



Toda's Theorem

Part 2

Dmitry Shiryaev

JASS 06



Main Theorem: $PH \subseteq P^{PP}$

$$PH \subseteq BPP^{\oplus P} \subseteq PP^{\oplus P} \subseteq P^{\#P} = P^{PP}$$

The first inclusion was proved in the previous lecture.

In this part we prove the rest 3 inclusions.



Plan of Proof

0) Evidently $BPP^{\oplus P} \subseteq PP^{\oplus P}$

1) Define the class $\#P$

2) Prove that $P^{\#P} = P^{PP}$

3) Prove that $PP^{\oplus P} \subseteq \#P$



Some Definitions

- L is from PP if there exists polynomial-time bounded NTM M such that

$$x \in L \Leftrightarrow P(M(x) = 1) > \frac{1}{2}$$

- L is from $\oplus P$ if there exists polynomial-time bounded NTM M such that

$$x \in L \Leftrightarrow \{\text{The number of accepting computation paths of } M \text{ on input } x \text{ is odd}\}$$



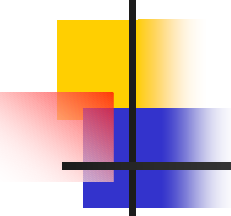
1) Definition of $\#P$

$acc_M(x)$ is the number of accepting computation paths of M on input x .

$\#P = \{ f : \Sigma^* \rightarrow N \cup \{0\} \mid \exists \text{ polynomial-time NTM } M_f$

such that $\forall x f(x) = acc_{M_f}(x) \}$


Example: #SAT –given a boolean formula one should find the number of it's satisfying assignments of variables.


$$2) P^{\#P} = P^{PP}$$

Part 1. $P^{PP} \subseteq P^{\#P}$

W.L.O.G., lengths of all branches of computation tree are equal. Using oracle we can find the number of accepting computations of any Turing machine.

It's easy to check whether it is more or less than a half. It can be done using machine itself.



$$2) P^{\#P} = P^{PP}$$

Part 2 $P^{\#P} \subseteq P^{PP}$

Let oracle from $\#P$ be the NTM M .

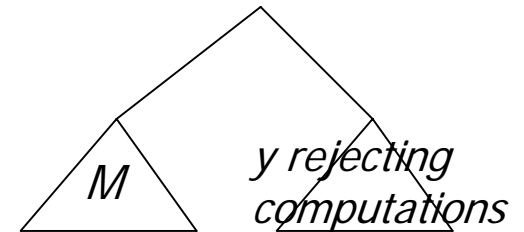
Define $L = \{(x \in \Sigma^*, y \in N) : acc_M(x) > y\}$

If we'll show that L is from PP , then using binary search we'll find $acc_M(x)$ in polynomial time.



$$2) P^{\#P} = P^{PP}$$

Part 2. $P^{\#P} \subseteq P^{PP}$

Now let's build the NTM M'
showing that L is from PP .



It will make the non-deterministic choice
with 2 branches:
one branch runs M ,
another branch makes fictitious computations,
number of them is exactly as M have,
with y rejecting computation paths.


$$2) P^{\#P} = P^{PP}$$

Part 2. $P^{\#P} \subseteq P^{PP}$

Now let's find the number of accepting paths of M' . It is $acc_M(x) + (all_M - y)$

(all_M is the whole number of computation paths of M)

Then the probability that M' accepts is

$$\frac{acc_M(x) + (all_M - y)}{2 \cdot all_M} = \frac{1}{2} + \frac{acc_M(x) - y}{2 \cdot all_M}$$

which is more than $1/2$ iff $acc_M(x) > y$


$$3) PP^{\oplus P} \subseteq P^{\#P}$$

Lemma

The functions s_i are defined recurrently:

$$s_0(z) = z$$

$$s_i(z) = 3(s_{i-1}(z))^4 + 4(s_{i-1}(z))^3$$

Then $\forall i \geq 0, \forall z \in N$ *if* $(2 \mid z)$
then $(2^{2^i} \mid s_i(z))$
else $(2^{2^i} \mid (s_i(z) + 1))$

Proof by induction


$$3) PP^{\oplus P} \subseteq P^{\#P}$$

Let $L \in PP^{\oplus P}$

$\exists A \in (P^{\oplus P} = \oplus P)$:

$$L = \{x \mid |\{y \in \{0,1\}^{p(|x|)} \mid (x, y) \in A\}| > \frac{1}{2} 2^{p(|x|)}\}$$

$$l(x) = \lceil \log p(|x|) + 1 \rceil \quad (p - \text{polynomial})$$

We'll show that L is from $P^{\#P}$
by describing machine M .

