CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Date: November 30th, 2023

**Due: by the end of the day, 12/12 (Tue.)**

**Final Exam: 150 points + 35 points (optional)**

Name: Adam Roy

1. Write your answer below the corresponding question.
2. Do not include the photo image of your handwriting for a text.

Abbreviation:

REG: Regular Language, REC: Recursive Language, RE: Recursive Enumerable Language,

(D)CFL: (Deterministic) Context-Free Language, FA: Finite Automata, TM: Turing Machine,

UG: Unrestricted Grammar, CSG: Context-Sensitive Grammar, etc.

Mark the followings.

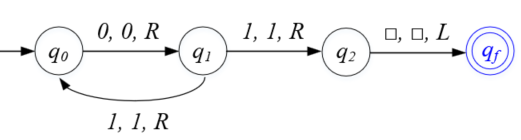
Difficulty:

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

Time:

6 Hours and 42 minutes

Q1. [25] For a Turing Machine M below,



1. [7] Give a language L(M) that is recognized by M in the *formal definition* (including a regular expression).

The Turing machine can be defined as a 7-tuple (Q, Σ, Γ, δ, q0, q𝑓) where: Q is a finite set of states, Σ is the alphabet, Γ is the tape alphabet, δ is the transition function such that Q x Γ → Q x Γ x {L, R}, q0 is the initial state, q𝑓 is the final and accept state.

The language that the Turing machine would accept would be L(M) = (O1)n | n ≥ 1

The regular expression of the language would be L(M) = (01)+

1. [10] By reviewing the transitions of TM, give a grammar for L(M).

Hint: L(M) is a regular language.

Context free grammar of the language.

S → 01S | 01

1. [8] Give a derivation for 010101 using the grammar in (2).

Q0, 0 → Q1, 0, R  
Q1, 1 → Q0, 1, R  
Q0, 0 → Q1, 0, R  
Q1, 1 → Q0, 1, R  
Q0, 0 → Q1, 0, R  
Q1, 1 → Q2, 1, R  
Q2, B → Q3, B, L

Q2. [15] Give a context-sensitive grammar G for the following languages.

1. [10] L = {*anbna2n* | *n* ≥ 1}

Hint: Refer to the CSG of { *anbncn* | *n* ≥ 1} and modify it.

S → aBb  
Ba → aB  
Bb → bbA  
bA → Ab  
aA → Aa  
Ab → Ba  
a → λ  
b → λ

[5] From the CSG G of (1), give a derivation of *w* = *aabbaaaa,* i.e.S ⇒G\* *aabbaaaa.*

S → aBb  
aBb → aaB  
aaB → aabbA  
aabbA → aabbaAb  
aabbaAb → aabbaAba  
aabbaAba → aabbaaaa

Q3. [10] Give the language generated by the following unrestricted grammar in a formal expression.

e.g.) { *anbn* | *n* ≥ 1 }

S → AB, A → *a*Ab, bB → bbbB, *a*Ab → *aa*, B→λ

L = {an+2bn+2m | n, m ≥ 0)

Q4. [10] Prove that L= { *apbp aqbq* | *p, q* ≥ 1 } is Context-Free Language but not Linear.

Hint: Construct its CFG to show L is CFL. Then, prove the non-linearity of L using Pumping Lemma for linear language.

Because we already know that (anbn | n ≥ 0) is context free and L is just two instances of this grammar the grammar of L is defined as  
S → S1S2  
S1 → aS1b, ab   
S2 → aS2b, ab

We then move to prove the language is not linear. Supposing L is linear there is a p > 0 that for every w in L with |w| > p we can we can write w as uvxyz such that |vy| > 0, |uvyz| <= p and uvixyiz for every i >=0 is in L. letting w = apbpapbp we find |w| > p. Thus w = uvxyz where |uvyz| < p we assign v = ac and y = ad. with that  
w = ap+c+dbpapbp when we pump with a value of 2 we find that w is no longer within the rules of L and thus we find there is a contradiction and L is proven to be non linear.

Q5. [10] Show that the family of context-sensitive language is closed under reversal.

Hint: For ∀L ∈CSL, there exists an LBA M, s.t. L = L(M).

You must formally define an LBA for LR using the LBA for L.

We define an LBA as M = (Q, Σ, Γ, δ, Q0, Qa, Qr). For a M’ upon an input w where w is reversed to wR, with additional states the machine will then write a new wR on the tape and traverse it to reverse the output. After reversing the input the machine will then simulate LBA M which is designed for the language L and it should therefore accept wR only if w is already in L. if M’ accepts wR then the original machine will accept w and same with rejection. We therefore can define M’ as M’= (Q’, Σ’, Γ’, δ’, Q0’, Qa’, Qr’) and because M’ is an LBA and is able to recognize LR it can be stated that the family of context sensitive languages is closed under reversal.

