Theory of Computation

Fall 2014, Final Exam. (Solutions)

Date: January 12, 2015

---

(1) (10 pts) Given two languages A and B, define $A \diamond B = \{xy \mid x \in A, y \in B, |x| = |y|\}$. Suppose A and B are regular, is $A \diamond B$ always context-free? Justify your answer.
Solution: Yes.
Proof Idea: A word z is in $A \diamond B$, iff $z = xy$ where $x \in A$ and $y \in B$, $|x| = |y|$. Let M_A and M_B be the two FA accepting A and B, resp. Design a PDA M doing the following:

- Read (nondeterminically) x and use M_A to decide $x \in A$, and each time a symbol is read, push a symbol onto M's stack
- Read the rest of the string y and use M_B to decide $y \in B$, and each time a symbol is read, pop a symbol from M's stack
- Accept if both M_A and M_B accept and the stack is empty at the end.

(2) (10 pts) Prove that the language $L = \{\langle M, x, \#^t\rangle \mid$ nondeterministic Turing machine M accepts x within t steps on at least one branch $\}$ is NP-complete.
Solution:
(in NP) To show L is in NP, we need to show the existence of a TM M' accepting L in polynomial time. Basically M' plays the role of M except that it treats the middle part x as its input, and M' uses $\#^t$ to keep track of the number of steps it performs. (This can be easily done by changing a # to a @ each time a step is carried out.) M' accepts $\langle M, x, \#^t\rangle$ in polynomial time iff M accepts x in t time.

(NP-hard) Let M be a polynomial time NTM with time bound $p(n)$. Let $L_{M,p(n)}$ be the language accepted by such a TM. Consider the following reduction from $L_{M,p(n)}$ to $L = \{\langle M, x, \#^{p(n)}\rangle \mid$ nondeterministic Turing machine M accepts x within t steps on at least one branch $\}$. Clearly $x \in L_{M,p(n)}$ iff $\langle M, x, \#^{p(n)}\rangle \in L$.

(3) (10 pts) Prove that the class NP is closed under the star operation.
Solution: Let L, accepted by an NTM M with time bound $p(n)$, be a language in NP. We design the following NTM M' to accept language L^*.

- Nondeterministically guess a partition $y_1 y_2 \cdots y_k$ $(=x)$.
- Run M to check, for every $i = 1, ..., k$, $y_i \in L$, which can be done in $p(|y_i|)$ time.
- Accept if all the checking succeeds.

Clearly M' runs in polynomial time.

(4) (10 pts) Is it true that if $A \leq_m B$ and B is context-free, then A is a recursive language? Why? (Here $\leq_m$ denote the many-one reduction.)
Solution: Yes.
Let M be a PDA accepting B. To tell whether $x \in A$, first apply the computable mapping f (guaranteed by the many-one reduction) to yield $f(x)$, and let M decide whether $f(x) \in B$. As the membership problem for context-free languages is decidable, A is therefore recursive.

(5) (10 pts) Prove that the following language is NOT decidable. Do not use Rice's theorem in your proof.

$$L = \{\langle M\rangle \mid M \text{ is a Turing machine and } M \text{ accepts } 1011\ \}.$$

Solution: We reduce A_{TM} to L. Given a $\langle M, x\rangle$, construct the following TM M':
On input y, simulate M on x. If M accepts x, then accept if $y = 1011$.
Clearly, $\langle M, x\rangle \in A_{TM}$ iff $\langle M\rangle \in L$

(6) (10 pts) Prove that every infinite Turing-recognizable language has an infinite Turing-decidable subset. (Hint: Think of Turing machines as enumerators.)
Solution: It is known that a language is r.e., (resp., recursive) iff there exists an enumerator generating strings (resp., strings in lexicographical order) in the language. Let M be an enumerator generating the infinite r.e. language. We construct the following enumerator M':

(a) M generates a new string x.

(b) If there is a string y already in the output tape of M' with $y > x$ (according to lexicographical order), then discard x

(c) Otherwise, write x to the output tape.

(d) Repeat the above three steps.

It is not hard to see that M' generates an infinite subset of the given r.e. language in lexicographical order.

(7) (10 pts) Given a context-free grammar G, is it decidable whether $L(G)$ is infinite? Justify your answer formally.
Solution: Let n be the pumping constant associated with $L(G)$. We can show that $L(G)$ is infinite iff there is a word $x \in L(G)$ such that $n \leq |x| \leq 2n$. Hence, by testing all words of lengths between n and $2n$, the problem is decidable as the membership problem for CFL is decidable.

(8) (10 pts) Prove that the following language is undecidable

$$L = \{\langle G \rangle \mid G \text{ is an ambiguous context-free grammar}\}$$

(Hint: Use PCP. Recall that a CFG G is ambiguous if there is a word x with two derivation trees in G.)
Solution: Let $P = \{(x_1, y_1), ..., (x_n, y_n)\}$ be an instance of PCP. We construct the following grammar G with rules
$S \rightarrow S_x \mid S_y$
$S_x \rightarrow x_1 S_x a_1 \mid ... \mid x_n S_x a_n \mid x_1 a_1 \mid ... \mid x_n a_n$
$S_y \rightarrow y_1 S_y a_1 \mid ... \mid y_n S_y a_n \mid y_1 a_1 \mid ... \mid y_n a_n$
where $a_1, ..., a_n$ are new symbols. It is not hard to see that PCP has a match iff G generates some word in an ambiguous way.

(9) (10 pts) Let $L = \{w \mid \text{either } w = 0x \text{ for some } x \in A_{TM} \text{ or } w = 1y, \text{ for some } y \in \overline{A_{TM}}\}$. Prove that neither L nor $\overline{L}$ is recursively enumerable. (Recall that $A_{TM} = \{\langle M, z \rangle \mid M \text{ is a Turing machine that accepts input } z\}$.)
Solution: It is known that $\overline{A_{TM}}$ is not r.e. To show neither L nor $\overline{L}$ is r.e., it suffices to establish $\overline{A_{TM}} <_m L$ (in this case, L is non-r.e.) and $\overline{A_{TM}} <_m \overline{L}$ (in this case, $\overline{L}$ is non-r.e.)

To show $\overline{A_{TM}} <_m L$, consider the mapping $f(y) = 1y$. Clearly, $y \in \overline{A_{TM}}$ iff $f(y) \in L$.

To show $\overline{A_{TM}} <_m \overline{L}$, we show $A_{TM} <_m L$. Consider the mapping $f(x) = 0x$. Clearly, $x \in A_{TM}$ iff $f(x) \in L$.

(10) (10 pts) Suppose $C_0, C_1, ...C_n, ..$ is an infinite sequence of classes of languages such that $\forall i \geq 0, C_i \subseteq NP$, $C_i \overset{\subset}{\neq} C_{i+1}$ and C_i is closed under $\leq_p^m$ (i.e., if $L_1 \leq_p^m L_2$ and $L_2 \in C_i$, then $L_1 \in C_i$). Define $C = \bigcup_{i \geq 0} C_i$. Prove that the class of language C does not contain any NP-complete language w.r.t. $\leq_p^m$ (the polynomial-time many-one reduction).
Solution: Suppose, in contrast, that C contains a complete language L. Then $L \in C_i$ for some $i \geq 0$, and $\forall$ $L' \in C, L' \leq_p^m L$. Now consider $C_{i+1} (\subset C)$. If $L' \in C_{i+1}$, then $L' \leq_p^m L$ implying that $L' \in C_i$, contradicting $C_i \overset{\subset}{\neq} C_{i+1}$.