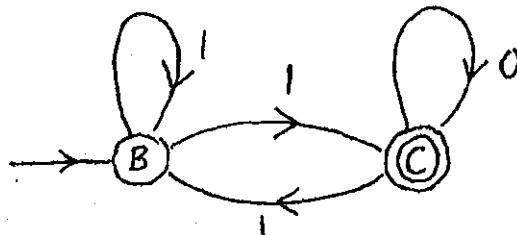


TEST #2 - Fall 2009

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.

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



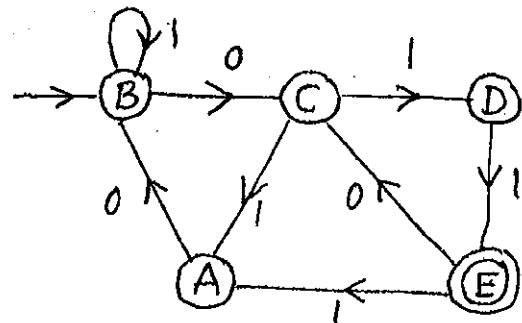
- (15) 2. (a) Find an NFA which is equivalent to the RLG given below.

$$G: \begin{array}{llll} S \rightarrow 1S, & S \rightarrow 01A, & A \rightarrow 10A, & A \rightarrow 0B, \\ B \rightarrow 101, & C \rightarrow \lambda, & C \rightarrow D, & B \rightarrow 1C, \\ C \rightarrow D, & D \rightarrow 01S, & D \rightarrow 11. & \end{array}$$

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

- (18) 3 (a) Find a regular expression for the language accepted by the NFA shown on the right.

- (b) Write down what the Halting Problem says and define what is the Busy beaver function.



- (18) 4 (a) Define what are the operations known as composition and primitive recursion.

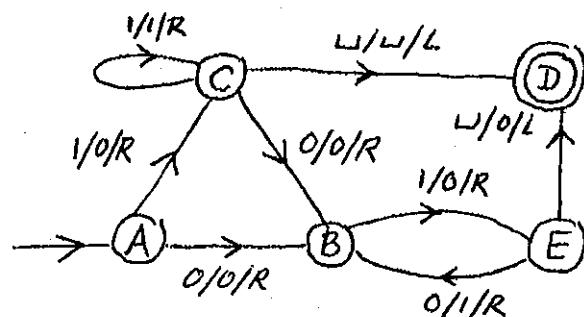
- (b) Show that $f(x, y) = 2x + 3y + 4$ is a primitive recursive function by finding primitive recursive functions g and h such that $f = \text{prim.rec.}(g, h)$.

[You must show that your g and h are primitive recursive.]

- (16) 5 (a) Define what is a Turing computable function with domain D .

- (b) Show what happens at each step if 10101 is the input for the TM, M shown on the right.

