Student Information

Full Name : Yaşar Cahit Yıldırım
Id Number : 2310647

Answer 1

a.

(i) Let
$$\begin{aligned} S_1 &\rightarrow aS_1a \mid bS_2 \\ S_2 &\rightarrow aS_2a \mid b \end{aligned} \quad \text{in } G = \Big(V = \{S_1, S_2, a, b\}, \Sigma = \{a, b\}, R, S_1\Big)$$

(ii) Let
$$\begin{aligned} S_1 &\rightarrow aS_2aS_2a \mid bS_2bS_2b \\ S_2 &\rightarrow aS_2a \mid aS_2b \mid bS_2a \mid bS_2b \mid a \\ S_3 &\rightarrow aS_3a \mid aS_3b \mid bS_3a \mid bS_3b \mid b \end{aligned} \quad \text{in } G = \Big(V = \{S_1, S_2, S_3, a, b\}, \Sigma = \{a, b\}, R, S_1\Big)$$

(iii) Let
$$\begin{aligned} S_1 &\rightarrow S_2C \mid S_3C \mid AS_4 \mid AS_5 \\ S_2 &\rightarrow a \mid aS_2 \mid aS_2b \\ S_3 &\rightarrow b \mid S_3b \mid aS_3b \\ S_4 &\rightarrow b \mid bS_4 \mid bS_4c \\ S_5 &\rightarrow c \mid S_5c \mid bS_5c \end{aligned} \quad \text{in } G = \Big(V = \{S_1, S_2, S_3, S_4, s_5, a, b, c\}, \Sigma = \{a, b, c\}, R, S_1\Big)$$

(iv) Let
$$\begin{aligned} S_1 &\rightarrow S_2S_3 \\ S_2 &\rightarrow S_2S_4 \mid \epsilon \\ S_3 &\rightarrow aS_3a \mid bS_3b \mid acS_2ca \mid bcS_2cb \\ S_4 &\rightarrow aS_4 \mid bS_4 \mid ac \mid bc \end{aligned} \quad \text{in } G = \Big(V = \{S_1, S_2, S_3, S_4, a, b, c\}, \Sigma = \{a, b, c\}, R, S_1\Big)$$

b.

We are dealing with sets so we can prove this by set equality. Let our language be L.

$\mathcal{L}(G) \subseteq L$:

Let's examine our grammar a bit. If we ignore the transitions $A \to BAA$ and $B \to ABB$ we can easily see that A is a state which we come with one less a than there are b's from state S. And in this state we either finish the string with an a or add an a and return to the start state. B is the same as A with b's so these two are used to balance the number of a's and b's.

Now if we consider the $A \to BAA$ case with the help of above understandment, one B and two A's will in the long run result us with two extra a's and one extra b **from the numbers we are on**. Meaning that whatever depth the recursion will be, at the end we will get our two a and one b terminals in order for recursion to end. The same will apply to $B \to ABB$. We can conclude that whatever string our grammar create, it will be in the language, thus $\mathcal{L}(G) \subseteq L$.

$L \subseteq \mathcal{L}(G)$:

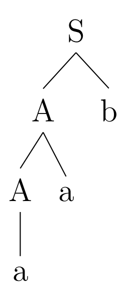
We can use induction to show this is the case.

Base Case: Smallest nonempty strings in our language are ab and ba, these are in our grammar.
Inductive Hypothesis: Let's assume all words w where $|w| \leq n$ in the language can be created by our grammar.
Inductive Proof: Let's consider a word w with $|w| = n + 2$, w either starts with an a or b. If it starts with an a it can be split as $aubv$ where $u, v \in L$ and $S \Rightarrow aB \Rightarrow aABB \Rightarrow aABbS$. v equals to S and $|v| \leq n$ so it can be created. We know A creates strings that has one more a and B strings that has one more b. So $u = AB$ where $|v| \leq n$ is created also. Same applies if w starts with a b, we split it as $buav$ and $bBAaS$. Thus $L \subseteq \mathcal{L}(G)$.
Since $\mathcal{L}(G) \subseteq L$ and $L \subseteq \mathcal{L}(G)$, $L = \mathcal{L}(G)$

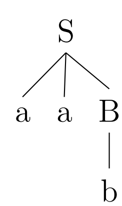
Answer 2

a.

The string aab is an example. It's two leftmost derivations are:



$S \Rightarrow Ab \Rightarrow Aab \Rightarrow aab$



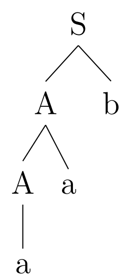
$S \Rightarrow aab \Rightarrow aab$

b.

