Derandomization

- I.e. solving a problem without using randomness.

<u>Qn</u>: BPP = P ?

- We will now show that this question is intimately connected to proving "lower bounds" or "hardness"!

<u>Theorem</u> (Impagliazzo & Kabanets '03): $\mathrm{PIT} \in \mathrm{P} \Rightarrow$ (BPP=P $\Rightarrow$)
$\mathrm{NEXP} \not\subseteq \mathrm{P/poly}$ or $\mathrm{per} \notin \mathrm{AlgP/poly}$.

<u>Remark</u>: • <u>per</u> is the functional problem of computing the <u>permanent</u> of a given matrix $A_{n \times n}$.

• $\underline{\mathrm{per}(A)} := \sum_{\sigma \in \mathrm{Sym}(n)} A_{1\sigma(1)} \cdots A_{n\sigma(n)}$

• Here we are interested in $\mathbb{F} = \mathbb{Q}$.

• The theorem connects "algorithm" with "nonexistence" of one.

- The involved proof depends on several older results.

per as oracle

Lemma 1: $PIT \in P$ & $per \in AlgP/poly \Rightarrow P^{per} \subseteq NP$.

Proof:

- Idea — "Guess" the poly-sized circuit for per. (& verify via self-reducibility)
- We can expand the permanent of an $n \times n$ matrix, $per_n(A)$, by the first row:

$$per_n(A) = \sum_{i\in[n]} A_{1i} \cdot per_{n-1}(A'_{1i}) \,,$$

where A'_{1i} is the submatrix of A after removing the first row & the i-th column.

- Given a circuit $C_{(n-1)^2}$ for per_{n-1}, we can "guess" a circuit C_{n^2} for $per_n(A)$, & using PIT algorithm verify

$$C_{n^2}(A) \stackrel{?}{=} \sum_{i\in[n]} A_{1i} \cdot C_{(n-1)^2}(A'_{1i}) \,.$$

- This guess-&-verify process proves $C_{n^2}(A) = \mathrm{per}_n(A)$, by induction on n.

- $\Rightarrow$ For any $L \in P^{\mathrm{per}}$ we can first guess the poly-sized circuit C_{n^2} for per_n, verify using PIT, & use C_{n^2} instead of the oracle.

  $\Rightarrow L \in NP \Rightarrow P^{\mathrm{per}} \subseteq NP$. $\square$

- The "strange" assumption of $\mathrm{per} \in \mathrm{AlgP/poly}$ gives the strange conclusion $P^{\mathrm{per}} \subseteq NP$.

- We now intend to make another strange assumption ($NEXP \subseteq P/poly$) & deduce that $NEXP \subseteq P^{\mathrm{per}} \subseteq NP$.

  This contradiction (nondet. time hierarchy) will prove the main theorem.

- This will require a crash course in

quantifier-based & interaction-based complexity classes.

Definition: • $\Sigma_0 := P$, $\Sigma_1 := NP$, $\Sigma_2 := NP^{NP}$, $\Sigma_3 := NP^{\Sigma_2}$, ...

oracle-based

• Formally, $L \in \Sigma_2$ if $\exists$ poly-time NDTM "solving" L using SAT as oracle.

• As having SAT as an oracle also means having $\overline{SAT}$ as an oracle, it could be shown that Σ_2 has the following

$\exists, \forall \rightarrow$ quantifier-based definition:

$L \in \Sigma_2$ if $\exists$ poly-time TM N s.t. $\forall x$,

$x \in L$ iff $\exists y_1 \forall y_2$, $N(x, y_1, y_2) = 1$.

size = poly($|x|$)

• Similarly, Σ_3 can be defined via an alternating sequence of 3 quantifiers: $\exists y_1 \forall y_2 \exists y_3$.

• Polynomial hierarchy $PH := \bigcup_{i \in \mathbb{N}} \Sigma_i$.

▷ PH ⊆ Pspace.

