COT 6315/CIS 4930 (Fall05,Sitharam)

Sample test 4 and very brief solutions : for detailed solutions, especially proofs, visit TA's office hours.

1. Classify the following languages as being: (i) neither r.e nor co-r.e, (ii) r.e or co-r.e (indicate which), but not both; (iii) both r.e. and co-r.e but not in PSPACE (iv) In PSPACE, but NP-hard (v) in NP but not in Logspace (assuming NP $\neq$ Logspace – do not assume NP $\neq$ P) (vi) in P In all cases, fully justify your answer (the reductions, hierarchy theorems, relating space and time complexity classes etc. would be useful).

   (a) $\{(M, z)$, M an (always halting) algorithm taking 3 inputs, $z \in N$ $: \exists x \in N \forall y \in N \; M accepts(x, y, z)\}$

   Answer: (i) – both HALT and complement of HALT are reducible to this set using $\leq_m^{rec}$-reductions. So what?

   (b) (choose one)

   $\{F$ CNF Boolean formula : strictly greater than 1/2 of all possible truth assignments to the variables of F in fact satisfy $F\}$

   Answer: (iv) – This is a classical example of a set in the class PP (mentioned briefly in class). Why? It is NP-hard since SAT is $\leq_m^P$-reducible to it. How?

   OR

   A set L that is complete for $DTIME(2^{n^{\log n}})$ under the usual polynomial time many-one ($\leq_m^P$) reductions.

   Answer: (iii) – clearly has an algorithm, hence in both r.e and co-r.e (why?); not in PSPACE because if it were, then (through application of reductions and relations between space and time complexity classes – how?), it would follow that DTIME($2^{n^{\log n}}$) = EXPTIME, which is not possible by the basic time-hierarchy theorem (which one?).

   (c) (choose one) Let M be a (deterministic) algorithm that takes inputs (x, y) and runs in time $|x|^k$. Let $c \leq |x|^k$. Let M be related to a set X in the following way.

   $x \in X \Rightarrow$ for all except c of the y's (don't know which ones) with $|y| \leq |x|^k$, $M(x, y)$ accepts

   $x \notin X \Rightarrow$ for all of the y's with $|y| \leq |x|^k$, $M(x, y)$ rejects. Classify the set X.

   Answer: (vi) – run M on some (arbitrarily chosen) combination of $c + 1$ y's. If atleast $M(x, y)$ accepts on atleast one of these y's, then accept. This recognizes X correctly, and is a polytime procedure. Why?

   OR

   SAT

Answer: (v) since the Cook-Levin reduction in the book is infact a stronger, $\leq_m^{Logspace}$-reduction. (So what)? Aside: the answer (v) would hold even if P = NP, provided NP $\neq$ Logspace (to avoid confusion, note that by basic space-time relations Logspace $\subseteq$ P).

2. Classify the following (classes of) languages as being: (i) (contained) in DSPACE(n^2) but not (contained) in the class of context free languages, (ii) (contained) in L(ogspace) but not (contained) in the class of context free languages (iii) (contained) in the class of context free languages but not (contained) in the class of regular languges (iv) (contained) in the class of regular languages. In all cases fully justify your answer. To show a class of languages C is contained in a complexity class D, you need to show that every language in C is in D. To show a clss of languages C is not contained in a complexity class D, sufficient to show that there is atleast one language in C that is not in D.

   (a) $ALLEQUAL = \{a^n b^n c^n d^n : n \in N\}$
   Answer: (ii); Simple Logspace algorithm (which?); CF Pumping lemma.

   (b) $L_1 \cap$ Flip(L_2), where L_1 is context free and L_2 is regular. Here the Flip operation is defined as: Flip(L) =

   $$\{xy : x, y \in \{0,1\}^* \ \wedge yx \in L\}$$

   Answer (iii). Use fact that context free languages are closed under intersection with regular languages and show that regular languages are closed under Flip. Why? Use $L_2 = \Sigma^*$ to show not regular part. How?

   (c) Shuffle(L_1,L_2), where L_1 and L_2 are context free. Here the Shuffle operation informally shuffles the symbols of a string in L_1 with the symbols of a string in L_2, any which way, but staying in sequence, as though each string were a deck of cards. Formally, Shuffle is defined as follows. Shuffle(L_1,L_2) =

   $$\{z_1 \ldots z_m : z_i \in \{0,1\} \wedge \exists S = \{s_1, \ldots, s_k\}, T = \{t_1, \ldots, t_{m-k}\}, S \cap T = \emptyset$$

   $$\wedge \ \ S \cup T = \{1, \ldots, m\} \ \wedge \ z_{s_1} \ldots z_{s_k} \in L_1 \wedge \ z_{t_1} \ldots z_{t_k} \in L_2\}$$

   Answer: (i) – show it is in NSPACE(n) and use nondeterministic to deterministic space relationship. For this, one observation needed is that size of stack used by PDA is atmost a constant factor more than number of symbols in input string. Based on this, give an

NSPACE(n) algorithm to recognize Shuffle(L1,L2), given PDAs for L1 and L2. To show CFLS are not closed under Shuffle should be stfwd using appropriately the pumping lemma and appropriately chosen string. How?