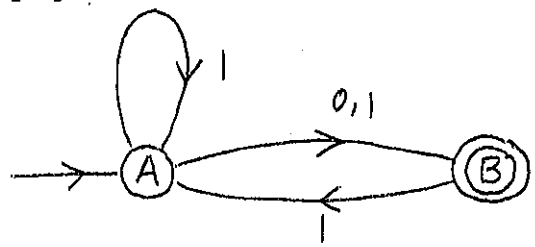
MAD 3512 - THEORY OF ALGORITHMS FLORIDA INT'L UNIV.

TEST #2 - Spring 2008 TIME: 75 min.

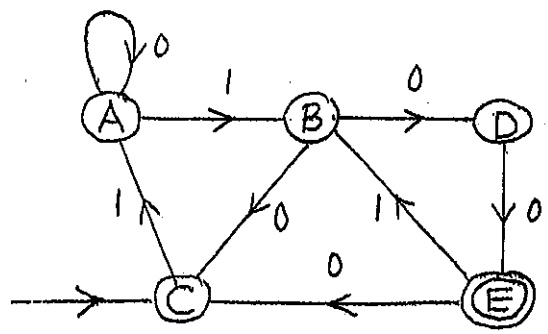
Answer all 6 questions. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each question on a separate page.

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



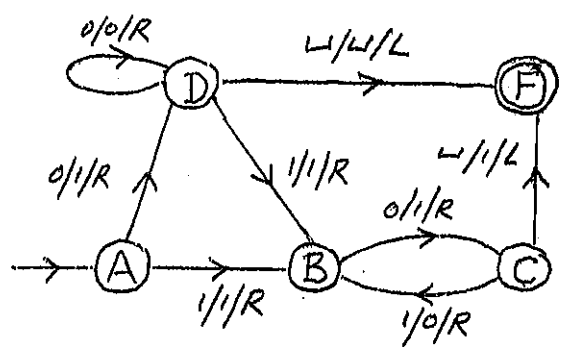
(15) 2. (a) Find an *NFA* which is *equivalent* to the RLG given below.
G: S→A, S→01B, A→10A, A→0C, B→λ,
B→1D, C→01, C→D, D→01S, D→11.
(b) Convert the NFA shown below on the right in Qu.#3 into an *equivalent RLG*.

(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 *initial functions* and what is the operation known as *primitive recursion*.
(b) Show that $f(x,y) = 3x+4y+1$ 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 01010 is the input for the TM, M shown on the right.
(c) Find the *language accepted* by M.



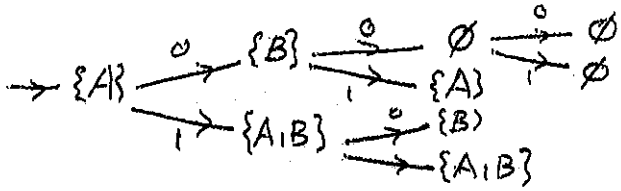
(18) 6. Determine which of the following languages are regular and which are not.
(a) $L_1=\{a^k.b^n: k\equiv n^2+1 \pmod 3\}$ (b) $L_2=\{b^k.c^n: k<n^2+1\}$

[*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. You may use the Pumping Lemma, if you so desire.*]

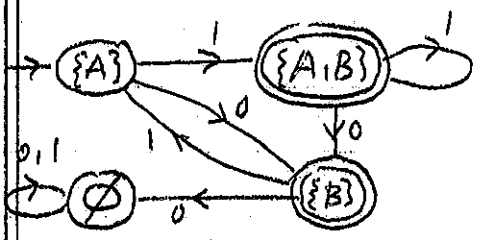
MAD 3512 – Theory of Algorithms — Florida Int'l Univ.

Solutions to Test #2 — Spring 2008

1.

