CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: Elena Corpus

**Home Assignment 4: 143/150 points + 10 points (optional)**

Q1. [18/20] For a given language L = {*anb****2n*** | *n* ³ 0 is even}.

1. [8/8] Give a CFG that accepts L.

S -> aaSbbbb | **ε**

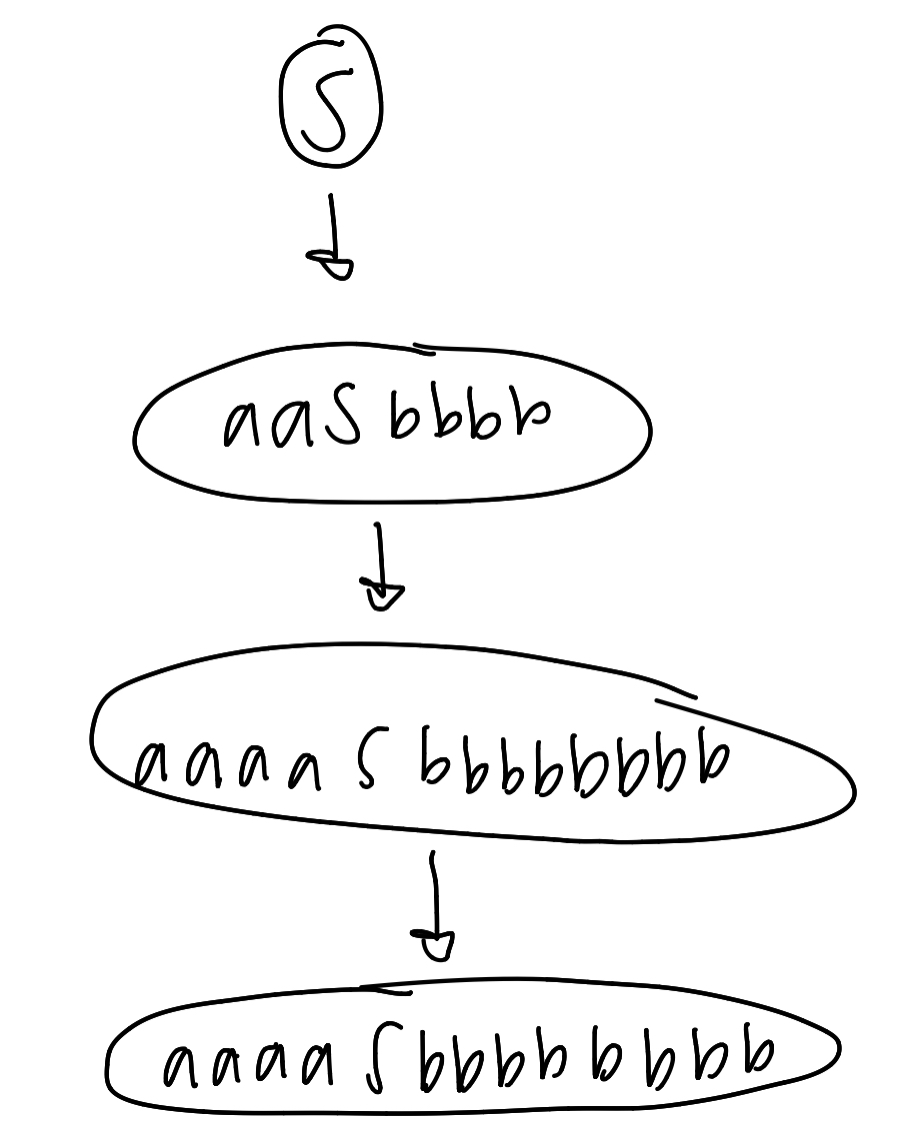
1. [6/6] Show the sequence of derivations for the acceptance of *aaaabbbbbbbb* by G in (1).

