

Computational Complexity Theory

Lecture notes by Henrik Hald Nørgaard

Lecture 17, 22/11 2006

Exercise: We know from lecture 15 that

$$PARITY \leq_{cd} MAJORITY \tag{1}$$

Prove that it is not the case that $PARITY$ reduces to $MAJORITY$ using a AC^0 many-one reduction. Hint: Use Hastad's switching lemma.

1 Smolensky's polynomial method and $PARITY$

We will finish the proof of the following theorem:

Theorem 1 (Smolensky, '86)

$$PARITY \notin AC^0[MOD_3] \tag{2}$$

Remark: Smolensky actually proved a more general theorem. If p and q are primes, $p \neq q$ and $k, l \geq 1$, then

$$MOD_{p^l} \notin AC^0[MOD_{q^k}] \tag{3}$$

The theorem above emerges with $p = 2, q = 3, k = l = 1$.

We will prove the theorem by proving that all functions in $AC^0[MOD_3]$ have a certain property that $PARITY$ does not have. This property is:

Definition 1 (Property L) *All functions $\{f_n\}_{n=1}^\infty \in AC^0[MOD_3]$ are polynomially representable in the following way: For $n \geq 1$ here exists a polynomial $p \in \mathbb{Z}_3[x_1, \dots, x_n, w_1, \dots, w_m]$ where $m = n^{O(1)}$ and p has degree $\text{polylog}(n)$ such that:*

$$\forall x \in \{0, 1\}^n \exists w \in \{0, 1\}^m : P_w(f(x) \neq p(x, w)) < \frac{1}{n^{\omega(1)}} \tag{4}$$

Notice that any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ has a representation as a polynomial of degree n using no extra variables simply by spelling out the entries in the truth table corresponding to f .

The proof that alle functions in $AC^0[MOD_3]$ actually have property L was given in the previous lecture. Now we will use this property to prove Smolensky's theorem:

Theorem 2 *PARITY does not have property L.*

Proof: Suppose that *PARITY* has property L. Then we claim that for any boolean function of n variables $f : \{0, 1\}^n \rightarrow \{0, 1\}$ there is a polynomial $p \in \mathbb{Z}_3[x_1, \dots, x_n, w_1, \dots, w_m]$ where $m = n^{O(1)}$ and p has degree $\frac{n}{2} + o(n)$, such that:

$$\forall x \in \{0, 1\}^n \forall w \in \{0, 1\}^m : P(f(x) \neq p(x, w)) < \frac{1}{n^{\omega(1)}} \quad (5)$$

In the following we will think of f as having domain $\{-1, 1\}^n$. This is for technical convenience only and the result for domain $\{-1, 1\}$ imidiately translates to the equivalent result for $\{0, 1\}$. We start out with a representation of f as a polynomial based on the truth table of f and split this polynomial into terms of low and high degree:

$$f(x_1, \dots, x_n) = \sum_{I \subseteq \{x_1, \dots, x_n\}} c_I \prod_{i \in I} x_i \quad (6)$$

$$= \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| \leq \frac{n}{2}} c_I \prod_{i \in I} x_i + \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| > \frac{n}{2}} c_I \prod_{i \in I} x_i \quad (7)$$

In order to handle the problematic terms in the second sum, we observe that for $x \in \{-1, 1\}$:

$$\prod_{i \in I} x_i = \prod_{j=1}^n x_j \cdot \prod_{i \in I^c} x_i \quad (8)$$

and that if $k = |\{i \in I | x_i = -1\}|$ then

$$\prod_{j=1}^n x_j = (-1)^k = PARITY(x_1 - 1, \dots, x_n - 1) + 1 \pmod{3} \quad (9)$$

Using these equalities in the equation above gives:

$$f(x_1, \dots, x_n) = \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| \leq \frac{n}{2}} c_I \prod_{i \in I} x_i \quad (10)$$

$$+ \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| > \frac{n}{2}} c_I \prod_{j=1}^n x_j \left(\prod_{i \in I^c} x_i \right) \quad (11)$$

$$= \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| \leq \frac{n}{2}} c_I \prod_{i \in I} x_i \quad (12)$$

$$+ \sum_{I \subseteq \{x_1, \dots, x_n\}, |I| > \frac{n}{2}} c_I (\text{PARITY}(x_1 - 1, \dots, x_n - 1) + 1) \prod_{i \in I^c} x_i \quad (13)$$

If $|I| > \frac{n}{2}$ then $|I^c| \leq \frac{n}{2}$ and by assumption *PARITY* can be polynomially represented by a polynomial of degree $o(n)$. This proves the claim.

