University of Virginia - cs3102: Theory of Computation	Spring 2010
PS2	**Due: 16 February 2010 (2:00pm)**

This problem set covers material in Sipser Chapter 1, focusing on Sections 1.2 and 1.3, and sections 2.1 and 2.2 of Chapter 2. Answer all ten questions (5f is an optional challenge question). For full credit, answers must be concise, clear, and convincing, not just correct.

You are strongly encouraged to use LaTeX to produce your submission, and we have provided a LaTeX template to assist with this. You may, however, handwrite your answers *so long as your writing is legible and easily interpreted.* For full credit, answers must be concise, clear, and convincing, not just correct.

Collaboration Policy. For this assignment, we will follow the same collaboration and resource policy as on Problem Set 1. See the Problem Set 1 handout for a full description of the policy.

Pledging. It is not necessary to scribble a pledge on your submission: we assume all students are honest and honorable regardless of whether or not you scribble a pledge on your submission. If you are dishonorable, however, please write a note on your submission indicating this. If you want to write a pledge, please write something meaningful like, "with honor", not something rote and meaningless like "pledge".

Problem 1: Constructing NFAs. For each of the following languages, draw an NFA that recognizes the language using *fewest possible number of states.* For all languages, assume $\Sigma = \{0, 1\}$.

a. $\{w | w$ starts and ends with different symbols. $\}$

b. $\{w | w$ does not contain two consequtive 0s. $\}$

c. The language described by the regular expression $\{0, 1\}^* \, 01^*$.

Problem 2: Nondeterminstic Finite Automata. Answer these questions for the NFA N_1 in Sipser's Example 1.38. The extended transition function, δ^*, is as defined in the regular languages notes.

a. What is the set of possible states of N_1 after processing input `1`?

b. What is the set of possible states of N_1 after processing input `010`?

c. Is there a string w such that $\delta^*(q_1, w)$ does not include q_1? (If so, give the string w. If not, argue why no such w exists.)

d. Is there a string w such that $\delta^*(q_1, w) = \{q_1, q_2, q_3, q_4\}$? (If so, give the string w. If not, argue why no such w exists.)

Problem 3: Eliminating Epsilon. Prove that for every NFA N, there is an NFA N' that recognizes the same language as N but does not use any ϵ-transitions. (A very short, very convincing proof is possible, but you will receive some credit for a longer proof.)

Problem 4: Priming the Pump. Use the pumping lemma to prove the language, $PRIMES$, is non-regular:

$$PRIMES = \{1^n | n \text{ is a prime number}\}$$

Problem 5: Regularity. For each language below, answer whether or not the language is regular. Include a convincing proof supporting your answer. You may use any proof technique you want, including construction, the pumping lemma, and the closure properties we have established for regular languages in the book, class, and other problems.

a. $\{a^n b^m | n \geq 0, 0 \geq m \leq 27\}$

b. $\{a^n b^m | m = 2n\}$

c. $\{w | w \text{ is a valid email address}\}$ (for a summary of the specification of a valid email address, see http://en.wikipedia.org/wiki/E-mail_address#RFC_specification)

d. $\{w | w \text{ describes a chess position where White has a winning strategy}\}$

> We describe a chess position with a sequence of 64 symbols from the alphabet $\Sigma = \{-, p, n, b, r, q, k, P, N, B, R, Q, K\}$ where the lower-case letters represent a white pieces and the upper-case letters represent the corresponding black pieces. White has a winning strategy for a given chess position if there is a way for white to win the game no matter what the other player does. (Note: you do not need to know anything more about chess to answer this.)

e. $\{w | \text{there is no string } x \text{ such that } w = xx\}$

f. *Bonus Challenge:*
$\{w | w$ is a valid day in the Gregorian calendar in the form *month day, year*$\}$

> The Gregorian leap year rule is: every year that is divisible by four is a leap year, except for years that are exactly divisible by 100 but not divisible by 400. For example, "February 7, 2010", "February 29, 2012", and "February 29, 2400" are in the language, but "February 29, 2010" and "February 29, 2100" are not.

Problem 6: Language Splitting. (Based on Sipser's 1.63a) Prove that any infinite regular language (that is, a regular language with an infinite number of strings) can be split into three infinite disjoint regular subsets.

Problem 7: Language Operators. Define a new operation on languages, $\mathcal{D}$, as:

$$\mathcal{D}(L) = \left\{ w | w \in \Sigma^* \text{ and } ww^R \in L \right\}$$

(where w^R denotes the reverse of w). Does $\mathcal{D}$ preserve regularity?

Problem 8: Nondeterminstic Pushdown Automata. Draw a nondeterministic pushdown automaton that recognizes the language $\left\{ w0w^R | w \in \{0,1\}^* \right\}$. The fewer states you use, the better.

Problem 9: Regular Grammars. A *regular grammar* is a replacement grammar in which all rules have the form $A \rightarrow aB$ or $A \rightarrow a$ where A and B represent any variable and a represents a terminal. Prove that all regular languages can be recognized by a regular grammar.

Problem 10: Context-Free Grammars. Consider the grammar G below (S is the start symbol, and 0 and 1 are terminals):

$$\begin{array}{lll} S & \rightarrow & \epsilon \\ S & \rightarrow & S\mathbf{00} \\ S & \rightarrow & \mathbf{11}S \\ S & \rightarrow & \mathbf{0}S\mathbf{1} \\ S & \rightarrow & \mathbf{10}S \end{array}$$

a. Show that the string 111000 can be produced by G by showing a derivation that produces it.

b. How many different derivations are there in G to produce 111000? (Support your answer with a clear argument.)

c. What is the fewest number of rules that can be added to G to produce a grammar that describes the language of all even-length strings in $\{0,1\}^*$? (An excellent answer would include the rules to add, a proof why it is not possible it use fewer rules, and a proof that your modified grammar produces all even-length strings.)

End of Problem Set 2.

Turn in your *stapled* submission at the beginning of class on Tuesday, 16 February 2010.