6.841 Advanced Complexity Theory March 18, 2009

Lecture 13

Lecturer: Madhu Sudan *Scribe: Alex Cornejo*

1 Overview of today's lecture

- Toda's Theorem: $\mathsf{PH} := \bigcup_{k\in\mathbb{N}} \sum_k^P \subseteq \mathsf{P}^{\#\mathsf{P}}$, steps:
- Prove some properties concerning $\exists C$, $\forall C$, $\oplus \cdot C$, $\mathsf{BP} \cdot C$
- Do some operator calculus to prove $\mathsf{PH} \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$.
- Prove that $\mathsf{BP} \cdot \oplus \cdot \mathsf{P} \subseteq \mathsf{P}^{\#\mathsf{P}}$

2 Review: Operator definitions

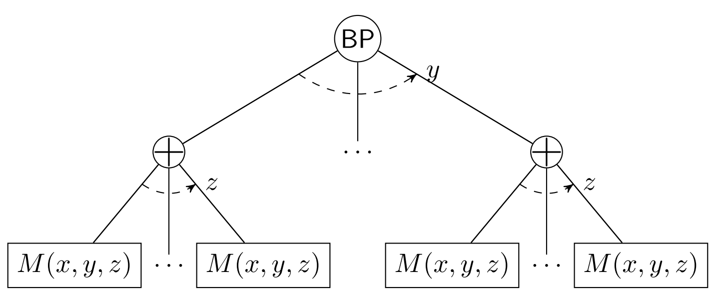
Let L be a language, C a complexity class and $q(n)$ some growing function of n.

$$\mathsf{BP}_{q(n)} \cdot L = \left\{\Pi_{yes}^{q(n)}(L), \Pi_{no}^{q(n)}(L)\right\}$$
$$\Pi_{yes}^{q(n)}(L) = \left\{x \mid \Pr\left[(x, y) \in L\right] \geq 1 - 2^{-q(n)}\right\}$$
$$\Pi_{no}^{q(n)}(L) = \left\{x \mid \Pr\left[(x, y) \in L\right] \leq 2^{-q(n)}\right\}$$

When we omit the $q(n)$ subscript we assume that $q(n) \in \mathsf{P}$.

$$\mathsf{BP} \cdot L = \{\Pi_{yes}(L), \Pi_{no}(L)\} \qquad \mathsf{BP} \cdot C = \{\mathsf{BP} \cdot L \mid L \in C\}$$
$$\oplus \cdot L = \{x \mid \#(y) \text{ s.t. } (x, y) \in L \text{ is even}\} \qquad \oplus \cdot C = \{\oplus \cdot L | L \in C\}$$
$$\overline{\oplus} \cdot L = \{x \mid \#(y) \text{ s.t. } (x, y) \in L \text{ is odd}\} \qquad \overline{\oplus} \cdot C = \{\overline{\oplus} \cdot L | L \in C\}$$
$$\exists \cdot L = \{x \mid \exists y \text{ s.t. } (x, y) \in L\} \qquad \exists \cdot C = \{\exists \cdot L | L \in C\}$$
$$\forall \cdot L = \{x \mid \forall y \text{ s.t. } (x, y) \in L\} \qquad \forall \cdot C = \{\forall \cdot L | L \in C\}$$

For this lecture, the correct way to think about an expression involving Toda's complexity operators is visualizing the execution tree that represents the expression. For example consider a language $L \in \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$, then for $x \in L$ we have the following tree:



3 Operator properties

To prove Toda's theorem we need to prove the following properties:

Property 1. $\oplus \cdot \oplus \cdot C = \oplus \cdot C$

Property 2. $\mathsf{BP} \cdot \mathsf{BP} \cdot C = \mathsf{BP} \cdot C$

Property 3. $\oplus \cdot \mathsf{BP} \cdot C = \mathsf{BP} \cdot \oplus \cdot C$

Property 4. $\exists \cdot C, \forall \cdot C \subseteq \mathsf{BP} \cdot \oplus \cdot C$

Observe that for our purposes it would suffice to prove them for $C \in \{\mathsf{P}, \oplus \cdot \mathsf{P}, \mathsf{BP} \cdot \oplus \cdot \mathsf{P}\}$. Lets warm up by proving $\oplus \cdot C = \overline{\oplus} \cdot C$.

