Lecture 11 (21.12.2012) Author: Tomasz Gogacz

W hierarchy

1 Motivation

Consider following problem:

k-Clique
Input: Graph G, natural number k
Question: Does graph G contain a k-clique as a subgraph?

Is this problem hard? Clearly it can be solved in time $O(|V|^{k+2})$. Can we solve this problem in FPT?

There is a **very** low probability that we will show $k - Clique \notin FPT$, because this statement implies $P \neq NP$. So we need more sophisticated notion of hardness.

Why we think that $P \neq NP$? Because Turing machine is a complicated device.

k-NTM-A
Input: Non-deterministic Turing machine M, input word x alphabet Σ, natural number k (unary coding)
Question: Does M accept x in at most k steps?

This problem can be solved by branching in time $\Theta(n^k)$. For decades no one has found fast solution. Thus we are justified to conjecture that this problem isn't in FPT. We will show, that $k - CLIQUE$ is at least that hard as $k - NTM - A$.

2 Reductions

Reductions are used to show hardness of problems, or conversely relative easiness. So for two languages $L_i \subset \Sigma^* \times \mathbb{N}$ the inequality $L_1 \leq_{FPT} L_2$ should implies "if $L_2 \in FPT$, then $L_1 \in FPT$".

Definition 1. A FPT-reduction in Karp sense is a function that:

- from pair (x, k) computes (x', k')
- $(x, k) \in L_1$ iff $(x', k') \in L_2$
- it is computable in time $f(k)poly(|x|)$ for some computable function f
- $k' \leq g(k)$ for some computable function g

Example 2. $k - CLIQUE \leq_{FPT} k - Indepentent\_Set$

Example 3. $k - CLIQUE \not\leq_{FPT} k - Vertex\_Cover$ (Probably).
$k - CLIQUE \equiv (|V| - k) - Vertex\_Cover$, but this interpretation does not give appropriate bound on parameter.

In terms of this section, motivation from previous section can be stated as follows. $k-CLIQUE$ is a hard problem because $k - NTM - A \leq_{FPT} k - CLIQUE$

3 Logic circuits

A circuit is a kind of labeled DAG. Vertexes with input degree 0 form input of circuit. All other vertexes are labeled by logic operations ($\vee, \wedge, \neg$). There is also a distinguished vertex called output of circuit.

We will divide set of all circuits into well behave classes.

Definition 4. $\mathbb{C}(h, w, s)$ is set of those circuits that every path from input to output of circuit:

- has lenght at most h
- has at most s gates with Fan-in bigger than w

Definition 5.

(i) $L(h, w, s) = \{(C, k) \in \mathbb{C}(h, w, s) : C \text{ is satisfiable by exactly } k \text{ one's}\}$

(ii) $\mathbb{L}(h, w, s) = \{F : F \leq_{FPT} L(h, w, s)\}$

(iii) $W[s] = \bigcup_{h=1}^{\infty} \bigcup_{w=1}^{\infty} \mathbb{L}(h, w, s)$

Theorem 6. *$k-CLIQUE$ is $W[1]$ complete. I.e. $k-CLIQUE \in W[1]$ and $\forall P \in W[1] \quad P \leq_{FPT} k - CLIQUE$*

Proof:

Step 1: $L(h, w, 1) \leq_{FPT} L^*(4, w', 1)$

Proof of Step 1:

Let $C \in \mathbb{C}(h, w, 1)$. WLOG we can assume that C is a tree. (It can be enforced by cloning gates with Fan-out bigger than one. Since h is a constant this operation takes polynomial time.) All negations can be now pushed down. Let Big denote set of gates with Fan-in bigger than w. The part of C above Big have size bounded by w^h (so also $|Big| \leq w^h$). It can be rewritten to $w^h - CNF$ (with Big as variables). Thus its depth is now at most two.

Similarly each subcircuit below each $t \in Big$ can be rewritten to $w^h - CNF$ or $w^h - DNF$. If we choose one of this possibilities in such a way, that the logic operator in node t is the same as in top node in subcircuit, we can merge those two nodes. Thus on each path there is at most four nodes (one big, at most two above and one below it).

Step 2: $L^*(4, w, 1) \leq_{FPT} L(2, w', 1)$ $(w' - CNF - SAT)$

Proof of Step 2: Form input circuit (C, k) we will produce circuit (C', k'). Let $\bar{k}$ be a number bigger than $2|Big|$ (it is bounded by w^4 - a constant). Let $k' = \bar{k}k + 2|Big|$.

For each $x_i \in Var(C)$ we produce new variables $x_i^1, \ldots, x_i^{\bar{k}}$. For each node $b \in Big$ we produce two variables - $y_b, \bar{y}_b$ (they denote true value of b), and $z_b^0, \ldots, z_b^{|Fan-in(b)|}$ (they correspond to true value of incoming edges).

Suppose that b has $\wedge$ as label. Then $z_b^0 = 1$ mean that all incoming edges have true value 1. $z_b^\alpha = 1$ mean that $\alpha - th$ incoming edge of b has true value 0. For $\vee$ label meanings are opposite.

