|  |
| --- |
| Assignment 2 |
|  |
| Eman Samy  188849 |

# Question 1

|  |
| --- |
| Write a computer program in Java/Python for the following function to prove that each function is computable by designing a detailed TM. In each case give an informal high-level description of your implementation:  **𝑓(𝑥) = 7𝑥 + 1 𝑓𝑜𝑟 𝑒𝑣𝑒𝑟𝑦 𝑥 ≥ 0**  **A function has:**  **Diagram, venn diagram  Description automatically generated**  **For 7x + 1**  7x is 1111111 if x = 1 and 11111111111111 if x = 2 in unary  1 is 1 in unary |

A function is computable if there is a TM such that

q0 w > qf f(w) for all w belongs to Domain.

Input string 7x0y

Output string 7xy0 7 if x = 1 delimiter 1

1

1

Start

1

0

1

1

1

1

1

Q0 (initial state)

7x + 1

Finish

-------------

1

0

1

1

1

Qf (final state)

Time 0 1 -> 1,R

1

1

1

0

1

1

1

1

1

Q0

Time 1 1 -> 1,R

1

1

1

0

1

1

1

1

1

Q0

And moves to the right one steps in times 2,3,4,5,6 sequentially 1 -> 1,R

then

Time 7 0 -> 1,R

1

1

1

0

1

1

1

1

1

Q0

Time 8 1 -> 1,R

1

1

1

1

1

1

1

1

1

Q1

Time 9 |\_| -> |\_|,R

1

1

1

1

1

1

1

1

1

Q1

Time 10 1 -> 0,L

1

1

1

1

1

1

1

1

1

Q2

Time 11 1 -> 1,L

1

1

0

1

1

1

1

1

1

Q3

Time 12 1 -> 1,L

1

1

0

1

1

1

1

1

1

Q3

Time 13 1 -> 1,L

1

1

0

1

1

1

1

1

1

Q3

Q3 moves to the left in times 14,15,16,17 1 -> 1,L

Time 18 1 -> 1,L

1

1

0

1

1

1

1

1

1

Q3

Time 19 |\_| -> |\_|,R

1

1

0

1

1

1

1

1

1

Q3

Time 20 HALT and accept

1

1

0

1

1

1

1

1

1

Q4

Text

Description automatically generated

Initializng an array which represents the tape

Count the 1s from the right to the left

till the final state

The machine halts

Looping on the tape starting from the end (the far right,

turn the last cell’s 1 into 0

Looping on the tape from the start, if the cell has 1 it moves to the next

Else, it turns the 0 into 1 and move to the next

Inserting the 1 in the function

Inserting the delimiter

Inserting 1s in the array

# Question 2

Write a computer program in Java/Python to construct a PDA to recognize the following language: 𝐿 = { |𝑛 ≥ 1} 𝖴 {𝑎^𝑛 𝑑 𝑏^2𝑛 𝑛 ≥ 1}

For :

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Time 0

Stack is empty

Current state: q0

Time 1

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Input

a a a b b b c

$

Stack

Current state: q1

Time 2

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Input

a

a a a b b b c

Stack

$

Current state: q1

Time 3

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

a

Input

a

a a a b b b c

Stack

$

Current state: q1

Time 4

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

a

a

Input

a

a a a b b b c

Stack

$

Current state: q1

Time 5

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

a

a

Input

a

a a a b b b c

Stack

$

Current state: q2

Time 6

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

a

Input

a

a a a b b b c

Stack

$

Current state: q2

Time 7

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Input

a

a a a b b b c

Stack

$

Current state: q2

Time 8

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Input

a a a b b b c

Stack

$

Current state: q3

Time 9

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

Input

a a a b b b c

Stack

Current state: q4

For :

Time 0

a a a b b b b b b d

Input

, -> b,a->

, -> $ , -> b,a->

Q4

Q3

Q2

Q1

Q0

Current state: q0

Stack is empty

, ->

Q5

Time 1

a a a b b b b b b d

Input

, -> b,a->

, -> $ , -> b,a->

Q4

Q3

Q2

Q1

Q0

Current state: q0

Stack is empty

$

, ->

Q5

Time 2

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

Current state: q1

Stack

a

a

, ->

Q5

Time 3

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

Current state: q2

a

Stack

a

a

a

, ->

Q5

Time 4

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

a

a

Current state: q2

a

Stack

a

a

a

, ->

Q5

Time 5

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

a

a

Current state: q3

a

Stack

a

a

a

, ->

Q5

Time 6

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

a

a

Current state: q3

a

Stack

a

a

a

, ->

Q5

Time 7

a a a b b b b b b d

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

a

a

Current state:

a

Stack

a

a

a

, ->

Q5

The same happens in time, 7 ,8,9 one a is removed from the stack each time b is the input

Time 8

a a a b b b b b b d n

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

Current state: q4

Stack

, ->

Q5

Time 9

a a a b b b b b b d n

Input

, -> b, a->

, -> $ , -> b, a->

Q4

Q3

Q2

Q1

Q0

Current state: q6 n

Stack

Q5

, ->

Q6

The final state is reached and the stack is empty

The PDA looks like this

, -> a , ->

, -> $ , -> , ->

Q4

Q3

Q2

Q1

Q0

, -> $

, -> b, a->

, -> b, a->

R4

R3

R2

R1

, ->

R5

Text

Description automatically generated

b must be equal to 2n and a to N and d appears once

which means b appears twice times a

If the input is d, nothing is popped or pushed

if the input is c, nothing is pushed or popped

If the input is b, a is popped and nothing is pushed

If the input is a, nothing is popped and a is pushed twice OR

a is popped and a is pushed three times

The first time $ is pushed to the stack

The first time $ is pushed to the stack

N of a must be equal to N of b and C appears once

Last the stack is popped in case of $

If the input is b, a is popped and nothing is pushed

If the input is a, nothing is popped and a is pushed

The first time $ is pushed to the stack

Creating a stack

Text

Description automatically generated



Text

Description automatically generated



Text

Description automatically generated