Automatic Sequences - a glimpse

CSc 482 Concrete Mathematics course project

Carlos Torchia - V00717432

July 10, 2012

Abstract

A sequence $(a_n)_{n\geqslant 0}$ that can be generated using the output of a deterministic finite automaton with output (DFAO) with the base-k representation of n input is called an automatic sequence. Automatic sequences are discussed in the book *Automatic Sequences* by Allouche and Shallit. There are alternative characterizations of automatic sequences, which require instead there to exist a DFAO that operates on other numerical representations of n, such as base 1 or base $-k$. There are also many examples of k-automatic sequences. The Thue-Morse sequence and its generalization base-k modulo m are automatic sequences, as they have simple DFAOs that generate them. Also paperfolding sequences based on iteratively folding a sheet of paper half way according to a set of rules or directions is automatic if and only if the folding rules are ultimately periodic. A k-automatic set is one whose characteristic sequence is k-automatic. A set is k-automatic if and only if the set of base-k representations of the numbers therein is a regular language. The set of perfect squares and the set of prime numbers are shown to be not k-automatic. These propositions are well known, and the purpose of this paper is to be an introduction.

Introduction

A k-automatic sequence for $k \geqslant 2$ is a sequence of integers $(a_n)_{n\geqslant 0}$ such that a_n can be computed by a deterministic finite automaton with output (DFAO) as the output of the last state to process the base-k representation of n. More precisely, $(a_n)_{n\geqslant 0}$ is k-automatic if there exists a DFAO $(Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ such that for all $n \geqslant 0$, $\tau(\delta(q_0, w)) = a_n$, where $n = [w]_k$ so that w is some base-k representation of a_n [AS]. Since the DFAO is operating on base-k representations of integers, we say it is a k-DFAO.

In this paper, we will briefly note some alternative characterizations of k-automatic sequences, and then look at some examples of automatic sequences and non-automatic sets.

We will mostly be looking at definitions and proofs from *Automatic Sequences* [AS], but I provide some extra clarification or interpretation, as well as provide my own proof for the generalization of the Thue Morse sequence. Thus, the main objective of this paper is not to provide new results but rather to explain this topic.

The intended audience of this paper is the professor of CSc 482 (Concrete Mathematics), Dr. Frank Ruskey, but I assume any student who has taken a course on the theory of computation could pick this up and understand it with some effort, and hopefully be of some help to them. In other words, it is hoped that the explanations in this paper are not overly-convoluted or confusing.

Some notation

Most of this paper should be written in a way understandable to anyone with some mathematical background. If $M = (Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ is a k-DFAO, this implies that Q is its set of states, $\Sigma_k = \{0, 1, \ldots, k-1\}$ is its input alphabet, $\delta\colon Q \times \Sigma_k \to Q$ is the state transition function, q_0 is the initial state, Δ is the output alphabet, and $\tau\colon Q \to \Delta$ is the output function. When we write $\delta(q, w) = q'$, we mean that if we are at state q and we receive the word w, we will thereafter be at state q'.

As noted before $[w]_k = n$ implies that w is some base-k representation of n. Also $(n)_k = w$ implies that w is the unique base-k representation of n without any leading 0s.

Alternative characterizations of automatic sequences

Now we discuss alternative characterizations of automatic sequences. These definitions also use the idea of a sequence being generated by an automaton's output, but the representation of the input n to the DFAO is different from the original definition in each case.

1. A sequence is k-automatic if it has a DFAO that computes a_n given the base-k representation w of $n = [w]_k$ only if $[w]_k$ is without leading 0s, because one can easily augment the DFAO to handle leading 0s (Theorem 5.2.1 [AS]).

2. Any sequence that has a DFAO that reads w starting at the least significant bit is k-automatic, because a DFAO can be constructed that reads w^R, i.e. the representation of n reversed, and produces the same output (Theorem 5.3.4 [AS]).

3. A sequence $(a_n)_{n\in\mathbb{Z}}$ is defined to be $(-k)$-automatic if there exists a DFAO that generates a_n using n's base-$(-k)$ representation. Equivalently, $(a_n)_{n\in\mathbb{Z}}$ is k-automatic if and only if $(a_n)_{n\geqslant 0}$ is k-automatic and $(a_{-n})_{n\geqslant 0}$ is k-automatic (Theorem 5.3.2 [AS]).

