Information in Infinite Words

Yuri Pritykin*

July 22, 2007

Abstract

In this paper we study "information" in infinite words. This very general and informal notion is formalized in several very different ways. First, we study algorithms on infinite words, mainly in connection with almost periodicity. Namely, using the technics of combinatorics on words, we show that several problems about infinite words are not decidable algorithmically (or these words do not contain enough information to solve these problems). Then we obtain some corollaries about monadic theories on natural numbers, showing some characteristics of words to be inexpressible in monadic language. Finally we use the notion of Kolmogorov complexity of finite objects to somehow describe information in infinite words. A lot of open questions are raised and discussed.

1 Introduction

Very different approaches of this paper can be united under the aim to study how much information and information of what kind is contained in an infinite word. The paper is mainly organized as a discussion of notions and open questions, but some interesting results are also presented. Proofs are usually omitted due to (extended) abstract style.

Lots of interesting algorithmic questions naturally appear in connection with almost periodicity, i. e., whether one can check some property or find some characteristic algorithmically being given a word. Further, we mainly deal with the case when the answers on these questions are negative. In Section 3 we prove that some properties are not effective in algorithmic sense. In other words, we show that under some natural conditions we do not have enough information about a word to find some interesting property or characteristic.

In Section 4 we obtain some corollaries from the results of Section 3 showing the limited power of monadic language to express things about words.

In Section 5 we try to use the widely known and fundamental notion of Kolmogorov complexity of finite words to study the notion of information in infinite word.

2 Preliminaries

We use basic notions and notations of combinatorics on words without definitions. The introduction to the area can be found in [2] or [10]. Nevertheless remind one of the notions which is of the main importance for us. We say that a sequence of words u_n (finite of infinite) tends to an infinite word x and write $\lim_{n\to\infty} u_n = x$, if $\forall i\ \exists n\ \forall m > n\ u_m(i) = x(i)$.

*Moscow State University, Russia, `http://lpcs.math.msu.su/~pritykin/`. The work was partially supported by RFBR grants 06-01-00122, 05-01-02803, NSh-5351.2006.1, Kolmogorov grant of Institute of New Technologies, and the Möbius Contest Foundation for Young Scientists.

An infinite word x is periodic if for some T we have $x(i) = x(i+T)$ for each $i \in \mathbb{N}$. This T is called a period of x. The class of all infinite periodic words we denote by $\mathcal{P}$. Let us consider some extensions of this class.

A word x is called *almost periodic (=uniformly recurrent)* if for any factor u of x there exists a number l such that any factor of x of length l contains at least one occurrence of u (and therefore u occurs in x infinitely many times). Obviously, to show almost periodicity of a word it is sufficient to check the mentioned condition not for all factors, but only for some infinite number of prefixes. Denote by $\mathcal{AP}$ the class of all almost periodic words. This class was introduced by Morse in [11, 12] and earlier works and is widely known in combinatorics on words (e. g., see [3]).

A word x is *eventually almost periodic* if some its suffix is almost periodic. The class of all eventually almost periodic words we denote by $\mathcal{EAP}$. This class was studied in [16].

A word x is called *generalized almost periodic* if for any factor u of x occurring in it infinitely many times there exists a number l such that any factor of x of length l contains at least one occurrence of u. We denote the class of all generalized almost periodic words by $\mathcal{GAP}$. This class was introduced by Semenov in [21] and also was studied in [13].

Suppose $x \in \mathcal{EAP}$. Denote by $\mathrm{pr}(x)$ the minimal n such that $x[n,\infty) \in \mathcal{AP}$. Thus for each $m \geqslant \mathrm{pr}(x)$ we have $x[m,\infty) \in \mathcal{AP}$.

A function $\mathrm{R}_x \colon \mathbb{N} \to \mathbb{N}$ is an *almost periodicity regulator* of a word $x \in \mathcal{GAP}$, if

(1) every string of length n occurring in x infinitely many times, occurs in any factor of length $\mathrm{R}_x(n)$ in x, and

(2) any string of length n occurring finitely many times in x, does not occur in $x[\mathrm{R}_x(n),\infty)$.

The latter condition is important only for words in $\mathcal{GAP} \setminus \mathcal{AP}$. Notice that regulator is not unique: any function greater than regulator is also a regulator.