The B non-terminal is not necessary because it only leads to create the word aab but aab is already accepted by different derivations. So an equivalent unambiguous CFG is simply:

$$G' = \{V' = \{a, b, S, A\}, \Sigma, R' = \{S \to Ab, A \to a\ Aa\}, S\}$$

c.



$$S \Rightarrow Ab \Rightarrow Aab \Rightarrow aab$$

Answer 3

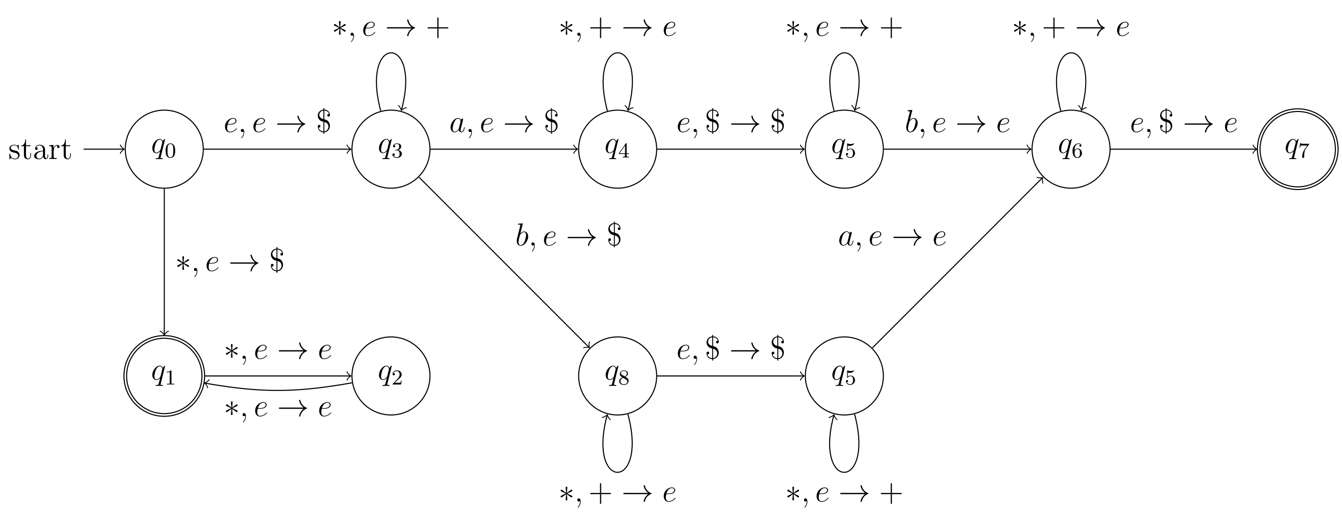
a.

$$\{a^{2n}b^{3n} \mid n \geq 0\}$$

b.

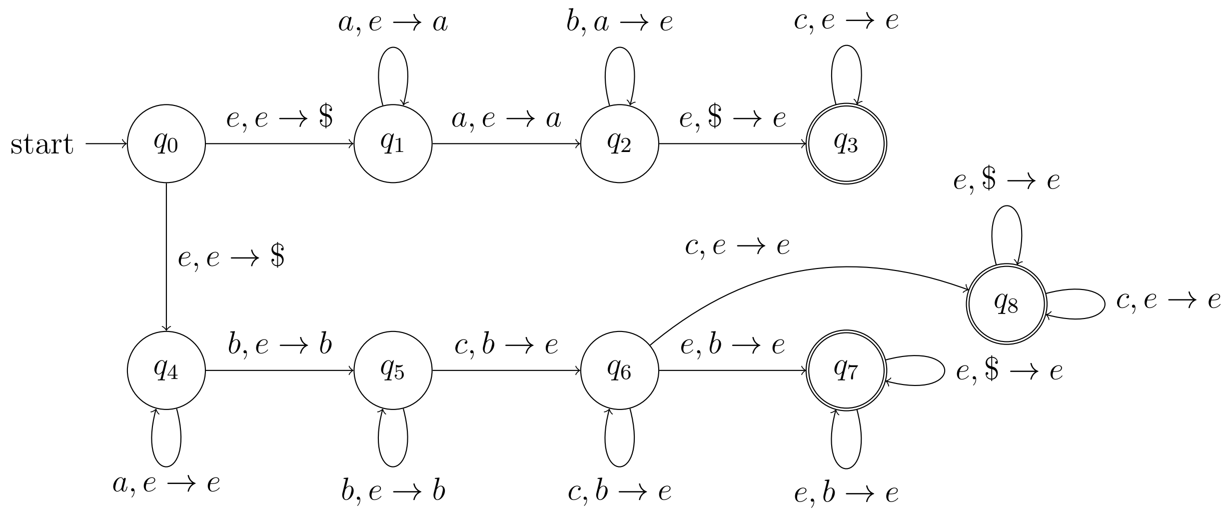
In the PDA, * symbol is used as "any symbol from alphabet" and + symbol as "just an allocator".

$$\{w_1w_2 \in \{a,b\}^* \mid |w_1w_2| \text{ is odd or } (|w_1| = |w_2| \text{ and } w_1 \neq w_2)\}$$



c.

(i)



(ii)

State	Input	Stack	Transition
q_0	aabcc	e	-
q_4	aabcc	\$	$e, e \to \$$
q_4	abcc	\$	$a, e \to e$
q_4	bcc	\$	$a, e \to e$
q_5	cc	b\$	$b, e \to b$
q_6	c	\$	$c, b \to e$
q_8	e	\$	$c, e \to e$
q_8	e	e	$e, \$ \to e$
			Accepts

State	Input	Stack	Transition
q_0	bac	e	-
q_4	bac	\$	$e, e \to \$$
q_5	ac	b\$	$b, e \to b$
			Rejects

State	Input	Stack	Transition
q_0	bac	e	-
q_1	bac	\$	$e, e \to \$$
			Rejects

Answer 4

a.

