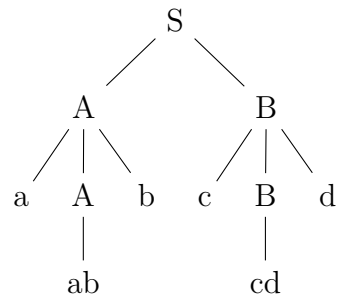
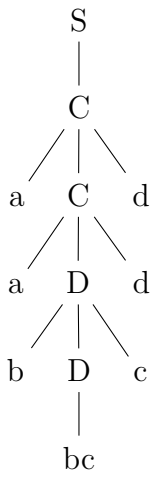
01

a) The following two top-down parse trees can be generated for the given CFG to accept the string "`aabbccdd`".





Therefore, the grammar is ambiguous.

b) Leftmost derivation:

$$
\begin{aligned}
\texttt{E} &\Rightarrow \texttt{+E} \\
&\Rightarrow \texttt{+*E} \\
&\Rightarrow \texttt{+*-E} \\
&\Rightarrow \texttt{+*-X} \\
&\Rightarrow \texttt{+*-pXq} \\
&\Rightarrow \texttt{+*-ppXqq} \\
&\Rightarrow \texttt{+*-pppYqqq} \\
&\Rightarrow \texttt{+*-ppprYwqqq} \\
&\Rightarrow \texttt{+*-pppprrwwqqq}
\end{aligned}
$$

02

a) S → D110D | D010D | D011D
   D → 0D | 1D | ε

b) S → D00 | 0
   D → 0D | 1D | ε

c) <S> → XX<S><A> | ε
   <A> → Y<A>Z | ε

d) S → XcccC
   X → AE | EB // extra a's or b's
   E → aEb | ab // $a^n b^n$
   A → aA | a // a*
   B → bB | b // b*
   C → cC | c // c*

03

a)
```
E -> T | E + T
T -> F | T * F
F -> I | (E)
I -> a | b | Ia | Ib | I0 | I1

// new starting state and
// non-terminal states for all the terminals
S -> E
E -> T | E + T
T -> F | T * F
F -> I | (E)
I -> a | b | Ia | Ib | I0 | I1
A -> a
B -> b
P -> +
M -> *
O -> 1
Z -> 0
L -> (
R -> )

// replacing grouped terminals with non-terminals
S -> E
E -> T | EPT
T -> F | TMF
F -> I | LER
I -> a | b | IA | IB | IZ | IO
A -> a
B -> b
P -> +
M -> *
O -> 1
Z -> 0
L -> (
R -> )

// removing groupings of more than two non-terminals
S -> E
E -> T | EU
T -> F | TV
F -> I | LW
I -> a | b | IA | IB | IZ | IO
A -> a
B -> b
P -> +
M -> *
```

```
O -> 1
Z -> 0
L -> (
R -> )
U -> PT
V -> MF
W -> ER

// replacing single non-terminals
S -> a | b | IA | IB | IZ | IO | LW | TV | EU
E -> a | b | IA | IB | IZ | IO | LW | TV | EU
T -> a | b | IA | IB | IZ | IO | LW | TV
F -> a | b | IA | IB | IZ | IO | LW
I -> a | b | IA | IB | IZ | IO
A -> a
B -> b
P -> +
M -> *
O -> 1
Z -> 0
L -> (
R -> )
U -> PT
V -> MF
W -> ER
```