Obviously, $\mathcal{P} \subset \mathcal{AP} \subset \mathcal{EAP} \subset \mathcal{GAP}$. In fact, all these inclusions are strict. For instance the famous Thue–Morse word $x_T = 0110100110010110\ldots$ (see [24, 1] or Section 3) is an example of an element in $\mathcal{AP}$ but not in $\mathcal{P}$ (moreover, $\mathcal{AP}$ has cardinality continuum while $\mathcal{P}$ is countable, see [13] for proofs). The inequality $\mathcal{AP} \subsetneq \mathcal{EAP}$ is obvious. The inequality $\mathcal{EAP} \subsetneq \mathcal{GAP}$ was proved in [16] (moreover it was proved that $\mathcal{GAP} \setminus \mathcal{EAP}$ has cardinality continuum).

We denote the set of all factors of a sequence x by $\mathrm{Fac}(x)$, the set of all factors of length n by $\mathrm{Fac}_n(x)$. Recall that subword complexity of a sequence x is a function $p_x \colon \mathbb{N} \to \mathbb{N}$ such that $p_x(n) = |\,\mathrm{Fac}_n(x)|$ (see [6] for a survey).

3 Algorithms on Infinite Words

Formally, we consider an algorithm with an oracle for a word on input. This algorithm halts on every oracle and outputs a finite binary string or any other constructive object (for example, "yes" or "no"). The main property of such an algorithm is continuity: it outputs the answer on having read only finite number of symbols from the word. Thus to prove non-effectiveness we only need to show discontinuity. In fact, such proofs are just concrete complicated combinatorial constructions showing this discontinuity.

It is obligatory to remark that there is also another way to formalize the notion "algorithm on infinite word". Namely, one can consider classical algorithms without any oracle defined only on totally computable words. Such an algorithm takes on input a program printing an infinite word and outputs a finite string. However this approach leads to the same situation as in the case of relativized algorithms: such algorithm also generates a continuous function on infinite words (e. g., see Theorem by Kreisel, Lacombe, Shoenfild in [18]).

If we have only a word, then we can not recognize almost any property about this word. For example it is even impossible to understand whether the symbol 1 occurs in given binary word: if an algorithm checks some finite number of symbols and all these symbols are 0, then it can not guarantee that 1 does not occur further. The question about algorithmic decidability becomes more interesting if we allow to give on input some additional information. In the case of generalized almost periodic words almost periodicity regulator might be a good choice.

From this point of view the above problem can be solved effectively: reading first $f(1)$ symbols of the word we can say whether or not 1 occurs in it, and moreover reading next $f(1)$ symbols we can even say whether 1 occurs in it finitely or infinitely many times.

The following several theorems are examples of problems concerning almost periodicity that can not be solved algorithmically.

We say $f_n \to f$ for $f_n, f \colon \mathbb{N} \to \mathbb{N}$ if $\forall i\ \exists n\ \forall m > n\ f_m(i) = f(i)$.

Theorem 1. *Given $x \in \mathcal{EAP}$ and its regulator f, it is impossible to compute algorithmically any $l \geqslant \mathrm{pr}(x)$.*

To show the technics, let us prove this theorem.

Remind that x_T is the Thue–Morse word. This word can be obtained as follows: let $a_0 = 0$, $a_{n+1} = a_n\overline{a}_n$ for all n, and $x_T = \lim_{n\to\infty} a_n$. Notice that $|a_n| = 2^n$. The Thue–Morse word has lots of interesting properties (see [1]), but we are interested in the following one: x_T is cube-free, i. e., for any $a \in \mathbb{B}^*$, $a \neq \Lambda$ the string aaa does not occur in x_T (see [1, 24]).

Proof of Theorem 1. It is enough to construct $x_n \in \mathcal{EAP}$, $x \in \mathcal{AP}$ with regulators f_n, f such that $x_n \to x$, $f_n \to f$, but $\mathrm{pr}(x_n) \to \infty$. Indeed, suppose the mentioned algorithm exists and it outputs some $l \geqslant 0$ (arbitrary for $x \in \mathcal{AP}$) given $\langle x, f\rangle$ on the input. During the computation of l the algorithm reads only finite number of symbols in x and of values of f. Hence there exists $N > l$ such that algorithm does not know any $x(k)$ or $f(k)$ for $k > N$. Since $\mathrm{pr}(x_n) \to \infty$, there exists n such that $\mathrm{pr}(x_n) > N$. The algorithm works on the input $\langle x_n, f_n\rangle$ in the same way as it works on the input $\langle x, f\rangle$, and then outputs l, but $\mathrm{pr}(x_n) > N > l$.

