4.4 Yao's Criterion

At first sight trying to prove the perfectness of a pseudorandom generator G seems hopeless. How to manage "all polynomial tests"? But surprisingly a seemingly much weaker test is sufficient. Let $G_m(x) = (b_1^{(m)}(x), \ldots, b_{r(n)}^{(m)}(x))$ be the bit sequence generated by G_m from the seed x. Let $C = (C_n)_{n \in \mathbb{N}}$ be a polynomial family of circuits,

$$C_n : \mathbb{F}_2^n \times \mathbb{F}_2^{i_n} \times \Omega_n \longrightarrow \mathbb{F}_2$$

with $0 \le i_n \le r(n) - 1$, and let $h \in \mathbb{N}[X]$ be a non-constant polynomial. Then we say that C has a $\frac{1}{h}$-advantage for extrapolating G if the set of parameters $m \in M$ with

$$P\{(x,\omega) \in A_m \times \Omega_n \mid C_n(m, b_{j_m+1}^{(m)}(x), \ldots, b_{j_m+i_n}^{(m)}(x), \omega) = b_{j_m}^{(m)}(x)\}$$

$$\geq \frac{1}{2} + \frac{1}{h(n)} \tag{2}$$

for an index j_m, $1 \le j_m \le r(n) - i_n$, is not sparse in M. In other words given a subsequence C extrapolates the preceding bit with a small advantage in sufficiently many cases. We say that G passes the **extrapolation test** if there exists no such polynomial family of circuits with a $\frac{1}{h}$-advantage for extrapolating G for any polynomial $h \in \mathbb{N}[X]$.

For instance the linear congruential generator fails the extrapolation test, as does a linear feedback shift register.

Theorem 4 [Yao's criterion] *The following statements are equivalent for a pseudorandom generator G:*

(i) *G is perfect.*

(ii) *G passes the extrapolation test.*

Proof. "(i) $\Longrightarrow$ (ii)": Assume G fails the extrapolation test. Then there is a polynomial family C of circuits that has a $\frac{1}{h}$-advantage for extrapolating G. Let $A \subseteq M$ be the non-sparse set of parameters for which the inequality (2) holds. We construct a polynomial test $C' = (C'_n)_{n \in \mathbb{N}}$:

$$C'_n(m, u, \omega) = C_n(m, u_{j_m+1}, \ldots, u_{j_m+i_n}, \omega) + u_{j_m} + 1$$

where for $m \in \mathbb{F}_2^n - A$ we set $j_m = 1$ (this value doesn't matter anyway). Hence

$$C'_n(m, u, \omega) = 1 \Longleftrightarrow C_n(m, u_{j_m+1}, \ldots, u_{j_m+i_n}, \omega) = u_{j_m}.$$

For $m \in A$ we get

$$p(G, C', m) = P\{C_n(m, b^{(m)}_{j_m+1}(x), \ldots, b^{(m)}_{j_m+i_n}(x), \omega) = b^{(m)}_{j_m}(x)\} \;\geq\; \frac{1}{2} + \frac{1}{h(n)}$$

and have to compare this value with

$$\bar{p}(C', m) = P\{C_n(m, u_{j_m+1}, \ldots, u_{j_m+i_n}, \omega) = u_{j_m}\}$$

$$= P\{C_n(\ldots) = 0 \text{ and } u_{j_m} = 0\} + P\{C_n(\ldots) = 1 \text{ and } u_{j_m} = 1\}.$$

(The sum corresponds to a decomposition into two disjoint subsets.) Each summand denotes the probability that two independent events occur simultaneously. Thus

$$\bar{p}(C', m) = \frac{1}{2} P\{C_n(\ldots) = 0\} + \frac{1}{2} P\{C_n(\ldots) = 1\} = \frac{1}{2}.$$

Hence for $m \in A$

$$p(G, C', m) - \bar{p}(C', m) \geq \frac{1}{h(n)}.$$

We conclude that G fails the test C', and therefore is not perfect.

"(ii) $\Longrightarrow$ (i)": Assume G is not perfect. Then there is a polynomial test C failed by G. Hence there is a non-constant polynomial $h \in \mathbb{N}[X]$ and a $t \in \mathbb{N}$ with

$$|p(G, C, m) - \bar{p}(C, m)| \geq \frac{1}{h(n)}$$

for m from a non-sparse subset $A \subseteq M$ with $\#A_n \geq \#M_n/n^t$ for infinitely many $n \in I$. For at least half of all $m \in A_n$ we have $p(G, C, m) > \bar{p}(C, m)$ or the inverse inequality. First we treat the first of these two cases (for fixed n).

