logics

MDPI

Article

Algorithmic Problems for Computation Trees

Mikhail Moshkov

Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia; mikhail.moshkov@kaust.edu.sa

Abstract: In this paper, we study three algorithmic problems involving computation trees: the optimization, solvability, and satisfiability problems. The solvability problem is concerned with recognizing computation trees that solve problems. The satisfiability problem is concerned with recognizing sentences that are true in at least one structure from a given set of structures. We study how the decidability of the optimization problem depends on the decidability of the solvability and satisfiability problems. We also consider various examples with both decidable and undecidable solvability and satisfiability problems.

Keywords: structure; computation tree; optimization

1. Introduction

This paper is devoted to the study of three algorithmic problems related to computation trees: problems of optimization, solvability, and satisfiability. Computation trees can be used as algorithms for solving problems of combinatorial optimization, computation geometry, etc. Computation trees are a natural generalization of decision trees: in addition to one-place operations of the predicate type (attributes) which are used in decision trees, in computation trees, many-place predicate and function operations may be used. Algebraic computation trees and some of their generalizations have been studied most intensively [1–3]. This paper continues the line of research introduced by the work of [4]: this work does not concentrate on specific types of computation trees, such as algebraic, but studies computation trees over arbitrary structures.

Our consideration is based on the notions of signature and structure of this signature. Signature σ is a finite or countable set of predicate and function symbols with their arity. Structure U of the signature σ is a pair (A, I), where A is a nonempty set called the universe of U, and I is an interpretation function mapping the symbols of σ to predicates and functions in A.

Each computation tree over structure U is a pair (S, U) of a computation tree scheme S of the signature σ and the structure U. Such computation trees are used to solve problems over structure U, each of which is a pair (s, U) of a problem scheme s of the signature σ and the structure U. The considered notions are discussed in Sections 2–4.

For a given class C of structures of the signature σ, we say that a scheme of computation tree S solves a scheme of problem s relative to the class C if, for any structure $U \in C$, the computation tree (S, U) solves the problem (s, U).

We define the notion of so-called strictly limited complexity measure ψ over the signature σ and consider the following problem of optimization. For a given sentence α from a set of sentences H of the signature σ and scheme of problem s, we should find a scheme of computation tree S which satisfies the following conditions: (i) S solves the scheme of problem s relative to the class $C(\alpha)$ consisting of all structures $U \in C$ for which α is true in U, and (ii) S has the minimum ψ-complexity among such schemes of

check for updates

Academic Editors: Valentin Goranko and Sujata Ghosh

Received: 12 March 2025
Revised: 11 April 2025
Accepted: 6 May 2025
Published: 9 May 2025

Citation: Moshkov, M. Algorithmic Problems for Computation Trees. *Logics* **2025**, *3*, 4. https://doi.org/10.3390/logics3020004

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

computation trees. We study the decidability of this problem depending on the decidability of problems of solvability and satisfiability.

The problem of solvability: for an arbitrary sentence $\alpha \in H$, scheme of problem s, and scheme of computation tree S, we should recognize if the scheme of computation tree S solves the scheme of problem s relative to the class $C(\alpha)$.

In Section 5, we prove that the problem of solvability is decidable if and only if the following problem of satisfiability is decidable: for an arbitrary sentence $\beta \in H \wedge \Phi_\exists$, we should recognize if there exists a structure $U \in C$ for which β is true in U. Here, $H \wedge \Phi_\exists$ is the set of sentences $\alpha \wedge \gamma$ such that $\alpha \in H$, $\gamma \in \Phi_\exists$, and $\Phi_\exists$ is the set of sentences of the signature σ in prenex form with the prefixes of the form $\exists \cdots \exists$.

In the same section, we prove that if H is equal to the set of all sentences of the signature σ, then the problem of solvability is decidable if and only if the theory of the class C is decidable. This gives us many examples of both decidability and undecidability of the satisfiability problem.

We will find more examples in Section 6, where we study the case of the satisfiability problem when $H = H_1 \wedge \cdots \wedge H_n$ and, for $i = 1, \ldots, n$, H_i is the set of all sentences of the signature σ in prenex form with prefixes in a nonempty set of words in the alphabet $\{\forall, \exists\}$ that is closed relative to subwords. We describe all classes of sentences $H_1 \wedge \cdots \wedge H_n \wedge \Phi_\exists$ for which the problem of satisfiability is decidable both for the case when C is the class of all structures of the signature σ and for the case when C is the class of all finite structures of the signature σ.

In Section 7, we prove that, for any strictly limited complexity measure, the problem of optimization is undecidable if the problem of satisfiability is undecidable. We also prove that the problem of optimization is decidable if the problem of solvability is decidable and the considered strictly limited complexity measure satisfies some additional condition.

Section 8 contains short conclusions.

2. Sequences of Expressions

In this section, we consider the notions of function and predicate expressions of a signature σ and the notion of an n-sequence of such expressions. These notions are used when we consider schemes of problems and computation trees.

Let σ be a finite or countable signature containing generally both predicate and function symbols and their arity. This paper will not consider nullary predicate symbols. The paper will specifically mention when nullary function symbols (constants) are not allowed.

Let $\omega = \{0, 1, 2, \ldots\}$, $E_2 = \{0, 1\}$ and $X = \{x_i : i \in \omega\}$ be the set of variables. For $n \in \omega \setminus \{0\}$, we denote $X_n = \{x_0, \ldots, x_{n-1}\}$. Let α be a formula of the signature σ. We will use the following notation: $\alpha^1 = \alpha$ and $\alpha^0 = \neg\alpha$.

Definition 1. *A function expression of the signature σ is an expression of the form $x_j \Leftarrow f(x_{l_1}, \ldots, x_{l_m})$, where f is an m-ary function symbol of the signature σ. A predicate expression of the signature σ is an expression of the form $x_{l_1} = x_{l_2}$ or an expression of the form $r(x_{l_1}, \ldots, x_{l_k})$, where r is a k-ary predicate symbol of the signature σ.*

Definition 2. *Let $n \in \omega \setminus \{0\}$. An n-sequence of expressions is a pair (n, β), where β is a finite sequence of function and predicate expressions.*

Let $\beta = \beta_1, \ldots, \beta_m$. For $i = 1, 2, \ldots, m$, we correspond to the expression β_i a sequence $M_i = t_{i0}, t_{i1}, \ldots$ of terms of the signature σ with variables from X_n. Let $M_1 = x_0, x_1, \ldots, x_{n-2}, x_{n-1}, x_{n-1}, \ldots$. Let the sequences $M_1, \ldots, M_i$, $m > i \geq 1$, be already defined. We now define the sequence M_{i+1} corresponding to the expression β_{i+1}. If β_i is a

predicate expression, then $M_{i+1} = M_i$. Let β_i be a function expression $x_j \Leftarrow f(x_{l_1}, \ldots, x_{l_m})$. Then, $M_{i+1} = t_{i0}, \ldots, t_{ij-1}, f(t_{il_1}, \ldots, t_{il_m}), t_{ij+1}, t_{ij+2}, \ldots$.

We associate with each predicate expression β_i of the sequence β an atomic formula $\kappa(n, \beta, \beta_i)$ of the signature σ with variables from X_n. Let β_i be the expression $x_{l_1} = x_{l_2}$. Then, $\kappa(n, \beta, \beta_i)$ is equal to $t_{il_1} = t_{il_2}$. Let β_i be the expression $r(x_{l_1}, \ldots, x_{l_k})$. Then, $\kappa(n, \beta, \beta_i)$ is equal to $r(t_{il_1}, \ldots, t_{il_k})$.

3. Schemes of Computation Trees and Computation Trees

In this section, we discuss the notions of a scheme of computation tree and its complete path, as well as the notion of a computation tree.

Later, we will specifically clarify whether equality $=$ can be used in the considered formulas of the signature σ.

For $n \in \omega \setminus \{0\}$, denote $Q_n^=(\sigma)$ as the set of formulas of the signature σ with variables from X_n that are atomic formulas or negations of atomic formulas. Equality sign $=$ in $Q_n^=(\sigma)$ means that formulas of the form $t_1 = t_2$ and $\neg(t_1 = t_2)$ belong to $Q_n^=(\sigma)$, where t_1 and t_2 are terms of the signature σ with variables from X_n. We denote by $Q_n(\sigma)$ the set of formulas from $Q_n^=(\sigma)$ that do not contain equality. The set $Q_n(\sigma)$ contains only formulas of the form $r(t_1, \ldots, t_k)$ or $\neg r(t_1, \ldots, t_k)$, where r is a k-ary predicate symbol from σ, and $t_1, \ldots, t_k$ are terms of the signature σ with variables from X_n.

Definition 3. *A scheme of computation tree of the signature σ is a pair $S = (n, G)$, where $n \in \omega \setminus \{0\}$ and G define a finite directed tree with root. This tree has three types of nodes: function, predicate, and terminal. A function node is labeled with a function expression of the signature σ. One edge leaves the function node. This edge is not labeled. A predicate node is labeled with a predicate expression of the signature σ. This expression can be of the form $x_{l_1} = x_{l_2}$ if we allow equality. Two edges leave the predicate node. One edge is labeled with the number 0, and another edge is labeled with the number 1. A terminal node is labeled with a number from ω. This node has no leaving edges.*

The set X_n will be called the set of input variables of the scheme of computation tree S. Usually, we will not distinguish the scheme S and its graph G.

