Total marks 50 - Total time is 50 minutes. Answer all 4 questions.

Instructions: This midterm is open book, open notes. Non-programmable calculators allowed. No electronic communication devices allowed.

The line numbers in grammars are for reference only and are not part of the grammars. Ellipses ... indicate omitted, correct text. \( \epsilon \) indicates an empty string.

If you need to make any assumptions in order to answer a question, state the assumptions clearly in your answer book.

Remember: an explanation of your reasoning is always more important than the (correct) Yes/No answer.

I. [10 marks] Regular expressions and automata.

1. [4 marks] Write a regular expression defining the language of comments consisting of a string surrounded by /* and */, unless it is inside double quotes ("ibrated. For example, */aaa */ is a comment and so is */a*/a*/, but */aa*/ is not. Assume your alphabet is \( \Sigma \).

2. [4 marks] Design a finite automaton for this language.

3. [2 marks] Is your automaton deterministic or non-deterministic? Explain your answer.

II. [16 marks] LL grammars and parsing.

Consider the following context-free grammar:

\[
1. \quad S ::= S (S) S \\
2. \quad | \quad \epsilon
\]

1. [2 marks] Give either a leftmost or a rightmost derivation for string \(((())\)). Explain which one you produced.

2. [3 marks] Is this grammar ambiguous? Justify your answer.

3. [3 marks] What is the language generated by this grammar?

4. [4 marks] Is this grammar LL(1)? If not, construct an equivalent one that is.

5. [4 marks] Prove that it is LL(1).
III. [14 marks] LL and LR parsing.

1. [8 marks] Show that the following grammar is SLR(1):

1: $S ::= S A$
2: $|$ $A$
3: $A ::= a$

2. [3 marks] Without building the table for LALR(1), explain why this grammar is LALR(1) as well. Just saying that $L(SLR(1)) \subseteq L(LALR(1))$ gets 0 marks.

3. [3 marks] Explain the intuition why LR parsing recognizes more grammars than LL.

IV. [10 marks] Run-time analysis.
Show the run-time checks that have to be make to make sure that the following program does not have type and range errors.

```plaintext
var j, i: 2..50
    m, n: integer
    A: array [3..75] of -10..-5

i = i-1;
m = i+j;
A[m] = n;
A[3m-7] = -2j;
```