6.841 Advanced Complexity Theory March 02, 2005

Lecture 9

Lecturer: Madhu Sudan *Scribe: Mariana Baca*

1 Overview

Today we will look at two results:

- Karp-Lipton: NP $\subseteq P/_{poly} \Longrightarrow \sum_3^p = \prod_3^p$
- Fortnow: SAT $\in$ L $\Longrightarrow \exists\ \epsilon > 0$ s.t. SAT $\notin$ TIME($n^{1+\epsilon}$)

We hope that in studying circuit lower-bounds, we can prove NP $\neq$ P by proving NP $\not\subseteq$ to $P/_{poly}$. This seems like a contradictory result given Non-Uniformity can be used to solve undecidable problems. The Karp-Lipton result proves that even though non-uniform advice may be very powerful, the advice needed to solve NP is "easy" to generate (where "easy" refers to PSPACE or PH). This advice may be in the form of a circuit.

2 Karp-Lipton Result

Theorem 1 *NP* $\subseteq P/_{poly} \Longrightarrow \sum_3^p = \prod_3^p$

Proof We can easily generate a circuit to decide a SAT formula ϕ for a given length n:

C(ϕ) = 1 iff $\exists\ y$ s.t. $\phi(y) = 1$. $\forall\phi$: C(ϕ) = 1 $\Longleftrightarrow \phi \in$ SAT.

This language is in PH for inputs of length n, because it can be expressed by a bounded sequence of quantifiers. We can expand this equation in order to express SAT $\in$ PH in a canonical form:

$\exists C\ \forall\phi, y'\ \exists y$:

C(ϕ) = 0 $\Longrightarrow \phi(y') = 0$

C(ϕ) = 1 $\Longrightarrow \phi(y) = 1$

This proves that NP $\subset \sum_3^p$, which is not a very useful result at all. However, this result is useful for simplifying other problems in the hierarchy, in particular $\sum_4^p$-complete problems:

L={ϕ | $\exists x_1\ \forall x_2\ \exists x_3\ \forall x_4\ \phi(x_1, x_2, x_3, x_4)$ =1 }

L is Σ_4^p-complete, but if we "guess" the values of x_1, x_2, x_3, it can be re-written as L', which is a coNP-complete language as follows:

L'={(ϕ, x_1, x_2, x_3) | $\forall x_4\ \phi(x_1, x_2, x_3, x_4)$ =1 }

The circuit for coSAT can be found with the following $\sum_3^p$ problem:

$\exists C\ \forall \phi, y'\ \exists y$:

$C(\phi) = 1 \Longrightarrow \phi(y') = 1$

$C(\phi) = 0 \Longrightarrow \phi(y) = 0$

We can now try to find a C which decides L' in parallel to trying to find x_1, x_2, x_3 that are valid inputs for L'. This circuit will decide whether a given $\phi \in$ L.

$\phi \in$ L iff:

$\exists x_1, C$

$\forall x_2, \phi', x_1', x_2', x_3', y_4'$

$\exists x_3, x_4'$ s.t.

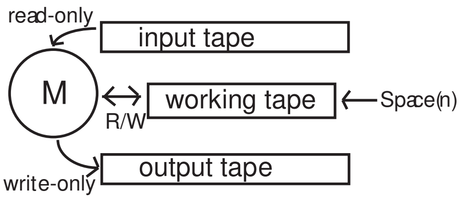
$C(\phi', x_1', x_2', x_3') \Rightarrow \phi'(x_1', x_2', x_3', x_4')$

$\bigwedge \phi'(x_1', x_2', x_3', y_4') \Rightarrow C(\phi', x_1', x_2', x_3')$

$\bigwedge C(\phi, x_1, x_2, x_3, x_4) = 1.$

Since this circuit solves L, and the circuit is clearly in $\sum_3^p$, PH collapses to $\sum_3^p$. ■

3 A Short Deviation on Space Complexity



Space complexity refers to the length of the work tape used by a Turing Machine M. The Work tape refers to a tap which is read and write, as opposed to the input tape, which is read only, or the output tape which is write only. If a Turing Machine M computes $f_1(x)$, and uses space s_1, and another Turing Machine M' computes $f_2(x)$ uses space s_2, the space complexity of $f_2(f_1(x))$ uses $s_1 + s_2$ space. Similarly, if computing A*A takes $\log n$ space, A^{2*i} takes $i * \log n$ space. This convention is adopted to be able to understand less than linear space classes.

4 Fortnow's Theorem

Theorem 2 $SAT \in L \Longrightarrow \exists\ \epsilon > 0$ *s.t.* $SAT \notin TIME(n^{1+\epsilon})$

Although Fortnow's theorem is not a statement about alternation, it nevertheless uses alternation to prove an important result. SAT and L are both proven to be very robust classes. TIME($n^{1+\epsilon}$) is not a very robust class, but important nonetheless.