Definition 4. *A complete path of the scheme S is a directed path, which starts in the root and finishes in a terminal node of S.*

Let $\tau = w_1, d_1, w_2, d_2, \ldots, w_m, d_m, w_{m+1}$ be a complete path of the scheme S. We denote $\beta = \beta_1, \ldots, \beta_m$ as the sequence of function and predicate expressions attached to nodes $w_1, \ldots, w_m$. Let us consider the n-sequence of expressions of (n, β). We correspond to each predicate node w_i of the path τ the formula $\kappa(n, \beta, \beta_i)^c$ from $Q_n^=(\sigma)$, where c is the number attached to the edge d_i.

We now define the set of formulas $F(\tau)$ associated with the complete path τ. If τ contains predicate nodes, then $F(\tau)$ is the set of formulas associated with the predicate nodes of τ. If τ does not contain predicate nodes, then $F(\tau) = \{\mu_\tau\}$, where μ_τ is the formula $x_0 = x_0$ if equality is allowed and $\neg r(x_0, \ldots, x_0) \vee r(x_0, \ldots, x_0)$ if equality is not allowed, and r is a predicate symbol of the signature σ. Note that if τ does not contain predicate nodes, then τ is the only complete path of the scheme S. Let $F(\tau) = \{\alpha_1, \ldots, \alpha_k\}$. Denote by $\pi_\tau(S)$ the formula $\alpha_1 \wedge \cdots \wedge \alpha_k$. Denote by t_τ the number attached to the terminal node w_{m+1} of the path τ.

Definition 5. *Let U be a structure of the signature σ with the universe A and $S = (n, G)$ be a scheme of computation tree of the signature σ. The pair $\Gamma = (S, U)$ will be called a computa-*

tion tree over U with the set of input values X_n. The scheme S will be called the scheme of the computation tree Γ.

Definition 6. *A complete path τ of the scheme S will be called realizable in the structure U for a tuple $\bar{a} \in A^n$ if any formula from $F(\tau)$ is true in U on the tuple $\bar{a}$.*

One can show that there exists exactly one complete path of S that is realizable in the structure U for the tuple $\bar{a}$.

We correspond to the computation tree Γ a function $\varphi_\Gamma : A^n \to \omega$. Let $\bar{a} \in A^n$ and τ be a complete path of the scheme S that is realizable in the structure U for the tuple $\bar{a}$. Then, $\varphi_\Gamma(\bar{a}) = t_\tau$. We will say that the computation tree Γ implements the function φ_Γ.

We denote by $\mathrm{Tree}^=(\sigma)$ the set of schemes of computation trees of the signature σ. Denote by $\mathrm{Tree}(\sigma)$ the set of schemes from $\mathrm{Tree}^=(\sigma)$ that do not contain equality in the expressions attached to nodes.

4. Schemes of Problems and Problems

In this section, the notions of a scheme of problem and its special representation, as well as the notion of a problem, are considered. The notion of a computation tree solving a problem is also defined.

Definition 7. *A scheme of problem of the signature σ is a tuple $s = (n, \nu, \beta_1, \ldots, \beta_m)$, where $n \in \omega \setminus \{0\}$, $m \in \omega \setminus \{0\}$, $\beta_1, \ldots, \beta_m$ are functions and predicate expressions of the signature σ, and there is $k \in \omega \setminus \{0\}$ such that $\nu : E_2^k \to \omega$, and there are exactly k predicate expressions in the sequence $\beta_1, \ldots, \beta_m$.*

The set X_n is called the set of input variables for the scheme of problem s. We denote by β the sequence of expressions $\beta_1, \ldots, \beta_m$. Let us consider the n-sequence of expressions of (n, β). Let $\beta_{i_1}, \ldots, \beta_{i_k}$ be all predicate expressions in the sequence β. We correspond to each predicate expression β_{i_j} the formula $\alpha_j = \kappa(n, \beta, \beta_{i_j})$ from $Q_n^=(\sigma)$. The tuple $(n, \nu, \alpha_1, \ldots, \alpha_k)$ will be called the special representation of the scheme of problem s. We correspond to the scheme s and to each tuple $\bar{\delta} = (\delta_1, \ldots, \delta_k) \in E_2^k$ the formula $\pi_{\bar{\delta}}(s) = \alpha_1^{\delta_1} \wedge \cdots \wedge \alpha_k^{\delta_k}$.

Definition 8. *Let U be a structure of the signature σ consisting of a universe A and an interpretation I of symbols from σ. Let $s = (n, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem of the signature σ. The pair $z = (s, U)$ will be called a problem over U with the set of input variables X_n. The scheme s will be called the scheme of the problem z.*

Let us consider the special representation $(n, \nu, \alpha_1, \ldots, \alpha_k)$ of the scheme of problem s. The interpretation I corresponds to the formula α_j, $j = 1, \ldots, k$, a function from A^n to E_2 in a natural way: any predicate including predicates of the form $x_{l_1} = x_{l_2}$ can be considered as a function with values from the set E_2.

We correspond to the problem z the function $\varphi_z : A^n \to \omega$ defined as follows: $\varphi_z(\bar{a}) = \nu(\alpha_1(\bar{a}), \ldots, \alpha_k(\bar{a}))$ for any $\bar{a} \in A^n$. The problem z may be interpreted as a problem of searching for the number $\varphi_z(\bar{a})$ for a given $\bar{a} \in A^n$.

Definition 9. *Let $S = (n, G)$ be a scheme of computation tree of the signature σ. We will say that the computation tree $\Gamma = (S, U)$ solves the problem $z = (s, U)$ if the functions φ_Γ and φ_z coincide.*

We denote by $\mathrm{Probl}^=(\sigma)$ the set of schemes of problems of the signature σ. Denote by $\mathrm{Probl}(\sigma)$ the set of schemes from $\mathrm{Probl}^=(\sigma)$ that do not contain equality in the predicate expressions.

5. Solvability Versus Satisfiability

In this section, two algorithmic problems are considered. The problem of solvability is related to the recognition of schemes of computation trees solving schemes of problems relative to a class of structures. The problem of satisfiability is about the recognition of sentences that are true in at least one structure from a given set of structures. It is shown here that the problem of solvability is decidable if and only if corresponding to it the problem of satisfiability is decidable. Various examples with both decidable and undecidable solvability and satisfiability problems are also considered.

Let C be a class of structures of the signature σ.

Definition 10. *We will say that a scheme of computation tree $S = (n_1, G)$ from $\mathrm{Tree}^{=}(\sigma)$ solves a scheme of problem $s = (n_2, \nu, \beta_1, \ldots, \beta_m)$ from $\mathrm{Probl}^{=}(\sigma)$ relative to the class C if $n_1 = n_2$ and either $C = \emptyset$ or $C \neq \emptyset$, and for any structure U from C, the computation tree (S, U) solves the problem (s, U).*

For a sentence (formula without free variables) α of the signature σ, denote $C(\alpha) = \{U : U \in C, U \models \alpha\}$, where the notation $U \models \alpha$ means that the sentence α is true in U. Let H be a nonempty set of sentences of the signature σ, and let

$$(\mathrm{Probl}, \mathrm{Tree}) \in \{(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma)), (\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma))\}.$$

We can consider the pair $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma))$ only if the signature σ contains predicate symbols.

Definition 11. *We now define the problem of solvability for the quadruple $(\mathrm{Probl}, \mathrm{Tree}, H, C)$: for arbitrary $\alpha \in H$, $s \in \mathrm{Probl}$, and $S \in \mathrm{Tree}$, we should recognize if the scheme of computation tree S solves the scheme of problem s relative to the class $C(\alpha)$.*

Definition 12. *Let us define the problem of satisfiability for the pair (H, C): for an arbitrary $\alpha \in H$, we should recognize if there exists a structure $U \in C$ such that $U \models \alpha$.*

This is the general definition of the satisfiability problem. Later, for each version of the solvability problem, we will consider its corresponding version of the satisfiability problem and prove that they are both decidable or both undecidable.

Let $H_1, \ldots, H_m$ be nonempty sets of sentences of the signature σ. We denote by $H_1 \wedge \cdots \wedge H_m$ the class of sentences of the form $\alpha_1 \wedge \cdots \wedge \alpha_m$, where $\alpha_1 \in H_1, \ldots, \alpha_m \in H_m$. Let Π be a set of words in the alphabet $\{\forall, \exists\}$. Denote by $\Phi^{=}(\Pi, \sigma)$ the class of all sentences in prenex form of the signature σ with prefixes from Π. Denote by $\Phi(\Pi, \sigma)$ the class of all sentences from $\Phi^{=}(\Pi, \sigma)$ that do not contain equality.

Let w be a word in the alphabet $\{\forall, \exists, \forall^*, \exists^*\}$. The set $P(w)$ of prefixes is defined in the following way:

$$P(\forall^n) = \{\forall^i : 0 \leq i \leq n\},\ P(\exists^n) = \{\exists^i : 0 \leq i \leq n\},$$
$$P(\forall^*) = \{\forall^i : i \in \omega\},\ P(\exists^*) = \{\exists^i : i \in \omega\},$$
$$P(w_1 w_2) = \{u_1 u_2 : u_1 \in P(w_1), u_2 \in P(w_2)\}.$$

Denote $H^{=}(\sigma)$ as the set of sentences of the signature σ. Denote $H(\sigma)$ as the set of sentences of the signature σ that do not contain equality.

