UIC
DEPARTMENT OF COMPUTER SCIENCE

Lenore Zuck | CS301 | *Handout #1*
TA: Saad Sheikh | Languages and Automata | *February 21, 2007*

Mock Midterm 1

(Closed books, 45 minutes, no group work, etc. Please write clearly. Justify your answers.) Unless otherwise stated (and it never is), assume that $\Sigma = \{0, 1\}$.

Problem 1 For a word $w \in \Sigma^*$, we define the reverse operation, w^R, inductively as follows:

1. If $w = \epsilon$, then $w^R = \epsilon^R = \epsilon$.
2. If $w = w_1 \cdot a$ (for some $a \in \Sigma$), then $w^R = aw_1^R$.

So, that, e.g., $(100)^R = 0(10)^R = 00(1)^R = 00(1\epsilon)^R = 001(\epsilon)^R = 001$.

Prove that for any strings $w_1, w_2 \in \Sigma^*$, $(w_1 w_2)^R = w_2^R \cdot w_1^R$.

Problem 2

1. For each of the following languages, write a regular expression that describes the language and construct a FSA that recognizes it.
   (a) The set of strings that have exactly one occurrence of "1".
   (b) The set of strings that do *not* have the substring 01.
2. Given a regular language L and a DFA that recognizes it with n states. Does there always exist a NFA that recognizes L with less than n states?

Problem 3 Let L be a regular language. Show that the language $\mathit{Pref}(L)$ defined by:

$$\mathit{Pref}(L) \; = \; \{u : \text{for some } w,\, uw \in L\}$$

is regular.

Problem 4 For two strings u and v of the same length, define $\mathit{merge}(u, v)$ to be the string of length $2|v|$ such that the letters in the odd positions consist of u, and the letters in the even positions consist of v. E.g., $\mathit{merge}(011, 110) = 011110$.

Let L_1 and L_2 be languages. Define the language

$$\mathit{merge}(L_1, L_2) \; = \; \{w : w = \mathit{merge}(u, v) \text{ for some } u \in L_1 \text{ and } v \in L_2\}$$

Show that if both L_1 and L_2 are regular, then so is $\mathit{merge}(L_1, L_2)$.