1. (20 pts) [CFG] Consider the grammar \( G: S \rightarrow S S + | S S # | a \) and the string: \( a a + a # \)
   (a) Give a rightmost derivation for the string. (5pts)
   (b) Give a parse tree for the string. (5 pts)
   (c) Is the grammar ambiguous or unambiguous? why? (5pts)
   (d) Describe the language generated by this grammar? (5 pts)

2. [5 pts] Consider the grammar: \( S \rightarrow S a | a \)
   Which class of grammar does this grammar belong to?
   (a) LL(1)    (b) LL(2)   (c) LR(0)   (d) LR(1)  (e)  LALR(1)
   Ans: ______________

3. (30 pts) [LL(1) parsing] Given the following grammar G:
   \[
   \begin{align*}
   S & \rightarrow E \$ \\
   E & \rightarrow T E' \\
   E' & \rightarrow + T E' | \epsilon \\
   T & \rightarrow F T' \\
   T' & \rightarrow * F T' | \epsilon \\
   F & \rightarrow (E) | d
   \end{align*}
   \]
   (a) Find all nonterminals in G that are nullable. (4pts)
   Ans: __________________________
   (b) Fill in the following table with FIRST and FOLLOW sets for the grammar: (10 pts)
(c) Complete the following LL(1) table for the grammar: (10 pts)

<table>
<thead>
<tr>
<th>N</th>
<th>FIRST(N)</th>
<th>FOLLOW(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>(, d</td>
<td></td>
</tr>
</tbody>
</table>

(d) Is this grammar LL(1)? why? (4 pts)
Ans: ____________________________

4. (10pts) Let $G = (N, \Sigma, S, P)$ be a CFG where $N$ is the set of non-terminal symbols, $S$ is the set of terminal symbols and $P$ the set of production rules. Define the relation $\text{First} \subseteq N \times \Sigma$ as the set of all pairs $(A, a)$ such that $a$ can appear in the first position of some sentential forms derived from $A$, i.e., $\text{First} = \{ (A, a) \in N \times \Sigma | A \Rightarrow^* a \beta \text{ for some } \beta \in (N \cup \Sigma)^* \}$. The relation $\text{First}$ can be defined inductively as follows:

(a) Basis: $(A, a) \in \text{First}$ for all $A \Rightarrow B_1B_2\cdots B_n a \cdots \in P$, where $n \geq 0$ and $B_1, \cdots, B_n$ are all nullable.

(b) Closure:
If $(C,c) \in \text{First}$ and $B_1, \cdots, B_n$ are nullable and $A \Rightarrow B_1B_2\cdots B_n C \cdots \in P$, then $(A,c) \in \text{First}$.

Now let $\text{Last} = \{ (A, a) \in N \times \Sigma | \exists \beta \in (N \cup \Sigma)^* \text{ s.t. } A \Rightarrow^* \beta a \}$. List an inductive definition for the relation $\text{Last}$:

(a) Basis:
5. [10 pts] Write regular expressions for the following languages:
   (a) All strings over \( \{a, b\} \) containing at least 2 b's.
   (b) Decimal number 0 ~ 255. Note that all leading zero's except 0 are not accepted. (4pts)
   (c) Binary numbers that are greater than 101. Note. leading zero's are permitted.

6. [20 pts ; minimization of FA] For the finite automata F given below,
(a) Complete the following chart to find all pairs of states that cannot be merged without affecting the accepted language. [12 pts]

(b) Draw the minimized finite automata according the result of (a). [8 pts]

7. (15 pts) [From Regular Expression to DFA] Consider the regular expression $\alpha$ given below:

$$(a+b)^* \ ba \ (a+b)$$

(a) Write an NFA which recognizes the language represented by $\alpha$ and contains no more than 4 states. Remember to label each state. [5 pts]
(b) Convert the NFA you got at (a) to an equivalent DFA by using the subset construction. Note: you must identify each state of your DFA with the set of old NFA states it corresponds to and make sure all unused states are removed. [10pts]

8. (25 pts)[LR parsing] Consider the grammar

0. \( S \rightarrow E \)
1. \( E \rightarrow E \ E \ + \)
2. \( E \rightarrow E \ E \ ! \)
3. \( E \rightarrow a \)

(a) Build the LR(0) NFA for this grammar [5pts]

(b) Build the LR(0) DFA for this grammar [10pts]

(c) Is this an LR(0) grammar? Give evidence. [5pts]

(d) Is this an SLR(1) grammar? Give evidence. [5pts]