Theorem 1. *Let C be a nonempty class of structures of the signature σ and H be a nonempty set of sentences of the signature σ:*

(a) *Let $H \subseteq H^{=}(\sigma)$. The problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H, C)$ is decidable if and only if the problem of satisfiability for the pair $(H \wedge \Phi^{=}(P(\exists^*), \sigma), C)$ is decidable.*

(b) *Let $H \subseteq H(\sigma)$. The problem of solvability for the quadruple $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), H, C)$ is decidable if and only if the problem of satisfiability for the pair $(H \wedge \Phi(P(\exists^*), \sigma), C)$ is decidable.*

Proof. (a) Let the problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H, C)$ be decidable. We now describe an algorithm for solving the problem of satisfiability for the pair $(H \wedge \Phi^{=}(P(\exists^*), \sigma), C)$. Let $\alpha \in H$ and $\beta \in \Phi^{=}(P(\exists^*), \sigma)$. We construct a sentence of the signature σ that is logically equivalent to the sentence β and is of the form $\exists x_0 \ldots \exists x_{n-1}((\beta_{11} \wedge \cdots \wedge \beta_{1m_1}) \vee \cdots \vee (\beta_{k1} \wedge \cdots \wedge \beta_{km_k}))$, where, for $j = 1, \ldots, k$ and $i = 1, \ldots, m_j$, $\beta_{ji} \in Q_n^{=}(\sigma)$.

For $j = 1, \ldots, k$, we construct a scheme of computation tree $S_j = (n, G_j)$ from $\mathrm{Tree}^{=}(\sigma)$ such that for any structure U of the signature σ and any tuple $\bar{a} \in A^n$, where A is the universe of U,

$$\varphi_{(S_j,U)}(\bar{a}) = \begin{cases} 0, & \text{if } U \models \neg(\beta_{j1}(\bar{a}) \wedge \cdots \wedge \beta_{jm_j}(\bar{a})), \\ 1, & \text{if } U \models \beta_{j1}(\bar{a}) \wedge \cdots \wedge \beta_{jm_j}(\bar{a}). \end{cases}$$

We denote by $s_0 = (n, \nu, x_0 = x_0)$ the scheme of problem from $\mathrm{Probl}^{=}(\sigma)$ such that $\nu : E_2 \to \{0\}$, i.e., $\varphi_{(s_0,U)} \equiv 0$ for any structure U of the signature σ.

Using an algorithm that solves the problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H, C)$, for $j = 1, \ldots, k$, we recognize if the scheme of computation tree S_j solves the scheme of problem s_0 relative to the class $C(\alpha)$. It is not difficult to show that a structure $U \in C$ such that $U \models \alpha \wedge \beta$ exists if and only if there exists $j \in \{1, \ldots, k\}$ for which the scheme of computation tree S_j does not solve the scheme of problem s_0 relative to the class $C(\alpha)$.

Let the problem of satisfiability for the pair $(H \wedge \Phi^{=}(P(\exists^*), \sigma), C)$ be decidable. We now describe an algorithm solving the problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H, C)$. Let $\alpha \in H$, $S = (n_1, G)$ be a scheme of computation tree from $\mathrm{Tree}^{=}(\sigma)$, and let $s = (n_2, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem from $\mathrm{Probl}^{=}(\sigma)$ with the special representation $(n_2, \nu, \alpha_1, \ldots, \alpha_k)$.

If $n_1 \neq n_2$, then the scheme of computation tree S does not solve the scheme of problem s relative to the class $C(\alpha)$. Let $n_1 = n_2$. Let τ be a complete path of the scheme of computation tree S and $\bar{\delta} \in E_2^k$. Denote $\gamma_{\tau\bar{\delta}} = \exists x_0 \ldots \exists x_{n_1-1}(\pi_\tau(S) \wedge \pi_{\bar{\delta}}(s))$.

Using an algorithm solving the problem of satisfiability for the pair $(H \wedge \Phi^{=}(P(\exists^*), \sigma), C)$, for each pair of path τ and tuple $\bar{\delta}$ such that $t_\tau \neq \nu(\bar{\delta})$, we check if there exists a structure $U \in C$ in which the sentence $\alpha \wedge \gamma_{\tau\bar{\delta}}$ is true. One can show that the scheme of computation tree S does not solve the scheme of problem s relative to the class $C(\alpha)$ if and only if, for at least one of the considered pairs τ and $\bar{\delta}$, the sentence $\alpha \wedge \gamma_{\tau\bar{\delta}}$ is true in a structure from the class C.

(b) The second part of the theorem statement can be proved in a similar way. □

We now consider some corollaries of Theorem 1.

Definition 13. *Let C be a nonempty class of structures of the signature σ. The theory $\mathrm{Th}(C)$ of the class C consists of all sentences from $H^{=}(\sigma)$ that are true in all structures from C. This theory is called decidable if there exists an algorithm that, for a given sentence from $H^{=}(\sigma)$, recognizes if this sentence belongs to $\mathrm{Th}(C)$.*

Proposition 1. *Let C be a nonempty class of structures of the signature σ. The problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H^{=}(\sigma), C)$ is decidable if and only if the theory $\mathrm{Th}(C)$ of the class C is decidable.*

Proof. Let the theory $\mathrm{Th}(C)$ be decidable. Let $\alpha \in H^{=}(\sigma) \wedge \Phi^{=}(P(\exists^*), \sigma)$. It is clear that the sentence α is true in a structure from C if and only if the sentence $\neg\alpha$ does not belong to the theory $\mathrm{Th}(C)$. Taking into account that the theory $\mathrm{Th}(C)$ is decidable, we obtain that the problem of satisfiability for the pair $(H^{=}(\sigma) \wedge \Phi^{=}(P(\exists^*), \sigma), C)$ is decidable. Using Theorem 1, we obtain that the problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H^{=}(\sigma), C)$ is decidable.

Let the problem of solvability for the quadruple $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), H^{=}(\sigma), C)$ be decidable. Let $\alpha \in H^{=}(\sigma)$. Evidently, α does not belong to the theory $\mathrm{Th}(C)$ if and only if the sentence $\neg\alpha \wedge \exists x_0(x_0 = x_0)$ from $H^{=}(\sigma) \wedge \Phi^{=}(P(\exists^*), \sigma)$ is true in a structure from C. Using Theorem 1, we obtain that the problem of satisfiability for the pair $(H^{=}(\sigma) \wedge \Phi^{=}(P(\exists^*), \sigma), C)$ is decidable. Hence, the theory $\mathrm{Th}(C)$ is decidable. □

We now consider examples of classes of structures with decidable theory [5,6]:

- The class of abelian groups and the class of finite abelian groups.
- The class of finite fields.
- The field of real numbers.
- The field of complex numbers.
- The addition of natural numbers.

We now consider examples of classes of structures with undecidable theory [6,7]:

- The class of groups and the class of finite groups.
- The class of fields.
- The arithmetic of natural numbers.

6. Decidable Classes of Sentences

This section studies decidable classes of sentences of a special kind. The details of the study are described first.

Let σ be a finite or countable signature that does not contain nullary predicate symbols and nullary function symbols. A set Π of words in the alphabet $\{\forall, \exists\}$ will be called closed relative to subwords if, for any word $w \in \Pi$, any word obtained from w by the removal of some letters belongs to Π. Let $\Pi_1, \ldots, \Pi_n$ be nonempty sets of words in the alphabet $\{\forall, \exists\}$ closed relative to subwords. Let $\Pi_{n+1} = P(\exists^*)$. Denote $L^{=} = \Phi^{=}(\Pi_1, \sigma) \wedge \cdots \wedge \Phi^{=}(\Pi_n, \sigma)$, $K^{=} = L^{=} \wedge \Phi^{=}(\Pi_{n+1}, \sigma)$, $L = \Phi(\Pi_1, \sigma) \wedge \cdots \wedge \Phi(\Pi_n, \sigma)$, and $K = L \wedge \Phi(\Pi_{n+1}, \sigma)$.

We denote by C_σ the class of all structures of the signature σ and by C_σ^{fin} the class of all finite structures of the signature σ. The next statement follows directly from Theorem 1. It describes the connections between the solvability problems under consideration and the corresponding satisfiability problems.

Proposition 2. (a) *The problem of solvability for the quadruple* $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), L^{=}, C_\sigma)$ *is decidable if and only if the problem of satisfiability for the pair* $(K^{=}, C_\sigma)$ *is decidable.*

(b) *The problem of solvability for the quadruple* $(\mathrm{Probl}^{=}(\sigma), \mathrm{Tree}^{=}(\sigma), L^{=}, C_\sigma^{fin})$ *is decidable if and only if the problem of satisfiability for the pair* $(K^{=}, C_\sigma^{fin})$ *is decidable.*

(c) *The problem of solvability for the quadruple* $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), L, C_\sigma)$ *is decidable if and only if the problem of satisfiability for the pair* (K, C_σ) *is decidable.*

(d) *The problem of solvability for the quadruple* $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), L, C_\sigma^{fin})$ *is decidable if and only if the problem of satisfiability for the pair* (K, C_σ^{fin}) *is decidable.*

Let H be a class of sentences of the signature σ.