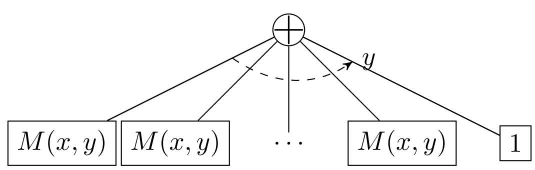
3.1 $\oplus \cdot C = \overline{\oplus} \cdot C$

We define the operator $\neg\cdot$ as:

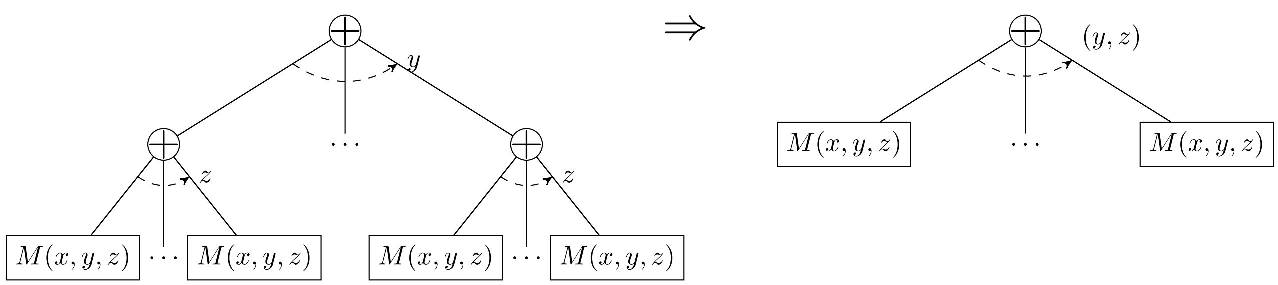
$$\neg \cdot L = \{(x, 0y) \mid (x, y) \in L\} \cup \{(x, 1)\}$$

Consider the language $\overline{\oplus} \cdot \neg \cdot L$, by unravelling the definition of $\overline{\oplus}\cdot$ we have $\overline{\oplus} \cdot \neg \cdot L = \{x | \#(y) \text{ s.t. } (x, y) \in \neg \cdot L \text{ is odd}\}$. However from the definition of $\neg\cdot$ if the number of y's such that $(x, y) \in \neg \cdot L$ is odd then the number of y's such that $(x, y) \in L$ is even, hence $\overline{\oplus} \cdot \neg \cdot L = \oplus \cdot L$. A symmetric argument proves that $\oplus \cdot \neg \cdot L = \overline{\oplus} \cdot L$.

The following diagram succinctly encodes the previous argument and demonstrates the equivalence of $\oplus\cdot$ and $\overline{\oplus}\cdot$.



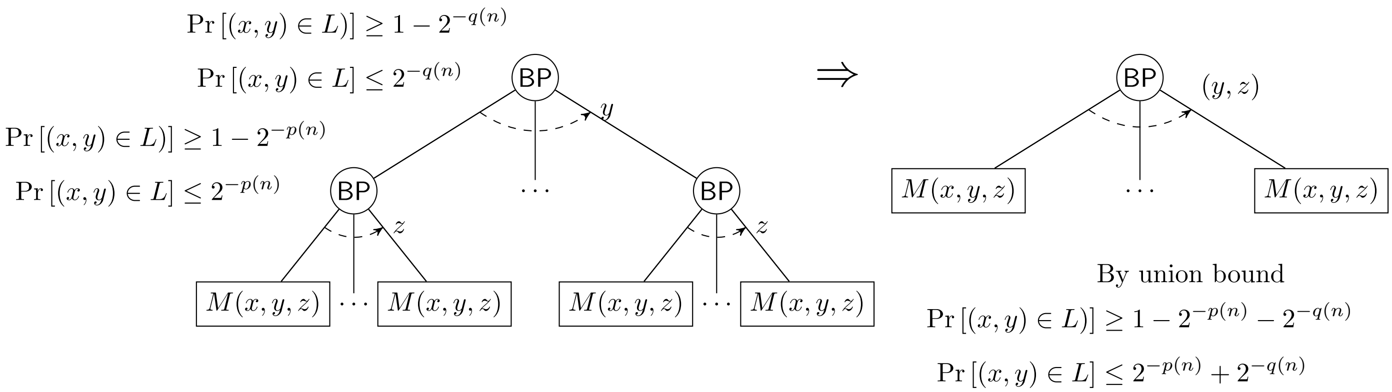
3.2 Property 1: $\oplus \cdot \oplus \cdot C = \oplus \cdot C$.



