MAD 3512 - THEORY OF ALGORITHMS
TEST #2 - Fall 2013

Answer all 6 questions. No calculators, formula sheets or cell-phones are allowed. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each of the 6 questions on 6 separate pages.

(15) 1 (a) Find an NFA $M$ which is equivalent to the RLG $G$ given below.

\[ G: \quad \text{a} \rightarrow \text{b}, \quad \text{B} \rightarrow \text{Ob}, \quad \text{c} \rightarrow \text{1C}, \quad \text{c} \rightarrow \text{11}, \quad \text{c} \rightarrow \lambda, \quad \text{c} \rightarrow \text{1D}, \quad \text{c} \rightarrow \text{E}, \quad \text{D} \rightarrow \text{1B}, \quad \text{D} \rightarrow \lambda, \quad \text{E} \rightarrow \text{01}, \quad \text{E} \rightarrow \text{0C}. \]

(b) Suppose $F$ is a finite language and $L \cup F$ is a regular language. Is it always true that $L$ must also be a regular language? (Justify your answer.)

(15) 2 (a) Write down what the Halting Problem says.

(b) Find a regular expression for the language accepted by the NFA shown on the right.

(15) 3 (a) Define what it means for $f$ to be obtained from $g$ and $h$ by primitive recursion.

(b) Show that $f(x,y) = 3x + 2y + 1$ is a primitive recursive function by finding primitive recursive functions $g$ and $h$ such that $f = \text{prec}[g,h]$.

(20) 4 (a) Define what it means for the function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ to be obtained from the total function $g: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ by minimization.

(b) Show that the functions $f(x) = \lfloor x/3 \rfloor$ and $h(x) = x (\text{mod } 2)$ are $\mu$-recursive functions. [You may use the fact that PRED, MONUS, ADD, MULT & SIGN are primitive recursive if needed in #4, but you are not allowed to do so in #3.]

(15) 5 (a) Define what is a Turing-decidable binary relation $R$ on $\mathbb{N}$.

(b) Show what happens at each step if (i) $\lambda$, and (ii) 1, are inputs for the TM, $M$ shown on the right.

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

(a) $L_1 = \{a^k b^n : k \text{ (mod } 3) > n^2 \text{ (mod } 3) \}$

(b) $L_2 = \{b^k a^n : k > n^2 \}$.

[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.]
MAD 3512 - Theory of Algorithms  Florida Int'l Univ.
Solutions to Test #2  Fall 2013

1(a) [Diagram]

(b) YES. First observe that \( F \) is finite because \( F \subseteq E \). Now all finite languages are regular. So \( F \) will be regular. Since \( U \) is regular, \( (U F) - F \) will also be regular. But \( L = (U F) - F \cup (L F) \). Hence \( L \) is a union of regular languages and so is regular.

2(a) The Halting Problem is the question, is there a TM \( H \) such that for any TM \( M \) and any input \( w \) for \( M \), \( H \) will be in an accepting state if \( M \) halts on \( w \) & \( H \) will halt in a non-accepting state if \( M \) does not halt on \( w \)?

(b) Corresponding GFA

Eliminate D

Eliminate A

Eliminate B

\[ R_1 = aab \quad R_2 = a(aa + ad) = a(aa + d) \quad R_3 = bdb \quad R_4 = bda \]

\[ L(M) = R_1 \ast R_2 (R_4 + R_3 R_1 \ast R_2) \ast \]

\[ = (aab) \ast a(aa + d) \ast (bda + bdb(aab) \ast a(aa + d)) \ast \]
3(a) \[ f \text{ is obtained from } g \text{ by primitive recursion if} \]
\[ f(x,0) = g(x) \text{ and } f(x, s(y)) = h(x, y, f(x,y)). \] Here \( g: N \rightarrow N, \)
\( h: N^{n+2} \rightarrow N, \) \( f: N^{n+1} \rightarrow N \) and \( x = \langle x_1, \ldots, x_n \rangle. \)

\[ \begin{align*}
f(x, y) &= 3x + 2y + 1. \quad &\text{So } f(x,0) = 3x + 1 \Rightarrow g(x) = 3x + 1 \\
\text{Also } f(x, s(y)) &= 3x + 2(y+1) + 1 = (3x + 2y + 1) + 2 = f(x, y) + 2 \\
\text{So } h(x, y, f(x,y)) &= f(x, y) + 2. \quad &\therefore h = 5050 \text{ I}_{3,3}\end{align*} \]

Now \( g \) is not yet written as a primitive recursive function.

