Theory of Computing Systems (2021) 65:462–478
https://doi.org/10.1007/s00224-020-09995-4

On Decidability of Theories of Regular Languages

Sergey Dudakov[1] · **Boris Karlov**[1]

Published online: 13 August 2020

Abstract
This paper is dedicated to studying decidability properties of theories of regular languages with classical operations: union, concatenation, and the Kleene star. The theory with union only is a theory of some Boolean algebra, so it is decidable. We prove that the theory of regular languages with the Kleene star only is decidable. If we use union and concatenation simultaneously, then the theory becomes both Σ_1- and Π_1-hard over the one-symbol alphabet. Using methods from the proof of this theorem we establish that the theory of regular languages over one-symbol alphabet with union and the Kleene star is as hard as arithmetic. Then we establish that the theory with all three operations is reducible to arithmetic also, hence, it is equivalent to arithmetic. Finally, we prove that the theory of regular languages over any alphabet with concatenation only is equivalent to arithmetic also. The last result is based on our previous work where an analogous theorem was proved for one-symbol languages.

Keywords Regular languages · Theory · Union · Concatenation · Kleene star · Quantifier elimination · Arithmetic · Undecidability

1 Introduction

One of the most important problems in mathematical logic is to study algorithmic decidability properties of different theories. Some classical results in this area are undecidability of Peano arithmetic, of group and semigroup theories etc. Examples of

This article belongs to the Topical Collection: *Special Issue on Computer Science Symposium in Russia (2019)*
Guest Editor: Gregory Kucherov

✉ Boris Karlov
bnkarlov@gmail.com

Sergey Dudakov
sergeydudakov@yandex.ru

[1] CS Department, Tver State University, Zhelyabova str., 33, Tver, 170100, Russia

decidable theories are Presburger arithmetic, abelian groups theory, Boolean algebras theory etc. Some more recent results are connected to word theories. In [6, 7] it was proved that the words theory over two-symbol alphabet with concatenation is essentially undecidable. In [16, 17] it was proved that a variant of Robinson arithmetic is interpretable is this theory.

In this article we study some variants of theory of regular languages. This interest is due to the importance of such languages for formal linguistic and automata theory. Also regular languages have a lot of practical applications in compiler design, text processing algorithms, and many other fields. Each regular language has good decidability properties itself and many natural problems are decidable for them: the equivalence, the membership, the emptiness, the infinity problem etc.

But what about the set of all regular languages? If we consider the set of all regular languages (over fixed alphabet) with language-wide operations as a universe, then analogous decidability questions naturally appear. With correspondence $w \leftrightarrow \{w\}$ it is easy to reduce the concatenation theory of words to the concatenation theory of languages. Hence, the last is undecidable.

In our paper we consider theory of regular languages with different sets of classical operations. Section 2 of this article contains some basic definitions. In Section 3 we study the theory T_1 of regular languages with the Kleene star operation only. We prove that T_1 admits effective quantifier elimination, hence, it is decidable. Regular languages with union only form a Boolean algebra, hence, its theory is decidable [11]. In Section 4 we consider the theory T_2 of regular languages over a one-symbol alphabet with union and concatenation. It is proved that T_2 is both Σ_1- and Π_1-hard. In Section 5 we study the theory T_3 of regular languages over a one-symbol alphabet with union and the Kleene star. We establish that elementary arithmetic can be interpreted in T_3. Also we prove that the theory T_4 of regular languages with all operations (union, concatenation, Kleene star) can be interpreted in arithmetic. Thus, the theories T_3, T_4 and arithmetic are algorithmically equivalent. The Kleene star can be expressed via union and concatenation, therefore, T_2 is equivalent to arithmetic also.

In Section 6 we study regular languages with concatenation only. In [4] it was proved that the theory T_5 of regular languages over one-symbol alphabet is equivalent to arithmetic. This result cannot be immediately generalized to arbitrary languages because its proof uses commutativity of concatenation. Instead, we interpret the theory T_{5e} (the significant part of T_5) in the theory T_6 of regular languages over multi-symbol alphabet. Therefore, T_6 is also equivalent to arithmetic.