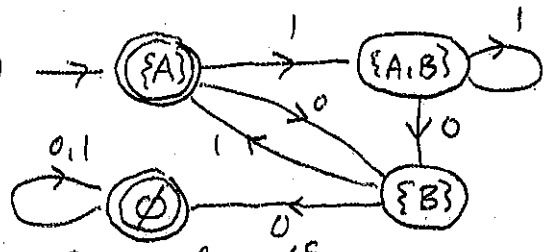


Note: DFAs are special NFAs

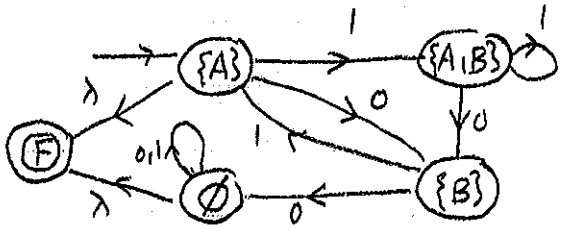


DFA for L

(a)

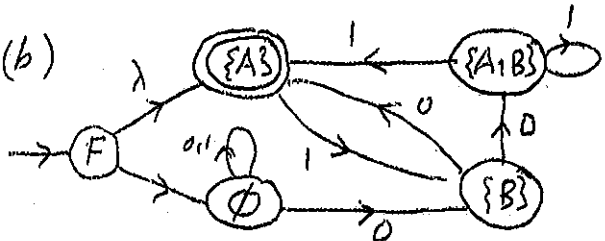


DFA for L^c



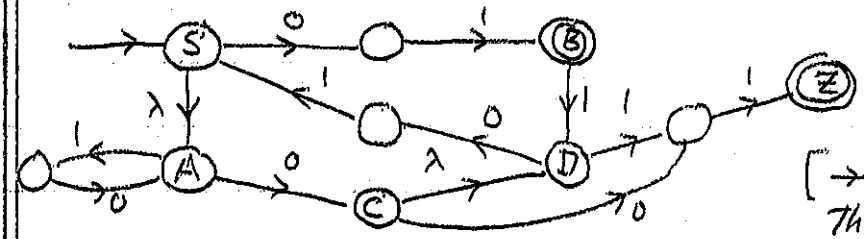
OAS – NFA for L^c

(b)



NFA for $(L^c)^R$.

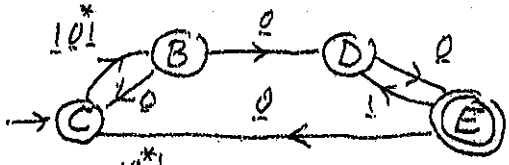
2(a)



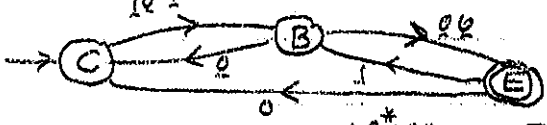
[$\to C$ means C is the starting variable]

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

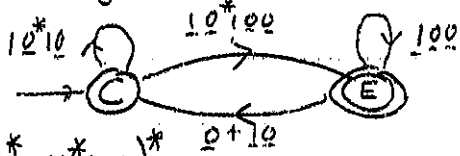
3(a) Eliminate A to get →



Eliminate D to get →



Eliminate B to get →



$L(M) = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$

$= (10^*10)^* \cdot 10^*100 \cdot (100 + (0+10) \cdot (10^*10)^* \cdot 10^*100)^*$.

3(b) The <u>Halting Problem</u> asks if there is a TM such that for an arb. TM M and an arb. input w for M,
H halts on $c(M)\#c(w)$ in an acc. state if M halts on w &
H halts on $c(M)\#c(w)$ in a non-acc. state if M does not halt on w.

Let $\mathcal{H}_n$ = set of all TMs with n states & tape alphabet $\{1, \sqcup\}$ which halts when started on the blank tape.
$\beta(n)$ = maximum number of 1's that a TM in $\mathcal{H}_n$ can produce.

4(a) The initial functions are: the constant 0, the zero function $z(x)=0$, the successor function $s(x)=x+1$, and the projective functions $I_k^{(n)}$ which are defined by $I_k^{(n)}(x_1,\ldots,x_n) = x_k$; $1 \le k \le n$.

Primitive recursion is the operation which produces a function $f: \mathbb{N}^{n+1} \to \mathbb{N}$ from the functions $g: \mathbb{N}^n \to \mathbb{N}$ & $h: \mathbb{N}^{n+2} \to \mathbb{N}$ by putting $f(\underline{x},0) = g(\underline{x})$ & $f(\underline{x},y+1) = h(\underline{x},y,f(\underline{x},y))$.

