- Thus, "hardness $\Rightarrow$ prg".
  Is there a converse?

Claim: If $\exists$ $S(\ell)$-prg then $\exists f \in E$ s.t. $H_{wrs}(f_n) > n^3$.

$S(\ell) > \ell$

Proof:

- Let $G: \{0,1\}^\ell \to \{0,1\}^n$ be an $S(\ell)$-prg.
- Consider the function $f_n: \{0,1\}^n \to \{0,1\}$ s.t. $f_n(x) = 1$ iff $x \in im(G)$. Clearly, $f \in E$.
- Let C_n be the smallest circuit computing f_n.

- Also, $Pr[C_n(G(U_\ell)) = 1] = 1$
  while $Pr[C_n(U_n) = 1] \leq 2^\ell / 2^n \leq 1/2$
  $\Rightarrow$ C_n distinguishes $G(U_\ell)$ from U_n well.
  $\Rightarrow$ $size(C_n) > S(\ell)^3 = n^3$. $\square$

- We will now see more impressive applications of prg in complexity:

Partial derandomization

<u>Theorem</u> (Impagliazzo, Wigderson 1998): If $BPP \neq EXP$ then $\forall L \in BPP$, $\exists$ subexponential-time algorithm A s.t. for ∞ly-many n's:

$$\Pr_{x \in \{0,1\}^n} [A(x) = L(x)] \geq 1 - \frac{1}{n} .$$

the <u>det. algo. A</u> is right on <u>average</u>.

<u>Proof sketch</u>:

- If $EXP \not\subseteq P/poly$ then $\exists f \in EXP$ with $H_{wrs}(f) = n^{\omega(1)}$.

  Later we will see how to amplify this to get an $f' \in EXP$ with $H_{avg}(f') = n^{\omega(1)}$.

  NW-theorem then implies $BPP \subseteq Subexp$.

- So, assume $EXP \subseteq P/poly$.

  EXP ⊆ MA ⊆ PH ⊆ EXP → Then (recall the initial lectures), $EXP = PH$.

  This, with Toda's theorem ($PH \subseteq P^{per}$) means that $P^{per} = EXP$.

  $\Rightarrow P^{per} \not\subseteq BPP$.

- This, essentially, says that per is hard & we will use it to define $\underline{G} := NW_g^{per} : \{0,1\}^{\ell} \to \{0,1\}^n$, with a superpoly- stretch.

- For an $L \in BPP$, if $B(x,r)$ is the randomized algorithm solving L, then we define the promised A as:

$$A(x) := \text{majority}\{B(x, G(U_\ell))\}.$$

i.e. all except finitely many

- Suppose the Thm. statement is false. Then, for almost all n's: $\Pr_{x \in U_n}[A(x) = L(x)] < 1 - \frac{1}{n}$.

$$\Rightarrow \Pr_{x \in U_n}[\text{maj}\{B(x, G(U_\ell))\} \neq \text{maj}\{B(x, U_n)\}] > 1/n.$$

$\Rightarrow$ We can fix $x = s_n \in \{0,1\}^n$ s.t. the circuit family $\{D_n := B(s_n, \cdot) \mid n\}$ can distinguish, $G(U_\ell)$ from U_n, well.

- In fact, the circuit D_n can be constructed by a randomized poly-time algorithm (whp).

- Recalling the properties of $G = NW_f^{per}$, we can deduce that $\exists$ randomized poly-time algorithm T that can "learn" per_N, i.e.
  Given oracle access to per_N, T runs in poly(N)-time & produces a poly(N)-sized circuit computing per_N.

- Now we can remove the need for the oracle because per_N is self-reducible:
  $$per_N(M) = \sum_{i\in[N]} M_{1i} \cdot per_{N-1}(minor_{1i}(M)).$$

$\Rightarrow$ T can build $per_1, per_2, \ldots, per_N$ recursively.

$\Rightarrow$ $P^{per} \subseteq BPP$, which is a contradiction.
$\Rightarrow$ $A(x)$ is "mostly" correct.

$\square$

– The next prg application completes the proof of "PIT $\in$ P $\Rightarrow$ lower bounds".