CS 172 Spring 2007 — Discussion Handout 10

1. **Satisfied, but not too satisfied**

   We define a **$\neq$-assignment** to a 3cnf formula ϕ as an assignment such that each clause contains two literals with unequal truth values (note that this must necessarily be a satisfying assignment).

   (a) Prove that the negation of a $\neq$-assignment is also a $\neq$-assignment.

   (b) Let $\neq$SAT be the collection of 3cnf-formulas that have an $\neq$-assignment. Obtain a polynomial time reduction from 3SAT to $\neq$SAT by replacing each clause $c_i = (y_1 \vee y_2 \vee y_3)$ with the two clauses $(y_1 \vee y_2 \vee z_i)$ and $(y_3 \vee \overline{z_i} \vee b)$, where z_i is a new variable for the clause c_i and b is a single new variable for the whole formula. Prove the correctness of this reduction.

2. **Hard to break off from many**

   A cut of an undirected graph G is defined as a partition of the vertex set into two disjoint subsets S and T. The size of the cut is the number of edges having one endpoint in S and one in T. Let

   $$MAX - CUT = \{\langle G, k\rangle |\ G \text{has a cut of size at least } k\}$$

   Show that MAX-CUT is NP-complete by arguing the correctness of the following reduction from $\neq$SAT to MAX-CUT:

   Given a formula ϕ with n variables and m clauses, create a graph having $3m$ vertices for each variable and $3m$ for its negation. Connect all vertices corresponding to x to all vertices corresponding to $\overline{x}$. Finally, identify 3 vertices for each literal with every clause (i.e. divide the $3m$ clauses into m groups of 3 each). For each clause c_i, form a triangle out of the vertices corresponding to the literals in the clause, using only the vertices in the groups corresponding to g_i.

3. **Newer heights of nastiness**

   Not only is computing the exact solution of many optimization problems NP-complete, it is even NP-complete to solve these problems approximately. For (say) a minimization problem, we say that an algorithm gives an r approximation if the cost of the solution given by the algorithm is no more that r times the cost of the optimum. Show that the following problem is NP-complete for any constant $r > 0$:

   r-APPROX-TSP: Given a set of points P, a cost function $f : P \times P \to \mathbb{N}$ and a number k, determine if there is a TSP solution of cost at most $r \cdot k$.

   (*Hint:* Modify the reduction from Hamiltonian Path to TSP)

4. **Old favorites: Adding colors to life**

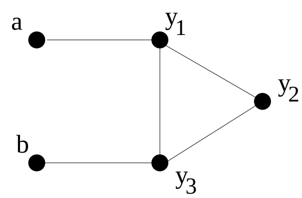
   k-COLORABILITY is the problem of finding an assignment of 1 color to each vertex of a given graph G, out of a total of k colors. such that no two adjacent vertices have the same color. These are some of the hardest NP problems to even approximate - the best known algorithm may use as many as $O(n^{0.2111})$ colors to color a graph which is actually 3-colorable. 2-colorability, however, can be solved in polynomial time.

   Here we construct a reduction to show that 3-COLORABOLITY is NP-Complete by reducing 3SAT to 3-COLORABILITY. We have the following components in the graph to "simulate" a 3SAT formula:

i) A triangle to represent the states `true`, `false` and a third state `don't-care`. This is because all 3 vertices of a triangle must be colored differently and we can interpret each color as stated above. We'll now use the numbers 1, 2 and 3 for the colors corresponding to `true`, `false` and `don't-care`.

ii) For each variable x we have two vertices, one for x and one for $\bar{x}$.

iii) A "sort of" OR-gate as shown below, which has the property that y_2 can be of color 1 (`true`) if and only if one of a and b is true (given that a and b take only `true` and `false values`).



We now carry out the reduction in steps:

a) Prove the property of the OR-gate.

b) Connect the variables to the *truth-triangle* appropriately to ensure that each vertex x_i and $\bar{x}_i$ is only colored `true` or `false`. Also, ensure that x_i and $\bar{x}_i$ do not get the same color.

c) Use the OR-gate to construct a gadget to check if a 3-clause is `true` using the colors of the 3 literals appearing in it.