Let $x = x_T$, $x_n = a_na_na_nx$. Notice that $\mathrm{pr}(x_n) \geqslant 2^n$. Indeed, if $\mathrm{pr}(x_n) < 2^n$, then $a_na_nx = a_na_na_n\overline{a}_n\overline{a}_na_n\ldots \in \mathcal{AP}$, and hence $a_na_na_n$ occurs in x_T — contradiction with the statement before the proof.

It only remains to show that we can find regulators f_n, f for x_n, x such that $f_n \to f$. It is sufficient to find the same regulator g for all x_n (then we can increase it and obtain the same regulator for all x_n and for x too). Fix some R_x and assume $g = 4 \cdot \mathrm{R}_x$. Let v with $|v| = k$ occur in $x_n = a_na_na_nx$ infinitely many times. Let us take the factor $x[i, j]$ of length $4 \cdot \mathrm{R}_x(k)$ and show that v occurs in it. If $j \geqslant 3 \cdot 2^n + \mathrm{R}_x(k)$, then v occurs on the factor $x[3 \cdot 2^n, 3 \cdot 2^n + \mathrm{R}_x(k)]$ (by definition of R_x). Otherwise $j < 3 \cdot 2^n + \mathrm{R}_x(k)$, hence $i \leqslant 3 \cdot 2^n - 3\,\mathrm{R}_x(k)$. But $i \geqslant 0$, therefore $\mathrm{R}_x(k) \leqslant 2^n = |a_n|$. Then $x_n[i, i + \mathrm{R}_x(k)]$ is contained in a_na_n. But a_na_n occurs in x, so $x_n[i, i + \mathrm{R}_x(k)]$ occurs too. Therefore v occurs in x.

However g is not required yet. We should look at the strings occurring in x_n finitely many times. Obviously, if some v occurs in x_n finitely many times, then $|v| = k > 2^n$ (otherwise v occurs in two consecutive strings a_n or $\overline{a}_n$, and thus in x). Therefore this can happen only for finite number of different n. Considering all the situations when strings of length k occur in some x_n finitely many times, we probably increase the value $g(k)$, but only finitely many times. Thus the required estimation for regulators exists. □

We have already mentioned that $\mathcal{EAP} \subsetneq \mathcal{GAP}$. It turns out that it is even impossible to separate these classes effectively.

Theorem 2. *Given $x \in \mathcal{GAP}$ and its regulator f, it is impossible to determine algorithmically whether $x \in \mathcal{EAP}$.*

The following theorem shows that it is even impossible to separate effectively $\mathcal{AP}$ and $\mathcal{P}$.

Theorem 3. *Given $x \in \mathcal{AP}$ and its regulator f, it is impossible to determine algorithmically whether $x \in \mathcal{P}$.*

By the argument of Theorem 3 we obtain that there exists an infinite set of periodic words with common regulator. The example is interesting since the periods of these sequences are arbitrary large. This construction can be used in the following result: after adding one symbol at the beginning of an almost periodic word we can not check whether it is still almost periodic. Or in other words,

Theorem 4. *Given $x \in \mathcal{EAP}$, its regulator f and some $l \geqslant \mathrm{pr}(x)$, it is impossible to find algorithmically $\mathrm{pr}(x)$ precisely.*

The following result is similar to the result of Theorem 3.

Theorem 5. *Given $x \in \mathcal{AP}$ and its regulator f, it is impossible to determine algorithmically whether x is automatic. The same is with morphic.*

The contrary problem of determining whether a given morphic word is almost periodic is still open in general case. The particular case of automatic words or pure morphic words is solved in [17, 14].

4 Monadic Theories

As we could see in the previous section, even in a sequence together with its regulator there is no enough information about this sequence: some very natural characteristics can not be expressed. However, if we restrict our expressibility power up to monadic second-order language, we will see that an almost periodic sequence and its almost periodicity regulator are exactly what we need to express everything that we can about this sequence (Theorem 6).

Generally, combinatorics on infinite words is closely connected with the theory of second order monadic logics on natural numbers. Here we just want to show some examples of these connections. More details can be found, e. g., in [21, 22, 23].

