ISSN (Print): 2077-9879
ISSN (Online): 2617-2658

Eurasian Mathematical Journal

2022, Volume 13, Number 1

Founded in 2010 by
the L.N. Gumilyov Eurasian National University
in cooperation with
the M.V. Lomonosov Moscow State University
the Peoples' Friendship University of Russia (RUDN University)
the University of Padua

Starting with 2018 co-funded
by the L.N. Gumilyov Eurasian National University
and
the Peoples' Friendship University of Russia (RUDN University)

Supported by the ISAAC
(International Society for Analysis, its Applications and Computation)
and
by the Kazakhstan Mathematical Society

Published by

the L.N. Gumilyov Eurasian National University
Nur-Sultan, Kazakhstan

Aims and Scope

The Eurasian Mathematical Journal (EMJ) publishes carefully selected original research papers in all areas of mathematics written by mathematicians, principally from Europe and Asia. However papers by mathematicians from other continents are also welcome.

From time to time the EMJ publishes survey papers.

The EMJ publishes 4 issues in a year.

The language of the paper must be English only.

The contents of the EMJ are indexed in Scopus, Web of Science (ESCI), Mathematical Reviews, MathSciNet, Zentralblatt Math (ZMATH), Referativnyi Zhurnal – Matematika, Math-Net.Ru.

The EMJ is included in the list of journals recommended by the Committee for Control of Education and Science (Ministry of Education and Science of the Republic of Kazakhstan) and in the list of journals recommended by the Higher Attestation Commission (Ministry of Education and Science of the Russian Federation).

Information for the Authors

Submission. Manuscripts should be written in LaTeX and should be submitted electronically in DVI, PostScript or PDF format to the EMJ Editorial Office through the provided web interface (www.enu.kz).

When the paper is accepted, the authors will be asked to send the tex-file of the paper to the Editorial Office.

The author who submitted an article for publication will be considered as a corresponding author. Authors may nominate a member of the Editorial Board whom they consider appropriate for the article. However, assignment to that particular editor is not guaranteed.

Copyright. When the paper is accepted, the copyright is automatically transferred to the EMJ. Manuscripts are accepted for review on the understanding that the same work has not been already published (except in the form of an abstract), that it is not under consideration for publication elsewhere, and that it has been approved by all authors.

Title page. The title page should start with the title of the paper and authors' names (no degrees). It should contain the Keywords (no more than 10), the Subject Classification (AMS Mathematics Subject Classification (2010) with primary (and secondary) subject classification codes), and the Abstract (no more than 150 words with minimal use of mathematical symbols).

Figures. Figures should be prepared in a digital form which is suitable for direct reproduction.

References. Bibliographical references should be listed alphabetically at the end of the article. The authors should consult the Mathematical Reviews for the standard abbreviations of journals' names.

Authors' data. The authors' affiliations, addresses and e-mail addresses should be placed after the References.

Proofs. The authors will receive proofs only once. The late return of proofs may result in the paper being published in a later issue.

Offprints. The authors will receive offprints in electronic form.

Publication Ethics and Publication Malpractice

For information on Ethics in publishing and Ethical guidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the EMJ implies that the work described has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The EMJ follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/NewCode.pdf). To verify originality, your article may be checked by the originality detection service CrossCheck http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will only accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the EMJ.

The Editorial Board of the EMJ will monitor and safeguard publishing ethics.

The procedure of reviewing a manuscript, established by the Editorial Board of the Eurasian Mathematical Journal

1. **Reviewing procedure**

1.1. All research papers received by the Eurasian Mathematical Journal (EMJ) are subject to mandatory reviewing.

1.2. The Managing Editor of the journal determines whether a paper fits to the scope of the EMJ and satisfies the rules of writing papers for the EMJ, and directs it for a preliminary review to one of the Editors-in-chief who checks the scientific content of the manuscript and assigns a specialist for reviewing the manuscript.

1.3. Reviewers of manuscripts are selected from highly qualified scientists and specialists of the L.N. Gumilyov Eurasian National University (doctors of sciences, professors), other universities of the Republic of Kazakhstan and foreign countries. An author of a paper cannot be its reviewer.

1.4. Duration of reviewing in each case is determined by the Managing Editor aiming at creating conditions for the most rapid publication of the paper.

1.5. Reviewing is confidential. Information about a reviewer is anonymous to the authors and is available only for the Editorial Board and the Control Committee in the Field of Education and Science of the Ministry of Education and Science of the Republic of Kazakhstan (CCFES). The author has the right to read the text of the review.

1.6. If required, the review is sent to the author by e-mail.

1.7. A positive review is not a sufficient basis for publication of the paper.

1.8. If a reviewer overall approves the paper, but has observations, the review is confidentially sent to the author. A revised version of the paper in which the comments of the reviewer are taken into account is sent to the same reviewer for additional reviewing.

1.9. In the case of a negative review the text of the review is confidentially sent to the author.

1.10. If the author sends a well reasoned response to the comments of the reviewer, the paper should be considered by a commission, consisting of three members of the Editorial Board.

1.11. The final decision on publication of the paper is made by the Editorial Board and is recorded in the minutes of the meeting of the Editorial Board.

1.12. After the paper is accepted for publication by the Editorial Board the Managing Editor informs the author about this and about the date of publication.

1.13. Originals reviews are stored in the Editorial Office for three years from the date of publication and are provided on request of the CCFES.

1.14. No fee for reviewing papers will be charged.

2. **Requirements for the content of a review**

2.1. In the title of a review there should be indicated the author(s) and the title of a paper.

2.2. A review should include a qualified analysis of the material of a paper, objective assessment and reasoned recommendations.

2.3. A review should cover the following topics:

- compliance of the paper with the scope of the EMJ;

- compliance of the title of the paper to its content;

- compliance of the paper to the rules of writing papers for the EMJ (abstract, key words and phrases, bibliography etc.);

- a general description and assessment of the content of the paper (subject, focus, actuality of the topic, importance and actuality of the obtained results, possible applications);

- content of the paper (the originality of the material, survey of previously published studies on the topic of the paper, erroneous statements (if any), controversial issues (if any), and so on);

- exposition of the paper (clarity, conciseness, completeness of proofs, completeness of bibliographic references, typographical quality of the text);

- possibility of reducing the volume of the paper, without harming the content and understanding of the presented scientific results;

- description of positive aspects of the paper, as well as of drawbacks, recommendations for corrections and complements to the text.

2.4. The final part of the review should contain an overall opinion of a reviewer on the paper and a clear recommendation on whether the paper can be published in the Eurasian Mathematical Journal, should be sent back to the author for revision or cannot be published.

Web-page

The web-page of the EMJ is www.emj.enu.kz. One can enter the web-page by typing Eurasian Mathematical Journal in any search engine (Google, Yandex, etc.). The archive of the web-page contains all papers published in the EMJ (free access).

Subscription

Subscription index of the EMJ 76090 via KAZPOST.

E-mail

eurasianmj@yandex.kz

The Eurasian Mathematical Journal (EMJ)
The Nur-Sultan Editorial Office
The L.N. Gumilyov Eurasian National University
Building no. 3
Room 306a
Tel.: +7-7172-709500 extension 33312
13 Kazhymukan St
010008 Nur-Sultan, Kazakhstan

The Moscow Editorial Office
The Peoples' Friendship University of Russia
(RUDN University)
Room 473
3 Ordzonikidze St
117198 Moscow, Russia

VAGIF SABIR oglu GULIYEV

(to the 65th birthday)

On February 22, 2022 was the 65th birthday of Vagif Sabir oglu Guliyev, editor-in-chief of the Transactions of the Azerbaijan National Academy of Science, Issue Mathematics, Series of physical-technical and mathematics science (Scopus, Q3), deputy editor-in-chief of the Applied and Computational Mathematics (Web of Science, Q1), deputy director of the Institute of Applied Mathematics (IAM) of the Baku State University (BSU), head of the Department of Mathematical Analysis at the Institute of Mathematics and Mechanics (IMM) of the Azerbaijan National Academy of Sciences (ANAS), member of the Editorial Board of the Eurasian Mathematical Journal.

V.S. Guliyev was born in the city of Salyan in Azerbaijan. In 1978 Vagif Guliyev graduated from the Faculty of Mechanics and Mathematics of the Azerbaijan State University (now the Baku State University) with an honors degree and then completed his postgraduate studies at this university. His scientific supervisors were distinguished mathematicians A.A. Babayev and S.K. Abdullayev. In 1983 he defended his PhD thesis at the BSU. From 1983 he continued his scientific activities at the V.A. Steklov Mathematical Institute of the Academy of Sciences of the USSR. In 1987-1991 he was in internship at this institute and in 1994 defended there his DSc thesis.

From 1983 to 1995 he worked as assistant, a senior lecturer, docent and from 1995 to 2018 as a professor of Mathematical Analysis Chair of the BSU. In 1995-2008 he worked on part-time basis at the Institute of the IMM. From 2008 to 2014 he was a chief researcher of the Department of Mathematical Analysis of the IMM, from 2014 to the present day he is the head of this department.

In 2014 V.S. Guliyev was elected a corresponding member of the ANAS.

From 2015 to 2019, he worked as deputy director on science at the IMM. From 2019 to the present day, he has been working as a chief researcher at the IAM. Since May 2021, he has been working as a deputy director on science of the IAM.

Professor Vagif Guliyev has been a member of the Presidium of the Higher Attestation Commission under the President of the Republic of Azerbaijan since 2014 to the present day.

V.S. Guliyev is a world recognized specialist in real and harmonic analysis, function spaces and partial differential equations. He obtained seminal scientific results in several areas of functional analysis and the theory of partial differential and integral equations. He was one of the first to study local Morrey-type spaces, generalized weighted Morrey-type spaces and anisotropic Banach-valued Sobolev spaces, for which appropriate embedding theorems were established.

Some of his results and methods are named after him: the Adams-Guliyev and Spanne-Guliyev conditions for the boundedness of operators in Morrey-type spaces, Guliyev's method of local estimates of integral operators of harmonic analysis, the Burenkov-Guliyevs conditions for the boundedness of operators in general local Morrey-type spaces.

On the whole, the results obtained by V.S. Guliyev have laid a groundwork for new perspective scientific directions in the theory of functions spaces and its applications to partial differential equations.

Vagif Sabir oglu Guliyev is an author of more than 250 scientific publications including 2 monographs. Among his pupils there are more than 20 candidates of sciences and 5 doctors of sciences. The results obtained by V.S. Guliyev, his pupils, collaborators and followers gained worldwide recognition.

The mathematical community, many his friends and colleagues and the Editorial Board of the Eurasian Mathematical Journal cordially congratulate Vagif Sabir oglu Guliyev on the occasion of his 65th birthday and wish him good health, happiness and new achievements in mathematics and mathematical education.

EURASIAN MATHEMATICAL JOURNAL
ISSN 2077-9879
Volume 13, Number 1 (2022), 44 – 68

THE RECOGNITION COMPLEXITY OF DECIDABLE THEORIES

I.V. Latkin

Communicated by J.A.Tussupov

Key words: decidable theories, the theory of equality, the coding of computations, polynomial time, polynomial space, lower complexity bound.

AMS Mathematics Subject Classification: 03C40, 03C07, 03D15, 68Q05, 68Q15.

Abstract. We will find a lower bound on the recognition complexity of the decidable theories that are nontrivial relative to equality, namely, each of these theories is consistent with the formula, whose sense is that there exist at least two distinct elements. However, at first, we will obtain a lower bound on the computational complexity for the first-order theory of the Boolean algebra that contains only two elements. For this purpose, we will code the long-continued deterministic Turing machine computations by the relatively short-length quantified Boolean formulae; the modified Stockmeyer and Meyer method will appreciably be used for this simulation. Then, we will construct a polynomial reduction of this theory to the first-order theory of pure equality.

DOI: https://doi.org/10.32523/2077-9879-2022-13-1-44-68

1 Introduction

At the beginning, we recall some designations. The function $\exp_k(n)$ is called *k-iterated* (or *k-storey*, or *k-fold*) exponential, if, for every natural k and n, it is calculated in the following way: $\exp_1(n)=2^n$, $\exp_{k+1}(n)=2^{\exp_k(n)}$ [21]. The length of a word X is denoted by $|X|$, i.e., $|X|$ is the number of symbols in X. If A is a set, then $|A|$ denotes its cardinality; "$A \risingdotseq \mathcal{A}$" means "$A$ is a designation for $\mathcal{A}$"; and $\exp(n) \risingdotseq \exp_1(n)$.

Let f be a non-decreasing function mapping a natural number to a natural number, i.e., $f : \mathbb{N} \to \mathbb{N}$. Then $DTIME(f(n))$ (or $TIME(f(n))$) is the class containing all languages, which are recognized by the deterministic Turing machines within $\mathcal{O}(f(n))$ steps (or in specified time). Here and bellow, the variable n is the length of the input string. The class $DSPACE(f(n))$ (or $SPACE(f(n))$) consists of all the languages that are recognized by such machines using the $\mathcal{O}(f(n))$ tape cells (i.e., using specified amount of memory or space). The complexity class **P** consists of all languages which can be recognized by the deterministic Turing machines in polynomial time, more precisely, $\mathbf{P} = \bigcup_{k\in\mathbb{N}} DTIME(n^k)$. The class **PSPACE** consists of all languages that can be recognized by such machines using a polynomial amount of space, i.e., $\mathbf{PSPACE} = \bigcup_{k\in\mathbb{N}} DSPACE(n^k)$. See also [1], [2] and [9] for details.

1.1 Problem statement

The results on the complexity of recognition (or computational complexity, or *inherent complexity* according to [17]) are well-known for many decidable theories [7], [14]–[19]. We recall only some of these results concerning first-order theories.

The language $ThRLC$ (the theory of the field $\mathbb{R}$ of real numbers), and even the $Th(\mathbb{R},+)$ (the theory of the additive group of this field), has the *complexity of recognition* that is more than exponential [7], namely, there exists a rational constant $d_1 > 0$, such that if P is a deterministic Turing machine which recognizes the theory $ThRLC$ (or $Th((\mathbb{R},+))$), then the P runs for at least $2^{d_1|\varphi|}$ steps when started on input φ, for infinitely many sentences φ. So, these theories do not belong to $DTIME(\exp(d_1 n))$; here and below, the letter d with subscripts denotes a suitable constant. For the Presburger arithmetic PAR (the theory of natural numbers with addition) and the Skolem arithmetic SAR (the theory of natural numbers with multiplication), the recognition complexity is more than a double exponential: $PAR, SAR \notin DTIME(\exp_2(d_2 n))$. For the theory of linearly ordered sets $ThOR$, the computational complexity is very great [15]: $ThOR \notin DTIME(\exp_{\lfloor d_3 n\rfloor}(n))$, where $\lfloor y \rfloor$ is the integer part of a number y.

