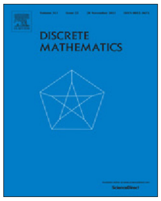
Discrete Mathematics 341 (2018) 2409–2417

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc



Automatic complexity of shift register sequences

Bjørn Kjos-Hanssen

University of Hawai‘i at Mānoa, United States

ARTICLE INFO

Article history:
Received 14 October 2017
Received in revised form 10 April 2018
Accepted 19 May 2018

Keywords:
Linear feedback shift registers
Finite automata
Automatic complexity

ABSTRACT

Let x be an m-sequence, a maximal length sequence produced by a linear feedback shift register. We show that x has maximal subword complexity function in the sense of Allouche and Shallit. We show that this implies that the nondeterministic automatic complexity $A_N(x)$ is close to maximal: $n/2 - A_N(x) = O(\log^2 n)$, where n is the length of x. In contrast, Hyde has shown $A_N(y) \leq n/2 + 1$ for all sequences y of length n.

1. Introduction

Linear feedback shift registers, investigated and popularized by Golomb [2], may be “the most-used mathematical algorithm idea in history”, used at least 10^{27} times in cell phones and other devices [8]. They are particularly known as a simple way of producing pseudorandom output sequences called m-sequences. However, thanks to the Berlekamp–Massey algorithm [5], one can easily find the shortest LFSR that can produce a given sequence x. The length of this LFSR, the *linear complexity* of x, should then be large for a truly pseudorandom sequence, but is small for m-sequences. In this article we show that using a different complexity measure, *automatic complexity*, the pseudorandomness of m-sequences can be measured and, indeed, verified.

Roughly speaking, finite automata are not able to detect significant patterns in shift register sequences. Moreover, shift register sequences seem to give an answer to the question

> “What kind of sequences have high automatic complexity?”

See in particular some results of computer experimentation in Section 4.

2. Definitions

While our computer results in Section 4 concern the linear case specifically, our theoretical results in Section 3 concern the following natural abstraction of the usual notion of feedback shift register [1].

Definition 1. Let q be a positive integer and let $[q] = \{0, \ldots, q-1\}$. A q-ary k-stage combinatorial shift register (CSR) is a mapping

$$\Lambda : [q]^k \to [q]^k$$

E-mail address: bjoernkh@hawaii.edu.
URL: https://math.hawaii.edu/wordpress/bjoern/.

https://doi.org/10.1016/j.disc.2018.05.015
0012-365X/© 2018 Elsevier B.V. All rights reserved.

such that there exists $F : [q]^k \to [q]$ such that for all x_i,

$$\Lambda(x_0, \dots, x_{k-1}) = (x_1, x_2, \dots, x_{k-1}, F(x_0, x_1, \dots, x_{k-1})).$$

The function F is called the *feedback function* of Λ.

Definition 2. An infinite sequence $x = x_0x_1\dots$ is *eventually periodic* if there exist integers M and $N > 0$ such that for all $n > M$, $x_n = x_{n-N}$. The least N for which there exists such an M is the *period* of x.

Definition 3. For any k-stage CSR Λ and any word x of length $\geq k$, the *period of* Λ *upon processing* x is the period of the sequence $\Lambda^t(x_0, \dots, x_{k-1})$, $0 \leq t < \infty$.

Lemma 4 is well-known and easy but we believe including its proof may help the reader.

Lemma 4. *Let k and q be positive integers. Let Λ be a q-ary k-stage CSR. Let $x = x_0x_1\dots$ be an infinite sequence produced by Λ. Then x is eventually periodic, and the period of Λ upon processing x exists and is finite.*

Proof. The infinite sequence $\Lambda^t(x_0, \dots, x_{k-1})$ for $0 \leq t < \infty$ takes values in the finite set $[q]^k$. Thus, by the pigeonhole principle, there exist M and $N > 0$ with

$$\Lambda^M(x_0, \dots, x_{k-1}) = \Lambda^{M-N}(x_0, \dots, x_{k-1}).$$

Let $n > M$. Then

$$\begin{aligned}(x_n, \dots, x_{n+k-1}) &= \Lambda^n(x_0, \dots, x_{k-1})\\ &= \Lambda^{n-M}\Lambda^M(x_0, \dots, x_{k-1})\\ &= \Lambda^{n-M}\Lambda^{M-N}(x_0, \dots, x_{k-1})\\ &= \Lambda^{n-N}(x_0, \dots, x_{k-1})\\ &= (x_{n-N}, \dots, x_{n-N+k-1}),\end{aligned}$$

hence $x_n = x_{n-N}$. □

We can now define LFSRs and m-sequences. As our computer results concern binary sequences, we take $q = 2$. However, a higher level of generality would also be possible.

Definition 5. Suppose a k-stage CSR Λ produces the infinite output $x = x_0x_1\dots$ and its feedback function is a linear transformation of $[q]$ when viewed as the finite field $\mathbb{F}_q$, where $q = 2$. Then Λ is a *linear feedback shift register* (LFSR). Suppose the period P of Λ upon processing x is $2^k - 1$. Then $x_0 \dots x_{P-1}$ is called an m-sequence (or maximal length sequence, or PN (pseudo-noise) sequence).

If m-sequences are pseudo-random in some sense then they should have high, or at least *not unusually low*, complexity according to some measure. In 2015, Jason Castiglione (personal communication) suggested that *automatic complexity* might be that measure.

Our *nondeterministic finite automata* will have no ϵ-transitions, a unique start state and a set of accepting states. Without loss of generality for our purposes, the accepting state is unique. The language recognized by an automaton M is the set $L(M)$ of words accepted by M.

Definition 6 ([3,7]). Let $L(M)$ be the language recognized by the automaton M. Let x be a sequence of finite length n.

- The (deterministic) *automatic complexity* of x is the least number $A(x)$ of states of a deterministic finite automaton M such that

$$L(M) \cap \{0, 1\}^n = \{x\}.$$

- The *nondeterministic automatic complexity* $A_N(x)$ is the minimum number of states of a nondeterministic finite automaton (NFA) M accepting x such that there is only one accepting path in M of length $|x|$.
- The *non-total deterministic automatic complexity* $A^-(x)$ is defined like $A(x)$ but without requiring totality of the transition function.

As totality can always be achieved by adding at most one extra "dead" state, we have

$$A_N(x) \leq A^-(x) \leq A(x) \leq A^-(x) + 1.$$

Theorem 7 (Hyde [3]). *The nondeterministic automatic complexity $A_N(x)$ of a sequence x of length n satisfies*

$$A_N(x) \leq \lfloor n/2 \rfloor + 1.$$

Fig. 1 gives a hint to the proof of Theorem 7 in the case where n is odd. Theorem 7 is sharp [3], and experimentally we find that about 50% of all binary sequences attain the bound. Thus, to "fool" finite automata this bound should be attained or almost attained.