



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 می شود:

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

پس رشته توکن بندی شده ما به این صورت است:

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

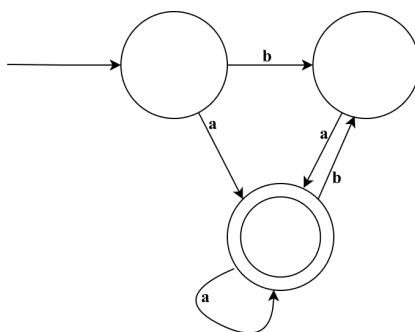
پاسخ مسئله‌ی ۲.

پاسخ مسئله‌ی ۳.

بخش 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

چون در این زبان باید دو حالت را با هم اجتماع کنیم، نیاز به کشیدن DFA نیست. دو عبارت منظم را نوشته و با هم اجتماع می‌کنیم.

این که رشته w دارای دقیقاً دو a باشد، توسط عبارت منظم $b^*ab^*ab^*$ توصیف می‌شود.

همچنین این که رشته w دارای حداقل دو b باشد توسط عبارت منظم $(a+b)^*b(a+b)^*b(a+b)^*$ توصیف می‌شود.

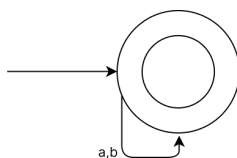
حال کافیست که دو عبارت منظم را با هم اجتماع کنیم، پس جواب نهایی ما برابر است با:

$$((b^*ab^*ab^*) + (a+b)^*b(a+b)^*b(a+b)^*)$$

بخش ۲.

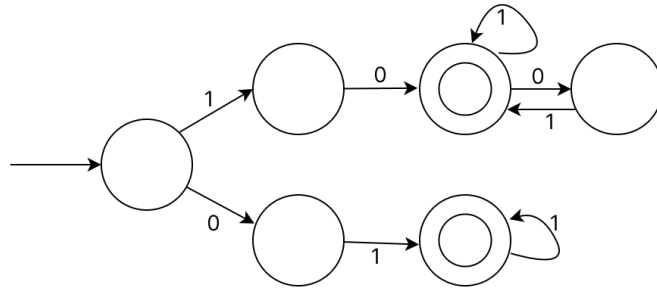
الف

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:



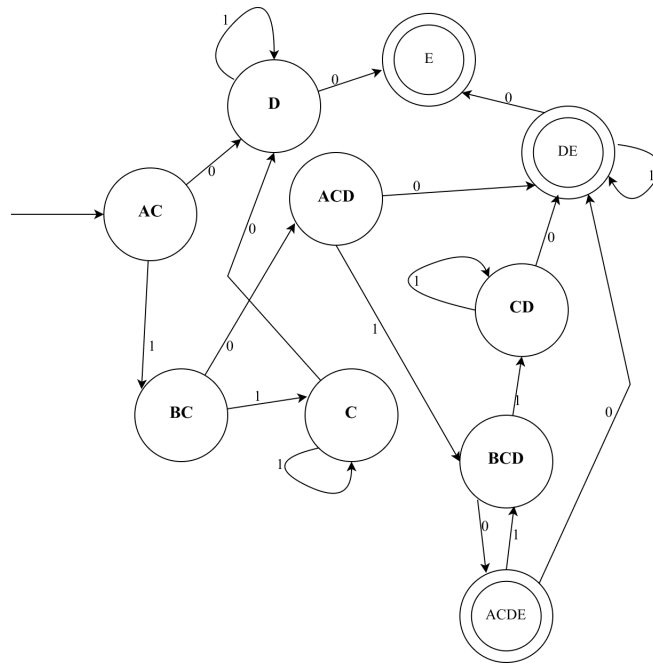
بخش ۳.

الف

برای رسم DFA از روی NFA لازم است از مجموعه ϵ -closure به هر State استفاده می‌کنیم:

$$\epsilon - \text{closure}(A) = \{A, C\} \quad \epsilon - \text{closure}(B) = \{B\}$$

$$\epsilon - \text{closure}(C) = \{C\} \quad \epsilon - \text{closure}(D) = \{D\}$$



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:

