Toda Theorem

$$PH \subseteq P^{PP}$$

Part1: $PH \subseteq BPP^{\oplus P}$

Language L belongs to class $\oplus P$(parity P) if exists polynomial-time non-deterministic Turing machine, such x lies in L is equivalent to that fact, that number of accepting computations is odd(or, exists polynomially checked relation R such that the number of y(R(x, y)= 1) is odd.

General Idea of proof

At first, we'll notice,that $NP \subseteq RP^{\oplus P}$ (it can be evidently deducted from Valiant-Vazirani Lemma)

Then, obviously $NP \subseteq BPP^{\oplus P}$

But the proof of Valiant-Vazirani Lemma can be relativised so $NP^A \subseteq BPP^{\oplus P^A}$ (*)

It would be enough to prove, that $\forall i \in N \quad \Sigma^i P \subseteq BPP^{\oplus P}$

We will do it by the mathematical induction

If i is equal to 1- we've already proved it

So, considering these three lemmas being truth

Lemma 2: $\oplus P^{BPP^A} \subseteq BPP^{\oplus P^A}$

Lemma 3: $\oplus P^{\oplus P} \subseteq \oplus P$

Lemma 4: $BPP^{BPP^A} \subseteq BPP^A$

we'll receive

$$\Sigma^{k+1} P = NP^{\Sigma^k P} \subseteq NP^{BPP^{\oplus P}} \subseteq BPP^{\oplus P^{BPP^{\oplus P}}} \subseteq BPP^{BPP^{\oplus P^{\oplus P}}} \subseteq BPP^{BPP^{\oplus P}} \subseteq BPP^{\oplus P}$$

The first inclusion is correct because of the step of induction, second- because of *, third- because of lemma 2, forth- because of Lemma 3 and fifth- because of lemma 4. So, the only thing we need is to prove these three lemmas.

Lemma 4

It is ell known fact, that in definition of BPP we can use, as a probability of mistake, 2^{-p_i}instead of (1-3/4), where p_i is a polynomial function of input length

Let M uses L as oracle, M is wrong with pr= $2^{-e(n)}$ $L \in BPP^A$

WLOG, we can consider, that the lengths of all branches of NTM L are equal and the number of its accesses to L is equal to l(n) in all branches

We can replace every access to oracle L by the branch of machine N(from the definition of BPP) with the probability of mistake $2^{-i(n)}$

The number of branches, which status will change is equal to $2^{-e(n)} + 1 - (1-2^{-i(n)})^{l(n)}$

that can be made less than 0.25 by selecting i(n) and e(n)

Lemma 2

L belongs to $\oplus P^{BPP^A}$ so exists NTM with oracle $B^A \in BPP^A$accepting every $x \in \{0,1\}^n$for odd number of y. $y \in \{0,1\}^{\gamma(n)}$

R is corresponding relationship (from the definition of parity P class)

$R \in P^{B^A} \in P^{BPP^A} \in BPP^{BPP^A} \in BPP^A$ Π is the corresponding Turing machine (with oracle A)

So $L = \{x \mid \#\{y : \#\{z : (x,y,z) \in L(\Pi^A)\} \geq (1-2^{-\pi(n)})2^{\varsigma(n)}\} is\, odd\}$

And we need: $L = \{x \mid \#\{z : \#\{y : (x,y,z) \in L(\Pi^A)\} is\, odd\} \geq 0.75 * 2^{\varsigma(n)}\}$

We can consider a table, (for fixed x, WLOG x is in L), in (y, z) we'll put result of Π for given (y, z).The number of rows with more than $(1-2^{-\pi(n)})2^{\varsigma(n)}\}$ ones is odd and the number of columns, which have 1 in intersection with this columns is more than $2^{\varsigma(n)} - 2^{-\pi(n)} * 2^{\varsigma(n)} * 2^{\gamma(n)}$

In other rows is not many 1,so the number of 'good' rows is

$2^{\varsigma(n)} - 2^{-\pi(n)} * 2^{\varsigma(n)} * 2^{\gamma(n)} - 2^{-\pi(n)} * 2^{\varsigma(n)} * 2^{\gamma(n)} \geq 0.75 * 2^{\varsigma(n)}$ for $\pi(n) \geq \gamma(n) + 3$

Lemma 3-

At first, we need to prove one more Lemma-

Lemma 5: $\oplus P = co\!-\!\oplus P$

It is obvious- for each non-deterministic Turing machine we can insert near the root an extra fork, in one subtree we'll just copy the old machine ,and in other- fictive computations of the same length with no accepting computations among them. So Parity P will come to co-parity P and visa versa

So, in lemma 3 we have to prove, that $\oplus P^{\oplus P} \subseteq \oplus P$

Having a Turing machine with oracle V from parity P we'll replace every oracle query by a fork with two branches –in first we'll substitute the tree of V and in second- $\underline{V}$, accepting those and only those words V reject.

From the leaves of this branches with ones we consider this one as the oracle answer and continue our computation. The Parity of the number of ones don't change, so the resulting tree will be also from Parity P