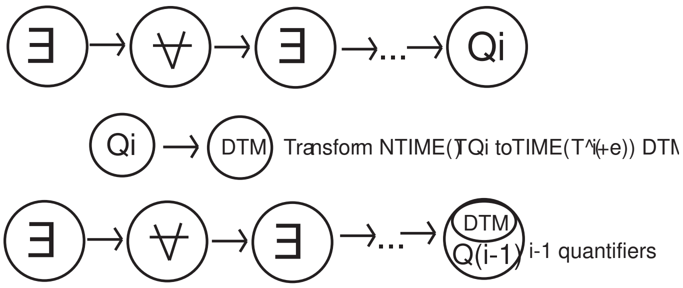
Proof Idea

- Alternation is a powerful tool when simulating low-space computation (SPACE(s) $\subseteq$ ATIME(s^2))
- Assume SAT $\in$ L and SAT $\in$ TIME($n^{1+\epsilon}$) for tiny ϵ.
- SAT $\in$ TIME($n^{1+\epsilon}$) $\Longrightarrow$ Alternation is not powerful in *general* (not a contradiction, yet).
- NTIME (n) $\leq$ SAT($n \log n^2$) $\in$ L $\Longrightarrow$ General computation reduces to low-space computation. Here we form a contradiction.

■

Claim 3 *SAT $\in$ TIME($n^{1+\epsilon}$) $\Rightarrow \forall$ NTM M, $\exists$ DTM M' s.t. L(M) = L(M') and M' runs in time $T^{(1+\epsilon)}$ and M runs in time T. This uses a strong form of Cook's Theorem. So given the assumption, for any NTM, its corresponding DTM will run only slightly slower.*

Claim 4 *SAT $\in$ TIME($n^{1+\epsilon}$) $\Rightarrow \forall$ i, $\sum_i^{TIME(T)} \subseteq$ DTIME($T^{(1+\epsilon)^i}$).*



Using the inductive hypothesis, you can reduce every step in the alternation to a sequence of DTM TIME($T^{(I+\epsilon)}$), which after i steps becomes a DTM TIME($T^{(i+\epsilon)^i}$).

Proof Assume NTIME(n) (and thus, SAT) $\in$ L and SAT $\in$ TIME($n^{1+\epsilon}$). Because of Claim 4, we know $\sum_i^{TIME(T)} \subseteq$ DTIME($T^{(1+\epsilon)^i}$). We also know DTIME($T^{(1+\epsilon)^i}$) $\subseteq$ NTIME($T^{(1+\epsilon)^i}$). Because NTIME(n) $\in$ L, NTIME($T^{(1+\epsilon)^i}$) $\subseteq$ SPACE($c * (1+\epsilon)^i \log T$). We shall now prove that SPACE($c * (1+\epsilon)^i \log T$) $\subseteq \sum_{i-1}^{TIME(T')}$ where T' $\leq T^{2/5}$, causing a contradiction.

Take a simulation which takes SPACE(s) to simulate (where s = $c * (1+\epsilon)^i \log T$). This computation takes at most 2^s steps to compute. Divide the computation in d parts (so that each computation is $2^s/d$ steps long):

$\exists c_1, c_2, c_3, ...c_d$

$\forall j \in [d], c_j \rightarrow c_{j+1}$ in $2^s/d$ timesteps.

This runs in d*s time per quantifier. It thus takes +2 quantifiers to reduce a computation from 2^s time to $2^s/d$ time. If you pick d carefully, you'll be able to decrease the time and still end up with i-1 quantifiers.

SPACE(s) $\subseteq \sum_{i-1}^{(i-1)s*d}$

Set d = $2^{2s/i}$ breakpoints. Thus, for $\sum_{i-1}^{Time(T'=(i-1)s*d)}$, for s = $c{*}(1{+}\epsilon)^i \log T$, T' $\leq T^{2*c(1+\epsilon)^i/i}$. Pick i ¿ 10*c . Pick ϵ to be small, s.t. $(1+\epsilon)^i$¡2. Then, $T^{2*c(1+\epsilon)^i/i} \leq T^{2/5}$. This proves SPACE($c*(1+\epsilon)^i \log T$) $\subseteq \sum_{i-1}^{TIME(T')}$ where T' $\leq T^{2/5}$, causing the contradiction. ■

Corollary 5 (BURMAN, FORTNOW (1997)) *SAT does not have uniform circuits of depth ($O(\log n)$) and size $n^{(i+\epsilon)}$.*