## 3.6 Linear Complexity and Turing Complexity

A **universal** Turing **machine** is able to simulate every other Turing machine by a suitable program. Let **M** be one, and let $u \in \mathbb{F}_2^n$ be a bit sequence of length $n$. Then the Turing-Kolmogorov-Chaitin **(TKC) complexity** $\chi(u)$ is the length of the shortest program of **M** that outputs $u$. There is always one such program of length $n$: Simply take $u$ as input sequence and output it unchanged. (Informally: Move the input tape forward by $n$ steps and stop.)

**Remark** The function $\chi\colon \mathbb{F}_2^* \longrightarrow \mathbb{N}$ itself is not computable. This means there is no Turing machine that computes $\chi$. Thus the TKC complexity is of low practical value as a measure of complexity. However in the recent years it gained some momentum in a more precise form by the work of Vitanyi and others, see for example:

> Ming Li, Paul Vitanyi: *An Introduction to Kolmogorov Complexity and Its Applications.* Springer, New York 1993, 1997.

A central result of the theory is:

$$\frac{1}{2^n} \cdot \#\{u \in \mathbb{F}_2^n \mid \chi(u) > n \cdot (1-\varepsilon)\} > 1 - \frac{1}{2^{n\varepsilon-1}}.$$

This result says that almost all sequences have a TKC complexity near the maximum value, there is no significantly shorter description of a sequence than to simply write it down. A common interpretation of this result is: "Almost all sequences are random." This corresponds quite well with the intuitive idea of randomness. Nobody would consider a sequence with a short description such as "alternate one million times between 0 and 1" as random.

> Thomas Beth, Zong-Duo Dai: On the complexity of pseudo-random sequences – or: If you can describe a sequence it can't be random. Eurocrypt 89, 533–543.

This paper contains some small errors that are corrected in [9].

Also "linear complexity" $\lambda$ measures complexity, using a quite special machine model: the LFSR. On first sight it suffers from severe deficits. The sequence "999999 times 0, then a single 1" has a very low TKC complexity—corresponding to a very low intuitive randomness—, but the linear complexity is 1 million.

Of course we could also try to use nonlinear FSRs for measuring complexity, see for instance the papers:

- Agnes Hui Chan, Richard A. Games: On the quadratic span of periodic sequences. Crypto 89, 82–89.

- Cees J. A. Jansen, Dick E. Boekee: The shortest feedback shift register that can generate a given sequence. Crypto 89, 90–96.

and Appendix B. However, as we saw, *a short description by a nonlinear FSR also implies a small linear complexity.*

In any case linear complexity has the advantage of easy explicite computability, and "in general" it characterizes the randomness of a bit sequence very well. This vague statement admits a surprisingly precise wording (stated here without proof). To make a fair comparision note that the description of a sequence by an LFSR needs $2 \times \lambda$ bits: the taps of the register and the starting value. Therefore we should compare $\chi$ and $2 \cdot \lambda$:

**Proposition 12** (Beth/Dai)

$$\begin{aligned}
\frac{1}{2^n} \cdot \#\{u \in \mathbb{F}_2^n \mid (1-\varepsilon)2\lambda(u) \leq \chi(u)\} &\geq 1 - \frac{8}{3 \cdot 2^{\frac{n\varepsilon}{2-\varepsilon}}}, \\
\frac{1}{2^n} \cdot \#\{u \in \mathbb{F}_2^n \mid (1-\varepsilon)\chi(u) \leq 2\lambda(u)\} &\geq 1 - \frac{1}{3} \cdot \frac{1}{2^{n\varepsilon-(1-\varepsilon)(1+\log n)+1}} - \frac{1}{3} \cdot \frac{1}{2^n}.
\end{aligned}$$

We interpret this as: "For almost all bit sequences the linear complexity and the TKC complexity coincide with only a negligeable discrepancy (up to the obvious factor 2)."

This result confirms that linear complexity—despite its simplicity—is a useful measure of complexity, and that in general bit sequences of high linear complexity have no short description in other machine models. Thus they are cryptographically useful. Every efficient prediction method—in the sense of cryptanalysis of bitstream ciphers—would provide a short description in the sense of TKC complexity. And conversely: If a sequence has a short description, then we even can generate it by a short LFSR. Thus we may summarize: *In general a bit sequence of high linear complexity is not efficiently predictable.*

Note that these results

- are "asymptotic" in character. For the "bounded" world we live in they only yield qualitative statements—a standard phenomen for results on cryptographic security.

- concern probabilities only. There might be $2^r \ll 2^n$ sequences of small TKC complexity that however have high linear complexity—*relatively* very few, but *absolutely* quite a lot! In Chapter 4 we'll construct such sequences, dependent on secret parameters, and show (up to one of the usual hardness assumptions for mathematical problems) that they don't allow an efficient prediction algorithm, in particular not by a "short" LFSR.