Mid Semester Examination, Autumn 2003-04

Total points: 40 | February 22, 2004 | Total time: $2 + \epsilon$ hours

---

Prove or disprove any eight of the following assertions. Give brief (but clear) justifications. No credits will be given for incorrect/incomplete/missing explanations, even though the truth value is correctly assigned. **(5×8)**

You may make use of the assumption that NP $\neq$ coNP, if necessary. Clearly indicate where you require this assumption.

1. If A is NP-complete and log-space reducible to B, then B is also NP-complete.

2. An $O(n^k)$ reduction algorithm from A to B followed by a deterministic $O(n^k)$ algorithm for B yields an $O(n^k)$ deterministic algorithm for A. (Here n is the input size, and k is a positive integer constant > 1.)

3. The language BIGCYCLE $:= \{\langle G\rangle \mid G$ is a directed graph having a cycle of length $\geqslant \lfloor n(G)/2\rfloor\}$ is NP-complete. (Here $n(G)$ denotes the number of vertices in G, and $\lfloor\ \rfloor$ the floor function.)

4. The language SMALLCYCLE $:= \{\langle G\rangle \mid$ The longest cycle in the directed graph G is of length $\leqslant \lfloor n(G)/2\rfloor\}$ is NP-complete.

5. The language HALFCYCLE $:= \{\langle G\rangle \mid$ The longest cycle in the directed graph G is of length $\lfloor n(G)/2\rfloor\}$ is decided by the following (poly-time) nondeterministic algorithm:

1. Compute $m := \lfloor n(G)/2\rfloor$.
2. Nondeterministically select m vertices $u_1, \ldots, u_m$ of G.
3. If $u_1, \ldots, u_m$ (in that order) do not constitute a cycle, *reject*.
4. Nondeterministically generate an integer k in the range $m < k \leqslant n(G)$.
5. Nondeterministically select k vertices $v_1, \ldots, v_k$ of G.
6. If $v_1, \ldots, v_k$ (in that order) constitute a cycle, *reject*, else *accept*.

6. The language GRAPHISO $:= \{\langle G_1, G_2\rangle \mid$ The undirected graphs G_1 and G_2 are isomorphic$\}$ is decided by the following (poly-time) nondeterministic algorithm:

1. If G_1 and G_2 contain different numbers of vertices, *reject*.
2. If each of G_1 and G_2 contains only one vertex, *accept*.
3. Nondeterministically select vertices u of G_1 and v of G_2.
4. If the degree of u in G_1 is different from the degree of v in G_2, *reject*.
5. Delete u from G_1 and v from G_2.
6. Go to Step 2. [recursive call]

7. The intersection of two NL-complete languages (over the same alphabet) must be NL-complete too.

8. The intersection of two NP-complete languages (over the same alphabet) must be NP-complete too.

9. The language TRIANGLE-FREE $:= \{\langle G\rangle \mid G$ is a triangle-free undirected graph$\}$ is in L.

10. The language 3COLOR $:= \{\langle G\rangle \mid$ The undirected graph G is 3-colorable$\}$ is PSPACE-complete if and only if PSPACE $=$ NP.

Check-sum: In the above set, there are more false assertions than true!

Dr. Abhijit Das, Dept. of Computer Science & Engg, Indian Institute of Technology, Kharagpur 721 302