We consider monadic logics on $\mathbb{N}$ with the relation "$<$", that is, first-order logics on natural numbers with order where also unary finite-value function variables and quantifiers over them are allowed. We also suppose that we know some fixed finite-value function $x\colon \mathbb{N} \to \Sigma$ and can use it in our formulas. Such a theory is denoted by $\mathrm{MT}\langle \mathbb{N}, <, x\rangle$ and is called *monadic theory of x*.

The main question here can be the question of decidability, that is, does there exist an algorithm that given a sentence in a theory says whether this sentence is true of false.

Theorem 6 (Semenov 1983 [22]). *If x is almost periodic, then $\mathrm{MT}\langle \mathbb{N}, <, x\rangle$ is decidable iff x and some its almost periodicity regulator are computable.*

It is curious that now using Theorem 6 we can obtain some corollaries from the results of Section 3.

Corollary 7. *There is no monadic formula $\phi(x)$ over a word x that for $x \in \mathcal{EAP}$ expresses some $l \geqslant \mathrm{pr}(x)$.*

Corollary 8. *There is no monadic formula $\phi(x)$ over a word x that for $x \in \mathcal{GAP}$ expresses the property $x \in \mathcal{EAP}$.*

Corollary 9. *There is no monadic formula $\phi(x)$ over a word x that for $x \in \mathcal{AP}$ expresses the property $x \in \mathcal{P}$.*

Corollary 10. *There is no monadic formula $\phi(x, l)$ over a word x and a natural number l that for $x \in \mathcal{EAP}$ and some $l \geqslant \mathrm{pr}(x)$ expresses the exact value $\mathrm{pr}(x)$.*

Each of the corollaries shows some property to be inexpressible by a uniform formula in monadic language for all x simultaneously. However, in each theory $\mathrm{MT}\langle\mathbb{N}, <, x\rangle$ the same property is obviously expressible, independently of other theories. For example, Corollary 7 says that there is no a unique formula $\phi(x)$ that for $x \in \mathcal{EAP}$ expresses some $l \geqslant \mathrm{pr}(x)$. However, in each $\mathrm{MT}\langle\mathbb{N}, <, x\rangle$ for $x \in \mathcal{EAP}$ such l is a constant natural number that can be expressed.

It seems interesting to continue research in this direction and to describe properties of sequences that can be uniformly expressed in monadic language.

5 Kolmogorov Complexity

Kolmogorov complexity of a finite word is informally the amount of information in this word. More formally, it is the length of the shortest binary description for this word, where the resulting word is obtained from the description using some universal algorithmic decompression function, for instance, universal Turing machine. In other words, Kolmogorov complexity of a word is the length of the shortest program without input in some natural programming language that prints this word. Kolmogorov complexity is defined up to an additive constant (depending on particular universal machine that one considers). More details and formalities for this extremely important and interesting notion can be found in [9] or [25]. We denote by $K(u)$ the complexity of a word u. In this section all finite and infinite words are supposed to be binary. Recall one of the simplest properties of Kolmogorov complexity: $K(u) < |u| + C$ for all $u \in \mathbb{B}^*$, for every word can be described at least by this word itself.

Although Kolmogorov complexity is used to measure the amount of information in words (as well as other finite constructive objects), and therefore applications and connections of Kolmogorov complexity to combinatorics on words might be of great interest and importance, it looks very strange that almost nothing is done in this direction. Among recent works we can mention only [19] (see discussion below) and [20] where new proof of existence of sequences with arbitrary critical exponent is given (originally proved in [8]).

The aim of this section is to discuss the following (still open) problem raised by Andrej Muchnik: does there exist an almost periodic sequence x with computable regulator such that all its subwords have high Kolmogorov complexity, namely, $K(u) > \alpha|u|$ for some $0 < \alpha < 1$ and for all subwords u of x? Muchnik conjectured that such sequences exist for all $\alpha < 1$. Properties of such sequences should combine some computability (computable almost periodicity regulator) and some randomness (almost maximal Kolmogorov complexity of subwords). However, such sequence is not really computable, since it would imply logarithmic complexity of all prefixes. It is also not really random (in Martin-Löf sense, this means having all prefixes of maximal Kolmogorov complexity), since it would imply $\mathrm{Fac}(x) = \mathbb{B}^*$, which is not the case for almost periodic sequences.