(b)
$$f(x,0) = 3x+1 \leftarrow g(x) \qquad g(0) = 3(0)+1 = 1$$
$$\begin{aligned} f(x,y+1) &= 3x+4(y+1)+1 \\ &= (3x+4y+1)+4 \\ &= f(x,y)+4 \leftarrow h(x,y,f(x,y)) \end{aligned} \qquad \begin{aligned} g(y+1) &= 3(y+1)+1 \\ &= (3y+1)+3 \\ &= g(y)+3 \end{aligned}$$
$\therefore\ g = \text{prim. rec.}(S\circ 0,\ S\circ S\circ S\circ I_2^{(2)})$ & $h = S\circ S\circ S\circ S\circ I_3^{(3)}$
$\therefore\ f = \text{prim. rec.}(\text{prim. rec.}(S\circ 0,\ S\circ S\circ S\circ I_2^{(2)}),\ S\circ S\circ S\circ S\circ I_3^{(3)})$

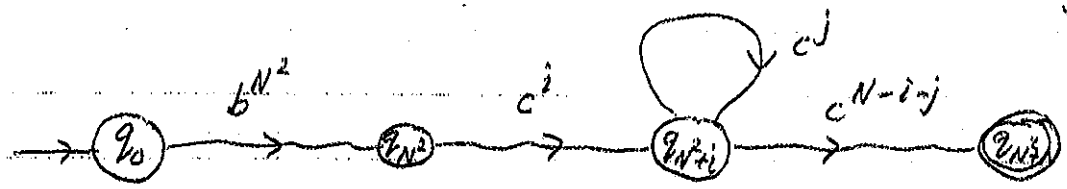
5(a) A function with domain D is said to be <u>Turing-computable</u> if we can find a TM M such that we have $w \in D \iff \langle q_0, w\rangle \vdash^* \langle q_F, f(w)\rangle$ is a halted computation in M with $q_F \in A$.

(b) $\langle A, \underline{0}1010\rangle \vdash \langle D, 1\underline{1}010\rangle \vdash \langle B, 11\underline{0}10\rangle \vdash \langle C, 111\underline{1}0\rangle$
$\vdash \langle B, 11110\underline{0}\rangle \vdash \langle C, 11101\underline{\sqcup}\rangle \vdash \langle F, 1110\underline{1}1\rangle$

(c) $L(M) = \underline{0}\underline{0}^* + \underline{0}\underline{0}^*\underline{1}\underline{0}(\underline{1}\underline{0})^* + \underline{1}\underline{0}(\underline{1}\underline{0})^* = \underline{0}\underline{0}^* + (\underline{0}\underline{0}+\underline{\lambda}).\underline{1}\underline{0}.(\underline{1}\underline{0})^*$

6(a) If $n \equiv 0 \pmod 3$, then $k \equiv 0^2+1 \equiv 1 \bmod 3$; if $n \equiv 1 \pmod 3$, then $k \equiv 1^2+1 \equiv 2 \pmod 3$; & if $n \equiv 2 \pmod 3$, $k \equiv 2^2+1 \equiv 2 \pmod 3$.
So $L_1 = a(aaa)^*(bbb)^* + aa(aaa)^* \cdot b(bbb)^* + aa(aaa)^* \cdot bb \cdot (bbb)^*$ and is therefore a regular language.

(b) Suppose L_2 was a regular language. Then we can find a DFA M such that $L(M) = L_2$. Let N be the number of states in M and consider the string $b^{N^2} \cdot c^N$. Since $N^2 < (N)^2+1$, $b^{N^2}c^N \in L_2$ & will be accepted by M. Since it takes $N+1$ states to process the c^N, the acceptance track of $b^{N^2}c^N$ mus have a loop as show below with $j \geq 1$.



Now if we skip this loop, we will see that M accepts
$$b^{N^2} \cdot c^i \cdot c^{N-i-j} = b^{N^2} c^{N-j}.$$
But $N^2 \neq (N-j)^2+1$, so this contradicts the fact that $L(M) = L_2$. Hence L_2 cannot be a regular lang.