Definition 14. *We will say that the class* H *is decidable if the problem of satisfiability for the pair* (H, C_σ) *is decidable and the problem of satisfiability for the pair* (H, C_σ^{fin}) *is decidable.*

Definition 15. *We will say that a sentence α of the signature σ is satisfiable if α is true in a structure from C_σ. We will say that a sentence α of the signature σ is finite satisfiable if α is true in a structure from C_σ^{fin}.*

Definition 16. *We will say that the class H is a class of reduction if there exists an algorithm that corresponds to an arbitrary sentence α of the signature σ a sentence $\alpha' \in H$ such that α is satisfiable if and only if α' is satisfiable, and α is finite satisfiable if and only if α' is finite satisfiable. Note that each class of reduction is undecidable.*

In this section, based on the results from [8–11], all decidable classes of sentences K and $K^=$ are described.

6.1. Signatures with Predicate Symbols Only

In this section, we consider a finite or countable signature σ that does not contain function symbols and nullary predicate symbols.

Denote $P_1 = P(\exists^*\forall^*)$, $P_2 = P(\exists^*\forall^2\exists^*)$, and $P_3 = P(\exists^*\forall^2)$. In [9], a constant r_0 is defined in the following way: $r_0 = \lfloor \log_2 m^* \rfloor + 13$, where m^* is the number of states in a universal Turing machine.

Theorem 2. *Let the signature σ contain at least $r_0 + 1$ predicate symbols and not contain function symbols. Let $\Pi_1, \ldots, \Pi_n$ be nonempty sets of words in the alphabet $\{\forall, \exists\}$ closed relative to subwords, $\Pi_{n+1} = P(\exists^*)$, $K^= = \Phi^=(\Pi_1, \sigma) \wedge \cdots \wedge \Phi^=(\Pi_n, \sigma) \wedge \Phi^=(\Pi_{n+1}, \sigma)$, and $K = \Phi(\Pi_1, \sigma) \wedge \cdots \wedge \Phi(\Pi_n, \sigma) \wedge \Phi(\Pi_{n+1}, \sigma)$.*

(a) *Either the class K is decidable or it is a class of reduction. The same applies to the class $K^=$.*

(b) *K is a class of reduction if and only if $K^=$ is a class of reduction.*

(c) *If the class $K^=$ is decidable, then, with the exception of a finite number of sentences, the satisfiability for sentences from $K^=$ coincides with the finite satisfiability.*

(d) *K is decidable if and only if at least one of the following four possibilities occurs:*

(d.1) *σ contains only 1-ary predicate symbols.*

(d.2) *For $i = 1, \ldots, n$, $\Pi_i \subseteq P_1$.*

(d.3) *For $i = 1, \ldots, n$, $\Pi_i \subseteq P_2$.*

(d.4) *There exists $i_0 \in \{1, \ldots, n\}$ such that $\Pi_{i_0} \subseteq P_1 \cup P_2$ and, for each $i \in \{1, \ldots, n\} \setminus \{i_0\}$, $\Pi_i \subseteq P_3$.*

Proof. Statements (a), (b), and (c) follow directly from Theorems 19 and 21 [9].

Let σ be an infinite signature. From Theorem 19 [9], it follows that K is a class of reduction if and only if the signature σ contains at least one predicate symbol, wherein the arity is at least two, and at least one of the following two possibilities occurs:

- For some $i \in \{1, \ldots, n+1\}$, $\Phi(\Pi_i, \sigma)$ is a reduction class.
- For some $i, j \in \{1, \ldots, n\}$ such that $i \neq j$, $\forall\exists \in \Pi_i$, and $\forall^3 \in \Pi_j$.

From Theorem 3 [9], it follows that the class $\Phi(P_1 \cup P_2, \sigma)$ is decidable. Therefore if, for the class K, at least one of the possibilities (d.1), (d.2), (d.3), and (d.4) occurs, then K is decidable.

Let K be decidable. We now show that K satisfies at least one of the conditions (d.1), (d.2), (d.3), and (d.4).

If σ contains only 1-ary predicate symbols, then K satisfies the condition (d.1). Let σ contain at least one predicate symbol F, wherein the arity is at least two. Then, for any $i \in \{1, \ldots, n+1\}$, $\Phi(\Pi_i, \sigma)$ is a decidable class, and there are no $i, j \in \{1, \ldots, n\}$ such that $i \neq j$, $\forall\exists \in \Pi_i$ and $\forall^3 \in \Pi_j$. Taking into account that the signature σ is infinite and contains

the predicate symbol F, and using Theorem 3 [9], we obtain that, for any $i \in \{1, \ldots, n\}$, $\Pi_i \subseteq P_1 \cup P_2$.

A set of words Π in the alphabet $\{\forall, \exists\}$, which is closed relative to subwords, will be called as follows:

- $\emptyset$-set if $\Pi \subseteq P_1 \cup P_2$, $\forall\exists \notin \Pi$ and $\forall^3 \notin \Pi$.
- $\forall^3$-set if $\Pi \subseteq P_1 \cup P_2$, $\forall\exists \notin \Pi$ and $\forall^3 \in \Pi$.
- $\forall\exists$-set if $\Pi \subseteq P_1 \cup P_2$, $\forall\exists \in \Pi$ and $\forall^3 \notin \Pi$.
- $\forall^3$-$\forall\exists$-set if $\Pi \subseteq P_1 \cup P_2$, $\forall\exists \in \Pi$ and $\forall^3 \in \Pi$.

Evidently, if Π is an $\emptyset$-set, then $\Pi \subseteq P(\exists^*\forall^2) = P_3$. If Π is a $\forall^3$-set, then $\Pi \subseteq P_1$. If Π is a $\forall\exists$-set, then $\Pi \subseteq P_2$. If Π is a $\forall^3$-$\forall\exists$-set, then $\Pi \subseteq P_1 \cup P_2$. Note that $P_3 = P_1 \cap P_2$.

Let there exist i_0 such that Π_{i_0} is a $\forall^3$-$\forall\exists$-set. Then, for any $i \in \{1, \ldots, n\} \setminus \{i_0\}$, Π_i is an $\emptyset$-set, and $\Pi \subseteq P_3$. Therefore, K satisfies the condition (d.4).

Let, for any $i \in \{1, \ldots, n\}$, Π_i be not a $\forall^3$-$\forall\exists$-set.

If, for any $i \in \{1, \ldots, n\}$, Π_i is an $\emptyset$-set, then K satisfies the conditions (d.2) and (d.3).

Let there exist $i \in \{1, \ldots, n\}$ such that Π_i is a $\forall^3$-set. Then, evidently, for any $j \in \{1, \ldots, n\}$, Π_j is an $\emptyset$-set or a $\forall^3$-set, and K satisfies the condition (d.2).

Let there exist $i \in \{1, \ldots, n\}$ such that Π_i is a $\forall\exists$-set. Then, evidently, for any $j \in \{1, \ldots, n\}$, Π_j is an $\emptyset$-set or a $\forall\exists$-set, and K satisfies the condition (d.3).

Let σ be a finite signature. Let the class K satisfy at least one of the conditions (d.1), (d.2), (d.3), or (d.4). We add to σ an infinite set of 1-ary predicate symbols. As a result, we obtain a signature σ_1. It is clear that the class $K(\sigma_1) = \Phi(\Pi_1, \sigma_1) \wedge \cdots \wedge \Phi(\Pi_{n+1}, \sigma_1)$ satisfies at least one of the conditions (d.1), (d.2), (d.3), or (d.4). According to what was proven above, the class $K(\sigma_1)$ is decidable. Since $K \subseteq K(\sigma_1)$, the class K is decidable.

Let σ contain at least $r_0 + 1$ predicate symbols. Let K be decidable. If σ contains only 1-ary predicate symbols, then K satisfies the condition (d.1). Let σ contain at least one predicate symbol, wherein the arity is greater than or equal to two. Taking into account that $\Pi_{n+1} = P(\exists^*)$ and using Theorem 21 [9], we obtain that, for any $i \in \{1, \ldots, n\}$, $\forall\exists\forall \notin \Pi_i$ and $\forall^3\exists \notin \Pi_i$. Therefore, for $i = 1, \ldots, n$, $\Pi_i \subseteq P_1 \cup P_2$.

Taking into account that $\Pi_{n+1} = P(\exists^*)$ and using Theorem 21 [9], we obtain that there are no $i, j \in \{1, \ldots, n\}$ such that $i \neq j$, $\forall\exists \in \Pi_i$ and $\forall^3 \in \Pi_j$.

Next, using arguments similar to the case of infinite signature σ, where we considered $\emptyset$-sets, $\forall^3$-sets, $\forall\exists$-sets, and $\forall^3$-$\forall\exists$-sets, we obtain that K satisfies condition (d.2), condition (d.3), or condition (d.4). □

Remark 1. *If the signature σ contains less than $r_0 + 1$ predicate symbols and K satisfies at least one of the conditions (d.1), (d.2), (d.3), or (d.4), then K is decidable—see the proof of the theorem.*

6.2. Signatures with Function Symbols. Equality Is Not Allowed

In this section, we consider a finite or countable signature σ that does not contain nullary predicate and function symbols and contains a function symbol of arity greater than 0. The equality $=$ cannot be used in the formulas of the signature σ.