This article is based on our previous work [5] presented in the 14th International Computer Science Symposium in Russia (CSR 2019, Novosibirsk, Russia, July 2019). After that article was accepted we have established some new results. Some of them were presented in the International Conference "Algebra and Mathematical Logic: Theory and Applications" (Kazan, Russia, June 2019). In particular, it was proved that the theory T_1 is PSPACE-complete, so a reference [9] is added in Section 3. In Section 4 we add a reference [11] that contains the solution to one of the questions left open in [5]: the theory of regular languages with union only is decidable since it is a Boolean algebra. The second established result is the

undecidability degree of theory of regular languages with concatenation only over one-symbol alphabet, and we add a reference [4].

The proof of the last result uses commutativity of concatenation. Hence, to carry this result over to any alphabet we include Section 6, which contains new results. So, we found a solution to another open problem from [5] on the undecidability degree of theory of regular languages with concatenation only. In the conclusion some new open problems are formulated.

2 Preliminaries

An *alphabet* is a finite set of symbols. A *word* over an alphabet Σ is a finite sequence of symbols from Σ. The *length* of w is denoted $|w|$. The *empty word* is a word of zero length, it is denoted ε. *Concatenation* of two words u and v is a word which is obtained by appending v is the end of u. Concatenation of words u and v is denoted $u \cdot v$ or simply uv. The i-th *power* of the word w is the word $ww \dots w$ where w is repeated i times. In particular, $w^0 = \varepsilon$ for every word w.

A *language* over an alphabet Σ is an arbitrary set of words over Σ. A *union* of two languages L_1 and L_2 is the usual union of the sets L_1 and L_2. *Concatenation* of languages L_1 and L_2 is the language $L_1 \cdot L_2 = \{ uv : u \in L_1, v \in L_2 \}$. A *Kleene star* of the language L is the language L^* consisting of concatenations of all possible sequences of words from L, i.e. $L^* = \{ w_1 w_2 \dots w_n : n \geq 0, w_i \in L \text{ for every } i \}$. Since $n = 0$ is possible L^* always contains the empty word ε. Obviously, $(L^*)^* = L^*$. L^* is finite if and only if $L = \emptyset$ or $L = \{ \varepsilon \}$, in both cases $L^* = \{ \varepsilon \}$.

The language L over the alphabet $\Sigma = \{ a_1, a_2, \dots, a_n \}$ is *regular* if it can be obtained from the languages $\emptyset$, $\{ \varepsilon \}$, $\{ a_1 \}, \dots, \{ a_n \}$ in finitely many steps with union, concatenation, and the Kleene star (for precise definitions see [1]). A *deterministic finite automaton* (DFA) is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *states*, Σ is an *input alphabet*, $q_0 \in Q$ is an *initial* state, $F \subseteq Q$ is a set of *final* states, and $\delta \colon Q \times \Sigma \to Q$ is a *transition function*. A word $w = a_1 a_2 \dots a_n$ is *accepted* by a DFA M if there exists a sequence of states $q_0, q_1, \dots, q_n$ such that $q_n \in F$, $\delta(q_i, a_{i+1}) = q_{i+1}$ for every $0 \leq i < n$. A DFA M *recognizes* a language of all words accepted by M. It is known (see [1, 8, 10]) that the language L is regular if and only if it is accepted by some DFA.

A *theory* T is a set of first-order formulas closed under logic inference. Formulas φ and ψ are *equivalent* in the theory T if $(\varphi \leftrightarrow \psi) \in T$. This is denoted $\varphi \equiv_T \psi$. A theory T admits *quantifier elimination* if for every formula φ there exists a quantifier-free formula $\psi \equiv_T \varphi$. For quantifier elimination it is enough to eliminate an existential quantifier from all formulas of kind $(\exists x)\varphi$ where φ is an elementary conjunction (see [3]). A n-ary relation P is *definable* in a theory T if there exists a formula φ such that φ does not contain P and $P(x_1, \dots, x_n) \equiv_T \varphi(x_1, \dots, x_n)$. Similarly, the n-ary function f is *definable* in the theory T if there exists a formula φ such that φ does not contain f and $f(x_1, \dots, x_n) = y \equiv_T \varphi(x_1, \dots, x_n, y)$. The *theory of a structure* $\mathfrak{A}$ is the set of all formulas which are true in $\mathfrak{A}$.

3 Decidability of Theory of Regular Languages with the Kleene Star

In this section we study the theory T_1 of regular languages over an arbitrary alphabet $\Sigma = \{a_1, \ldots, a_n\}$ with constant $\emptyset^{(0)}$ (empty language) and operation $^{*(1)}$ (the Kleene star).

Let us note some properties of the Kleene star.