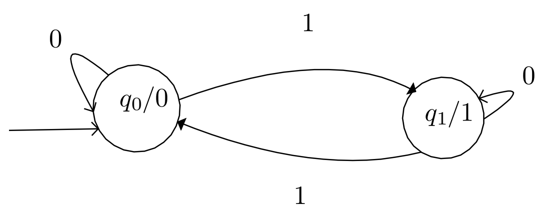
4. A sequence $(a_n)_{n\geqslant 0}$ is defined to be 1-automatic if there exists a DFAO that generates a_n using n's base-1 representation. Here, n is represented as 1^n. $(a_n)_{n\geqslant 0}$ is 1-automatic if and only if it is ultimately periodic (last theorem in chapter 5 of [AS]).

5. There are other characterizations of k-automatic sequences. For example, if a DFAO exists that compute $(a_n)_{n\in\mathbb{Z}}$ using n's representation in the (k, l)-number system, then $(a_n)_{n\in\mathbb{Z}}$ is said to be (k, l)-automatic, and this is equivalent to $(a_n)_{n\geqslant 0}$ and $(a_{-n})_{n\geqslant 0}$ being $k + l + 1$-automatic [AS].

Thus, although the input n to the DFAO is in a different numerical representation for sequences satisfying each of these definitions, they are essentially equivalent to automatic sequences.

The Thue-Morse sequence

We now proceed to discussing some examples of k-automatic sequences. Firstly, the Thue-Morse sequence, also discussed by Sardar Ali in his presentation, is defined to be t_n, which is 1 if the number of 1s in the base 2 representation of n is odd, 0 if it is even. In other words, $t_n = s_2(n) \bmod 2$, where $s_k(\cdot)$ is the base-k sum of digits function [AS]. This sequence is 2-automatic because the following DFAO generates it upon receiving n in base-2 [AS]:



Here, when we are at q_0 an even number of 1s have been observed. At q_1, an odd number of 1s have been observed. When the automaton comes to the end of the input, if we are at q_1 then w has an odd number of 1s so we output 1, and if we are at q_0 then w has an even number of 1s so we output 0. Thus, this automaton generates t_n.

Base-k generalization of the Thue-Morse sequence

In general, as suggested in Exercise 5.2 (a) [AS], we can make a k-DFAO that generates $s_k(n) \bmod m$, the sum of the digits of n in base-k modulo m, for any $k \geqslant 2$ and $m \geqslant 1$.

Proof. Let $A = (Q, \Sigma_k, \delta, q_0, \Sigma_m, \tau)$ as follows:

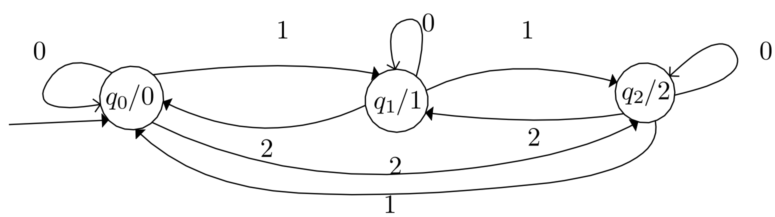
$$Q = \{q_0, q_1, \ldots, q_{m-1}\}$$

$$\tau(q_i) = i \text{ for } i \in 0 \ldots m-1$$

$$\delta(q_i, j) = q_{(i+j) \bmod m} \text{ for } j \in 0 \ldots k-1$$

Here, being in state q_i means that so far the sum of digits modulo m is i. Suppose $[w]_k = n$ and $w = w_0\ w_1\ \ldots w_{n-1}$. If $|w| = 1$, then $[w]_k = w_0$ so $\delta(q_0,\ w) = q_{w_0 \bmod m} = q_{s_k([w]_k) \bmod m}$, so $\tau(\delta(q_0,\ w)) = s_k([w]_k) \bmod m$. If $|w| > 1$, then by induction $\delta(q_0,\ w_0 w_1 \ldots w_{n-2}) = q_{s_k([w_0 w_1 \ldots w_{n-2}]_k) \bmod m}$, so $\delta(q_0,\ w) = q_{(s_k([w_0 w_1 \ldots w_{n-2}]_k) \bmod m + w_{n-1}) \bmod m} = q_{s_k([w]_k) \bmod m}$. Thus $\tau(\delta(q_0, w)) = s_k([w]_k) \bmod m$. □

As an example, suppose $k = 3, m = 3$. The following DFAO generates $s_3(n) \bmod 3$:



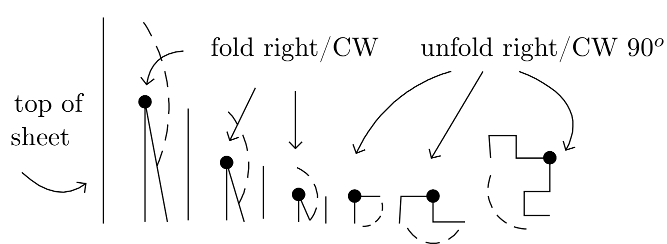
Here, at state q_0 the sum of digits is 0, at state q_1 the sum of digits is 1, and at state q_2 the sum of digits is 2.

Paperfolding sequences

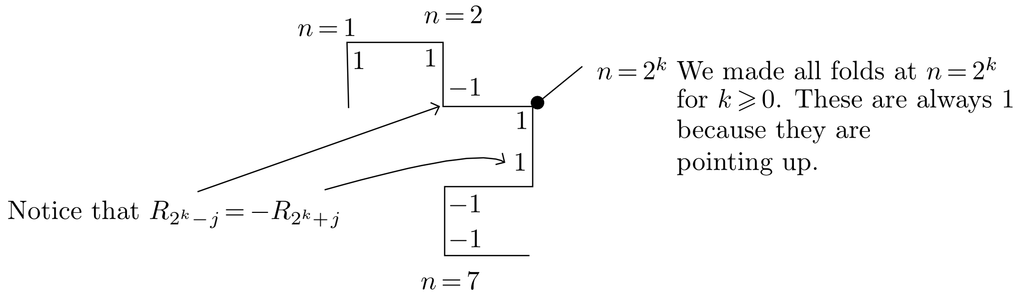
In this section, we discuss the regular paperfolding sequence, which is 2-automatic. Then we will ask when any general paperfolding sequence is k-automatic.

The regular paperfolding sequence

The definition of the *regular paperfolding sequence* is based on the folding of a sheet of paper in half n times always in the same direction (right or clockwise), and then unraveling it so that each corner is 90^o.



Those corners that point upwards, i.e. that were on the top side of the unraveled sheet, are assigned the value 1, and their order from left to right on the sheet is assigned $n \geqslant 1$. This sequence R_n is defined to be the *regular paperfolding sequence* [AS], and its first 7 values are $<1, 1, -1, 1, 1, -1, -1>$. Here is the pattern for 3 folds:



Since every time we fold we generate twice as many elements in the sequence, we may conjecture that this sequence is 2-automatic, because we can determine whether given corner is up or down by its position relative to the nearest power of 2.

Indeed, R_n is 2-automatic. The following proof is adapted from Example 5.1.6, Figure 5.7, Observation 6.5.1, and Theorem 6.5.2 [AS] to be specific to the regular folding sequence, and not any folding sequence in general. It shows that if $n=2^k\,(2\,j+1)$ then $R_n=(-1)^j$, and this can be computed by a DFAO given $(n)_k$. Also I provide some detail to intermediate steps that are not entirely obvious, including why the DFAO in Figure 5.7 generates $(-1)^j$ if $n=2^k\,(2\,j+1)$.

Proof. First we show that if $n=2^k\,(2\,j+1)$, then $R_n=(-1)^j$, by induction on n.

If $n=1$, then $k=0$ and $j=0$, and $R_n=R_1=1=(-1)^0=(-1)^j$.

Suppose by induction that for all $n<2^m$, $R_n=(-1)^j$. Suppose then that $2^m\leqslant n<2^{m+1}$. If $n=2^m$, then $R_n=1=(-1)^0=(-1)^j$, so suppose $n>2^m$.

$$\begin{aligned}R_n=&R_{2^{m+1}-(2^{m+1}-n)}\\ =&R_{2^m+2^m-(2^{m+1}-n)}\\ =&R_{2^m+(n-2^m)}\\ =&-R_{2^m-(n-2^m)}\\ =&-R_{2^{m+1}-n}\end{aligned}$$

Note $k<m$, because

If $2^k\,(2\,j+1)=n<2^{m+1}$ and $j\geqslant 1$, then $2^k<2^k\,j<2^k\,\frac{2\,j+1}{2}<2^m$, then $2^k<2^m$, and thus $k<m$.

Also note $2^{m+1}-n<n$, because

If $2^m<n$, then $2^{m+1}-n<2^{m+1}-2^m=2^m<n$.

Let $j'=2^{m-k}-j-1$. Then

$$\begin{aligned}2^{m+1}-n=&2^{m+1}-2^k\,(2\,j+1)\\ =&2^k\,(2^{m-k+1}-2\,j-1)\\ =&2^k\,(2\,(2^{m-k}-j)-1)\\ =&2^k\,(2\,(2^{m-k}-j-1)+1)\\ =&2^k\,(2\,j'+1)\end{aligned}$$

So since $2^{m+1}-n<n$, $R_{2^{m+1}-n}=(-1)^{j'}$ by induction.

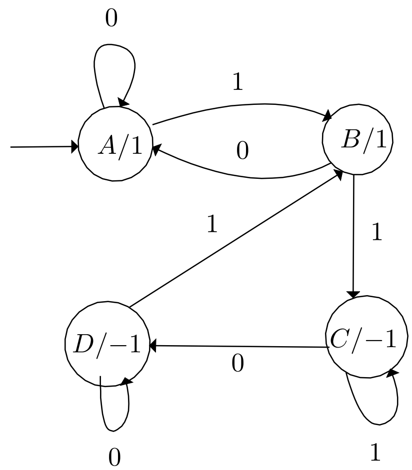
And since $k<m$, $j'=2^{m-k}-j-1\equiv j+1\,(\mathrm{mod}\,2)$.

Thus, $R_n = -R_{2^{m+1}-n} = -(-1)^{j'} = (-1)^{j'+1} = (-1)^j$.

Now, if j is odd, so that $R_n = -1$, $(n)_2 = w_b w_{b-1} \ldots w_{k+2} 110^k$. (Since $2j+1$ ends with 11.)

Otherwise, if j is even so that $R_n = 1$, either $j = 0$ or $j > 0$. If $j = 0$, then $n = 2^k$, so $(n)_2 = 10^k$. If $j > 0$, then $(n)_2 = w_b w_{b-1} \ldots w_{k+2} 010^k$. (Since $2j+1$ ends with 01.)

Thus, the following DFAO generates R_n:

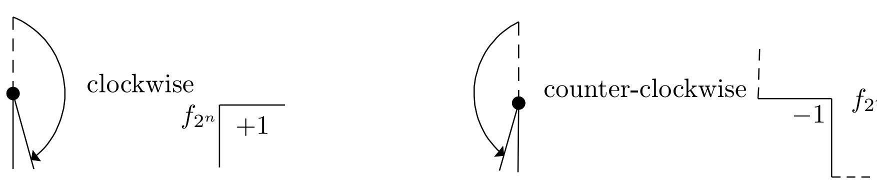


Here, being in state A means that no digits have been processed or the last digit is 0, and $(n)_2 = 10^k$ or $(n)_2 = \ldots 010^k$ so far, so $(-1)^j = 1$. (Here $k > 0$.) Being in state B means that the last digit is 1 and $(n)_2 = \ldots 01$ so far, so $(-1)^j = 1$. Being in state C means that the last digit is 1 and $(n)_2 = \ldots 11$ so far, so $(-1)^j = -1$. Being in state D means that the last digit is 0 and $(n)_2 = \ldots 110^k$ so far, so $(-1)^j = -1$.

Thus, $(R_n)_{n \geqslant 0}$ is 2-automatic. $\square$

Paperfolding sequences in general: when are they k-automatic?

The regular paperfolding sequence is not the only paperfolding sequence. The regular paperfolding sequence is generated by folding the paper clockwise on each iteration. Different sequences can be generated if we are allowed to fold the paper in different directions each time we fold. For example, we can alternate between folding the paper in half clockwise and counter-clockwise.



A sequence $(a_n)_{n \geqslant 0}$ is *ultimately periodic* if eventually (or immediately) the sequence starts repeating a finite sequence of elements infinitely in succession. More precisely, for some $r, s \geqslant 0$, $a_{n+sj} = a_n$ for all $n \geqslant r$ and $j \geqslant 0$. For example, $\langle 1, 1, -1, 1, -1, 1, -1 \ldots \rangle$ is ultimately periodic.

According to Theorem 6.5.4 [AS], a paperfolding sequence $(f_n)_{n \geqslant 0}$ is 2-automatic if and only if the sequence of foldings $(b_n)_{n \geqslant 0}$ is ultimately periodic. I sketch the proof by [AS] here.

Note $b_n = f_{2^n}$. This is because each fold performed doubles the number of corners, so the corner number 2^n corresponds to the direction b_n in which the fold was made. A number of the form 2^n is represented as a binary string 10^n, so if $(f_n)_{n \geqslant 0}$ is generated by an a k-DFAO, eventually it will start reusing the same states after there have been enough 0s, so b_n must be ultimately periodic (Corollary 5.5.3 [AS]). Conversely, if $(b_n)_{n \geqslant 0}$ is ultimately periodic, then it is 2-automatic, so an automaton can be constructed that generates $(-1)^j b_k$ where $n = 2^k (2j+1)$, which is the value of f_n in general. $\square$

Regular expressions and k-automatic sets

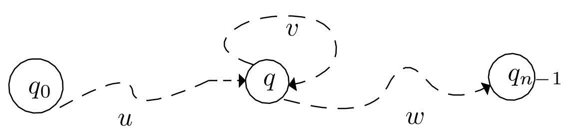
A characteristic sequence $(a_n)_{n\geqslant 0}$ of a set S is $a_n = [\![n \in S]\!]$, i.e. $a_n = 1$ if $n \in S$, $a_n = 0$ if $n \notin S$. A k-automatic set is one whose characteristic sequence is k-automatic.

It can be shown that a sequence $(a_n)_{n\geqslant 0}$ is k-automatic if and only if $I_k(\boldsymbol{a}, d) = \{(n)_k : a_n = d\}$ is a regular language for all $d \in \Delta$, where $(Q, \Sigma, \delta, q_0, \Delta, \tau)$ is the DFAO that generates a_n (Lemma 5.2.6 [AS]). Thus, if S is k-automatic, then $I_k(\boldsymbol{a}, 1) = \{(n)_k : n \in S\}$ must be regular. Thus, it is natural to think of a k-automatic set S as corresponding to a regular language whose elements are base-k representations of S.

In this section, we will examine two non-automatic sets and their characteristic sequences.

Pumping lemma

First, we recall the pumping lemma (Lemma 4.2.1 [AS]). If a language L is regular, and so it is accepted by some deterministic finite automaton (DFA) with n states, then for any word $w \in L$ whose length $|w| \geqslant n$, there exist $u, v, w \in \Sigma^*$ such that $w = u\,v\,w$, $|u\,v| \leqslant n$, $|v| \geq 1$, and $u\,v^i\,w \in L$ for all $i \geqslant 0$. This is because in outputting w, there must be some state q used more than once, since $|w| \geqslant n$. The string v is the part of w outputted between the first and second occurrences of q. This string v can be repeated by visiting the states between the first and second occurrences of q, and we do not have to visit those states at all so v can also be omitted. Thus, $u\,w$, $u\,v\,w$, $u\,v\,v\,w$, are all members of L.



As Adamczewski shows, the pumping lemma can be extended to k-automatic sets. If S is k-automatic, then the set of base-k representations of its elements is a regular language. Thus, if $p \in S$ and $|(p)_k| > n$ where n is the number of states of the DFAO that accepts S, then $(p)_k$ can be written as $w_1 w_2 w_3$ such that $|w_2| \geqslant 1$, $|w_1 w_2| \leqslant n$, and $[w_1 {w_2}^i w_3]_k \in S$ for all $i \geqslant 0$.

Is the set of perfect squares k-automatic?

Below I give a sketch of the proof that the set of perfect squares is not 2-automatic, in order to provide an understanding of the proof given in [AS]. The idea is that the set of binary representations of perfect squares is not regular, so the characteristic sequence $(a_n)_{n\geqslant 0}$ cannot be 2-automatic.

Let SQUARES $= \{(n)_k : n$ is a perfect square$\}$. If $y^2 = (2^{2m} - 1)\,2^{2n+2} + 1$, then $m = n$ and $y = 2^{2m+1} - 1$. This means that a string of the form $1^{2m}0^{2n+1}1$ that is the binary representation of some square must be of the form $1^{2m}0^{2m+1}1$. Thus, SQUARES $\cap\,(11)^*(00)^*01 = \{1^{2m}0^{2m+1}1 : m \geqslant 0\}$ cannot be regular, because an automaton that accepts $1^{2m}0^{2m+1}1$ must for some m have fewer states than m, so at least two states used in processing $1^{2m}0^{2m+1}1$ must be used twice, and removing the part of the string accepted between these two states must produce another string that should be accepted by the automaton but with fewer than $2\,m$ 1s and still $2\,m + 1$ 0s, so it cannot be of the form $1^{2m}0^{2m+1}1$ (this is the pumping lemma, Lemma 4.2.1 [AS]). Thus, SQUARES itself cannot be regular, but SQUARES $= \{(n)_k : a_n = 1\}$, which by Lemma 5.2.6 must be regular [AS]. $\square$

The question remains whether the set of perfect squares are k-automatic for some $k \neq 2$.

