CS 301: Languages and Automata **Fall 2015**

Problem Set #1

Due: Tuesday, September 22, 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 A, show that $A^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in A\}$ is regular. [*Hint: Given a DFA* $M = (Q, \Sigma, \delta, q_0, F)$ *that recognizes* A*, build an NFA* $N = (Q', \Sigma, \delta', q_0', F')$ *that recognizes* $A^{\mathcal{R}}$*.*]

Problem 2 Define

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

That is, given a language A, BackwardsAndForwards(A) is a new language consisting of the elements of A whose reversal is also an element of A. Using closure properties of regular languages, show that the class of regular languages is closed under the operation 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 A and B, show that

$$A \setminus B = \{w \in A \mid w \notin B\}$$

is regular.

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

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

Problem 4 Recall the definitions of Prefix and Suffix

$$\text{Prefix}(A) = \{w \mid \exists x \in \Sigma^* \text{ s.t. } wx \in A\},$$
$$\text{Suffix}(A) = \{w \mid \exists x \in \Sigma^* \text{ s.t. } xw \in A\}.$$

We showed in class that regular languages are closed under Prefix. Using closure properties of regular languages, show that regular languages are closed under Suffix.

Problem 5 For languages A and B, define

$$A \obslash B = \{w \in A \mid w \text{ does not contain any string in } B \text{ as a substring}\}.$$

Prove that regular languages are closed under $\obslash$.[1] [*Hint: Think about what* $\Sigma^* \circ L \circ \Sigma^*$ *means for a language* L. *Write* $A \obslash B$ *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 $f : \{\mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}\} \to \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ given by

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

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

Prove that if A is a regular language and $f : \Sigma \to \Gamma$ is an arbitrary function — that is, it is *not* necessarily the example given above — then $f(A)$ is regular. [*Hint: given a DFA* M *that recognizes* A*, build an NFA* N *that recognizes* $f(A)$ *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

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

We can extend f to operate on strings $w = w_1 w_2 \cdots w_n$ by $f(w) = f(w_1) f(w_2) \cdots f(w_n)$ 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* $f(a) = \varepsilon$.]

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

Problem 8 For each language below, give an equivalent regular expression. (You don't need to prove that it's correct.) In each case, $\Sigma = \{\texttt{0}, \texttt{1}\}$.

$A = \{w \mid w$ beings with a 1 and ends with a 0$\}$
$B = \{w \mid w$ contains at least three 1s$\}$
$C = \{w \mid w$ contains the substring 0101$\}$
$D = \{w \mid w$ has length at least 3 and its third symbol is 0$\}$
$E = \{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$
$F = \{w \mid w$ doesn't contain the substring 110$\}$
$G = \{w \mid$ the length of w is at most 5$\}$
$H = \{w \mid w$ is any string except 11 and 111$\}$
$I = \{w \mid$ every odd position of w is a 1$\}$
$J = \{w \mid w$ contains at least two 0s and at most one 1$\}$
$K = \{\varepsilon, \texttt{0}\}$
$L = \{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$
$M = \emptyset$
$N = \Sigma^* \setminus \{\varepsilon\}$

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.