CS 271: Automata and Computation Theory **Spring 2015**

Problem Set #1

Due: Sunday, February 15, 2015

To receive full credit for each construction you give, you must justify why your construction is correct unless the problem explicitly says otherwise.

Problem 1 Prove that the class of regular languages is closed under reversal. That is, show that given a regular language L, show that $L^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in L\}$ is regular. [*Hint: Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes L, build an NFA $N = (Q', \Sigma, \delta', q_0', F')$ that recognizes $L^{\mathcal{R}}$.*]

Problem 2 Define

$$\textsc{BackwardsAndForwards}(L) = \{w \in L \mid w \in L \text{ and } w^{\mathcal{R}} \in L\}.$$

That is, given a language L, $\textsc{BackwardsAndForwards}(L)$ is a new language consisting of the elements of L whose reversal is also an element of L. Using closure properties of regular languages, show that the class of regular languages is closed under the operation $\textsc{BackwardsAndForwards}$.

Problem 3

a. Use closure properties of regular languages to show that regular languages are closed under set difference. That is, given regular languages L_1 and L_2, show that

$$L_1 \setminus L_2 = \{w \in L_1 \mid w \notin L_2\}$$

is regular.

b. Show that regular languages are closed under symmetric set difference

$$L_1 \triangle L_2 = \{w \mid \text{either } w \in L_1 \text{ or } w \in L_2 \text{ but not both}\}.$$

Problem 4

a. For any language L, define

$$\textsc{Prefix}(L) = \{w \mid \exists x \in \Sigma^* \text{ s.t. } wx \in L\}.$$

Prove that regular languages are closed under $\textsc{Prefix}$.

b. For any language L, define

$$\textsc{Suffix}(L) = \{w \mid \exists x \in \Sigma^* \text{ s.t. } xw \in L\}.$$

Using closure properties of regular languages and the result of part **a**, prove that regular languages are closed under $\textsc{Suffix}$.

Problem 5 For languages L_1 and L_2, define

$$L_1 \obackslash L_2 = \{w \in L_1 \mid w \text{ does not contain any string in } L_2 \text{ as a substring}\}.$$

Prove that regular languages are closed under $\obackslash$.[1] [*Hint: Think about what* $\Sigma^* \circ L \circ \Sigma^*$ *means for a language* L. *Write* $L_1 \obackslash L_2$ *in terms of set difference and concatenation and apply closure properties of regular languages.*]

Problem 6 Let Σ and Γ be alphabets and let $f : \Sigma \to \Gamma$ be a function that maps symbols in Σ to symbols in Γ. One such example is $f : \{1, 2, 3, 4, 5\} \to \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ given by

$$\begin{aligned} f(1) &= \mathtt{b} \\ f(2) &= \mathtt{b} \\ f(3) &= \mathtt{a} \\ f(4) &= \mathtt{d} \\ f(5) &= \mathtt{a}. \end{aligned}$$

We can extend such an f to operate on strings $w = w_1 w_2 \cdots w_n$ by

$$f(w) = f(w_1) f(w_2) \cdots f(w_n).$$

Using the same example, $f(132254) = \mathtt{babbad}$. We can extend f to operate on languages by $f(L) = \{f(w) \mid w \in L\}$.

Prove that if L is a regular language and $f : \Sigma \to \Gamma$ is an arbitrary function — that is, it is *not* necessarily the example given above — then $f(L)$ is regular. [*Hint: given a DFA* M *that recognizes* L, *build an NFA* N *that recognizes* $f(L)$ *by applying* f *to the symbols on each transitions. To prove that this works, consider the states* M *goes through on input* w *and the states* N *goes through on input* $f(w)$.]

Problem 7 A homomorphism is a function $f : \Sigma \to \Gamma^*$ that maps symbols in Σ to *strings* over Γ. One example of a homomorphism is the function that maps every string to ε. A less-trivial example is $f : \{\mathtt{a}, \mathtt{b}\} \to \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ given by

$$\begin{aligned} f(\mathtt{a}) &= \mathtt{bacca} \\ f(\mathtt{b}) &= \mathtt{b}. \end{aligned}$$

[1] You can typeset $\obackslash$ in LaTeX by putting the line `\usepackage{mathabx}` in the preamble and using `\obackslash` in math mode.

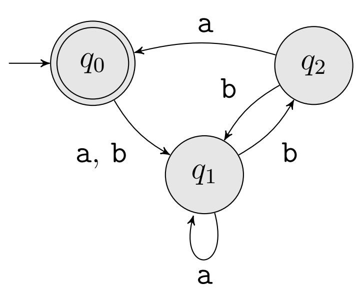
We can extend f to operate on strings $w = w_1w_2\cdots w_n$ by $f(w) = f(w_1)f(w_2)\cdots f(w_n)$ and languages by $f(L) = \{f(w) \mid w \in L\}$.

Prove that regular languages are closed under homomorphism. [*Hint: As with your construction in Problem 6, you want to apply f to the symbols on each transition but in this case you may need to add additional states if the length of $f(a)$ is not 1. Be sure to handle the case where $f(a) = \varepsilon$.*]

Problem 8 For each language in Exercise 1.6 in Sipser, give an equivalent regular expression. (You don't need to prove that it's correct.)

Problem 9 Using the procedure given in Lemma 1.55 in Sipser, convert the regular expression $(0 \cup 11)^*01(00 \cup 1)^*$ to an NFA. Show each step.

Problem 10 Using the procedure given in Lemma 1.60 in Sipser, convert the following DFA to a regular expression. Show each step.



Extra Credit This extra credit problem is worth as much as 1.5 other problems but is more difficult. Define the *chop* of languages A and B as

$$\textsc{Chop}(A, B) = \{w \mid \exists x \in B \text{ s.t. } wx \in A\}.$$

For example, if

$$X = \{\texttt{aab}, \texttt{aba}, \texttt{bba}, \texttt{bbb}\}$$
$$Y = \{\texttt{ba}, \texttt{bbb}\}$$
$$Z = \{\texttt{a}^n\texttt{b}^n \mid n \geqslant 0\}$$

are languages then,

$$\textsc{Chop}(X, Y) = \{\varepsilon, \texttt{a}, \texttt{b}\}$$
$$\textsc{Chop}(X, Z) = \{\texttt{a}, \texttt{aab}, \texttt{aba}, \texttt{bba}, \texttt{bbb}\}$$
$$\textsc{Chop}(Z, X) = \{\texttt{a}^n\texttt{b}^{n-3} \mid n \geqslant 3\}.$$

If you compare the definition of Chop to the definiton of Prefix, you can see that they are similar. Indeed $\textsc{Prefix}(L) = \textsc{Chop}(L, \Sigma^*)$.

Prove that if A is a regular language and B is *any* language (in particular, it need not be regular), then $\textsc{Chop}(A, B)$ is regular. [*Hint: Try modifying your proof that regular languages are closed under* Prefix.]