CSci 435: Formal Languages and Automata

**Exam 2: 100 points + 20 points (optional)**

Name: Adam Roy

1. Your answer should be precise and fully described; any sloppy answer will not get a full point.
2. In any PDA, assume that a start symbol ‘z’ is always located at the stack; thus, you do not have to give a transition of adding ‘z’ to the stack at the beginning or/and removing it from the stack at the end.
3. For the design of a PDA or a TM, draw its transition diagram.

Mark the following;

Difficulty:

Very Easy: \_\_\_\_\_\_ Easy: \_\_\_\_\_\_ Moderate: x Difficult: \_\_\_\_\_\_ Very Difficulty: \_\_\_\_\_\_

Time:

Short: \_\_\_\_\_\_\_\_\_ Enough: x Too Much: \_\_\_\_\_\_\_\_\_\_

Abbreviation:

CFG/ (D)CFL: Context-Free Grammar/ (Deterministic) Context-Free Language

(N/D)PDA: (Nondeterministic / Deterministic) PushDown Automaton

TM: Turing Machine

Q1. [15] In the given CFG, G = ( {S}, {*a, b*}, S, P ) with productions S →1 AB | *aaaB*, A →2A*a* | *a*, B →3 *b,*

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

L(aa\*b) = (anb | n ≥ 1)

L(G) = aa\*b

1. [5] Draw a left-most derivation tree for the string *aaaab*.A network of dots and lines

   Description automatically generated
2. [5] Decide if the G is ambiguous or not. Justify your answer.

The language with grammar G is ambiguous as strings such as aaab can be produced with productions  
S → aaaB  
aaaB → aaab

Or

S → AB  
AB → AaB  
AaB → AaaB  
AaaB → aaaB  
aaaB → aaab

Q2. [20] In the given grammar below,

S → *a*A | *a*BB , A → *aa*A | *λ,* B → *b*B| *bbC* , C → B

1. [15] Give the ***simplest equivalent grammar*** by eliminating (A) the λ-productions., (B) the unit-productions, and (C) the useless variables/productions, step by step.
2. Remove λ productions:

S → aA | aBB , A → aaA | λ, B → bB | bbC , C → B  
**S → aA | aBB | a , A → aaA | aa, B → bB | bbC , C → B**

1. Remove unit productions:

S → aA | aBB | a , A → aaA | aa, B → bB | bbC , C → B  
**S → aA | aBB | a , A → aaA | aa, B → bB | bbC , C → bbC**

1. Remove useless productions:

S → aA | aBB | a , A → aaA | aa, B → bB | bbC , C → bbC  
**S → aA | a , A → aaA | aa**

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

The language generated by the grammar:

L(G) = (an | n ≥ 1, n is odd)

The regular expression of the grammar:

L(G) = a(aa\*)

Q3. [10] Construct a NPDA that accepts the language L = {*anbn+1* | *n* ≥ 0}. Draw its transition diagram.

A diagram of a diagram of a line with circles and letters

Description automatically generated

Q4. [20, optional] For the language L = {*anbm* | *n* > *m* ≥ 0}.

1. [10] Construct a CFG G that defines L, L(G) = L.

S → aSb | A, A → aA | λ

1. [10] Show that L is a Deterministic CFL by constructing a Deterministic PDA with an empty stack.

A diagram of a triangle with lines and circles

Description automatically generated

Q5. [15] By Pumping Lemma, prove that the language L = { | *n* ≥ 0 } is not context free.

In your proof, (A) clearly decide 5 substrings ***u, v, x, y, z*** and ***m*** > 0 for *w* ∈ L, such that *w = uvxyz* and

0 < |v*xy*| ≤ *m* ≤ |*w*| and (B-C) give the number pumping, ***i*,** that violates *wi* = *uvixyiz* ∉L.

Hint: Choose w = in L, then apply Pumping Lemma step by step.

We assume that the language is context free and we also assume that there is a PDA for the language.

We take the string aaaaaaaaa or a9 and devide the string into 5 parts, u = aa, v = aa, x = aa, y = a, z = aa

By pumping lemma any value of I such that uvixyiz belongs to L if it is true then L is a context free language.

Assuming that I = 2 the string “aa aaaa aa aa aa” which equals a12 is formed which is not part of the language L

Assuming that I = 3 the string “aa aaaaaa aa aaa aa” which equals a15 is formed which is also not part of the language L therefore proof by contradiction states that the language L is not a context free language.

Q6. [10 ] Decide if the language L = {*akbn* | *k* ≤ *n* ≤ 2*k* – 1} is linear or not. If L is linear, give its linear grammar. Otherwise, prove it by the Pumping Lemma for linear language.

Assuming that L is regular, there is a pumping length p > 0 such that for any string |w| >= p where we can divide w into 3 parts, x y z, such that |xy| <= p, |y| > 0, and for all strings xyiz is in the language L.

We take the string aaabbbb and assuming I = 2 we get aaaaabbbbb which is accepted

We again take the string aaabbbb and assuming I = 3 we get aaaaaaabbbbbb which is rejected therefore with proof by contradiction we can say that the language is non linear.

Q7. [10] Design a Turing machine with no more than three states that accepts the language L(*a*(*a*+b)\*)

where Σ = {*a, b*}. Draw its transition diagram.

A diagram of a diagram

Description automatically generated

Q8. [10] Using adders, subtracters, comparers, copiers, or multipliers, construct a TM that compute the functions: *f*(*n*) = *n*!*~~.~~* Draw its block diagram.

A diagram of a block diagram

Description automatically generated

Q9. [10] Design a TM that computes the function: *f*(*x*) = 1 if *x* is odd | 0 if *x* is even, where *x* is given in unary notation with 1’s only to TM.

Use the tape symbol Γ = {1, 0, €}. Draw its transition diagram of the TM.

A diagram of a triangle with circles and lines

Description automatically generated