CS 381 HW 3 Solutions

October 19, 2001

2. Proof by Contradiction (below S means delta hat)

Assume there exists a Machine M such that L(M)= L and that the number of states in M < max { $|w|$: w in L }

Take the longest word in L, call it wmax. By the pigeonhole principle, we can see that wmax must visit at least one state more than once (since it has more characters than the Machine has states). Pick one such state, call it q. We know, that since wmax is accepted, that there exists a path from the start state to q, q to itself, and q to an accept state– i.e there exists non-empty strings x, y, and z such that:

S(p, x) = q (where p is the start state) S(q, y) = q S(q, z) = r (where r is an element of F)

We can show that xyyz is accepted: S(S(S(S(p,x), y), y), z) S(S(S(q, y), y), z) S(S(q, y), z) S(q, z) r

But $|xyyz| > |xyz|$ = wmax. We have a contradiction. So a DFA that accepts a finite language must have $>= max\{|w| : w \in L\}$ states.

5a. Prove that L = {w: $\#_0$(w) - $\#_1$(w) $\equiv$ 1 mod 3} is regular. We prove that L is regular by constructing a DFA that computes L.

Let Q = $\{q_0, q_1, q_2\}$
s = a;
F = {b}
$\delta(q_i, 0) = q_{(i+1) mod 3}$
$\delta(q_i, 1) = q_{(i-1) mod 3}$

We can easily see that the DFA keeps track of the difference between number of 0 and number of 1 mod 3 at every step.

5b. Prove that L' $=\{|$w: $\#_0$(w) - $\#_1$(w)$| \equiv 1$ mod 3$\}$ is not regular

- pick k > 0.
- let s = 0^{k+1} 1^k $\in$ L' for all k.
- s = xyz: x = 0^{k+1}, y = 1^k, z = ϵ. w = xyz, |y| $\geq$ k.
- y = uvw. u = 1^n, v = 1^m, w = 1^{k-n-m} m > 0.
- let i = 4 and consider w = xuv^4wz = 0^{k+1} 1^n 1^{4m} 1^{k-n-m} ϵ = 0^{k+1} 1^{k+3m}.
- $\#_0$(s) = k+1; $\#_1$(s) = k+3m. Therefore |w: $\#_0$(s) - $\#_1$(s)|= |1-3m|. m > 0 therefore, |1-3m| = 3m - 1. But 3m - 1 $\equiv$ 2 mod 3 is not equivalent to 1 mod 3. Therefore s is not in L'.

We have shown that $\forall$ k>0 $\exists$ x,y,z as defined above such that xyz $\in$ L', |y| > 0 so that for all uvw = y, |v| > 0, $\exists$ i such that xuv^iyz is not in L'. Therefore, L' is not regular.

6. Prove that for every regular expression r there exists a regular expression t, such that: $L(r) = \{w : w \notin L(t)\}$.

Solution:

1. For all regular expressions r, L(r) is a regular language (Theorem 8.1).
2. Regular languages are closed under complementation, therefore $\overline{L(r)}$ is a regular language.
3. For all r, L(r)=$\{w : w \notin \overline{L(r)}\}$ since by definition of complement $L(r) \bigcap \overline{L(r)} = \emptyset$ and $L(r) \bigcup \overline{L(r)} = \Sigma^*$.
4. Let A = $\overline{L(r)}$, then there exists t, with t a regular expression, s.t. A=L(t) (Theorem 8.1) and thus L(t) = $\overline{L(r)}$.
5. For all r there exists t such that $L(r) = \{w : w \notin L(t)\}$ by substitution.

Comment: The proof to this problem was fairly easy/intuitive. For that reason it was graded largely in part on how formal an argument you gave. In particular, we were looking for justification to claims you made (ie Theorem 8.1 and closure of regular languages under complementation).

7. Find a regular expression t over {0,1}* s.t. $L(t) = \{w : w \notin L(((0+1)(0+1))*)\}$

((0+1)(0+1))* = (00 + 01 + 10 + 11)* = all even length strings (Strings of length 2n) so L(t) = all strings not of even length (Strings of length 2n + 1) t = ((00 + 01 + 10 + 11)*) = {0, 1} ((00 + 01 + 10 + 11)*) = (0+1) ((00 + 01 + 10 + 11)*) = (0+1) ((0+1)(0+1))*