\[ g(y) = 3y + 1. \quad \text{So } g(0) = 1 \text{ and } g(s(y)) = 3(y+1) + 1 = g(y) + 3 \]

\[ \therefore g = \text{prec}[s_0, s_0 s_0 \text{ I}_{1,2}] . \quad \text{Hence } g = \text{prec}[g, h] = \text{prec}[\text{prec}[s_0, s_0 s_0 \text{ I}_{1,2}], s_0 s_0 \text{ I}_{3,3}] . \text{ Thus } f \text{ is a primitive recursive function.} \]

4(a) \[ f \text{ is obtained from } g \text{ by minimization if } x = \langle x_1, \ldots, x_n \rangle \&
\]
\[ f(x) = \{ \text{smallest } y \text{ such that } g(x,y) = 0 \}
\]
\[ \text{undefined, when } g(x,y) \geq 1 \text{ for each } y \in N. \]

(b) Let \( g(x,y) = x \div 3y. \) Then \( (\forall y)[g(x,y) = 0] = \mu y[x \div 3y = 0] = \mu y[x \equiv 3^{-1}x] = f(x). \quad \text{So } f = \mu g[0]. \]

\[ \therefore f = \mu \left[ \text{MONUS o} (\text{I}_{1,2} \land \text{MULT o} (s_0 s_0 z o \text{I}_{1,2}) \land \text{I}_{2,2}), 0 \right]. \]

\[ \therefore f \text{ is a } \mu \text{-recursive function.} \]

(c) First observe that \( h(x) = x \pmod{2} = \begin{cases} 0 & \text{if } x \text{ is even} \\ 1 & \text{if } x \text{ is odd.} \end{cases} \)

Now \( (\forall y)[x \div 2y = 0] = \mu y[x/2] = k(x), \text{ say.} \quad \text{So } k = \mu \left[ \text{MONUS o} (\text{I}_{1,2} \land \text{MULT o} (s_0 s_0 z o \text{I}_{1,2}) \land \text{I}_{2,2}), 0 \right]. \)

But \( h(x) = 2([x/2] - x) = \text{MONUS} (\text{MULT o} (s_0 (s(z(x))), k(x)), \text{I}_{1,2}) \)

\[ \therefore h = \text{MONUS o} (\text{MULT o} (s_0 s_0 z) \land k) \land \text{I}_{1,2} . \]

Since \( k \) is \( \mu \)-recursive, it follows that \( h \) is \( \mu \)-recursive.

5(a) The binary relation \( R \) on \( N \) is Turing-decidable if we can find a \( TM M \) such that for the input \( \langle \text{mem} \rangle \), \( M \) will halt in an
5(a) accepting state if \( (m,n) \in R \); and \( M \) will halt in a non-accepting state if \( (m,n) \notin R \).

5(b) (i) \( (A, w) \rightarrow (B, w u) \rightarrow (D, u) \)

(ii) \( (A, 1) \rightarrow (A, x u) \rightarrow (B, x u) \rightarrow (C, 1 u) \)

\[ \rightarrow (B, 11) \rightarrow (B, u 11) \rightarrow (D, 11) \]

6(a) \[ n \equiv 0 \pmod{3} \Rightarrow k \equiv 0 \pmod{3} > 0^2 \pmod{3} \Rightarrow k \equiv 1 \text{ or } 2 \pmod{3}, \]
\[ n \equiv 1 \pmod{3} \Rightarrow k \equiv 1 \pmod{3} > 1^2 \pmod{3} \Rightarrow k \equiv 2 \pmod{3}, \]
\[ n \equiv 2 \pmod{3} \Rightarrow k \equiv 2 \pmod{3} > 2^2 \pmod{3} = 1 \pmod{3} \Rightarrow k \equiv 2 \pmod{3}. \]

\[ \therefore L_1 = (a + aa)(aaa)^* (bbb)^* b + aa(aaa)^* (bbb)^* b \]

and so is a regular language.

(b) \( L_2 = \{ b^n a^n : k > n^2 \} \). Suppose \( L_2 \) was regular. Then we can find a \( \delta \)-free NFA \( M \) such that \( L(M) = L_2 \). Let \( N \) be the number of states in \( M \). Then \( b^{N+1} a^N \in L_2 \) because if we put \( k = N + 1 \) 

\[ \& n = N, \] we will see that \( k > n^2 \). So \( M \) will accept \( b^{N+1} a^N \).

Since it takes \( N+1 \) states to process the \( a^N \), any acceptance track of \( b^{N+1} a^N \) must contain a loop as shown below with \( j \geq 1 \).

Now if we ride this loop twice, we will see that \( M \) accepts the string \( b^{N+2} a^2 a b a a^{N-1-j} = b^{N+1} a^{N+j} \).

But \( b^{N+2} a^{N+j} \notin L_2 \) because \( N^2 + 1 \neq (N+j)^2 \). So this contradicts the fact that \( L(M) = L_2 \). Hence \( L_2 \) is non-regular.