Exercise of Chapter 2

Xiaoyu Chen

1 Exercise 2.5

(This exercise is mainly directed to readers with some exposure to computational complexity.) Garey and Johnson [36, §7.3] note Fact 2.4 without proof. Since I am not aware of any published proof, we should maybe pause to provide one. Garey and Johnson's reduction [36, §3.1.2] from 3Sat (the restriction of Sat to formulas with three literals per clause) to Exact3Cover (actually a special case of Exact3Cover called "3-dimensional matching") is almost parsimonious. The "truth setting component" is fine (each truth assignment corresponds to exactly one pattern of triples). The "garbage collection component" is also fine (it is not strictly parsimonious, but the number of patterns of triples is independent of the truth assignment, which is just as good). The "satisfaction testing component" needs some attention, as the number of patterns of triples depends on the truth assignment. However, with a slight modification, this defect may be corrected. Finally, to do a thorough job, we really ought to modify Garey and Johnson's reduction [36, §3.1.1] from Sat to 3Sat to make it parsimonious too.

1.1 solution

1.1.1 #Sat $\leq_p$ #3SAT

Suppose $E = e_1 \wedge e_2 \wedge \cdots \wedge e_k$. We want to replace each clause e_i to create a new expression F in 3SAT from. Here are some rules for doing this.

Add 3 aux variables a, b, c and add 7 clauses to F

$(a + b + c)(a + b + \overline{c})(a + \overline{b} + c)(a + \overline{b} + \overline{c})(\overline{a} + b + c)(\overline{a} + b + \overline{c})(\overline{a} + \overline{b} + c)$

- -

This ensures $a = 1, b = 1, c = 1$ in any solution.

$|e_i| = 1$, i.e. $e_i = (x)$ or $e_i = (\overline{x})$

$(x + \overline{a} + \overline{b})$

> $|e_i| = 2$, i.e. $e_i = (x + y)$
>
> $$(x + y + \overline{c})$$

> $|e_i| > 3$, i.e. $e_i = (x_1 + x_2 + x_3 + \cdots + x_m)$
>
> $$(x_1 + x_2 + y)(x_1 + x_2 + \overline{y})(x_1 + \overline{x_2} + y)(\overline{x_1} + x_2 + y)(y + x_3 + \cdots + x_m)$$
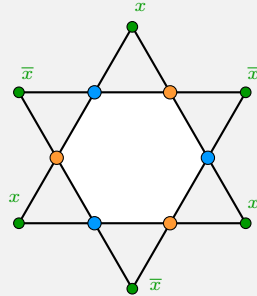>
> ---
>
> Here $y = x_1 + x_2$ in any solution. This could be verified by a truth table.

1.1.2   #3SAT $\leq_p$ #3DM

Suppose there are three groups: ● , ● , ● . Now we want to transfrom a 3SAT CNF E into a graph G.

> truth setting component
>
> For each variable x, if it appears k times in E, then we construct a truth setting component with $2k$ triangles.
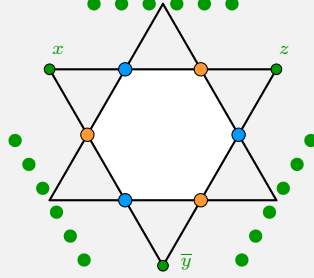>
> ---
>
>

If there is a clause $e = (x + y + z)$ in E, then we could convert e in to another form:

$$xyz + xy\overline{z} + x\overline{y}z + x\overline{yz} + \overline{x}yz + \overline{x}y\overline{z} + \overline{xy}z$$

So, x, y, z should choose an assignment from one of these cases. We construct our satisfaction testing component as follow:

For a single case, e.g. $x\overline{y}z$, we may construct a graph as follow

All the labeled vertices here are actually some vertices on truth setting component. And if a cases are both contain x, e.g. xyz and $x\overline{y}z$, they should use the same vertex which represent x in the truth setting component. Note that except $x, \overline{y}, $ and z, we have another 3 intervals. We connect each of these interval to 6 ● vertices. In each direction, these 6 vertices are shared by all the 7 cases, this ensures that there must be one case matches the verteces in truth setting component. Since the order of these non-labeled ● vertices does not make sense to 3SAT, so there is a factor of $(6!)^3$ in our answer.

Note that in the above 2 kinds of component the number of ● , ● and ● are not the same. Precisely, we have $|●| > |●| + |●|$, while $|●| = |●|$ To fix this, we need to introduce $|●| - |●|$ pairs of ● and ● . Then, for each pair of ● and ● , use them to construct triangles with all the ● in truth setting component. Since the order of these non-labeled ● vertices does not make sense to 3SAT, so there is a factor of $(|●| - |●|)!$ in our answer.

3