Let M be the PDA equivalent to the CFG G where $M = \Big(\{p,q\}, \Sigma, V, \Delta, p, \{q\}\Big)$ with

$$
\begin{aligned}
\Delta =\{&((p,e,e),(q,E)), && (T1)\\
&((q,e,E),(q,E+T)), && (T2)\\
&((q,e,E),(q,T)), && (T3)\\
&((q,e,T),(q,TxF)), && (T4)\\
&((q,e,T),(q,F)), && (T5)\\
&((q,e,F),(q,(E))), && (T6)\\
&((q,e,F),(q,a)), && (T7)\\
&((q,a,a),(q,e)), && (T8)\\
&((q,+,+),(q,e)), && (T9)\\
&((q,x,x),(q,e)), && (T10)\\
&((q,(,(),(q,e)), && (T11)\\
&((q,),)),(q,e))\} && (T12).
\end{aligned}
$$

b.

We know by Lemma 3.4.2 that we can convert a PDA M to M'' such that $M = M''$ and M'' is **simple**. So by not executing the last step of this lemma, we can acquire a PDA M'' which has $|\beta| \leq 1$ and $|\gamma| \leq 1$. After this, we can consider each Δ'' transition that in form $((q,x,\beta),(p,\gamma))$ where $|\beta|, |\gamma| \neq 0$ and replace it with $((q,x,\beta),(r,e))$ and $((r,e,e),(p,\gamma))$ where $x \in \Sigma$. Thus create Δ' and K' for PDA M' which has $|\beta| + |\gamma| \leq 1$.

Answer 5

a.

(i) Consider context-free grammar

$$G = \Big(V = \{S_1\}, \Sigma = \{a,b\}, R = \{S_1 \rightarrow aS_1b \mid e\}, S_1\Big) \text{ where } L(G) = \{a^nb^n : n \geq 0\}$$

Notice that $a^mb^{m+n}a^n$ is concatenation of a^mb^m and b^na^n and both of these can be generated by G. Since context-free languages are closed under concatenation, $a^mb^{m+n}a^n$ is context-free.

(ii) Let
$$\begin{aligned} S_1 &\to bS_2 \mid bS_3aa \\ S_2 &\to AS_2a \mid b \\ S_3 &\to aS_2A \mid b \\ S_4 &\to aA \mid a \end{aligned}$$
in $G = \Big(V = \{S_1, S_2, S_3, A\}, \Sigma = \{a, b\}, R, S_1\Big)$.

$\{a,b\}^* - L = \overline{L} = a\Sigma^* \cup \Sigma^* a \cup \Sigma^* bb\Sigma^* \cup G$ and since we know all these languages are context-free, $\{a,b\}^* - L$ is context-free.

b.

(i) Assume given language be context-free. Then there exists a split $uvxyz$ such that it is in the language. Let p be the pumping length and consider string $a^{p^2}b^p$.

Case 1, there are only a's in v and y: The string uv^2xy^2z has the property $p^2 + 2k \leq p^2$ where $vxy \leq p$ and $2k > 0$ so it can not be in the language.
Case 2, there are only b's in v and y: The string uv^0xy^0z has the property $p^2 \leq p^2 - 2k$ where $vxy \leq p$ and $2k > 0$ so it can not be in the language.
Case 3, there are both a's and b's in vxy: The string has the property $p^2 - k \leq (p-j)^2$ where $vxy \leq p$ and $j + k > 0$ with k is the number of a's j is the number of b's.

$$j + k \leq p,\ j \leq p - k,\ j > 0,\ k > 0$$
$$p^2 - k \leq p^2 - 2pj + j^2,\ p^2 - k \leq p^2 - 2p(p-k) + (p-k)^2$$
$$p^2 - k \leq k^2 \text{ is false because } j^2 + 2jk + k^2 \leq p^2$$

Since every case considered builds up to a contradiction, this language is not context-free.

(ii) Let L be the given language and $L' = L \cap a^*ba^*ba^*b = \{a^nba^nba^n \in \{a,b\}^* \mid n \geq 0\}$. Not that if L is context-free, then L' must be too by closure properties. Assume L' is context free. Then there exists a split $uvxyz$ such that it is in L'. Let p be the pumping length and consider string $a^pba^pba^pb$.

Case 1, there is a b in vxy: The string uv^2xy^2z has more than three b's so it can not be in the language.
Case 2, there are only a's in v and y: The string uv^2xy^2z has **at least** one group of a's not matching other(s) so it can not be in the language. This case is also valid when v and y consists different a groups divided with a b i.e. $x = b$. "at least" is emphasised for that purpose.

Since we found that L' is not a context-free language, we can conclude L is not also.

Answer 6

(i) (T/F)? False.

(ii) (T/F)? True.

(iii) (T/F)? True.

(iv) (T/F)? False.