Student Information

Full Name : Adil Kaan Akan
Id Number : 2171155

Answer 1

a.

The book states 3 rules:
(1)- ((p,a,e),(p,a)) for each a in Σ
(2)- ((p,e,α^R),(p,A)) for each rule $A \to \alpha$ in R.
(3)- ((p,e,S),(q,e)).
If we use these rules, we get the followings:

1. (p,a,e),(p,a)
2. (p,b,e),(p,b)
3. (p,c,e),(p,c)
4. (p,e,XaXSa),(p,S)
5. (p,e,XbXSb),(p,S)
6. (p,e,c),(p,S)
7. (p,e,Xa),(p,X)
8. (p,e,Xb),(p,X)
9. (p,e,e),(p,X)
10. (p,e,S),(q,e)

b.

Step	State	Unread Input	Stack	Transition Used
1	p	abbcbabbaa	e	-
2	p	bbcbabbaa	a	1
3	p	bcbabbaa	ba	2
4	p	cbabbaa	bba	2
5	p	babbaa	cbba	3
6	p	babbaa	Sbba	6
7	p	babbaa	XSbba	9
8	p	abbaa	bXSbba	2
9	p	bbaa	abXSbba	1
10	p	bbaa	XabXSbba	9
11	p	bbaa	XbXSbba	8
12	p	bbaa	Sba	5
13	p	bbaa	XSba	9
14	p	baa	bXSba	2
15	p	aa	bbXSba	2
16	p	aa	XbbXSba	9
17	p	aa	XbXSba	7
18	p	aa	Sa	5
19	p	a	aSa	1
20	p	a	XaSa	9
21	p	a	XSa	8
22	p	e	aXSa	1
23	p	e	XaXSa	9
24	p	e	S	4
25	q	e	e	10

Answer 2

a.

The Turing Machine TM = (K,Σ,δ,s,H)
where K = { $q_a, q_b, q_c, q_d, q_e, q_f, q_g, q_h, q_i, q_j, H$}
Σ = x,y,1,$\sqcup$,$\triangleright$,T
s is the start state which q_a
The transition function can be defined as follows:

Function	Return value
$\delta(q_a, y)$	$(q_a, \rightarrow)$
$\delta(q_a, \sqcup)$	(q_d, H)
$\delta(q_e, x)$	$(q_e, 1)$
$\delta(q_e, y)$	$(q_e, 1)$
$\delta(q_e, 1)$	$(q_e, \leftarrow)$
$\delta(q_d, 1)$	$(q_c, \leftarrow)$
$\delta(q_d, x)$	$(q_d, 1)$
$\delta(q_d, y)$	$(q_d, 1)$
$\delta(q_c, x)$	$(q_c, \rightarrow)$
$\delta(q_a, \sqcup)$	$(q_b, \rightarrow)$
$\delta(q_b, 1)$	(q_c, x)
$\delta(q_c, 1)$	(q_b, y)
$\delta(q_c, \sqcup)$	$(q_d, 1)$
$\delta(q_b, \sqcup)$	$(q_e, \leftarrow)$
$\delta(q_e, \sqcup)$	$(q_f, \rightarrow)$
$\delta(q_f, 1)$	(q_g, T)
$\delta(q_g, \sqcup)$	$(q_h, \leftarrow)$
$\delta(q_g, T)$	$(q_g, \rightarrow)$
$\delta(q_g, 1)$	$(q_g, \rightarrow)$
$\delta(q_h, 1)$	$(q_i, \sqcup)$
$\delta(q_e, \sqcup)$	$(q_h, \leftarrow)$
$\delta(q_i, T)$	$(q_h, \rightarrow)$
$\delta(q_i, \sqcup)$	$(q_i, \leftarrow)$
$\delta(q_i, 1)$	$(q_i, \leftarrow)$
$\delta(q_j, 1)$	$(q_h, \leftarrow)$
$\delta(q_h, T)$	$(q_j, 1)$
$\delta(q_h, \sqcup)$	$(h, \sqcup)$

Answer 3

The given machine do not have ability to move its head to the left direction. Since its head cannot move to the left direction, we can say that the machine do not have memory. Since it has no memory, it cannot remember the symbols which are read. The machine is equivalent to DFA since it has no memory and it recognizes regular languages. The given Turing Machine:
TM = (K,Σ, δ, q_0, F), where K is set of states, Σ is alphabet, δ is transition function, q_0 is start state, and F is set of final states. Define TM' = (K',Σ, δ, q_0, F), K' = K$\cup$QxΓ.
s=q_0, F$\cup$FxΣ and we can state the new transition function as follows,

- If δ(q,a) = (q',T,R) (that means TM moves its head to the right direction while reads the input symbol a), TM' contains a transition (q,a,q') and if the transition δ(q,a)=(q',T,S)

(that means TM is staying while it reads the input symbol a), TM' contains a transition ((q,T),e,(q',P)).

