TEST #2 - Fall 2011

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: -A, A-10A, A-0B, B-01, B-1C, B-\lambda, C-0E, C-D, D-01D, D-10, D-\lambda, E-1A.
(b) Convert the NFA shown below on the right in Qu. #3 into an equivalent RLG.

(15) 3 (a) Define what is the Busy beaver function.
(b) Find a regular expression for the language accepted by the NFA shown on the right.

(22) 4 (a) Define what it means for $f$ to be obtained from $g$ and $h$ by primitive recursion. Show that $f(x,y) = 2x + 3y + 1$ is a primitive recursive function by finding primitive recursive functions $g$ and $h$ such that $f = \text{prim.rec.}(g,h)$
(b) Define what it means for $f$ to be obtained from $g$ by minimization. Show that $f(x) = \sqrt{x+2}$ is a recursive function by finding a primitive recursive function $g$ such that $f = \mu[g,0]$. [You may use the fact that MONUS, ADD, & MULT are prim. rec. if needed in 4(b), but you can't do so in 4(a).]

(15) 5 (a) Define what is a Turing-acceptable language on the alphabet $V$.
(b) Show what happens at each step if 00101 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) $L_1 = \{a^k b^n : k \text{ (mod 3)} < 2n \text{ (mod 3)} \}$  
(b) $L_2 = \{b^k c^n : 2k < n \}$.

[If you say that it is regular, you must find a regular expression; if you say it is non-regular, you must give a complete proof.]
1.

- $\emptyset \rightarrow \{B\} \rightarrow \emptyset \leftarrow \emptyset \leftarrow \{B\} \leftarrow \emptyset \leftarrow \emptyset \leftarrow \{B\}$

- DFA for $L$

- $\emptyset \rightarrow \{B\} \rightarrow \emptyset \rightarrow \emptyset \rightarrow \{B\}$

- DFA for $L^c$

- $\emptyset \rightarrow \{B\} \rightarrow \emptyset \rightarrow \emptyset \rightarrow \{B\}$

- OAS-NFA for $L^c$

- $\emptyset \rightarrow \{B\} \rightarrow \emptyset \rightarrow \emptyset \rightarrow \{B\}$

- NFA for $(L^c)^r$

2.

(a) $A \rightarrow bB, B \rightarrow aC, B \rightarrow cE, E \rightarrow bE, E \rightarrow \lambda$

(b) $C \rightarrow bA, A \rightarrow aB, B \rightarrow aC, B \rightarrow cE, E \rightarrow bE, E \rightarrow \lambda$

3. (a) The Busy Beaver function is defined as follows: from blank tape $\beta(n) = \text{maximum number of 1's a TM in } \mathcal{J}_n \text{ can produce}$

Here $\mathcal{J}_n = \text{set of all TMs with tape alphabet \{a, b\} and } n \text{ states, which halts when started on the blank tape}$

(b) Corresponding GFA:

1. Eliminate $A$
2. Eliminate D

\[ L(D) = \frac{1}{2} \left( \frac{1}{2^2} \right) \]

\[ = (baa)^* \cdot bac \left( (b+abc) + (ac+aba), (baa)^* \cdot bac \right) \]

3. Eliminate B

\[ \therefore L(N) = \frac{1}{2} \left( \frac{1}{2^2} \right) \]

\[ = (baa)^* \cdot bac \left( (b+abc) + (ac+aba), (baa)^* \cdot bac \right) \]

4(a) \( f \) is obtained from \( g \) and \( h \) by primitive recursion if

\[ f(x, 0) = g(x) \quad \text{and} \quad f(x, y+1) = h(x, y, f(x, y)). \]

Here \( X = \langle x_1, \ldots, x_n \rangle \), \( g : N^n \to N \), \( f : N^{n+1} \to N \) and \( h : N^{n+2} \to N \).

\[ f(x, y) = 2x + 3y + 1. \]

So \( f(x, 0) = 2x + 1 \Rightarrow g(x) = 2x + 1 \)

Also \( f(x, y+1) = 2x + 3(y+1) + 1 = (2x + 3y + 1) + 3 = f(x, y) + 3 \)

\[ \Rightarrow h(x, y, f(x, y)) = f(x, y) + 3. \]

\[ \therefore h = s \circ s \circ s \circ I_3^{(e)} \]

Now \( g(0) = 1 = s_0, \) and \( g(y+1) = 2(y+1) + 1 = (2y+1) + 2 \)

\[ \therefore g = \text{prim. rec.} \left( (s_0, s \circ s \circ I_2^{(e)}) \right) \]

\[ = \text{prim. rec.} \left( \text{prim. rec.} \left( (s_0, s \circ s \circ I_2^{(e)}), s \circ s \circ s \circ I_3^{(e)} \right) \right). \]

and so is a primitive recursive function.

(b) \( f \) is obtained from \( g \) by minimization if \( f : N^n \to N \)

and \( f(x) = \{ \text{smallest } y \text{ such that } g(x, y) = 0 \}

undfined, if \( g(x, y) > 0 \) for every \( y \).

Here \( g : N^{n+1} \to N \) must be a total function.

Take \( g(x, y) = x + 2 - y^2. \) Then \( (\mu y)[g(x, y) = 0] \)

is just the function \( (\mu y)[x + 2 - y^2 = 0] = \sqrt{x+2} \)

Hence \( f = \mu \left[ g, 0 \right] \)

\[ = \mu \left[ \text{MONUS} \circ \left( s \circ s \circ I_2^{(e)} \right), \text{MULT} \circ \left( I_2^{(e)}, I_2^{(e)} \right), 0 \right] \]

and so is a recursive function.
A language \( L \) is Turing-acceptable if we can find a TM, \( M \) with input alphabet \( \Sigma \) such that for any input \( w \in \Sigma^* \), \( M \) will halt in an accepting state if \( w \in L \), and \( M \) will halt in a non-accepting state or will fail to halt, if \( w \notin L \).

(a) \( \langle A, 00101 \rangle \rightarrow \langle B, 10101 \rangle \rightarrow \langle C, 10001 \rangle \)
\( \rightarrow \langle B, 10011 \rangle \rightarrow \langle C, 100100 \rangle \rightarrow \langle E, 100101 \rangle \).

(b) \( L(M) = 01^* + 01^* 01(01)^* + 11(01)^* \).

(c) \( n \equiv 0 \pmod{3} \Rightarrow k \pmod{3} < 0 \) which is not possible
\( n = 1 \pmod{3} \Rightarrow k \pmod{3} < 2 \) \( \Rightarrow k \equiv 0 \pmod{3} \) or \( 1 \pmod{3} \)
\( n = 2 \pmod{3} \Rightarrow k \pmod{3} < 2 \) \( \Rightarrow k \equiv 0 \pmod{3} \)
\( 1 \cdot L_1 = (aqq)^* b (bbb)^* + a(aqq)^* b (bbb)^* + (aaa)^* b (bbb)^* \).
Hence \( L_1 \) is a regular language.

(b) \( L_2 = \{ b^k c^n : 2k < n \} \). Suppose \( L_2 \) was regular. Then we can find an NFA \( M \) such that \( L(M) = L_2 \). Let \( N \) be the number of states in \( M \). Then \( b^N c^{2N+1} \in L(M) \) because \( 2(N) < 2N+1 \). Hence \( L(M) = L_2 \).

Now if we ride this loop twice we will see that \( M \) accepts \( b^2 b^j b^N \in L_2 \).

Because \( 2(N+1) < 2N+1 \). So this contradicts the fact that \( L(M) = L_2 \). Hence \( L_2 \) cannot be regular.