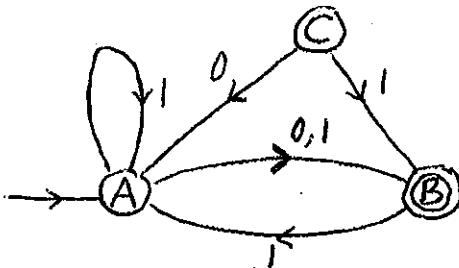


TEST #2 - Spring 2010

TIME: 75 min.

Answer all 6 questions. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each of the 6 questions on a separate page.

- (16) 1. Let  $L$  be the language accepted by the NFA shown on the right. Find NFAs which accept  
 (a)  $L^c$       (b)  $(L^c)^*$ .

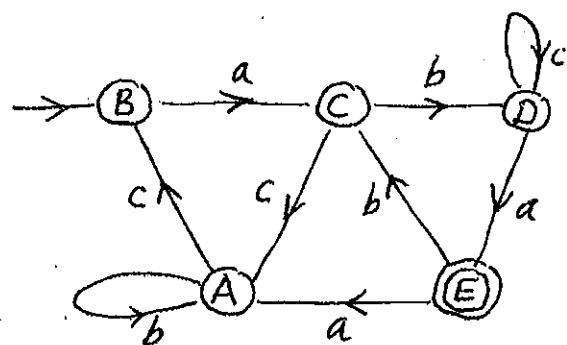


- (16) 2. (a) Find an NFA which is equivalent to the RLG given below.

$$G: \begin{array}{l} \rightarrow A, \quad A \xrightarrow{0} A, \quad A \xrightarrow{1} B, \quad B \xrightarrow{100} B, \quad B \xrightarrow{\lambda} C, \\ C \xrightarrow{\lambda} E, \quad C \xrightarrow{10} D, \quad D \xrightarrow{00} D, \quad D \xrightarrow{101} E, \quad E \xrightarrow{\lambda} A. \end{array}$$

- (b) Convert the NFA shown below on the right in Qu.#3 into an equivalent RLG.

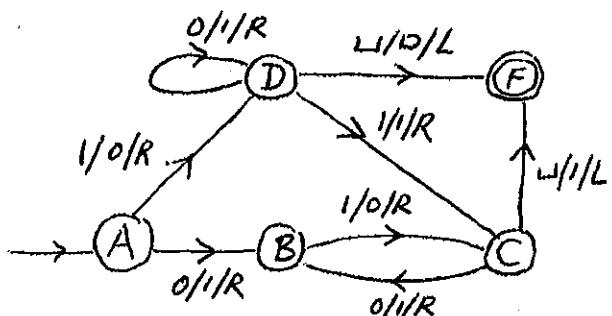
- (16) 3 (a) Find a regular expression for the language accepted by the NFA shown on the right.  
 (b) Suppose  $F$  is a finite language and  $A \cup F$  is regular, does it follow that  $A$  must also be regular?  
 [You may use any theorem from class.]



- (18) 4 (a) Define what are the initial functions and what are the primitive recursive functions.

- (b) Show that  $f(x, y) = 3x + 4y + 2$  is a primitive recursive function by finding primitive recursive functions  $g$  and  $h$  such that  $f = \text{prim.rec.}(g, h)$  and showing that  $g$  &  $h$  are primitive recursive.

