Solutions to Homework #2

(Partially from William DeVries)

1. Proof using structural induction on the construction of L.

(1) If $L = \emptyset$, then $End(L, a)$ is $\emptyset$, which is regular.

(2) If $L = \{\Lambda\}$, then $End(L, a)$ is $\emptyset$, which is regular.

(3) If $L = \{b\}$, then $End(L, a)$ is $\emptyset$, or it is a, both are regular.

(4) If L_1 and L_2 are regular languages, and if $End(L_1, a)$ and $End(L_2, a)$ are regular languages, then,

(4.1) if $L = L_1 \cup L_2$, $End(L, a) = End(L_1, a) \cup End(L_2, a)$. This is a regular language.

(4.2) if $L = L_1 L_2$, then we have two cases to consider. If $L_2 = \{\Lambda\}$, then $End(L, a) = End(L_1, a)$. If $L_2 \neq \{\Lambda\}$, then $End(L, a) = L_1 End(L_2, a)$. Both cases give L is regular.

(4.3) if $L = L_1^*$, $End(L, a) = L_1^* End(L_1, a)$. This is a regular language.

2. $End(L, a) = \emptyset$, so the regular expression is $\emptyset$

3. Proof using structural induction on the construction of L. $L^r = x^r : x \in L$

(1) If $L = \emptyset$, then $L^r = \emptyset$, which is a regular language.

(2) If $L = \{\lambda\}$, then $L^r = \{\lambda\}$, which is a regular language.

(3) If $L = \{a\}$, then $L^r = \{a\}$, which is a regular language.

(4) Let L_1 and L_2 be two regular languages, and if both L_1^r and L_2^r are regular languages, then

(4.1) if $L = L_1 \cup L_2$, $L^r = L_1^r \cup L_2^r$. Which is a regular language.

(4.2) if $L = L_1 L_2$, $L^r = L_2^r L_1^r$. Which is a regular language.

(4.3) if $L = L_1^*$, $L^r = (L_1^r)^*$ Which is a regular language.

4. $cb(c(aa + bbb)^*)^*$. Why? You may follow the recursive steps shown in the above problem.

5. The shortest words are b, c.

6. Here is mine solution: I design a function *shortestwords* that returns a finite set of words with input of a regular expression.

FiniteSetOfWords *shortestwords*(RegExp r)

{

if $r = \emptyset$ then return $\emptyset$;

if $r = \{\Lambda\}$ then return $\{\Lambda\}$;

if $r = \{a\}$ for some symbol a then return $\{a\}$;

if $r = r_1 + r_2$ for some regular expressions r_1 and r_2 then
select all the shortest words from $shortestwords(r_1) \cup shortestwords(r_2)$;
return the selected words;
if $r = r_1 r_2$ for some regular expressions r_1 and r_2 then
return the concatenation $shortestwords(r_1)shortestwords(r_2)$;
if $r = r_1^*$ for some regular expression r_1 then
return $\{\Lambda\}$;
}