Theorem 3. *Let σ be a signature with at least one function symbol of arity greater than 0. Let $\Pi_1, \ldots, \Pi_n$ be nonempty sets of words in the alphabet $\{\forall, \exists\}$ closed relative to subwords, $\Pi_{n+1} = P(\exists^*)$, and $K = \Phi(\Pi_1, \sigma) \wedge \cdots \wedge \Phi(\Pi_n, \sigma) \wedge \Phi(\Pi_{n+1}, \sigma)$.*

(a) *Either the class K is decidable or it is a class of reduction.*

(b) *If the class K is decidable, then for sentences from K, the satisfiability coincides with the finite satisfiability.*

(c) *The class K is decidable if and only if at least one of the following three conditions is satisfied:*

(c.1) *The signature σ does not contain predicate symbols.*
(c.2) *The signature σ contains only* 1*-ary predicate and* 1*-ary function symbols.*
(c.3) *For $i = 1, \ldots, n$, $\Pi_i \subseteq P(\exists^*\forall\exists^*)$.*

Proof. Let K satisfy the condition (c.1). Then, $K = \emptyset$ and, consequently, K is decidable. Since $K = \emptyset$, the statement (b) holds for K.

Let K satisfy the condition (c.2). By Theorem 7 [10], the class $\Phi(P, \sigma)$, where P is the set of all words in the alphabet $\{\forall, \exists\}$, is decidable. Evidently, there exists an algorithm that transforms any sentence $\alpha \in K$ to a sentence $\alpha' \in \Phi(P, \sigma)$ such that α' is satisfiable if and only if α is satisfiable, and α' is finite satisfiable if and only if α is finite satisfiable. Therefore, K is decidable. By Theorem 7 [10], the statement (b) holds for $\Phi(P, \sigma)$. Therefore, the statement (b) holds for K.

Let K satisfy the condition (c.3). Let $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_{n+1} \in K$. By introducing some new nullary function symbols $c_1, \ldots, c_k$ and changing variables, we can transform the sentence α to the form $\beta = \forall x_0 \exists \ldots \exists u_1 \wedge \cdots \wedge \forall x_0 \exists \ldots \exists u_{n+1}$ such that $u_1, \ldots, u_{n+1}$ are quantifier-free formulas of the signature $\sigma' = \sigma \cup \{c_1, \ldots, c_k\}$ with pairwise disjoint sets of variables (except x_0), and for any structure $\mathcal{U}$ of the signature σ, α is true in $\mathcal{U}$ if and only if β is true in $\mathcal{U}$ for some interpretation of new symbols $c_1, \ldots, c_k$ in the universe of $\mathcal{U}$. Denote $\gamma = \forall x_0(\exists \ldots \exists u_1 \wedge \cdots \wedge \exists \ldots \exists u_{n+1})$. Evidently, for any structure $\mathcal{U}'$ of the signature σ', γ is true in $\mathcal{U}'$ if and only if β is true in $\mathcal{U}'$. Furthermore, we can transform the sentence γ to the form $\delta = \forall x_0 \exists \ldots \exists u$ such that u is a quantifier-free formula of the signature σ', and for any structure $\mathcal{U}'$ of the signature σ', δ is true in $\mathcal{U}'$ if and only if γ is true in $\mathcal{U}'$.

Let $y_1, \ldots, y_k$ be variables from X that do not belong to δ. Denote $\varepsilon = \exists y_1 \ldots \exists y_k \forall x_0 \exists \ldots \exists v$, where v is obtained from u by replacing symbols $c_1, \ldots, c_k$ with variables $y_1, \ldots, y_k$, respectively. One can show that, for any structure $\mathcal{U}$ of the signature σ, ε is true in $\mathcal{U}$ if and only if δ is true in $\mathcal{U}$ for some interpretation of new symbols $c_1, \ldots, c_k$ in the universe of $\mathcal{U}$. Therefore, for any structure $\mathcal{U}$ of the signature σ, ε is true in $\mathcal{U}$ if and only if α is true in $\mathcal{U}$. It is clear that $\varepsilon \in \Phi(P(\exists^*\forall\exists^*), \sigma)$. Using Theorem 7 [10], we obtain that the class $\Phi(P(\exists^*\forall\exists^*), \sigma)$ is decidable. Therefore, the class K is decidable. By Theorem 7 [10], the statement (b) holds for $\Phi(P(\exists^*\forall\exists^*), \sigma)$. Therefore, the statement (b) holds for K.

Let K not satisfy any of the conditions (c.1), (c.2), or (c.3). Then, σ contains a predicate symbol p and a function symbol f such that, for at least one of these symbols, its arity is at least 2, and there exists $i \in \{1, \ldots, n\}$ such that $\Pi_i \not\subseteq P(\exists^*\forall\exists^*)$. Let, for definiteness, $i = 1$. Since $\Pi_1 \not\subseteq P(\exists^*\forall\exists^*)$, the word $\forall^2$ belongs to Π_1.

We now show that, for $i = 2, \ldots, n+1$, the set $\Phi(\Pi_i, \sigma)$ contains a sentence that is true in any structure $\mathcal{U}$ of the signature σ. Let $i \in \{2, \ldots, n+1\}$ and $Q_0 \ldots Q_t \in \Pi_i$, where $Q_0, \ldots, Q_t \in \{\exists, \forall\}$. Then, the sentence $\pi_i = Q_0 x_0 \ldots Q_t x_t (p(x_0, \ldots, x_0) \vee \neg p(x_0, \ldots, x_0))$ is true in any structure $\mathcal{U}$ of the signature σ and $\pi_i \in \Phi(\Pi_i, \sigma)$.

Let the arity of the symbol p be greater than or equal to two. From Theorem 7 [10], it follows that the class $\Phi(P(\forall^2), \sigma_1)$, where σ_1 consists of the 2-ary predicate symbol ρ and the 1-ary function symbol φ, is a class of reduction. Let us show that there is an algorithm which, for an arbitrary sentence $\alpha \in \Phi(P(\forall^2), \sigma_1)$, constructs a sentence $\alpha' \in K$ such that α' is satisfiable if and only if α is satisfiable, and α' is finite satisfiable if and only if α is finite satisfiable.

Let us replace in α every occurrence of $\rho(t_1, t_2)$ by $p(t_1, t_2, \ldots, t_2)$ and every occurrence of $\varphi(t)$ by $f(t, t, \ldots, t)$, where t_1, t_2, and t are arbitrary terms of the signature σ. Denote by β the obtained formula. Then, as α', we can take the sentence $\beta \wedge \pi_2 \wedge \cdots \wedge \pi_{n+1}$, which belongs to K. Since $\Phi(P(\forall^2), \sigma_1)$ is a class of reduction, K is also a class of reduction.

Let the arity of the symbol f be greater than or equal to two. From Theorem 7 [10], it follows that the class $\Phi(P(\forall^2), \sigma_2)$, where σ_2 consists of the 1-ary predicate symbol ρ and the 2-ary function symbol φ, is a class of reduction. Let us show that there is an algorithm which, for an arbitrary sentence $\alpha \in \Phi(P(\forall^2), \sigma_2)$, constructs a sentence $\alpha' \in K$ such that α' is satisfiable if and only if α is satisfiable, and α' is finite satisfiable if and only if α is finite satisfiable.

Let us replace in α every occurrence of $\rho(t)$ by $p(t, t, \ldots, t)$ and every occurrence of $\varphi(t_1, t_2)$ by $f(t_1, t_2, \ldots, t_2)$, where t, t_1, and t_2 are arbitrary terms of the signature σ. Denote by β the obtained formula. Then, as α', we can take the sentence $\beta \wedge \pi_2 \wedge \cdots \wedge \pi_{n+1}$, which belongs to K. Since $\Phi(P(\forall^2), \sigma_2)$ is a class of reduction, K is also a class of reduction. This completes the proofs of statements (a), (b), and (c). □

6.3. Signatures with Function Symbols. Equality Is Allowed

Theorem 4. *Let σ be a signature with at least one function symbol. Let $\Pi_1, \ldots, \Pi_n$ be nonempty sets of words in the alphabet $\{\forall, \exists\}$ closed relative to subwords, $\Pi_{n+1} = P(\exists^*)$, and $K^= = \Phi^=(\Pi_1, \sigma) \wedge \cdots \wedge \Phi^=(\Pi_n, \sigma) \wedge \Phi^=(\Pi_{n+1}, \sigma)$.*

(a) *Either the class $K^=$ is decidable or it is a class of reduction.*

(b) *The class $K^=$ is decidable if and only if at least one of the following three conditions is satisfied:*

(b.1) *For $i = 1, \ldots, n$, $\Pi_i \subseteq P(\exists^*)$.*

(b.2) *The signature σ contains only 1-ary predicate symbols, at most one 1-ary function symbol, and does not contain function symbols with arity greater than one.*

(b.3) *For $i = 1, \ldots, n$, $\Pi_i \subseteq P(\exists^*\forall\exists^*)$, and the signature σ contains at most one 1-arity function symbol and does not contain function symbols with arity greater than one. There are no any restrictions regarding predicate symbols.*

Proof. Let $K^=$ satisfy the condition (b.1). Evidently, there is an algorithm which transforms an arbitrary sentence $\alpha \in K^=$ into a sentence $\alpha' \in \Phi^=(P(\exists^*), \sigma)$ such that α is satisfiable if and only if α' is satisfiable, and α is finite satisfiable if and only if α' is finite satisfiable. Using the main theorem in [11], we obtain that the class $\Phi^=(P(\exists^*), \sigma)$ is decidable. Therefore, the class $K^=$ is decidable.

Let $K^=$ satisfy the condition (b.2). From the main theorem in [11], it follows that the class $\Phi^=(P, \sigma)$ is decidable, where P is the set of all words in the alphabet $\{\exists, \forall\}$. Taking into account that there is an algorithm that converts any sentence from $K^=$ to prenex form, we obtain that the class $K^=$ is decidable.

