$P: \{0,1\}^n \rightarrow \{0,1\}$

A circuit $C$ of size $S(n)$

$$\text{Prob}[C(x) = f(x)] \leq \frac{1}{2} + \frac{1}{S(n)}$$

**BFNW:** worst-case \rightarrow average-case hardness

$A(x,r) \in \{0,1\}^T$ uniform at rank $T = \text{run-time}$

$s = \{s_1, \ldots, s_k\}$

$$\text{Goal:} \quad \text{Prob}[A(x,s_i) = A(x,r)] \leq \frac{1}{4}, (w \leq \frac{1}{3})$$

Equivalently,$$A,C,01 \vdash$$

$$|\text{Prob}(C(s_i)) - \text{Prob}(C(r))| \leq \frac{1}{4}$$

$\Rightarrow$ Deterministic simulation of $A \circ C$ +

$$(\text{Time to construct } S) + K \cdot \text{poly}(T).$$

**YaO:** Next Bit Test: very efficiently applicable (original construction)

- but requires a cryptographic assumption.

$t = \text{"(T,e)-pseudorandom"}.$

**Q1** BPP = RP + advice:

**Q2** Universal search algorithms \Rightarrow deterministic if logspace.

Next bit circuit for $s$, $C$, at position $i$:

$s_i$-bit position:

$C(s_i, \ldots, s_{i-1})$ is a "guess" for $s_i$.

$$\text{succ}(C,i) = \text{Prob}(C(s_i, s_{i-1}) = s_i) - \frac{1}{2}$$
Let: Equivalence of \((T,E)\)-pseudorandom + nonexistence of a small next-bit-circuit

Proof by hybrid argument

\[ s = \beta \quad s_0 = \gamma \quad \text{pseudorandom} \]

\[ S_a : \text{Pick } s_i, \text{ randomize } F \]

\[ \text{output } \gamma \cdot \gamma_i, \gamma_{i_1} \cdots \gamma_{i_r} \]

Assume: \(C(s) - C(R) \geq E\) (so not pseudorandom)

\[ (C(s_T) - C(s_{T-1})) + \ldots + (C(s_1) - C(s_0)) \geq E \]

\[ \text{ sup } C(s_T) - C(s_{T-1}) \geq E/T, \]

\[ C' \text{- randomized circuit.} \]

\[ \begin{cases} \text{pick } r_i \cdot \gamma_i \text{ at random} \\ \text{if } C(s_0, \ldots, s_{i-1}, \gamma_i > T \text{ output } r_i \\ \text{otherwise output } 1 - r_i \end{cases} \]

and do the usual calculations. (Hard direction)

Round about way to sample in bit couples:

Use a hard function

Choose \(x_0, x_T \in \{0,1\}^n\)

Calculate \(f(x_0), \ldots, f(x_T) \in \{0,1\}^T\).

Choose \(x_0, x_T\) - computationally (partially) independent

Knowing any \(g_i(x_i)\), \(g_i(x_i)\) does not substantially improve guessing \(f(x_i)\).

Example: \(T = 2\)

Suppose \(C(x, g(x_i)) = f(x) \) with probability \(f\)

(1) \(T \rightarrow T - \log f\), let \(x_1, x_2, \ldots\) maintain this advantage

(2) Recycle \(g(x_i)\) for \(T\) very commonly when \(x_1, x_2\)

(3) Use \(C'\) to compute \(f(x_i)\).
NW construction

\[ m = \# \text{ of bits to sample}, \quad z_1, \ldots, z_m \]
\[ A_i \in \{1, \ldots, m\}, \quad A_i = \{s_{z_i}, \ldots, s_{z_m}\} \]
\[ x_i = z_{a_i}, \quad z_{a_i} \in \{0,1\}^n, \quad a_i \in A_i \]

\[ \forall i, j, |A_i \cap A_j| \leq \ell \quad (\ell \leq \log T). \]

Assume further, given \( g(x_1), \ldots, g(x_m) \), still hard to predict \( s(x_i) \).

Suppose \( C'(x_i, g_1(x_i), \ldots, g_k(x_i)) \) models \( s_i(x_i) \)

by part 'c + e'.

1. Fix bits in \( A_i \) to maintain the advantage.
2. For each \( x_i \), \( \exists \) at most \( 2^\ell \) possible rules over the \( 2^\ell \) \( x_j \).
   - Give a table of all these rules.
3. \( C \) simulates \( C' \): (a) Look up each \( x_j \) in the table.
   - (b) Feed the rule to \( C' \).

Argue as before.

\[ m \ll T \quad (m = n^2 \text{ or } m = O(n)) \]
\[ \ell \leq \log T \quad (\text{small}). \quad \ell \text{ cm of course.} \]

Let \( p \) be prime \n < p < 2n \quad (\text{say})
\n\[ 1 + m = p^2, \quad m < 2^\ell \quad (\text{say}) \]
\[ m = p^2 \quad \text{and } m = p \cdot n \]
\[ w = m \cdot l \]
\[ \text{For } q \text{ a degree } \ell, \text{ univariate polynomial over } \mathbb{Z}_p \]
\[ \text{let } A_q = \{ q(s_y(i)) : i = 1, \ldots, n \} \]
\[ |A_q| = n \]
\[ |A_q \cap A_q'| \leq \ell. \]

Need \( T \leq p^2 \) to have enough polynomials

\[ \ell \approx \frac{\log T}{\log n} \ll T. \]
Theorem (for instance) "Low-end Version"

If \( \exists f \in \text{EXP} \text{ which is } s(n)\text{-hard \text{ and } } s(n) = n^{o(1)} \)

then \( \forall e > 0 \) \( \text{BPP} = \text{DTIME} \left( 2^{n^e} \right) \).

Proof

Let \( n = T^S \), so \( f_n \in T^{o(1)} \) hard.

Choose \( A_1, \ldots, A_T \) such that \( |A_i \cap A_j| \leq \frac{2^n}{2 n^c} = O(1) \).

Let \( m = n^2 = T^{2S} \).

For all \( x_1, \ldots, x_m \), compute \( s = P(x_1) \ldots P(x_T) \)

in total time \( 2^n \cdot T \cdot 2^m = 2^{n + 2S} \)

i.e. subexponential in \( T \).

Theorem "High-end"

If \( \exists f \in \text{EXP}, f \in 2^{s(n)} \text{-hard,} \)

then \( \text{BPP} = \text{P} \).