Honors Theory, Spring 2002, Yap
Homework 6

This is due on Wed April 17.
All reducibility concepts assume Karp-reducibility.

1. (15 Points) The proof of Cook's theorem in Chapter 3 reduces any language $A \in NP$ to SAT. If M is an NP-machine for A, for any input w, the proof constructs a 3CNF formula F_w such that F_w is satisfiable iff $w \in A$. We claim that something stronger is true: $\#(F_w)$ is equal to the number of accepting computations of M on input w. Actually, this claim needs a mild condition on M. What is it? Prove this claim under this mild condition. **Hint:** if you do not see what this condition might be, we suggest the strategy of ignoring it at first, and trying to prove the claim unconditionally.

   SOLUTION: We know from the proof of Cook's theorem that an accepting computation path for w determines an assignment that satisfies F_w. To show that $\#(F_w)$ is equal to the number of accepting computation paths of M on input w, we only need to show that any two different accepting computation paths on w determine two different satisfiable assignments for F_w, and vice versa.

   But observe that distinct computation paths correspond to distinct sequences of instruction execution (this is more general than just having distinct sequences of configurations). But the sequence of instructions are encoded by the Boolean variables $I(j,t)$ for the appropriate values of j and t.

   What is the "mild condition"? Notice that if there are accepting paths that are longer than $p(n)$ (where $n = |w|$), then these will not be counted among the number of distinct satisfying assignments to F_w. Therefore, we require that every accepting computation path has length $\leq p(n)$.

2. (20 Points) Prove that QBF is $PSPACE$-complete. Since this proof can be found in the literature, I will enforce some originality in your solution by asking you to use the same framework as my description of Cook's theorem in Chapter 3. The additional idea you need comes from the key idea in the proof of Savage's theorem: if $C \vdash^{2m} C'$ (i.e., there is an $2m$-step path from C to C' then $C \vdash^m C''$ and $C'' \vdash^m C'$ for some C''.

   SOLUTION: First we show QBF is in $PSPACE$. We may assume that the input formula is in prenex normal form, with no free variables. The following is linear space algorithm for accepting QBF: let ϕ be the input formula.

   1. If ϕ contains no quantifiers, then it is a constant formula with no variables. We just evaluate it and accept if it is true; otherwise reject.

   2. If ϕ is $\exists x\psi$, recursively call M on ψ, first substitute x with 0, if the result is accept then accept, else substitute x with 1, accept if the result is accept; otherwise reject.

   3. If ϕ is $\forall x\psi$, recursively call M on ψ, first substitute x with 0, if the result is reject then reject, else substitute x with 1, reject if the result is reject; otherwise accept.

   Next, we show that QBF is $PSPACE$ hard. For any language $A \in DSPACE(n^k)$, we reduce it to QBF as follows. We use the same framework as the proof of Cook's theorem. For any string w, it is in language A iff there is a computation path of length $t(n) = O(1)^{n^k}$, from the initial configuration C_1 to an accepting configuration C_a. Let the predicate $PATH_m(C, C')$ be true if there is a computation sequence from C to C' of length $\leq m$. We want a Boolean formula F_w that is true iff $PATH_{t(n)}(C_0, C_a)$ where C_0, C_a are the initial and accepting configurations (which we may assume are uniquely represented).

   The problem is, $t(n)$ is exponential so we cannot afford to explicitly describe all the $t(n)$ configurations as in Cook's proof. We use the idea of Savitch's theorem to fix this to: thus $PATH_m(C, C')$ is true iff

$$(\exists C'')(\forall x, x')[(x = C \wedge x' = C'').\vee.(x = C'' \wedge x' = C').\Rightarrow .PATH_{m/2}(x, x')] \tag{1}$$

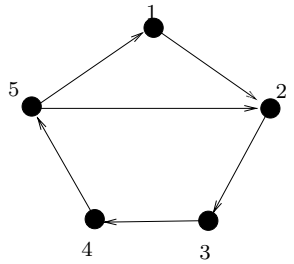
   Here, C'', x, x' represents configurations using space n^k. So we need to introduce $O(n^k)$ variables, as in Cook's proof to represent them.

BASE CASE: In case $m = 2$, the subexpression $PATH_{m/2}(x, x')$ in (1) can be directly replaced by a polynomial size Boolean formula (unquantified) that says that $x \vdash x'$ according to the rules of the Turing acceptor of the set A. This is analogous to Cook's proof.

