(15) 1. Let $L$ be the language accepted by the NFA shown on the right. Find NFAs which accept
(a) $L^2$   
(b) $(L^2)^r$.

(15) 2. (a) Find an NFA which is equivalent to the RLG given below.
G: $S \rightarrow A$, $S \rightarrow 10B$, $A \rightarrow 01A$, $A \rightarrow 1C$, $B \rightarrow 0D$, $B \rightarrow 010$, $C \rightarrow \lambda$, $C \rightarrow D$, $D \rightarrow 10$, $D \rightarrow 01$.
(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 $A^* \cup B^*$ 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 $C \cdot D$ 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) $L_1 = \{a^k b^k : k \equiv n^2 - 1 \pmod{3}\}$
(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.]
1 (a)  

DFA for $L$

1 (b)  

OAS-NFA for $L^c$

2(a)  

NFA for $(L^c)^R$

2 (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 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 $C$: $M_3$

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

$\therefore L(M) = (abab)^*ab(c+ab)^*a^*bc^*b^*(abab)^*a^*bc^*b^*(c+ab)^*a^*bc^*b^*$

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

$C \cap D = \{a^n : n \geq 0\}$ and

$C \cap D = \{a^k, k \geq 2\}$

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

Extra: $C \cap D = \{a^n : n \geq 0\} \cup \{a^n : n \geq 2\} = \varnothing(a^*)$. $C$ 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 \(\mathbb{D}\) is any function for which we can find a TM \(M\) such that for \(w, x \in \mathbb{D}\), \(<w, x> \rightarrow <w, f_x>\) is a halted computation with \(f_x \in \mathbb{A}(\mathbb{N})\).

(b) \(<A, 10110> \rightarrow <B, 00110> \rightarrow <C, 01110> \rightarrow <B, 01010> \rightarrow <D, 01000> \rightarrow <D, 010000> \rightarrow <F, 0100001>\) halts.

(c) \[L(M) = 0^* 0^* + 110^* + 1(01)^* 0^* + 1(01)^* 0^* \]
\[= 0^* 0^* + 1(01)(0^* + 0^*)\]

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

But \(N^2 \neq (N+j)^2 - 1\), 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