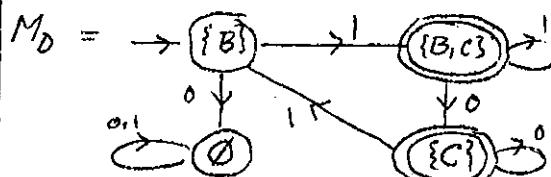
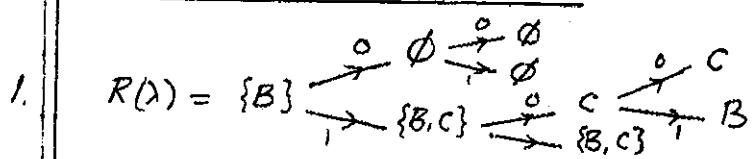
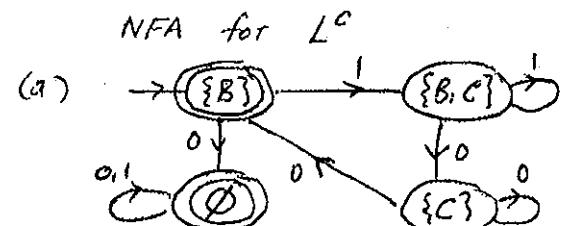
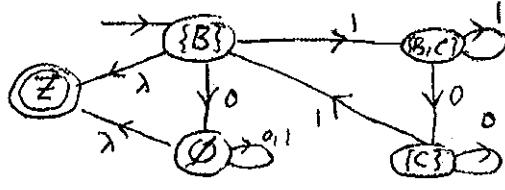
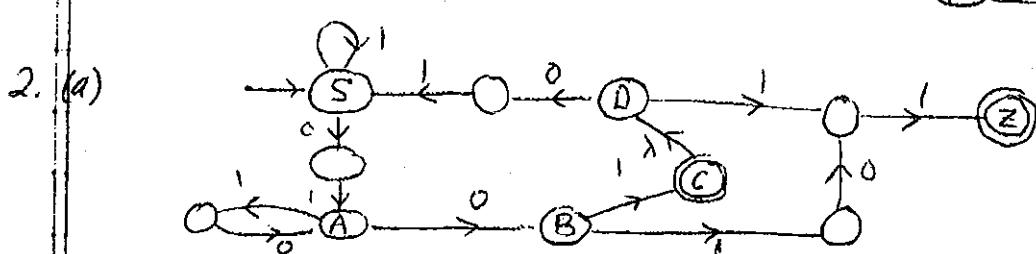
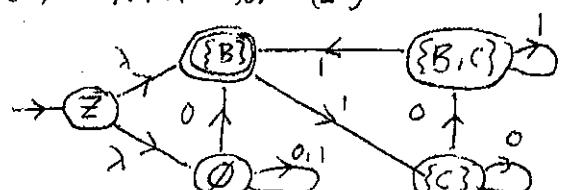
- (c) Find the language accepted by M .



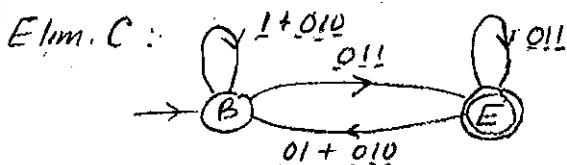
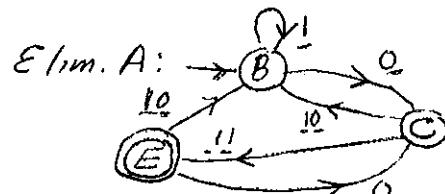
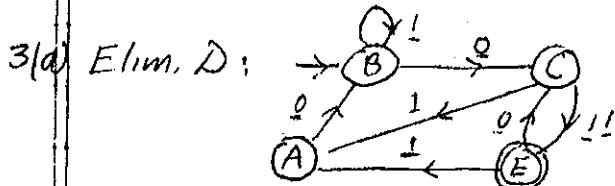
- (18) 6. Determine which of the following languages are regular and which are not.

$$(a) L_1 = \{a^k.b^n : n+2 \equiv k^2 \pmod{3}\} \quad (b) L_2 = \{b^k.c^n : n+2 > k^2\}$$

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

OAS-NFA for L^c :(b) NFA for $(L^c)^R$ 

(b) $\rightarrow B, B \rightarrow 1B, B \rightarrow 0C, C \rightarrow 1D, C \rightarrow 1A$
 $A \rightarrow 0B, D \rightarrow 1E, E \rightarrow 1A, E \rightarrow 0C, E \rightarrow \lambda$.



$$\text{Ans: } r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$$

$$= (1+010)^* \cdot 011 (011 + (01+010)(1+010)^* \cdot 011)^*$$

(b) The Halting problem asks if there is a TM H such that for an arb. TM M and an arb. input w; H will halt on (M, w) in an accepting state, if M halts on w; and H will halt on (M, w) in a non-accepting state, if M does not halt on w.

The Busy-beaver function is defined by $B(n) = \max_{H_n} \text{no. of 1's}$ a TM in H_n can produce. Here $H_n = \text{set of all TMs with tape alphabet } \{1, \lambda\} \text{ & } n \text{ states which halts on the blank tape.}$

4(a) Composition is the operation that produces a function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ from $g_1, \dots, g_k: \mathbb{N}^n \rightarrow \mathbb{N}$ and $h: \mathbb{N}^k \rightarrow \mathbb{N}$ by putting $f(\underline{x}) = h(g_1(\underline{x}), \dots, g_k(\underline{x}))$. Here $\underline{x} = \langle x_1, x_2, \dots, x_n \rangle$.