Is the set of prime numbers k-automatic?

Now we will figure out why Exercise 5.12 is true: "the characteristic sequence $(a_n)_{n\geqslant 0}$ of the prime numbers is not a k-automatic sequence for any $k \geqslant 2$" [AS]. After unsuccessfully attempting to find a non-periodic subsequence of $(a_n)_{n\geqslant 0}$, we consult a presentation by Boris Adamczewski that contains the proof to this fact.

First, Corollary 5.5.3 [AS] is not of use here, because it can only be used to show a sequence is not k-automatic because either of the subsequences $(a_{k^n})_{n\geqslant 0}$ or $(a_{k^n-1})_{n\geqslant 0}$ is not ultimately periodic, which they must be if the sequence is to be k-automatic. However, for the prime number sequence these subsequences are ultimately periodic. First, k^n for any k is not prime for $n \geqslant 2$, thus $a_{k^n} = 0$ for $n \geqslant 2$, so $(a_{k^n})_{n\geqslant 0}$ is ultimately periodic. Secondly, if $k > 2$ and $n \geqslant 2$, then $k^n - 1$ is not prime [Wikipedia]. Then for $k > 2$, $a_{k^n-1} = 0$ so $(a_{k^n-1})_{n\geqslant 0}$ is ultimately periodic.

Adamczewski's proof, refered to as Schutzenberger's theorem, uses the pumping lemma and Fermat's little theorem.

Proof. By the pumping lemma, if $p = [w_1 w_2 w_3]_k$ is a sufficiently large prime with $|w_2| \geqslant 1$ and $|(p)_k|$ is greater than the number of states in hypothetical DFAO that accepts S, then $[w_1 w_2{}^p w_3] \in S$ for all $p \geqslant 1$. Note $[w_1 w_2{}^p w_3]_k \equiv [w_1 w_2 w_3]_k \equiv 0 \pmod{p}$ by Fermat's Little Theorem [Adamczewski]. However, for $p > 1$, $[w_1 w_2{}^p w_3]_k$ cannot be prime, so S must contain numbers that are not primes. Thus, the set of prime numbers is not a k-automatic set [Adamczewski]. □

Conclusion

Here in this paper we have discussed the definition of an automatic sequence, including some alternative characterizations of automatic sequences. We looked at the Thue-Morse sequence and its generalization, which are always k-automatic. We looked at paperfolding sequences, which would be k-automatic if and only if its sequence of folds is ultimately periodic. And we looked at the characteristic sequences of perfect square and prime numbers, which are not k-automatic.

There are many other topics of automatic sequences, such as Cobham's theorem, and many other simple examples of automatic sequences such as the Rudin-Shapiro sequence, that are suitable in a paper like this, and the reader is invited to investigate or expand upon this. Also I still do not know if there exists any k-DFAO that accepts the set of base-k representations of all perfect squares.

Acknowledgements

Thanks go to the writers of the textbook *Automatic Sequences* [AS], Boris Adamczewski for the presentation of automatic sequences in relation to number theory, my professor Dr. Frank Ruskey for the opportunity to work on this and lending the book, and my classmate Rayhan for a brief discussion of automatic sequences.

References

[Adamczeski] Adamczeski, B. (2010). *Automata in Number Theory*. "The pumping lemma."

Accessed July 10, 2012:

`http://www.lorentzcenter.nl/lc/web/2010/384/presentations/Adamczewski.pdf`

[AS] Allouche, J.P. and Shallit, J. (2003). *Automatic Sequences*. Chapters 5 and 6.

Cambridge: Cambridge University Press.

[Wikipedia] Wikipedia. *Mersenne prime*. "Theorems about Mersenne numbers." Accessed July 10, 2012:

`http://en.wikipedia.org/wiki/Mersenne_prime`