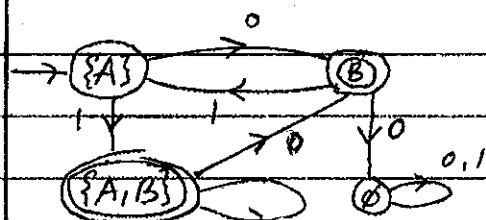
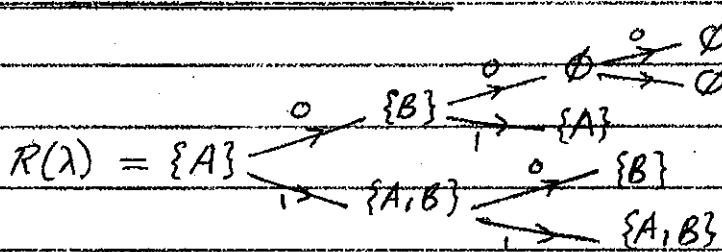
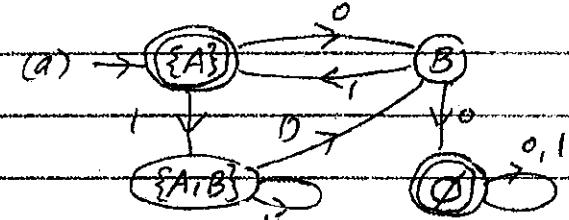
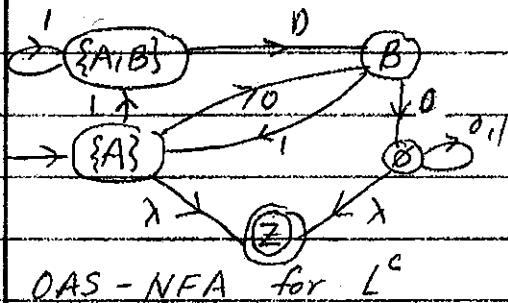
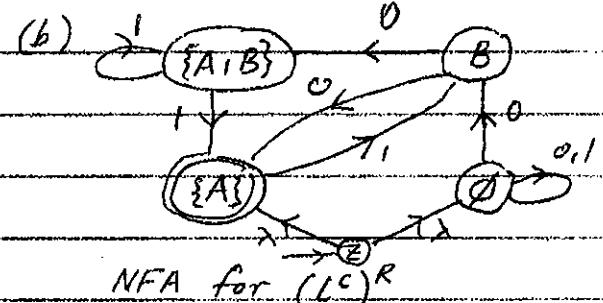
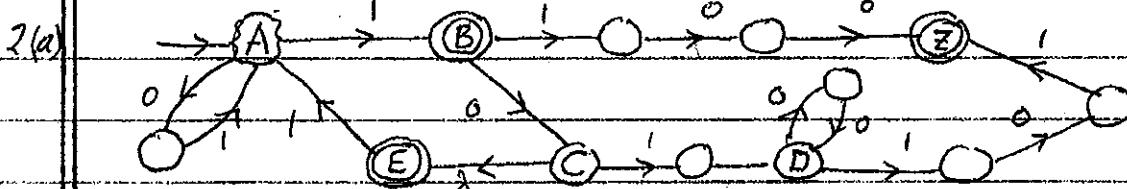
- (16) 5 (a) Write down what the Halting Problem says.  
 (b) Show what happens at each step if 10101 is the input for the TM,  $M$  shown on the right.  
 (c) Find the language accepted by  $M$ .



- (18) 6. Determine which of the following languages are regular and which are not. (a)  $L_1 = \{a^k \cdot b^n : k+1 \equiv 2n^2 \pmod{3}\}$       (b)  $L_2 = \{b^k \cdot c^n : k+1 > n^2\}$

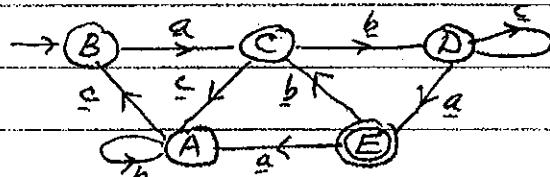
[If you say that it is regular, you must find a regular expression; if you say it is non-regular, you must give a complete proof.]

1.