S -> aaSbbbb

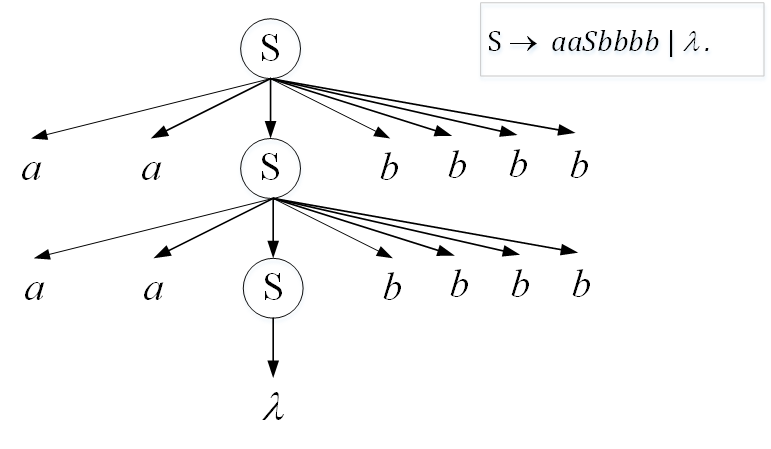
-> aaaaSbbbbbbbb

-> aaaabbbbbbbb

1. [4/6] Draw a derivation tree for *aaaabbbbbbbb*.



* See the attached answer



Q2. [30] Construct a CFG for the following languages where *n*, *m, k* ³ 0.

1. [10/10] L1 = { *anbn* | *n* is a multiple of *3* }

(a3)x(b3)x

(aaa)x(bbb)x

S -> aaaSbbb | **ε**

1. [8/10] L2= { *anbmck* | *k* = *n+m* }

= Anbmck

= anbmcn+m

= anbmcncm

= ((ancn)/A) ((bmcm)/b)

S -> AB

A -> aAc | ε

B -> bBc | ε

S → *a*Sc | B, B → *b*Bc | λ

1. [10/10] L3 = { *anbm* | *n =* *m –*1 }

N = m – 1

N + 1 = m

= Anbn+1

= (anbn)/x \* b

S -> xb

X -> axb | ε

1. [10/10, optional] L4 = { *anbmck* | *n=m* or *m* £ *k* }

N = m or m <= l

(anbn/u)(ck/v) or (an/x) (bmckck/y)

S -> UV | XY

U -> aUb | ε

V -> cV | ε

X -> aX | ε

Y -> bYc | c | bAc | ε

A -> cA | c

Let’s split L2 into 2 parts: L2A = { *anbmck* | *n=m* } and L2B = { *anbmck* | *m* ≤ *k* } where L2A = L(G2A) and L2B = L(G2B) with the grammars G1 and G2.

Then, use S → S1 | S2 where S1 derives L2A with G1 and S2 derives L2B with G2.