Primitive Recursion is the operation that produces a func. $f: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ from the functions $g: \mathbb{N}^n \rightarrow \mathbb{N}$ and $h: \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ by putting $f(\underline{x}, 0) = g(\underline{x})$ and $f(\underline{x}, y+1) = h(\underline{x}, y, f(\underline{x}, y))$. Again $\underline{x} = \langle x_1, \dots, x_n \rangle$.

(b) $f(x, y) = 2x + 3y + 4$. If $f = \text{prim. rec.}(g, h)$ then

$$\begin{aligned} f(\underline{x}, 0) &= 2x + 4 \leftarrow g(\underline{x}) & g(y) &= 2y + 4, \quad g(0) = 0 \\ f(\underline{x}, y+1) &= 2x + 3(y+1) + 4 & g(y+1) &= 2(y+1) + 4 \\ &= (2x + 3y + 4) + 3 & &= (2y + 4) + 2 \\ &= f(\underline{x}, y) + 3 \leftarrow h(\underline{x}, y, f(\underline{x}, y)) & &= g(y) + 2 \end{aligned}$$

$\therefore h = \text{so so so } I_3^{(3)}$ and $g = \text{prim. rec. } (\text{so so so so } 0, \text{ so so } I_2^{(2)})$

So $f = \text{prim. rec. } (\text{prim. rec. } (\text{so so so so } 0, \text{ so so } I_2^{(2)}), \text{ so so so } I_3^{(3)})$.

5(a) f is Turing computable if we can find a TM M such that for each $w \in D$, $\langle q_0, w \rangle \xrightarrow{*} \langle q_f, f(w) \rangle$ is a halted comput. in M with $q_f \in AH$.

(b) $\langle A, 10101 \rangle \xrightarrow{*} \langle C, 00101 \rangle \xrightarrow{*} \langle B, 00101 \rangle \xrightarrow{*} \langle E, 00001 \rangle \xrightarrow{*} \langle B, 00010 \rangle \xrightarrow{*} \langle E, 000100 \rangle$

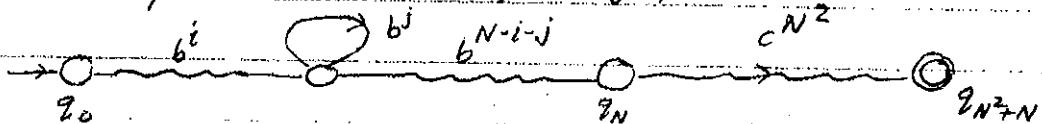
(c) $L(M) = 11^* + 01(01)^* + 11^*01(01)^*$. $\xrightarrow{*} \langle D, 000100 \rangle$.

6(a) $L_1 = \{a^k b^n : n+2 \equiv k^2 \pmod{3}\} = \{a^k b^n : n \equiv k^2 - 2 \pmod{3}\}$ is regular b/c.

$(aaa)^*(bbb)^*b + (aaa)^*a(bbb)^*bb + (aaa)^*aa(bbb)^*bb$ is a reg. expr. for L_1 .

(b) L_2 is not regular. Suppose L_2 was regular. Then we can find an NFA M such that $L(M) = L_2$. Let $N = \text{no. of states in } M$ and consider the string $b^N c^{N^2}$. Since $n+2 = N^2 + 2(N)^2 = k^2$, M accepts $b^N c^{N^2}$.

Since it takes $N+1$ states to process b^N , the acceptance track must contain a loop a shown below, with $j \geq 1$.



If we ride the loop twice, we will see that M accepts $b^{N+j} c^{N^2}$.

But this contradicts $L(M) = L_2$ b/c. $n+2 = N^2 + 2 \not\equiv (N+j)^2 = k^2$. So L_2 is really non-regular.