Today

- Toda's Theorem
- Part 1: $\mathrm{PH} \subseteq \mathrm{BP} \cdot \bigoplus \cdot P$.
- Part 2: $\mathrm{BP} \cdot \bigoplus \cdot P \subseteq \mathsf{P}^{\#\mathsf{P}}$

Operators vs. Constant Depth Circuits

- Think of:
  - $\exists$ operator as layer of OR gates.
  - $\forall$ operator as layer of AND gates.
  - BP operator as approximate Majority gate.
  - $\bigoplus$ operator as a parity gate.
- Complexity classes become constant depth circuit.
- Part 1 of Toda's theorem says constant depth AND-OR circuit can be replaced by a depth two circuit circuit with parity gates at bottom level and an approximate majority at top level, uniformly.

Breakdown of Part 1

$$\begin{aligned} &\exists \cdot \mathrm{BP} \cdot \bigoplus \cdot P \\ &\quad\subseteq \mathrm{BP} \cdot \bigoplus \cdot \mathrm{BP} \cdot \bigoplus \cdot P \\ &\quad\subseteq \mathrm{BP} \cdot \mathrm{BP} \cdot \bigoplus \cdot \bigoplus \cdot P \\ &\quad\subseteq \mathrm{BP} \cdot \mathrm{BP} \cdot \bigoplus \cdot P \\ &\quad\subseteq \mathrm{BP} \cdot \bigoplus \cdot P \end{aligned}$$

Rest follows by closure under complementation and induction.

Part 1, Step 1

Write $\exists \cdot \mathrm{BP} \cdot \bigoplus \cdot P$

as

$\exists x, \mathrm{BP}\, y, \bigoplus z, M(w, x, y, z)$

or as

$\exists x, N(w, x)$ where $N \in \mathrm{BP} \cdot \bigoplus \cdot P$.

By Valiant-Vazirani & amplification, we note this condition can be written as

$\mathrm{BP}_{\mathbf{h}} \bigoplus_{\mathbf{x},\mathbf{b},c} N_1(w, \mathbf{h}, \mathbf{x}, \mathbf{b}, c)$

where $\mathbf{h}$ is a sequence of m hash functions, $\mathbf{x}$ is m non-det. choices for N, $\mathbf{b}$ is m bits, and c is a bit.

$N_1(w, \mathbf{h}, \mathbf{x}, \mathbf{b}, c)$ accepts if the input is all 0s or if $c = 1$ and for all i, $N_2(w, h_i, x_i, b_i)$ accepts.

$N_2(w, h_i, x_i, b_i)$ accepts if the input is all 0s or if $b_i = 1$ and $h_i(x_i) = 1$ and $N(w, x_i)$.

To conclude, suffices to observe that N_1's computation is in $\mathrm{BP} \cdot \bigoplus \cdot P$.

Part 1, Steps 2, 3 & 4

Nothing special: Just do blind actions and for various choice of parameters, things work.

Step 2 Switch 2^a-ary parity gate with 2^b-ary BP gates of error 2^{-c} and get a BP gate that errs with probability 2^{b-c}.

Step 3 Collapse parity gates and it just works $\bigoplus_y \bigoplus_z f(y,z) = \bigoplus_{y,z} f(y,z)$.

Step 4 Collapse BP gates: $\mathrm{BP}_y \, \mathrm{BP}_z \, f(y,z)$ vs. $\mathrm{BP}_{y,z} \, f(y,z)$? If BP_y errs with probability ϵ and BP_z errs with probability δ then $\mathrm{BP}_{y,z}$ errs with probability at most $\epsilon + \delta$.

Overview

- Concludes Part 1 of Toda's Theorem.
- For Part 2, need to understand some arithmetic games one can play with $\#$ accepting paths.

Arithmetic games

- If non-deterministic machine M_1 on input w_1 has n_1 accepting paths, and M_2 on input w_2 has n_2 accepting paths, then can create machines + inputs that have $n_1 + n_2$, or $n_1 \times n_2$ accepting paths.
- W.l.o.g. consider circuits. Have circuits C_1, C_2 ($C_i(\cdot) = M_i(w_i, \cdot)$) taking n-bit inputs and accepting n_1 and n_2 inputs respectively.
- Then, circuit C_3 given by $C_3(x, y) = C_1(x) \wedge C_2(x)$ accepts $n_1 n_2$ inputs.
- And, C_4 given by $C_4(x, b) = (b \wedge C_1(x)) \vee (\bar{b} \wedge C_2(x))$ has $n_1 + n_2$ accepting inputs.

More arithmetic

- Can also construction circuits with any fixed number of accepting inputs.
- So given any polynomial p with positive coefficients, and circuit C with N accepting inputs, can construct C' with $p(N)$ accepting inputs. Furthermore size of $C' = O(|p| \cdot |C|)$.
- If p is a constant degree polynomial with constant coefficients, can apply this process $O(\log n)$ times.

Will use the last parts later, but first show how to amplify.

"Boosting" modular counts

- Suppose $a = b(\bmod 2^{2^c})$ for $b \in \{0, -1\}$.
- Then for $h(a) = 3a^4 + 4a^3$ have $h(a) = b(\bmod 2^{2^{c+1}})$.
- Let $h^{(i)}(a) = h(h^{(i-1)}(a)$, where $h^{(0)}(a) = a$.
- Let $t = O(\log m)$. Let C' be the circuit with $h^{(t)}(\#_x C(x, y))$ accepting inputs. (Can construct such C' in polynomial time.).
- C' is what we need.

QED. (Done with Toda's theorem.)

Polynomial magic=?

How would we come up with the polynomial h?

- Requirements:
  - $h(a) = b(\mod 2^{2^{c+1}})$ for $b \in \{0, -1\}$.
  - Coefficients of h non-negative.
- First condition says $a^2 | h(a)$ and $(a + 1)^2 | h(a) + 1$. Natural choice (to make coeff. of a^1 disappear), $h_1(a) + 1 = (a+1)^2(a-1)^2 = a^4 - 2a^2 + 1$. Now have $h_2(a) = a^4 - 2a^2$. Satisfies first condition, but violates second.
- To make coefficients positive, add a (large multiple of) polynomial with +ve coefficients that is 0 on a^2 and $(a + 1)^2$. Simple choice $= a^2(a + 1)^2$.
- New candidate $h_2(a) = h_1(a) + 2 \cdot a^2(a + 1)^2 = 3a^4 + 4a^3$.