G2A: S1 → *a*S1*b* | λ, A → *a*A | λ

G2B: S2 → *b*S2*c*| C, C → *c*C | λ

Union G2A and G2B with the new start symbol S: S → S1 | S2

Therefore, the CFG, G2, s.t. L2 = L(G2) is

{ S → S1C| AS2

S1 → *a*S1*b* | λ,

S2 → *b*S2*c*| C,

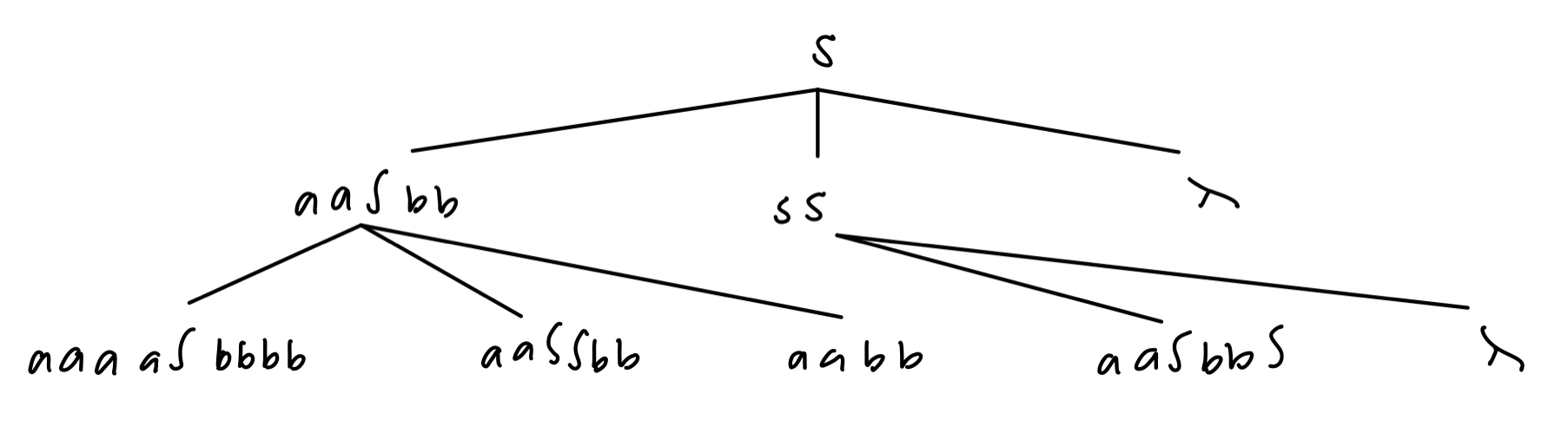
A → *a*A | λ

C → *c*C | λ }

Q3. [5/10] Give the language L that is generated by the given grammar, in a formal expression.

S ® *aa*S*bb* | SS |l.

e.g.) L = { *w* Î {*a, b*}\* | *na*(*w*) = 2*nb*(*w*) }



L = (aa)\* (bb)\*

Because a and b are in pairs

