Compiler Mid-Term Exam

1. What are the phases of a compiler?
2. Illustrate using the figure the difference between compiler and interpreter?
3. **Consider the context-free grammar**

$$S \rightarrow S\,S\,+\;|\;S\,S\,*\;|\;\mathbf{a}$$

   a) Show how the `aa+a*` string can be generated by this grammar (derivation).
   b) Construct a parse tree for this string.

4. **Eliminate the left recursion of the following grammar:**

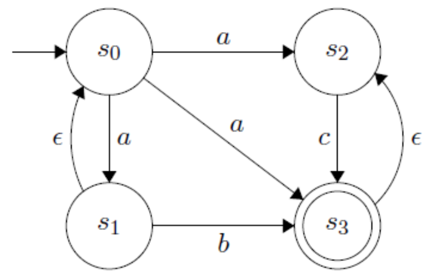
$$S \rightarrow STS \mid ST \mid T$$
$$T \rightarrow Ta \mid Tb \mid U$$
$$U \rightarrow T \mid c$$

Answer:

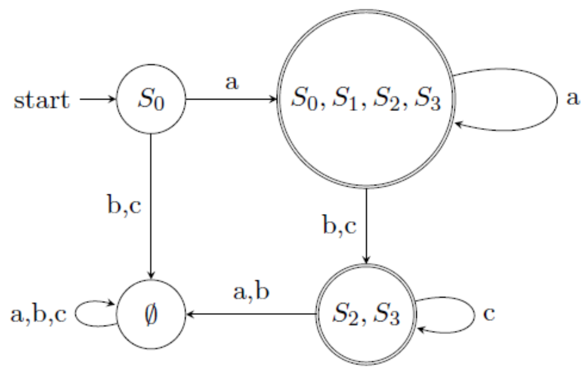
$$S \rightarrow TS'$$
$$S' \rightarrow TSS' \mid TS' \mid \epsilon$$
$$T \rightarrow cT'$$
$$T' \rightarrow aT' \mid bT' \mid \epsilon$$

5. **Convert** the following NFA with Ɛ–transitions into equivalent DFA. **<u>Note that</u>** a DFA must have a transition defined for every state and symbol pair.

(a) Original NFA, $\Sigma = \{a, b, c\}$:



DFA:



Compiler Mid-Term Exam

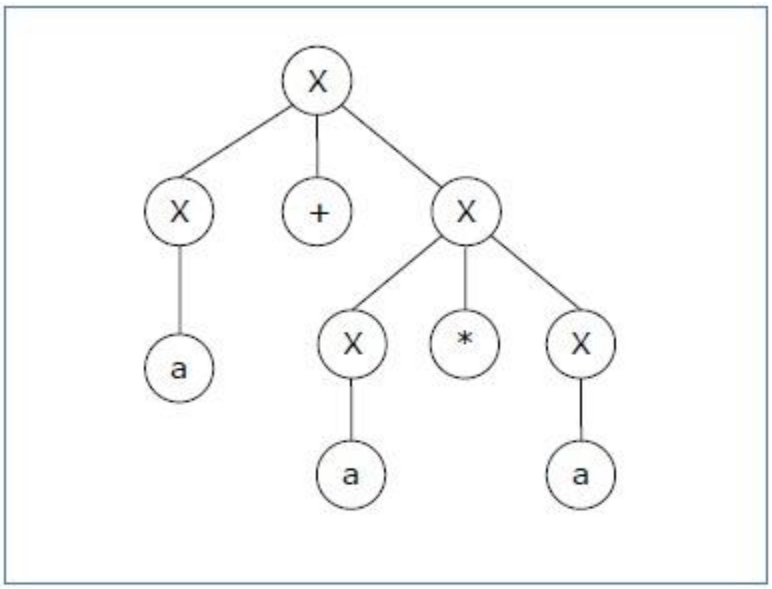
1. What is the meaning of grouping compiler phases into passes?
2. What are types of parsing?
3. Check whether the grammar G with production rules: X → X+X | X*X |X| a
is ambiguous or not?

Solution

Let's find out the derivation tree for the string "a+a*a". It has two leftmost derivations.