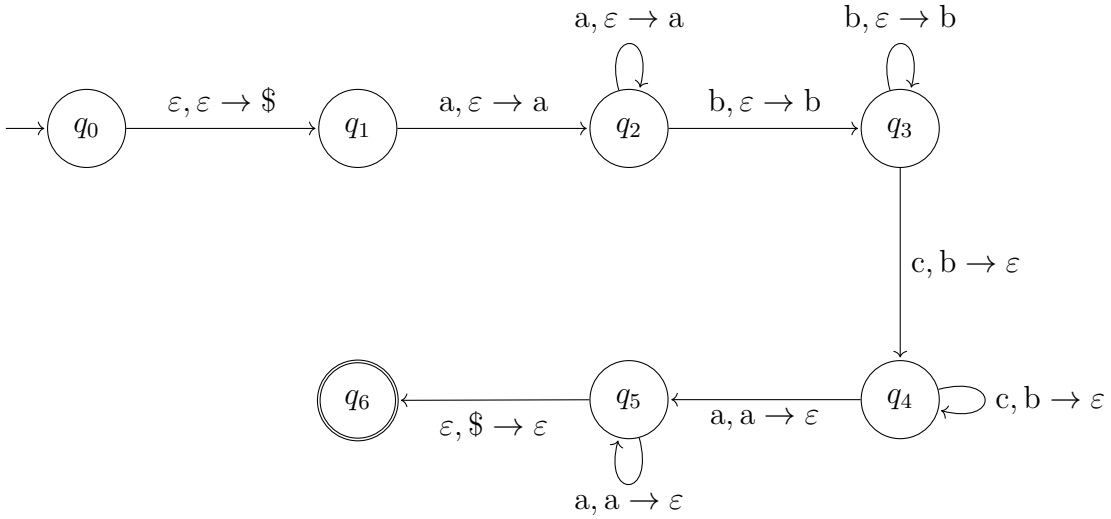
b)
```
S -> ASB | ε
A -> aAS | a
B -> SbS | A | bb

// new initial state and
// non-terminals for all the terminals and
// removing ε
I -> S | ε
S -> ASB | AB
A -> aAS | aA | a
B -> SbS | Sb | bS | b | A | bb
X -> a
Y -> b

// replacing grouped terminals
I -> S | ε
S -> ASB | AB
A -> XAS | XA | a
B -> SYS | SY | YS | b | A | YY
X -> a
Y -> b
```

```
// removing groupings of more than two non-terminals
I -> S | ε
S -> MB | AB
A -> XM | XA | a
B -> SN | SY | YS | b | A | YY
X -> a
Y -> b
M -> AS
N -> YS

// removing single non-terminals
I -> MB | AB | ε
S -> MB | AB
A -> XM | XA | a
B -> SN | SY | YS | b | XM | XA | a | YY
X -> a
Y -> b
M -> AS
N -> YS
```

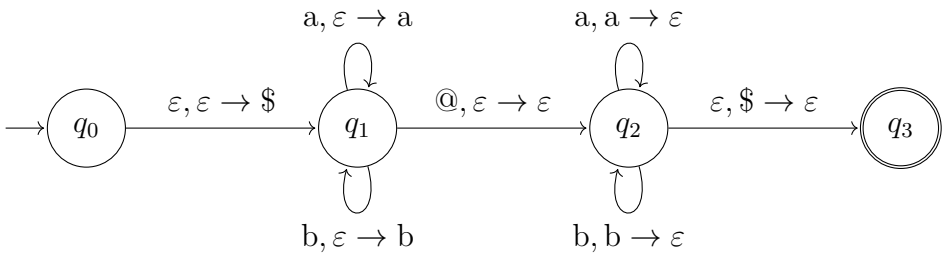
c)
```
A -> BAB | B | ε
B -> 00 | ε

// new initial state and
// non-terminals for all the terminals and
// removing ε
S -> A | ε
A -> BAB | BB | BA | AB | A | B
B -> 00
Z -> 0

// removing groupings of more than two non-terminals
S -> A | ε
A -> BX | BB | BA | AB | A | B
B -> 00
Z -> 0
X -> AB

// removing single non-terminals
S -> BX | BB | BA | AB | ZZ | ε
A -> BX | BB | BA | AB | ZZ
B -> ZZ
Z -> 0
X -> AB
```

04

a) $a^m b^n c^n a^m \mid m, n > 0$



b) $w@w^r$



05

Turing machine for $L = \{a^i b^j c^k \mid i < j < k, i >= 1\}$