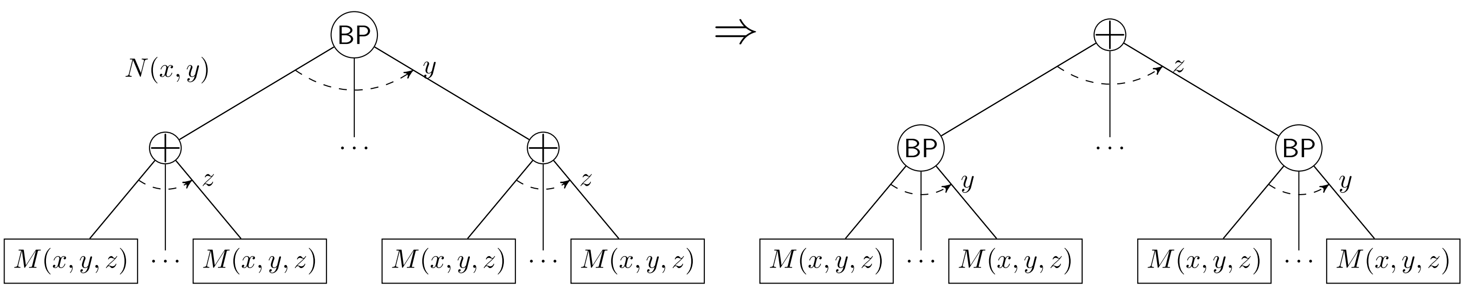
This is possible since $\oplus$ is associative and for our purposes the multiplicative increase in fan-in does not matter.

3.3 Property 2: $\mathsf{BP} \cdot \mathsf{BP} \cdot C = \mathsf{BP} \cdot C$.

Let $q(n)$ and $p(n)$ be polynomials, then observe



3.4 Property 3: $\oplus \cdot \mathsf{BP} \cdot C = \mathsf{BP} \cdot \oplus \cdot C$.



For a fixed x, the probability of choosing z such that $N(x,y)$ does not work is

$$\Pr_z[M(x,y,z) \neq N(x,y)] \leq 2^{-q(n)}$$

Hence the probability that there exists a y such that z does not work is

$$\begin{aligned}\Pr_z[\exists y \text{ s.t. } M(x,y,z) \neq N(x,y)] &\leq \sum_y \Pr_z[M(x,y,z) \neq N(x,y)] \\ &= 2^{|y|}2^{-q(n)} \\ &= 2^{|y|-q(n)}\end{aligned}$$

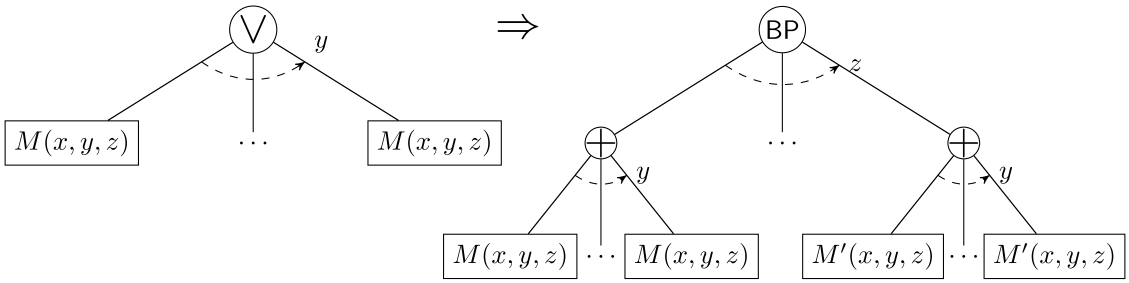
Therefore the probability that a particular z works for all choices of y is

$$\Pr_z[\forall y \; M(x,y,z) = N(x,y)] \geq 1 - 2^{|y|-q(n)}$$

So by choosing $q(n)$ sufficiently large we have $\mathsf{BP} \cdot \oplus \cdot C = \oplus \cdot \mathsf{BP} \cdot C$.

