Theory of Computation

Final Exam, January 9, 2012

1. (20 pts) True or false? No explanations needed. Score = max{0, Right - $\frac{1}{2}$ Wrong}.

   (a) If $L \subseteq 0^*$, then L is always recursive.
   $\times$.

   (b) If $L \leq_m \{0^n 1^n \mid n \geq 0\}$, then L is recursive.
   $\bigcirc$

   (c) $\{0^n 0^{2n} 0^{3n} \mid n \geq 0\}$ is context-free but not regular.
   $\times$. The language is $(000000)^*$.

   (d) $\{xwx^R \mid x, w \in \{0,1\}^+\}$ is context-free but not regular.
   $\times$ The language is $0(0+1)^+0 + 1(0+1)^+1$.

   (e) $\{wa^n a^n w^R | w \in \{a,b\}^*, n > 0\}$ is context-free but not regular.
   $\bigcirc$

   (f) $\{a^n b^{n^2} \mid n \geq 0\}$ is not context-free.
   $\bigcirc$

   (g) u is a subsequence of v if u can be obtained by dropping symbols from v, where $u, v \in \Sigma^*$. For example $11, 001$ are subsequences of 1001. If A is context-free, then $SUBSEQ(A) = \{u \mid \exists v \in A \text{ such that } u \text{ is a subsequence of } v\}$ is also context-free.
   $\bigcirc$

   (h) The language $\{< G > \mid G$ is a context-free grammar and G is ambiguous$\}$ is recursively enumerable (r.e.).
   $\bigcirc$

   (i) The language $\{< M, w > \mid M$ is a linear bounded automaton and M accepts $w\}$ is not recursive.
   $\times$

   (j) The language $L = \{< G, D > \mid G$ is a context-free grammar, D is a regular grammar, and $L(G) \subseteq L(D)\}$ is recursive.
   $\bigcirc$

2. (20 pts) For the following languages, determine whether the language is **(A) recursive, (B) recursively enumerable but not recursive, (C) not recursively enumerable.** No explanations needed. No penalty for wrong answers.

   (a) $L_1 = \{< M > |M$ is a TM and there exists an input on which M halts in less than $| < M > |$ steps$\}$
   (Sol.) A.

   (b) $L_2 = \{< M > |M$ is a TM that accepts all even numbers$\}$
   (Sol.) C.

   (c) $L_3 = \{< M > |M$ is a TM and $L(M)$ is finite$\}$.
   (Sol.) C.

(d) $L_4 = \{< M > | M$ is a TM and $L(M)$ is infinite$\}$.
(Sol.) C.

(e) $L_5 = \{< M > | M$ is a TM and $L(M)$ is an uncountably infinite set$\}$
(Sol.) A. The language is empty.

(f) $L_6 = \{< M_1, M_2 > | M_1, M_2$ are TMs and $\epsilon \in L(M_1) \cup L(M_2)\}$.
(Sol.) B

(g) $L_7 = \{< M_1, M_2 > | M_1, M_2$ are TMs and $\epsilon \in L(M_1) \cap L(M_2)\}$.
(Sol.) B.

(h) $L_8 = \{< M_1, M_2 > | M_1, M_2$ are TMs and $\epsilon \in L(M_1) - L(M_2)\}$.
(Sol.) C.

(i) $L_9 = \{< M > | M$ is a TM, and there exists an input whose length is less than 100 on which M halts$\}$
(Sol.) B.

(j) $L_{10} = \{< M > | M$ is a TM and M is the only TM that accepts $L(M)\}$
(Sol.) A. The language is empty.

3. (10 pts) Consider the following grammar:
$S \to AB \mid BC \quad A \to AB \mid a \quad B \to CC \mid b \quad C \to BA \mid b$
Apply the CYK algorithm to determine whether the string $ababa$ is generated by the grammar. You need to draw the CYK table to show the computation.
(Sol.) Refer to the textbook or class notes for the solution of a similar problem.

4. (10 pts) For a Language $L \subseteq \Sigma^*$, define $reflect(L) = \{ww^R \mid \exists w \in L\}$ where w^R denotes the reverse of the string w.

   (a) (4 pts) Consider $L_0 = \{0^i1^i \mid i \geq 0\}$ and $L_1 = 0^*1^*$. What are $reflect(L_0)$ and $reflect(L_1)$?
   Sol.) $reflect(L_0) = \{0^i1^{2i}0^i \mid i \geq 0\}$ and $reflect(L_1) = \{0^i1^{2j}0^i \mid i, j \geq 0\}$.

   (b) (6 pts) Is the class of context-free languages closed under $reflect$? Justify your answer.
   (Sol.) L_0 is context-free. However, $reflect(L_0)$ is not context-free.

5. (10 pts) Prove that the language $L = \{a^ib^jc^k \mid j < i, j < k\}$ is NOT context-free.