Pf: Systematically go through all the possibilities of the quantified strings. □

OPEN: $\Sigma_0 \subsetneq \Sigma_1 \subsetneq \Sigma_2 \subsetneq \cdots\cdots \subsetneq PH \subsetneq Pspace$?

- Instead of $\exists, \forall$ we could use other types of quantifiers.

  e.g. '$\mathcal{M}$'-quantifier for most.

Definition: • The statement "$\mathcal{M} y \in \{0,1\}^n, N(y)=1$" is true iff $\Pr_{y \in \{0,1\}^n}[N(y)=1] \geq 3/4$.

TM

• k-alternations of $\exists$ & $\mathcal{M}$ give us the class AM[k]: $L \in AM[k]$ if $\forall x$,

$$x \in L \text{ iff } \underbrace{\mathcal{M} y_1 \exists y_2 \mathcal{M} y_3 \cdots\cdots}_{k}, N(x, y_1, \dots, y_k) = 1$$

Arthur (verifier)

Merlin (prover)

• Starting the alternations with '$\exists$' gives us the class MA[k].

- The "physical" interpretation of these classes is by considering k-rounds of <u>interaction</u> between:
  the king <u>Arthur</u> (randomized verifier)
  his advisor <u>Merlin</u> (unreliable prover)

- AM[1] : Arthur x, y_1 — Merlin (not used)

▷ AM[1] = BPP.

- MA[1]: Arthur $x \xleftarrow{\;y_1\;}$ Merlin

▷ MA[1] = NP.

- MA[2] : Arthur $x, y_2 \xleftarrow{\;y_1\;}$ Merlin

⇒ like NP with a randomized verifier.

- This we also call <u>MA</u>.

- AM[2]: Arthur $x, y_1 \xleftarrow{\;y_2\;}$ Merlin

- This we also call <u>AM</u>.

<u>Definition</u>: If we make k a variable then we get $IP := \bigcup_{c \in \mathbb{R}_{>0}} AM[n^c]$.

(interactive protocol)

▷ All these classes are in Pspace.

<u>Theorem</u> (Shamir 1990): $IP = \text{Pspace}$.

<u>Pf sketch</u>:

- It suffices to show that Pspace $\subseteq$ IP.
- We pick a Pspace-complete problem — Quantified boolean formula (<u>QBF</u>):

$x_i \in \{0,1\}$ → Given a formula $\psi := Q_1 x_1 \cdots Q x_n \, \phi(\bar{x})$, where Q_i's are quantifiers $\{\exists, \forall\}$ & ϕ is a boolean formula. Test whether ψ is true.

▷ QBF is Pspace-complete.

- We intend to show that QBF $\in$ IP.

This protocol will be <u>algebraic</u>.

- For the boolean formula ϕ, we define an <u>arithmetized</u> version $P_\phi \in \mathbb{Q}[x_1,\ldots,x_n]$.

Eg. $\phi := (x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3 \vee x_4)$ arithmetizes to $T_\phi := \left(1-(1-x_1)(1-x_2)\right) \cdot \left(1 - x_2(1-x_3)(1-x_4)\right)$.

The key property being that $\phi(a_1,\ldots,a_n) = \text{true}$ iff $T_\phi(a_1,\ldots,a_n) = 1$.

- Clearly, arithmetization is <u>easy</u> to do.
- We extend this to quantifiers as:

$\forall x_i$ converts to $\prod_{x_i \in \{0,1\}}$,

$\exists x_i$ " " $\sum_{x_i \in \{0,1\}}$

By going mod a <u>random</u> prime, wlog P_ψ is an integer in $[0, 2^n]$

- Finally, $\psi := Q_1 x_1 \cdots Q_n x_n\, \phi(\bar{x})$ converts to

$P_\psi := \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_n\, P_\phi(x_1,\ldots,x_n)$,

where $\tilde{Q}_i$ is $\prod_{x_i}$ or $\sum_{x_i}$.

▷ ψ is true iff $P_\psi \neq 0$.

- Now to convince Arthur that ψ = true, Merlin will try to prove: $P_\psi = k \in [2^n]$ in n rounds of interaction.
(Merlin sends partial polynomial related to P_ψ. Arthur does PIT.)