Q6. [10] Prove that the complement of a Context-Free Language must be Recursive.

Since all context-free languages are already recursive and closed it should hold that a complement CFL would also be recursive. So if a language L is context free then it must be recursive and L’ would also be recursive.

Q7. [10] Design a TM that computes the function: *f*(*x*) = 2*x* - 1, where *x* is given in unary notation with 1’s only to TM. Use the tape symbol Γ = {1, *a*, €}. Draw its transition function, not a block diagram.

A screenshot of a computer

Description automatically generated

Q8. [10] Let M1 and M2 be arbitrary Turing machines. Show that the problem “L(M1) ⊆ L(M2)" is undecidable.

If we construct a machine M1, M1 will ignore input x lets say and simulates M on input w and will accept only if M accepts. Essentially L(M1) = Σ\* if M enters an accept state otherwise it rejects similarly if we create another machine M2 which simulates M we find L(M2) = Σ\*. From this point we find that if M accepts then L(M1) ⊆ L(M2) otherwise if M does not accept L(M2) is not a subset of L(M1).

Q9. [10] Show that for A = {*wi*| 00, 10, 01, 11} and B = {*vi* | 001, 0, 101, 1}, there exists a Post Correspondence solution.

No solutions can be attained from the set using individual elements, but solutions can be attained from the set by concatenating elements. In this case

u = A1A2 = 0010 and u = A2A4 = 1011

v = B1B2 = 0010 and v = B3B4 = 1011

are some solutions that could work.

Q10. [10] Determine whether the given Boolean expression is satisfiable or not.

The expression would be considered satisfiable because at least one combination of assigned truth values yields the entire expression to evaluate as true. One such combination would be X1 = T X2 = T X3 = F

Q11. [10] Show that L = { *vvRv* | *v* ∈{*a, b*}+} is in DTIME(*n*). Explain how a Deterministic-multitape-TM runs on *w* ∈ L in O(*n*).

When this multi tape turing machine is initialized it first computes the length of the input and if x % 3 != 0 then it rejects. Then all heads of the tapes are initialized at their positions, 0, x/3 and 2x/3. The machine then enters a verification loop to verify if all heads are on the same step from 1 to x/3 moving each head right after each verification step. If all verifications are successful, then accept. The machine runs on O(n) time where n is the size of the input. Overall the machine makes two passes total once to check the length and reject if not divisible by 3 and a second time to verify and accept or reject if the form is not of VVV

Q12. [20] *k*-clique problem

1. [7] Show that a *k*-clique must have exactly *k*(*k*-1)/2 edges.

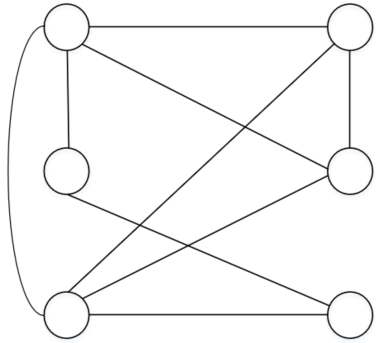
A single vertex with no edges has 1(1-1)/2 = 0

2 vertex with one edge has 2(2-1)/2 = 1

3 vertex with 3 edges has 3(3 -1)/2 = 3

4 vertex with 6 edges has 4(4 – 1)/2 = 6

1. In the given graph below,
   1. [7] Find a 4-clique. Mark those edges in the graph.
   2. [6] Prove that the given graph has no 5-clique.





Using the clique formula it can be noted that a 5-clique needs 10 edges but any 5 vertices on this graph only contain at max 7 edges so this graph does not support having a 5-clique

Q13. [15, optional] Evaluate the efficiency of algorithm for L = {*ww* | w ∈ {*a, b*}\*} as the number of moves in terms of *n*=|w|, e.g.) O(*n*).

Describe how each TM process L with your O(T(*n*)) moves where T(*n*) is a function of *n*.

1. [5] On a standard single-tape deterministic TM.
2. [5] On a two-tape deterministic TM.
3. [5] On a single-tape nondeterministic TM

Q14. [10, optional] Explain a Halting Problem, its importance/contribution, and more in detail in your own description. – Do not copy it from the textbook/slide.

The halting problem was introduced by Alan Touring. It is the important due to its undecidability meaning it can test the limits of algorithms and proves that there are algorithms out there that are impossible to build a solution for. There the halting problem is also important due to it bringing about a new branch of theoretical computer science. Of course, there is also the development of Turing machines and general universal computation. The halting problem has direct ties to the development of Turing machines.

Q15. [10, optional] Explain P-Problem, NP-Problem and NP-Complete Problem, respectively (P/NP/NP-Complete-language, equivalently), and their importance/contribution in the theory of computer science, in detail.