This implies that for any function $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ there is a polynomial p of degree $\frac{n}{2} + o(n)$ so that

$$f(x) = p(x) \quad (14)$$

for at least a $(1 - \frac{1}{n^{\omega(1)}})$ -fraction of the domain $\{-1, 1\}^n$. Because the claim implies:

$$\mathfrak{R}x \in \{-1, 1\}^n \mathfrak{R}w \in \{-1, 1\}^m : P(f(x) = g(x, w)) \geq 1 - \frac{1}{n^{\omega(1)}} \quad (15)$$

and changing the order of x and w this means that:

$$\exists w \in \{-1, 1\}^m \mathfrak{R}x \in \{-1, 1\}^n : P(f(x) = g(x, w)) \geq 1 - \frac{1}{n^{\omega(1)}} \quad (16)$$

Hardcoding such an w into the polynomial p gives the above claim. Now the rest of the argument is a counting argument: The total number of functions $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ is 2^{2^n} , whereas the number of functions that can be approximated as described above by a polynomial of low degree is bounded by the number of polynomials of degree at most $d := \frac{n}{2} + o(n)$ times the number of functions $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ that agree with a fixed polynomial on a $(1 - \frac{1}{n^{\omega(1)}})$ -fraction of the domain. The first factor is bounded by $3^{\binom{n}{d}}$ and

$$\binom{n}{\frac{n}{2}} \leq \frac{n^{\frac{n}{2}}}{(\frac{n}{2})!} \approx \frac{2^n e^n}{n^{\frac{n}{2}} \sqrt{n}} = o(2^n) \quad (17)$$

using Stirling approximation formula for $n!$. This implies that also $\binom{n}{d} = o(2^n)$.

The second factor equals the number of functions $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ that agree with a fixed polynomial except on at most $k := \frac{2^n}{n^{\omega(1)}}$ entries. This is bounded by:

$$\sum_{i=0}^k \binom{2^n}{i} 2^i \leq \sum_{i=0}^k \frac{(2^n)^i}{i!} 2^i \leq k 2^{(n+1)k} \leq 2^{n+(n+1)\frac{2^n}{n^{\omega(1)}}} = 2^{o(2^n)} \quad (18)$$

These two bounds together imply that the number of functions that can be approximated by a polynomial of low degree is at most $2^{o(2^n)}$. This shows that the initial assumption about *PARITY* must be wrong.

Remark: The above kind of proof is very different from a proof using diagonalization (e.g. the proof of $DTIME(n) \subsetneq DTIME(n^2)$), that will tell you nothing about the complexity classes involved. Here we have a specific property of languages/functions in $AC^0[MOD_3]$.

Exercise: Where did we use, that 3 is a prime?

2 Razborov-Rudich's natural proofs

Definition 2 A natural property for a complexity class C is defined by the following statements:

1. it is a property of functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ (a truth table of size 2^n)
2. for all functions $\{f_n\} \in C$, f_n has the property for n large
3. the property is decidable in polynomial time in the size of the truth table
4. a random function does not satisfy the property with probability $1 - o(n)$

Example: The property

f can be set to constant by setting all but \sqrt{n} variables to constant

is a natural property of AC^0 as we have seen in a previous lecture because the 2^n subsets of $\{0, 1\}^n$ can be each be checked in polynomial time in the size of the truth table of the function. It is an open problem whether this can be decided in linear time.

Definition 3 A (strong) pseudorandom function generator is a family of functions

$$f_n : \{0, 1\}^{s(n)} \times \{0, 1\}^n \rightarrow \{0, 1\} \quad (19)$$

for $n \geq 1$ where $s = O(n^k)$. The first argument is called the 'key' and the second argument is called the 'message'. For fixed key this function should look random, that is: For all algorithms running in polynomial time (in 2^n) we have that:

$$|P_z(A(f(z, \cdot)) = 1) - P_{u \in \{0, 1\}^{2^n}}(A(u) = 1)| < \frac{1}{n^{\omega(1)}} \quad (20)$$

A pseudorandom function generator belongs to the complexity class C if the language

$$\{(k, x) \mid n = |x| \wedge f_n(k, x) = 1\} \quad (21)$$

belongs to C .

Definition 4 *The cryptographic hypothesis is the assumption that certain concrete functions in P and even some in TC^0 are pseudorandom function generators.*

Theorem 3 (Razborov-Rudich) *Under the cryptographic hypothesis, P does not have a natural property.*

This theorem follows from the cryptographic assumption and the following theorem:

Theorem 4 *If C has a pseudorandom function generator, then C has no natural property.*

Proof: Use the natural property as the algorithm A in the definition of a pseudorandom function generator. The property of A that a random function satisfies A with low probability contradicts the fact that the functions in C created by the pseudorandom function generator for a fixed key all have the property A using the inequality in the definition of the pseudorandom function generator.

