CS294 Fine-Grained Complexity and Algorithms

Lecture 16: ETH for problems beyond 3-SAT

Lecturer: Prof. Russell Impagliazzo
Scribe: Pasin Manurangsi

1 Introduction

In the previous two lectures, we have seen various algorithmic techniques for NP-Complete problems. These techniques yield non-trivial and different running times for each problem, which raises the following question: is there any underlying unity in the exact time complexities of these problems?

To answer this question, recall the exponential time hypothesis for 3-SAT ($\text{ETH}_{\text{3-SAT}}$):

Conjecture 1 (The Exponential Time Hypothesis [IP01]) *3-SAT requires 2^{cn} time to solve for some constant $c > 0$ where n is the number of variables. In other words, there exists $d > 0$ such that no algorithm running in time $O(2^{dn})$ can solve 3-SAT.*

In short, 3-SAT cannot be solved in subexponential time unless $\text{ETH}_{\text{3-SAT}}$ is false. It is natural to ask whether 3-SAT has a specific property that makes $\text{ETH}_{\text{3-SAT}}$ a plausible conjecture. It turns out the answer to this question is false as Impagliazzo, Paturi and Zane [IPZ01] proved similar running time lower bounds for many other problems.

Theorem 2 ([IPZ01]) *For many "reasonable" problems P, ETH_P is equivalent to $ETH_{\text{3-SAT}}$ where ETH_P is defined in a similar fashion to $ETH_{\text{3-SAT}}$.*[1]

We will describe over the next lectures what "reasonable" means in the theorem above. For now, let us turn our attention to a class of problems that captures many natural NP-hard problems and a type of reduction that is normally used to prove NP-hardness results for the problems in this class.

2 Constraint Satisfaction Problems

A constraint satisfaction problem or CSP is a problem that can be written in the following form:
INPUT:

- Variable $x_1, \ldots, x_n$

[1] Each type of problems has different running time lower bound one can hope for. The lower bound we would like for CSPs is mentioned at the end of Section 2.

– Alphabet set Σ

– Collection of allowable relations $\{(R_t, a_t)\}_t$ where $1 \le a_t \le n$ and $R_t : \Sigma^{a_t} \to \{0,1\}$

– A formula $R_{t_1}(x_{j_{1,1}}, \ldots, x_{j_{1,a_{t_1}}}) \wedge R_{t_2}(x_{j_{2,1}}, \ldots, x_{j_{2,a_{t_2}}}) \wedge \cdots \wedge R_{t_m}(x_{j_{m,1}}, \ldots, x_{j_{m,a_{t_m}}})$

OUTPUT: Whether there is an assignment to the variables such that the formula is satisfied.

As stated earlier, many natural NP-Complete problems can be stated this way. For example, k-SAT, 1-out-of-k SAT and k-coloring can be phrased as CSPs as shown in Figure 1.

Problems	Variables	Alphabets	Allowable Relations
k-SAT	Variables in the formula	$\{0,1\}$	$R(y_{i_1}, \ldots, y_{i_k}) = (y_{i_1} \vee \cdots \vee y_{i_k})$ where each y_{i_j} is a literal
1-out-of-k SAT	Variables in the formula	$\{0,1\}$	$R(y_{i_1}, \ldots, y_{i_k}) = (\exists! l, y_{i_l} = 1)$ where each y_{i_j} is a literal
k-coloring	Vertices	$\{0, 1, \ldots, k-1\}$	$R(x_{i_1}, x_{i_2}) = (x_{i_1} \neq x_{i_2})$

Figure 1: Demonstration of how to view k-SAT, 1-out-of-k SAT and k-coloring as CSPs.

Note that, for CSPs, we typically think of a_t's as constants. Observe that, for any CSP, one can use naive exhaustive search to solve the problem in time $O(|\Sigma|^n \cdot poly(m))$. This is typically the running time that we aim for in the exponential time hypotheses for such problems, i.e., we would like running time lower bounds to be exponential in n.

3 Gadget Reductions

For CSPs problems, a gadget reduction (also known as projection reduction) from problem $P = (\Sigma, \{(R_t, a_t)\}_t))$ to problem $P' = (\Sigma', \{(R'_t, a'_t\})_t)$ is a reduction characterized by constructing a set of constant number of relations (which may include new variables), called a *gadget*, in P' that represents a relation in P.

Informally, the reduction takes an instance $(\{x_1, \ldots, x_n\}, \bigwedge_{l=1}^{m} R_{t_l}(x_{j_{l,1}}, \ldots, x_{j_{l,a_{t_l}}})$ of P and produces an instance $(\{x'_1, \ldots, x'_{n'}\}, \bigwedge_{l'=1}^{m'} R'_{t'_{l'}}(x'_{j_{l',1}}, \ldots, x'_{j_{l',a'_{t_{l'}}}})$ of P' such that each relation $R_{t_l}(x_{j_{l,1}}, \ldots, x_{j_{l,a_{t_l}}})$ in P is translated to the relations $\wedge_p R'_{t'_{ip}}(x'_{j_{ip,1}}, \ldots, x'_{j_{ip,a'_{t_{ip}}}})$ in P' by a gadget and that the original relation $R_{t_l}(x_{j_{l,1}}, \ldots, x_{j_{l,a_{t_l}}})$ is true if and only if there exists an assignment to the new variables such that $\wedge_p R'_{t'_{ip}}(x'_{j_{ip,1}}, \ldots, x'_{j_{ip,a'_{t_{ip}}}})$ is true. (Note that we think of variables $x'_1 = x_1, \ldots, x'_n = x_n$ in the new instance as the old variables and $x'_{n+1}, \ldots, x'_{n'}$ as the new variables.)

