TEST #2 - Spring 2008
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.


(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.]

(18) 5(a) Define what is a Turing comutable 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.]
Solutions to Test #2

1. \[
\begin{align*}
&\{A\} \xrightarrow{0} \{A\} \xrightarrow{1} \{A\} \xrightarrow{\emptyset} \emptyset \\
&\{A\} \xrightarrow{\emptyset} \{A\} \xrightarrow{1} \{A\} \xrightarrow{\emptyset} \emptyset \\
\end{align*}
\]

DFA for \(L\)

1. \[
\begin{align*}
&\{A\} \xrightarrow{0} \{A\} \xrightarrow{1} \{A\} \xrightarrow{\emptyset} \emptyset \\
&\{A\} \xrightarrow{\emptyset} \{A\} \xrightarrow{1} \{A\} \xrightarrow{\emptyset} \emptyset \\
\end{align*}
\]

DFA for \(L^c\)

OAS - NFA for \(L^c\)

NFA for \((L^c)^c\)

2. \[
\begin{align*}
&\xrightarrow{\lambda} C, \quad C \xrightarrow{IA}, \quad A \xrightarrow{0A}, \quad A \xrightarrow{IB}, \quad B \xrightarrow{0C} \\
&\quad B \xrightarrow{0D}, \quad D \xrightarrow{0E}, \quad E \xrightarrow{IB}, \quad E \xrightarrow{0C}, \quad E \xrightarrow{\lambda}
\end{align*}
\]

[\(\rightarrow C \text{ means } C \text{ is the starting variable}\)]

3. (a) Eliminate A to get

Eliminate D to get

Eliminate B to get

\[
L(M) = r_1^* r_2 (r_3 + r_3^* r_2)^* \\
= (10^* 10)^* 10^* 100 . (100 + (0+10) . (10^* 10)^* 10^* 100)^*.
\]
The Halting Problem 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)#9w 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 \{0,1\} 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
\( I^n_k \) which are defined by \( I^n_k(x_1, \ldots, x_n) = x_k \); 1 <= k <= n.

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

(b) \( f(x,0) = 3x+1 \) \( \rightarrow \) \( g(x) \)
\( g(0) = 3(0)+1 = 1 \)
\( f(x,y+1) = 3x + 4(y+1) + 1 \)
\( = (3x + 4y + 1) + 4 \)
\( = (3x + 4y + 1) + 4 \)
\( = h(x,y, f(x,y)) \)
\( g = \text{prim. rec. (} s_0, s_0 s_0 s_0 I_2^{(3)} \text{)} \) & \( h = s_0 s_0 s_0 s_0 I_2^{(3)} \)

5.(a) A function with domain \( D \) is said to be Turing-computable
if we can find a TM M such that we have we D \( \Rightarrow \)
\( \langle g_0, w \rangle \xrightarrow{\ast} \langle g_f, f(w) \rangle \) is a halted computation in M with \( g_f \in A \).

(b) \( \langle A, 01010 \rangle \xrightarrow{\ast} \langle D, 11010 \rangle \xrightarrow{\ast} \langle B, 11010 \rangle \xrightarrow{\ast} \langle C, 1110 \rangle \)
\( \xrightarrow{\ast} \langle B, 11100 \rangle \xrightarrow{\ast} \langle C, 11101 \rangle \xrightarrow{\ast} \langle F, 11101 \rangle \)

(c) \( L(M) = 00^* + 00^* 10(10)^* + 10(10)^* = 00^* + (00 + 1).10.(10)^* \)
6(a) If \( n = 0 \pmod{3} \), then \( k = 0^2 + 1 = 1 \pmod{3} \); if \( n = 1 \pmod{3} \), then \( k = 1^2 + 1 = 2 \pmod{3} \); & if \( n = 2 \pmod{3} \), \( k = 2^2 + 1 = 2 \pmod{3} \). So \( L_1 = a(aa)^*b(bb)^* + a(aa)^*b(bb)^*a(aa)^*b(bb)^* \) 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 \( b^{N^2}c^N \). Since \( N^2 < (N+1)^2 \), \( b^{N^2}c^N \in L_2 \) & will be accepted by \( M \). Since it takes \( NH \) states to process the \( c^N \), the acceptance track of \( b^{N^2}c^N \) must have a loop as shown below with \( j = 1 \).

Now if we skip this loop, we will see that \( M \) accepts \( b^{N^2}c^i \cdot c^{N-i-j} = b^{N^2}c^{N-j} \). 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 language.