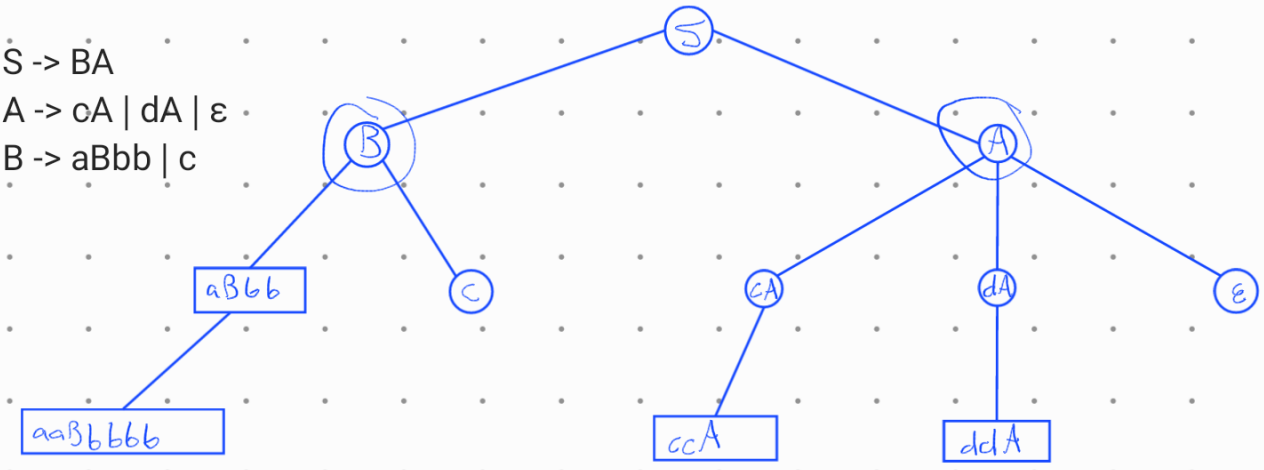
Exercise 4.1 Exercise 4.2 – obligatory (6 points)

Let the following context-free grammar G with start symbol S be given:

S -> BA
A -> cA | dA | ε
B -> aBbb | c



a) Which of the following words are derivable? If possible, give a derivation for each.

(1) cdc

S-> BA
BA -> cA
cA -> cdA
cdA -> cdcA
cdcA -> cdc

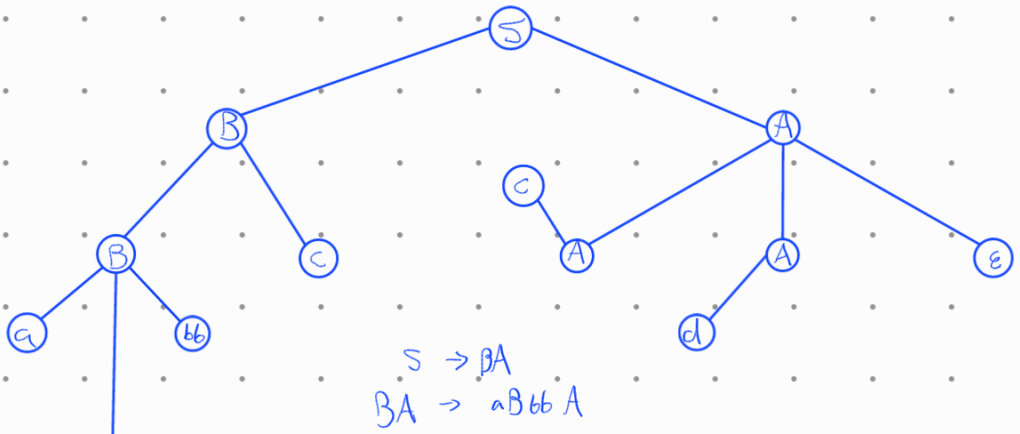
(2) acbb

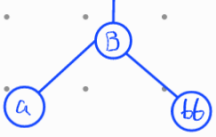
S-> BA
BA -> aBbbA
aBbbA -> acbbA
acbbA -> acbb

(3) accba

S-> BA
~~BA -> aBbbA~~
not possible

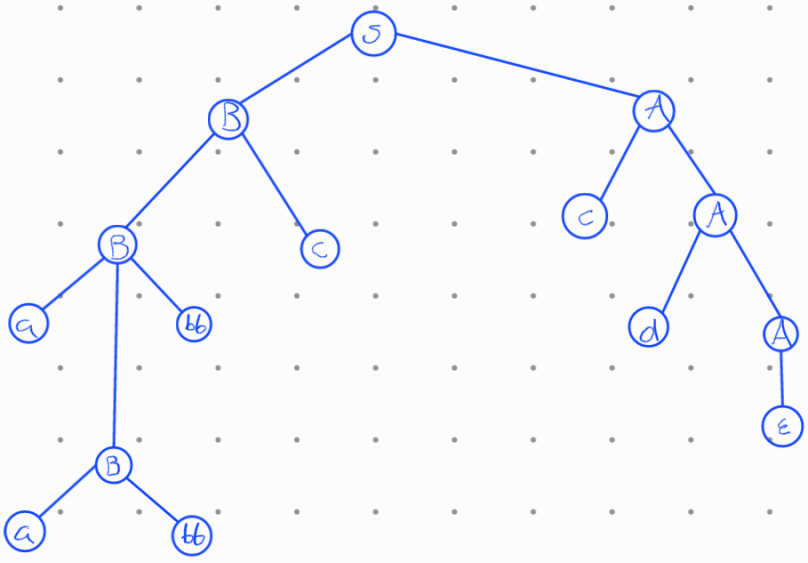
b) Give a derivation tree for the string aacbbbbcd.





aBbbA → a(aBbb)bbA
a(aBbb)bbA → aacbbbbA
aacbbbbA → aacbbbbcA
aacbbbbcA → aacbbbbcdA
aacbbbbcdA → aacbbbbcd

c) Give a rightmost derivation for the string aacbbbbcd.