Lemma 1 *1) There exists a language L_1 such that $L = L_1^*$ if and only if $L = L^*$.*
2) If $L = L^*$ and $L \neq \{\varepsilon\}$, then there exist infinitely many languages L_i such that $L_i^* = L$.
3) There exist infinitely many languages which are Kleene stars.
4) There exists an infinite family of languages L_i such that $L_i^* \neq L_i$ and $L_i^* \neq L_j^*$ for all $i \neq j$.

Proof 1) If $L_1^* = L$, then $L^* = (L_1^*)^* = L_1^* = L$.
2) Let w be an arbitrary nonempty word from L. Let $L_i = L \setminus \{w^i\}$ for $i \geq 2$. Then $L_i^* = L$, and all languages L_i are different.
3) All languages $\{a_1^i\}^*$ are different.
4) Let $L_i = \{w \in \Sigma^* : |w| \geq i\}$, $i \geq 1$. All these languages are different, $L_i \neq L_i^*$ because $\varepsilon \in L_i^* \setminus L_i$. If $i < j$, then the shortest word from L_j is longer than the shortest word form L_i. Therefore, $L_i^* \neq L_j^*$.

□

Theorem 1 *The theory T_1 admits quantifier elimination.*

Proof It is enough to eliminate a quantifier from $(\exists x)\varphi$, where the elementary conjunction φ can contain formulas of the forms $x = t$, $x^* = t$, $x = x^*$, $x \neq t$, $x^* \neq t$, and $x \neq x^*$. Throughout this proof symbols r, s, t (possibly with indices) denote terms without the variable x. We assume that φ has no terms of the form $(y^*)^*$ because $(y^*)^* = y^*$. We consider several possible cases.

Case 1 If φ is $x = t \wedge \varphi'$, then $(\exists x)(x = t \wedge \varphi') \equiv_{T_1} (\varphi')_t^x$, where $(\varphi')_t^x$ is obtained from φ by replacing x with t. Therefore, we assume in the following that φ does not contain $x = t$.

Case 2 Let φ contain both $x = x^*$ and $x^* = t$: $(\exists x)(x = x^* \wedge x^* = t \wedge \varphi')$. Then

$$(\exists x)\varphi \equiv_{T_1} (\exists x)(x = t \wedge x = x^* \wedge x^* = t \wedge \varphi') \equiv_{T_1} t = t^* \wedge (\varphi')_t^x.$$

Case 3 Let φ contain $x^* = t$ but no $x = x^*$. Then $(\exists x)\varphi$ is

$$(\exists x)(x^* = t \wedge \bigwedge_{i=1}^{k} x^* = t_i \wedge \bigwedge_{i=1}^{\ell} x \neq r_i \wedge \bigwedge_{i=1}^{m} x^* \neq s_i \wedge x \neq x^*).$$

If one of k, ℓ, or m is zero, then the corresponding conjunction is missing. The part $x \neq x^*$ may also be missing. Substituting t instead of x^* we have

$$(\exists x)\varphi \equiv_{T_1} (\exists x)(x^* = t \wedge \bigwedge_{i=1}^{p} x \neq r_i) \wedge \bigwedge_{i=1}^{k} t = t_i \wedge \bigwedge_{i=1}^{m} t \neq s_i.$$

Here $p = \ell$ if φ does not contain $x \neq x^*$, and $p = \ell + 1$, $r_{\ell+1} = t$ otherwise.

If $t^* = t$ and $t \neq \{\varepsilon\}$, then by Lemma 1 t can be written as a Kleene star of infinitely many languages. Hence, t is a Kleene star of some language other than $r_1, \ldots, r_p$, and the conditions $x \neq r_i$ may be omitted. Hence, $(\exists x)x^* = t \equiv_{T_1} t = t^*$ by Lemma 1.

If $t = \{\varepsilon\}$, then x can be only either $\emptyset$ or $\{\varepsilon\}$. Then either all languages $r_1, \ldots, r_p$ must be different from $\emptyset$, or they must be different from $\{\varepsilon\}$.