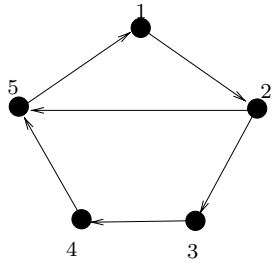
INDUCTION: If $m > 2$, we recursively replace $PATH_{m/2}(x, x')$ by further expansions of (1). The number of expansions needed is $t(n) = n^k$. Hence the final quantified Boolean formula is our desired F_w, and it has polynomial size.

It is routine (but tedious) to construct a transducer which, given w will output F_w. Then $w \in A$ iff $F_w \in$QBF, proving that A is Karp-reducible to QBF.

3. (20 Points) This exercise helps you gain some facility with the group theoretic ideas in the NONISO $\in$ *IP* proof. Let $V = V_n = \{1, \ldots, n\}$ and S_n be the set of permutations on V_n. The trivial permutation is denoted $\mathbf{1}_n$ (or simply $\mathbf{1}$). Write the composition of $\sigma, \sigma' \in S_n$ in the form of a product $\sigma\sigma'$, instead of $\sigma \circ \sigma'$.
(i) Let $2 \leq k \leq n$. If $\{a_1, \ldots, a_k\} \subseteq \binom{V}{k}$, then $(a_1, \ldots, a_k) \in S_n$ denotes the permutation which takes each a_i to $a_{(i+1 \bmod k)}$, called a **cyclic permutation**. Two special cases are $k = 2$ or $k = n$. Then $(a_1, \ldots, a_k)$ is **transpose** or a **Hamiltonian permutation**, respectively. Two cyclic permutations $(a_1, \ldots, a_k)$, $(b_1, \ldots, b_\ell)$ are disjoint if $a_i \neq b_j$ for all i, j. For instance, $(132)(45) = (45)(132)$ is a product of two disjoint cycles. The order of writing disjoint products does not matter. Show that every non-trivial permutation is a product of disjoint permutations.



(a) G_0



(a) G_1

Figure 1: Two labeled digraphs G_0, G_1.

(ii) Let G_0 be the digraph shown in Figure 1. Determine $\texttt{iso}(G_0)$ and $\texttt{iso}(G_1)$. What the sizes of these two sets?
(iii) Determine $\texttt{aut}(G_0)$ and $\texttt{aut}(G_1)$. What the sizes of these two sets?

SOLUTION:

(i) Let $\pi : V_n \to V_n$ be any permutation. First for any i that $\pi(i) \neq i$, we construct a cyclic permutation that includes i as follows: starting from i, consider $\pi(i)$, then $\pi^2(i)$, etc. This produces a sequence

$$C_1 = (i, \pi(i), \pi^2(i), \ldots, \pi^k(i)) \tag{2}$$

where we stop when $\pi^{k+1}(i)$ is equal to some previously encountered element $\pi^j(i)$ $(j = 0, 1, \ldots, k)$ in this sequence. We claim that $j = 0$ (and this means the C in (2) is a cycle). If $j > 0$ this means that $\pi(\pi^k(i)) = \pi(\pi^{j-1}(i))$, since both are equal to $\pi^j(i)$. But π is a bijection, this means $\pi^k(i) = \pi^{j-1}(i)$, contradiction since we said k is the first time that we had a repeat.

We can pick another i that does not occur in C_1 such that $\pi(i) \neq i$, and construct another cycle C_2. It is easy to see that C_1, C_2 are disjoint. We continue this until there are no more such i's. If there are m such cycles, we see that

$$\pi = C_1 C_2 \cdots C_m$$

for some $m \geq 1$.

(ii)

$$\forall G \in g_n, |iso(G)| \times |aut(G)| = n!$$

So $|iso(G_0)| = |iso(G_1)| = 5!/1 = 120$.

(iii) Automorphism preserves in-degree and out-degree of vertex. In graph G_0, only vertice 2 has in-degree 2 and out-degree 1, and only vertice 5 has in-degree 1 and out-degree 2, so these two vertices should be the same after permutation. Vertice 1 is the only one connects vertice 2 and 5, thus should be the same. Vertice 4 is the only one has an edge outgoing to vertice 5, thus should be the same. Hence the only permutation allowed is trivial. The same analysis for graph G_1.

So $|aut(G_0)| = |aut(G_1)| = 1$.