If we go beyond the confines of logical theories of the first order, we can see more enormous lower bounds on the recognition complexity. An example of such an estimate is the lower bound for the weak monadic second-order theory of one successor $WSIS$, other examples can be found in [4], [14], [17], [19]. However, according to the author, the most impressive estimate of this kind was obtained by Vorobyev in [21] for the type theory Ω, which is a rudimentary fragment of the theory of propositional types due to Henkin: $\Omega \notin DSPACE(\exp_\infty(\exp(d_4 n)))$, hence $\Omega \notin DTIME(\exp_\infty(\exp(d_4 n)))$, where the function $\exp_\infty$ is recursively defined by $\exp_\infty(0) = 1$ and $\exp_\infty(k+1) = 2^{\exp_\infty(k)}$, i.e., this lower bound has the exponentially growing stack of twos.

But what is the recognition complexity of the simplest decidable, but non-trivial theories? Shall such theories be quickly decidable, i.e., polynomial?

One of the simplest theories is the first-order theory of the algebraic structure with two elements and a unique equality predicate. We will see in Section 7 that even this theory does not have a polynomial upper bound of computational complexity. We will in passing obtain the lower bounds on the recognition complexity of the theories that are nontrivial concerning some equivalence relation $\backsim$, namely, these theories have models with at least two elements that are not $\backsim$-equivalent. Obvious examples of such theories are the theories of pure equality and one equivalence relation.

Since the lower bound on the computational complexity of these theories is not polynomial, we obtain that the class **P** is a proper subclass of **PSPACE**.

1.2 Used methods and the main idea

The lower bounds on the computational complexity for the theories mentioned in the previous subsection and some others were yielded by the techniques of the immediate codings of the machine actions by means of formulae in [7], [14]–[19], [21]. The essence of these methods (we will call them the Rabin and Fischer methods) is as follows [17].

Let T be the theory written in the signature (or *underlying language* [17]) σ. Assume that, for any input string X and every program P of the Turing machine, one can write a sentence $S(P,X)$, of the σ, satisfying the following conditions. There exist a constant $d>0$ and a function f such that: (i) $|S(P,X)| < d(|X|+|P|)$; (ii) $S(P,X) \in T$ if and only if a computation by the program P accepts the input X in fewer than $f(|X|)$ steps; (iii) the formula $S(P,X)$ can be effectively constructed from X and P in fewer than $g(|X|+|P|)$ steps, where $g(k)$ is a fixed polynomial. If $f(k)$ is a function growing at least at exponential rate, then under the above conditions, there exist a constant $C>0$ and infinitely many sentences φ of σ, for which every Turing machine requires at least $f(C|\varphi|)$ steps to decide whether $\varphi \in T$, i.e., $T \notin DTIME(f(Cn))$.

We will below scrutinize this method in more detail and a somewhat more general form than this was done in the previous paragraph or Subsection 4.1 in [17]. We need the more general form of this technique for the following reason.

Our main purpose is to evaluate the computational complexity of an equality theory $Th\mathcal{E}$ and other equational-nontrivial decidable theories (Section 7). However, at first, we will obtain a lower bound on recognition complexity for $Th\mathcal{B}$, which is the first-order theory of Boolean algebra $\mathcal{B}$ that has only two elements, using the Rabin and Fischer method. Then, we will construct a polynomial reduction of the $Th\mathcal{B}$ to $Th\mathcal{E}$. In Subsection 8.1, we will explain why such a succession of actions is applied.

But the first-order theory of two-element Boolean algebra has a very weak expressive ability. Therefore, the modeling sentence for this theory, i.e., the formula possessing property (ii) from the method described above, does not turn out to be very short, it may have not a linear restriction on its length (see Subsection 6.4 for more details). Furthermore, the $Th\mathcal{B}$ seems so poor and meager that there can, in general, be a doubt about the very possibility of the simulation of the sufficiently long computations using the relatively short formulae of this theory.

Nevertheless, such modeling was well-known a long time ago. Stockmeyer and Meyer showed in 1973 that a language $TQBF$consisting of the true quantified Boolean formulae (corresponding problem is designated as QBF, or sometimes $QSAT$) is polynomially complete in the class **PSPACE** [9], [20]. This implies in particular that for every language $\mathcal{L}$ in this class, there is an algorithm, which produces a quantified Boolean formula for any input string in polynomial time; and all these sentences model the computations that recognize the $\mathcal{L}$ and use the polynomial amount of space. Namely, each of them is true if and only if the given input string belongs to the $\mathcal{L}$; at that, the long enough computations are simulated, as the polynomial constraint on memory allows the machine to run during the exponential time [1], [2], [9].

The method, which is employed in the proof of Theorem 4.3 in [20] for the implementation of this simulation, permits writing down a polynomially bounded formula for the modeling of the exponential quantity of the Turing machine steps provided that one step is described by the formula, whose length is polynomial. One running step of the machine is described in [20] by the Cook's method formula; this is a formula of the propositional calculus, and one can construct it just as the sentences, which were applied for the modeling of the polynomial quantity of steps of the nondeterministic Turing machine in [5], (see also the proof of Theorem 10.3 in [1]). There exists a Boolean ∃-formula, which corresponds to the Cook's method formula. We will name this ∃-formula as the Cook formula.

We intend to bring into play the construct of Stockmeyer and Meyer for obtaining our purpose. However, we will model the running steps of a machine by the more complicated formulate that have an alternation of quantifiers. This complication is caused because Cook's method formula is very long — it is far longer than an amount of the used memory. It has a subformula consisting of one propositional variable $C_{i,j,t}$ (see, for instance, the proof of Theorem 10.3 in [1]). This variable is true if and only if the ith cell contains symbol X_j of the tape alphabet at the instant t. But suppose that each of the first $T+1$ squares of tape contains the symbol X_0 at time t, the remaining part of the tape is empty. This simple tape configuration (or *instantaneous description* [1], [20]) is described by the formula that has a fragment $C_{0,0,t} \wedge C_{1,0,t} \wedge \ldots \wedge C_{T,0,t}$, and one this subformula only is $2T+1$ in length without taking the indices into account. It is impossible to abridge this record, even if we try to use the universal quantifier since its application to the indices is not allowed within the confines of the first-order theory. Thus, in order to describe the machine actions on the exponential fragment of tape, we need Cook's method formula, whose length is still more.

We propose to encode the binary notation of the cell number by a value set of special variables $x_{t,0}, \ldots, x_{t,n}$, where $n+1 \geqslant \log_2 T$ (see Subsections 4.1, 6.2, and 8.2 for further details). So we need the $\mathcal{O}(n)$ various symbols (without indices) for the describing of one cell, and $\mathcal{O}(n^2)$ ones for the representation of the whole input string X, if $n = |X|$. Then we can describe one running step of the machine, which uses $T \rightleftharpoons \exp(|X|)$ memory cells on input X, with the aid a formula that is less than $\mathcal{O}(n^3)$ in length. The main idea of so brief a describing consists of the following: merely one tape square can change on each of the running steps, although the whole computation can use the exponential amount of memory. Therefore, it is enough for us to describe the changes in the only cell, and the contents of the remaining ones can be "copied" by applying the universal quantifier (see the construction of the formula $\Delta^{cop}(\widehat{u})$ in Subsection 4.1). The denoted locality of the actions of deterministic machines has long been used in the modeling of the machine computation with the help of formulae, see, for example, Lemma 2.14 in [19] or Lemma 7 in [21].

At the beginning, we will introduce all variables in great abundance in order to facilitate the proof, namely, the variables will have the first indices t from 0 to T. Next, we will eliminate many of the variables using the method from [20]) — see Subsections 4.1, 4.3, 6.2, 8.2 for further details.

The description of the initial configuration and the condition of the successful termination of computations have a length of $\mathcal{O}(n^3)$, if we anew use the quantifiers; hence the entire formula, which simulates the first $\exp(n)$ steps of the computation of the machine P, will be no more than $\mathcal{O}(|P|\cdot n^3)$ in length (taking into account the indices).

Therefore, we need to slightly strengthen the Rabin and Fischer method, so that it can also be applied in the case of a non-linearly bounded modeling formula.

The paper structure. The generalized Fisher and Rabin method is adduced in Section 2. The degree of its usefulness and novelty is discussed in Remark 1. Section 3 contains an exact formulation of the main theorem (Theorem 3.1), its primary corollaries, and some preparation for its proof. Sections 4–6 are devoted to the proof of the main theorem. The lower bound on the computational complexity of the theories, which are nontrivial relatively some equivalence relation, will be yielded in Section 7. In Section 8, we will discuss the obtained results and consider the used methods in greater detail, comparing them with other approaches to the simulation of computation. The brief list of the open problems concludes the paper.

2 The generalized Fischer and Rabin method

We will describe this method in the most general form.

2.1 Auxiliary notions

We will need some new concepts. Let P be a program of the k-tapes Turing machine.

Definition 1. The instruction $q_b\alpha_1\alpha_2\dots\alpha_k \to q_j\beta_1\beta_2\dots\beta_k$ of the program P is termed *explicitly non-executable* and the internal state b *inaccessible (for the P)*, if this program does not contain the instructions of the form $q_l\gamma_1\gamma_2\dots\gamma_k \to q_b\delta_1\delta_2\dots\delta_k$.

It is not difficult to write such a machine program that contains some non-executable instructions, but all its internal states are accessible. On the other hand, the detection of the non-executable instructions, whose internal states are accessible, maybe is a very difficult task in some cases.

Let us assume that we have removed all the explicitly non-executable instructions from a program P. The elimination has resulted in some program P_1. This P_1 may again contain some explicitly non-executable instructions, for instance, if the instructions $q_l\gamma_1\gamma_2\dots\gamma_k \to q_b\delta_1\delta_2\dots\delta_k$ and $q_b\alpha_1\alpha_2\dots\alpha_k \to q_j\beta_1\beta_2\dots\beta_k$ belong to P, the first of them is explicitly non-executable for the P, and the state b is not included in other instructions, therefore the second instruction is non-executable for the P, but not explicitly. However, it already is explicitly non-executable for the program P_1. We can continue this removing process of the explicitly non-executable instructions until we obtain the *irreducible* program $r(P)$ that does not contain such instructions.

Let T and P be some programs. If $r(T)=r(P)$, then the P, T, and $r(P)$ are called the *clones* of each other. As usual, a Turing machine and its program are designated by a uniform sign. Therefore we will say that two Turing machines are *clones* of each other if their programs are so.

Lemma 2.1. *(i) There exists a polynomial $h(n)$ such that one can write the code of irreducible clone $r(P)$ within $h(|P|)$ steps for every program P.*

(ii) on each tape, all the actions of the machines P and T are identical with each other on the same inputs, when $r(T)=r(P)$.

Proof. It straightforwardly follows from definitions by direct calculation. □

Definition 2. Let $F(n)$ be a function that is monotone increasing on all sufficiently large n. The function F is called a *limit upper bound for the class of all polynomials* (LUBP) if, for any polynomial p, there is a number n such that the inequality $F(m) > p(m)$ holds for $m \geqslant n$, i.e., each polynomial is asymptotically smaller than F.

An obvious example of the limit upper bound for all polynomials is a s-iterated exponential for every $s \geqslant 1$. It is easy to see that if $F(x)$ is a LUBP, then the functions $F(x^m)$ and $F(mx)$ are also LUBPs for positive constant m; moreover, if $0<m<1$, then the function $F(x) - F(mx)$ is a LUBP. It follows from this that if $T(n)$ is a LUBP, then it grows at least exponentially in the sense that is considered in [4], namely, $T(dn)/T(n)$ tends to 0 as n tends to ∞ for $0 < d < 1$. Inverse assertion seemingly is valid too.

2.2 The generalization

Let us suppose that we want to find a lower bound on the recognition complexity of a language $\mathcal{L}$ over alphabet σ. First of all, we fix a finite tape alphabet A of Turing machines and the number k of their tapes.

We also fix a certain *polynomial encoding* of the strings over the alphabet σ and of the programs of Turing machines by finite strings of symbols (words) over the alphabet A, i.e., such encoding and unique decoding are realized in polynomial time from the length of an object in a *natural language*. This language consists of all words over the alphabet σ and all the Turing machines programs with k the tapes and the tape alphabet A. An example of such a natural language will be described in Subsection 3.1. We presume that the used encoding c_O is *composite*, namely, the code of each instruction in any program is the constituent of the program code. So, the code of an object E is denoted by $c_O E$, i.e., $c_O E \in A^*$, if $E \in \sigma^*$ or E is a program.

Proposition 2.1. *Let F be a limit upper bound for all polynomials and $\mathcal{L}$ be a language over some alphabet σ. Suppose that for any given program P of a Turing machine and every string X on the input tape of this machine, one can effectively construct a word $S(P,X)$, over the alphabet σ, possessing the following properties:*

(i) a code for $S(P,X)$ can be built within time $g(|X|+|c_O P|)$, where g is a polynomial fixed for all X and P;

(ii) the word $S(P,X)$ belongs to $\mathcal{L}$ if and only if the Turing machine P accepts input X within $F(|X|)$ steps;

(iii) there exist constants $D, b, s>0$ such that either the inequalities

(a) $|X| \leqslant |c_O S(P,X)| \leqslant D \cdot |c_O P|^b \cdot |X|^s$ or the inequalities

(b) $|X| \leqslant |c_O S(P,X)| \leqslant D \cdot (|c_O P|+|X|)$ hold for all sufficiently long X, and these constants do not depend on P, but they can depend on the applied encoding.

Then (1) for every constant $\delta > 0$ and any program P, there is a integer t_0 such that the inequality $|c_O S(P,X)| \leqslant D_1 \cdot |X|^{s_1}$ is true for all of the strings X, which are longer than t_0, where either $D_1 = D$ and $s_1 = s+\delta$ in case (a) or $D_1 = (D+\delta)$ and $s_1 = 1$ in case (b);

(2) for each $a>1$ and every deterministic Turing machine M, which recognizes the language $\mathcal{L}$, there exist infinitely many words Y, on which M runs for more than $F(D_2 \cdot |c_O Y|^{\rho})$ steps for $D_2 = (aD_1)^{-\rho}$ and $\rho = (s_1)^{-1}$.

Proof. (1). It is easy to see that the t_0 is equal to $|c_O P|^{b/\delta}$ in case (a); and it equals to $(D/\delta)\cdot|c_O P|$ in case (b).

(2) In accordance with condition (i), one can assume that a code for $S(P,X)$ can be written by some machine M_1 for all given strings X and $c_O P$.

Let us suppose that there exist numbers a, t_1 and a machine M_2 such that the M_2 determines whether $Y \in \mathcal{L}$ within $F(D_2 \cdot |c_O Y|^{\rho})$ steps for any string Y over the σ, provided that $|c_O Y| > t_1$ and $a > 1$.

To proceed to an ordinary diagonal argument, we stage-by-stage construct the Turing machine M. At the first stage, we write a machine M_0, which for a given input X, determines whether the string X is the code $c_O P$ of some program P. If not, then the M_0, as well as the whole machine M, rejects the X; else it writes the code $c_O r(P)$ of the irreducible clone $r(P)$.

