COMPSCI 250: Fall 2023
Homework 6

David A. Mix Barrington, Ghazaleh Parvini

Due Date: Friday, December 8;
Late Day: Saturday, December 9

This assignment has eight problems. There is also one Extra Credit problem. The extra credit is 10 points.

Please submit a single PDF file, with the problems in order (as below), and legible. Look at your PDF before submitting it – it is fine to scan or photograph a handwritten document but it the graders can't read it, they won't grade it.

Please assign pages to problems in Gradescope. Graders will click on the problem number. If no page shows up because it's not assigned, the assumption is you have not solved the problem.

Be sure you are doing Problems in the book and not Exercises: the numbers should start with P rather than E.

For full credit, show your work, explaining your reasoning. This also helps assign partial credit.

You are responsible for following the academic honesty guidelines on the Grading and Requirements page. This means that what you present must be your own work in presentation, and you must acknowledge all sources of aid other than course staff and the textbook. You will get 2 extra points if you typeset your Homework.

(8 points) **Problem 14.1.6**

Prove that every language with a finite number of strings is the language of some DFA.

(12 points) **Problem 14.2.4**

A string in $\{a, \cdots, z\}^*$ is said to be **panalphabetic** if it contains at least one occurrence of each letter. Examples of panalphabetic strings are

$$thequickbrownfoxjumpsoverthelazydog$$

and

$$jackdawslovemybigsphinxofquartz.$$

Is the language of panalphabetic strings decidable by a DFA? Prove your answer.

(12 points) **Problem 14.3.6**

Prove that the DFA from Exercise 14.3.6 is minimal, either by running the minimization algorithm on it or by showing that each pair of states is L_5-distinguishable.

(15 points) **Problem 14.5.4**

Let N be an ordinary NFA with k states. For every letter $a \in \Sigma$, define a k by k boolean matrix M_a such that $M_a(i, j)$ is true if and only if $<i, a, j> \in \Delta$. Define a function f from Σ^* to the set of k by k boolean matrices by the rules $f(\lambda) = I$, $f(wa) = f(w)M_a$. Prove that $w \in L(N)$ if and only if there is a final state f such that the (ι, f) entry of $f(w)$ is true.

(16 points) **Problem 14.6.9**

Given a nonempty finite language S, consider the language $CS = \Sigma^* S \Sigma^*$ of strings that have a substring in S.

(a) Describe an ordinary NFA for this language. (Note: The Yes-aba language is just $C_{\{}aba\}$.)

(b) Prove that the minimal DFA for any such language C_S has exactly one final state.

(c) Use the Subset Construction to find a DFA for the language C_S where $S = \{aaa, bbb\}$. (You may invoke part (b) and not worry about distinguishing the different final states.)

(d) Repeat part (c) for $S = \{abb, bab, bba\}$. Let $\Sigma = \{a, b\}$.

(15 points) **Problem 14.7.6**

Let N be a λ-NFA in which there are two states p and q with transitions $<p, \lambda, q>$ and $<q, \lambda, p>$. Show that there is a λ-NFA N', with $L(N') = L(N)$, such that N' has a single state r instead of p and q. Should r be a final state? What if p or q is the start state? What should be the transitions in and out of r? Should any other transitions from N change? Prove that with your choices, $L(N) = L(N')$.

(12 points) **Problem 14.8.7**

In Section 5.5 we gave a recursive algorithm that decides whether the language of a given regular expression is empty.

(a) Prove that if N is a $\lambda - NFA$ obeying our three rules, $L(N) = \emptyset$ if and only if there is no path from the start state to the final state in N.

(b) Prove, by induction on all regular expressions α, that the emptiness tester of Section 5.5 returns false on input α if and only if the $\lambda - NFA$ constructed from α has a path from the start state to the final state.

(10 + 5 extra) **Problem 14.10.3**

(variant from the textbook)

Find a regular expression for the language of strings in $\{a, b\}^*$ whose number of a's and number of b's are **congruent** modulo 3. (**Hint:** Build a DFA and use state elimination. This gets a bit messy. The problem in the textbook has the number of a's and the number of b *both* divisible by 3. We'll give you five extra points for solving the original version.)

Extra Credit

(10 points) **Problem 15.6.2**

Build a Turing machine M_{copy} that when started on input w (i.e., in configuration $i\square w$), halts with the tape contents $i\square w\square w$. (Here i is the start state.) You need not give the entire state table as long as your description is clear and complete. Let $\Sigma = \{a, b\}$.