CSC 254
Assignment 1
September 9th, 2003
Due in class on Tuesday, September 16th, 2003

1. (15 points) Describe the languages denoted by the following regular expressions (where the alphabet consists of 0 and 1):
   (a) $0(0|1)^*0$
   (b) $((\epsilon|0)1^*)^*$
   (c) $(0|1)^*0(0|1)(0|1)$
2. (10 points) Write regular expressions for the following languages
   (a) All strings of 0's and 1's that do not contain the (consecutive) substring 011.
   (b) All strings of 0's and 1's that do not contain the (non-consecutive) subsequence 011.
3. (20 points) For the regular expression (a|b)*
   (a) Construct an NFA using Thompson's construction algorithm
   (b) Show the sequence of moves in parsing "ababbab"
   (c) Convert the NFA to a DFA using subset construction algorithm
   (d) Minimize the DFA using the partitioning algorithm
4. (15 points) Write a grammar for the following languages. Are the grammars ambiguous?:
   (a) All strings of 0's and 1's that have the same number of 0's and 1's.
   (b) All strings of 0's and 1's that have more 0's than 1's.
   (c) All balanced pairs of left and right parentheses (e.g., "()", "()(())").
5. (20 points) Consider the grammar
   S ::= aSbS | bSaS | epsilon
   (a) Show that the grammar is ambiguous by constructing two leftmost derivations for "abab"
   (b) Show that the grammar is ambiguous by constructing two rightmost derivations for "abab"
6. (20 points) Consider the grammar
   S ::= (L) | a
   L ::= L , S | S

(a) Eliminate left recursion from the grammar. Is it LL(1)? If not, how would you make it LL(1)?

(b) Build a non-backtracking recursive descent parser, given the following code:

```
tok;    // current token

match(x) {            // matches token
  if (tok != x)       // if wrong token
     error();         // exit with error
  tok = getToken();   // get new token
}

parser() {
  tok = getToken();   // initialize
  S( );               // start symbol
  match("$");         // match EOF
}
```

(c) Show an example parse of the string "(a,a)"