Many very close results were already obtained. Levin lemma says that for every α such that $0 < \alpha < 1$ there exists an infinite binary sequence x such that $K(u) > \alpha|u|$ for all factors u of x (see [5] or [19]). Since for every sequence there exists an almost periodic one with factors that are factors of initial sequence (minimality property), Levin lemma implies existence of almost periodic sequence with above property. The problem now is to find such sequence with computable almost periodicity regulator.

In fact, some weaker characteristic of almost periodic sequence can be made computable in this situation. Let x be almost periodic. Let $\mathrm{R}'_x(n)$ be some l such that $x[0, n-1]$ appears on each segment of length l in x (see also [3]). Obviously, for every $x \in \mathcal{AP}$ if some R_x is computable, then some $\mathrm{R}'_x(n)$ is also computable. As it can be seen after analyzing the paper [19], for every α, $0 < \alpha < 1$, there exists a binary sequence x such that $K(u) > \alpha|u|$ for all factors u of x, and such that some R'_x is computable. As the following theorem shows, the gap between this result and the desired conjecture is significant.

Theorem 11. *There exists an almost periodic sequence x such that R'_x is computable, but no its almost periodicity regulator is computable.*

A weaker version of Muchnik's conjecture can be obtained if we consider not all factors, but only all prefixes. After analyzing the arguments from [13] and changing some details, one can prove that for every α, $0 < \alpha < 1$, there exists a binary sequence x such that $K(u) > \alpha|u|$ for all prefixes u of x, and such that some its almost periodicity regulator is computable. The technics of [13] is a possible way to prove Muchnik's conjecture.

Very natural notion of subword complexity is often used to characterize infinite words. The following lemmas describe some connections between Kolmogorov and subword complexity.

Lemma 12. *For every n there exists C such that for all $m \geqslant n$ for all $u \in \mathrm{Fac}_m(x)$ we have $K(u) \leqslant \frac{m}{n}\log(p_x(n)) + C$.*

Lemma 13. *For every n there exists $u \in \mathrm{Fac}_n(x)$ such that $K(u) \geqslant \log(p_x(n))$.*

Recall that $\lim_{n\to\infty} \frac{1}{n}\log(p_x(n))$ always exists and is called *topological entropy* of a word (see [6]); we denote it by $E_t(x)$. This is a real number between 0 and 1 somehow describing how determinant sequence is: the more close to 0 the number $E_t(x)$ is, the more determinant a sequence x is. If one wants to define something similar in terms of Kolmogorov complexity, then $E_k(x) = \lim_{n\to\infty} \frac{1}{n}\max\{K(u) : |u| \in \mathrm{Fac}_n(x)\}$ might be a good choice. From this definition it is even not clear when $E_k(x)$ exists, but from Lemmas 12 and 13 it follows that $E_k(x)$ exists for all x and is equal to $E_t(x)$.

Another important corollary from Levin lemma and Lemma 13 is that for every α, $0 < \alpha < 1$, there exists an almost periodic sequence x with $E_t(x) > \alpha$. Moreover, such sequence can be made with computable regulator, that follows from Lemma 13 applied to a sequence with prefixes of high Kolmogorov complexity.

However, Muchnik's conjecture in a general form is still open.

6 Conclusion

The notion of information in infinite sequence is of great interest. It seems that in all directions mentioned in this paper significant and interesting further results can be obtained (even if this motivation about "information in infinite word" looks artificial, the problems look interesting themselves). Probably the most important point is to establish connections of Kolmogorov complexity and combinatorics on words and to find new applications from each of these two fields in another one.

Another problem which is not mentioned in the main text of this paper in the most general formulation is the following. Suppose we have a sequence and some transformation of this sequence.

When this transformation does not lose information from this sequence, i. e., the image has the same information? In other words, when does there exist a transformation of the same type that applied to image returns an initial sequence? In particular, consider transformations of the simplest type, namely, morphisms. When for a sequence x and a morphism h does there exist a morphism g such that $g(h(x)) = x$? Is the problem of existence of such morphism decidable? How many such morphisms might be for a fixed sequence?

7 Acknowledgements

The author is grateful to An. Muchnik and A. Semenov for their permanent help in the work, to A. Shen for discussions about Section 5, and to all other participants of Kolmogorov seminar, Moscow [7] and also to J. Karhumäki, K. Saari for useful discussions. Some of the results of this paper were already presented at WoWA'06 (satellite to CSR'06), St. Petersburg, Russia, and WAW'06, Turku, Finland, the author thanks audience for attention.