3.5 Property 4: $\exists \cdot C \subseteq \mathsf{BP} \cdot \oplus \cdot C$

In some sense the following proof is in the same spirit of the Razborov-Smolensky result we saw in lecture 6, since we are looking to replace an OR gate with a XOR and a BP gate.



For any language $L \in C$ by the argument of Valiant-Vazirani we can construct a machine M'

$$M'(x,y,z) := M(x,y) \wedge z(y) = \overline{0}$$

such that the language L' (where $L' \in C$) accepted by M' satisfies the following:

$$\exists y : (x,y) \in L \iff \Pr_z[\#(y) \text{ s.t. } (x,y,z) \in L' \text{ is even}] \geq \frac{1}{p(n)}$$
$$\forall y : (x,y) \notin L \iff \Pr_z[\#(y) \text{ s.t. } (x,y,z) \in L' \text{ is even}] = 0$$

Hence Valiant-Vazirani gives us what we want for *weak*-BP instead of BP. We can now use the standard amplification technique by using $z_1, \ldots, z_k$ instead of z and $y_1, \ldots, y_k$ instead of y.

$$\begin{aligned}\exists y : (x,y) \in L \iff \Pr_{z_1,\ldots,z_k}[\#(y_1,\ldots,y_k) \text{ s.t. } (x,y,z) \in L' \text{ is even}] &\geq 1 - (1 - \frac{1}{p(n)})^k \\ &\geq 1 - e^{-\frac{k}{p(n)}} \\ &= 1 - 2^{-\log_2 e \frac{k}{p(n)}}\end{aligned}$$

Hence we can choose $k = q(n)p(n)/\log_2 e$ for a polynomial $q(n)$ to get strong-BP. Observe that if C is closed under complement then we also get $\forall \cdot C \subseteq \mathsf{BP} \cdot \oplus \cdot C$ from the same argument.

4 Toda's Theorem

Theorem 1 (Toda's Theorem) $PH \subseteq P^{\#P}$

Proof Assume $\mathsf{BP} \cdot \oplus \cdot \mathsf{P} \subseteq \mathsf{P}^{\#\mathsf{P}}$ (proved in the next theorem), then to prove the statement it suffices to prove that $\mathsf{PH} \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$. From the definition of the polynomial hierarchy we have $\mathsf{PH} := \bigcup_{k \in \mathbb{N}} \Sigma_k^P$, we proceed by induction on k.

Base case. This is trivial since by definition $\Sigma_k^P = \Pi_k^P = P$.

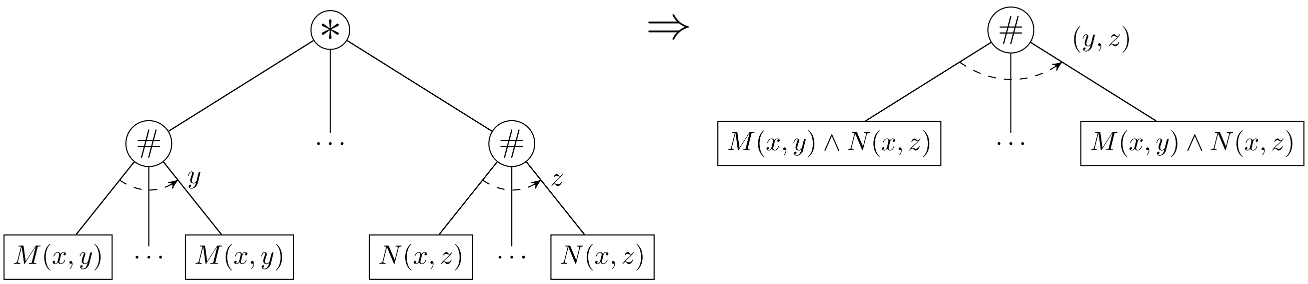
Inductive step. As inductive hypothesis we assume that $\Sigma_k^P, \Pi_k^P \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$.