For $k = 0, \ldots, r(n)$ let

$$p^k_m = P\{C_n(m, t_1, \ldots, t_k, b^{(m)}_{k+1}(x), \ldots, b^{(m)}_{r(n)}(x), \omega) = 1\}$$

where $t_1, \ldots, t_k \in \mathbb{F}_2$ are random bits. So we consider the probability in $A_m \times (\mathbb{F}_2^k \times \Omega_n)$. We have

$$p^0_m = p(G, C, m), \quad p^{r(n)}_m = \bar{p}(C, m),$$

$$\frac{1}{h(n)} \leq p^0_m - p^{r(n)}_m = \sum_{k=1}^{r(n)} (p^{k-1}_m - p^k_m)$$

for the $m \in A_n$ under consideration. Thus there is an r_m with $1 \leq r_m \leq r(n)$ such that

$$p^{r_m-1}_m - p^{r_m}_m \geq \frac{1}{r(n)h(n)}.$$

One of these values r_m occurs at least $(\#M_n/2n^t r(n))$ times, denote it by k_n.

Let $\Omega'_n = \mathbb{F}_2^{k_n} \times \Omega_n$. The polynomial family C' of circuits whose deterministic inputs are fed from $A_n \times \mathbb{F}_2^{r(n)-k_n}$, and whose probabilistic inputs from Ω'_n, is defined for this n by

$$C'_n(m, u_1, \ldots, u_{r(n)-k_n}, t_1, \ldots, t_{k_n}, \omega) = C_n(m, t, u, \omega) + t_{k_n} + 1.$$

Hence

$$C'_n(m, u, t, \omega) = t_{k_n} \Longleftrightarrow C_n(m, t, u, \omega) = 1.$$

Now

$$C'_n(m, b^{(m)}_{k_n+1}(x), \ldots, b^{(m)}_{r(n)}(x), t, \omega) = b^{(m)}_{k_n}(x)$$

$$\Longleftrightarrow \begin{cases} C_n(m, t, b^{(m)}_{k_n+1}(x), \ldots, b^{(m)}_{r(n)}(x), \omega) = 1 \quad \text{and} \quad t_{k_n} = b^{(m)}_{k_n}(x) \\ \text{or} \\ C_n(m, t, b^{(m)}_{k_n+1}(x), \ldots, b^{(m)}_{r(n)}(x), \omega) = 0 \quad \text{and} \quad t_{k_n} \neq b^{(m)}_{k_n}(x) \end{cases}$$

Both cases describe the occurence of two independent events. Therefore the probability of the second one is $\frac{1}{2}(1 - p_m^{k_n})$. The first one is equivalent with

$$C_n(m, t_1, \ldots, t_{k_n-1}, b^{(m)}_{k_n}(x), \ldots, b^{(m)}_{r(n)}(x), \omega) = 1 \quad \text{and} \quad t_{k_n} = b^{(m)}_{k_n}(x)\,.$$

Its probability is $p_m^{k_n-1}/2$. Together this gives

$$P\{C'_n(m, b^{(m)}_{k_n+1}(x), \ldots, b^{(m)}_{r(n)}(x), t, \omega) = b^{(m)}_{k_n}(x)\}$$

$$= \frac{1}{2} + \frac{1}{2}(p_m^{k_n-1} - p_m^{k_n}) \geq \frac{1}{2} + \frac{1}{2r(n)h(n)}$$

for at least $\#M_n/2n^t r(n)$ of the parameters $m \in M_n$. With $u = t + \deg(r) + 1$ this is $\geq \#M_n/n^u$ for infinitely many $n \in I$.

In the case where $p(G, C, m) < \bar{p}(C, m)$ for at least half of all $m \in A_n$ we analoguously set

$$C'_n(m, u, t, \omega) = C_n(m, t, u, \omega) + t_{k_n}.$$

Then the derivation runs along the same lines.

Therefore G fails the extrapolation test (with $i_n = r(n) - k_n$ and $j_m = k_n$). $\diamond$

By the way the proof made use of the non-uniformity of the computational model: C'_n depends on k_n, and we didn't give an algorithm that determines k_n.