At the second stage, the M_1 joins the running process and writes a word $c_O S(r(P), c_O P)$. At the next stage, the procedure M_2 determines whether the string $S(r(P), c_O P)$ belongs to the language $\mathcal{L}$. If it does not, then the M accepts the input $X = c_O P$. When the M_2 gives an affirmative answer, then M rejects the X.

We estimate the running time of M on input $X = c_O P$. Since c_O is a polynomial encoding and Lemma 2.1(i) is valid, there exists a polynomial h_1 such that the running time of M_0 does not exceed $h_1(|X|)$. The machine M_1 builds $c_O S(r(P), X)$ within $g(|X| + |c_O r(P)|) \leqslant g(2|c_O P|)$ steps, since $|c_O r(P)| \leqslant |c_O P|$; the stage M_2 lasts no longer than $F(D_2 \cdot |c_O S(r(P), c_O P)|^{\rho}) \leqslant F((D_1 \cdot |c_O P|^{s_1})^{\rho}/(aD_1)^{\rho}) = F(|c_O P|/a^{\rho})$ steps for $|c_O S(r(P), c_O P)| \geqslant |c_O P| > t_1$ by our assumption. Hence, the entire M will execute its work no more than $T(P) = h_1(|c_O P|) + g(2|c_O P|) + F(|c_O P|/a^{\rho}) < F(|c_O P|)$ steps for all sufficiently large $|c_O P|$.

Let us look at the situation that obtains, when the input string of M is the code of so lengthy a clone $\widehat{M}$ of the machine M itself that the inequalities $|c_O \widehat{M}| > \max\{t_0, t_1\}$ and $T(\widehat{M}) < F(|c_O \widehat{M}|)$ become true.

If the M rejects the input $c_O \widehat{M}$, then the M_2 answers affirmatively, i.e., the string $S(r(\widehat{M}), c_O \widehat{M})$ belongs to the language $\mathcal{L}$. According to the condition (ii), this means that the $r(\widehat{M})$ accepts the input $c_O \widehat{M}$ within $F(|c_O \widehat{M}|)$ steps. Since the machines M and $\widehat{M}$ are the clones of $r(\widehat{M})$, the M does it too. There appears a contradiction.

If the M accepts the $c_O \widehat{M}$ as its input, then the procedure M_2 answers negatively. In accordance with the sense of the formula $S(r(\widehat{M}), c_O \widehat{M})$, this signifies that the machine $r(\widehat{M})$ either rejects the $c_O \widehat{M}$ or its running time on this input is more than $F(|c_O \widehat{M}|)$. By construction and our assumption, the clone $r(\widehat{M})$ cannot operate so long. We have again arrived at a contradiction. □

Remark 1. The generalized Rabin and Fischer method, in essence, has been known in an implicit form for a long time. For example, it is said in the penultimate paragraph of the introduction of the article [21] (before the paragraph "Paper outline") that the quadratic increase in the length of the modeling formulae implies a lowering of the lower bound from $F(n)$ to $F(\sqrt{n})$ (in our notation), when Compton and Henson's method is applied. But the author could not find an explicit formulation of the statement similar to Proposition 2.1 for a reference, although its analog for the space complexity is Lemma 3 in [21]. The proof of the proposition is given only for the sake of completeness of the proof of Corollary 3.2. Proposition 2.1 in such form is clearly redundant for this purpose, however, the author hopes to apply it to further researches.

Corollary 2.1. *Under the conditions of the proposition, $\mathcal{L} \notin DTIME(F(D^{-\zeta} \cdot n^{\zeta}))$ holds, where $\zeta = s^{-1}$ ($s = 1$ in case (b)).*

Proof. Really, $s_1 = s + \delta$ and $aD_1 = a(D + \delta)$ tend to s and D respectively, when a tends to one and δ tends to zero. Hence, $\rho = (s + \delta)^{-1}$, n^{ρ}, and $D_1^{-\rho}$ accordingly tend to s^{-1}, $n^{s^{-1}}$, and $D^{-s^{-1}}$ in this case. □

3 Necessary agreements and the main result

In this section, we specify the restrictions on the used Turing machines, the characteristics of their actions, and the methods of recording their instructions and Boolean formulae. These agreements are very important in proving the main theorem. Although any of these restrictions can be omitted at the cost of a complication of proofs.

3.1 On the Turing machines and recording of Boolean formulae

We reserve the following alphabet for the formulae of the signature of the two-element Boolean algebra $\mathcal{B}$: a) signature symbols $\cap, \cup, C, 0, 1$ and equality sign $\approx$; b) Latin letters for the indication of the types of the object variables; c) Arabic numerals and comma for the writing of indices; d) Logical connectives $\neg, \wedge, \vee, \rightarrow$; e) the signs of quantifiers $\forall, \exists$; f) auxiliary symbols: (,). All these symbols constitute the first part of *a natural language*.

Remark 2. Let us pay attention to that we use three different symbols for the denotation of equality. The first is the "$\approx$" symbol of signature. It applies only inside the formulae of a logical theory. The second is the ordinary sign "$=$". It denotes the real or assumed equality and is used in our discussions on the formal logical system. The third sign "$\rightleftharpoons$" designates the equality in accordance with a definition.

The priority of connectives and operations or its absence is inessential, as a difference in length of formulae is linear in these cases.

Hereinafter we consider only deterministic machines with the fixed tape alphabet A, which contains at least four symbols: the first of them is a designating "blank" symbol, denoted Λ; the second is a designating "start" symbol, denoted $\rhd$; and the last two are the numerals 0,1 (almost as in Section 1.2 of [2]). As usual, the machine cannot write or erase $\rhd$ symbol.

It is implied that the simulated machines have an only tape, seeing that the transformation of the machine program from a multi-tape variant to a single-tape version is feasible in the polynomial time on the length of the program, at that the running time increases polynomially [1], [2], [9]. Although the auxiliary machines may be multi-tape.

The machine tape is infinite only to the right as in [1], [2], [4], [5], [7], [9], [12]–[21], because such machines can simulate the computations, which is T steps in length on the machine with two-sided tape, in linear time of T [2]. The tape initially contains the start symbol $\rhd$ in the leftmost square, a finite non-blank input string X alongside of the $\rhd$, and the blank symbol Λ on the rest of its cells; the head is aimed at the left end of the tape, and the machine is in the special starting state $q_{start} = q_0$. When the machine recognizes an input, it enters the accepting state $q_1 = q_{acc}$ or the rejecting state $q_2 = q_{rej}$.

Our machines have the single-operand instructions of a kind $q_i\alpha \rightarrow q_j\beta$ as in [12], which differ from double-operand instructions of a form $q_i\alpha \rightarrow q_j\beta\gamma$, where $\alpha \in A$; $\beta, \gamma \in A \cup \{R, L\}$. Even if we regard the execution of a double-operand instruction as one step of computation, then the difference in length of the running time will be linear.

The Turing machines do not fall into a situation when the machine stopped, but its answer remained undefined. Namely, they do not try to go beyond the left edge of the tape; and besides, they do not contain the *hanging* (or *pending*) internal states q_j, for which $j \neq 0, 1, 2$, and there exist instructions of a kind $\ldots \rightarrow q_j\beta$, but there are no instructions beginning with $q_j\alpha \rightarrow \ldots$ at least for one $\alpha \in A$. The attempts to go beyond the left edge of the tape are blocked by the replacement of the instructions of a form $q_i\rhd \rightarrow q_kL$ by $q_i\rhd \rightarrow q_i\rhd$. The hanging states are eliminated by adding the instructions of a kind $q_j\alpha \rightarrow q_j\alpha$ for each of the missing alphabet symbol α.

The programs of such Turing machines with the tape alphabet A are written down by the symbols of this alphabet, as well as the application of the symbols $q, R, L, \rightarrow$, Arabic numerals, and comma. This is the second, last part of *a natural language*.

We will write the subscripts by some fix number system, and it does not matter which is the base a of this system. Therefore, we will apply the record $\lceil \log t \rceil$ for denotation of the smallest integer, which is not less than $\log_a t$, when we calculate the length of the index t or the state number t.

3.2 The main theorem and its corollary

Let $c_O M$ be a chosen polynomial code of an object M by a string over a tape alphabet A — see the beginning of Subsection 2.2. We suppose that for this encoding, there exists a linear function l such that the inequalities $|M| \leqslant |c_O M| \leqslant l(|M|)$ hold for any object M of the natural language described in the previous subsection.

Theorem 3.1. *For each deterministic Turing machine P and every input string X, one can write a closed formula (sentence) $\Omega(X,P)$ of the signature of the two-element Boolean algebra $\mathcal{B}$ with the following properties:*

(i) the code $c_O\,\Omega(X,P)$ can be written within polynomial time of $|X|$ and $|P|$;

(ii) $Th(\mathcal{B}) \vdash \Omega(X,P)$ if and only if the Turing machine P accepts input X within time $\exp(|X|)$;

(iii) for every $\varepsilon > 0$, there is a constant $D > 0$ (depending on the used encoding) such that the inequalities $|X| < |c_O\,\Omega(X,P)| \leqslant D \cdot |c_O P| \cdot |X|^{2+\varepsilon}$ are valid for all sufficiently long X.

See the proof in Sections 4–6. Now we just note that according to the agreement at the beginning of this subsection, the calculation of the lengths of all components of the modeling formulae will be based on the estimate of the quantity of all the symbols, of the natural language of Subsection 3.1, involved in their recording.

At first, we will construct the very long formulae that simulate the computations. These formulae will have a huge number of the "redundant" variables. We will take care of the brief record of the constructed formulae after we ascertain the correctness of our modeling (see Propositions 5.1 (ii), 5.2, and 6.1(ii) below). The modified Stockmeyer and Meyer method is substantially used at that.

Corollary 3.1. *For every $\varepsilon > 0$, $Th(\mathcal{B}) \notin DTIME(\exp(D^{-\rho} \cdot n^{\rho}))$, where $\rho = (2+\varepsilon)^{-1}$.*

Proof. It straightforwardly follows from the theorem and Corollary 2.1. □

Corollary 3.2. *The class* **P** *is a proper subclass of the class* **PSPACE**.

Proof. Really, the theory $Th(\mathcal{B})$ does not belong to the class **P** under the previous corollary, and this theory is equivalent to the language $TQBF$ relatively polynomial reduction. But the second language belongs to the class **PSPACE**, moreover, it is polynomially complete for this class [20]. □

Remark 3. This result is quite natural and expected for a long time. Its proof is yielded by one of the few possible ways. Indeed, since the language $TQBF$ is polynomially complete for the class **PSPACE**, the inequality $\mathbf{P} \neq \mathbf{PSPACE}$ implies the impossibility of the inclusion $Th(\mathcal{B}) \in \mathbf{P}$; but this is almost equivalent to $Th(\mathcal{B}) \notin DTIME(\exp(dn^{\delta}))$ for suitable $d, \delta > 0$, as it is clear that $Th(\mathcal{B}) \in DTIME(\exp(d_1 n))$.

3.3 Supplementary denotations and arrangements

We introduce the following abbreviations and arrangements for the improvement in perception (recall that "$A \rightleftharpoons \mathcal{A}$" means "$A$ is a designation for $\mathcal{A}$"):

(1) the square brackets and (curly) braces are equally applied with the ordinary parentheses in long formulae; (2) the connective $\wedge$ is sometimes written as &; (3) $\cap$ and $\wedge$ (&) connect more closely than $\cup$ and $\vee, \rightarrow$; (4) $x < y \rightleftharpoons x \approx 0 \wedge y \approx 1$; (5) $\langle \alpha_0, \ldots, \alpha_n \rangle < \langle \beta_0, \ldots, \beta_n \rangle$ is the comparison of tuples in lexicographic ordering, i.e., it is the formula

$$\alpha_0 < \beta_0 \vee \Big\{ \alpha_0 \approx \beta_0 \wedge \big[\alpha_1 < \beta_1 \vee \big(\alpha_1 \approx \beta_1 \wedge \{ \alpha_2 < \beta_2 \vee [\alpha_2 \approx \beta_2 \wedge (\alpha_3 < \beta_3 \ldots)]\}\big)\big]\Big\}.$$

The record $\widehat{x}$ signifies an ordered set $\langle x_0, \ldots, x_n \rangle$, whose length is fixed. It is natural that the "formula" $\widehat{x} \approx \widehat{\alpha}$ denotes the system of equations $x_0 \approx \alpha_0 \wedge \ldots \wedge x_n \approx \alpha_n$. The tuples of variables with two subscripts will occur only in the form where the first of these indices is fixed, for instance, $\langle u_{k,0}, \ldots, u_{k,n} \rangle$, and we will denote it by $\widehat{u}_k$.

Counting the length of a formula of the natural language, we are guided by the rule: a tuple $\widehat{x}$ has a length of $n{+}1$ plus M, which is the quantity of symbols involved in a record of the indices $0, \ldots, n$. The inequality $|\widehat{x} \approx \widehat{\alpha}| \leqslant M{+}3n{+}3$ will hold, if the $\widehat{\alpha}$ is a tuple of constants; and $|\widehat{x} \approx \widehat{\alpha}| \leqslant 2M{+}3n{+}3$, when it consists of variables.

A binary representation of a natural number t is denoted by $(t)_2$.

It is known that if $t = (\widehat{\gamma}) \rightleftharpoons \langle \gamma_0, \ldots, \gamma_n \rangle_2$ is a binary representation of a natural number $t \leqslant \exp(2, n)$, then the numbers $t{+}1$ and $t{-}1$ will be respectively expressed as

$$((\widehat{\gamma}){+}1)_2 = \langle \gamma_0 \oplus \gamma_1 \cdot \ldots \cdot \gamma_{n-1} \cdot \gamma_n,\ \ldots,\ \gamma_{n-2} \oplus \gamma_{n-1} \cdot \gamma_n,\ \gamma_{n-1} \oplus \gamma_n,\ \gamma_n \oplus 1 \rangle_2 \quad \text{and}$$
$$((\widehat{\gamma}){-}1)_2 = \langle \gamma_0 \oplus C\gamma_1 \cdot \ldots \cdot C\gamma_{n-1} \cdot C\gamma_n,\ \ldots,\ \gamma_{n-2} \oplus C\gamma_{n-1} \cdot C\gamma_n,\ \gamma_{n-1} \oplus C\gamma_n,\ \gamma_n \oplus 1 \rangle_2,$$

where the operation $\cap$ is written in the form of multiplication $x \cap y = x \cdot y$; and $x \oplus y \rightleftharpoons x \cdot Cy \cup Cx \cdot y$.

Let us pay attention that if $(\widehat{\gamma})$ is a binary representation of a natural number t, then $((t)_2) = t$ and $((\widehat{\gamma}))_2 = (\widehat{\gamma})$ according to our dessignations.

Lemma 3.1. *(i)* $|\langle \alpha_0, \ldots, \alpha_n \rangle < \langle \beta_0, \ldots, \beta_n \rangle| = \mathcal{O}(\max\{|\langle \alpha_0, \ldots, \alpha_n \rangle|, |\langle \beta_0, \ldots, \beta_n \rangle|\})$.

