CIS (4|5)61
Spring 2013 Midterm 1

1. [10 points]

Consider the following regular expression patterns for a scanner generator like lex, flex, or jflex, in which the priority of patterns is indicated by the order of their appearance.

```
end         { return END; }
else        { return ELSE; }
[a-z]+      { return IDENT; }
```

Part 1 (3 points): Draw an NFA corresponding to these patterns (showing which action goes with each accepting state).

Part 2 (7 points): Draw a DFA corresponding to the the NFA of part 1, again being sure to distinguish accepting states.

(score)

2. [10 points]

The following regular expression is designed to match integers with commas separating groups of three digits. For example, it matches 99 and 99,999 but it doesn't match 9,99,9 or 999,9. It is written in a format similar to the input of a scanner generator like flex or jflex, with '?' for optional elements and '( )' for grouping, and for for simplicity I will make 9 the only legal digit.

```
9?9?9(,999)*      { return INT;  }
```

Part 1 (3 points): Draw an NFA corresponding to these patterns. Use a double circle to indicate accepting state(s).

Part 2 (7 points): Draw a DFA corresponding to the the NFA of part 1, again being sure to distinguish accepting states.

(score)

3. [15 points]

Consider the following grammar for variable and function declarations. **P** (program), **D** (declaration), and **L** (list of variables) are non-terminal symbols. *i* (identifier) and the punctuation symbols ‘(’, ‘)’, ‘:’, and ‘,’ are terminal symbols.

$\mathbf{P} \to \mathbf{D}\ \ \$$

$\mathbf{D} \to i\ \ (\ )\ \ :\ \ i$

$\mathbf{D} \to \mathbf{L}\ \ :\ \ i$

$\mathbf{L} \to \mathbf{L}\ \ ,\ \ i$

$\mathbf{L} \to i$

Show that the grammar is not LR(0), but it is LALR(1). (You can omit boring parts of the CFSM; build just enough to show that LALR(1) lookahead resolves the LR(0) conflict.)