1. We find $q(z) = (s_{l(x)}(z))^2$ recurrently.

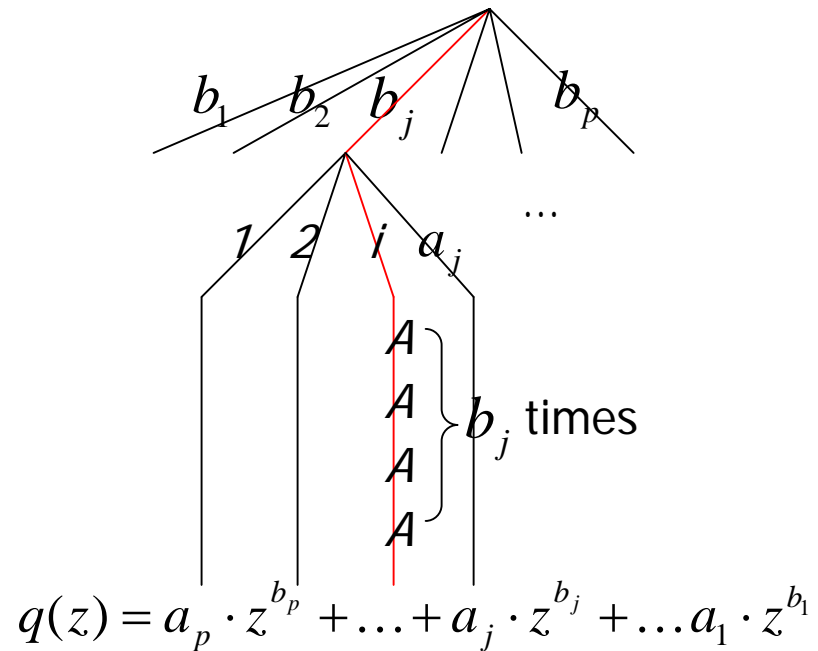
Evidently, $\deg(q) \leq 16^{l(x)} \leq p^4(|x|)$

$$3) PP^{\oplus P} \subseteq P^{\#P}$$

2. a) We make the non-deterministic choice, number of branches is equal to the number of non-zero coefficients of $q(z)$.

b) Each of that branches splits in some new branches, number of them is equal to appropriate coefficient of $q(z)$.

c) In each branch we run NTM A , number is equal to appropriate power. Next machines starts iff previous accepted.



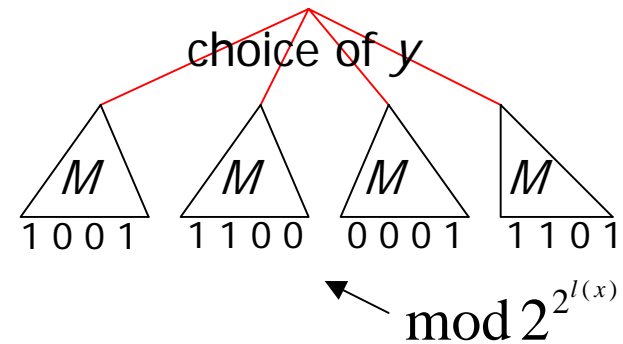

$$3) PP^{\oplus P} \subseteq P^{\#P}$$

M has exactly $(s_{l(x)}(z))^2$ accepting paths,
where z is the number of accepting paths of A
on input x .

By lemma, if number of accepting paths of A
was odd, then M will have 1 modulo $2^{2^{l(x)}}$
accepting paths,
else 0 modulo $2^{2^{l(x)}}$

$$3) PP^{\oplus P} \subseteq P^{\#P}$$

To find whether x belongs to L we will describe machine N and find the number of its acc. paths.



Let N guess y and then run $M(x, y)$.
 Number of com. paths y is $2^{p(|x|)} < 2^{2^{l(x)}} \leq 2^{p(|x|)+1}$

Hence number of proper com. paths =
 number of accepting paths of N modulo $2^{2^{l(x)}}$

We only have to see whether
 it is more or less than $\frac{1}{2} 2^{p(|x|)}$



Links

- E.A. Hirsch. Lecture notes. Lecture 9. (in Russian)
<http://logic.pdmi.ras.ru/~hirsch/students/complexity1/>
- S. Toda: PP is as Hard as the Polynomial-Time Hierarchy
<http://locus.siam.org/fulltext/SICOMP/volume-20/0220053.pdf>