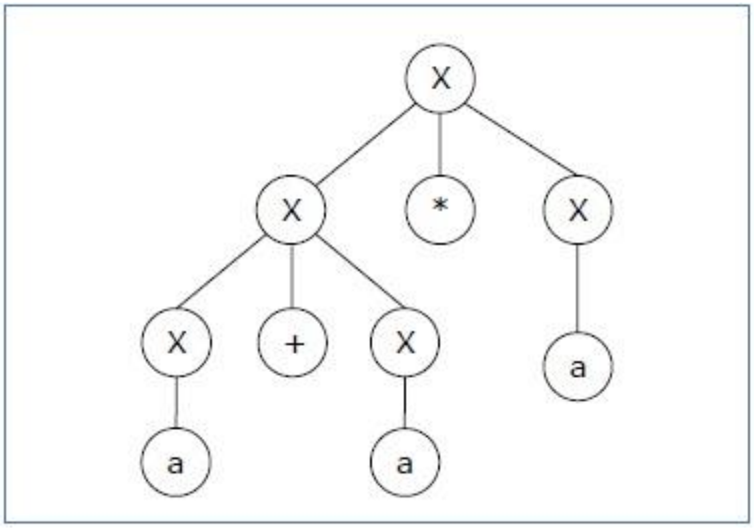
Derivation 1 − X → X+X → a +X → a+ X*X → a+a*X → a+a*a

Parse tree 1 −



Derivation 2 − X → X*X → X+X*X → a+ X*X → a+a*X → a+a*a

Parse tree 2 −



Since there are two parse trees for a single string "a+a*a", the grammar **G** is ambiguous.

4. Eliminate the left recursion of the following grammar:

E → E+T | T

T → T*F | F

F → (E) | id

Answer:

E → TE'

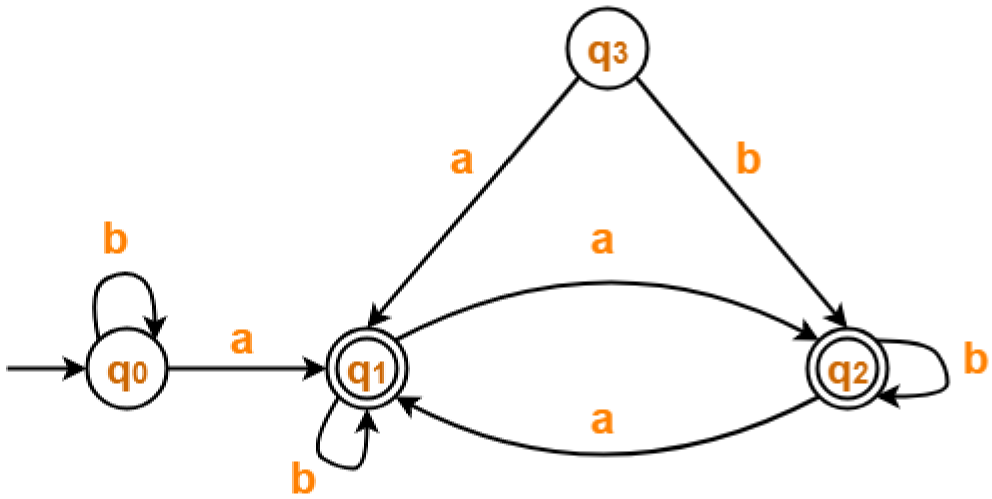
E' → +TE' | Ɛ

Then eliminate for T as:

T → FT'

T' → *FT' | Ɛ

5. Minimize the following DFA:



Answer:

Now using Equivalence Theorem, we have-

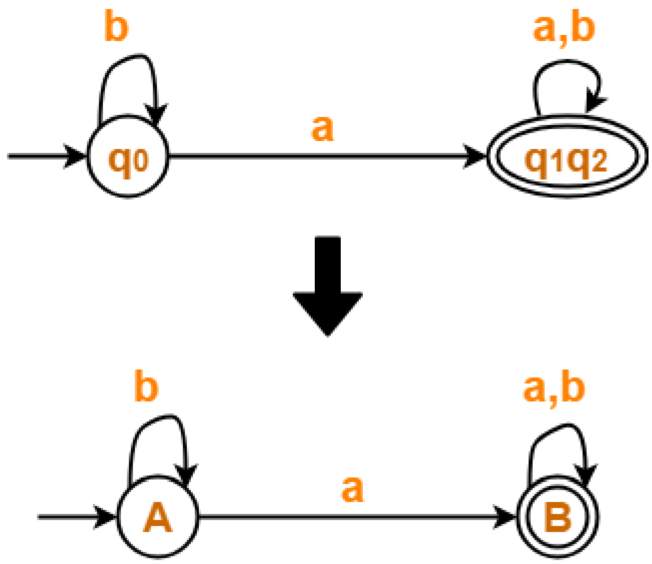
$P_0 = \{ q_0 \} \{ q_1 , q_2 \}$

$P_1 = \{ q_0 \} \{ q_1 , q_2 \}$

Since $P_1 = P_0$, so we stop.

From P_1, we infer that states q_1 and q_2 are equivalent and can be merged together.

So, Our minimal DFA is-



Minimal DFA