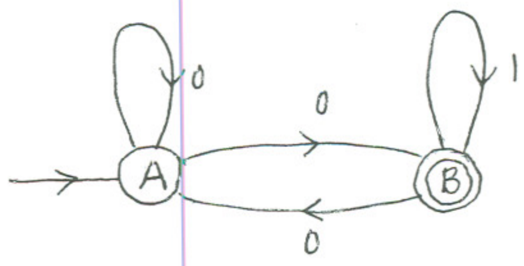
MAD 3512 - THEORY OF ALGORITHMS FLORIDA INT'L UNIV.

TEST #2 - Spring 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 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) $(L^c)^R$.



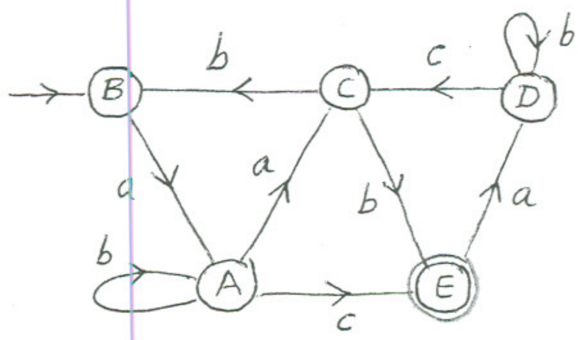
(15) 2. (a) Find an *NFA* which is *equivalent* to the RLG given below.

G: $S \to A$, $S \to 10B$, $A \to 01A$, $A \to 1C$, $B \to 0D$,
$B \to 010$, $C \to \lambda$, $C \to D$, $D \to 10S$, $D \to 01$.

(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 a *Primitive Recursive function*.



(16) 4(a) If B, $A \cup B$, and $A^c \cup B^c$ are all regular languages, does it always follow that A must be regular? (Justify your answer)

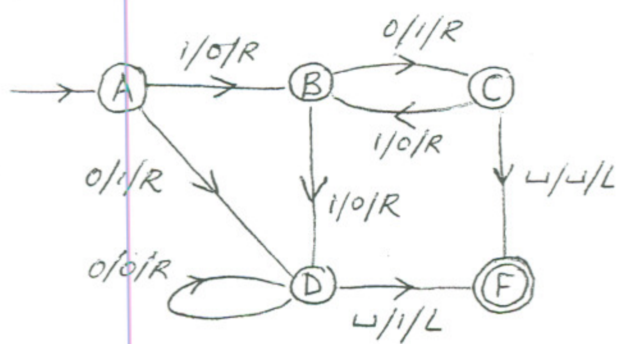
(b) If C and D are both non-regular languages, does it always follow that C.D must be non-regular? (Justify your answer)

[You may use any result that was proved in class for Problem #4]

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

(b) Show what happens at each step if 10110 is the input for the TM, M shown on the right.

(c) Find the *language accepted* by M.



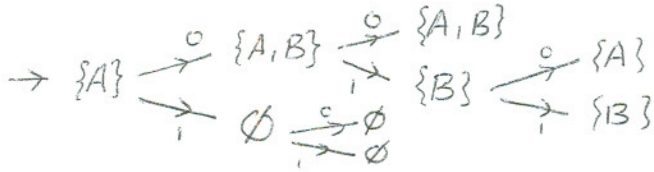
(20) 6. Determine which of the following languages are regular and which are not.

(a) $L_1=\{a^k.b^n:\ k\equiv n^2-1 \pmod 3\}$ (b) $L_2=\{b^k.c^n:\ k>n^2-1\}$

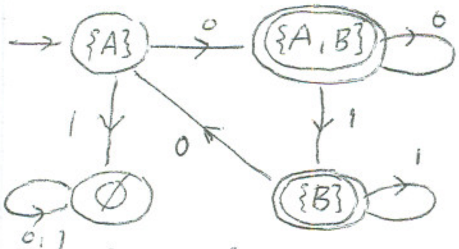
[If you say that it is regular, you must find a regular expression for it; if you say it is non-regular, you must give a complete proof.]

