CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: Grant Haataja

**Home Assignment 4: 128/150 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.

*Ans.* S → *aa*S*bbbb* | λ

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

*Ans.* S → *aa*S*bbbb*

→ *aaaa*S*bbbbbbbb*

→ *aaaabbbbbbbb*

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

*Ans.*

Diagram

Description automatically generated

Q2. [18/30] Construct a CFG for the following languages where *n*, *m, k* ≥ 0.

1. [4/10] L1 = { *anbn* | *n* is a multiple of *3* }

*Ans.* S → *aaa*S | B, B → *bbb*B | λ your L(G) = (aaa)\*(bbb)\*

1. [2/ 10] L2= { *anbmck* | *k* = *n+m* }

*Ans.* S → A | B, A → *a*B*c* | *b*B*c*, B → A | *bc* | λ

S ⇒ A ⇒ bBc ⇒ bAc ⇒ b(aBc)c ⇒ ba(bc) cc = babccc ∉ L2

S → *a*Sc | B, B → *b*Bc | λ

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

*Ans.* S → *a*S*b* | *b*

1. [2/10, optional] L4 = { *anbmck* | *n=m* or *m* ≤ *k* }

*Ans.* S→ A | X, A → Ac | B | λ, B → aBb | λ,

S ⇒ A ⇒ anbnc\* it doesn’t guarantee #(b) ≤ #(c)

X → aX | Y | λ. Y → Yc | bYc | λ

S ⇒ X ⇒ a\*bncnc\* it doesn’t guarantee #(a) = #(b)

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*) }

*Ans.* L = { *w* ∈ {*a, b*}\* | *na*(*w*) = *nb*(*w*) and *na*(*w*) mod 2 = 0, i.e. |w|=even,

and #(*a*)*v* > #(b)*v* where *v* is any prefix of *w*.}.

}

Q4. [0/ 10] Find an s-grammar for L = {*anb****2n*** | *n* ≥ 2}.

*Ans.* S → *a*A*bb*, A → *a*B*bb*, B → *a*C*bb* | λ, C → B | λ

Please see the definition of s-grammar. Yours is not s-grammar.

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.

*Ans.* G is ambiguous for the string *bbba* because there are two sequences of productions that yield *bbba*.

S → *bbb*B S → AB

→ *bbb*a → A*b*B

→ A*bb*B

→ *bbb*B

→ *bbb*a

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

*Ans.* L = {*bna* | *n* ≥ 1}

L(bb\*a)

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

*Ans.* S → A*a*, A → *b* | A*b*

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

*Ans.* S → *b*AA | *b*B | *b*, A → *a*A | *aa*C, B → *bb*B | *bb*, C → A

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

*Ans.* S → *b*AA | *b*B | *b*, A → *a*A | *aa*A, B → *bb*B | *bb*

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

*Ans.* S → *b*B | *b*, B → *bb*B | *bb*

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

*Ans.* L = {*b2n+1* | *n* ≥ 0}

L(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.

*Ans*. First, we eliminate the λ-productions, giving: S → AB | B | *a*B, A → *abb*, B → *bb*A | *bb*

Next, we eliminate unit productions, giving: S → AB | *bb*A | *bb* | *a*B, A­­ → *abb*, B → *bb*A | *bb*

Then, we create new productions so that each body has only a single terminal, or multiple variables.

S → AB | BbBbA | BbBb | AaB, A → AaBbBb, B → BbBbA | BbBb, A­a → *a*, Bb → *b*

Finally, we create new productions so that each body has only a single terminal or **two**variables.

S → AB | CA | B­bBb | AaB, A → AaC, B → CA | B­bBb,

Aa → *a*, Bb → *b*, C → BbBb

This grammar is now in Chomsky Normal Form.

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

S → *a*S*b* | *ab* | *bb*

*Ans*. There are no λ-productions or unit-productions, so we start by creating new productions for each of the terminals, such that the right side of each production consists of exactly one terminal followed by zero or more variables.

S → *a*SB | *a*B | *b*B, B → *b*