Remark: Razborov and Rudich have shown that many proofs fall within their framework of natural properties. As a rule of thumb either a complexity class can either be used for cryptography or you can prove lower bounds against it to separate it from other classes. It is though not clear if all proofs of this kind can be put into the Razborov-Rudich framework.

Remark: Existence of cryptography is a stronger assumption than $P \neq NP$, but if they were equivalent Razborov-Rudich's theorem says that:

$$P \neq NP \Rightarrow \text{we can't prove it using the Razborov-Rudich framework}$$

Remark: Razborov has proven that all proofs of $P \neq NP$ in a logical system containing a certain fragment of bounded arithmetic fall with the framework of natural properties. Hence under the cryptographic assumption this system is not powerful enough to show $P \neq NP$. But on the other hand it is also known, that it is not possible to show the pigeon hole principle in this system...

Definition 5

$$ACC^0 = \bigcup_k AC^0[MOD_k] \quad (22)$$

3 Interactive proof systems

The theory of interactive proof systems was developed in the 80'ties. Our goal in this part of the course will be the *PCP*-theorem, that is concerned with the problem of determining which approximation problems for *NP*-problems that are *NP*-hard.

First we will extend the complexity class *NP* in the following way:

NP contains the languages *L* for which a sceptical agent *I* can be convinced that an element *x* is in *L* by a short proof (that could e.g. be published in a book). The new class *DIP* (Deterministic Interactive Proof) contains the languages *L* for which a sceptical agent *I* can be convinced that *x* is in *L* by interacting with a computationally unbounded device.

Formally, we define *DIP* as the set of languages that are decided by the following setup: We have two TM's, *V* (the verifier) and *P* (the prover), that share the input tape and also share a working tape that they can use to communicate. *V* is a polynomial time TM, while *P* is a TM with unbounded power and they thus model respectively the sceptical agent and the computationally unbounded device mentioned above. The TM *V* either accepts or rejects the input and by doing so determines if (*V*, *P*) accepts or rejects the input. A language *L* is in *DIP* iff

$$\exists P \text{ prover} : x \in L \Leftrightarrow (V, P)(x) \text{ accepts} \quad (23)$$

and

$$\forall x \in L \forall P^* \text{ prover} : (V, P^*)(x) \text{ rejects} \quad (24)$$

Using the model of the sceptical agent above, *x* is accepted if the computationally unbounded device is able to convince *V* that $x \in L$. On the other hand if $x \notin L$ then it is not possible to convince *V* that $x \in L$.

Proposition 1 $NP = DIP$

Proof: $NP \subseteq DIP$: Here *P* writes the proof that $x \in L$ on the joint communication tape of *V* and *P* and *V* then accepts *x* iff *V* can verify that $x \in L^c$ using this proof.

$DIP \subseteq NP$: *P* is computationally unbounded and knows *x* and can thus simulate everything that *V* will do, in particular the communication between the two TM's. In other words *P* cannot be surprised by *V* and thus the setup in *DIP* effectively does not add any power than the power already held by *V* that alone determines a language in *NP*.

We now add another feature to the setup in *DIP* to get the complexity class *IP*. In *IP* *V* has access to a tape full of random bits, that are not accessible to *P*. Again, it is the TM *V* that either accepts or rejects the input thus determines

if (V, P) accepts or rejects the input. A language L is in IP iff

$$\exists P \text{ prover} : x \in L \Leftrightarrow (V, P)(x) \text{ accepts} \quad (25)$$

and

$$\forall x \in L^c \forall P^* \text{ prover} : P((V, P^*)(x) \text{ accepts}) \leq \frac{1}{2} \quad (26)$$

Graph-isomorphism is known to be in $coNP$ and believed to be in NP . We now prove:

Theorem 5 *Graph-isomorphism is in IP .*

Proof: Given graphs G_1 and G_2 represented as adjacency matrices, V picks one of them at random and does a random permutation of the vertices of the graph. Then V gives the resulting graph H to P and ask P which one of the graphs G_1 and G_2 is isomorphic to H ?

If $G_1 \approx G_2$, P can only guess the answer at random with probability $\frac{1}{2}$.

If $G_1 \not\approx G_2$, P can separate G_1 and G_2 and thus determine which one of them is isomorphic to H .

Remark: Next time we will prove that in fact $IP = PSPACE$.