Homework 2
Due September 29, 2015

Submissions are due by the beginning of class on the specified due date. Handwritten or typed solutions are acceptable. If you do write your solutions by hand, be sure to write clearly. If the grader cannot read your answer, they cannot give you the points. Late submissions will be accepted with a 10% penalty for each day they are late.

1. (60 points) Consider the following LL(1) grammar with terminals (, ), +, id, num, and starting symbol S.

\[
S \rightarrow F \\
S \rightarrow ( S T ) \\
T \rightarrow \epsilon \\
T \rightarrow + F T \\
F \rightarrow id \\
F \rightarrow num
\]

a. Compute the first and follow set of all non-terminals.
b. Construct the LL(1) parse table for the grammar.
c. Show step-by-step (content of stack and input string, as well as the production taken) how the following program is parsed: (( 20 + 30 + a )).
d. Show the parse tree of the program in part c.

2. (15 points) Write a grammar (not necessarily LL(1)) that recognizes various forms of variables in C++, which should include the simple variable (e.g. a, b, c), array variables (e.g. a[10], a[10][10], a[10][10][10]), class members (e.g. a.data, a.data.name, ...), pointer variables (e.g. a->data, a->data->name), and composite variables that are combinations of the above forms (e.g. a->bb[100][200][300].data->mdata.bb[100]). You can assume the array index can only be a number.
3. (15 points) Write a grammar (not necessarily LL(1)) that recognizes a list of variable definition lines. Each line contains a type (int or char) followed by a list of simple variables (identifiers) separated by comma (,). Each line ends with a semicolon (;). For example:

```plaintext
int a, b, d;
char aa;
int ccc, ddd;
```

4. (10 points) What is the language recognized by each of the following grammars? You can either give the set representation of the language or describe the set in English. The starting symbol is $S$.

a. 
$S \rightarrow A \ a \ | \ b$
$A \rightarrow A \ c \ | \ \epsilon$

b. 
$S \rightarrow S \ S \ | \ (S) \ | \ \epsilon$