Harvard University
Computer Science 121

Problem Set 2

Due Tuesday, September 24, 2013 at 11:59 PM.
Submit your solutions electronically to cs121+ps2@seas.harvard.edu with "ps2 submission" in the subject line. The solutions to each part should be attached as separate PDF files, called lastname+ps2a.pdf, lastname+ps2b.pdf, and lastname+ps2c.pdf.
Late problem sets may be turned in until Friday, September 27, 2013 at 11:59 PM with a 20% penalty.

Problem set by ** ENTER YOUR NAME HERE **

Collaboration Statement: **FILL IN YOUR COLLABORATION STATEMENT HERE (See the syllabus for information)**

See syllabus for collaboration policy.

PART A (Graded by Nick)

PROBLEM 1 (3+3+3+3+3 points)

Translate the following languages over $\Sigma = \{a, b\}$ from English description to regular expression or vice versa. Limit each description to one line.
(A) $L = \{w \in \Sigma^* :$ the second letter of w is the same as the last letter $\wedge |w| \geq 2\}$
(B) $L = \{w \in \Sigma^* : w = \sigma_0\sigma_1 \cdots \sigma_k,\ k \in \mathbb{N},$ where all characters with even indices are the same$\}$
(C) $(a(a \cup b)^*b) \cup (b(a \cup b)^*a)$
(D) $(b^*abbb^*) \cup (b^*bbab^*) \cup (b^*babb^*) \cup (bbb^*)$

(A) $\Sigma a \Sigma^* a \cup \Sigma b \Sigma^* b \cup \Sigma\Sigma$
(B) $(a\Sigma)^*(a \cup \varepsilon) \cup (b\Sigma)^*(b \cup \varepsilon)$.
(C) $L = \{w \in \Sigma^* :$ the first letter of w is different from the last letter, $|w \geq 2|\}$
(D) $L = \{w \in \Sigma^* :\ w$ has at least two b's and at most one $a\}$.

PROBLEM 2 (15 points)

Using regular expressions, prove that if L is a regular language then the *reversal* of L, $L^R = \{w^R : w \in L\}$, is also regular. In particular, given a regular expression that describes L, show by induction how to convert it into a regular expression that describes L^R. Your proof should not make recourse to NFAs.

We use structural induction. First some notation: given a regular expression R, let R^R be the regular expression, if it exists, for the language $L(R)^R$. Now, given a language L, let R be a regular expression for L. We show that $L(R)^R$ is a regular language described by regular expression R^R.

1. $R = a$ for some $a \in \Sigma$, or $R = \epsilon$, or $R = \emptyset$. In either of these cases, we can quickly show $L(R)^R = L(R)$, so $L(R)^R$ is regular, and $R^R = R$.

2. $R = R_1 \cup R_2$, for some regular expressions R_1, R_2. In this case, $L(R) = L(R_1) \cup L(R_2)$. We know that

$$\begin{aligned} w \in L(R)^R &\iff w^R \in L(R) \\ &\iff w^R \in L(R_1) \text{ OR } w \in L(R_2) \\ &\iff w \in L(R_1)^R \text{ OR } w \in L(R_2)^R \\ &\iff w \in L(R_1)^R \cup L(R_2)^R \ . \end{aligned}$$

So $L(R)^R = L(R_1)^R \cup L(R_2)^R$. But by inductive hypothesis, $L(R_1)^R$ and $L(R_2)^R$ are regular and described by R_1^R and R_2^R, so their union is regular, so $L(R)^R$ is regular, and is given by $R_1^R \cup R_2^R$.

3. $R = R_1 \circ R_2$, for some regular expressions R_1, R_2. In this case, $L(R) = \{xy : x \in L(R_1), y \in L(R_2)\}$.

$$\begin{aligned} w \in L(R)^R &\iff w^R \in L(R) \\ &\iff w^R = xy : x \in L(R_1), y \in L(R_2) \\ &\iff w = y^R x^R : x \in L(R_1), y \in L(R_2) \\ &\iff w = yx : x \in L(R_1)^R, y \in L(R_2)^R \ . \end{aligned}$$

So $L(R)^R = L(R_2)^R \circ L(R_1)^R$. But by induction hypothesis, both of these are regular and given by R_1^R and R_2^R, so their concatenation is regular and we deduce $R^R = R_2^R \circ R_1^R$.

4. $R = R_1^*$, for some regular expression R_1. In this case, $L(R) = \{w^i : w \in L(R_1), i \in \mathbb{N}\}$. Therefore $L(R)^R = \{(w^i)^R : w \in L(R_1)\}$. We know that $(w^i)^R = (w^R)^i$, so this is $\{w^i : w \in L(R_1)^R, i \in \mathbb{N}\}$; but this is just $(L(R_1)^R)^*$. By structrual induction hypothesis, $L(R_1)^R$ is regular and described by R_1^R, so by closure under Kleene star, $L(R)^R$ is also regular and given by $(R_1^R)^*$.