L = {w ∈{a, b}\* | |w| is even, #(*a*)*w* = #(*b*)*w* and #(*a*)*v* > #(b)*v* where *v* is any prefix of *w*.}.

= {(*a*n*b*n)m | *n* is even, *m* >= 0 and #(*a*)*v* > #(b)*v* where *v* is any prefix of *w*.}.

= {(*a*2n*b*2n)m | *n*, *m* >= 0 and #(*a*)*v* > #(b)*v* where *v* is any prefix of *w*.}.

Notice that it’s so similar to the grammar of Example 5.4.

– If *aa* is replaced by *c* and *bb* by *d*, then S → *c*S*d* | SS |λ.

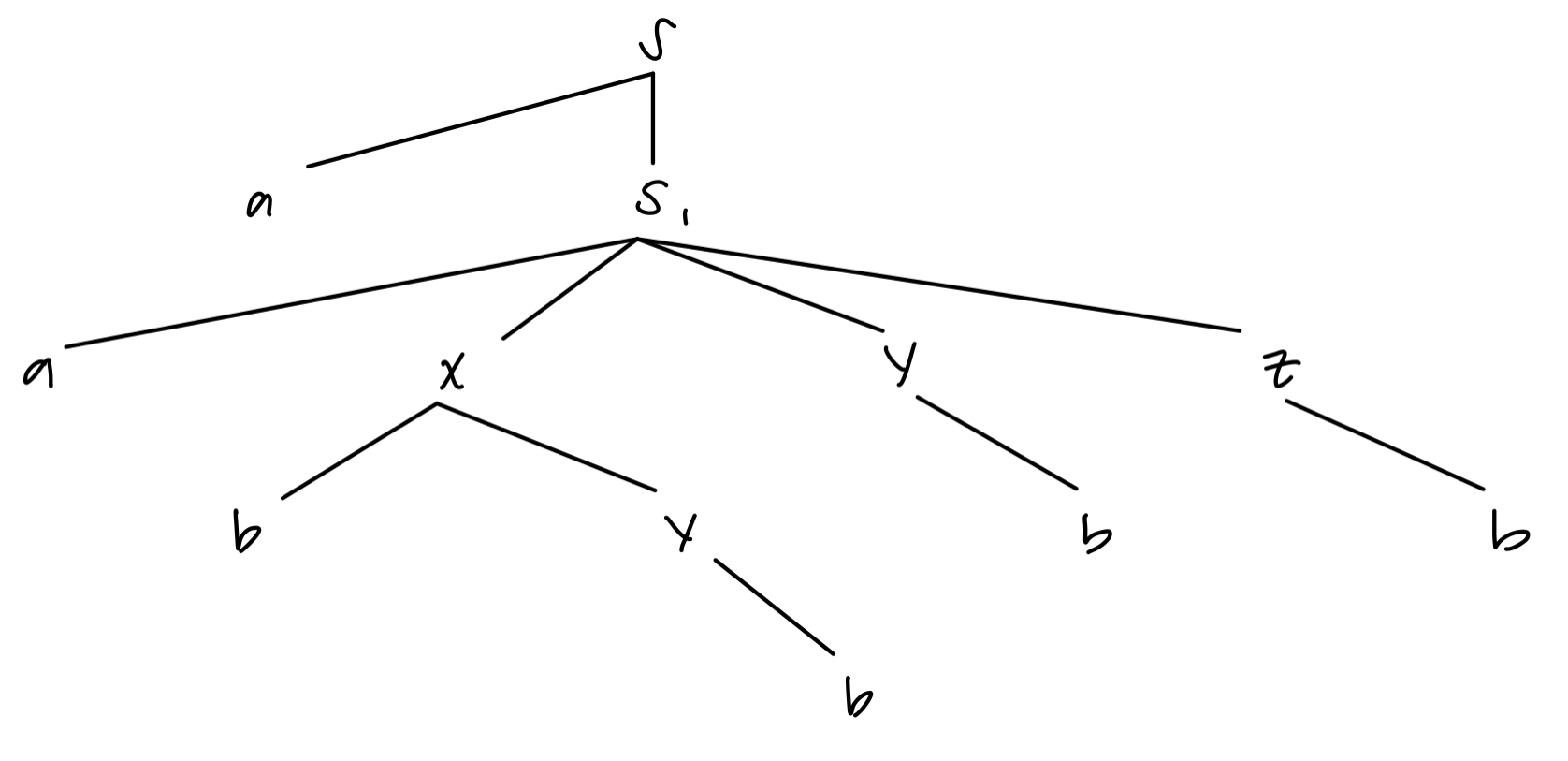
Q4. [10/10] Find an s-grammar for L = {*anb****2n*** | *n* ³ 2}.