References

[1] J.-P. Allouche, J. Shallit. *The ubiquitous Prouhet–Thue–Morse sequence.* Sequences and their applications, Proceedings of SETA'98, Springer Verlag, pp. 1–16, 1999.

[2] J.-P. Allouche, J. Shallit. *Automatic Sequences.* Cambridge University Press, 2003.

[3] J. Cassaigne. *Recurrence in infinite words.* Proceedings of STACS'01, LNCS 2010, pp. 1–11, 2001.

[4] A. Cobham. *Uniform tag sequences.* Math. Systems Theory, 6, pp. 164–192, 1972.

[5] B. Durand, L. Levin, A. Shen. *Complex tilings.* Proceedings of STOC 2001; enhanced version on `http://arXiv.org/abs/cs.CC/0107008`.

[6] S. Ferenczi. *Complexity of sequences and dynamical systems.* Discrete Math. 206, pp. 145–154, 1999.

[7] Kolmogorov Seminar: `http://lpcs.math.msu.su/kolmogorovseminar/eng/`.

[8] D. Krieger, J. Shallit. *Every real number greater than 1 is a critical exponent.* Theoretical Computer Science, to appear, 2007.

[9] M. Li, P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications.* Second Edition, Springer Verlag, 1997.

[10] M. Lothaire. *Algebraic Combinatorics on Words.* Cambridge University Press, 2002.

[11] M. Morse, G. A. Hedlund. *Symbolic dynamics.* American Journal of Mathematics, 60, pp. 815–866, 1938.

[12] M. Morse, G. A. Hedlund. *Symbolic dynamics II: Sturmian trajectories.* American Journal of Mathematics, 62, pp. 1–42, 1940.

[13] An. Muchnik, A. Semenov, M. Ushakov. *Almost periodic sequences.* Theoretical Computer Science, vol. 304, pp. 1–33, 2003.

[14] F. Nicolas, Yu. Pritykin. *On Almost Periodic Morphic Sequences.* Preprint, 2007.

[15] Yu. L. Pritykin. *Finite automata mappings of strongly almost periodic sequences and algorithmic undecidability.* Proceedings of XXVIII Conference of Young Scientists, Moscow State University, Faculty of Mechanics and Mathematics, pp. 177–181, 2006 (in Russian).

[16] Yu. L. Pritykin. *Finite-Automaton Transformations of Strictly Almost-Periodic Sequences.* Mathematical Notes, vol. 80, N 5, pp. 710–714, 2006.

[17] Yu. Pritykin. *On Almost Periodicity Criteria for Morphic Sequences in Some Particular Cases.* Proc. of DLT 2007, LNCS 4588, pp. 361–370. Preprint on `http://arxiv.org/abs/0704.0218`.

[18] H. Rogers. *Theory of Recursive Functions and Effective Computability.* The MIT Press, 1967.

[19] A. Rumyantsev, M. Ushakov. *Forbidden Substrings, Kolmogorov Complexity and Almost Periodic Sequences.* Proceedings of STACS'06, LNCS 3884, pp. 396–407.

[20] A. Rumyantsev. *Kolmogorov complexity, Lovasz local lemma and critical exponent.* Accepted to CSR 2007, Ekaterinburg, Russia.

[21] A. L. Semenov. *On certain extensions of the arithmetic of addition of natural numbers.* Math. of USSR, Izvestia, vol. 15, pp. 401–418, 1980.

[22] A. L. Semenov. *Logical theories of one-place functions on the set of natural numbers.* Math. of USSR, Izvestia, vol. 22, pp. 587–618, 1983.

[23] A. L. Semenov. *Decidability of Monadic Theories.* Proc. MFCS'84, LNCS 176, pp. 162–175, 1984.

[24] A. Thue. *Über unendliche Zeichenreihen.* Norske vid. Selsk. Skr. Mat. Nat. Kl., 7, pp. 1–22, 1906. Reprinted in *Selected mathematical papers of Axel Thue*, Universitetsforlaget, Oslo, pp. 139–158, 1977.

[25] V. A. Uspenskiy, A. Shen, N. K. Vereschagin. *Kolmogorov Complexity.* Unpublished book: `http://lpcs.math.msu.su/kolmogorovseminar/eng/materials/materials.html`