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

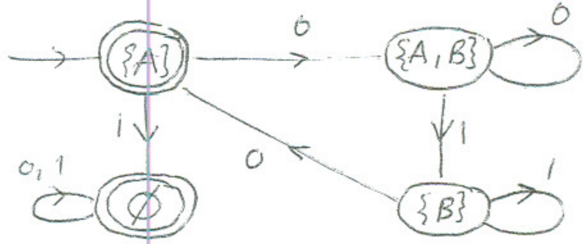
Solutions to Test # 2 — Spring 2009



1 (a)

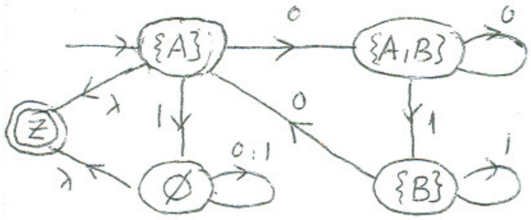


DFA for L

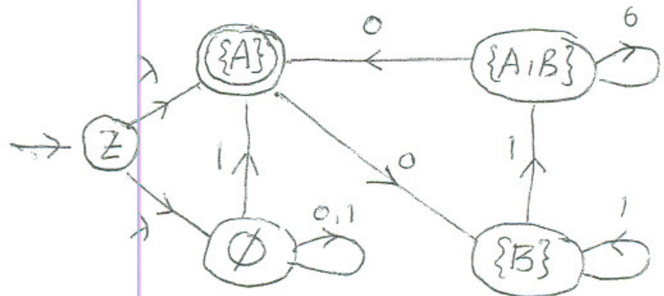


DFA for L^c

1 (b)

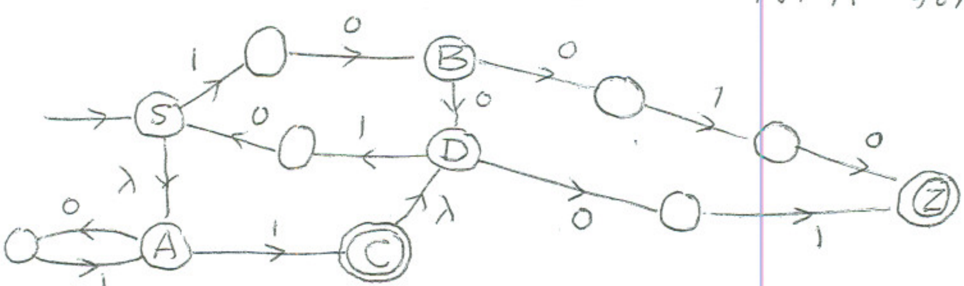


OAS-NFA for L^c



NFA for $(L^c)^R$

2(a)

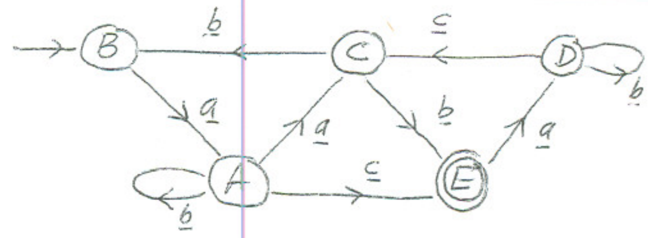


2 (b) $\rightarrow B$, $B \rightarrow aA$, $A \rightarrow bA$, $A \rightarrow aC$, $A \rightarrow cE$
$C \rightarrow bB$, $C \rightarrow bE$, $D \rightarrow bD$, $D \rightarrow cC$, $E \rightarrow aD$, $E \rightarrow \lambda$

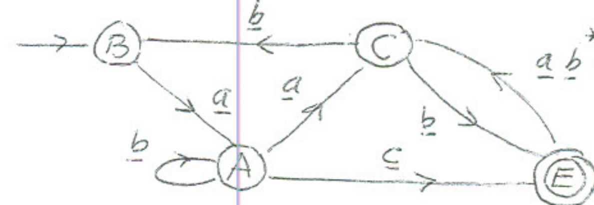
3 (b) Halting Problem: Is there a TM H such that for an arbitrary TM M and an arbitrary input w, H will halt on $c(M)\#c(w)$ in an an accepting state, if M halts on w, and on $c(M)\#c(w)$ in a non-accepting state, if M does not halt on w.