Let $K^=$ satisfy the condition (b.3). Now, we will almost repeat the reasoning from the proof of Theorem 3. Let $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_{n+1} \in K^=$. By introducing some new nullary function symbols $c_1, \ldots, c_k$ and changing variables, we can transform the sentence α to the form $\beta = \forall x_0 \exists \ldots \exists u_1 \wedge \cdots \wedge \forall x_0 \exists \ldots \exists u_{n+1}$ such that $u_1, \ldots, u_{n+1}$ are quantifier-free formulas of the signature $\sigma' = \sigma \cup \{c_1, \ldots, c_k\}$ with pairwise disjoint sets of variables (except x_0), and for any structure U of the signature σ, α is true in U if and only if β is true in U for some interpretation of new symbols $c_1, \ldots, c_k$ in the universe of U. Denote $\gamma = \forall x_0(\exists \ldots \exists u_1 \wedge \cdots \wedge \exists \ldots \exists u_{n+1})$. Evidently, for any structure U' of the signature σ', γ is true in U' if and only if β is true in U'. Furthermore, we can transform the sentence γ to the form $\delta = \forall x_0 \exists \ldots \exists u$ such that u is a quantifier-free formula of the signature σ', and for any structure U' of the signature σ', δ is true in U' if and only if γ is true in U'.

Let $y_1, \ldots, y_k$ be variables from X that do not belong to δ. Denote $\varepsilon = \exists y_1 \ldots \exists y_k \forall x_0 \exists \ldots \exists v$, where v is obtained from u by replacing symbols $c_1, \ldots, c_k$ with variables $y_1, \ldots, y_k$, respectively. One can show that, for any structure U of the signature σ, ε is true in U if and only if δ is true in U for some interpretation of new symbols $c_1, \ldots, c_k$ in the universe of U. Therefore, for any structure U of the signature σ, ε is true in U if and only if α is true in

U. It is clear that $\varepsilon \in \Phi^{=}(P(\exists^*\forall\exists^*), \sigma)$. Using the main theorem in [11], we obtain that the class $\Phi^{=}(P(\exists^*\forall\exists^*), \sigma)$ is decidable. Therefore, the class $K^{=}$ is decidable.

Let $K^{=}$ not satisfy any of the conditions (b.1), (b.2), or (b.3). One can show that in this case $K^{=}$ satisfy at least one of the following three conditions:

(c.1) There exists $i \in \{1, \ldots, n\}$ such that $\Pi_i \not\subseteq P(\exists^*)$, i.e., $\forall \in \Pi_i$, and σ contains two 1-ary function symbols.

(c.2) There exists $i \in \{1, \ldots, n\}$ such that $\Pi_i \not\subseteq P(\exists^*)$, i.e., $\forall \in \Pi_i$, and σ contains a function symbol with arity greater than one.

(c.3) There exists $i \in \{1, \ldots, n\}$ such that $\Pi_i \not\subseteq P(\exists^*\forall\exists^*)$, i.e., $\forall^2 \in \Pi_i$, and σ contains a predicate symbol with arity greater than one and a function symbol.

Denote by σ_1 the signature consisting of two 1-ary function symbols, σ_2 the signature consisting of one 2-ary function symbol, and σ_3 the signature consisting of one 2-ary predicate symbol and one 1-ary function symbol. From the main theorem in [11] (see also Theorem 4.0.1 [8]), it follows that the classes $\Phi^{=}(P(\forall), \sigma_1)$, $\Phi^{=}(P(\forall), \sigma_2)$, and $\Phi^{=}(P(\forall^2), \sigma_3)$ are classes of reduction.

If the class $K^{=}$ satisfies the condition (c.1), then $\Phi^{=}(P(\forall), \sigma_1)$ can be reduced to $K^{=}$, and $K^{=}$ is a class of reduction. If the class $K^{=}$ satisfies the condition (c.2), then $\Phi^{=}(P(\forall), \sigma_2)$ can be reduced to $K^{=}$, and $K^{=}$ is a class of reduction. If the class $K^{=}$ satisfies the condition (c.3), then $\Phi^{=}(P(\forall^2), \sigma_3)$ can be reduced to $K^{=}$, and $K^{=}$ is a class of reduction. We will not go into details of the proof—similar statements were considered in the proof of Theorem 3. □

7. Problem of Optimization

This section considers the problem of optimization of schemes of computation trees and studies how their decidability depends on the decidability of the problems of solvability and satisfiability. It is proven that, for any strictly limited complexity measure, the problem of optimization is undecidable if the problem of satisfiability is undecidable. It is also proven that the problem of optimization is decidable if the problem of solvability is decidable and the considered strictly limited complexity measure satisfies some additional condition. Note that the problem of solvability and its corresponding problem of satisfiability are either both decidable or both undecidable.

7.1. Equality Is Not Allowed

First, we consider the case when the equality is not allowed.

Let σ be a finite or countable signature. If σ is finite, then we represent it in the form $\sigma = \{q_0, \ldots, q_m\}$, where $q_0, \ldots, q_m$ are predicate and function symbols, each with its own arity. If σ is infinite, then we represent it in the form $\sigma = \{q_0, q_1, \ldots\}$. We denote by σ^* the set of finite words in the alphabet σ, including the empty word λ.

Definition 17. *A complexity measure over the signature σ is an arbitrary map $\psi : \sigma^* \to \omega$. The complexity measure ψ will be called strictly limited if it is computable and, for any $\alpha_1, \alpha_2, \alpha_3 \in \sigma^*$, it possesses the following property: if $\alpha_2 \neq \lambda$, then $\psi(\alpha_1\alpha_2\alpha_3) > \psi(\alpha_1\alpha_3)$.*

We now consider some examples of complexity measures. Let $\psi : \sigma \to \omega \setminus \{0\}$. The function ψ is called a weight function for the signature σ. We extend the function ψ to the set σ^* in the following way: $\psi(\alpha) = 0$ if $\alpha = \lambda$ and $\psi(\alpha) = \sum_{j=1}^{m} \psi(q_{i_j})$ if $\alpha = q_{i_1} \cdots q_{i_m}$. This function is called a weighted depth. If $\psi(q_i) = 1$ for any $q_i \in \sigma$, then the function ψ is called the depth and is denoted by h. The depth and any computable weighted depth are strictly limited complexity measures.

Let ψ be a complexity measure over the signature σ. We extend it to the sets Probl(σ) and Tree(σ). Let β be a finite sequence of function and predicate expressions

of the signature σ that do not contain the equality. We correspond to β a word word(β) from σ^*. If the length of β is equal to 0, then word$(\beta) = \lambda$. If $\beta = \beta_1, \ldots, \beta_m$, then word$(\beta) = q_{i_1} \cdots q_{i_m}$, where, for $j = 1, \ldots, m$, q_{i_j} is the symbol of the signature σ from the expression β_j.

Let $s = (n, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem from the set Probl(σ). Then, $\psi(s) = \psi(\text{word}(\beta_1, \ldots, \beta_m))$.

Let $S = (n, G)$ be a scheme of computation tree from the set Tree(σ) and $\tau = w_1, d_1, w_2, d_2, \ldots, w_m, d_m, w_{m+1}$ be a complete path of the scheme S. We denote $\beta_\tau = \beta_1, \ldots, \beta_m$ as the sequence of function and predicate expressions attached to nodes $w_1, \ldots, w_m$. Then, $\psi(S) = \max\{\psi(\text{word}(\beta_\tau)) : \tau \in \text{Path}(S)\}$, where Path$(S)$ is the set of complete paths of the scheme S. The value $\psi(S)$ will be called the ψ-complexity of the scheme of computation tree S. We denote by $h(S)$ the depth of the scheme of computation tree S. By $\sigma(S)$, we denote the set of symbols of the signature σ used in the function and predicate expressions in the scheme S.

Lemma 1. *Let ψ be a strictly limited complexity measure over the signature σ and S be a scheme of computation tree from the set* Tree(σ). *Then, the following statements hold:*

(a) $\psi(S) \geq h(S)$.

(b) $\psi(S) \geq \max\{\psi(q_j) : q_j \in \sigma(S)\}$.

Proof. Let $\alpha \in \sigma^*$. Then, it is easy to show that $\psi(\alpha) \geq |\alpha|$, where $|\alpha|$ is the length of the word α, and $\psi(\alpha) \geq \psi(q_j)$ for any letter q_j in the word α. Using these relations, one can show that the statements of the lemma hold. □

Let us recall that $H(\sigma)$ is the set of sentences of the signature σ that do not contain equality. Let ψ be a strictly limited complexity measure over the signature σ, C be a nonempty class of structures of the signature σ, and H be a nonempty subset of the set $H(\sigma)$.

