1 Definitions

Def 1.1 Let $t(n), r(n) : \mathsf{N} \rightarrow \mathsf{N}$ and $\text{err}(n) : \mathsf{N} \rightarrow \mathsf{Q} \cap (0, 1/2)$. (Think of $t(n), r(n)$ as poly and $\text{err}(n) = 1/4$.) Let $\text{BPP}(t(n), r(n), \text{err}(n))$ be the set of all $A \subseteq \{0,1\}^*$ such that there exists TM M that runs in time $t(n)$ on inputs of the form (x, y) where $|x| = n$ and $|y| = r(n)$. such that the following occurs. Let $x \in \{0,1\}^n$.

1. if $x \in A$ then, for at least $1 - \text{err}(n)$ of $y \in \{0,1\}^{r(n)}$, $M(x, y) = 1$.
2. if $x \notin A$ then, for at least $1 - \text{err}(n)$ of $y \in \{0,1\}^{r(n)}$, $M(x, y) = 0$.

We also define $\text{BPP} = \bigcup_{k=1}^{\infty} \text{BPP}(n^k, n^k, \frac{1}{4})$.

Def 1.2 Let $L : \mathsf{N} \rightarrow \mathsf{N}$ (think Log), $s : \mathsf{N} \rightarrow \mathsf{N}$ (think $2^{\epsilon n}$), and $\text{diff} : \mathsf{N} \rightarrow \mathsf{Q} \cap (0, 1/2)$ (think $\frac{1}{\text{poly}}$). Assume that, for all n, G maps $\{0,1\}^{L(n)}$ into $\{0,1\}^n$. G is $(L(n), s(n), \text{diff}(n))$-*pseudorandom* if

1. (Informally) For all n the set $\{G_{L(n)}(z) : z \in \{0,1\}^{L(n)}\}$ "looks like" $\{0,1\}^n$.
2. (Formally) For almost all n, for every $s(n)$-sized circuit C_n,

   $$|\Pr(C_n(y) = 1 : y \in \{0,1\}^n) - \Pr(C_n(G_n(z)) = 1 : z \in \{0,1\}^{L(n)}| < \text{diff}(n).$$

   (so no $s(n)$-sized circuit can tell the two sets apart, up to $\text{diff}(n)$. When assuming this is not true we freely use 0 intead of 1 and/or do not use the absolute value signs.

Note 1.3 If we say that $G \in \text{DTIME}(t(n))$ we mean that it runs in time $t(n)$ where n is the length of the *output*.

Def 1.4 Let $L : \mathsf{N} \rightarrow \mathsf{N}$ (think Log), $s : \mathsf{N} \rightarrow \mathsf{N}$ (think $2^{\epsilon n}$), and $\text{eps} : \mathsf{N} \rightarrow \mathsf{Q} \cap (0, 1/2)$ (think $\frac{1}{\text{poly}}$). Assume that, for all n, G maps $\{0,1\}^{L(n)}$ into $\{0,1\}^n$. G is $(L(n), s(n), \text{eps}(n))$-*next bit predictable* if, for infinitely many n, there exists $i \in \{2, \ldots, n\}$ and a circuit $C_n : \{0,1\}^{i-1} \rightarrow \{0,1\}$ such that

1. C_n is a deterministic circuit of size $s(n)$.

2. For at least $\frac{1}{2}+\mathrm{eps}(n)$ of strings $y \in \{G_{L(n)}(x)[1:i-1] \mid x \in \{0,1\}^{L(n)}\}$, $C(y) = G_{L(n)}(x)[i]$. (Note that we interpret $\{G_{L(n)}(x)[1:i-1] \mid x \in \{0,1\}^{L(n)}\}$ as a multiset.)

Def 1.5 Let $f : \{0,1\}^* \to \{0,1\}$. Let f_n be the restriction of f to $\{0,1\}^n$. f is $(s(n), \mathrm{eps}(n))$*–hard* if there does not exist an $s(n)$-sized circuit C_n that computes, for almost all n, f_n correctly on $\frac{1}{2}+\mathrm{eps}(n)$ of the strings in $\{0,1\}^n$.

2 Notation Used Throughout the Paper

Notation 2.1 Throughout this paper the following hold.

1. $L(n) : \mathsf{N} \to \mathsf{N}$ (think log). c will be a constant. $cL(n)$ will be used alot.

2. $s(n), S(n) : \mathsf{N} \to \mathsf{N}$ (think poly, $2^{\epsilon n}$). Bounds on circuit size.

3. $r(n) : \mathsf{N} \to \mathsf{N}$ (think poly). We require $r(n) \geq n$. The random string that a BPP machine uses.

4. $t(n), T(n) : \mathsf{N} \to \mathsf{N}$ (think poly, 2^n). Run times.

5. $G : \{0,1\}^* \to \{0,1\}^*$. At different places we will also require that for all n $\{0,1\}^{cL(n)}$ maps to $\{0,1\}^n$, for some c. We denote the subfunction that maps $\{0,1\}^m$ to $\{0,1\}^n$ by G_m. (m will be $L(n)$ or $cL(n)$ or $c^2L(n)$). A potential psuedorandom generator.

6. $f : \{0,1\}^* \to \{0,1\}$. We denote the subfunction that maps $\{0,1\}^n$ to $\{0,1\}$ by f_n. A "hard" function.

7. $\mathrm{err}(n) : \mathsf{N} \to \mathsf{Q} \cap (0,1/2)$ (think $\frac{1}{4}$) An error, so the smaller it is the less chance of error.

8. $\mathrm{diff}(n) : \mathsf{N} \to \mathsf{Q} \cap (0,1/2)$ (think $\frac{1}{\mathrm{poly}}$). $\mathrm{diff}(n)$ is decreasing. How much two distributions differ. The smaller it is, the less they differ.

9. $\mathrm{eps}(n) : \mathsf{N} \to \mathsf{Q} \cap (0,1/2)$ (think $\frac{1}{\mathrm{poly}}$). $\mathrm{eps}(n)$ is decreasing. How much more than $\frac{1}{2}$ of the elements of some domain a function is computed correctly. The larger $\mathrm{eps}(n)$ the large the domain we can compute the function on.