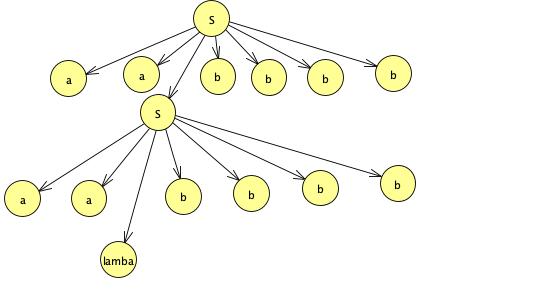
CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_\_\_\_Derek Trom\_\_\_\_

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

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

1. [8] Give a CFG that accepts L.
   1. **S→aaSbbbb|λ**
2. [6] Show the sequence of derivations for the acceptance of *aaaabbbbbbbb* by G in (1).
   1. **S→aaSbbbb→aaaaSbbbbbbbb→aaaaλbbbbbbbb**
3. [6] Draw a derivation tree for *aaaabbbbb*
   1. 

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

1. [10] L1 = { *anbn* | *n* is a multiple of *3* }
   1. **S -> aaaSbbb | λ**
2. [10] L2= { *anbmck* | *k* = *n+m* }
   1. **S → A | B | λ , A → aAc | aBc | ac , B → bBc | bc**
3. [10] L3 = { *anbm* | *n =* *m –*1 }
   1. **S → Ab | b , A → aAb | ab**
4. [10, optional] L4 = { *anbmck* | *n=m* or *m* ≤ *k* }
   1. **S → AC | DB , A → aAb | λ , B → bBc | C | λ , C → cC | λ , D → aD | λ**

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

S → *aa*S*bb* | SS |λ.

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

a. **L = {aa{aa,bb}\*bb | naa(w) = nbb(w)}**

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

1. **S -> aS1B1B2**
2. **S1 -> aS2B1B2**
3. **S2 -> aS2B1B2 | λ**
4. **B1 -> b**
5. **B2 -> b**

Q5. [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] Show that the grammar G is ambiguous.
   1. In order to show ambiguity we must show that there are two of the same left derivations from the grammar
      1. **S -> bbbB -> bbba  
         S -> AB -> AbB -> AbbB -> bbbB -> bbba**
      2. **Thus it is ambiguous because the same string is derived**

1. [6] Give language L that is generated by G, L = L(G), in a formal expression (including a regular expression).
   1. **L = {bna | n>=1} , r = b\*a**
2. [6] Can you construct an unambiguous grammar that is equivalent to G? Otherwise, show that G is inherently ambiguous.
   1. **S → Aa , A → b | Ab**

Q6. [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 | *λ,* C → A

1. [10] Eliminate the λ-productions
   1. **S->bAA|bB|b**
   2. **A->Aa|aac**
   3. **B->bbB**
   4. **C->A**
2. [10] Eliminate the Unit-productions from (1)
   1. **S->bAA|bB|b**
   2. **A->Aa|aac**
   3. **B->bbB**
3. [10] Eliminate the useless productions (2), so that give the simplified equivalent grammar.
   1. **S → bB | b**
   2. **B → bbB | bb**
4. [5] Give the language L that is generated by this grammar, L = L(G), in a formal expression (including a regular expression).
   1. **L = { bn | n > 0 and is odd }, r = b(bb)\***

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

S → AB | *a*B, A → *abb* | *λ* , B → *bb*A

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

1. **Eliminating λ-productions**
   1. **S -> AB | aB | B**
   2. **A -> abb**
   3. **B -> bbA | bb**
2. **Remove unit productions**
   1. **S -> AB | aB | bbA | bb**
   2. **A -> abb**
   3. **B -> bbA | bb**
3. **Replace terminals with variables**
   1. **Y→b**
   2. **Z→a**
4. **Final form**
   1. **S → AB | CA | YY | ZB**
   2. **A → ZC**
   3. **B → CA | YY**
   4. **C → YY**
   5. **Y → b**
   6. **Z → a**

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

S → *a*S*b* | *ab* | *bb*

1. **S → aSB | aB | bB**
2. **A → a**
3. **B → b**