8. For each of the following languages L (over $S = \{o, p, q\}$) find a regular expression r_L such that $L(r_L) = L$.

i) L = {w: if p occurs in w then w ends with q}

Solution: (o+q)*p(o+p+q)*q+(o+q)*

Comment: Many people did not take into account the case when no p's occur in w, and thus left out the (o+q)* term. Another equally correct, yet less apparent, answer is (o+p+q)*q+(o+q)*.

ii) L = {w: $\#_p(w)$ is even}

Solution: ((o+q)*p(o+q)*p(o+q)*)*+(o+q)*

Comment: There were many different ways of expressing this. But again, some people did not take into account the case when 0 p's occur in w. Other's made the more serious mistake of ignoring cases like $s_1ps_2s_3s_4ps_5$ and constructed a regular expression using 'pp' to ensure an even number of p's. If you did that, make sure you understand the correct solution.

iii) L = {w: the next-to-last letter in w is p}

Solution: (o+p+q)*p(o+p+q)

Comment: This was an easy one. Most people got this right.

General Comments: These questions asked for a regular expression. Do not give patterns (chapter 7 was not assigned reading).

9. For each of the following languages L, describe the equivalence classes of R_L and determine the rank of R_L:

i $L = \{w \in \{0,1\}^* : w \textit{ contains exactly two } 1's\}$

ii $L = \{0^m 1^k 0^{m+k} : m, k \in N\}$

iii $L = L(ab(a+b)^*ab)$

Solution:
i.
The min-DFA that accepts L is:

$M = (Q, \Sigma, \delta, s, F)$
$Q = \{\sigma_1,\ \sigma_2,\ \sigma_3,\ \sigma_4\}$
$\Sigma = \{0, 1\}$
$s = \sigma_1$
$F = \{\sigma_3\}$
$\delta(\sigma_1, 0) = \sigma_1,\ \delta(\sigma_1, 1) = \sigma_2$
$\delta(\sigma_2, 0) = \sigma_2,\ \delta(\sigma_2, 1) = \sigma_3$
$\delta(\sigma_3, 0) = \sigma_3,\ \delta(\sigma_3, 1) = \sigma_4$
$\delta(\sigma_4, 0) = \sigma_4,\ \delta(\sigma_4, 1) = \sigma_4$

the rank of R_L =the number of the states of the min-DFA that accepts L,thus the rank=4. And each equivalence class corresponds to a diffirent state in the DFA. There are 4 states:$\sigma_1,\ \sigma_2,\ \sigma_3,\ \sigma_4$, so we have 4 equivalence classes:$A_i = \{x|\delta(\sigma_1, x) = \sigma_i\}$, i=1,2,3,4. or:

A_1. [0]=L(0^*)

A_2. [1]=L(0^*10^*)

A_3. [11]=L($0^*10^*10^*$)

A_4. [111]=L($0^*10^*10^*1(0 + 1)^*$)

ii.
This language is not regular, so we can not use the same idea as i. Consider any string $w_1 \in \{0, 1\}^*$. There are two cases:
1)w_1 can not be a prefix of any string in L. In this case:

$w_1 R_L w_2 \Leftrightarrow w_2$ can not be a prefix of any string in L either

so we get a equivalence class D=$\{w|for\ all\ x \in \{0, 1\}^*, wx \notin L\}$
2)w_1 is a prefix of some string in L. in this case, we have:

$$w_1 = 0^a, a \geq 0 \qquad \text{or}$$
$$w_1 = 0^a1^b, a \geq 0,\ b > 0 \qquad \text{or}$$
$$w_1 = 0^a1^b0^c, a \geq 0,\ b > 0,\ c > 0,\ a + b \geq c$$

Let A=L(0^*),B=L(0^*1^+),C=L($0^*0^+1^+$), then $w_1 \in A \vee w_1 \in B \vee w_1 \in C$

$w_1 = 0^a, a \in N$,in other word:$w_1 \in A$

in this case, we claim:$w_1\ R_L\ w_2 \Leftrightarrow w_2 = w_1 = 0^a$
let x=010^{a+2},$w_1x = 0^{a+1}10^{a+2} \in L$

$w_1\ R_L\ w_2 \Rightarrow w_2010^{a+2} \in L \rightarrow 1\ doesn't\ appear\ in\ w_2 \rightarrow w_2 \notin B, w_2 \notin C \Rightarrow w_2 \in A$

let $w_2 = 0^{a_2}$,we can easily conclude that:$w_1\ R_L\ w_2 \Leftrightarrow a = a_2$ thus we get infinite number of equivalence classes here: $A_i = \{0^i\}, i \in N$

