



## پاسخ مسئله ۱.

برای tokenize کردن یک عبارت، دو اولویت وجود دارد که این دو اولویت به ترتیب عبارتند از:

۱. طول توکن

۲. بالاتر بودن اولویت گرامر

با توجه به اولویت‌های فوق، عبارت به صورت زیر توکن بندی می‌شود:

- $Q < a^1 c^3 >$
- $Q < b^1 >$
- $R < a^1 b^3 >$
- $Q < a^1 c^1 >$
- $Q < a^1 c^1 >$
- $R < a^2 b^1 >$
- $R < a^2 b^2 >$
- $S < a^1 b^1 c^1 >$

پس رشته توکن بندی شده ما به این صورت است:

**accc/b/abbb/ac/ac/aab/aabb/abc**

پاسخ مسئله‌ی ۲.

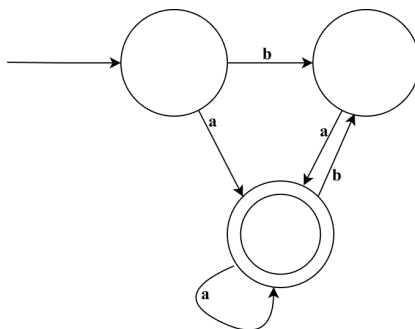
### پاسخ مسئله‌ی ۳.

توضیح کلی راه حل

۱.

الف

ابتدا DFA مربوط به این زبان را رسم می‌کنیم:



حال عبارت منظم مربوط به این زبان را از روی DFA می‌نویسیم:

$$a^+(ba)^* + ba^+(ba)^* = (a^+ + ba^+)(ba)^*$$

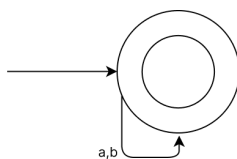
ب

توضیحات

۲.

الف

این عبارت منظم، تمامی رشته‌های متشکل از ۰ و ۱ را می‌پذیرد. پس DFA آن به این صورت خواهد بود:



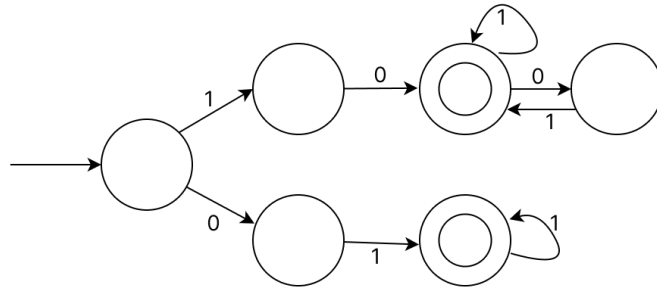
ب

با توجه به عبارت منظم داده شده، DFA را رسم می‌کنیم:

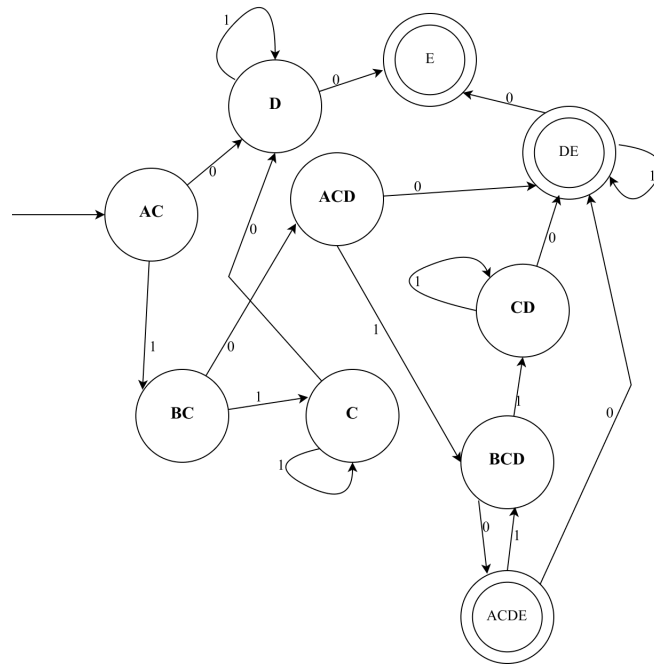
۳.

الف

برای رسم DFA از روی NFA لازم است از مجموعه  $\epsilon$ -closure مربوط به هر State استفاده می‌کنیم:



$$\begin{aligned}\epsilon - \text{closure}(A) &= \{A, C\} & \epsilon - \text{closure}(B) &= \{B\} \\ \epsilon - \text{closure}(C) &= \{C\} & \epsilon - \text{closure}(D) &= \{D\}\end{aligned}$$



ب

ابتدا عبارت منظم مربوط به NFA فوق برابر است با:

$$(10)^*(1)^*(0)(1)^*(0)$$

## پاسخ مسئله‌ی ۴.

برای Left Most Derivation از راست‌ترین E شروع کرده و عبارت‌های معادل را جایگزین می‌کنیم تا به عبارت مورد نظر برسیم:

$E \rightarrow EBE$   
 $\rightarrow [E]BE$   
 $\rightarrow [EBE]BE$   
 $\rightarrow [VBE]BE$   
 $\rightarrow [aBE]BE$   
 $\rightarrow [a!E]BE$   
 $\rightarrow [a!V]BE$   
 $\rightarrow [a!b]BE$   
 $\rightarrow [a!b]@E$   
 $\rightarrow [a!b]@[E]$   
 $\rightarrow [a!b]@[V]$   
 $\rightarrow [a!b]@[a]$

برای Right Most Derivation از راست‌ترین E شروع کرده و عبارت‌های معادل را جایگزین می‌کنیم تا به عبارت مورد نظر برسیم:

$E \rightarrow EBE$   
 $\rightarrow EB[E]$   
 $\rightarrow EB[V]$   
 $\rightarrow EB[a]$   
 $\rightarrow E@[a]$   
 $\rightarrow [E]@[a]$   
 $\rightarrow [EBE]@[a]$   
 $\rightarrow [EBV]@[a]$   
 $\rightarrow [EBb]@[a]$   
 $\rightarrow [E!b]@[a]$   
 $\rightarrow [V!b]@[a]$   
 $\rightarrow [a!b]@[a]$

حال درخت Parse را رسم می‌کنیم:

