CS 273: Intro to Theory of Computation, Fall 2007
Problem Set 5 (due Tuesday, October 9th)

This homework contains 5 problems. It is due 12:30 in class or by noon at Elaine Wilson's office (3229 Siebel).

1. **Context-free grammars**

   (a) Here is a context-free grammar G, in which S is the start symbol and the variables are $\{S, A, B\}$. What is the language defined by G?

   $$S \to aS \mid AaA \mid B$$
   $$A \to bA \mid cA \mid b \mid c$$
   $$B \to bB \mid b$$

   (b) Here is another grammar G with start symbol S and variables $\{S, A, B\}$. What is its language?

   $$S \to aS \mid A \mid B$$
   $$A \to bA \mid \epsilon$$
   $$B \to bBc$$

   (c) Let $\Sigma = \{a, b, c, d\}$. Suppose L is the language

   $$L = \{w_1x_1w_2x_2 \ldots x_iw_{i+1} \mid \text{ each } x_i \in \{c, d\}^*, \text{ each } w_i \text{ has the form } a^nb^n \text{ for some } i \in \mathbb{N}, \text{ and } i \geq 0\}$$

   Give a context-free grammar whose language is L.

2. **Proving non-regularity using closure properties**

   Use the closure properties of regular languages to prove that the following language is not regular.

   $$L = \{b^mc^nd^{n-5} \mid m \geq 0, n \geq 5\}$$

   This is a proof by contradiction and you are allowed to use any language that that was proved to be nonregular in class or in the textbook.

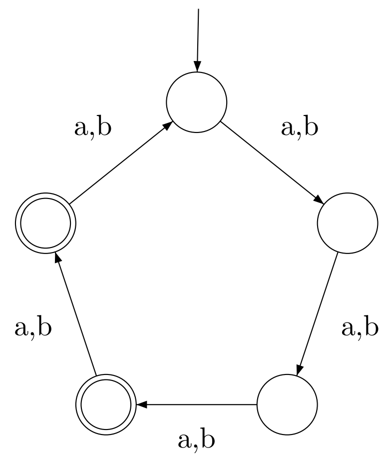
3. **Pumping Lemma**

(a) Let Σ be an alphabet with $|\Sigma| \geq 2$. $w \in \Sigma^*$ is a palindrome if $w = w^R$. A palindrome is *symmetric* if $w = xy$ for some $x, y \in \Sigma^*$ such that $|x| = |y|$, and $w = yx$. Let

$$P = \{w \in \Sigma^* \mid w \text{ is a symmetric palindrome}\}.$$

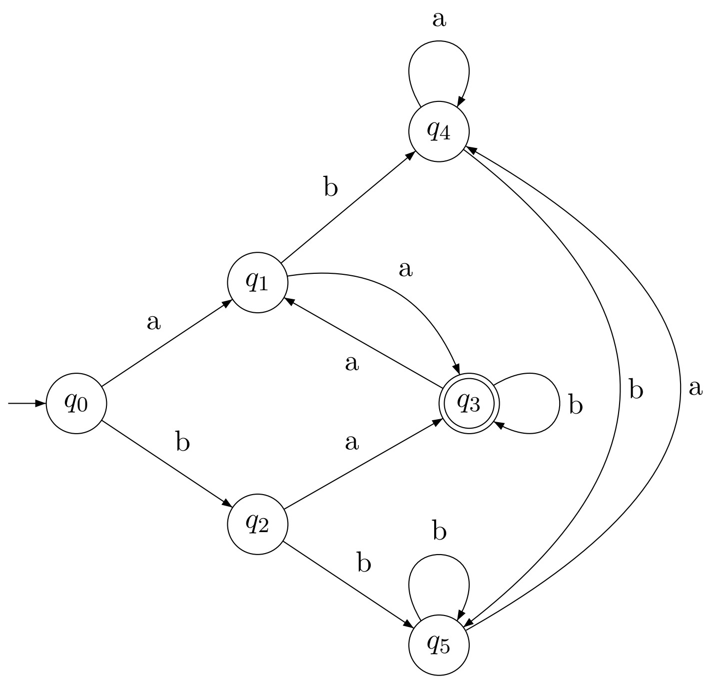
Use the pumping lemma to show that P is not regular.

(b) Let N be the automaton pictured below. Find the minimum pumping length p for $L(N)$. The "minimum pumping length" is the smallest value that can be chosen for p in applying the pumping lemma to this language. (See Sipser problem 1.55.) Briefly explain why your answer is correct.



4. **Suffix languages**

Consider the following DFA over the alphabet $\Sigma = \{a, b\}$:



(a) Write down the language L_q for each state q.

(b) Merge the states that accept the same languages, and hence construct a DFA B that accepts the same language but with fewer states.

(c) (5-point bonus) Suppose that B contains k states. Prove that *any* minimal DFA for this language requires k states, by exhibiting a set of k strings and explaining why any DFA for this language must send them to different states.

5. **NFA pattern matching**

Pattern-search programs take two inputs: a pattern given by the user and a file of text. The program determines whether the text file contains a match to the pattern, typically using some variation on NFA/DFA technology. Fully developed programs, such as grep, accept patterns containing regular-expression operators (e.g. union) and also other convenient shorthands. Our patterns will be much simpler.

Let's fix an alphabet $\Sigma = \{a, b, ...z, 0, 1, ...9\}$. Let $\Gamma = \Sigma \cup \{?, \#, +\}$. A **pattern** will be any string in Γ^*. A string w matches a pattern p if you can line up the characters in the two strings such that:

- When p contains a character from Σ, it must be paired with an identical character in w.
- The character ? in p can match any substring x in w, where x contains at least one character.
- The character # in p can match any numeric digit $0, \ldots, 9$.
- The character + in p can match exactly one character.

For example, the pattern "fleck" matches only the string "fleck". The pattern $p = 2\#3$ will match strings such as "283", "203", but not "173" and not "2893". The pattern $p = 273?fleck?$ will match a string w which starts with "273" immediately followed by some text of length at least one, followed by "fleck", followed by at least one more character (e.g. "273ab7sfleck8"). The pattern $p = cs + 27+$ will match a string w which starts with "cs" immediately followed 1 character, then "27", then at least one more character (e.g., "csa273", "cs8275").

A text file t contains a match to a pattern p if t contains some substring w such that w matches p.

Design an algorithm which converts a pattern p to an NFA N_p that searches for matches to p. That is, the NFA N_p will read an input text file t and accept t if and only if t contains a match to p. N_p searches for only one fixed pattern p. However you must describe a general method of constructing N_p from any input pattern p.