- If δ(q,T) = (q',P,R) (that means TM moves its head to the right direction while reads the tape symbol T), TM' contains a transition ((q,T),e,q'), and if δ(q,a) = (q',P,S) (that means TM is staying while it reads the tape symbol T), TM' contains a transition ((q,T),e,(q',P)).

Since we construct a finite automaton which is equivalent to the given turing machine TM, we prove that the given turing machine TM is recognizing regular languages.

Answer 4

a.

Turing machine TM = (K,Σ, δ, s, (q_{accept}, q_{reject})) where K is set of states, Σ is the alphabet which do not contain the symbol γ that stands for the dequeue operation, δ is the transition function, s is start state, q_{accept} is the acceptance state and q_{reject} is the rejection state.
The transition function is K x Σ x Σ $\rightarrow$ K x $\gamma \cup \sigma$, in order to make sure that the queue is only allows the look front end and rear end and current state.

b.

The configuration of the TM is in the form of K x Σ^* where σ^* stands for the input string.

c.

(q_1,$x_1w_1x_2$) $\vdash$ (q_2,$y_1w_2y_2$) if and only if for some a $\in$ ($\gamma \cup \Sigma$) and
δ(q_1,x_1,x_2) = (q_2,a) and one of the:

- $x \in \Sigma$, $x_1 = y_1$, $w_2 = w_1x_2$, $b_2 = x$
- $x = \gamma$, $w_1 = y_1w_2$,$x_2 = y_2$

d.

In order to prove the given fact, we need to show how both,queue based turing machine and deterministic turing machine, can simulate the other turing machine. **Simulate a deterministic Turing machine with the queue automaton**
Front head of the queue automaton is one of the heads of the deterministic Turing machine.
There should be special symbols to show the end and begining of the string at the rear end and front head.
To move right, autoamaton will dequeue and enqueue until it reaches the beginning symbol that show the front of the string and is dequeued always before reading the current symbol, then, is pushed after pushing the written value. This is a cycle, we can simulate the right operator by

using this cycle for one time and enqueue the dequeued symbol.
To move left, we should preserve the beginning symbol, and cycle through the entire queue when we need to read. If we read to the left in the deterministic turing machine, we should simulate it by cycling the the entire queue until it reaches the beginning symbol which indicates the beginning of the queue.
Simulate the queue automaton with the deterministic Turing machine
Again, there should be special symbols to show the end and begining of the string. We should go to the left direction until we see the begining symbol and go one unit to the right direction to reach the front end of the string. We should go the right direction until wee see the end symbol and go one unit to the left to reach the rear end of the string.
Enqueue: reaching the end symbol and writing the new element, then go one unit to the right direction and writing the end symbol to there.
Dequeue: reaching the front and writing the begining symbol to there.

e.

TM = (K,Σ,δ,s,(q_{accept}, q_{reject}))
K = $\{q_{start}, q_{accept}, q_{reject}, q_0, q_x, q_{xx}, q_{xy}, q_{xz}, q_{zx}, q_{xT}, q_{Tx}, q_{Ty}, q_c, q_y, q_{yy}, q_{yx}, q_{yz}, q_{zy}, q_{yT}, q_n\}$
Σ = a,b,c,T
s = q_{start}
δ is defined as follows:

