CS 273: Intro to Theory of Computation, Spring 2008

Problem Set 2 (due Monday, January 28th, 4pm)

This homework contains five problems, one of which is bonus. Please submit each on a **separate sheet of paper**. This will help us grade your homeworks more quickly. Turn in your homework at Elaine Wilson's office (3229 Siebel).

1. Building a DFA

   Let $\Sigma = \{0, 1\}$. Give DFA state diagrams for the following languages.

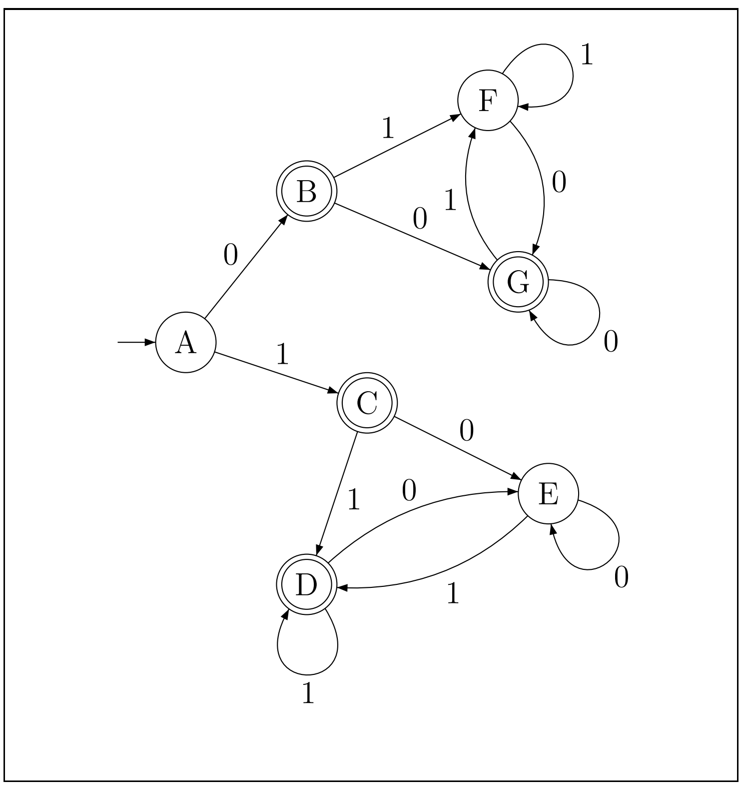
   (a) $L = \{w \mid w \text{ contains the substring } 001\}$

   (b) $L = \{a \mid \text{the length of } a \text{ is not divisible by 2 and not divisible 3}\}$.