S → BA
BA → BcA
BcA → BcdA
BcdA → Bcd
Bcd → aBbbcd
aBbbcd →

d) (optional, +2 bonus points) Which language is generated by the grammar G?

S -> BA
A -> cA | dA | ε
B -> aBbb | c

L = (a, c)

Exercise 4.3

Define a context-free grammar for the language of the following regular expression:
a(b|cc)* (a|ε)
S → aAB
A → | bA | ccA
B → a | ε

Exercise 4.4 – obligatory (6 points)

a) Define a context-free grammar for the language L
L = { dc^(2n)ba^(n)| n >= 0 } U { ac^kb | k > 0 }.
S → dA | aBb
A → ccAa | b
B → cB | ε

b) Specify a context-free grammar for logical expressions. The expressions can be build from the variable x, y, z, and the logical operators !, &&, and || (similar to Java). Any (sub-)expression can be encompassed by parenthesis. Here are some examples:

x
!z
!y || x
(x)
!(z && y || z)
(x || !(y && !!z))

Expr → letters |
Expr && Expr |
Expr || Expr |
! Expr |
(Expr)
Letters →
x | y | z

S → CAB
A → B&&B | B||B | (A) | ε
B → x | y | z
C → ! | ε

Exercise 4.5 (tricky!)

Let S = {0,1}. Specify context-free grammars for the following languages:

a) L1 = { 0^(n)1^(m) | n <= m}

S → 0S1 | B
B → 1B | ε

b) L2 = { 0^(n)1^(n+m)0^(m) | n > 0, m > 0}

S → AB |
A → 0A1| ε
B → 0B1 | ε

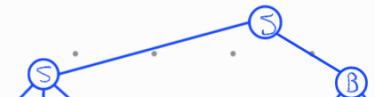
c) L3 = {w ϵ S* | |w|0 = |w|1 } , i.e. the language of all strings that contain the same number of zeros and ones.

Exercise 4.6 - obligatory (3 points)

Let the following context-free grammar G with start symbol S be given:

S → SaS | B
B → b | c

a) Draw a parse tree for the string bacab.



b) Show that grammar G is ambiguous. Not Ambiguous

LD

S a S → b a S

b a S → b a S a S

b a S a S → b a c a b

RD

S a S → S a b

S a b → S a S a b

S a S a b → S a c a b

→ b a c a b

Exercise 4.7

The Ada 2012 programming language reference manual defines the syntax of if statements using EBNF:

If_Statement → if Condition then Sequence_of_Statements
(elsif Condition then Sequence_of_Statements)*
[else Sequence_of_Statements]
end if ;

Sequence_of_Statements → Statement +

Terminal symbols are printed as boldface text. Condition and Statement are nonterminal symbols defined elsewhere. Square brackets enclose optional items.

Draw a syntax diagram (railroad diagram) for this section of the Ada grammar.

Exercise 4.8

Transform the following EBNF grammar into a context-free grammar.

S → (aSb)* [Ac]

A → (ab | Ad) bb

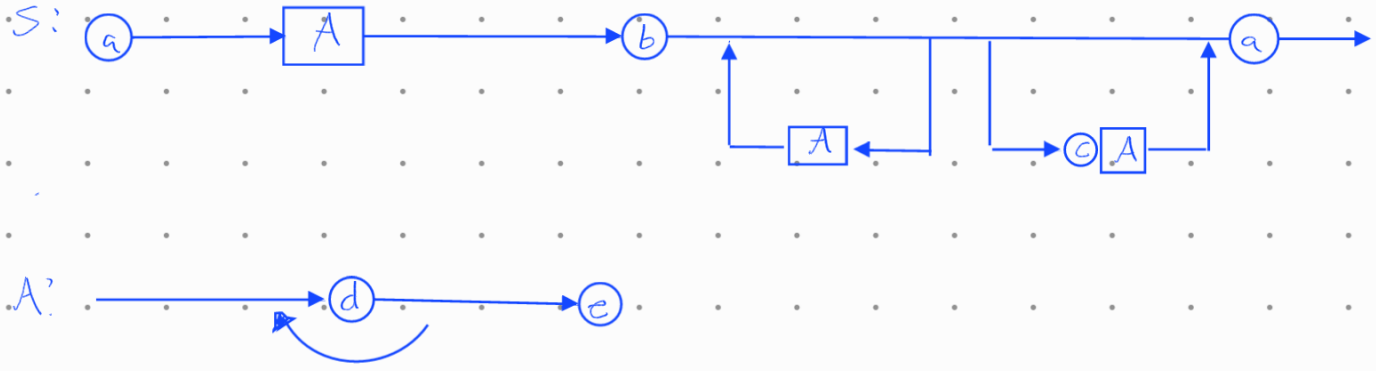
Exercise 4.9 - obligatory (5 points)

The following EBNF grammar is given, where [...] indicates optional parts:

S → aAbA*[cA]a

A → d+e

a) Represent the grammar as a syntax diagram (railroad diagram).



b) Convert the EBNF grammar to an equivalent context-free grammar.

S → aAbA*[cA]a

S → aAbXZa

X → AX | ε

Z → cA | ε

A → d+e

A → d+e

A → dYe

Y → dY | ε