(ii) If a tuple $(t)_2$ (together with the indices) is l symbols in length, then the binary representation of the numbers $t{\pm}1$ will take up $\mathcal{O}(l^2)$ symbols.

Proof. It is obtained by direct calculation. □

4 The beginning of the proof of Theorem 3.1

Prior to the writing of a formula $\Omega(X, P)$, we add $2|A|$ the *instructions of the idle run* to a program P, they have the form $q_k\alpha \rightarrow q_k\alpha$, where $k \in \{1(\mathit{accept}),\ 2(\mathit{reject})\}$, $\alpha \in A$. While the machine executes them, the tape configuration does not change.

4.1 The primary and auxiliary variables

In order to simulate the operations of a Turing machine P on an input X within the first $T = \exp(|X|)$ steps, it is enough to describe its actions on a zone, which is $T{+}1$ squares in width, since if the P starts its run in the zeroth cell, then it can finish a computation at most in the Tth one. Because the record of the number $(T)_2$ has $n{+}1 = |X|{+}1$ the bit, the cell numbers are encoded by the values of the ordered sets of the variables of a kind "x": $\widehat{x}_t = \langle x_{t,0}, \ldots, x_{t,n} \rangle$, which have a length of $n{+}1$. The first index t, i.e., *the color* of the record, denotes the step number, after which there appeared a configuration under study on the tape. So the formula $\widehat{x}_t \approx \widehat{\alpha} \rightleftharpoons x_{t,0} \approx \alpha_0 \wedge \ldots \wedge x_{t,n} \approx \alpha_n$ assigns the number $(\widehat{\alpha})$ of the required tape cell in binary notation at the instant t.

Let r be so great a number that one can write down all the state numbers of the P and encode all the symbols of the alphabet A through the bit combinations of the length $r{+}1$ at one time. Thus, $\exp(r{+}1) \geqslant |A|{+}U$, where U is the maximal number of the internal states of the P, and if $\beta \in A$, then $c\beta \rightleftharpoons \langle c\beta_0, \ldots, c\beta_r \rangle$ will be the $(r{+}1)$-tuple, which codes the β. So, the encoding c_O applied in Sections 2 and 3 is "outside" (inherent a machine being simulated), and the encoding c is "inner" (inherent a modeling formula).

The formula $\widehat{f_t} \approx c\varepsilon$ represents an entry of symbol ε in some cell after step t, where $\widehat{f_t}$ is the $(r{+}1)$-tuple of variables. When the $(\widehat{\mu})$th cell contains the symbol ε after step t, then this fact is associated with the *quasi-equation* (or *the clause*) of color t:

$$\psi_t(\widehat{\mu} \to \varepsilon) \ \rightleftharpoons\ \widehat{x}_t \approx \widehat{\mu} \to \widehat{f_t} \approx c\varepsilon \ \rightleftharpoons\ (x_{t,0} \approx \mu_0 \wedge \ldots \wedge x_{t,n} \approx \mu_n) \to \\ \to (f_{t,0} \approx c\varepsilon_0 \wedge \ldots \wedge f_{t,r} \approx c\varepsilon_r).$$

The tuples of variables $\widehat{q_t}$ and $\widehat{d_t}$ are accordingly used to indicate the number of the machine's internal state and the code of the symbol scanned by the head at the instant t. For every step t, a number $i = (\widehat{\delta})$ of the machine state q_i and a scanned square's number $(\widehat{\xi})$ together with a symbol α, which is contained there, are represented by a united *π-formula* of color t:

$$\pi_t(\alpha, (i)_2, \widehat{\xi}) \ \rightleftharpoons\ \widehat{d_t} \approx c\alpha \wedge \widehat{q_t} \approx \widehat{\delta} \wedge \widehat{z_t} \approx \widehat{\xi} \ \rightleftharpoons\ (d_{t,0} \approx c\alpha_0 \wedge \ldots \wedge d_{t,r} \approx c\alpha_r) \wedge \\ \wedge (q_{t,0} \approx \delta_0 \wedge \ldots \wedge q_{t,r} \approx \delta_r) \wedge (z_{t,0} \approx \xi_0 \wedge \ldots \wedge z_{t,n} \approx \xi_n),$$

where the ordered sets of variables $\widehat{d_t}$ and $\widehat{q_t}$ have a length of $r{+}1$; and $\widehat{z_t}$ is the $(n{+}1)$-tuple of variables and is assigned for the storage of the scanned cell's number. The formula expresses a condition for the applicability of instruction $q_i\alpha \to \ldots$; in other words, this is a *timer* that "activates" exactly this instruction, provided that the head scans the $(\widehat{\xi})$th cell.

The basic variables $\widehat{x}_t, \widehat{z}_t$, and $\widehat{q_t}, \widehat{f_t}, \widehat{d_t}$ are introduced in great abundance in order to facilitate the proof. But a final modeling formula will only contain those of them for which $t = 0, \ldots, n$ or $t = T \rightleftharpoons \exp(n)$ holds. The sets of the basic variables have different lengths. However, this will not lead to confusion, since the tuples of the first two types will always be $n{+}1$ in length, whereas the last ones will have a length of $r{+}1$. The sets of constants or other variables may also be different in length, but such tuple will always be unambiguously associated with some of the above-mentioned ones.

The other variables are auxiliary. They will be described as needed. Their task consists in a determination of the values of the basic variables of the color $t+1$ provided that the primary ones of the color t have the "correct" values; namely, this attribution has to adequately correspond to that instruction which is employed at the step $t{+}1$.

Lemma 4.1. *If the indices are left out of account, then a clause $\psi_t(\widehat{u} \to \beta)$ and a timer (π-formula) will be $\mathcal{O}(n{+}r)$ in length.*

Proof. It is obtained by direct counting. □

4.2 The description of an instruction action

The following formula $\varphi(k)$ describes an action of the kth instruction $M(k) = q_i\alpha \to q_j\beta$ (including the idle run's instructions; see the beginning of this section) at some step, where $\alpha \in A$, $\beta \in A \cup \{R, L\}$:

$$\varphi(k) \ \rightleftharpoons\ \forall \widehat{u} \left\{\pi_t(\alpha, (i)_2, \widehat{u}) \ \to\ \left[\Delta^{cop}(\widehat{u}(\beta)) \ \&\ \forall \widehat{h}\left(\Gamma^{ret}(\beta) \ \to \right.\right.\right. \\ \left.\left.\left. \to\ [\Delta^{wr}(\beta) \ \&\ \pi_{t+1}(h, (j)_2, \widehat{u}(\beta))]\right)\right]\right\}.$$

For the sake of concreteness, we regard that this step has a number $t+1$, so we have placed such subscripts on both π-formulae. Now we will describe the subformulae of the $\varphi(k)$ with the free basic variables $\widehat{x}_t, \widehat{q}_t, \widehat{z}_t, \widehat{d}_t, \widehat{f}_t, \widehat{x}_{t+1}, \widehat{q}_{t+1}, \widehat{z}_{t+1}, \widehat{d}_{t+1}$, and $\widehat{f}_{t+1}$.

The first π-formula of color t plays the role of *a timer* and "starts up the fulfillment" of the instruction with the prefix $q_i\alpha \to \dots$ provided that a head scans the $(\widehat{u})$th square. In this case, the number of the cell, on which will be aimed the head after the execution of the instruction $M(k)$, can be found for given meta-symbol $\beta \in \{R, L\} \cup A$, as follows: $\widehat{u}(R) \rightleftharpoons ((\widehat{u})+1)_2$; $\widehat{u}(L) \rightleftharpoons ((\widehat{u})-1)_2$; and $\widehat{u}(\beta) \rightleftharpoons \widehat{u}$ for $\beta \in A$.

The formula $\Delta^{cop}(\widehat{u}(\beta))$ changes the color of records in all the cells, whose numbers are different from $(\widehat{u}(\beta))$; in other words, it "copies" the majority of records:

$$\Delta^{cop}(\widehat{u}(\beta)) \rightleftharpoons \forall\, \widehat{w}\, [\neg \widehat{w} \approx \widehat{u}(\beta) \to \exists\, \widehat{g}(\psi_t(\widehat{w} \to \widehat{g}) \;\wedge\; \psi_{t+1}(\widehat{w} \to \widehat{g}))].$$

If $\beta \in \{R, L\}$, then $\Gamma^{ret}(\beta) \rightleftharpoons \psi_t(\widehat{u}(\beta) \to \widehat{h}) = \widehat{x}_t \approx \widehat{u}(\beta) \to \widehat{f}_t \approx \widehat{h}$. An informal sense of this formula is the following: it "seeks" a code $\widehat{h}$ of the symbol, which will be scanned after the next step $t+1$ (by this reason it is named "retrieval"); for this purpose, it "inspects" the square that is to the right or left of the cell $(\widehat{u})$. When $\beta \in A$, there is no need to look for anything, so the formula $\Gamma^{ret}(\beta)$ will be very simple in this case: $\widehat{h} \approx c\beta$.

The formula $\Delta^{wr}(\beta)$ "puts" the symbol, whose code is $\widehat{h}$ and color is $t+1$, in the $(\widehat{u}(\beta))$th square: $\Delta^{wr}(\beta) \rightleftharpoons \psi_{t+1}(\widehat{u}(\beta) \to \widehat{h}))$.

The second π-formula of the color $t+1$ "aims" the head at the $(\widehat{u}(\beta))$th cell; "places" the symbol $\widehat{h}$ in this location; and "changes" the number of the machine state for j: $\widehat{z}_{t+1} \approx \widehat{u}(\beta) \wedge \widehat{d}_{t+1} \approx \widehat{h} \wedge \widehat{q}_{t+1} \approx (j)_2$.

Lemma 4.2. *(i) If* $\beta \in A$, *then the formulae* $\Gamma^{ret}(\beta)$; $\pi_{t+1}(\widehat{h}, (j)_2, \widehat{u}(\beta))$; $\Delta^{wr}(k, \beta)$; $\Delta^{cop}(\widehat{u}, \beta)$; *and* $\varphi(k)$ *will be* $\mathcal{O}(|\psi_{t+1}(\widehat{w} \to \widehat{g})|)$ *in length.*

(ii) For $\beta \in \{R, L\}$, *each of these formulae is* $\mathcal{O}(n \cdot |\psi_{t+1}(\widehat{w} \to \widehat{g})|)$ *in length.*

Proof. This follows from Lemmas 3.1 and 4.1 by immediate calculation. □

4.3 The description of the running steps and configurations

At first, we will construct a formula $\Phi^{(0)}(P)$ describing one step of the machine run, when the P is applied to a configuration that arose after some step t. Next, we will describe the machine actions over an exponential period by the like manner; at that, the Stockmeyer and Meyer method will be used.

Let N be a quantity of the instructions of machine P together with $2|A|$ the idle run's ones (see the beginning of this section). The formula $\Phi^{(0)}(P)$ that describes one step (whose number is $t+1$) of the machine P is of the form:

$$\Phi^{(0)}(P)(\widehat{y}_t, \widehat{y}_{t+1}) \;\rightleftharpoons\; \bigwedge_{0<k\leq N} \varphi(k)(\widehat{y}_t, \widehat{y}_{t+1}),$$

where $\widehat{y}_t \rightleftharpoons \langle \widehat{x}_t, \widehat{q}_t, \widehat{z}_t, \widehat{d}_t, \widehat{f}_t \rangle$ and $\widehat{y}_{t+1} \rightleftharpoons \langle \widehat{x}_{t+1}, \widehat{q}_{t+1}, \widehat{z}_{t+1}, \widehat{d}_{t+1}, \widehat{f}_{t+1} \rangle$ are two $(2n+3r+5)$-tuples of its free variables.

Let us denote the quantifier-free part of a formula χ as $\langle \chi \rangle$.

Lemma 4.3. *(i) If* $\widehat{x}_t \neq \widehat{\mu}$, *then a clause* $\psi_t(\widehat{\mu} \to \varepsilon)$ *will be true independently of the value of variables* $\widehat{f}_t$. *In particular, a quasi-equation, which is contained into the record of* $\langle \Delta^{cop}(\widehat{u}) \rangle$, *will be true, if its color is* t *or* $t+1$, *and at the same time* $\widehat{x}_t \neq \widehat{w}$ *or* $\widehat{x}_{t+1} \neq \widehat{w}$, *respectively.*

(ii) For some constant D_1, *the inequality* $|\Phi^{(0)}(P)(\widehat{y}_t, \widehat{y}_{t+1})| \leqslant D_1 \cdot |c_O P| \cdot |\varphi(N)|$ *holds provided that the program* P *is not empty.*

Proof. (i) The premises of clauses are false in these cases.
(ii) If $N-2|A|\neq 0$, then $N\cdot\lceil\log N\rceil < D_2\cdot|c_O P|$. This implies the lemma assertion. □

The formulae $\Phi^{(s+1)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s+1)})$ conform to the actions of machine P over a period of time $e(s)\rightleftharpoons\exp(s)$. They are defined by induction on s:

$$\Phi^{(s+1)}(P)\rightleftharpoons\ \exists\widehat{v}\forall\widehat{a}\forall\widehat{b}\,\big\{\big[(\widehat{y}_t\approx\widehat{a}\wedge\widehat{v}\approx\widehat{b})\ \vee\ (\widehat{v}\approx\widehat{a}\wedge\widehat{b}\approx\widehat{y}_{t+e(s+1)})\big]\ \to\Phi^{(s)}(P)(\widehat{a},\widehat{b})\big\},$$

where $\widehat{v},\widehat{a},\widehat{b}$ are the $(2n+3r+5)$-tuples of the new auxiliary variables.

Let $L(t)$ be a configuration, which is recorded on the tape after step t (it may be unrealizable): namely, every cell, whose number is $(\widehat{\mu})$, contains a symbol $\varepsilon(\widehat{\mu})$; the scanned square has the number $(\widehat{\eta})$; and a machine is ready to execute an instruction $q_i\alpha\to\ldots$. Then the following formula corresponds to this configuration:

$$\Psi L(t)(\widehat{y}_t)\ \rightleftharpoons\ \pi_t(\alpha,(i)_2,\widehat{\eta})\ \&\bigwedge_{0\leqslant(\widehat{\mu})_2\leqslant T}\psi_t(\widehat{\mu}\to\varepsilon(\widehat{\mu})),$$

we recall that $T=\exp(n)$. It has $2n+3r+5$ free variables $\widehat{y}_t=\langle\widehat{x}_t,\widehat{q}_t,\widehat{z}_t,d_t,f_t\rangle$.

5 The simulation of one running step

We simply associated the formulae, which were constructed earlier, with certain components of programs or with processes. However one cannot assert that these formulae simulate something, i.e., they will not always turn true, when the events, which are described by them, are real.