0) Merlin sends $k>0$, claiming $P_\psi = k$.

1) If $n=1$ then Arthur accepts iff:
$$Q_1 = \forall \ \& \ P_\phi(0) \cdot P_\phi(1) = k,$$
$$Q_1 = \exists \ \& \ P_\phi(0) + P_\phi(1) = k.$$

deg s can be made small

2) If $n>1$ then Merlin sends $s(x_1)$, claiming it to be $\tilde{Q}_2 \tilde{Q}_3 \cdots \tilde{Q}_n P_\phi(x_1, \ldots, x_n)$.

3) Arthur tests: $Q_1 = \forall \ \& \ s(0) \cdot s(1) = k$,
$Q_1 = \exists \ \& \ s(0) + s(1) = k$,
and for a random α, verifies
$$s(\alpha) = \tilde{Q}_2 \cdots \tilde{Q}_n P_\phi(\alpha, x_2, \ldots, x_n).$$
done recursively by interacting with Merlin

$\square$

- Exercise: Show that if Merlin errs then Arthur will detect it with high probability.

- Now we prove some lemmas to be used later.

Lemma (Babai, Fortnow, Nisan, Wigderson 1993):

$$\mathsf{EXP} \subseteq \mathsf{P/poly} \Rightarrow \mathsf{EXP} = \mathsf{MA}.$$

Proof:

- Suppose $\mathsf{EXP} \subseteq \mathsf{P/poly}$.
- We will first show that $\mathsf{EXP} \subseteq \Sigma_2$:

- Let $L \in \mathsf{EXP}$ & N be its exp-time TM.
- Since the j-th bit in the i-th configuration of $N(x)$ is computable in EXP, $\exists$ poly-sized circuit $C(x,i,j)$ computing it.
- Now, $x \in L \Leftrightarrow \exists C, \forall i, \forall j, [C(x,i,j) \to C(x,i+1,j)$ is a valid step of $N]$.

- This just means that $L \in \Sigma_2$.
  $\Rightarrow \quad EXP \subseteq \Sigma_2$.
  $\Rightarrow \quad EXP = \Sigma_2$.

- Also, $\Sigma_2 \subseteq Pspace = IP \subseteq EXP$.
  $\Rightarrow \quad Pspace = IP = EXP \subseteq P/poly$.

- Thus, any $L \in EXP$ has an interactive protocol.
  Moreover, Merlin can be seen as a Pspace-machine, hence, can be simulated by a poly-sized circuits family $\{C_n\}_n$.

- This suggests a single-round protocol to convince Arthur that $x \in L$:
  (1) Merlin sends a circuit C, claiming to be C_n, $n := |x|$.
  (2) Arthur runs the protocol on x, using C instead of Merlin.

$\Rightarrow L \in MA$

$\Rightarrow \quad EXP \subseteq MA \Rightarrow EXP = MA.$ $\square$

- The final lemma is the most advanced. We will prove it after we have covered pseudo-random generators.

<u>Lemma</u> (Impagliazzo, Kabanets, Wigderson 2001):

$$NEXP \subseteq P/poly \Rightarrow NEXP = EXP.$$

- Finally, we can prove the PIT-theorem:

$$PIT \in P \Rightarrow NEXP \not\subseteq P/poly \text{ OR } per \notin alg\, P/poly.$$

<u>Proof:</u>

- Suppose $PIT \in P$, $NEXP \subseteq P/poly$.
  $\Rightarrow NEXP = EXP = MA$.
- Also, $MA \subseteq PH \subseteq P^{per}$ [Toda's theorem]
  $\Rightarrow NEXP \subseteq P^{per}$.

- Assuming $per \in P/poly$ implies $P^{per} \subseteq NP$.

$\Rightarrow \quad NEXP \subseteq NP$,
which contradicts the nondet.-time hierarchy.

- Thus, either $NEXP \not\subseteq P/poly$ OR
  $per \notin algP/poly$. $\square$