C' in $w' - CNF - SAT$ will contain the following formulas:

- all possible $x_i^a \to x_i^b$
- $y_i \leftrightarrow \neg \bar{y}_i$

- part above Big
- all possible $\neg z_b^\alpha \vee \neg z_b^\beta$
- if b has $\wedge$ as a label then $y_b \leftrightarrow z_b^0$, otherwise $\bar{y}_b \leftrightarrow z_b^0$
- for z_i^α let C_i^α be a corresponding clause. Suppose that b has $\wedge$ as a label (for $\vee$ label we add analogical clauses). We add: $\neg z_i^0 \vee C_i^\alpha$, $z_i^\alpha \rightarrow \neg C_i^\alpha$

For each $b \in Big$, in each satisfying valuation, at most one z_b^α has true value one. Also if one x_i^a has true value one, all other x_i^b (notice the same lower index) must be true. Since $k' = \bar{k}k + 2|Big|$ and $\bar{k} > 2|Big|$, then exactly k of x_i is true and for each $b \in Big$ exactly one z_b^α is true.

It is easy to see, that other formulas guarantee correctness of reduction.

Definition 7. Let $\bar{L}(w) = L(2, w, 1)$ $[w - CNF - SAT]$. Let $\overline{AL}(w)$ be a variant of $\bar{L}(w)$ where all literals are negative.

Step 3: $\bar{L}(w) \leq_{FPT} \overline{AL}(w)$.

Proof of Step 3: Let $C \in w - CNF - SAT$ and $Var(C) = \{x_1, \ldots, x_n\}$. Create k rows each with n vertexes $a_{i,1}, \ldots, a_{i,n}$. This matrix will encode valuation of variables as increasing sequence. (i appear in sequence iff x_i has true value one.)

For each row i and each two different columns f, g we produce a clause $(\neg a_{i,g} \vee \neg a_{i,f})$. Those clauses guarantee that in each row at most one node can be true.

For each $i_1 < i_2$ and $j_1 \geq j_2$ we produce clause $(\neg a_{i_1,j_1} \vee \neg a_{i_2,j_2})$. Those clauses guarantee that sequence is increasing.

We will need a special gadget to catch numbers outside of sequence. It will be a table $(1, \ldots, n) \times (1, \ldots, n)$ between each two rows. An entry in such table will correspond with chosen values in adjacent rows. In similar way as before, we forbid choosing at the same time i in above row and any entry in table outside ith column. The same about row below and columns.

For each clause e in C there is a unique valuation of variables, that appear in e, which makes e false. How can witness of such valuation look like? It is easy to spot a true variable because it just appear in sequence. Because sequence is strictly increasing, to spot a false variable x_i it is enough to find a "jump" over position i i.e there is a row with value less than i and the next row has value bigger than i. Such jump can be witnessed by only one variable in table between those two rows. Therefore a witness has a size equal to $|C|$. Moreover if a sequence correspond to bad valuation one of such witnesses will have true value false.

We have to check whether number of those witnesses don't cause over-FPT blowup. There are at most $(nk)^w$ possibilities to choose node for true literals and $(kn^2)^w$ possibilities to choose nodes for false literals. Hence, we got at most $n^{3w}k^{2w}$ new clauses.

We put $k' = 2k - 1$. So in good valuation exactly one variable of each row and of each table will be true.

Step 4: $\overline{AL}(w) \leq_{FPT} \bar{L}(2)$.

Proof of Step 4:

Let $C \in \overline{AL}(w)$. As in step 2 we start from multiplying each variable $\bar{k} > \sum_{i=1}^{w} \binom{k}{i}$ times. Then bind them with implications.

For each set $P \subset \{1, \ldots, n\}$, $|P| \leq w$ create a new variable y_P with intension that $y_p \leftrightarrow \bigwedge_{i \in P} x_i$.

One implication is easy to enforce. For each $i \in P$ we create a clause $y_P \rightarrow x_i$. The second direction will be enforced by size of k'. We put $k' = k\bar{k} + \sum_{i=1}^{w} \binom{k}{i}$.

So we will have following clauses

(i) $x_i^{\alpha} \to x_i^{\beta}$

(ii) $y_P \to x_i$

(iii) for each $c \in C$, $\neg y_c$

By the size of k' and $\bar{k}$ at most k original variables can be set true. From the (ii) at most $\sum_{i=1}^{w} \binom{k}{i}$ of y's can be set true (the subsets of previous variables). Since we have to set true exactly k' variables, we have to set true exactly k original variables and all variables which correspond to its subsets. (iii) guarantees that such valuation will satisfy original formula.

At the end we get $L(h, w, 1) \leq_{FPT} L^*(4, w', 1) \leq_{FPT} L(2, w'', 1) \leq_{FPT} \overline{AL}(w'')$
$\leq_{FPT} \bar{L}(2) \leq_{FPT} \overline{AL}(2)$. Notice that $\overline{AL}(2)$ is just $k - Independent\_Set$. Variables are nodes and clauses are edges.