**18.405J/6.841J Advanced Complexity Theory** March 4, 1997

Lecture 8

*Lecturer: Daniel A. Spielman* *Scribe: Saadeddine Mneimneh*

# 1 Relations between Different Complexity Classes

Some containment relations:

$P \subseteq RP$
$P \subseteq coNP$
$RP \subseteq NP$
$RP \subseteq BPP$
$NP \subseteq \Sigma_2^P$
$NP \subseteq \Pi_2^P$
$BPP \subseteq \Sigma_2^P$
$BPP \subseteq \Pi_2^P$
$coNP \subseteq \Sigma_2^P$
$coNP \subseteq \Pi_2^P$
$PH \subseteq PSPACE$

# 2 NP is as easy as detecting unique solutions

Several authors have observed that, among most known NP-complete problems, reductions can be found that preserve the number of solutions. Such reductions are called parsimonious reductions. A characteristic of each such NP-complete problem is that its instances have widely varying numbers of solutions. It is natural to ask whether the inherent difficulty in solving NP-complete problems is caused by this wide variation. The answer for this question is NO. Let $A$ be any NP-complete problem to which satisfiability is parsimoniously reducible. Let $\#A(x)$ denote the number of solutions to instance $x$. Then if we can find a randomized polynomial-time Turing machine that on input $x$ outputs:

0 if $\#A(x) = 0$
1 if $\#A(x) = 1$
Q(x) if $\#A(x) > 1$, where Q is an arbitrary boolean function

we would have $NP = RP$.

For this purpose, we use the concept of a randomized polynomial-time reduction: $\leq_{RP}$

Definition:
$A$ is randomized polynomial-time reducible to $B$ ($A \leq_{RP} B$) iff $\exists$ a randomized polynomial-time TM M and some polynomial $p$ such that:

if $x \notin A$ then $M(x) \notin B$ and
if $x \in A$ then $Pr[M(x) \in B] \geq p(|x|)^{-1}$

if $B \in RP$ and $A \leq_{RP} B$ then $A \in RP$

proof:

let $M_B$ be the $RP$ machine of $B$
if $x \notin A$ then $M(x) \notin B$ and $M_B$ rejects $M(x)$
if $x \in A$ then $Pr[M(x) \in B] \geq p(|x|)^{-1}$ and $Pr[M_B$ accepts $M(x)] \geq [2p(|x|)]^{-1}$

if we run $M_B$ a polynomial number of times, then we will boost the error probability.

# 3 Detecting Unique Solutions in RP-time means NP=RP

The idea is the following: For any problem in NP, we can reduce it the an instance of SAT. Repeatedly, we apply restrictions on the obtained formula. Then we run the RP machine to detect whehter the new formula has exactly one solution or not. The consequence of this is that any NP problem will be solved in RP time. Since $RP \subseteq NP$, then NP=RP.

The following will prove that given a formula, we can obtain a restricted version of it that has only one satisfying assignment by applying a polynomial number of random restrictions.

Consider $x_1, x_2 ..., x_n$ to be the satisfying assignments of a formula represented as vectors in $(GF(2))^n$
We use restrictions on the satisfying assignments (vectors) to decrease their number to only one satisfying assignment.

Lemma:
Let $w_1, w_2, ..., w_n$ be vectors chosen at random. Define

$H_i = \{x : x.w_j = 0; i \leq j\}$

Let $S$ be the set of vectors representing the satisfying assignments and assume $S \neq \Phi$

Let $S_i = H_i \cap S$,

then $Pr[\exists i | rank(S_i) = 1] \geq 1/2$

proof:

Without loss of generality, let $S$ contains the unit vectors $e_1, ..., e_k$ where $e_i[i] = 1$ and $rank(S) = k$.

Assume the first time the rank of $S$ decreases is when we intersect $S$ with $H_j$. So $rank(S_j) = rank(S \cap H_j) < k$ iff $w_j \notin 0^k\{0,1\}^{n-k}$. There are $2^n - 2^{n-k}$ such vectors. This means that there are $2^n - 2^{n-k}$ vectors that will decrease the rank. Moreover, if $w \in 1^k\{0,1\}^{n-k}$ then the rank will decrease to zero. Then the probability that the rank will be zero given that it decreases is:

$Pr[rank(S_j) = 0 | rank(S_j) < rank(S)] \leq 2^{n-k}/2^n - 2^{n-k}$
or
$Pr[rank(S_j) \neq 0 | rank(S_j) < rank(S)] \geq 1 - 2^{n-k}/2^n - 2^{n-k} = 2^k - 2/2^k - 1$

Let S be a set of rank k.
if k=1 then $Pr[\exists i | rank) S_i) = 1] = 1$;

if k=2, then $Pr[\exists i | rank(S_i) = 1] \geq$ the probability that the rank decreases for the first time without going to zero $\geq 2^k - 2/2^k - 1$

if k=3, then $Pr[\exists i | rank(S_i) = 1]$=the probability that the rank decreases for the first time without going to zero, and then for the second time without going to zero.

So if $t_k = Pr[\exists i | rank(S_i) = 1]$ when $rank(S) = k$, then

$t_k \geq (2^k - 2/2^k - 1).t_{k-1}$.

We can show by induction that $t_k \geq 2^{k-1}/2^k - 1$ which is $\geq 1/2$.

All this analysis depends on the fact that the rank decreases when more restrictions are imposed. To make sure that this will actually happen, $w_1, ..., w_n$ have to be linealry independent, the probability of which is:

$[(2^n - 1)/2^n].[(2^n - 2)/2^n].[(2^n - 4)/2^n]...[(2^n - 2^{n-1})/2^n] =$

$(1 - 1/2^n)(1 - 1/2^{n-1})...(1/2) \geq 1/4$

because only $2^k$ vectors can depend on $w_1, ..., w_k$.

So $Pr[\exists i | rank(S_i) = 1] \geq (1/4).(1/2) = 1/8$