For example, if we would like to reduce 3-SAT to 1-out-of-3 SAT, we can define a gadget as follows.

$$(x \vee y \vee z) \Rightarrow R_{1/3}(\overline{x}, a, b) \wedge R_{1/3}(\overline{y}, c, d) \wedge R_{1/3}(\overline{z}, e, f) \wedge R_{1/3}(a, c, e)$$

where a, b, c, d, e, f are new variables and $R_{1/3}$ is the 1-out-of-3 relation (i.e. $R_{1/3}(g, h, i)$ is true if and only if exactly one of g, h, i is true). It is not hard to see that $(x \vee y \vee z)$ is true if and only if there exists a, b, c, d, e, f such that $R_{1/3}(\overline{x}, a, b) \wedge R_{1/3}(\overline{y}, c, d) \wedge R_{1/3}(\overline{z}, e, f) \wedge R_{1/3}(a, c, e)$ is true.

In this example, if we would like to reduce an instance $(x \vee y \vee z) \wedge (\overline{y} \vee z \vee t)$ of 3-SAT to an instance of 1-out-of-3 SAT, then the result is $R_{1/3}(\overline{x}, a, b) \wedge R_{1/3}(\overline{y}, c, d) \wedge R_{1/3}(\overline{z}, e, f) \wedge R_{1/3}(a, c, e) \wedge R_{1/3}(y, \tilde{a}, \tilde{b}) \wedge R_{1/3}(\overline{z}, \tilde{c}, \tilde{d}) \wedge R_{1/3}(\overline{t}, \tilde{e}, \tilde{f}) \wedge R_{1/3}(\tilde{a}, \tilde{c}, \tilde{e})$ where $a, b, c, d, e, f, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e}, \tilde{f}$ are the new variables.

Finally, observe that, since each constraint is reduced by a gadget to a constant number of constraints, we have $m' = O(m)$. Moreover, since each constraint introduces at most a constant number of new variables, the total number of variables n' in the resulting instance is at most $O(n + m)$.

4 Running Time Lower Bound via Gadget Reduction

Unfortunately, the parameters we get in the previous section is still not good enough to prove Theorem 2. In particular, if the starting instance has $m = \omega(n)$ clauses, then the resulting instance have $n' = \omega(n)$ variables. This means that, with only the gadget reduction, $2^{\Omega(n)}$ running time lower bound for the original problem does not imply $2^{\Omega(n')}$ running time lower bound for the resulting problem. To solve this issue, we turn to the sparsification lemmas.

Recall that, in Lecture 13, we learned about the following sparsification lemma for maximum independent set due to Johnson and Szegedy [JS99].

Lemma 3 (Sparsification Lemma for Maximum Independent Set [JS99]) *Let $T_{IndSet}(n)$ be the running time needed to solve maximum independent set on a graph of n vertices (with any number of edges) and let $T_{IndSet\text{-}Deg \le d}(n)$ be the running time needed to solve maximum independent set on a graph of n vertices with each vertex has degree at most d.*

For all $c > 0$, there exists d such that

$$T_{IndSet}(n) \le 2^{cn} \cdot T_{IndSet\text{-}Deg \le d}(n).$$

In summary the sparsification lemma tells us that, if $T_{\text{IndSet}}(n)$ is exponential in n, then there exists d such that $T_{\text{IndSet-Deg}\le d}(n)$ is also exponential in n. In other words, if we assume $\text{ETH}_{\text{IndSet}}$, then even finding the maximum independent set on a graph with bounded degree (at most d) still takes at least exponential time.

As a result, using the following ideas, we can show that $\text{ETH}_{\text{IndSet}}$ implies $\text{ETH}_{\text{3-SAT}}$. First, we use the sparsification lemma to show that the maximum independent set problem is hard on bounded degree graphs. Using a gadget reduction[2], we then reduce the maximum inde-

[2]Here we abuse our notion of gadget reductions since the maximum independent set problem is in fact not a CSP.

pendent set problem on bounded degree graph to 3-SAT. Since $m = O(n)$ in bounded degree graph, the resulting 3-SAT instance has $n' = O(n)$, which yields $\text{ETH}_{\text{3-SAT}}$ as desired

We defer proving that $\text{ETH}_{\text{3-SAT}}$ implies $\text{ETH}_{\text{IndSet}}$ to next lecture, in which we will discuss about the sparsification lemma for 3-SAT and show that $\text{ETH}_{\text{3-SAT}}$ implies ETH_P for many problems P, including the maximum independent set problem.

References

[IP01] Russel Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, March 2001.

[IPZ01] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, December 2001.

[JS99] David S. Johnson and Mario Szegedy. What are the least tractable instances of max independent set? In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '99, pages 927–928, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.