CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_Grant Haataja

**Home Assignment 7: 120 points + 25 points (optional)**

Q1. [20] For a given language below, construct a TM with a *single final state* that accepts it.

1. [6] L = {w ||*w*|is a multiple of 4} where Σ = {*a*, *b*}.

*Ans*.

Diagram

Description automatically generated

1. [7] L = {w | *na*(*w*) ≠ *nb*(*w*)} where Σ = {*a*, *b*}.

*Ans*.

Diagram

Description automatically generated

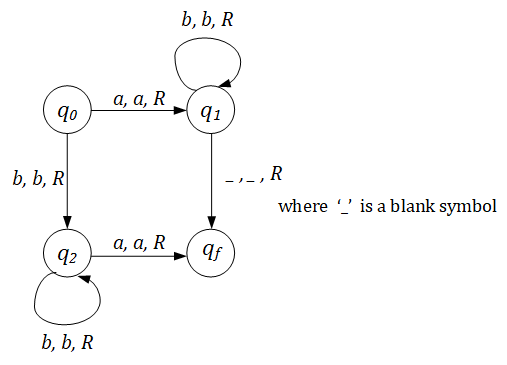
1. [7] L = {w | *anbn anbn* | *n* ≠ 0} where Σ = {*a, b*}.

*Ans*.

Diagram

Description automatically generated

Q2. [10] What language is accepted by the Turning machine whose transition graph is in the figure?



*Ans*. L(*ab*\* + *bb*\**a*).

Q3. [10] Construct a TM that accepts L = {ww | w ∈ {*a, b*}+ }.

Hint: This is a standard deterministic TM.

So, TM has to **find** the middle of the string first; then, compare two halves.

*Ans*.

First, we find the middle of the string and mark it. If the length of the string is not even, it cannot be in L, so we reject it. Then, we alternate between the beginning of the first occurrence of *w* and the second occurrence of *w*, marking characters that match and rejecting if a non-match is found. Upon completing this process, we accept the string successfully.

A picture containing diagram

Description automatically generated

Q4. [20] Construct a TM that computes the following function

1. [10] .

The input *w* is in the unary representation.

*Ans*. A picture containing bubble chart

Description automatically generated

1. [10]  *f*(*x, y*) = *x* + 2*y.*

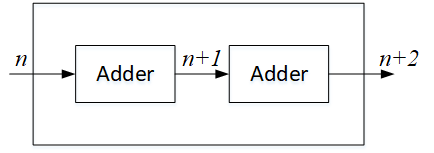
*Ans*.

A picture containing chart

Description automatically generated

Q5. [20] Using adders, subtracters, comparers, copiers or multipliers, draw block diagram for TM that compute the functions:

e.g.) *f*(*n*) = *n* + 2



1. [10]*f*(*n*) = *2n.*

*Ans*.

Diagram

Description automatically generated

1. [10] *f(n) = n!*

*Ans*.

A picture containing diagram

Description automatically generated

Q6. [15] For a two-tape Turing Machine,

1. [5] Give a *formal definition* of a transition function δ in two-tape TM.

*Ans*. δ(*q*­0, (*a*, *e*)) = (*q*1, (*x*, *y*), (L, R))

1. [10] Construct a two-tape TM that accepts L = { *anbn cn* | *n* ≥ 1}

*Ans*.

Diagram

Description automatically generated

Q7.[15, optional] Construct a **Nondeterministic** TM (NTM) that accepts L ={ *wwRw* | *w* ∈ {*a, b*}+ }.

1. Draw its transition graph, (B) explain how your transitions work out and (C) how the nondeterministic simplifies the case.

Note that the middle of the string in wwR can be guessed in NTM.

Q8. [10] Give the encoding, using the suggested method in the slide of Chap.9-#25-#27, for

δ(*q1, a1*) = (*q1, a1*, R); δ(*q1, a2*) = (*q3, a1*, L); δ(*q3, a1*) = (*q2, a2*, L)

*Ans*. Here we assume R is encoded as 11 and L is encoded as 1. We write 00 to give space between transitions.

1010101011 00 101101110101 00 1110101101101

Q9. [5] If *a* is encoded as 1, *b* as 11, R as 1, L as 11, decode the string 011010111011010.

*Ans*. δ(*q2, a*) = (*q3, b*, R)

Q10. [10, optional] Describe an algorithm that examines a string in {0, 1}+ to determine whether or not it represents an encoded Turing Machine.

*Ans*. Ignore all leading zeros. Identify groups of 1’s, and make sure that there are five groups of 1’s in a row, all separated by exactly one zero. The first four groups of 1’s can have one to many 1’s, but the fifth group must have either one or two 1’s. Then, after the fifth group of 1’s there will be exactly two 0’s, before the cycle repeats.

Q11. [10] Describe how Linear Bounded Automata could be constructed to accept

L = { *an* | *n* is a prime number}.

*Ans*. Divide the number of *a*’s by successively increasing prime numbers until either the remainder is 0, or the prime number dividing by is greater than the number of *a*’s in the string, accepting or rejecting the string appropriately.

1st track: stores the number of *a*’s remaining after divisions.

2nd track: stores the current number being divided by.