Definition 18. *We now define the problem of optimization for the triple (ψ, H, C): for arbitrary sentence $\alpha \in H$ and scheme of problem $s \in$* Probl(σ), *we should find a scheme of computation tree $S \in$* Tree(σ) *which solves the scheme of problem s relative to the class $C(\alpha)$ and has the minimum ψ-complexity. We will call such a scheme of computation tree optimal relative to ψ, s, and $C(\alpha)$.*

Lemma 2. *The ψ-complexity of a scheme of computation tree that is optimal relative to ψ, s, and $C(\alpha)$ is at most $\psi(s)$.*

Proof. Let $s = (n, \nu, \beta_1, \ldots, \beta_m)$. It is easy to construct a scheme of computation tree $S \in$ Tree(σ), which solves the scheme of problem s relative to the class $C(\alpha)$ and for which, for any complete path τ of S, the sequence β_τ of predicate and function expressions attached to the nodes of τ coincides with $\beta_1, \ldots, \beta_m$. Therefore, $\psi(S) = \psi(s)$. Thus, the ψ-complexity of a scheme of computation tree that is optimal relative to ψ, s, and $C(\alpha)$ is at most $\psi(s)$. □

Let $S_1 = (n, G_1)$ and $S_2 = (n, G_2)$ be schemes of computation trees from the set Tree(σ). We will say that these schemes are equivalent if, for any structure U of the signature σ, the functions implemented by the computation trees (S_1, U) and (S_2, U) coincide.

Lemma 3. *Any scheme of computation tree $S_1 = (n, G_1) \in$* Tree(σ) *can be transformed by changing the variables into a scheme of computation tree $S_2 = (n, G_2) \in$* Tree(σ), *which is equivalent to S_1 and in which all variables in the function and predicate expressions belong to the set $X_{n+2^{h(S_1)}}$.*

Proof. One can show that the number of function nodes in the scheme S_1 is at most $2^{h(S_1)}$. Each function node v of the scheme S_1 is labeled with a function expression

$$x_{i(v)} \Leftarrow q_{j(v)}(x_{l(1,v)}, \ldots, x_{l(m(v),v)}),$$

where $m(v)$ is the arity of the function symbol $q_{j(v)}$. Let $v_1, \ldots, v_k$ be all function nodes of the scheme S_1 for which variables $x_{i(v_1)}, \ldots, x_{i(v_k)}$ do not belong to the set X_n. Let $x_{j_1}, \ldots, x_{j_p}$ be all pairwise different variables in the sequence $x_{i(v_1)}, \ldots, x_{i(v_k)}$. Denote $Y = \{x_{j_1}, \ldots, x_{j_p}\}$.

In all expressions attached to function and predicate nodes of the scheme S_1, we replace each variable that does not belong to the set $X_n \cup Y$ with the variable x_{n-1} and replace variables $x_{j_1}, \ldots, x_{j_p}$ with variables $x_n, \ldots, x_{n+p-1}$, respectively. Denote by $S_2 = (n, G_2)$ the obtained scheme of computation tree. One can show that the scheme S_2 is equivalent to the scheme S_1, and all variables in the function and predicate expressions of the scheme S_2 belong to the set $X_{n+2^{h(S_1)}}$. □

Let ψ be a strictly limited complexity measure over the signature σ. For $i \in \omega$, we denote $\omega_\psi(i) = \{q_j : q_j \in \sigma, \psi(q_j) = i\}$. Define a partial function $K_\psi : \omega \to \omega$ as follows. Let $i \in \omega$. If $\omega_\psi(i)$ is a finite set, then $K_\psi(i) = |\omega_\psi(i)|$. If $\omega_\psi(i)$ is an infinite set, then the value $K_\psi(i)$ is indefinite.

For us, the most interesting situation is when the function K_ψ is a total recursive function. If σ is a finite signature, then, evidently, K_ψ is a total recursive function.

We now consider a class of strictly limited complexity measures ψ over the infinite signature σ for which K_ψ is a total recursive function. Let $\sigma = \{q_j : j \in \omega\}$, $g : \omega \to \omega \setminus \{0\}$ be a nondecreasing unbounded total recursive function, and ψ be a weighted depth over signature σ for which $\psi(q_j) = g(j)$ for any $j \in \omega$. Then, ψ is a strictly limited complexity measure over the signature σ for which K_ψ is a total recursive function.

Theorem 5. *Let ψ be a strictly limited complexity measure over the signature σ for which K_ψ is a total recursive function, C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H(\sigma)$, and the problem of solvability for the quadruple $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), H, C)$ be decidable. Then, the problem of optimization for the triple (ψ, H, C) is decidable.*

Proof. Taking into account that the function K_ψ is a total recursive function, it is not difficult to show that there exists an algorithm constructing the set $\{q_j : q_j \in \sigma, \psi(q_j) \leq r\}$ for any number $r \in \omega$. From this fact, it follows that there exists an algorithm which, for an arbitrary number $r \in \omega$, an arbitrary number $n \in \omega \setminus \{0\}$, and an arbitrary finite nonempty subset M of the set ω, constructs the set $\mathrm{Tree}(\sigma, r, n, M)$ of schemes of computation trees $S = (n, G)$ from $\mathrm{Tree}(\sigma)$ satisfying the following conditions:

- The terminal nodes of S are labeled with numbers from M.
- $h(S) \leq r$.
- $\max\{\psi(q_j) : q_j \in \sigma(S)\} \leq r$.
- All variables in the function and predicate expressions in the scheme S belong to the set X_{n+2^r}.

Let $s = (n, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem from $\mathrm{Probl}(\sigma)$, and let $M(s)$ be the set of values of the map ν. Denote $\mathrm{Tree}(s) = \mathrm{Tree}(\sigma, \psi(s), n, M(s))$. We now shall see that the set $\mathrm{Tree}(s)$ contains a scheme of computation tree that is optimal relative to ψ, s, and $C(\alpha)$.

Using Lemma 3, one can show that there exists a scheme of computation tree $S \in \mathrm{Tree}(\sigma)$ which is optimal relative to ψ, s, and $C(\alpha)$, in which numbers attached to terminal nodes belong to the set $M(s)$, and in which all variables in function and predicate expressions in the scheme S belong to the set $X_{n+2^{h(S)}}$. Using Lemma 2, we obtain that $\psi(S) \leq \psi(s)$. From this inequality and Lemma 1, it follows that $h(S) \leq \psi(s)$ and

$\max\{\psi(q_j) : q_j \in \sigma(S)\} \leq \psi(s)$. Therefore, the scheme S, which is optimal relative to ψ, s, and $C(\alpha)$, belongs to the set $\mathrm{Tree}(s)$.

We now describe an algorithm solving the problem of optimization for the triple (ψ, H, C). Let $\alpha \in H$ and $s = (n, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem from the set $\mathrm{Probl}(\sigma)$. First, we compute the value $\psi(s)$ and construct the set $M(s)$. Next, we construct the set $\mathrm{Tree}(s) = \mathrm{Tree}(\sigma, \psi(s), n, M(s))$. Using the algorithm solving the problem of solvability for the quadruple $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), H, C)$, we can find a scheme of computation tree $S \in \mathrm{Tree}(s)$, which solves the scheme of problem s relative to the class $C(\alpha)$ and has the minimum ψ-complexity among such schemes of computation trees. The scheme of computation tree S is optimal relative to ψ, s, and $C(\alpha)$. □

Corollary 1. *Let σ be a finite signature, ψ be a strictly limited complexity measure over the signature σ, C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H(\sigma)$, and the problem of solvability for the quadruple $(\mathrm{Probl}(\sigma), \mathrm{Tree}(\sigma), H, C)$ be decidable. Then, the problem of optimization for the triple (ψ, H, C) is decidable.*

Theorem 6. *Let C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H(\sigma)$, ψ be a strictly limited complexity measure over the signature σ, and the problem of satisfiability for the pair $(H \wedge \Phi(P(\exists^*), \sigma), C)$ be undecidable. Then, the problem of optimization for the triple (ψ, H, C) is undecidable.*

Proof. Let us assume the contrary: the problem of optimization for the triple (ψ, H, C) is decidable. We now describe an algorithm for solving the problem of satisfiability for the pair $(H \wedge \Phi(P(\exists^*), \sigma), C)$.

Let $\alpha \in H$ and $\gamma \in \Phi(P(\exists^*), \sigma)$. We construct a sentence of the signature σ that is logically equivalent to the sentence γ and is of the form $\exists x_0 \ldots \exists x_{n-1}((\gamma_{11}^{\delta_{11}} \wedge \cdots \wedge \gamma_{1m_1}^{\delta_{1m_1}}) \vee \cdots \vee (\gamma_{k1}^{\delta_{k1}} \wedge \cdots \wedge \gamma_{km_k}^{\delta_{km_k}}))$, where, for $j = 1, \ldots, k$ and $i = 1, \ldots, m_j$, $\delta_{ji} \in E_2$ and γ_{ji} define an atomic formula of the signature σ of the form $q_l(t_1, \ldots, t_p)$, where q_l is a p-ary predicate symbol from σ, and $t_1, \ldots, t_p$ are terms of the signature σ with variables from X_n.

For $j = 1, \ldots, k$, we construct a scheme of problem s_j from $\mathrm{Probl}(\sigma)$ with special representation $(n, \nu_j, \gamma_{j1}, \ldots, \gamma_{jm_j})$ such that $\nu_j : E_2^{m_j} \to E_2$ and, for any $\bar{\delta} \in E_2^{m_j}$, $\nu_j(\bar{\delta}) = 1$ if and only if $\bar{\delta} = (\delta_{j1}, \ldots, \delta_{jm_j})$. It is clear that, for any structure U of the signature σ and any tuple $\bar{a} \in A^n$, where A is the universe of U,

$$\varphi_{(s_j,U)}(\bar{a}) = \begin{cases} 0, & \text{if } U \models \neg(\gamma_{j1}(\bar{a})^{\delta_{j1}} \wedge \cdots \wedge \gamma_{jm_j}(\bar{a})^{\delta_{jm_j}}), \\ 1, & \text{if } U \models \gamma_{j1}(\bar{a})^{\delta_{j1}} \wedge \cdots \wedge \gamma_{jm_j}(\bar{a})^{\delta_{jm_j}}. \end{cases}$$

We denote by S_0 the scheme of computation tree from $\mathrm{Tree}(\sigma)$, which consists of only one node labeled with the number 0. It is clear that, for any structure U of the signature σ and any tuple $\bar{a} \in A^n$, where A is the universe of U, $\varphi_{(S_0,U)}(\bar{a}) = 0$.