A primitive recursive function is a function which can be obtained from the initial functions by a finite number of applications of compositions and primitive recursions.

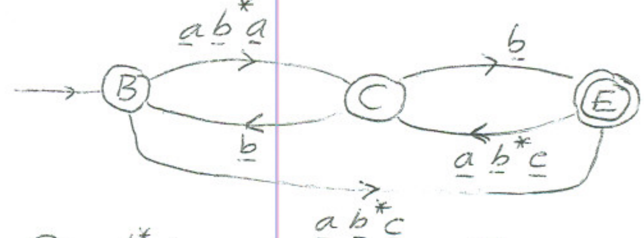
3 (a) M_0 is the GFA



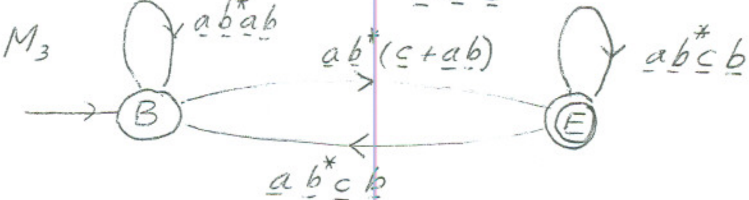
Eliminate D : M_1



Eliminate A : M_2



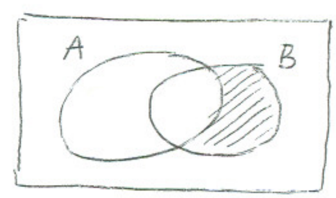
Eliminate C : M_3



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

$\therefore \; L(M) = (\underline{a}\underline{b}^*\underline{a}\underline{b})^* \cdot \underline{a}\underline{b}^*(\underline{c}+\underline{a}\underline{b}) \cdot \left(\underline{a}\underline{b}^*\underline{c}\underline{b} + \underline{a}\underline{b}^*\underline{c}\underline{b} \cdot (\underline{a}\underline{b}^*\underline{a}\underline{b})^* \underline{a}\underline{b}^*(\underline{c}+\underline{a}\underline{b})\right)^*$

4 (a) We know that $A = (A \cup B) - (B \cap (A^c \cup B^c))$.



Now since B and $A^c \cup B^c$ are regular, it follows from the closure Theorem that $B \cap (A^c \cup B^c)$ will be regular. And since $A \cup B$ is also regular $(A \cup B) - (B \cap (A^c \cup B^c))$ will also be regular by the Closure Theorem again. So A must be a regular language.

(b) Take $C = \{a^{n^2} : n \geq 0\}$ and $D = \underline{a}^* - C$. Then C and D are both non-regular languages. Also

$C \cdot D = \{a^0, \ldots\} \cdot \{a^k : k \neq n^2\} \supseteq \{a^k : k \neq n^2\}$ and

$C \cdot D \supseteq \{a^1, \ldots\} \cdot \{a^{n^2 - 1} : n \geq 2\} \supseteq \{a^{n^2} : n \geq 2\}$.

So $C \cdot D = \{a^k : k \geq 2\} = \underline{a}\underline{a}(\underline{a}^*)$. So $C \cdot D$ does not have to be non-regular. (It could be regular as shown here)

Extra: $C \cdot D = \{a^0, a^1, a^4, a^9, \ldots\} \cdot \{a^2, a^3, a^5, a^6, \ldots\} = \{a^2, a^3, a^4, a^5, a^6, a^7, \ldots\}$

C was proved to be non-reg in class. Also if D was reg., then C would be reg.

