Assignment 4

Chapter 6 and Chapter 7

100 Points

1. Eliminate the variable B from the following Grammar: (5 points)

S → aSB | bB, B → aA | b  
  
S-> aSaA | aSb |baA | bb  
  
Simple substitution

2. Eliminate all useless productions from the grammar (5 Points)

S -> aS|AB| **λ**

A -> bA,

B -> AA.

What language does this grammar generate?  
  
Useless productions cannot be reached from start string or cannot derive terminal string. A and B as productions produce infinite loops.  
  
S -> aS| **λ**

This language would be L = {a^n | n>0}

3. Transform these grammars into Chomsky Normal Form: (20 points)

a) S🡪 ASA | aB

A🡪B|S

B🡪b| **λ**Step 1.) Eliminate start symbol from right hand side  
Step 2.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

Step 3.) Eliminate any terminals present with non terminals (time to create new variables)  
  
Step 4.) Eliminate productions of more than 2 non terminals (create even more variables)

Step 1.) Eliminate start symbol from right hand side

S0 -> ASA | aB

S🡪 ASA | aB

A🡪B|S

B🡪b| **λ**

Step 2.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

S0 -> ASA | ab

S🡪 ASA | ab

A🡪 b | ASA | ab

Step 3.) Eliminate any terminals present with non terminals (time to create new variables)  
S0 -> ASA | XY

S🡪 ASA | XY

A🡪 b | ASA | XY  
X->a

Y->b

Step 4.) Eliminate productions of more than 2 non terminals

S0 -> AZ | XY

S🡪 AZ | XY

A🡪 b | AZ | XY  
X->a

Y->b

Z -> SA

b) S → aAD

A → aB | bAB

B → b

D → d

Step 1.) Eliminate start symbol from right hand side

S → aAD

A → aB | bAB

B → b

D → d

Step 2.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA  
S → aAD

A → aB | bAB

B → b

D → d

Step 3.) Eliminate any terminals present with non terminals (time to create new variables)  
S → XAD

A → XB | BAB

B → b

D → d  
X -> a

Step 4.) Eliminate productions of more than 2 non terminals (create even more variables)

S → XZ

A → XB | BY

B → b

D → d  
X -> a

Y -> AB  
Z -> AD

4. Transform these grammars into Greibach Normal Form: (20 Points)

a) S → XY | Xn | p

X → mX | m

Y → Xn | o  
  
terminals: m,n,o,p

Step 1.) Convert to Chomsky Normal Form  
Step 2.) Eliminate start symbol from right hand side  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)  
Step 6.) Number all non terminals  
Step 7.) Eliminate left recursion.  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right  
Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
  
  
  
  
Step 1.) Convert to Chomsky Normal Form  
S → XY | Xn | p

X → mX | m

Y → Xn | o  
Step 1.) Eliminate start symbol from right hand side  
S → XY | Xn | p

X → mX | m

Y → Xn | o  
Step 2.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

S → XY | Xn | p

X → mX | m

Y → Xn | o

Step 3.) Eliminate any terminals present with non terminals (time to create new variables)  
S → XY | XN | p

X → MX | m

Y → XN | o

N -> n  
M -> m  
Step 4.) Eliminate productions of more than 2 non terminals (create even more variables)

S → XY | XN | p

X → MX | m

Y → XN | o

N -> n  
M -> m

Step 2.) Number all non terminals

A1 → A2A3 | A2A4 | p

A2 → A5A2 | m

A3 → A2A4 | o

A4 -> n  
A5 -> m

Step 3.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right

A1 → A2A3 | A2A4 | p

A2 → mA2 | m

A3 → A2A4 | o

A4 -> n  
A5 -> m  
  
A1 → mA2A3| mA3 | mA4 | mA2A4 | p

A2 → mA2 | m

A3 → mA2A4 |mA4| o

A4 -> n  
A5 -> m  
  
This is in Greibach normal form.

b) S → SA | eBA

A → AS | a

B → b

Step 1.) Convert to Chomsky Normal Form  
  
S → SA | eBA

A → AS | a

B → b  
  
Step 2.) Eliminate start symbol from right hand side  
S0 -> SA | eBA  
S → SA | eBA

A → AS | a

B → b  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

S0 -> SA | eBA  
S → SA | eBA

A → AS | a

B → b

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)

S0 -> SA | EBA  
S → SA | EBA

A → AS | a

B → b  
E -> e  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)  
S0 -> SA | EX  
S → SA | EX