5.1 Simulating formula

Let $K(t)$ and $K(t+1)$ be some adjacent configurations, we definite

$$\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})\rightleftharpoons[\Psi K(t)(\widehat{y}_t)\ \&\ \Phi^{(0)}(P)(\widehat{y}_t,\widehat{y}_{t+1})]\to\Psi K(t+1)(\widehat{y}_{t+1}).$$

We will prove in this section that the sentence $\forall\widehat{y}_t\forall\widehat{y}_{t+1}\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ is true on the Boolean algebra $\mathcal{B}$ if and only if the machine P transforms the configuration $K(t)$ into $K(t+1)$ in one step, i.e., this formula models the machine actions at the step $t+1$.

5.2 The single-valuedness of modeling

Let $K(t+1)$ be a configuration that has arisen from a configuration $K(t)$ as a result of the machine P action at the step $t+1$.

Proposition 5.1. *(i) There exist special values of variables $\widehat{y}_t$ such that the formula $\Psi K(t)(\widehat{y}_t)$ is true, and the truth of $\Phi^{(0)}(P)(\widehat{y}_t,\widehat{y}_{t+1})$ follows from the truth of $\Psi K(t+1)(\widehat{y}_{t+1})$ for every $\widehat{y}_{t+1}$.*

(ii) If a formula $\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ is identically true over algebra $\mathcal{B}$, then the machine P cannot convert the configuration $K(t)$ into the configuration, which differs from $K(t+1)$, at the step $t+1$.

Proof. We will prove these assertions simultaneously. Namely, we will select the values of the variables $\widehat{y}_t$ and $\widehat{y}_{t+1}$ such that a formula

$$\Upsilon_{t+1}(\widehat{y}_t,\widehat{y}_{t+1})\rightleftharpoons[\Psi K(t)(\widehat{y}_t)\,\&\,\Phi^{(0)}(P)(\widehat{y}_t,\widehat{y}_{t+1})]\to\Psi L(t+1)(\widehat{y}_{t+1})$$

will be false, if the configuration $L(t+1)$ differs from the real $K(t+1)$. This implies Item (ii) of the proposition. However, at the beginning, we will select the special values of the variables of the tuple $\widehat{y}_t$. After that when we pick out the values of the corresponding variables of the color $t+1$, the formulae $\Psi K(t+1)(\widehat{y}_{t+1})$ and $\Phi^{(0)}(P)(\widehat{y}_t, \widehat{y}_{t+1})$ will become true or false at the same time depending on the values of the variables $\widehat{y}_{t+1}$.

Let $M(k) = q_i\alpha \to \ldots$ be an instruction that is applicable to the configuration $K(t)$; and $(\widehat{\eta})$ be a number of the scanned square. We define $\widehat{d}_t = c\alpha$, $\widehat{q}_t = (i)_2$, $\widehat{z}_t = \widehat{\eta}$. Then π-formula $\pi_t(\alpha, (i)_2, \widehat{\eta})$, which is in the record of $\Psi K(t)$, become true.

Now we consider a formula $\varphi(l)$ that conforms to some instruction $M(l) = q_b\theta \to \ldots$ that differs from the $M(k)$. This formula has a timer $\pi_t(\theta, (b)_2, \widehat{u})$ as the first premise. For the selected values of the variables $\widehat{d}_t$; $\widehat{q}_t$; and $\widehat{z}_t$, the timer takes the form of $c\alpha \approx c\theta \wedge (i)_2 \approx (b)_2 \wedge \widehat{\eta} \approx \widehat{u}$. It is obvious that if $\alpha \neq \theta$; or $i \neq b$; or $\widehat{u} \neq \widehat{\eta}$, then this π-formula will be false, and the whole $\varphi(l)$ will be true.

Thus, let $\varphi(k)$ be a formula that correspondents to the instruction $M(k) = q_i\alpha \to q_j\beta$; and $\widehat{u} = \widehat{\eta}$. Let us define $\widehat{d}_{t+1} = c\lambda$; $\widehat{q}_{t+1} = (j)_2$; $\widehat{z}_{t+1} = \widehat{\eta}(\beta)$, where $(\widehat{\eta}(\beta))$ is a number of the square, which will be scanned by the machine head after the fulfillment of the instruction $M(k)$; and λ is the symbol, which the head will see there. For these $\widehat{u}$ and selected values of $\widehat{d}_{t+1}, \widehat{q}_{t+1}, \widehat{z}_{t+1}$, the π-formula, which enters into the record of $\Psi K(t+1)$, becomes true. But the conclusion of the quantifier-free part $\langle\varphi(k)\rangle$ of the formula $\varphi(k)$ contains a slightly different timer $\pi_{t+1}(\widehat{h}, (j)_2, \widehat{u}(\beta))$; in this timer, the only equality $\widehat{d}_{t+1} \approx \widehat{h}$ included in it raises doubts for the time being.

Let us assign $\widehat{x}_t = \widehat{\eta}(\beta)$. Since $\widehat{u} = \widehat{\eta}$, the equality $\widehat{u}(\beta) = \widehat{\eta}(\beta)$ holds too. Therefore the quasi-equation, of the color t, which enters into the $\langle\Delta^{cop}(\widehat{u}(\beta))\rangle$, is true for all $\widehat{w} \neq \widehat{\eta}(\beta)$ and irrespective of the values of the tuples $\widehat{f}_t$ and $\widehat{g}$ according to Lemma 4.3(i). For the same reason, all the clauses that are included in the $\Psi K(t)$ are true, except the clause $\psi_t(\widehat{\eta}(\beta) \to \lambda)$ for $\beta \in \{R, L\}$ or $\psi_t(\widehat{\eta} \to \alpha)$ for $\beta \in A$. We set the value of the tuple $\widehat{f}_t$ as $c\lambda$, if $\beta \in \{R, L\}$, or as $c\alpha$, if not. Now, the questionable clause from the $\Psi K(t)$ becomes true, because its premise and conclusion are true.

If $\widehat{h} \neq c\lambda$, then the formula $\Gamma^{ret}(\beta)$ will be false, since it is either $\psi_t(\widehat{\eta}(\beta) \to \widehat{h})$ for $\beta \in \{R, L\}$, or $\widehat{h} \approx c\beta$ for $\beta \in A$. Hence the whole formula $\langle\varphi(k)\rangle$ will be true in this case. When $\widehat{h} = c\lambda$, the terminal π-formula in the $\langle\varphi(k)\rangle$ becomes true, since $\widehat{d}_{t+1} = c\lambda$.

If the "incorrect" formula $\Psi L(t+1)$ has a mistake in the record of timer or clause $\psi_{t+1}(\widehat{\eta}(\beta) \to \lambda)$, we will define $\widehat{x}_{t+1} = \widehat{\eta}(\beta)$ and $\widehat{f}_{t+1} = c\lambda$ (we recall that $\lambda = \beta$ for $\beta \in A$). When these fragments are that as they should be, but there is another "incorrect" clause $\psi_{t+1}(\widehat{\mu} \to \rho)$, where ρ is different from "real" δ, we will assign $\widehat{x}_{t+1} = \widehat{\mu}$ and $\widehat{f}_{t+1} = \widehat{g} = c\delta$ (we note that this is the only case when we need to set the values of the variables $\widehat{g}$). After that the quasi-equations of the color $t+1$ in the formulae $\langle\Delta^{cop}(\widehat{u}(\beta))\rangle$ and $\Delta^{wr}(\beta)$ are true in both of these cases on the grounds of Lemma 4.3(i) or because their premises and conclusions are true. Therefore the whole formula $\langle\varphi(k)\rangle$ is true. All the clauses contained in $\Psi K(t+1)$ are true for the same reasons.

We obtain as a result that any formula $\varphi(l)$ is true for the above selected values of the primary variables, so the entire conjunction $\Phi^{(0)}(P)$ is true. Since the premise and conclusion of the $\Omega^{(0)}(X, P)(\widehat{y}_t, \widehat{y}_{t+1})$ are true, and the configurations $K(t+1)$ and $L(t+1)$ are different; the "incorrect" formula Υ_{t+1} is false, because of the choice of the values of basic variables.

Since the configuration $L(t+1)$ may differ from the real $K(t+1)$ in any place, Item (i) is established too. □

5.3 The sufficiency of modeling

We will now prove a converse to Proposition 5.1(ii).

Proposition 5.2. *Let $K(t+1)$ be a configuration that has arisen from a configuration $K(t)$ as a result of an action of the machine P at the step $t+1$. Then the formula $\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ is identically true on algebra $\mathcal{B}$.*

Proof. Let $M(k)=q_i\alpha\to q_j\beta$ be the instruction that transforms the configuration $K(t)$ into the $K(t+1)$; and $\varphi(k)(\widehat{y}_t,\widehat{y}_{t+1})$ be a formula, which is written for this instruction. This formula is the consequence of the $\Phi^{(0)}(P)(\widehat{y}_t,\widehat{y}_{t+1})$.

Let us replace the $\varphi(k)$ by a conjunction of formulae $\varphi(k)(\widehat{\mu})$, they are each obtained as the result of the substitution the various values of the universal variables $\widehat{u}$ for the variables themselves. Every formula $\varphi(k)(\widehat{\mu})$ contains the premise $\widehat{d}_t\approx c\alpha\wedge\widehat{q}_t\approx(i)_2\wedge\widehat{z}_t\approx\widehat{\mu}$, one of them coincides with the only timer $\pi_t(\gamma,\widehat{\delta},\widehat{\eta})$ included in the $\Psi K(t)$ for $\widehat{u}=\widehat{\mu}=\widehat{\eta}$, $\gamma=\alpha$, and $i=(\widehat{\delta})$, as the instruction $M(k)$ is applicable to the configuration $K(t)$. Therefore the formula $\Psi K(t)$ & $\Delta^{cop}(\widehat{\eta}(\beta))$ & $\forall\widehat{h}\{\Gamma^{ret}(\beta)(\widehat{\eta})\to[\Delta^{wr}(\beta)(\widehat{\eta})$ & $\pi_{t+1}((\widehat{h},(j)_2,\widehat{\eta}(\beta))]\}$ follows from the $\Psi K(t)$ and $\varphi(k)(\widehat{\eta})$.

The formula $\Delta^{cop}(\widehat{\eta}(\beta))$ begins with the quantifiers $\forall\widehat{w}$. Let us replace this formula by a conjunction that is equivalent to it, we substitute all possible values for the variables $\widehat{w}$ to this effect. For every value of $\widehat{w}$, there is a unique value of the tuple $\widehat{g}$ such that the clause $\psi_t(\widehat{w}\to\widehat{g})$ enters into the formula $\Psi K(t)$. When these values of $\widehat{g}$ are substituted in their places, we will obtain all the quasi-equations from the $\Psi K(t+1)$, except one.

For the appropriate value of $\widehat{h}$, either the formula $\Gamma^{ret}(\beta)(\widehat{\eta})$ coincides with some clause existing in the $\Psi K(t)$, or it becomes true: $\widehat{h}\approx c\beta$, owing to the applicability of the instruction $M(k)$ to the configuration $K(t)$. In any case, the formula $\Delta^{wr}(\beta)(\widehat{\eta})$ in an explicit form contains the quasi-equation $\psi_{t+1}(\widehat{\eta}(\beta)\to\ldots)$, which is missing in the $\Psi K(t+1)$ so far; and the tuple $\widehat{h}$ obtains the concrete value. If we substitute this value in the concluding π-formula of the $\varphi(k)$, then we will obtain the necessary timer $\pi_{t+1}(\widehat{h},(j)_2,\widehat{\eta}(\beta))$ from the $\Psi K(t+1)$. □

6 The construction of the formula $\Omega(X,P)$

6.1 The simulation of the exponential computations

We define the formulae that model $e(s)\rightleftharpoons\exp(s)$ the running steps of a machine P, when the machine applies to a configuration $K(t)$:

$$\Omega^{(s)}(X,P)(\widehat{y}_t,\widehat{y}_{t+e(s)})\ \rightleftharpoons\ [\Psi K(t)(\widehat{y}_t)\ \&\ \Phi^{(s)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s)})]\to\Psi K(t+e(s))(\widehat{y}_{t+e(s)}).$$

Proposition 6.1. *Let $t,s\geqslant 0$ be the integers such that $t+e(s)\leqslant T$.*

(i) If the machine P transforms the configuration $K(t)$ into the $K(t+e(s))$ within $e(s)$ steps, then there are special values of variables $\widehat{y}_t$ such that the formula $\Psi K(t)(\widehat{y}_t)$ is true; and for all $\widehat{y}_{t+e(s)}$, whenever the $\Psi K(t+e(s))(\widehat{y}_{t+e(s)})$ is true, the $\Phi^{(s)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s)})$ is also true.

(ii) The formula $\Omega^{(s)}(X,P)(\widehat{y}_t,\widehat{y}_{t+e(s)})$ is identically true over the Boolean algebra $\mathcal{B}$ if and only if the machine P converts the configuration $K(t)$ into $K(t+e(s))$ within $e(s)$ steps.

Proof. Induction on the parameter s. For $s=0$, Item (i) is a consequence of Proposition 5.1(i), and Item (ii) follows from Propositions 5.1(ii) and 5.2.

We start the proof of the inductive step by rewriting the formula $\Phi^{(s+1)}(P)(\widehat{y}_t,\ \widehat{y}_{t+e(s+1)})$ in the equivalent, but longer form:

$$\exists\widehat{v}\{\forall\widehat{a}\forall\widehat{b}[(\widehat{y}_t\approx\widehat{a}\ \wedge\ \widehat{v}\approx\widehat{b})\ \to\ \Phi^{(s)}(P)(\widehat{a},\widehat{b})]\ \&\ \forall\widehat{a}\forall\widehat{b}[(\widehat{v}\approx\widehat{a}\ \wedge\ \widehat{b}\approx\widehat{y}_{t+e(s+1)})\ \to$$
$$\to\ \Phi^{(s)}(P)(\widehat{a},\widehat{b})]\}.$$

The following formula results from this immediately:

$$\Xi_{s+1} \rightleftharpoons \exists\widehat{v}\big\{\Phi^{(s)}(P)(\widehat{y}_t,\widehat{v})\ \&\Phi^{(s)}(P)(\widehat{v},\widehat{y}_{t+e(s+1)})\big\}.$$

On the other hand, each of the two implications which are included in the equivalent long form of the formula $\Phi^{(s+1)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s+1)})$ can be false only when the equalities existing in its premise are valid. Hence this formula is equivalent to the Ξ_{s+1}.

Let the machine P transforms the configuration $K(t)$ into the $K(t+e(s))$ within $e(s)$ steps, and it converts the latter into the $K(t+e(s+1))$ within the same time.