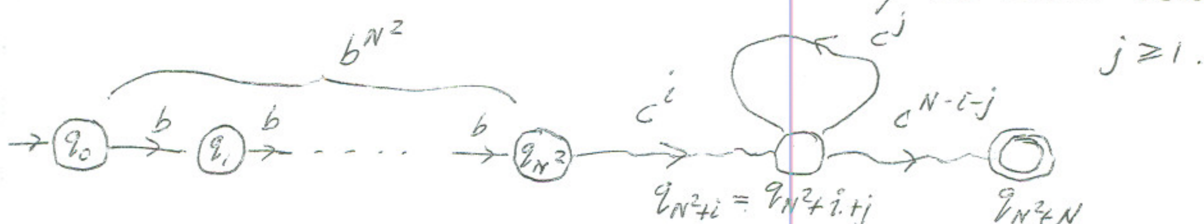
5(a) A Turing computable function with domain D is any function for which we can find a TM M such that $w \in D$
$\Leftrightarrow \langle q_0, \underline{w} \rangle \vdash^* \langle q_F, \varphi\, \underline{f}(w) \rangle$ is a halted computation with $q_F \in A(M)$

(b) $\langle A, \underline{1}0110 \rangle \vdash \langle B, 0\underline{0}110 \rangle \vdash \langle C, 01\underline{1}10 \rangle \vdash \langle B, 010\underline{1}0 \rangle$
$\vdash \langle D, 0100\underline{0} \rangle \vdash \langle D, 01000\underline{\sqcup} \rangle \vdash \langle F, 0100\underline{0}1 \rangle$ halts.

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

6(a) $L_1 = \{a^k b^n : k \equiv n^2 - 1 \pmod 3\}$. If $n \equiv 0 \pmod 3$, then $k \equiv 0^2 - 1 \equiv 2 \pmod 3$; if $n \equiv 1 \pmod 3$, $k \equiv 1^2 - 1 \equiv 0 \pmod 3$; and if $n \equiv 2 \pmod 3$, then $k \equiv 2^2 - 1 \equiv 0 \pmod 3$. So $L_1 = \{a^{3p+2} b^{3q} : p, q \geq 0\} \cup \{a^{3p} b^{3q+1} : p, q \geq 0\} \cup \{a^{3p} b^{3q+2} : p, q \geq 0\}$. Hence a regular expression for L_1 will be $(\underline{a}\,\underline{a}\,\underline{a})^* \underline{a}\,\underline{a}(\underline{b}\,\underline{b}\,\underline{b})^* + (\underline{a}\,\underline{a}\,\underline{a})^* . (\underline{b}\,\underline{b}\,\underline{b})^* \underline{b} + (\underline{a}\,\underline{a}\,\underline{a})^* . (\underline{b}\,\underline{b}\,\underline{b})^* . \underline{b}\,\underline{b}$. Hence L_1 is a regular language.

(b) Suppose L_2 was regular. Then we can find an NFA M_2 such that $L(M_2) = L_2$. Let N = the number of states in M_2. Since $b^{N^2} c^N \in L_2$, M_2 will accept $b^{N^2} c^N$. Also since it takes $N+1$ states to process the c^N part of this string, the acceptance track of $b^{N^2} c^N$ must have a loop as shown below:



Now if we ride this loop twice, we will see that M_2 accepts the string $b^{N^2} . c^i . c^j . c^j . c^{N-i-j} = b^{N^2} . c^{N+j}$. But $N^2 \not\equiv (N+j)^2 - 1$, because $j \geq 1$, so this means that $b^{N^2} . c^{N+j} \notin L_2$. But this contradicts the fact that $L(M_2) = L_2$. Hence L_2 cannot be regular.

END