A → AS | a

B → b  
E -> e

X-> BA  
Step 6.) Number all non terminals  
A1 -> A2A3 | A5A6  
A2 → A2A3 | A5A6

A3 → A3A2 | a

A4 → b  
A5 -> e

A6-> A4A3  
Step 7.) Substitute all non terminals with their internal productions if they are larger than the number of the non terminal to their right  
A1 -> A2A3 | A5A6  
A2 → A5A6A8A3 | A5A6

A3 → aA7A2 | a

A4 → b  
A5 -> e

A6-> A4A3

A7 -> A2A7| ‘LAMDA’

A8-> A5A6A8 |’LAMDA’  
  
Step 8.) Recursively substitute the leftmost variable until it starts with a terminal  
A1 -> eA6A8A3A3 | eA6A3|eA6  
A2 → eA6A8A3 | eA6

A3 → aA7A2 | a

A4 → b  
A5 -> e

A6-> bA3

A7 -> eA6A8A3A7| eA6A7 | ‘LAMDA’

A8-> eA6A8 |’LAMDA’

5. Please construct PDAs that accepts the languages defined by the grammar: (20 Points)

a) S 🡪 aSSSab | λ  
  
Step 1.) Convert to Chomsky Normal Form

S 🡪 aSSSab | λ  
Step 2.) Eliminate start symbol from right hand side

S0 -> aSSSab | λ  
S 🡪 aSSSab | λ  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA  
S0 -> aSSSab   
S 🡪 aSSSab

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)  
S0 -> ASSSAB   
S 🡪 ASSSAB  
A -> a

B -> b  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)  
S0 -> WZ  
S 🡪 WZ  
W -> XY

X -> AS  
Y -> SS  
Z -> AB  
A -> a

B -> b  
Step 6.) Number all non terminals  
A1 -> A3A6  
A2 🡪 A3A6  
A3 -> A4A5

A4 -> A7A2  
A5 -> A2A2  
A6 -> A7A8  
A7 -> a

A8 -> b  
Step 7.) Eliminate left recursion.  
A1 -> A3A6  
A2 🡪 A3A6  
A3 -> A4A5

A4 -> A7A2  
A5 -> A2A2  
A6 -> A7A8  
A7 -> a

A8 -> b  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right  
A1 -> A3A6  
A2 🡪 A3A6  
A3 -> A4A5

A4 -> aA2  
A5 -> A2A2  
A6 -> A7A8  
A7 -> a

A8 -> b  
Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
A1 -> aA2A5A6  
A2 🡪 aA2A5A6  
A3 -> aA2A5

A4 -> aA2  
A5 -> aA2A5A6A2  
A6 -> aA8  
A7 -> a

A8 -> b  
  
A drawing of a flower with numbers and a hand drawn sun

Description automatically generated

b) S 🡪 abSb| λ

Step 1.) Convert to Chomsky Normal Form  
Step 2.) Eliminate start symbol from right hand side  
S0 -> abSb| λ

S 🡪 abSb| λ  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

S0 -> abSb

S 🡪 abSb

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)  
S0 -> XY

S 🡪 XY

X -> AB  
Y -> SB

A -> a  
B -> b  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)  
S0 -> XY

S 🡪 XY

X -> AB  
Y -> SB

A -> a  
B -> b  
Step 6.) Number all non terminals  
1 -> 34

2 🡪 34

3 -> 56  
4 -> 26

5 -> a  
6 -> b  
Step 7.) Eliminate left recursion.  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right

Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
1 -> a64

2 🡪 a64

3 -> a6  
4 -> a646

5 -> a  
6 -> b  
  
A drawing of a flower with numbers and letters

Description automatically generated

6. Please construct PDAs that accepts the following languages: (20 Points)

a) ∑ = {a,b,c}: L= {wcwR : w∈ {a,b}\*} ( Hint: R means reverse, such as abcba)  
  
This language is asking for a language with the alphabet a,b,c. The first and back half of the language MUST be reverses of each other, and the middle must be C regardless. This means C is the terminal for the repeating section, which otherwise generates infinite strings of a and b on either side.   
S -> aSa | bSb| c  
Step 1.) Convert to Chomsky Normal Form  
Step 2.) Eliminate start symbol from right hand side  
S0 -> aSa | bSb| c  
S -> aSa | bSb| c  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA  
S0 -> aSa | bSb| c  
S -> aSa | bSb| c

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)  
S0 -> ASA | BSB| c  
S -> ASA | BSB| c  
A -> a