$w_1 = 0^a 1^b, a \geq 0,\ b > 0$, in other words: $w_1 \in B$

in this case we claim: $w_1\ R_L\ w_2 \Leftrightarrow w_2 = 0^{a_2} 1^{b_2}, b_2 > 0, a+b = a_2 + b_2$
first $w_1\ R_L\ w_2 \Rightarrow w_2 \notin A$, because as we have show:$w_2 \in A \wedge w_1 R_L w_2 \Rightarrow w_1 \notin B$. Also $w_2 \notin C$, because $w_1 10^{a+b+1} = 0^a 1^{b+1} 0^{a+b+1} \in L$, but $w 10^{a+b+1} \notin L, for\ any\ w \in C$. Thus $w_2 \in B$. Let $w_2 = 0^{a_2} 1^{b_2}, b_2 > 0$.

$w_1\ R_L\ w_2 \Rightarrow For\ any\ x, w_1 x \in L \wedge w_2 x \in L \Rightarrow\ x = 1^i 0^j\ \Rightarrow\ j = a+b+i \wedge j = a_2+b_2+i\ \Rightarrow a+b = a_2+b_2$

also we have infinite number of equivalence classes in this case $B_i = \{0^a 1^b | a+b = i, b > 0\}, i > 0$

$w_1 = 0^a 1^b 0^c, a \geq 0,\ b > 0,\ c > 0,\ a+b \geq c$

in this case we claim:
$w_1\ R_L\ w_2 \Leftrightarrow w_2 = 0^{a_2} 1^{b_2} 0^{c_2}, b_2 > 0, c_2 > 0, a+b-c = a_2 + b_2 - c_2$
Proof:
$w_1 R_L w_2 \Rightarrow w_2 \notin A \wedge w_2 \notin B \Rightarrow w_2 = 0^{a_2} 1^{b_2} 0^{c_2}, for\ some\ a_2 \geq 0, b_2 > 0, c_2 > 0$

$For\ any\ x, w_1 x \in L \wedge w_2 x \in L \Rightarrow\ x = 0^j\ \Rightarrow\ j+c = a+b \wedge j+c_2 = a_2+b_2\ \Rightarrow a+b-c = a_2+b_2-c_2$

also we have infinite number of equivalence classes in this case $C_i = \{0^a 1^b 0^c | a+b-c = i, b > 0, c > 0, a+b \geq c\}, i \in N$

thus, the rank of $R_L = \infty$, the equivalence classes includes:

$A_i = \{0^i\}, i \in N$
$B_i = \{0^a 1^b | a+b = i, b > 0\}, i > 0$
$C_i = \{0^a 1^b 0^c | a+b-c = i, b > 0, c > 0, a+b \geq c\}, i \in N$
$D = \{w | for\ all\ x \in \{0,1\}^*, wx \notin L\}$

iii.
The min-DFA that accepts L is:

$M = (Q, \Sigma, \delta, s, F)$
$Q = \{\sigma_1,\ \sigma_2,\ \sigma_3,\ \sigma_4,\ \sigma_5,\ \sigma_6\}$
$\Sigma = \{a, b\}$
$s = \sigma_1$
$F = \{\sigma_6\}$
$\delta(\sigma_1, a) = \sigma_2,\ \delta(\sigma_1, b) = \sigma_6$
$\delta(\sigma_2, a) = \sigma_6,\ \delta(\sigma_2, b) = \sigma_3$
$\delta(\sigma_3, a) = \sigma_4,\ \delta(\sigma_3, b) = \sigma_3$
$\delta(\sigma_4, a) = \sigma_4,\ \delta(\sigma_4, b) = \sigma_5$
$\delta(\sigma_5, a) = \sigma_4,\ \delta(\sigma_5, b) = \sigma_3$
$\delta(\sigma_6, a) = \sigma_6,\ \delta(\sigma_6, b) = \sigma_6$

the rank of R_L =6.The 6 equivalence classes:$A_i = \{x | \delta(\sigma_1, x) = \sigma_i\}$, i=1,2,3,4,5,6.
or:

A_1. $[\epsilon] = \{\epsilon\}$

A_2. $[a] = \{a\}$

A_3. [ab]=L$(ab + abb + ab(a + b)^*bb)$

A_4. [aba]=L$(ab(a + b)^*a)$

A_5. [abab]=L$(ab(a + b)^*ab)$

A_6. [aa]=L$((aa + b)(a + b)^*)$