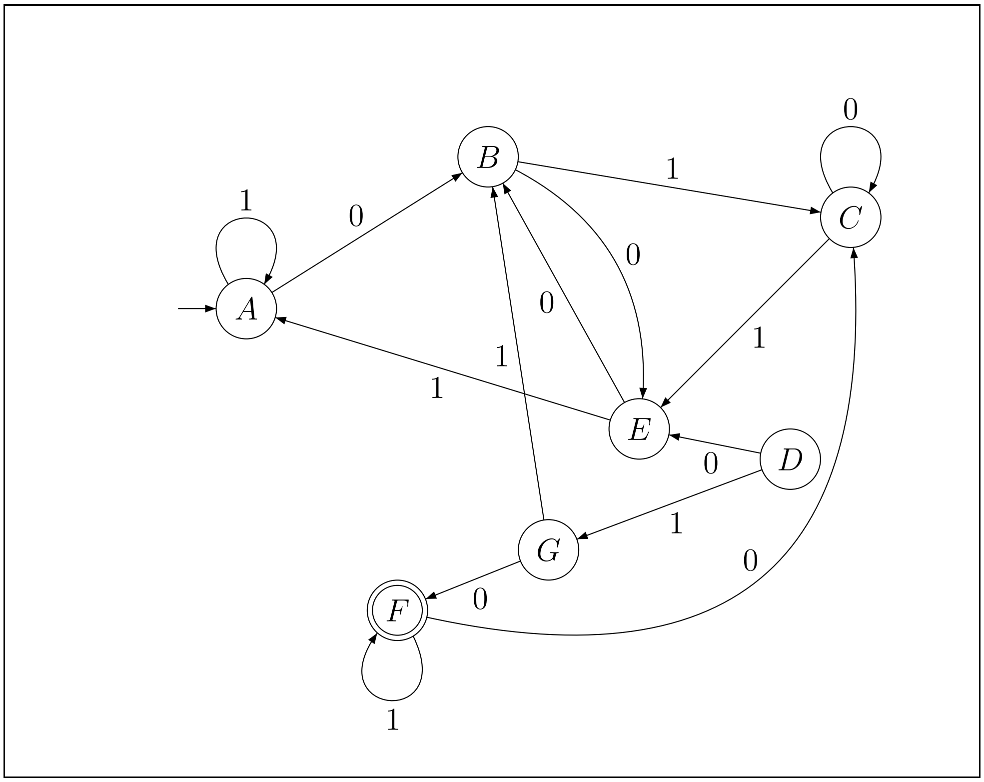
   (c) $L = \Sigma^* - \{\epsilon\}$

2. Interpreting a DFA

   (a) What is the language of the following DFA? That is, explicitly specify <u>all</u> the strings that the following DFA accepts. Briefly explain why your answer is correct.



(b) What about this one? Again, briefly justify your answer.



3. Sets and trees

   Define a "set-based binary tree" (SBBT) as follows:

   - Every positive integer is an SBBT.
   - If x and y are SBBTs, $x \neq y$, then $\{x, y\}$ is also an SBBT.

   For example, $R = \{\{3, 2\}, 4\}$ and $P = \{3, \{\{7, 9\}, \{8, 3\}\}$ are SBBTs. But $\{2, \{4, 5\}, 27\}$ and $\{2, \{3\}\}$ are not SBBTs. Let T be the set of all SBBTs.

   (a) Let's define the following function f mapping SBBTs to sets of integers:

$$f(X) = \begin{cases} \bigcup_{Y \in X} f(Y) & \text{if } X \text{ is a set} \\ \{X\} & \text{if } X \text{ is an integer} \end{cases}$$

   Notice that $f(Y)$ is always a set, for any input Y. The operation $\bigcup_{Y \in X} f(Y)$ unions together the sets $f(Y)$, for all the items Y that are in the set X.

   For the SBBTs P and R defined above, compute $f(P)$ and $f(R)$. Give a general description of what f does.

   (b) Similarly, we can define a function g mapping SBBTs to integers:

$$g(X) = \begin{cases} \sum_{Y \in X} g(Y) & \text{if } X \text{ is a set} \\ 1 & \text{if } X \text{ is an integer} \end{cases}$$

Give the values for $g(P)$ and $g(R)$, as well as a general description of what g does.

(c) For certain SBBTs, $g(X) = |f(X)|$. For which SBBTs does this equation work? Explain why it's not true in general.

4. Balanced strings

A string over $\{0, 1\}$ is *balanced*, if it has the same number of zeros and ones.

(a) Provide a pseudo-code (as simple as possible) for a program that decides if the input is balanced. You can use only a single integer typed variable in your program, and one variable containing the current character (of type `char`). You can read the next character using a library function called, say, `get_char`, which returns -1, if it reached the end of the input. Otherwise, `get_char` returns the next character in the input stream.

In particular, the program prints "accept" if the input is a balanced string, and print "reject", otherwise.

(b) For any fixed prespecified value of k, describe how to construct an automata that accepts a balanced string if, in every prefix of the input string, the absolute difference in the number of zeros and ones does not exceed k. How many states does your automata needs?

(c) Provide an intuitive explanation of why the number of states in automata for the problem of part (B), must have at least, say, k states.

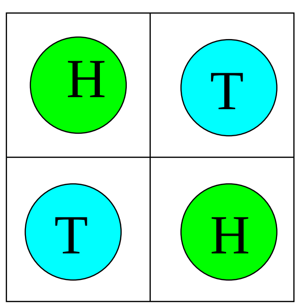
(d) Argue, that there is no *finite* automata that accepts only balanced strings.

For bonus credit, you can provide formal proofs for the claims above.

5. Bonus problem (Coins)

A journalist, named Jane Austen, unfortunately (for her) interviews one of the presidential candidates. The candidate refuses to let Jane end the interview going on and on about the candidate plans how to solve all the problems in the world. In the end, the candidate offers Jane a game. If she wins the game she can leave.

The game board is made out of 2×2 coins:



At each round, Jane can decide to flip one or two coins, by specifying which coins she is flipping (for example, flip the left bottom coin and the right top coin), next the candidate goes and rotates the board by either $90, 180, 270$, or 0 degrees. (Of course, rotation by 0 degrees is just keeping the coins in their current configuration.)

The game is over when all the four coins are either all heads or all tails. To make things interesting, Jane does not see the board, and does not know the starting configuration.

Describe an algorithm that Jane can deploy, so that she always win. How many rounds are required by your algorithm?