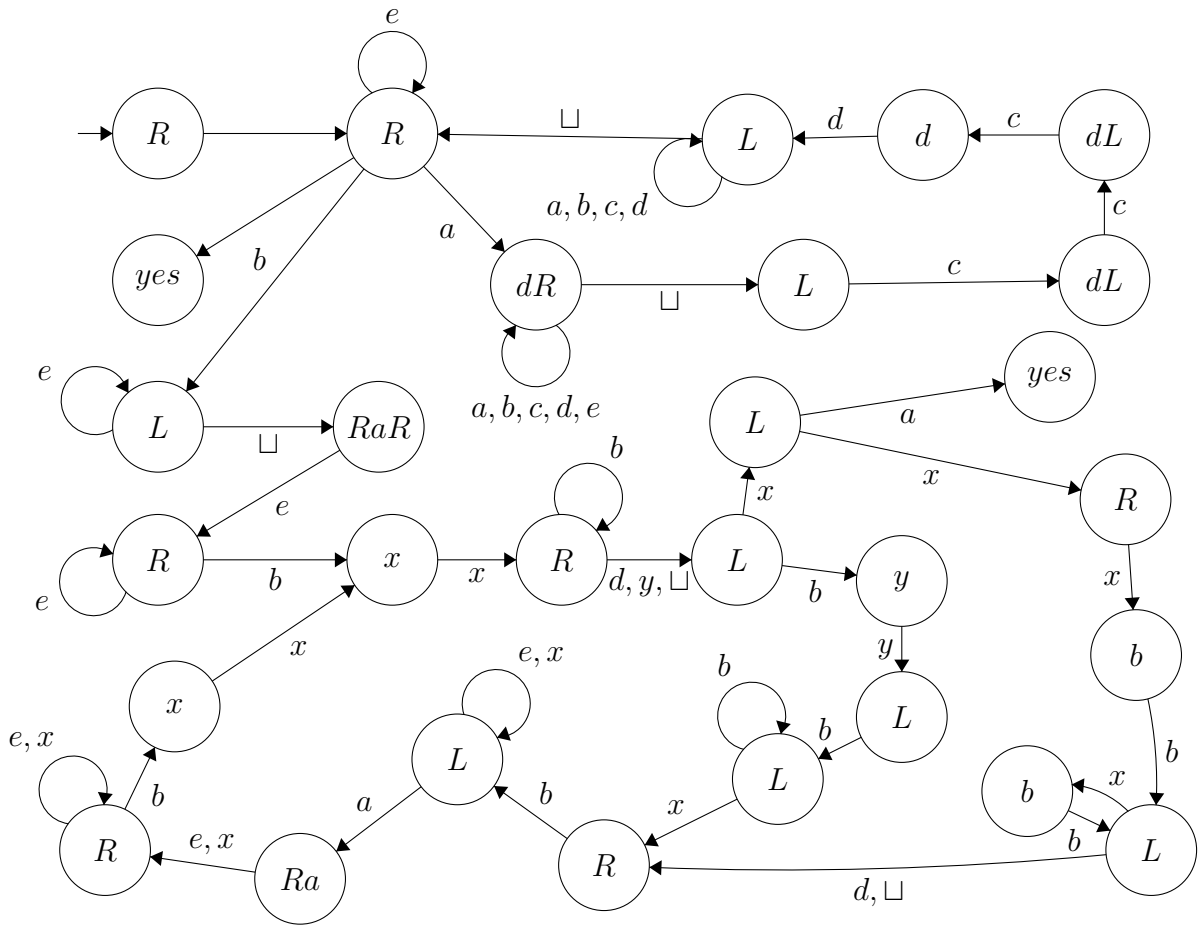
- $\delta(q_{start}, c, c) = (q_{accept,h})$
- $\delta(q_{start}, \bar{c}, \bar{c}) = (q_0, T)$
- $\delta(q_{start}, \bar{c}, c) = (q_{reject}, h)$
- $\delta(q_{start}, c, \bar{c}) = (q_{reject}, h)$
- $\delta(q_0, c, M) = (q_c, \gamma)$
- $\delta(q_0, T, c) = (q_n, \gamma)$
- $\delta(q_n, c, c) = (q_{accept}, h)$
- $\delta(q_n, \bar{c}, c) = (q_{reject}, h)$
- $\delta(q_c, T, T) = (q_{accept}, h)$
- $\delta(q_c, \bar{T}, T) = (q_{reject}, h)$
- $\delta(q_0, a, \Sigma) = (q_x, \gamma)$
- $\delta(q_x, a, \Sigma) = (q_{xx}, \gamma)$

