1. Regular Expressions & Finite Automata

Consider a small language using only the letters "z", "o", and the slash character "/". A comment in this language starts with "/o" and ends after the very next "o/".

(a) Give a singular regular expression that matches exactly one complete comment and nothing else. Assume comments do not nest. For full credit, use only the core notations: ε, "ab", AB, A + B, and A* **[15 points]**

Consider the language over the alphabet $\sum = \{a, b\}$ containing strings in which number of 'a's is a multiple of 3 or number of 'b's is a multiple of 2.

(b) Write an NFA for this language. **[5 points]**

(c) Write a DFA with at most 6 states for this language **[5 points]**

2. LL Parsing

Consider the following grammar with terminals *, !, n, (, and).

$E \rightarrow F\ H$ $\quad\quad$ $F \rightarrow F\ !\ |\ G$

$H \rightarrow *\ E\ |\ \varepsilon$ $\quad\quad$ $G \rightarrow n\ |\ (\ E\ )$

(a) The grammar is not LL(1). Explain in one sentence why. **[2 points]**

(b) Fix the grammar to make it LL(1) by filling in the blanks below **[3 points]**

$E \rightarrow F\ H$ $\quad\quad$ $F \rightarrow$ _____

$H \rightarrow *\ E\ |\ \varepsilon$ $\quad\quad$ $K \rightarrow$ _____

$G \rightarrow n\ |\ (\ E\ )$ $\quad\quad$ $K \rightarrow$ _____

(c) Compute the first and follow set of the fixed grammar. **[10 points]**

	First	Follow
E		
F		
G		
H		
K		

(d) Finish the LL(1) parsing table. **[10 points]**

	*	!	(	)	$	n
E						
F						
G						
H						
K						

3. Grammars & Ambiguity

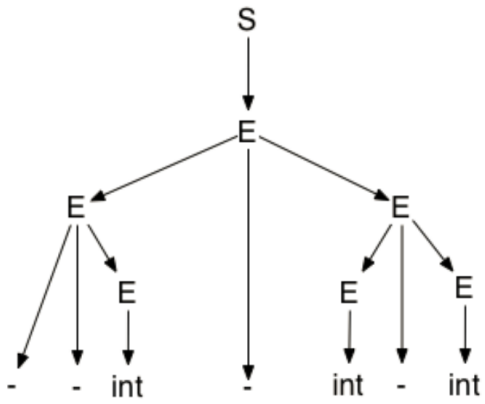
Consider the following grammar with terminals − (the negation operator) and `int`.

S → E
E → E - E | - - E | int

(a) Draw all the parse trees for the string `int - - - int - int` **[6 points]**

(b) Is this grammar ambiguous? Why or why not? **[2 points]**

(c) Suppose we want the string - - int - int - int to have only the following parse tree. Describe in English the precedence and associativity rules necessary to ensure this. **[4 points]**



Your answer:

4. Syntax-Directed Translation

Consider the binary numbers over {0, 1}. Give a syntax directed translation (i.e., a context free grammar and associated actions) that assigns an attribute of the root of the parse tree the value of the binary number converted to base 10. For example, for the string 1001 the attribute of the root of the parse tree should be assigned the value 9.

You can use Bison's action notation, but this is not required for full credit. Any attribute notation we can understand is accepted. **[15 points]**