1. (25 points.) Two non-empty sets may be related in exactly one of five ways. Use these five possibilities to answer the questions below.

**dis** Disjoint. \( X \cap Y = \emptyset \)

**int** Intersecting. \( X \subseteq X \cup Y \) and \( Y \subseteq X \cup Y \)

**sub** Subset. \( X \subseteq Y \)

**sup** Superset. \( Y \subseteq X \)

**equ** Equals. \( X = Y \)

What is the relationship between the set \( X \) and the set \( Y \) where \( X \) and \( Y \) are each of the pairs of sets below. Write **dis**, **int**, **sub**, **sup**, or **equ** in the box.

(a) **dis**  \( X \) is the set of SLR grammars; \( Y \) is the set of context-free grammars.

(b) **int**  \( X \) is the set of SLR grammars; \( Y \) is the set of LL(0) grammars.

(c) **sub**  \( X \) is the set of LR(1) grammars; \( Y \) is the set of LL(1) grammars.

(d) **sup**  \( X \) is the set of LALR(1) grammars; \( Y \) is the set of LR(1) grammars.

(e) **equ**  \( X \) is the set of ambiguous grammars; \( Y \) is the set of unambiguous grammars.
2. (10 points). Does there exist a regular expression \( r \) for which the formal language denoted by \( r \) is exactly the set of syntactically correct Mini-Java programs. Explain briefly why or why not.

Circle the correct answer: yes / no.


Circle the correct answer: yes / no.
4. (15 points). Using the construction method in the book, convert the following regular expression
over the alphabet \( \Sigma = \{a, b, c, d\} \) to a nondeterministic finite automata (NFA).

\[(ab^*) + (c^*d)^*\]

Do not simplify. Be sure to label the arcs and indicate the final states. Label the states 0, 1, 2, etc, where 0 is the initial state. Show the completed result in the space below. Use scratch paper if necessary.
5. (15 points). Convert the previous NFA over the alphabet \{a, b, c, d\} to a DFA using the subset construction. Be sure to label the states of your DFA with *sets* of the NFA’s state labels, so that the correspondence is clear. Do not simplify.
Consider the following grammar $G$ for the remaining questions:

1. $S \rightarrow * Q \mid P$
2. $Q \rightarrow E ; \mid T$
3. $P \rightarrow T ; \mid E$
4. $T \rightarrow U$
5. $E \rightarrow U$
6. $U \rightarrow *$

It is the case that one of the states of the LR(0) parsing automaton for $G$ contains the following two LR(0) items:

$E \rightarrow U \bullet$

$T \rightarrow U \bullet$

6. (10 points.) Compute nullable, FIRST and FOLLOW for all nonterminals of $G$. 

<table>
<thead>
<tr>
<th></th>
<th>Nullable</th>
<th>FIRST</th>
<th>FOLLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$U$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7. (10 points.) Without computing the LL(1) parsing table, is the grammar $G$ is LL(1)? Explain briefly why or why not.

Circle the correct answer: yes / no.

8. (10 points.) Without computing the parsing automata, but using the information above. What can you say about whether or not the grammar is LR(0)?

9. (10 points.) Without computing the parsing automata, but using the information above. What can you say about whether or not the grammar is SLR?
10. (15 points.) Build the LR(1) parsing automaton for the grammar $G$. 
11. (15 points.) Give the entire LR(1) parsing table for the grammar $G$. Is the grammar LR(1)?

Circle the correct answer: yes / no.
12. (10 points.) Carefully describe the changes in LALR(1) parsing table from the LR(1) parsing table. Is the grammar LALR(1)?

Circle the correct answer: yes / no.