CS412 Homework 1 — Due: 3 February 2006

This assignment contains 5 problems. Write your answers *neatly*. Make sure you clearly indicate the start and final states in your automata. Turn in your assignment in class on the due date.

1. Write a regular expression for currency, in dollars, represented as a positive decimal number rounded to the nearest one-hundredth. Such numbers begin with the character `$`, have commas separating each group of three digits to the left of the decimal point, and end with two digits to the right of the decimal point, for example `$8,987.40` or `$77,888,999.11`.

2. This question is about regular expressions that describe HTML comments.

   (a) Consider the following definition for HTML comments. An HTML comment starts with "`<!`", followed by zero or more sections optionally separated by whitespace characters, and followed by "`>`". Each section starts and ends with "`--`", does not contain any occurrence of "`--`". For instance, the following are valid comments:

   "`<!-- aha -->`"

   "`<!-- aha-aha -->`"

   "`<!-- aha -- -- aha -->`"

   Draw the DFA which accepts HTML comments and then use it to write the regular expression that describes HTML comments.

   (b) Write a comment that contains the sequence "`<!-- -->`" inside the comment, as a subsequence between the outermost "`<!`" and "`>`".

   (c) To avoid dealing with confusing (but valid) comments, we propose a simpler definition that supports only "clean" comments: an HTML comment begins with "`<!--`", ends with "`-->`" and does not contain "`--`" or "`>`" anywhere in the comment. Write a regular expression that describes comments according to this definition.

3. Write regular expressions for the following languages:

   (a) Strings of a's, b's, and c's where the first a (if any) precedes the first b;

   (b) All strings of `0`'s and `1`'s that do not contain the string `011`.

4. Consider the following regular expression: $(a|b)*\,abb\,(a|b)*$.

   (a) Use Thompson's construction to construct an NFA for this expression.

   (b) Convert the NFA to a DFA.

   (c) Minimize the DFA.

5. Consider the following lexical analysis specification:

```
a(ba)+  { return Tok1; }
a(b)*a  { return Tok2; }
a|b     { return Tok3; }
```

In case of tokens with the same length, the token whose pattern occurs first in the above list is returned.

(a) Show the NFA which accepts strings matching one of the above three patterns.

(b) Transform the above NFA to a DFA. Label the DFA states with the set of NFA states to which they correspond. Indicate the final states in the DFA and label each of these states with the (unique) token being returned in that state.

(c) Show the steps in the functioning of the lexer for the following input string:

```
abaabbabaa
```

Indicate what tokens does the lexer return for successive calls to `getToken()`. For each of these calls indicate the DFA states being traversed in the automaton.