$$\begin{aligned}\Sigma_k^P &\subseteq \exists \cdot \Pi_{k-1}^P && \text{by definition of the hierarchy} \\ &\subseteq \exists \cdot \mathsf{BP} \cdot \oplus \cdot \mathsf{P} && \text{by hypothesis} \\ &\subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{BP} \cdot \oplus \cdot \mathsf{P} && \text{by property 4} \\ &\subseteq \mathsf{BP} \cdot \mathsf{BP} \cdot \oplus \cdot \oplus \cdot \mathsf{P} && \text{by property 3} \\ &\subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P} && \text{by property 1 and 2}\end{aligned}$$

■

Theorem 2 $BP \cdot \oplus \cdot P \subseteq P^{\#P}$

Proof Suppose we had an operator $\#\cdot$ that allowed us to count the number of accepting paths, then clearly we could easily use such an operator to add and multiply the number of accepting paths.



Hence $\#\mathsf{P}$ is almost a ring, except that we do not have a way to subtract or divide the number of accepting paths. Recalling the definition of $\#\mathsf{P}$

$$f \in \#\mathsf{P} \iff \exists M \in \mathsf{NP} \text{ s.t. } f(x) = \#(y) \text{ s.t. } M(x, y) \text{ accepts}$$

And we have that

$$\begin{aligned} f_1, f_2 \in \#\mathsf{P} \Rightarrow & f_1 + f_2 \in \#\mathsf{P} \\ & f_1 * f_2 \in \#\mathsf{P} \end{aligned}$$

Hence given $f \in \#\mathsf{P}$ we can construct any polynomial g over f, as long as the degrees and coefficients are bounded by a polynomial (*i.e.* $g(x) = f(x)^2 + 3f(x) + 2f(x)^{10}$).

Given a language $L \in \mathsf{P}$ let $L' = BP \oplus \cdot L$, then by definition we have

$$\begin{aligned} x \in L' &\iff \Pr_y [\#(y) \text{ s.t. } (x, y) \in L = 0 \bmod 2)] \geq 1 - 2^{-q(n)} \\ x \notin L' &\iff \Pr_y [\#(y) \text{ s.t. } (x, y) \in L = 1 \bmod 2)] \leq 2^{-q(n)} \end{aligned}$$

Unfortunately we cannot count module 2, but what if we could construct a polynomial p such that:

$$\begin{aligned} p(x) = 0 \bmod 2^k &\Rightarrow \#(y) \text{ s.t. } (x, y) \in L = 0 \bmod 2 \\ p(x) = 1 \bmod 2^k &\Rightarrow \#(y) \text{ s.t. } (x, y) \in L = 1 \bmod 2 \end{aligned}$$

Assume there are 2^m distinct y's, then we want k large enough to ensure that for the space of y's we care about $p(x) \bmod 2^k = p(x)$, clearly $k = 2m$ suffices.

We could build such a polynomial p applying k times a polynomial f such that:

$$\begin{aligned} f(x) = 0 \bmod 2^{2z} &\Rightarrow x = 0 \bmod 2^z \\ f(x) = 1 \bmod 2^{2z} &\Rightarrow x = 1 \bmod 2^z \end{aligned}$$

Unfortunately a polynomial with this properties does not exist. However if we replace 1 with -1 then there is such a polynomial, namely $f(x) = 4x^3 + 3x^4$, and therefore we can construct p with f by a recursion of depth k.

Therefore it follows that $L' \in \mathsf{P}^{\#\mathsf{P}}$ since we can remove the probability operator by just counting the number of accepting states using p and depending if they are $\geq (1 - 2^{-q(n)})2^m$ or $\leq 2^{-1}2^m$ decide if $x \in L'$.

■