1 – a and 2 – b’s

X, Y, Z

Start at S

AaSbbbb | ε l



S -> aS1

S1 -> aXYZ

X -> bY | aXYZ

Y -> b

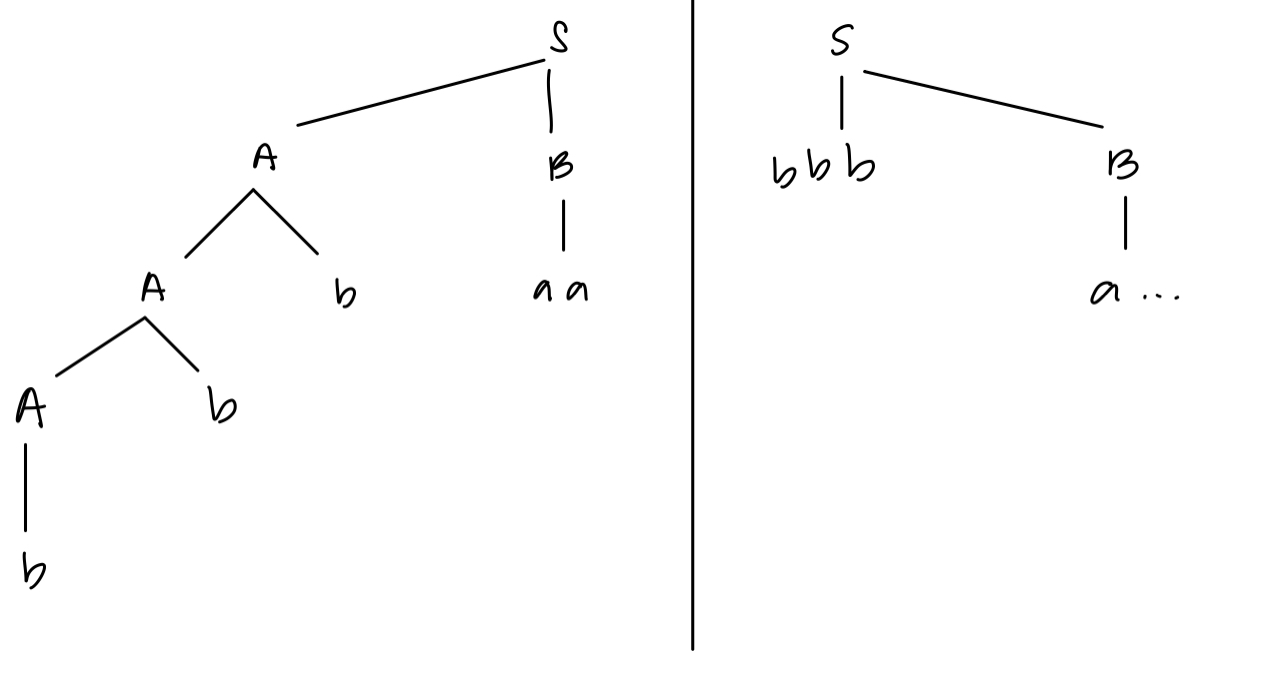
Z -> b

S → *a*S1B, S1 → *a*ABB, A → *a*ABB | b, B → *b*

Q5. [14/20] For a grammar G with the productions where G = ( {S, A, B}, {*a, b*}, S, P ) with productions

S ® AB | *bbbB*, A ® *b* | A*b*, B ® *a..*

1. [8/8] Show that the grammar G is ambiguous.



Because there are two separate trees, it shows that it is ambiguous

1. [3/6] Give language L that is generated by G, L = L(G), in a formal expression (including a regular expression).

S -> AB | bbbB

A -> b | Ab => bAb => b+ (at least one b)

B -> a..

AB => b\_a..

S -> AB | bbba..

S -> b+a.. | bbba..

(b++bbb)a..

L(*bb*\**a*) = { bna | n ≥ 1 } = *bb\*a*

1. [3/6] Can you construct an unambiguous grammar that is equivalent to G? Otherwise, show that G is inherently ambiguous.

If we substitute “A” in “S” it equals

S -> b+ | bbbB

B-> a..

Thus being unambiguous because removing “A,” the ambiguous is removed.

S → *bA*, A → *bA* | *a*

Any other unambiguous CFG that is equivalent to G that generates *bb\*a*

e.g.) S ® AB, A ® *b* | A*b*, B ® *a.*

S → A*a*, A → b | Ab

Q6. [33/35] In the given grammar G, generate the simplified equivalent grammar by eliminating the following productions through (1) – (3).

G = ( {S, A, B, C}, {*a, b*}, S, P ) with productions

S ®*b*AA | *b*B, A ® *a*A| *aaC* , B ® *bb*B | *l,* C ® A

1. [10/10] Eliminate the l-productions

S -> *b*AA | *b*B | b

A -> *a*A| *aaC*

B -> *bb*B | bb

C -> A

1. [8/10] Eliminate the Unit-productions from (1)

S -> *b*AA | *b*B | b

A -> *a*A| *aaC*

B -> *bb*B | bb

C → A is the only unit-production. So, removing it is:

S → bB | b | bAA,

A → *a*A | *aa*A,

B → bbB | bb,

1. [10/10] Eliminate the useless productions (2), so that give the simplified equivalent grammar.

S -> bB | b

B -> bbB | bb

1. [5/5] Give the language L that is generated by this grammar, L = L(G), in a formal expression (including a regular expression).

(bb)\*b

Containing b’s of odd length

L((bb)\*b) = L(b(bb)\*), L = {b2*n*+1 | *n* ≥ 0}

Q7. [15/15] Convert the given grammar into Chomsky Normal Form (CNF).

S ® AB | *a*B, A ® *abb* | *l* , B ® *bb*A

Hint: Eliminate the l-productions and/or any unit-production prior to their conversion into CNF.

***Removing λ productions***

S -> AB | aB | B  
A -> abb  
B -> bbA | bb

***Removing unit productions***

S -> AB | aB | bbA | bb  
A -> abb  
B -> bbA | bb

***Removing mixed rules***

S -> AB | WB | VVA | VV  
A -> WVV  
B -> VVA | VV  
W -> a  
V -> b

***Change to final CNF***

S -> AB | WB | CA | VV  
A -> WC  
B -> CA | VV  
W -> a  
V -> b  
C -> VV

Q8. [10/10] Convert the given grammar into Greibach normal form.

S ® *a*S*b* | *ab* | *bb*

S -> aSTb

S -> aTb

S -> bTb

Tb -> b

=> S -> aSTb | aTb | bTb

Tb -> b