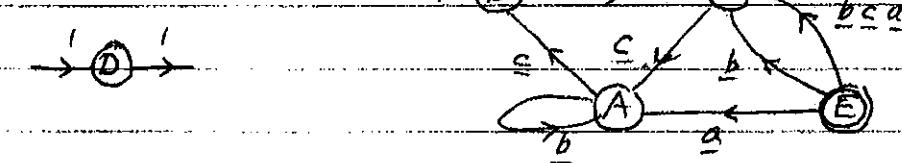
DFA for  $L$ .DFA for  $L^c$ .DAS-NFA for  $L^c$ NFA for  $(L^c)^R$ 

(b)  $\rightarrow B, B \rightarrow aC, C \rightarrow bD, C \rightarrow cA, A \rightarrow bA, A \rightarrow cB, D \rightarrow cD, D \rightarrow aE$   
 $E \rightarrow aA, E \rightarrow bC, E \rightarrow \lambda$ .

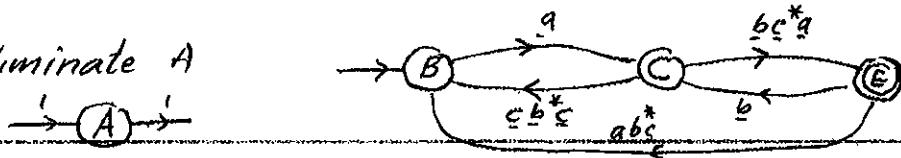
3(a) Corresponding GFA



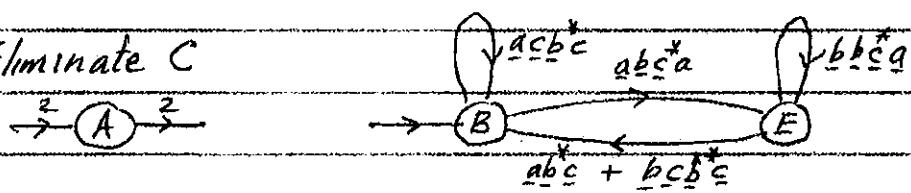
Eliminate D:



3(a) Eliminate A



Eliminate C

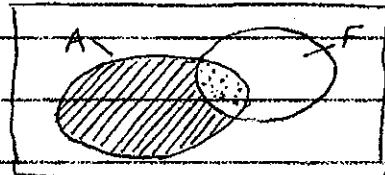


$$L(M) = (\underline{acb}^*)^* \cdot \underline{abc}^*a \cdot (\underline{bb}c^*a + (\underline{abc} + \underline{bcb}^*) \cdot (\underline{acb}^*)^* \cdot \underline{abc}^*)^*$$

$$(b) A = ((A \cup F) - F) \cup (A \cap F)$$

Since  $F$  is finite and  $A \cap F \subseteq F$ ,

$A \cap F$  is also finite. Now any finite set is regular. So  $F$  and  $A \cap F$  are regular. Also  $A \cup F$  is given as regular. Hence  $(A \cup F) - F$  is regular and so  $A = ((A \cup F) - F) \cup (A \cap F)$  will be regular by the closure Theorem.



4(a) The initial functions are: (i) the constant 0, (ii) the zero func. of 1 var,  $Z(x) \equiv 0$ , (iii) the successor function,  $S(x) = x+1$ ; and (iv) the projective functions  $I_k^{(n)}(x_1, \dots, x_n) = x_k$ .

The primitive recursive functions are the functions that can be obtained from the initial functions by using a finite number of applications of compositions and primitive recursions.

$$(b) f(x, 0) = 3x + 4(0) + 2 = 3x + 2, \quad \text{so } g(x) = 3x + 2$$

$$\begin{aligned} f(x, y+1) &= 3x + 4(y+1) + 2 \\ &= (3x + 4y + 2) + 4 \\ &= f(x, y) + 4 \\ &= h(x, y, f(x, y)) \end{aligned}$$

$$\begin{aligned} g(0) &= 3(0) + 2 = 2 \leftarrow g_1 \\ g(y+1) &= 3(y+1) + 2 = (3y+2) + 3 \\ &= g(y) + 3 = h_1(y, g(y)) \end{aligned}$$

$$\therefore h = S_0 S_1 S_2 S_3 I_3^{(3)} \quad \therefore g = \text{prim. rec.}(S_0 S_1 S_2 S_3 I_2^{(2)})$$

$$\therefore f = \text{prim. rec.}(g, h) = \text{prim. rec.}(\text{prim. rec.}(S_0 S_1 S_2 S_3 I_2^{(2)}, S_0 S_1 S_2 S_3 I_3^{(3)}), S_0 S_1 S_2 S_3 I_3^{(3)})$$