B->b  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)  
S0 -> AX | BY| c  
S -> AX | BY| c  
X ->SA

Y -> SB  
A -> a

B->b  
Step 6.) Number all non terminals  
1 -> 53 | 64| c  
2 -> 53 | 64| c  
3 ->25

4 -> 26  
5 -> a

6->b  
Step 7.) Eliminate left recursion.  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right  
Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
1 -> a3 | b4| c  
2 -> a3 | b4| c  
3 -> a35 | b45| c5

4 -> a36 | b46| c6  
5 -> a

6->b  
  
Looking at this you can eliminate step 2 as well, since it had to be substituted for all of its paths.   
1 -> a3 | b4| c  
3 -> a35 | b45| c5

4 -> a36 | b46| c6  
5 -> a

6->b  
  
A drawing of a sun

Description automatically generated

b) L = {a 2n b 3n | n ≥ 0}  
This language requires 3 bs for every 2 as

S -> aaSbbb

Step 1.) Convert to Chomsky Normal Form  
Step 2.) Eliminate start symbol from right hand side

S0 -> aaSbbb

S -> aaSbbb

Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)

S0 -> AASBBB

S -> AASBBB

A -> a

B -> b  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)

S0 -> YW

S -> YW

X -> AA  
Y -> XS  
Z -> BB

W -> BZ

A -> a

B -> b  
Step 6.) Number all non terminals  
1 -> 46

2 -> 46

3 -> 77  
4 -> 32  
5 -> 88

6 -> 85

7 -> a

8 -> b  
Step 7.) Eliminate left recursion.  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right until there is a terminal on the leftmost side.   
1 -> a726

2 -> a726

3 -> a7  
4 -> a72  
5 -> b8

6 -> b5

7 -> a

8 -> b  
  
Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
1 -> a726

2 -> a726

3 -> a7  
4 -> a72  
5 -> b8

6 -> b5

7 -> a

8 -> b

A drawing of a flower with numbers and a number

Description automatically generated

7. Show that the two grammars (5 Points)

S -> abAB|ba,

A -> aaa,

B -> aA|bb

And

S -> abAaA|abAbb|ba,

A -> aaa

are equivalent  
Grammars are shown to be equivalent if they generate the same strings. For the sake of laziness, I’m just going to back substitute each of the grammars into each other, and if they’re the same, then it’s a success.  
  
S -> abAB|ba,

A -> aaa,

B -> aA|bb

Becomes:  
S -> abAaA | abAbb |ba,

A -> aaa,

Becomes:  
Becomes:  
S -> abaaaaaaa | abaaabb |ba,

NOW LET US COMPARE IT TO:  
S -> abAaA|abAbb|ba,

A -> aaa  
Which becomes:  
S -> abaaaaaaa|abaaabb|ba,

A screenshot of a computer

Description automatically generated

8. Show the sequence of instantaneous descriptions for the acceptance of aabbbb by the pda of the language L = {anb2n >= 0} (5 points)  
Well, the first thing we need to do is create the PDA. Let’s define the grammar first  
  
S -> aSbb

Step 1.) Convert to Chomsky Normal Form  
Step 2.) Eliminate start symbol from right hand side  
S0 -> aSbb

S -> aSbb  
Step 3.) Eliminate single unit productions of non terminals and other useless productions NO LAMDA

Step 4.) Eliminate any terminals present with non terminals (time to create new variables)

S0 -> XY

S -> XY

X->aS

Y->bb  
  
Step 5.) Eliminate productions of more than 2 non terminals (create even more variables)

S0 -> XY

S -> XY

X->AS

Y->BB

A -> a

B-> b  
Step 6.) Number all non terminals  
1 -> 34

2 -> 34

3->52

4->66

5 -> a

6-> b  
Step 7.) Eliminate left recursion.  
Step 8.) Substitute all non terminals with their internal productions if their they are larger than the number of the non terminal to their right  
Step 9.) Recursively substitute the leftmost variable until it starts with a terminal  
1 -> a24

2 -> a24

3->a2

4->b6

5 -> a

6-> b  
  
The string we are considering is aabbbb  
initialization (q0, aabbbb,z) ->(q1,aabbbb,1z)->(q1,abbbb,24z) ->(q1, bbbb,244z) -> (q1,bbbb,44z) -> (q1,bbb,64z)->(q1,bb,4z)->(q1,b,6z)->(q1, LAMDA,z)->(qf, LAMDA, z)