Proof. Expecting a contradiction, assume L_1 is a CFL and let $p \in \mathbb{N}_1$ be its pumping length. Define $s = a^{p+1}b^pc^{p+1}$. Let $uvxyz = s$ be a partitioning such that $|vxy| \leq p$ and $|vy| \geq 1$. Since $|vxy| \leq p$, vy cannot have a's, b's, and c's. Since $|vy| \geq 1$, it follows that vy does not have equal numbers of a's, b's, and c's. With this ruled out, there are only two other possibilities:

Case 1: If vy has more a's or more c's than b's, then choose $s' = uxz$ and observe that s' has at most as many a's or at most as many c's as b's (because more a's or more c's than b's were removed).

Case 2: If vy has more b's than a's or than c's, then choose $s' = uv^2xy^2z$ and observe that s' has at least as many b's as a's or c's (because more b's were added than a's or c's).

In both cases $s' \notin L_1$, contradicting the pumping lemma. We conclude that L_1 is not context-free. □

6. (10 pts) For any language $L \subseteq \Sigma^*$, and any $u \in \Sigma^*$, let $u/L = \{v \in \Sigma^* | uv \in L\}$. Prove that if L is context-free, then u/L is also context-free for every $u \in \Sigma^*$.
(Hint: Let $u = a_1 \cdots a_k$, for some k, and let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA accepting L. Construct a PDA M' to accept u/L.)
(Sol.) Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be an FA accepting only the word u. Construct the following PDA $M_2 = (Q_2, \Sigma, \Gamma, \delta_2, q_0^2, Z_0, F_2)$, where

   - $Q_2 = Q \times Q_1$, and $q_0^2 = (q_0, q_1)$
   - δ_2 is the following: if $(q, Y) \in \delta(p, a, X)$, and $q' \in \delta_1(p', a)$, then $((q, q'), Y) \in \delta_2((p, p'), \epsilon, X)$.
   - $F = \emptyset$.

   In words, M_2 is the "product automaton" of M and M_1 except that the input symbol is suppressed (i.e., symbol a is replaced by ϵ)

   Now we construct a PDA M_3 by cascading M_2 and M using the following additional transitions: for every $(q, q') \in Q \times F_1$, add a transition $(q, q') \overset{\epsilon, X/X}{\rightarrow} q$.

   Intuitively, what M_3 does is that it runs M_2 first until some $(q, q') \in Q \times F_1$ is reached (meaning that along this computation, the prefix is u in the corresponding computation of M.) Then M_2 goes directly from (q, q') to q using an ϵ transition while keeping the contents of the stack intact. Following that, M resumes its computation from state q.

   The correctness of the above construction is easy to see.

7. (4 pts) Define *context-sensitive grammars.*
(Sol.)

   - (Definition 1) A grammar is context-sensitive if all productions are of the form $x \rightarrow y$, where x, y are in $(V \cup T)^+$ and $|x| \leq |y|$.

     or

   - (Definition 2) A grammar is context-sensitive if all productions are of the form $xAy \rightarrow xvy$, where x, v, y are in $(V \cup T)^*$, $A \in V$.

8. (10 pts) Let $A, B \subseteq \{0,1\}^*$ be r.e. languages such that $A \cup B = \{0,1\}^*$ and $A \cap B \neq \emptyset$. Prove that $A \leq_m (A \cap B)$, where $\leq_m$ denotes the many-one reduction.
(Hint: Let M_1 be a TM accepting A and M_2 be a TM accepting B. Further, since we know that $A \cap B \neq \emptyset$, we know that some string y will be accepted by both M_1 and M_2. Construct a mapping f that witnesses $\leq_m$, i.e., $x \in A \Leftrightarrow f(x) \in A \cap B$.)
(Sol.) Consider the following mapping $f(x)$:

Given an x, runM_1 and M_2 in parallel (using the technique discussed in class)
(Case1) if M_1 accepts first, then since Σ^* and N are both countably infinite,
the set $\{(y, i) \mid y \in \Sigma^*, i \in N\}$ is also countably infinite.
For every (y, i) pair in an enumeration sequence of $\{(y, i) \mid y \in \Sigma^*, i \in N\}$
run M_1 and M_2 on input y for i steps until finding a z such that both M_1 and M_2 accept.
(Note that since $A \cap B \neq \emptyset$, such a z must exist)
then output z; i.e., $f(x) = z$.

(Case 2) if M_2 accepts first, then output x; i.e., $f(x) = x$.

Note that since $A \cup B = \{0,1\}^*$, one of Case 1 or Case 2 must hold. In case 2, it is easy to see that $x \in A$ iff $f(x) = x \in A \cap B$, since x is known to be in $B = L(M_2)$.

9. (6 pts) Suppose A is recursively enumerable and $A \leq_m \bar{A}$. Prove that A is recursive.
(Sol.) First we claim that if $A \leq_m B$, then $\bar{A} \leq_m \bar{B}$. To see this, let g be a mapping that witnesses $A \leq_m B$. Then g also witnesses $\bar{A} \leq_m \bar{B}$.

Now if $A \leq_m \bar{A}$, then according to the above $\bar{A} \leq_m \bar{\bar{A}} = A$. Hence, if A is r.e., so is $\bar{A}$. Therefore, both A and $\bar{A}$ are recursive.