5(a) Halting Problem: Is there a TM  $H$  such that for an arbitrary TM  $M$  and an arb. input  $w$  for  $M$ ,  $H$  halts on  $c(M)\#cw$  in an acc. st., if  $M$  halts on  $w$ , and  $H$  halts on  $c(M)\#cw$  in a non-acc. st. if  $M$  does not halt on  $w$ ? (Here  $c(M)\#cw$  is  $\langle M, w \rangle$  coded into the alphabet of  $H$ )

$$(b) \langle A, 10101 \rangle \vdash \langle D, 00101 \rangle \vdash \langle D, 01101 \rangle \vdash \langle C, 01101 \rangle \vdash \langle B, 01111 \rangle \\ \vdash \langle C, 01110 \rangle \vdash \langle F, 011101 \rangle.$$

$$(c) L(M) = 10^* + 01(01)^* + 10^* 1(01)^*.$$

$$6(a) L_1 = \{a^k b^n : k+1 \equiv 2n^2 \pmod{3}\} = \{a^k b^n : k \equiv 2n^2 - 1 \pmod{3}\}$$

$$n \equiv 0 \pmod{3} \Rightarrow k \equiv 2(0)^2 - 1 = -1 \equiv 2 \pmod{3}$$

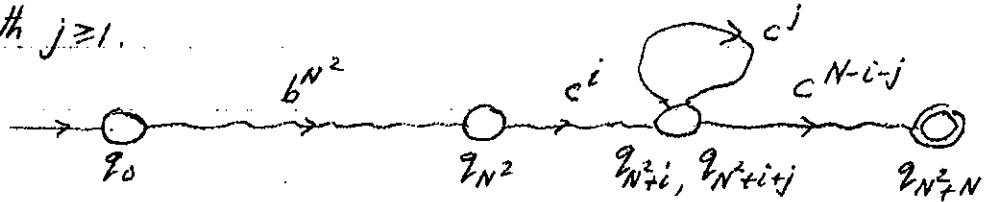
$$n \equiv 1 \pmod{3} \Rightarrow k \equiv 2(1)^2 - 1 = 1 \pmod{3}$$

$$n \equiv 2 \pmod{3} \Rightarrow k \equiv 2(2)^2 - 1 = 7 \equiv 1 \pmod{3}. \text{ So a reg. expr.}$$

$$\text{for } L_1 \text{ is } aa(aaa)^*(bbb)^* + a(aaa)^*.b(bbb)^* + a(aaa)^*.bb.(bbb)^*.$$

Hence  $L_1$  is a regular language

(b)  $L_2 = \{b^k c^n : k+1 > n^2\}$ . Suppose  $L_2$  is regular. Then we can find an NFA  $M$  such that  $L(M) = L_2$ . Let  $N$  be the number of states in  $M$ . Then  $b^{N^2} c^N \in L(M)$  because  $N^2+1 > (N)^2$ . (Here  $k=N^2$  &  $n=N$ ) So  $b^{N^2} c^N$  will be accepted by  $M$ . Since  $M$  has only  $N$  states and it takes  $N+1$  states to process  $c^N$ , the acceptance track of  $b^{N^2} c^N$  must have a loop as shown below with  $j \geq 1$ .



Now if we ride this loop twice we will see that  $M$  accepts  $b^{N^2} c^i c^j c^{j-i} c^{N-i-j} = b^{N^2} c^{N+j}$ . But  $N^2+1 \neq (N+j)^2$ , so this contradicts the fact that  $L(M) = L_2$ . Hence  $L_2$  is not regular.