- $\delta(q_{xx}, \Sigma, \Sigma) = (q_x, a)$
- $\delta(q_x, b, \Sigma) = (q_{xy}, \gamma)$
- $\delta(q_{xy}, \Sigma, \Sigma) = (q_x, b)$
- $\delta(q_0, b, \Sigma) = (q_y, \gamma)$
- $\delta(q_y, b, \Sigma) = (q_{yy}, \gamma)$
- $\delta(q_{yy}, \Sigma, \Sigma) = (q_y, b)$
- $\delta(q_y, a, \Sigma) = (q_{yx}, \gamma)$
- $\delta(q_{yx}, \Sigma, \Sigma) = (q_y, a)$
- $\delta(q_y, c, \Sigma) = (q_{yz}, \gamma)$
- $\delta(q_{yz}, \Sigma, \Sigma) = (q_{zy}, c)$
- $\delta(q_{zy}, b, \Sigma) = (q_0, \gamma)$
- $\delta(q_{zy}, \bar{b}, \Sigma) = (q_{reject}, h)$
- $\delta(q_y, T, \Sigma) = (q_{yT}, \gamma)$
- $\delta(q_{yT}, \Sigma, \Sigma) = (q_{Ty}, T)$
- $\delta(q_{Ty}, b, \Sigma) = (q_0, \gamma)$
- $\delta(q_{Ty}, \bar{b}, \Sigma) = (q_{reject}, h)$
- $\delta(q_x, c, \Sigma) = (q_{xz}, \gamma)$
- $\delta(q_{xz}, \Sigma, \Sigma) = (q_{zx}, c)$
- $\delta(q_{xz}, a, \Sigma) = (q_0, \gamma)$
- $\delta(q_{xz}, \bar{a}, \Sigma) = (q_{reject}, h)$
- $\delta(q_x, T, \Sigma, \Sigma) = (q_{xT}, \gamma)$
- $\delta(q_{xT}, \Sigma, \Sigma) = (q_{Tx}, T)$
- $\delta(q_{Tx}, a, \Sigma) = (q_0, \gamma)$
- $\delta(q_{Tx}, \bar{a}, \Sigma) = (q_{reject}, h)$

The overline means that a symbols from alphabet that is not that symbol.

Answer 5

a.



b.

The grammer is as follows:

- $S_1 \rightarrow$ b — S
- S $\rightarrow$ aBccc — aBSccc
- Ba $\rightarrow$ aB
- aB $\rightarrow$ aBδ
- P $\rightarrow$ e
- *delta* $\rightarrow$ e
- PB $\rightarrow$ P

- bB $\rightarrow$ Bbb
- bδB $\rightarrow$ bBδ
- Bδ $\rightarrow$ PbBδ

Answer 6

a.

L_1
Since regular grammers are subset of the context free grammers, they are context free and recursively enumerable. We can conclude that there exists a turing machine M_1 that recognizes regular grammers.
L_2
If we generalize the context free grammers, we get the unrestricted grammers and to generate a language by using grammer, it must be recursively enumerable. Furthermore, recursively enumerable languages are accepted by the Turing machines, there exits a turing machine M_2 which accepts the language L_2.
L_3
Same logic with L_2 and M_2.
L_4
The second part of the given language which is $bara^*b^*$ is regular, since it complementation of a regular language and regular languages are closed under complementation. Moreover, the $\bar{L_4} \cap a^*\bar{b}^*$ is recursively enumerable since a turing machine accepts it. L_4 must be a recursively enumerable, to that language be recursively enumerable. Since L_4 is recursively enumerable, there exits a turing machine M_4 which accepts the language L_4.
L_5
Same logic with L_2 and L_3.

b.

Sinec L_1, L_2, L_3 are context free or regular language, they have algorithms. Regular languages are recursive, since it has finitely many states and has no memory, thus, we can say that finite automaton will always stop. Context free languages are recursive since the push down automaton that recognizes context free languages has finite input and stack, and that will also always stop. However, L_4 and L_5 cannot have algorithm for some cases, since they are recursively enumerable languages, and recursively enumerable languages are the superset of the recursive languages.

c.

We should define two machines one recognizes $\bar{L_2}L_1 \cap L_5$ and the other recognizes $L_3^*\bar{L_4}$.
To recognize $L_3^*\bar{L_4}$, the machine M_Y:

In accepting states of M_3 is going either starting state of the M_4 or starting states of itself. To recognize $\bar{L}_2 L_1 \cap L_5$, the machine M_X:
It should copy the given string as sxs where x not in Σ. After that, it should move through to first string s and determine whether it is in the language $\bar{L}_2 L_1$. After that, it moves through the second string s and determine whether it is in the language L_5. If both conditions are satisfied, it should accept the string.
We should have a machine that include both of the machine we defined. It should choose undeterministicly M_X and M_Y. If one the machines accept the string, then the string is accepted.

d.