Compilers [Spring 2020]
Test I

NAME: ____

Instructions:

1) This test is 7 pages in length.

2) You have 75 minutes to complete and turn in this test.

3) Short-answer questions will be graded on how clearly you've communicated the necessary ideas. Respond in complete English sentences. Avoid using bullet points and enumerated lists.

4) This test is closed books, notes, papers, friends, phones, neighbors, smartwatches, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.

6) Write and sign the following: "I pledge my Honor that I have not cheated, and will not cheat, on this test."

____

____

Signed: ____

1. [4 points]
As defined in class, what is a compiler? [1-2 sentences]

2. [6 points]
Why do compilers normally have distinct lexing and parsing phases? [2-4 sentences]

3. [2 points]
As defined in class, what is a language? [1 sentence]

4. [5 points]
What makes a grammar context free? [1-2 sentences]

5. [6 points]
Karen implemented 777 compilers, each with a front and back end. What are the minimum and maximum numbers of “ends” Karen has written?

6. [10 points]
As we did in class, draw a diagram showing the major phases of a standard compiler and the result of each phase. Label the portions that comprise the front and back ends.

7. [10 points]
Define a grammar C and prove that C ∈ LL(2)\LL(1).
Hint: \ is the set-difference (i.e., subtraction) operator.

8. [15 points]

Consider the following grammar.

```
    0   S -> N$        2     A -> ε             4     B -> ε
    1   N -> AB        3     A -> Ax            5     B -> Bxy
```

Draw a minimized DFA that accepts exactly those strings derivable from `S`.

9. [12 points]

For this problem, assume that grammars may never contain a rule of the form `N -> N`.

a) Are reduce-accept conflicts possible with LR parsers? If so, illustrate such a conflict by showing an example CFG and its parse table. If not, *briefly* explain why.

b) Are shift-accept conflicts possible with LR parsers? If so, illustrate such a conflict by showing an example CFG and its parse table. If not, *briefly* explain why.

10. [30 points]

Define expression e to be arithmetic when e is e A e (addition), e M e (multiplication), or N (some natural-number literal). These arithmetic expressions obey standard rules of operator precedence and associativity.

a) Define a grammar G for arithmetic expressions and prove that G ∈ LALR(1). Make Rule 0 be `s->e$`.

b) Complete the following LALR(1) parse trace according to your table from Part (a).

Stack	Input	Action
	N A N A N M N $	

Undergraduates stop here. The remaining problem is for graduate students.

11. Draw RR(1) and RL(1) parse tables for the following grammar. [10 points]

```
0     S -> $N            1     N -> xN            2     N -> ε
```