B.4 Hard Problems

Exactly defining what a hard problem is is somewhat more tricky. We want to characterize a problem that has *no efficient solution for almost all* input tuples (or strings). Simply negating the property "efficient" is clearly insufficient. Somewhat better is the requirement that the advantage of an algorithm approaches 0 with increasing n. But also this is not yet a suitable definition since the advantage describes a lower bound only.

A better requirement is the non-existence of an advantage that approaches 0 too slowly. "Too slowly" is

$$\frac{1}{\eta(n)} \quad \text{with an arbitrary polynomial } \eta \in \mathbb{N}[X].$$

"Slow enough" is for instance the inverse exponential finction $1/2^n$.

Moreover there should be "almost no" exceptions, or the set of exceptions should be "sparse". Now we try to translate these ideas into an exact definition.

For $x \in L_{r(n)}$ we consider the probability

$$p_x := P(\{\omega \in \Omega_{k(n)} \mid C_n(x, \omega) = f(x)\}),$$

and the set of input strings x for which C_n has an ε-advantage:

$$L_{r(n)}(\varepsilon) := \{x \in L_{r(n)} \mid p_x \geq \frac{1}{2^{s(n)}} + \varepsilon\}.$$

For a polynomial $\eta \in \mathbb{N}[X]$ the set $L_{r(n)}(\frac{1}{\eta(n)})$ consists of the input strings x for which C computes $f(x)$ with advantage $\frac{1}{\eta(n)}$. Thus the exceptional set for η is

$$L^{[f,C,\eta]} := \bigcup_{n \in \mathbb{N}} L_{r(n)}(\frac{1}{\eta(n)}).$$

We denote it as "**advantageous set for f, C, η**". Its components should become more and more marginal with increasing n. The definition is:

Definition 3 A subset $A \subseteq L$ is called **sparse** if

$$\frac{\#A_n}{\#L_n}$$

is negligible.

Remarks and Examples

1. If $\#A_n = c$ is constant, and $L_n = \mathbb{F}_2^n$, then A is sparse in L for

$$\frac{\#A_n}{\#L_n} = \frac{c}{2^n}.$$

2. If $\#A_n$ grows at most polynomially, but $\#L_n$ grows faster than any polynomial, then A is sparse in L.

3. If $\#A_n = c \cdot \#L_n$ is a fixed proportion, then A is not sparse in L.

4. If $L = \mathbb{N}$, and A is the set of primes (in binary coding), then by the prime number theorem

$$\#A_n \approx \frac{2^{n-1}}{n \cdot \ln(2)} = \frac{\#L_n}{n \cdot \ln(2)}.$$

Hence the set of primes is not sparse in $\mathbb{N}$.

5. No known efficient algorithm is able to factorize a non-sparse subset of the set M of all products of primes whose lengths differ by at most one bit.

Definition 4 Let f be as in (2). Then f is called **hard** if for each PPC as in (1) and for each polynomial $\eta \in \mathbb{N}[X]$ the advantageous set $L^{[f,C,\eta]}$ is a sparse subset of L.

Examples

1. The conjecture that prime decomposition of integers is hard makes sense by remark 5.

2. **Quadratic residuosity conjecture:** Let B be the set of BLUM integers (products of two primes $\equiv 3 \pmod 4$),

$$L = \{(m, a) \mid m \in B, a \in \mathbb{M}_n^+\},$$

(for $\mathbb{M}_n^+$ see Appendix A.5) and let

$$f : L \longrightarrow \mathbb{F}_2$$

be the indicator function

$$f(m, a) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue} \bmod m, \\ 0 & \text{else.} \end{cases}$$

Then f is hard. (A forteriori when we more generally admit $a \in \mathbb{M}_n$.)