By the inductive hypothesis of Item (ii) (we recall that the induction is carried out over a single parameter s), the formula $\Omega^{(s)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s)})$ is identically true for any t such that $t+e(s)\leqslant T$, and hence it is identically true for an arbitrarily chosen t and for $t_1=t+e(s)$ provided that $t+e(s+1)=t_1+e(s)\leqslant T$. Thus, the formulae

$$[\Psi K(t)(\widehat{y}_t)\ \&\ \Phi^{(s)}(P)(\widehat{y}_t,\widehat{y}_{t+e(s)})]\rightarrow\Psi K(t+e(s))(\widehat{y}_{t+e(s)})\qquad\text{and}$$
$$\{\Psi K(t+e(s))(\widehat{y}_{t+e(s)})\ \&\ \Phi^{(s)}(P)(\widehat{y}_{t+e(s)},\widehat{y}_{t+e(s+1)})\}\rightarrow\Psi K(t+e(s+1))(\widehat{y}_{t+e(s+1)})$$

are identically true. Therefore, when we change the variables under the sign of the quantifier, we obtain from this that the following formula

$$\forall\widehat{v}\{[\Psi K(t)(\widehat{y}_t)\ \&\ \Phi^{(s)}(P)(\widehat{y}_t,\widehat{v})\ \&\ \Phi^{(s)}(P)(\widehat{v},\widehat{y}_{t+e(s+1)})]\rightarrow\Psi K(t+e(s+1))(\widehat{y}_{t+e(s+1)})\},$$

is identically true as well. This formula is equivalent to $[(\Psi K(t)\ \&\ \Xi_{s+1})\rightarrow\Psi K(t+e(s+1))](\widehat{y}_t,\widehat{y}_{t+e(s+1)})$, because the universal quantifiers will be interchanged with the quantifiers of existence, when they are introduced into the premise of the implication. Since the premise of the formula $\Omega^{(s+1)}(X,P)(\widehat{y}_t,\widehat{y}_{t+e(s+1)})$ is equivalent to $\Psi K(t)\ \&\ \Xi_{s+1}$ in accordance with the foregoing argument, the inductive step of Item (ii) is proven in one direction.

Now let the configurations $L(t+e(s+1))$ and $K(t+e(s+1))$ be different. For some $\widehat{v}_1,\widehat{y}_{t+e(s+1)}$, the formula $\{[\Psi K(t+e(s))\ \&\ \Phi^{(s)}(P)]\rightarrow\Psi K(t+e(s+1))\}(\widehat{v}_1,\widehat{y}_{t+e(s+1)})$ is true, but $\{[\Psi K(t+e(s))\ \&\ \Phi^{(s)}(P)]\rightarrow\Psi L(t+e(s+1))\}(\widehat{v}_1,\widehat{y}_{t+e(s+1)})$ is false by the inductive assumption of Item (ii). Therefore, the conclusion of the second formula is false, and its premise is true, i.e., the $\Psi L(t+e(s+1))(\widehat{y}_{t+e(s+1)})$ is false, and the $\Phi^{(s)}(P)(\widehat{v}_1,\widehat{y}_{t+e(s+1)})$ and the $\Psi K(t+e(s))(\widehat{v}_1)$ are true. Since the last formula and the $\Psi K(t)(\widehat{v}_0)$ are true for some special $\widehat{v}_0$, which exists due to induction proposition of Item (i), the formula $\Phi^{(s)}(P)(\widehat{v}_0,\widehat{v}_1)$ is true. Thus, the implication $\{[\Psi K(t)\ \&\ \Phi^{(s+1)}(P)]\rightarrow\Psi L(t+e(s+1))\}(\widehat{v}_0,\ \widehat{y}_{t+e(s+1)})$ has a true premise, and a false conclusion, therefore it is not identically true. Item (ii) is proven.

Inasmuch as the configuration $L(t+e(s+1))$ may differ from the current one at any position, to finish the proof of Item (i) we set the values of the variables $\widehat{v}_1$ in a special manner, using the inductive hypothesis. □

6.2 The short recording of the initial configuration and the condition of the successful termination of the run

Since we have the instructions for the machine run in the idle mode (see the beginning of Section 4), the statement that the machine P accepts an input string X within $T=\exp(n)$ steps can be written rather briefly — by means of one quantifier-free formula of the color T: $\chi(\omega)\rightleftharpoons\widehat{q}_T\approx(1)_2$. This formula has a length of $4r+3$ symbols nonmetering the indices. The writing of the first index T occupies $\lfloor\log T\rfloor+1$ digits in some number system (see the end of Subsection 3.1), where $\lfloor y\rfloor$ is the integer part of a number y. The maximum length of the second indices is $\lfloor\log r\rfloor+1$, and so we have $|\chi(\omega)|<(4r+3)\cdot(\lfloor\log T\rfloor+\lfloor\log r\rfloor+2)$.

The formula $\Psi K(t)$ was introduced in Subsection 4.3 to describe a configuration arising after the step t. It is very lengthy: $|\Psi K(t)| > n \cdot \exp(n)$. However, in the initial configuration, the input string X occupies $|X|$ squares to the right of the edge of a tape; and the remaining part of the tape is empty, starting with the cell, whose number is $|X|+1=(\widehat{\gamma})+1$. Therefore one can describe the initial tape configuration $K(0)$ by a brief universal formula:

$$\chi(0) \rightleftharpoons \pi_0(\rhd, \widehat{0}, \widehat{0}) \ \& \bigwedge_{0\leqslant(\widehat{\eta})\leqslant|X|} \psi_0(\widehat{\eta}\to\alpha(\eta)) \ \& \ \forall\widehat{u}_0[\widehat{u}_0>\widehat{\gamma} \to \psi_0(\widehat{u}_0\to\Lambda)],$$

where Λ denotes blank symbol; $\rhd$ is a sign of the left end of the tape; and $\alpha(\eta)$ is a symbol, which is located in the number $(\widehat{\eta})$ cell; and the π-formula of the color 0 signifies that a mechanism is ready for the execution of the instruction $q_0\rhd\to\ldots$ at the zeroth instant, and the machine head scans the extreme left square.

Lemma 6.1. *(i) The formulae $\chi(0)$ and $\Psi K(0)$ are equivalent to each other.*

(ii) $|\chi(0)(\widehat{y}_0)| \leq D_2 \cdot |X| \cdot |\psi_0(\widehat{u}_0\to\Lambda)|$ for a proper constant D_2.

Proof. (i) The quantifier-free part of the formula $\chi(0)$ simply coincides with the initial fragment of the formula $\Psi K(0)$. If we replace the second part of formula $\chi(0)$, which begins with the quantifiers $\forall\widehat{u}_0$, with its equally matched conjunction, the rest of the clauses from $\Psi K(0)$ will appear.

(ii) According to Lemmas 3.1(i) and 4.1, the system of inequalities $\widehat{u}_0 > \widehat{\gamma}$ has a length of the same order as $|\psi_0(\widehat{u}_0\to\Lambda)|$, a quantifier prefix is a bit shorter. Hence $|\forall\widehat{u}_0[\widehat{u}_0>\widehat{\gamma} \to \psi_0(\widehat{u}_0\to\Lambda)]| = \mathcal{O}(|\psi_0(\widehat{u}_0\to\Lambda)|)$. Since the expression $\chi(0)(\widehat{y}_0)$ includes $|X|+1$ quasi-equations of a form $\psi_0(\widehat{\eta}\to\alpha(\eta))$ and the timer, which have a length of the same order as $|\psi_0(\widehat{u}_0 \to \Lambda)|$ by Lemma 4.1, the whole formula $\chi(0)(\widehat{y}_0)$ is not more than $D_2 \cdot |X| \cdot |\psi_0(\widehat{u}_0\to\Lambda)|$ in length for some constant D_2. □

6.3 Simulating formula $\Omega(X, P)$

Let us define

$$\begin{aligned} \Omega(X,P) \ \rightleftharpoons \ \forall\widehat{y}_0,\widehat{y}_T \Big\{ \Big[\ \chi(0)(\widehat{y}_0) \ \ \& \ \exists\widehat{v}_n\forall\widehat{a}_n\forall\widehat{b}_n\ldots\exists\widehat{v}_1\forall\widehat{a}_1\forall\widehat{b}_1 \\ \Big\{ \bigwedge_{1\leqslant s\leqslant n} [(\widehat{a}_{s+1}\approx\widehat{a}_s \wedge \widehat{v}_s\approx\widehat{b}_s) \ \vee (\widehat{v}_s\approx\widehat{a}_s \wedge \widehat{b}_s\approx\widehat{b}_{s+1})] \to \ \Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1)\Big\}\Big] \ \to \\ \to \ \chi(\omega)(\widehat{y}_T)\Big\}, \end{aligned} \tag{6.1}$$

here we designate $\widehat{a}_{n+1}=\widehat{y}_0$, $\widehat{b}_{n+1}=\widehat{y}_T$ in the record of the "big" conjunction for the sake of brevity.

Proposition 6.2. *The formula $\Omega(X,P)$ has the property (ii) from the statement of Theorem 3.1. In other words, this sentence is true on the Boolean algebra $\mathcal{B}$ if and only if the machine P accepts the input X within T steps.*

Proof. Let $\Theta_s=\Theta_s(\widehat{a}_s,\widehat{a}_{s+1},\widehat{b}_s,\widehat{b}_{s+1})$ be the denotation for a disjunction of equalities $(\widehat{a}_{s+1}\approx\widehat{a}_s\wedge\widehat{v}_s\approx\widehat{b}_s)\vee(\widehat{v}_s\approx\widehat{a}_s\wedge\widehat{b}_s\approx\widehat{b}_{s+1})$. If we carry the quantifiers through the subformulae, which do not contain the corresponding variables (according to the agreement of Subsection 3.3 that a conjunction connects more intimately than an implication), then we will obtain that the part of the formula $\Omega(X,P)$, which is located in the big square brackets in (1), is equivalent to each of the three following formulae:

$$1)\ \chi(0)(\widehat{y}_0)\ \&\ \exists\widehat{v}_n\forall\widehat{a}_n\forall\widehat{b}_n\ldots\exists\widehat{v}_1\forall\widehat{a}_1\forall\widehat{b}_1\{\bigwedge_{1\leq s\leq n}\Theta_s\to\Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1)\};$$

$$2)\ \Psi K(0)(\widehat{y}_0)\ \&\ \exists\widehat{v}_n\forall\widehat{a}_n\forall\widehat{b}_n\ldots\exists\widehat{v}_1\forall\widehat{a}_1\forall\widehat{b}_1\big(\Theta_n\to(\Theta_{n-1}\to(\ldots\to \\ \to(\Theta_1\to\Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1))\ldots))\big);$$

$$3)\ \Psi K(0)(\widehat{y}_0)\ \&\ \exists\widehat{v}_n\forall\widehat{a}_n\forall\widehat{b}_n\big(\Theta_n\to\exists\widehat{v}_{n-1}\forall\widehat{a}_{n-1}\forall\widehat{b}_{n-1}(\Theta_{n-1}\to(\ldots\to \\ \to\exists\widehat{v}_1\forall\widehat{a}_1\forall\widehat{b}_1(\Theta_1\to\Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1))\ldots))\big).$$

According to the definition, the formula $\exists\widehat{v}_s\forall\widehat{a}_s\forall\widehat{b}_s(\Theta_s\to\Phi^{(s-1)}(P)(\widehat{a}_s,\widehat{b}_s))$ contracts into the $\Phi^{(s)}(P)(\widehat{a}_{s+1},\widehat{b}_{s+1})$. Therefore the whole $\Omega(X,P)$ is equivalent to the $\forall\widehat{y}_0,\widehat{y}_T\big[\big(\Psi K(0)\ \&\ \Phi^{(n)}(P)\big)\to\chi(\omega)\big]$. Consequently, based on Proposition 6.1(ii) and Lemma 6.1(i), one could say that the formula (6.1) is the modeling formula. □

6.4 The time of writing of $\Omega(X,P)$

The simulating formula $\Omega(X,P)$ is described by the definition 6.1 in an explicit form, this notation allows us to design an algorithm for its construction. It remains only to prove the properties (i) and (iii) of the statement of Theorem 3.1. Before we substantiate the polynomiality of the algorithm, we will make sure that the formula $\Omega(X,P)$ of form (6.1) has a polynomial length. We recall that the length of a formula is calculated in the natural language — see Subsections 3.1 and 3.2.

Lemma 6.2. *There exists a constant $D>0$ such that it does not depend on the P and n and the inequalities $|X|\leqslant|\Omega(X,P)|\leqslant D\cdot|P|\cdot|X|^{2+\varepsilon}$ hold for all the long enough X and any preassigned $\varepsilon>0$.*

Proof. Many components of the modeling formula were estimated already during their description, but their lengths were estimated on the assumption that their subformulae are written with basic variables $\langle\widehat{x}_t,\widehat{q}_t,\widehat{z}_t,\widehat{d}_t,\widehat{f}_t\rangle$, which were denoted in Subsection 4.3 as $\widehat{y}_t$. However, they are not included in the composition of the subformulae of $\Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1)$ — we have written the variables from the tuples $\widehat{a}_1$ and $\widehat{b}_1$ instead theirs. Namely, the first $n{+}1$ variables in the tuple $\widehat{a}_1$ serve as $\widehat{x}_t$; the second $r{+}1$ ones in $\widehat{a}_1$ are put instead of $\widehat{q}_t$ and so on. The same is true for the tuple $\widehat{b}_1$ and basic variables of color $t{+}1$. Certainly, this replacement does not influence on the length of those formulae, where the variables are located, if one disregards a length of indices.

However, the length of the indices has markedly changed. Just because of this reason, they were earlier taken into account only implicitly or not counted at all for the estimation of the lengths of the formulae(see Lemmas 3.1, 4.2, and 4.1). Nevertheless, this changing is negligible. Really, the second indices of variables $\widehat{y}_t$ are bounded above by $E_0\rightleftharpoons\lfloor\max\{\log n,\log r\}\rfloor+1$; whereas such subscripts of the variables $\widehat{a}_s$ have their lengths restricted from above by $E_1\rightleftharpoons\lfloor\log(2n+3r+5)\rfloor+1$. But the inequalities $n=|X|>r$ and $E_0,E_1\leqslant\lfloor\log 5n\rfloor$ hold for the long enough X and fixed P. The first indices of variables of the form $\widehat{a}_s$ and $\widehat{b}_s$ are not more than $\lfloor\lg n\rfloor+1$ in length. Therefore by Lemmas 4.1 and 4.2, the quasi-equations and timers of the subformulae $\chi(0)(\widehat{y}_0)$ and $\Phi^{(0)}(P)(\widehat{a}_1,\widehat{b}_1)$ from (6.1), in which the tuples $\widehat{u}(\beta)$ are not included for $\beta=R,L$, are not greater than $D_3\cdot n\cdot\lfloor\log n\rfloor$ in length; the clauses and timers comprising the $\widehat{u}(\beta)$ have a length not more than $D_4\cdot(n\cdot\lfloor\log n\rfloor)^2$ for the suitable constants D_3 and D_4. By Lemma 4.2 we have $|\varphi(k)|\leqslant D_5\cdot(n\cdot\lfloor\log n\rfloor)^2$ for another constant D_5 and the long enough X.

