Theory of Computation

Fall 2013, Final Exam.

Date: January 6, 2014

(I) (40 pts) True or False? Score = max{0, Right - $\frac{1}{2}$ Wrong}.

(1) Let $L = \{a^i \mid i \text{ is a prime number } \}(\subseteq \{a\}^*)$. The complement of L, i.e., $a^* - L$, is context-free but not regular.
False

(2) The language $\{a^i b^j c^k \mid i, j, k \geq 0,\ j = max\{i, k\}\}$ is not context-free.
True

(3) The complement of the language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.
True

(4) If a language L and its complement $\bar{L}$ are both context-free, then L must be regular.
False

(5) $L = \{\langle M \rangle \mid L(M) \text{ is infinite and } M \text{ is a PDA } \}$ is recursive.
True

(6) $\{w\#w^R \mid w \in \{0,1\}^*\}$ can be accepted by a two-tape deterministic Turing machine (DTM) in $O(n)$ time.
True

(7) Consider a type of automata each of which is a deterministic PDA augmented with a counter, i.e., a deterministic machine with a pushdown stack and a counter. Such automata are Turing-equivalent.
True

(8) $L = \{\langle M \rangle \mid M$ is a TM that write a blank symbol over a nonblank symbol when it runs on some input$\}$ is not recursive
True

(9) $L = \{\langle M, p, x \rangle \mid$ on input x, TM M never enters state $p\}$ is not recursive.
True

(10) $\{\langle M, w \rangle \mid$ TM M accepts word $w\} \leq_m 0^*1^*$, where $\leq_m$ denotes many-one reduction.
False

(11) $\{a^n b^n \mid n \geq 0\} \leq_m^p \{a^n b^m \mid n, m \geq 0\}$, where $\leq_m^p$ denotes polynomial-time many-one reduction.
True

(12) $\{a^n b^n \mid n \geq 0\} \leq_m \{a^n b^m \mid n, m \geq 0\}$.
True

(13) There is an undecidable subset of 0^*.
True

(14) The class NP (nondeterministic polynomial time) is closed under union.
True

(15) A 2-way PDA is a PDA whose input head can more in both directions. 2-way deterministic PDA accept only context-free languages.
False

(16) It a function f is computable by a Turing machine that always halts, then f is primitive recursive.
False

(17) $\{a^n b^n c^n d^n \mid n \geq 0\}$ is in P (deterministic polynomial time).
True

(18) Given a context-free language L and a regular language R, the problem of deciding whether "$L = R$?" is undecidable.
True

(19) The halting problem of TMs is NP-hard.
True

(20) It is known that $DSPACE(\log n) \overset{\subset}{\neq} DSPACE(\log^2 n)$. ($\overset{\subset}{\neq}$ means "is a proper subset of".)
True

(II) (15 pts) Let G be the following CFG in Chomsky Normal Form, where S is the start symbol:
$S \to AB \mid CA \mid a$
$A \to BC \mid a \mid b$
$B \to CC \mid c$
$C \to AC \mid b$
What follows is a partially filled table $T(i,j)$ based on the CYK parsing algorithm to decide whether a string x is in $L(G)$. $T(i,j)$ is the set of nonterminals that can derive $a_i \cdots a_j$ for an input string $x = a_1 a_2 \cdots a_n$, where $1 \le i \le j \le n$.

	$T(i,j)$	j = 1	2	3	4	5
	1	B	A	$\emptyset$	?	?
	2	–	A, C	S	B, C	?
i	3	–	–	A, S	C	B, C, S
	4	–	–	–	A, C	B, C, S
	5	–	–	–	–	A, C

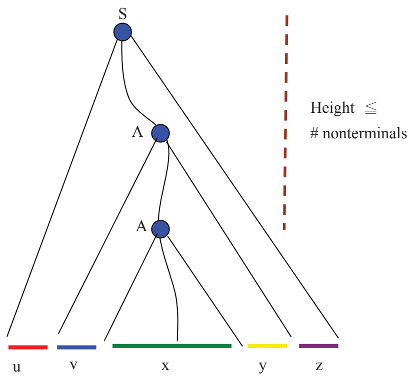
(a) (3 pts) What is the string x for this table? Why?
Sol: *cbabb*

(b) (9 pts) What are the three missing entries $T(1,4), T(2,5)$ and $T(1,5)$? Why?
Sol: $T(1,4) = A, C, T(2,5) = A, B, C, S$ **and** $T(1,5) = A, B, C, S$

(c) (3 pts) Is $x \in L(G)$? Why?
Sol: Yes

(III) (15 pts) A context-free grammar is said to be *linear* if its productions are of the form $A \to \epsilon$, $A \to a$, $A \to aB$, $A \to Bb$, or $A \to aBb$, where $a, b \in \Sigma$.

