CS412 Homework 1 *Due: 4 February 2005*

Turn in your assignment in class on the due date.

1. Write a regular expression for http and ftp URLs. An URL consists of four parts: the protocol (`http://` or `ftp://`), the DNS name or the IP address of a host, an optional port number, and an optional pathname for a file. For simplicity, we assume that:
   - A DNS name is a list of non-empty alphabetical strings separated by periods.
   - An IP address consists of four non-negative integers of at most three digits each, separated by periods.
   - A port number is a positive integer following a colon. (e.g. `:8080`)
   - The pathname part is a unix-style absolute pathname. The allowed symbols are letters, digits, period and slash. A sequence of two consecutive slashes `//` is forbidden, i.e. no empty directory name.
   - A URL may end with a slash as long as it does not create the sequence `//`.

2. A comment in the C language begins with the two-character sequence `/*`, followed by the body of the comment, and then the sequence `*/`. The body of the comment may not contain the sequence `*/`, although it may contain the sequence `/*`, or the characters `*` and `/`. We use the notation `(`E`)*` for the Kleene closure of E; all other occurences of `*` refer to the character itself, not to the Kleene operator.

   (a) Show that the following regular expression does not correctly describe C comments:

   ```
   /* (/)* ([^*/] | [^*]/ | *[^/])* (*)* */
   ```

   (b) Draw the DFA that accepts C comments and then use it to write the regular expression that correctly describes C comments.

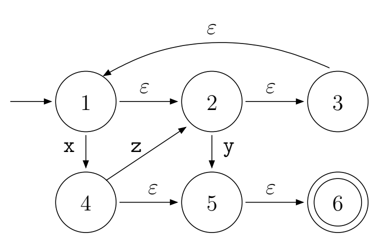
   (c) Write the DFA and the regular expression for C comments extended such that the sequence `*/` is permitted, as long as it appears inside quotes. For example, `/* ab "cd */" ef */` would be a valid comment.

3. Write the DFAs for each of the following:

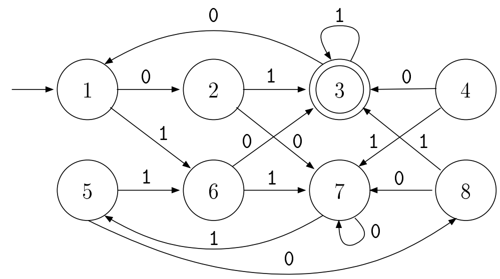
   (a) Binary numbers that contain the substring 011.

   (b) Binary numbers that are multiples of 3 and have no consecutive 1's.

4. Convert the following NFA to a DFA. For each DFA state, indicate the set of NFA states to which it corresponds. Make sure you show the initial state and the final states in the constructed DFA.



5. Minimize the states of the following DFA. Label each state in the minimized DFA with the set of states from the original DFA to which it corresponds.



6. Consider the following lexical analysis specification:

```
(aba)+    { 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 that 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 input string `abaabbaba`. 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.