The system of equalities, which are under the "big" conjunction in (6.1), is $\mathcal{O}(n\cdot[n\cdot(\lfloor\lg n\rfloor+2)])$ in length; and the quantifier prefix, which is situated before this conjunction, has approximately the same length. It is easy to see that an inequality $(\lfloor \lg n \rfloor + 2)^2 \leqslant n^{\varepsilon}$ holds for all $\varepsilon > 0$ and the big enough n. It follows from this and Lemma 4.3(ii) that $|\Omega(X,P)| \leqslant D \cdot |P| \cdot |X|^{2+\varepsilon}$ for some constant D. □

Corollary 6.1. *There exists a polynomial g such that for all X and P the construction time of the sentence $\Omega(X,P)$ is not greater than $g(|X|+|P|)$.*

Proof. We will at first estimate the time needed for a multi-tape Turing machine P_1 to write down the formula $\Omega(X,P)$. The running alphabet of this machine contains all symbols of natural language (see Subsection 3.1).

Let the input tape of the machine comprises a string X and a program P, and $|A|$ be a quantity of different symbols in the record of the X and the P. The machine P_1 can determine a length n of the input X, a maximal number U of internal states in P, and a size of the $|A|$ during one passage along its input tape. The calculation of the values of $r \leqslant \log_2(U+1+|A|)$ and the writing of it and n in some number system takes a time bounded by a polynomial of the $|P|$ and $|X|$. Further, the P_1 moves again along the record of the X and P and writes the formula $\chi(0)(\widehat{y}_0)$ at first, after that it writes $\Phi^{(0)}(P)(\widehat{a}_1, \widehat{b}_1))$, and finally, it designs $\Omega(X,P)$. It is clear that this process takes the time, which is no greater than the value of $p(|X|+|P|)$ for some polynomial $p(y)$.

The single-tape variant P_2 of the machine P_1 will do the same actions in the time equal to $g(|X|+|P|)$, which is of the form of $\mathcal{O}([p(|X|+|P|)]^2)$ [1],[2]. □

7 The complexity of the theory of a single equivalence relation

Let $\mathfrak{K}$ be a class of the algebraic systems, whose signature (or underlying language) σ contains the symbol of the binary predicate $\backsim$, and this predicate is interpreted as an equivalence relation on every structure of the class, in particular, $\backsim$ may be an equality relation. We denote these relations by the same symbol.

Definition 3. Let us assume that there exists a $\backsim$-*nontrivial* system $\mathcal{E}$ in a class $\mathfrak{K}$, namely, this structure contains at least two $\backsim$-nonequivalent elements. Then the class $\mathfrak{K}$ is also termed $\backsim$-*nontrivial.* A theory $\mathcal{T}$ is named $\backsim$-*nontrivial* if it has a $\backsim$-nontrivial model. When $\backsim$ is either the equality relation or there is a formula $N(x,y)$ of the signature σ such that the sentence $\exists x,y N(x,y)$ is consistent with the theory $Th(\mathfrak{K})$ (or $\mathcal{T}$, or belongs to $Th(\mathcal{E})$), and this formula means that the elements x and y are not equal, then we will replace the term "$\backsim$-nontrivial" with *"equational-nontrivial"*.

Theorem 7.1. *Let $\mathcal{E}$, $\mathfrak{K}$, and $\mathcal{T}$ accordingly be a $\backsim$-nontrivial system, class, and theory of the signature σ, in particular, they may be equational-nontrivial. Then there exists an algorithm such that for every program P and any its input X, builds the sentence $\Omega^{(T)}(X,P)$ of the signature σ, where $T \in \{Th(\mathcal{E}), Th(\mathfrak{K}), \mathcal{T}\}$; this formula possesses the properties (i) and (ii) of the word $S(P,X)$ from the statement of Proposition 2.1 for $\mathcal{L}=T$; $F(|X|)=\exp(|X|)$. Moreover, for each $\varepsilon>0$, there is a constant $E_{T,\sigma}$ such that the inequality $|\Omega^{(T)}(X,P)| \leqslant E_{T,\sigma} \cdot |P| \cdot |X|^{2+\varepsilon}$ holds for any sufficiently long X.*

Proof. At first, given X and P, we write a simulating sentence $\Omega(X,P)$ of the theory of the Boolean algebra $\mathcal{B}$ in the signature $\langle \cap, \cup, C, 0, 1 \rangle$ with the equality symbol $\approx$, applying Theorem 3.1. Then, we will transform it into the required formulae $\Omega^{(T)}(X,P)$ within a polynomial time.

For the sake of simplicity of denotations, we assume that the $\backsim$- nontrivial structure $\mathcal{E}$ is a model for the theory $\mathcal{T}$, belongs to the class $\mathfrak{K}$ and has the signature σ.

Let φ be a sentence of Boolean signature. We construct the closed formulae $\varphi^{(2.j)}$ so that $\mathcal{B} \models \varphi \Leftrightarrow \mathcal{E} \models \varphi^{(2.j)}$, where j can be 0,1, or 2 depending on the signature σ.

At the begining, we consider the case, when the σ contains the equivalence symbol $\backsim$ and the two constant symbols c_0 and c_1 such that $\mathcal{E} \models \neg c_0 \backsim c_1$.

In the first stage, we accordingly replace each occurrence of the subformulae of the kind $\exists y\psi$; $\forall x\psi$; $t \approx s$ with the formulae $\exists y((y \backsim c_0 \vee y \backsim c_1) \wedge \psi)$; $\forall x((x \backsim c_0 \vee x \backsim c_1) \rightarrow \psi)$; $t \backsim s$, where t and s are the terms.

We carry out the second stage's transformations during several passages until the formula ceases to change. In this stage, a) we replace the subformulae of the kind $C(t) \backsim s$ and $t \backsim C(s)$ with the formula $\neg t \backsim s$; b) if a term u is not the constant 0 or 1, then we replace the subformulae of the kind $t_1 \cup t_2 \cup \ldots \cup t_s \backsim u$ and $u \backsim t \cup t_1 \cup t_2 \cup \ldots \cup t_s$ with the formula $[(t_1 \backsim c_1 \vee t_2 \backsim c_1 \vee \ldots \vee t_s \backsim c_1) \rightarrow u \backsim c_1] \wedge [(t_1 \backsim c_0 \wedge t_2 \backsim c_0) \wedge \ldots \wedge t_s \backsim c_0) \rightarrow u \backsim c_0]$; and the subformulae $t_1 \cap t_2 \cap \ldots \cap t_s \backsim u$ and $u \backsim t_1 \cap t_2 \cap \ldots \cap t_s$ with the formula $[(t_1 \backsim c_0 \vee t_2 \backsim c_0 \vee \ldots \vee t_s \backsim c_0) \rightarrow u \backsim c_0] \wedge [(t_1 \backsim c_1 \wedge t_2 \backsim c_1 \wedge \ldots \wedge t_s \backsim c_1) \rightarrow u \backsim c_1]$. We complete the second stage by replacing the constants 0 and 1 with the constants c_0 and c_1 respectively.

In the third stage, we replace accordingly each occurrence of the subformulae of the kind $t_1 \cup t_2 \cup \ldots \cup t_s \backsim c_1$, $t_1 \cap t_2 \cap \ldots \cap t_s \backsim c_1$, $t_1 \cup t_2 \cup \ldots \cup t_s \backsim c_0$, $t_1 \cap t_2 \cap \ldots \cap t_s \backsim c_0$ with the formulae $t_1 \backsim c_1 \vee t_2 \backsim c_1 \vee \ldots \vee t_s \backsim c_1$, $t_1 \backsim c_1 \wedge t_2 \backsim c_1 \wedge \ldots \wedge t_s \backsim c_1$, $t_1 \backsim c_0 \wedge t_2 \backsim c_0 \wedge \ldots \wedge t_s \backsim c_0$, $t_1 \backsim c_0 \vee t_2 \backsim c_0 \vee \ldots \vee t_s \backsim c_0$.

We execute these transformations as long as the record contains at least one symbol of the signature of Boolean algebras. The number of such symbols is decreased at least by one on every passage for the second and third stages, and the first stage can be realized on the only passage. So we need at most n passages, where n is a length of the sentence φ. The length of the whole record grows linearly on each pass, since the transformation of the kind b) of the second stage is longest, but even this transformation increases the length no more than in five times (for $s = 2$).

Nevertheless, the length of the resulting record $\varphi^{(2.0)} \rightleftharpoons \varphi^{\backsim}_{c_0,c_1}$ can increase non-linearly in common case, for instance, if φ contains an atomic formula of the kind $\bigcup_i \Big\{ \bigcap_j \Big[\bigcup_k \Big(\ldots \Big) \Big] \Big\} \backsim u$, where the number of alternations of the "big" conjunctions and disjunctions depends on the n.

However, there are not such subformulae in the sentence $\Omega(X, P)$ simulating for the theory of algebra $\mathcal{B}$. Indeed, in accordance with its definition, the conversion of the subformulae of the kind $\widehat{x}_t \approx \widehat{u}(\beta)$ (this is the system of equalities) and $\neg\,\widehat{w} \approx \widehat{u}(\beta)$ (this is the disjunction of inequalities) make the most increase if $\beta \in \{R, L\}$, because they comprise the atomic formulae of the form $x_{t,j} \approx u_j \oplus u^{\beta}_{j+1} \cdot \ldots \cdot u^{\beta}_n$ and $\neg\, w_j \approx u_j \oplus u^{\beta}_{j+1} \cdot \ldots \cdot u^{\beta}_n$, where u^{β}_k is either u_k for $\beta = R$ or Cu_k for $\beta = L$ — see Subsection 3.3. We recall that these subformulae are $x_{t,j} \approx [u_j \cap C(u^{\beta}_{j+1} \cap \ldots \cap u^{\beta}_n)] \cup [Cu_j \cap u^{\beta}_{j+1} \cap \ldots \cap u^{\beta}_n]$ and $\neg\, w_j \approx [u_j \cap C(u^{\beta}_{j+1} \cap \ldots \cap u^{\beta}_n)] \cup [Cu_j \cap u^{\beta}_{j+1} \cap \ldots \cap u^{\beta}_n]$ by our denotation. So, we need to perform only the three transformations of the kind b) in order to convert the sentence $\Omega(X, P)$ into the $\Omega(X, P)^{(2.0)}$. Therefore the estimation $|\Omega(X, P)^{(2.0)}| \leqslant D_0 |\Omega(X, P)|$ is valid for appropriate constant D_0. Since one can execute every passage of any stage within $\mathcal{O}(|\Omega(X, P)|^2)$ steps, the entire transformation takes the polynomial time.

Now, let us suppose that the signature of the structure $\mathcal{E}$ has no the constant symbols. Then we replace the constants c_0 and c_1 in the formula $\varphi^{(2.0)} = \varphi^{\backsim}_{c_0,c_1}$ with the new variables a and b, respectively. We obtain the formula $\varphi^{\backsim}_{a,b}$, and write additionally the prefix after that: $\varphi^{(2.1)} \rightleftharpoons \exists a, b[\neg\, a \backsim b \;\&\; \varphi^{\backsim}_{a,b}]$. It is clear that $|\varphi^{(2.1)}| \leqslant 2|\varphi^{(2.0)}|$ for $|\varphi^{(2.0)})| \geqslant 11$, and so $|\Omega(X, P)^{(2.1)}| \leqslant D_1 |\Omega(X, P)|$ for appropriate constant D_1.

Finally, when the signature σ does not contain the equivalence symbol, but there exists a formula $N(x, y)$, which asserts that the elements x and y is not equal, then we replace every occurrence of the atomic subformula of the kind $t \backsim s$ in the $\varphi^{\backsim}_{a,b}$ with the formula $\neg N(t, s)$ and add the prefix: $\varphi^{(2.2)} \rightleftharpoons \exists a, b[N(a, b) \;\&\; \varphi^{N}_{a,b}]$. It is obvious that $|\varphi^{(2.2)}| \leqslant |N(x, y)| \cdot |\varphi^{2.1}|$, hence $|\Omega(X, P)^{(2.2)}| \leqslant D_2 |\Omega(X, P)|$ for some constant D_2.

One can easily prove by induction on the complexity of the formulae that the condition $\mathcal{B} \models \varphi$ is equally matched to one of the following conditions (depending on the signature σ): either $\mathcal{E} \models \varphi^{(2.0)}$, or $\mathcal{E} \models \varphi^{(2.1)}$, or $\mathcal{E} \models \varphi^{(2.2)}$. It is clear that if $\mathfrak{K}$ and $\mathcal{T}$ are the equational-nontrivial class and theory respectively, then the condition $\mathcal{B} \models \varphi$ is also tantamount to the conditions $Th(\mathfrak{K}) \vdash \forall a, b(N(a,b) \rightarrow \varphi^N_{a,b})$ and $\mathcal{T} \vdash \forall a, b(N(a,b) \rightarrow \varphi^N_{a,b})$. □

Corollary 7.1. *The recognition complexity of each $\backsim$-nontrivial decidable theory $\mathcal{T}$, in particular, equational-nontrivial, has the non-polynomial lower bound, more precisely* $\mathcal{T} \notin DTIME(\exp(D_{T,\sigma} \cdot n^{\delta}))$, *where* $\delta = (2+\varepsilon)^{-1}$, $D_{T,\sigma} = (E_{T,\sigma})^{-\delta}$.

Proof. It immediately follows from the theorem and Corollary 2.1. □

8 Results and discussions

Let us notice that nearly all of the decidable theories mentioned in the surveys [6], [17] are nontrivial regarding equality or equivalence. So, if we regard "the polynomial algorithm" as a synonym for "the fast-acting algorithm", then the quickly decidable theories are almost completely absent. Furthermore, the examples, given in the introduction and [4], [7], [8],[14]–[19], [21], show that the complexity of the recognition procedures can be perfectly enormous for many natural, and seemingly, relatively simple theories.

It seems plausible that the estimation obtained in Corollary 3.1 is precise enough. One can substantiate this assertion, if firstly, to find the upper bound on the recognition complexity of theory $Th(\mathcal{B})$ by the multi-tape Turing machines; secondly, to obtain the lower bound for this complexity for the same machines. The author suspects that the inequalities from Item (iii) of the main theorem are valid as well for the k-tape machines, but the constant D must be about in k times bigger at that.

Let us point out that the number of the alternation of quantifiers depends on the input length in the modeling formula $\Omega(X, P)$. Therefore this sentence does not belong to any language belonging to some class of the polynomial hierarchy (see for more details [2], [16]). However, if one can build the modeling formula belonging to some class, of this hierarchy, located above **P**, then the class **P** will be different from this class, as well as from **NP** (this class consists of the languages which are accepted by the nondeterministic Turing machines in polynomial time) [16].

8.1 The totality and locality of the simulating methods

The method of Cook's formulae has arisen for the modeling of the nondeterministic Turing machine actions within polynomial time, and evidently, the construction of Stockmeyer and Meyer is also applicable for the same simulation in polynomial space, provided that the running time of the machine is exponential. This is a significant advantage of these techniques.