Using an algorithm that solves the problem of optimization for the triple (ψ, H, C), for $j = 1, \ldots, k$, we find a scheme of computation tree $S_j \in \mathrm{Tree}(\sigma)$ which solves the scheme of problem s_j relative to the class $C(\alpha)$ and has the minimum ψ-complexity.

Using the properties of the strictly limited complexity measure ψ, it is not difficult to show that a structure $U \in C$ such that $U \models \alpha \wedge \gamma$ exists if and only if there exists $j \in \{1, \ldots, k\}$ for which the scheme of computation tree S_j does not coincide with the scheme of computation tree S_0. Therefore, the problem of satisfiability for the pair $(H \wedge \Phi(P(\exists^*), \sigma), C)$ is decidable, but this is impossible. □

7.2. Equality Is Allowed

We now consider the case when the equality is allowed.

Let σ be a finite or countable signature. If σ is finite, then we represent it in the form $\sigma = \{q_1, \ldots, q_m\}$. If σ is infinite, then we represent it in the form $\sigma = \{q_1, q_2, \ldots\}$. Let $\sigma_= = \sigma \cup \{q_0\}$, where q_0 is the symbol denoting equality $=$, and $\sigma_=^*$ is the set of finite words in the alphabet $\sigma_=$, including the empty word λ.

Definition 19. *An e-complexity measure over the signature σ is an arbitrary map $\psi : \sigma_=^* \to \omega$. The e-complexity measure ψ will be called strictly limited if it is computable and, for any $\alpha_1, \alpha_2, \alpha_3 \in \sigma_=^*$, it possesses the following property: if $\alpha_2 \neq \lambda$, then $\psi(\alpha_1\alpha_2\alpha_3) > \psi(\alpha_1\alpha_3)$.*

The prefix "e-" here and later indicates the presence of the equality.

Let ψ be an e-complexity measure over the signature σ. We extend it to the sets $\mathrm{Probl}^=(\sigma)$ and $\mathrm{Tree}^=(\sigma)$. Let β be a finite sequence of the function and predicate expressions of the signature σ. We correspond to β a word $\mathrm{word}(\beta)$ from $\sigma_=^*$. If the length of β is equal to 0, then $\mathrm{word}(\beta) = \lambda$. If $\beta = \beta_1, \ldots, \beta_m$, then $\mathrm{word}(\beta) = q_{i_1} \cdots q_{i_m}$, where, for $j = 1, \ldots, m$, q_{i_j} is the symbol from $\sigma_=$ contained in the expression β_j. In particular, if β_j has the form $x_{l_1} = x_{l_2}$, then $q_{i_j} = q_0$.

Let $s = (n, \nu, \beta_1, \ldots, \beta_m)$ be a scheme of problem from the set $\mathrm{Probl}^=(\sigma)$. Then, $\psi(s) = \psi(\mathrm{word}(\beta_1, \ldots, \beta_m))$.

Let $S = (n, G)$ be a scheme of computation tree from the set $\mathrm{Tree}^=(\sigma)$ and $\tau = w_1, d_1, w_2, d_2, \ldots, w_m, d_m, w_{m+1}$ be a complete path of the scheme S. We denote $\beta_\tau = \beta_1, \ldots, \beta_m$ as the sequence of function and predicate expressions attached to nodes $w_1, \ldots, w_m$. Then, $\psi(S) = \max\{\psi(\mathrm{word}(\beta_\tau)) : \tau \in \mathrm{Path}(S)\}$, where $\mathrm{Path}(S)$ is the set of complete paths of the scheme S. The value $\psi(S)$ is called the ψ-complexity of the scheme of computation tree S.

Let us recall that $H^=(\sigma)$ is the set of sentences of the signature σ. Let ψ be a strictly limited e-complexity measure over the signature σ, C be a nonempty class of structures of the signature σ, and H be a nonempty subset of the set $H^=(\sigma)$.

Definition 20. *We now define the problem of e-optimization for the triple (ψ, H, C): for arbitrary sentence $\alpha \in H$ and scheme of problem $s \in \mathrm{Probl}^=(\sigma)$, we should find a scheme of computation tree $S \in \mathrm{Tree}^=(\sigma)$ which solves the scheme of problem s relative to the class $C(\alpha)$ and has the minimum ψ-complexity.*

Let ψ be a strictly limited e-complexity measure over the signature σ. For $i \in \omega$, we denote $\omega_\psi(i) = \{q_j : q_j \in \sigma_=, \psi(q_j) = i\}$. Define a partial function $K_\psi : \omega \to \omega$ as follows. Let $i \in \omega$. If $\omega_\psi(i)$ is a finite set, then $K_\psi(i) = |\omega_\psi(i)|$. If $\omega_\psi(i)$ is an infinite set, then the value $K_\psi(i)$ is indefinite.

The proof of the next statement is similar to the proof of Theorem 5.

Theorem 7. *Let ψ be a strictly limited e-complexity measure over the signature σ for which K_ψ is a total recursive function, C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H^=(\sigma)$, and the problem of solvability for the quadruple $(\mathrm{Probl}^=(\sigma), \mathrm{Tree}^=(\sigma), H, C)$ be decidable. Then, the problem of e-optimization for the triple (ψ, H, C) is decidable.*

Corollary 2. *Let σ be a finite signature, ψ be a strictly limited e-complexity measure over the signature σ, C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H^=(\sigma)$, and the problem of solvability for the quadruple $(\mathrm{Probl}^=(\sigma), \mathrm{Tree}^=(\sigma), H, C)$ be decidable. Then, the problem of e-optimization for the triple (ψ, H, C) is decidable.*

The proof of the next statement is similar to the proof of Theorem 6.

Theorem 8. *Let C be a nonempty class of structures of the signature σ, H be a nonempty subset of the set $H^{=}(\sigma)$, ψ be a strictly limited e-complexity measure over the signature σ, and the problem of satisfiability for the pair $(H \wedge \Phi^{=}(P(\exists^*), \sigma), C)$ be undecidable. Then, the problem of e-optimization for the triple (ψ, H, C) is undecidable.*

8. Conclusions

This paper studied relationships among three algorithmic problems involving computation trees: the optimization, solvability, and satisfiability problems. It also studied the decidability of these problems in different situations.

A powerful toolkit, developed in particular in [8], was used to describe the decidability conditions of the satisfiability problem and, consequently, the decidability of the optimization problem for various classes of structures. Naturally, like any fairly universal approaches, the methods considered may not be applicable in some situations, for example, if we study a specific structure of interest to us.

This paper was limited to considering the fundamental question of the decidability of the optimization problem. To use such results in real applications, it is necessary to study the time complexity of the optimization problem. The prerequisites for this are provided by the book [8], which studies not only the decidability of the satisfiability problem but also its time complexity. One can consider such a study of the optimization problem, including the development of various heuristics for optimizing computation trees, as one of the future goals of this work.

Funding: This research was funded by the King Abdullah University of Science and Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST). The author is indebted to the anonymous reviewers for useful comments and suggestions.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Ben-Or, M. Lower bounds for algebraic computation trees (preliminary report). In Proceedings of the 15th Annual ACM Symposium on Theory of Computing, STOC 1983, Boston, MA, USA, 25–27 April 1983; pp. 80–86.
2. Gabrielov, A.; Vorobjov, N.N. On topological lower bounds for algebraic computation trees. *Found. Comput. Math.* **2017**, *17*, 61–72. [CrossRef]
3. Grigoriev, D.; Vorobjov, N.N. Complexity lower bounds for computation trees with elementary transcendental function gates. *Theor. Comput. Sci.* **1996**, *157*, 185–214. [CrossRef]
4. Moshkov, M. Rough analysis of computation trees. *Discret. Appl. Math.* **2022**, *321*, 90–108. [CrossRef]
5. Ax, J. The elementary theory of finite fields. *Ann. Math.* **1968**, *88*, 239–271. [CrossRef]
6. Ershov, Y.L.; Lavrov, I.A.; Taimanov, A.D.; Taitslin, M.A. Elementary theories. *Russ. Math. Surv.* **1965**, *20*, 35–105. [CrossRef]
7. Ershov, Y.L.; Palyutin, E.A. *Mathematical Logic*, 6th ed.; Fizmatlit: Moscow, Russia, 2011. (In Russian)
8. Börger, E.; Grädel, E.; Gurevich, Y. *The Classical Decision Problem*; Perspectives in Mathematical Logic; Springer: Berlin/Heidelberg, Germany, 1997.
9. Gurevich, Y. On the effective recognizing of satisfiability of predicate formulas. *Algebra Log.* **1966**, *5*, 25–55. (In Russian)

10. Gurevich, Y. The decision problem for logic of predicates and operations. *Algebra Log.* **1969**, *8*, 284–308. (In Russian) [CrossRef]
11. Gurevich, Y. The decision problem for standard classes. *J. Symb. Log.* **1976**, *41*, 460–464. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.