317 Homework #10

1. Build a turing machine accepting $(b+c)^{+} \# a^{+}$

* we know that (b+c) and c+ are (b+c)+ means any number of b or c repeats at least one of them *

* a^{+} means any number of a's at least 1 given $(b+c)+ \# a+$ that can be build with intermediate symbol # and we combine the two parts into one *

Turing machine $M = < Q\ T, B, E, \delta, q_0, F >$

Q - Set of States

1 - Tape symbols

B - Blank symbol $\subset T$

E - input alphabet $E \in T$

δ . Transition function

q_0 - starting state

$F = < Q$ final accepting states

$1 = (b+c) \# a+$

$(q_1, b) - (q_2, b, R)$

"Reading at least one to move. state"

$(q_3, b) \to (q_3, b, R)$ continue to read

$(q_3, b) \to (q_4, b, R)$ end do tape to halt process

$E . \{a, b+c\} = \{a, b, c, B \# T\}$

↓

Transition function as follows

$\delta (Q \times 1) - Q \times r \times \{L R S\}$

$(q_0, b) \to (q_1, b, R)$ Reading at least one basic to move to next state

$(q_0, c) \to (q_1, c, R)$ all

$(q_1, b) \to (q_1, b, R)$ / Reading more than one 'c'

$(q_1, b) \to (q_2, b, R)$

$(q_1, \#) - (q_2, \#, R)$ // End do first Block (bcc)

2. Build a turing Machine accepting $\{x\#x^r : x^r \{x \in \{a,b\}^+\}\}$

//state q_0: Read input
//action : Read String

q_0: Read input
if input = "" then go to q Accept
else go to q_1
// state q_1: check first character
// Action: Read the first character

q_1:
Read first character
if first character = "#" then go to qReject
else go to q_2
// state q_2: Store the first character
//action: Store the first character

q_2:
Store the first character Go to q_3
Go to q_3
// state q_3: Move the right
// Action: Move one cell to the right

q_3:
Move one cell to the right
Go to q_4
// state q_4: check for "#"
// Action: Read the character

q_4: Read the Character
if Character = "#" then go to qReject
else go to q_5

// state q_5: check if character is the Same as the first character
// Action: compare the character with the Stored first character

q_5: compare the character with the Stored first character
if character Match then go to q_6 else go to qReject
// State q_6: Move back to left
// Action: move one cell to left

q_6: Move one cell to the left
Go to q_7
// state q_7: Read the character
// Action: Read the character

q_7: Read the character
if character = "" then go to q Accept
else go to qReject
// State qAccept: Accept
//Action: Accept the input

qAccept:
Accept the input
// State qReject: Reject
//Action: Reject the input

qReject:
Reject the input

3.

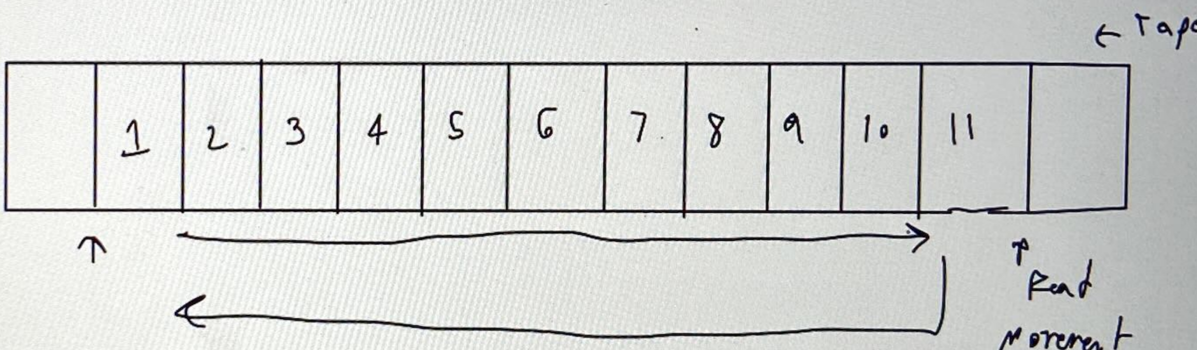
→ Con free language are recognized by PDA

→ PDA use stack, for recognising given string

Now,

Turing machine is a one-turn turing Machine. Such that during any execution and on any input. M makes at most one turn on the tape.

Means, if the tape is moving right, later it (May move left but never moves right again.



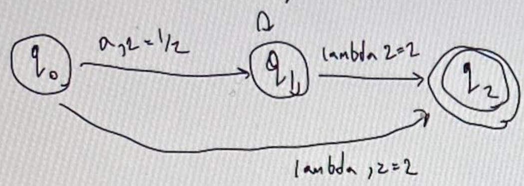
→ This is Similar to the stack, because after turning the last element is accessing first.

→ Hence, the tape has become, stack, so this machine acts like PDA.

* The language accepted by one-turn turing machine are context free*

4.

Let $T = \{1, 2\}, \Sigma = \{a, b\}$



We simulate a 2-PDA on a turing machine by using the tape as a queue of possible configuration of the 2-PDA (say by simply keeping them in sequence with delimiters, popping by reading the first configuration and then moving the other left, and pushing by adding configrations at the end). Each configuration in the queue encodes a state of the 2-PDA, the contents of both stacks, and the remaining input. our TM will then pop a configration, iterate through the possible 2-PDA transitions from this configuration and for each transition pushing the resulting configration onto the stack. If the transition popped has no more input and is in an accept state, the TM accepts if the queue is ever empty, the TM rejects.

The 2-PDA accepts if and only if there is some sequence of valid transitions which brings it to an accept state with no input left. if such a sequence exists the TM will eventually follow it and accept. Conversely, if no such sequence exists the TM will not accept. Thus this TM accepts if and only if the 2-PDA did not.