However, our method of modeling utilizing formulae is ineligible for nondeterministic machines. More precisely, such modeling formula must be exponential in length, when nondeterministic machine runs in exponential space. Unfortunately, the corresponding example is too cumbersome for this paper. This example rests on that simple fact that if we set the values of the basic color t variables, then we can "see" only at most two the tape squares (the $\widehat{x}_t$th and maybe $\widehat{z}_t$th) when we are situated within the framework of our approach — see the proof of Proposition 5.1. So our simulating method is *strictly local, pointwise*. At the same time, the techniques of Cook and Stockmeyer and Meyer are *total*, since they allow us to "see" all of the tape cells simultaneously at any instant, if we, of course, slightly transform the formula from [20].

The author is sure that the technique of the direct encoding of machines continues to be a potent tool for investigating the computational complexity of theories, despite the emergence of other powerful approaches for obtaining the lower bounds on this complexity such as the Compton and Henson method [4] or the method of the bounded concatenations of Fleischmann, Mahr, and Siefkes [8].

Nevertheless, the coding of the machine computations into the models of the theory being studied is a very difficult task in many cases. Such coding is partly like to the modeling of the machine actions with the aid the defining relations, when one wants to prove the insolubility of some algorithmically problem for the finitely presented algebraical structures of given variety (see, for instance, [3], [11]). In both cases, we have the strong restrictions, which are dictated by the necessity to be within the framework of the given signature or variety. But the case of the algorithmic problem for the finitely presented structures is, perhaps, somewhat easier than the simulation of computations using the formulae of a certain theory. In the first case, we apply the suitable words consisting of the generators of the algebraical system for the description of tape configurations or their parts. The value of these words can change depending on the defining relations and the identity of variety. However, these changes have the local character relatively of the entire structure; whereas the variables can take on any values inside the system when we make a simulation in the second case.

The task becomes slightly easier if there are some constants in the theory signature. Just for this reason, we work with the Boolean algebra having two elements, but not with the language $TQBF$ consisting of the true quantified Boolean formulae.

Note also, that the simulation of the actions of the computational mechanisms, which is realized in [3] and [11] (these devices are the Minsky machines in the former, and they are the Minsky operator algorithms in the latter), is total. On the other hand, this modeling is somewhat like the Compton and Henson method too. Indeed, in all of these cases, the coding of computations is done once and for all. In [4], this is made for Turing machines in proving the inseparability results; then, the authors transfer the obtained lower bounds from one theory to another, using interpretations. Both in [3] and [11], such simulation is made in proving the insolubility of the words problem for the appropriate module over a certain integral domain; afterward, this module is embedded (isomorphically in the first article and homomorphically in the second) in the solvable group under construction.

8.2 Entirely simultaneous and conventionally sequential modelings

Let us investigate the modernization of the Stockmeyer and Meyer method that arose in this paper.

It might seem that the formula $\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ is an analogue of the Cook's method formula $A_{0,m}(\widetilde{U},\widetilde{V})$, which was applied in the proof of Theorem 4.3 in [20], here $\widetilde{U}$ and $\widetilde{V}$ are the sequences $(u_1,\ldots,u_m)$ and $(v_1,\ldots,v_m)$ of the Boolean variables and $m=q(|X|)$ is the value of suitable polynomial q on the length of input X; this m and our P are n and $\mathfrak{M}$ in [20]. Indeed, it is said in [20] that the formula $A_{0,m}(\widetilde{U},\widetilde{V})$ is satisfiable if and only if the configuration encoded by the formula $v_1\ldots v_m$ follows from the configuration that corresponds to $u_1\ldots u_m$ in at most one step of the P.

So, the $A_{0,m}(\widetilde{U},\widetilde{V})$ can be considered as the conjunction of the formulae $u_1\ldots u_m$, $v_1\ldots v_m$, which describe the adjacent configurations, and also of the formula that describes the transformation from one configuration to another. One can regard that this transfer formula has the kind $A_t\&B_t\&C_t\&D_t\&A_{t+1}\&B_{t+1}\&C_{t+1}\&D_{t+1}\&E_{t,t+1}$, where the $A_t, A_{t+1}, B_t, B_{t+1}, C_t, C_{t+1}, D_t, D_{t+1}$, and $E_{t,t+1}$ are the subformulae of the formulae A, B, C, D, and E respectively from the proof of Theorem 10.3 in [1] and are obtained from them by means the restriction of the last formulae on the fixed value of the parameter t.

However, the author believes nevertheless that the analogue of the formula $A_{0,m}(\widetilde{U},\widetilde{V})$ is the $\Phi^{(o)}(P)(\widehat{y}_t,\widehat{y}_{t+1})$. In other words, the former corresponds to the $A_t\&B_t\&C_t\&D_t\&A_{t+1}\&B_{t+1}\&C_{t+1}\&D_{t+1}\&E_{t,t+1}$, i.e., this formula simply describes the regulations of the transformation of one configuration to another, but it does not contain the descriptions of these configurations (the formulae $u_1 \dots u_m$ and $v_1 \dots v_m$) in explicit form, because the intermediate configurations on the tape are unknown for us. We can know only the initial configuration and the fragment of the terminal one.

One can easily prove by induction that if $B_{s,m}(\widetilde{U},\widetilde{V}) \rightleftharpoons u_1 \dots u_m \;\&\; A_{s,m} \;\&\; v_1 \dots v_m$, then $\exists\widetilde{U}\exists\widetilde{V} B_{s,m}(\widetilde{U},\widetilde{V})$ is true if and only if the configuration encoded by $v_1 \dots v_m$ follows from the configuration that corresponds to $u_1 \dots u_m$ in at most $\exp(s)$ steps of the P. So this $B_{s,m}(\widetilde{U},\widetilde{V})$ is the simulating formula in [20], and it is analog of our $\Omega^{(s)}(X,P)(\widehat{y}_t,\widehat{y}_{t+e(s)})$. It is quite clear that if the $B_{s,m}(\widetilde{U},\widetilde{V})$ is satisfable, then the $A_{s,m}(\widetilde{U},\widetilde{V})$ is the same. In addition, the converse is also true for $s=0$. This is easy seen from the description of formulae A, B, C, D, and E given in [1], since these formulae contain all components of the tuples $\widetilde{U}$ and $\widetilde{V}$.

Since the $B_{0,m}$ has the form "$configuration(t)$ & $step(t+1)$ & $configuration(t+1)$", we can say that the *completely simultaneous modeling* has been applied in [5], [20].

But the formula $\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ is constructed in another way. It asserts that if the descriptions of the tth step's configuration (the formula $\Psi K(t)(\widehat{y}_t)$) and of the step $t+1$ (the $\Phi^{(0)}(P)(\widehat{y}_t,\widehat{y}_{t+1})$) are correct, then the configuration, which appeared after this step, will be adequately described as well (by the $\Psi K(t+1)(\widehat{y}_{t+1})$). We call this approach as a *conventionally sequential modeling* of actions, i.e., the $\Omega^{(0)}(X,P)$ has such structure: "$configuration(t)$ & $step(t+1) \rightarrow configuration(t+1)$".

Thus, the designs of the formulae $\Omega^{(0)}(X,P)(\widehat{y}_t,\widehat{y}_{t+1})$ and $A_{0,m}(\widetilde{U},\widetilde{V})$ are essentially different, if even one does not take into consideration the presence of the inner quantifiers in the former. Furthermore, their free variables "demand" the quantifiers of the various kind in order the formulae become true. Seemingly, just this difference in the external quantifiers dictates the difference, above mentioned, in the internal structure of these formulae. The additional argument for this conclusion is that the conventionally sequential modeling is also used in [3] and [11]. Recall in this connection that the investigation of the finitely presented algebraic system, which is given with the aid of the generators $g_1,\dots,g_k$ and the defining relations $R_1(g_1,\dots,g_k),\dots, R_m(g_1,\dots,g_k)$, is equivalent (in many respects) to the study of the formulae of the kind

$$\forall g_1 \dots \forall g_k[(R_1(g_1,\dots,g_k)\& \dots \& R_m(g_1,\dots,g_k)) \rightarrow S(g_1,\dots,g_k)].$$

Moreover, we saw in Proposition 5.1(i) that the $\Omega^{(0)}(X,P)$ can completely model too, but existential quantifiers are applied at that.

8.3 Open problems

It is well known that the theory of two equivalence relations is not decidable, but the theory of one such relation $\sim$ is decidable [6]. Now it turns out according to Corollary 7.1 that although it is decidable, but for a very long time.

What will happen if we add some unary predicates or functions to the signature with the only equivalence symbol $\sim$ so that the resulting theory remains decidable? Will it be possible to find such functions and/or predicates in order that the recognition complexity "smoothly" increases? We can formulate this in a more precise way.

Problem 1. *Let $\sigma_0,\sigma_1,\dots$ be a sequence of signatures such that $\sigma_0 \supseteq \{\sim\}$ or $\sigma_0 \supseteq \{\approx\}$ and $\sigma_i \subset \sigma_{i+1}$ for each natural i. Does there exist a sequence of the algebraical structures $\mathfrak{M}_0,\mathfrak{M}_1,\dots$ such that their signatures accordingly are $\sigma_0,\sigma_1,\dots$ and*

$$Th(\mathfrak{M}_j) \in DTIME(\exp_{j+2}(n)) \;\setminus\; DTIME(\exp_{j+1}(n))?$$

Recall that $Th(\mathfrak{M})$ denotes the first-order theory of the system $\mathfrak{M}$. It is possible that there already is a candidate for the like sequence of the higher-order theories with the "smoothly" increasing recognition complexity.

Problem 2. *Let $\Omega^{(k)}$ be a fragment of the type theory Ω from [21], which is obtained with the aid the restriction of the types of variables by level k. Can one point out for each natural k such a number s that $\Omega^{(k)} \in DTIME(\exp_{k+s+1}(n)) \setminus DTIME(\exp_{k+s}(n))$?*

A sequence of theories of concrete linear orders with additional unary predicates (similar theory is described in Example 2 in [10]) seems to be another possible candidate for the role of a sequence of algebraic systems that has a "smooth" growing complexity of recognizing theories (in the sense described above). It is also interesting to find a sequence of signatures and corresponding structures $\mathfrak{N}_0, \mathfrak{N}_1, \ldots$ for which this complexity increases "quite smoothly", i.e., $Th(\mathfrak{N}_j) \in DTIME(\exp(n^{j+1})) \setminus DTIME(\exp(n^j))$. Signatures with unary predicates and constants seem to be suitable for this, besides, the theories with such signatures are well studied [13].

Problem 3. *It seems quite plausible that the theory of finite Boolean algebras has a double exponential as the lower bound on the complexity of recognition.*

Problem 4. *Let $F(n)$ be a limit upper bound for all polynomials (see Definition 2). What algebraic and/or model-theoretic properties must be possessed an algebraical structure $\mathfrak{A}$ in order that $Th(\mathfrak{A}) \in DTIME(F(n))$ or $Th(\mathfrak{A}) \notin DTIME(F(n))$ holds?*

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The design and analysis of computer algorithms.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1976.

[2] S. Arora, B. Barak, *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[3] G. Baumslag, D. Gildenhuys, R. Strebel *Algorithmically insoluble problems about finitely presented solvable groups, Lie and associative algebras I.* J. Pure and Applied Algebra, 39 (1986), no. 1–2, 53–94.

[4] K.J. Compton, C.W. Henson, *A uniform method for proving lower bounds on the computational complexity of logical theories.* Ann. Pure Appl. Logic, 48 (1990), no. 1, 1–79.

[5] S.A. Cook, *The complexity of theorem-proving procedures.* in Proc. of the 3^{rd} Annual ACM Symposium on the Theory of Computing, Shaker Heights, Ohio (1971), 151–159.

[6] Y.L. Ershov, I.A. Lavrov, A.D. Taimanov, M.A. Taitslin, *Elementary theories.* Russian Math, Surveys 20 (1965), 35–100 (English translation).

[7] M.J. Fisher, M.O. Rabin, *Super exponential complexity of Presburger's arithmetic.* SIAM, AMS Proceedings, 7 (1974), 27–41.

[8] K. Fleischmann, B. Mahr,, D. Siefkes, *Bounded concatenation theory as a uniform method for proving lower complexity bounds.* in: R. Gandy, M. Hyland, eds., Logic Colloquium, 76 (1977), North-Holland, Amsterdam,, 471–490.

[9] M.R. Garey, D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness.* Freeman, New York, 1979.

[10] B.Sh. Kulpeshov, S.V. Sudoplatov, *Distributions of countable models of quite o-minimal Ehrenfeucht theories.* Eurasian Math. J., 11 (2020), no. 3, 66-78.

[11] I.V. Latkin, *The occurrence problem for subvarieties of the variety* $\mathbf{N}_2\mathbf{A}$*, revisited.* Algebra and Logic, 40 (2001), no 5, 327–333.

[12] A.I. Malcev, *Algorithms and recursive functions.* Wolters–Noordhoff Pub. Co., Groningen, 1970.

[13] N.D. Markhabatov, S.V. Sudoplatov, *Ranks for families of all theories of given languages.* Eurasian Math. J., 12 (2021), no. 2, 52-58.

[14] A.R. Meyer, *Weak monadic second order theory of successor is not elementary-recursive.* Technical Report (Memo 38) Massachusetts Institute of Technology, Cambridge, MA, USA, 1973.

[15] A.R. Meyer, *The inherent complexity of the theories of ordered sets.* in: International Congress of Mathematians. Vancouver, 1974, also in Canadian Math. Congress, (1975), 477–482.

[16] A.R. Meyer, L.J. Stockmeyer *The equivalence problem for regular expressions with squaring requires exponetial space.* in: Proc. Thirteenth Annual IEEE Symposium on Switching and Automata theory (1972), 125–129.

[17] M.O. Rabin, *Decidable theories.* in Handbook of mathematical logic, Part C., Chapter 3, Ed. J. Barwise, Elsevier, Amsterdam, New York, Eighth impression: 1999, 596–629.

[18] E.L. Robertson, *Structure of complexity in the weakmonadic second-order theories of the natural numbers.* Research Report CS-73-31, Dept. of Applied Analysis and Computer Science, University of Waterloo, Dec. 1973; also Proc. 6th ACM Symp. on Theory of Computing (1974), 161–171.

[19] L.J. Stockmeyer, *The complexity of decision problems in automata theory and logic.* PhD thesis, MIT Lab for Computer Science, 1974; also /MIT/LCS TechRep 133.

[20] L.J. Stockmeyer, A.R. Meyer, *Word problems requiring exponential time.* in Proc. 5^{th} Ann. ACM Symp. on the Theory of Computing, Association for Computing Machinery, New York, 1973, 1–9.

[21] S. Vorobyov, *The most nonelementary theory.* Information and Computation, 190 (2004), no. 2, 196–219.

Ivan Vasil'evich Latkin
Faculty of Basic Engineering Training
D. Serikbayev East Kazakhstan Technical University
69 Protozanov St.
070004 Ust-Kamenogorsk, Kazakhstan
E-mails: lativan@yandex.ru

Received: 11.10.2019