CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Date: October 3rd, 2018

Due: 5:00 PM, October 10th (Wed.), 2018. (No Extension) Name: **\_Aaron Johnson\_**

**Home Assignment 3: 150 points + 10 points (optional)**

Q1. [30] Show that the following languages are context-free, by giving the context-free grammar that generates it. *n, m, k* ≥ 0

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

**A → aaaAbbb** **| λ**

1. [10] L2= { *anbmck* | *n=m* or *m* ≤ *k* }

**A → BC | DE**

**B → aBb | λ**

**C → cC | λ**

**D → aD | λ**

**E → bEc | λ**

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

**A → B | C**

**B → D | E**

**C → F | G**

**D → aDb | λ**

**E → aEc | D | λ**

**F → aFb | λ**

**G → bGc | λ**

1. [10, optional] L4 = L2 ∪ L3 from (2) & (3)

**A → B | C**

**B → D | E**

**C → F | G**

**D → aDb | λ**

**E → aEc | D | λ**

**F → aFb | λ**

**G → bGc | λ**

**A → BC | DE**

**B → aBb | λ**

**C → cC | λ**

**D → aD | λ**

**E → bEc | λ**

Q2. [10] Give the language L that is generated by the given grammar.

S → *aa*S*bb* | SS |λ.

**L(G) contains strings where # of a = b and is a multiple of *2* and where at least 2 a’s appear before at least 2 b’s**

Q3. [10] Find an s-grammar for L(*aaa*\**b* + *ab*\*).

**A → aBCD | BE**

**B → a**

**C → aC | λ**

**D → b**

**E → bE | λ**

Q4. [30] For a given grammar below,

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

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

1. [10] Show the grammar G is ambiguous.

**If we have a string s = bbba, we can derive s multiple ways given G:**

**S ⇒ AB ⇒ AbB ⇒ AbbB ⇒ bbbB ⇒ bbba**

**AND**

**S ⇒ bbbB ⇒ bbba**

**Therefore, G is ambiguous**

1. [10] Give language L that is generated by G, L = L(G), in a formal expression (including a regular expression).

**L = bb\*a**

**L = {bna | where n >= 1}**

1. [10] Construct an unambiguous grammar that is equivalent to G.

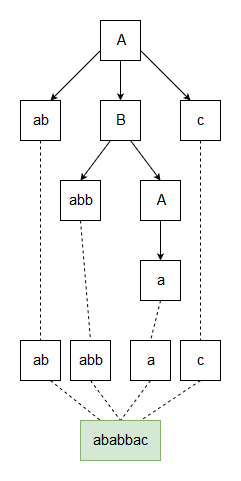
**S → AB**

**A → Ab | b**

**B → a**

Q5. [10] In the given grammar, draw a derivation tree for the string *ababbac*.

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

 A → *a* | *aa*A| *~~ab~~*~~A~~*~~c~~* ***abBc*** , B → *abb*A | *b.*

Q6. [35] In the given grammar below, 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 → bB | *b*AA , A → *a*A| *aaC* , B → *bb*B | *λ,* C → A

1. [10] Eliminate the λ-productions

**S → bAA | bB | b**

**A → aA | aaC**

**B → bbB | bb**

**C → A**

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

**C produces A, so we can substitute whatever A produces in for C**

**S → bAA | bB | b**

**A → aA | aaC**

**B → bbB | bb**

**C → aA | aaC**

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

**Endless a’s would be produced by (2), so rules A and C are useless**

**S → bB | b**

**B → bbB | bb**

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

**L = b(bb)\***

**L = {b(bb)n, where n >= 0}**

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.

**S → AB | CB | DA | EE**

**A → CF**

**B → FA | EE**

**C → a**

**D → EE**

**E → b**

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

S → *a*S*b* | *ab* | *bb*

**S1 → aS3 | aS4 | bS4**

**S2 → a**

**S3 → aS3S4 | aS4S4 | bS4**

**S4 → b**