CS5338

– Formal Languages Spring 2019 –

Assignment 3

Due: March 29, 2019

Submitted By

Ahmed Raza,

a\_r904,

A04775175

**1. (20 pts) Give a context-free grammar (CFG) for each of the following languages over the alphabet Σ = {a, b}:**

**(a)** All strings in the language L: {anbma2n|n, m ≥ 0}

S → aSaa|B

B → bB|Є

**(b)** All nonempty strings that start and end with the same symbol.

S → aXa|bXb|a|b

X → aX|bX|Є

**(c)** All strings with more as than bs.

S → Aa|Ms|SMA

A → Aa|Є

M → Є|MM|bMa|aMb

**(d)** All palindromes (a palindrome is a string that reads the same forwards and backwards).

S → aSa|bSb|a|b|Є

**2. (20 pts) Let G = (V, Σ, R, S) be a context-free grammar such that V = {E, T, F}, Σ = {a, +, ∗, (, )}, S = E and R is**

**E → E + T | T**

**T → T × F | F**

**F → (E) | a**

**Give parse trees and leftmost derivations for the following strings.**

**I.** a

S ⇒ E ⇒ T ⇒ F ⇒ a

**II.** a\*(a+a)

S ⇒ E ⇒ T ⇒ T\*F ⇒ F\*F ⇒ a\*F ⇒ a\*(Є) ⇒ a\*(Є+T) ⇒ a\*(T+T) ⇒ a\*(F+T) ⇒ a\*(a+T) ⇒ a\*(a+F) ⇒ a\*(a+a)

**III.** a+a+a

S ⇒ E+T ⇒ E+T+T ⇒ T+T+T ⇒ T+T+T ⇒ F+T+T ⇒ a+T+T ⇒ a+F+T ⇒ a+a+T ⇒ a+a+F ⇒ a+a+T ⇒ a+a+a

**IV.** ((a+(a)))

S ⇒ T ⇒ F (E) ⇒ (T) ⇒ (F) ⇒ ((E)) ⇒ ((E+T)) ⇒ ((T+T)) ⇒ ((F+T)) ⇒ ((a+T)) ⇒ ((a+F)) ⇒ ((a+E)) ⇒ ((a+(T))) ⇒ ((a+(F))) ⇒ ((a+(a)))

**3. (30 pts) Answer each part for the following context-free grammar.**

**R → XRX | S**

**S → aTb | bTa**

**T → XTX | X | E**

**X → a | b**

**I.** What are the variables and terminals of G? Which is the start symbol?

Set of Variables : {R,S,X,T}

Set of terminals : {a,b}

Start symbol : R

**II.** Give three examples of strings in L(G).

ab, ba, aab

**III.** Give three examples of strings not in L(G).

a, b, aa

**IV.** True or False: T ⇒ aba. False

**V.** True or False: T ⇒∗ aba. True

**VI.** True or False: T ⇒ T. False

**VII.** True or False: T ⇒∗ T. True

**VIII.** True or False: XXX ⇒∗ aba. True

**IX.** True or False: X ⇒∗ aba. False

**X.** True or False: T ⇒∗ XX. True

**4. (10 pts) Determine whether the grammar implicitly defined by the following rules is ambiguous. Prove your answer.**

**S → AB**

**A → aA**

**A → abA**

**A → ε**

**B → bB**

**B → abB**

**B → ε**

It is ambiguous, the left most derivatives of ab are,

S ⇒ AB ⇒ abAB ⇒ abB ⇒ abB ⇒ ab

S ⇒ AB ⇒ B ⇒ abB ⇒ abB ⇒ ab

The language generates regular grammar and is not inherently ambiguous

**5. (20 pts) Give pushdown automata that recognize the following languages.**

**(a)** A = {w ∈ {0, 1}\* | w contains at least three 1s }

1, Є→Є

1, Є→Є

1, Є→Є

0, Є→Є

0, Є→Є

0, Є→Є

0, Є→Є

A is a regular language and Transition between DFA and PFA can be done using the same states and transactions.

**(b)** B = {w ∈ {0, 1}\* | w = wR and the length of w is odd }

Є, Є→$

0, Є→Є

1, Є→Є

Є, $→Є

0, Є→0

1, Є→1

0, 0→Є

1, 1→Є

Since the length of w∈B is odd, so it must have a symbol exactly in the middle position. i.e. |w|=2n+1 for some n>=0, and the (n+1)th symbol in w is the middle one.

If a string w of length 2n+1 satisfies w=w^R, the first n symbol must match (in reverse order) the last n symbol. While the middle symbol does not have to match anything. Thus, in the above PDA the transition from 12 to itself reads the first n symbol pushes it on the stack.

The transition form q2 to q3 non deterministically identifies the middles symbol of w, which doesn’t need to match any other symbol, popping the stack at each step to make sure the symbol after the middle matches (in reverse order) the symbol before the middle one.

**(c)** C = {w ∈ {0, 1}\* | w = wR }

Є, Є→$

Є, Є→Є

0, Є→Є

1, Є→Є

Є, $→Є

0, Є→0

1, Є→1

0, 0→Є

1, 1→Є

The length of a string w∈c can either be even or odd. If it is even, then there is no middle symbol in w, so the first part of the w is pushed back on the stack. We move from q2 to q3 without reading, pushing or popping anything and then match the second half of w to the first half in reverse order. If the length of w is odd, then it has a middle symbol, and the description of the PDA in part (b) applies.

**(d)** D = { ai bj ck | i, j, k ≥ 0 and i+j = k }

Є, Є→$

Є, Є→Є

Є, Є→Є

a, Є→x

b, Є→x

c, x→Є

Є, $→Є

For every a and b read in the first part of the string, the PDA pushes an x onto the stack. Then it must read a c for each x popped off the stack.