(a) (8 pts) Prove the following pumping lemma for linear languages:
Let L be a linear language. Then there exists a positive integer n such that for every $w \in L$ with $|w| \ge n$, w can be decomposed into $w = uvxyz$ such that (1) $|uvyz| \le n$, (2) $|vy| \ge 1$ and (3) $uv^ixy^iz \in L$, for all $i \ge 0$. (Note that x can be of arbitrary length.)
Proof



(b) (7 pts) Use the above pumping lemma to show that linear languages are not closed under concatenation, i.e., given linear languages A and B, $A \cdot B$ need not be linear. (Hint: consider $\{0^n1^n \mid n \ge 1\}$.)
Proof
Consider $\{0^n1^n \mid n \ge 1\} \cdot \{0^n1^n \mid n \ge 1\} = \{0^n1^n0^m1^m \mid n, m \ge 1\}$. The rest is easy.

(IV) (5 pts) Let $A \subseteq \Sigma^*$ be a recursive language, and $B = \{x \mid \exists w \in \Sigma^*, xw \in A\}$, i.e., B consists of all strings that are prefixes of strings in A. Prove that B is recursively enumerable.
Proof
Let $s_1, s_2, ...$ be all strings in Σ^*. On input x, for $i = 1, 2, ...$, run M on xs_i, if M accepts then accept; otherwise go to the next i.

(V) (5 pts) Is the following statement true? Justify your answer.
Given a recursively enumerable language A, if $A \le_m \bar{A}$, then A is recursive. (Here $\bar{A} = \Sigma^* - A$.)
Sol: Yes
Note that $A \le_m \bar{A} \Rightarrow \bar{A} \le_m A$; hence $\bar{A}$ is r.e. A and $\bar{A}$ being r.e. imply A is recursive.

(VI) (10 pts) Let $B = \{\langle M \rangle \mid M$ is a TM, $L(M) = \{\epsilon\}\}$. Suppose you want to show that $\{\langle M, w\rangle \mid M$ is a TM, M accepts $w\} \leq_m B$ using a reduction f that maps $\langle M, w\rangle$ to M_1.

(a) Fill in the blanks (i.e., $A1$ and $A2$) in the following two statements in a way that states what you have to do to make the reduction work.

- If M accepts w, then ....$\mathbf{L(M_1)} = \{\epsilon\}$.........
- If M does not accept w, then ....$\mathbf{L(M_1)} \neq \{\epsilon\}$.........

(b) Given M and w, give the definition of the desired TM M_1 by filling in the blanks (i.e., $A3 - A5$) in the following

$M_1 =$ "On input x

If $x \neq \epsilon$, ...**Reject**.........

If $x = \epsilon$, simulate M on w

If M accepts w, ...**Accept**.........

If M rejects w, ...**Reject**........."

(VII) (10 pts) Classify the following languages into **(1)** ***recursive*****, (2)** ***recursively enumerable*** **(i.e.,** *r.e.***) but not recursive, (3)** ***co-r.e.*** **but not recursive,** and **(4) none of the above.** Proofs are not needed.

(a) $L_1 = \{\langle M \rangle \mid M$ is a Turing Machine that accepts some string in $0^*\}$.
Sol: (2) This language is undecidable by Rice's theorem. It is r.e.. A nondeterministic TM can guess a string w in 0^* and then run M on it.

(b) $L_2 = \{\langle M \rangle \mid M$ is a Turing Machine that does not accept any palindrome $\}$. Recall that a palindrome is a word w such that $w = w^R$ (w^R is the reversal of w).
Sol: (3) This language is undecidable by Rice's theorem. It is co-r.e.. A nondeterministic TM can guess a palindrome accepted by M and verify it by running M on it. So if $M \in \bar{L}2$, it can be accepted by the NTM. It is not r.e. because if it were, it would be decidable.

(c) $L_3 = \{\langle M \rangle \mid M$ is a Turing Machine that terminates on some input in 1000 steps$\}$.
Sol: (1) This is decidable. Since M has to terminate in 1000 steps, at most 1000 symbols of the input strings are relevant. So one could try them all exhaustively and run 1000 steps of M to see if it terminates.

(d) $L_4 = \{\langle M \rangle \mid M$ is a Turing Machine that accepts 1 and does not accept 0$\}$.
Sol: (4) Let $L_u = \{\langle M, w\rangle \mid M$ accepts $w\}$. We first show $\bar{L_u} \leq_m L_4$ by the following

```
On input M,w:
    Return the following machine:
            R: on input x
               if x = 1 then accept
               else Run M on w
                    if M accepts then accept.
```

We now show $\bar{L_u} \leq_m \bar{L_4}$ by the following

```
On input M,w:
    Return the following machine:
            R: on input x
               if x = 0 then reject
               else Run M on w
                    if M accepts then accept.
```

Since $\bar{L_u}$ is not r.e., the answer follows.

(e) $L_5 = \{\langle M, w\rangle \mid M$ is a linear bounded automaton that does not halt on $w\}$.
Sol: (1) L_5 is decidable. Note that a linear-bounded automata can be in at most qng^n possible configurations, where $|w| = n, g$ is the size of the tape alphabet and q is the number of states. We can decide L5 using a TM that simulates the LBA. If the LBA accepts or rejects within qng^n steps, then the TM halts and rejects, since M, w is not in L_5. Otherwise TM accepts M, w, since M does not halt on w. Since the TM always halts and correctly decides the question, L5 is decidable.