Combining both variants we obtain the following equivalence:

$$(\exists x)\varphi \equiv_{T_1} \left(t = \emptyset^* \rightarrow (\bigwedge_{i=1}^{p} r_i \neq \emptyset \vee \bigwedge_{i=1}^{p} r_i \neq \emptyset^*)\right) \wedge t = t^* \wedge \bigwedge_{i=1}^{k} t = t_i \wedge \bigwedge_{i=1}^{m} t \neq s_i.$$

Case 4 Let φ contain the subformula $x = x^*$ but no $x^* = t$: $(\exists x)\varphi$ is

$$(\exists x)(x = x^* \wedge \bigwedge_{i=1}^{k} x \neq s_i \wedge \bigwedge_{i=1}^{\ell} x^* \neq r_i).$$

Substituting x instead of x^* we obtain

$$(\exists x)\varphi \equiv_{T_1} (\exists x)(x = x^* \wedge \bigwedge_{i=1}^{k} x \neq s_i \wedge \bigwedge_{i=1}^{\ell} x \neq r_i).$$

By Lemma 1 there are infinitely many languages which are Kleene stars, hence, the last formula is true in T_1.

Case 5 Finally, let us suppose that φ contains only inequalities: $(\exists x)\varphi$ is

$$(\exists x)(\bigwedge_{i=1}^{k} x \neq r_i \wedge \bigwedge_{i=1}^{\ell} x^* \neq s_i \wedge x \neq x^*).$$

Again $x \neq x^*$ may be missing. In this case $(\exists x)\varphi$ is true also due to Lemma 1: there are infinitely many languages which are not Kleene stars.

□

Corollary 1 *The theory T_1 is decidable.*

The procedure from the proof of Theorem 1 is extremely inefficient. It requires transforming the formula into conjunctive normal form for every quantifier, resulting

in overall hyperexponential complexity, i.e. the time complexity is $\underbrace{2^{2^{\cdots^2}}}_{O(n)\text{ times}}$ where n is the length of an input formula. In [9] a more efficient algorithm was announced, and also it was announced that the theory T_1 is PSPACE-complete.

4 Undecidability of Theory of Regular Languages with Union and Concatenation

Before studying the theory of regular languages with both union and concatenation let us note that the following relations and functions are definable with union only:

- inclusion: $x \subseteq y \equiv_{T_2} x + y = y$;
- intersection: $x \cap y = z \equiv_{T_2} z \subseteq x \wedge z \subseteq y \wedge (\forall u)((u \subseteq x \wedge u \subseteq y) \to u \subseteq z)$;
- the empty language $\emptyset$: $x = \emptyset \equiv_{T_2} (\forall y) x + y = y$;
- the set of all words over alphabet $\{0\}$: $x = 0^* \equiv_{T_2} (\forall y) x + y = x$;
- complement $\bar{x}$: $\bar{x} = y \equiv_{T_2} x + y = 0^* \wedge x \cap y = \emptyset$;
- subtraction: $x \setminus y = z \equiv_{T_2} x \cap \bar{y} = z$;
- cardinality test $\mathrm{Card}_k(x)$ for every fixed natural number k. The formula $\mathrm{Card}_k(x)$ says that the language x contains exactly k words:

$$\mathrm{Card}_1(x) \equiv_{T_2} x \neq \emptyset \wedge (\forall y)(y \subseteq x \to (y = \emptyset \vee y = x));$$
$$\mathrm{Card}_{k+1}(x) \equiv_{T_2} (\exists y)(y \subseteq x \wedge \mathrm{Card}_k(x \setminus y) \wedge \mathrm{Card}_1(y)).$$

Thus, if we consider the set of regular languages with operation $+^{(2)}$ (union), then they form a Boolean algebra. The theory of each Boolean algebra is decidable (see [11]).

In this section we study the theory T_2 of regular languages over the alphabet $\{0\}$ with a constant $0^{(0)}$ (the language $\{0\}$), operations $+^{(2)}$ (union) and $\cdot^{(2)}$ (concatenation).

The constant ε denoting the language $\{\varepsilon\}$ can be defined with concatenation:

$$x = \varepsilon \equiv_{T_2} (\forall y) x \cdot y = y.$$

Let the relation $\mathrm{Mult}_k(x, y)$ denote the following: the language x contains one word 0^n for some natural number $n \neq 0$, and y contains one word 0^{kn}. This relation is definable in T_2 as follows:

$$\mathrm{Mult}_k(x, y) \equiv_{T_2} \mathrm{Card}_1(x) \wedge x \neq \varepsilon \wedge y = \underbrace{x \cdot x \cdot \cdots \cdot x}_{k\text{ times}}.$$

We prove undecidability of T_2 by reducing to it the halting problem for two-counter machines. A *two-counter machine* (see [13]) has a finite set of *states* $Q = \{q_0, q_1, \ldots, q_{n-1}\}$, an *initial* state q_0, a *final* state q_f, two *counters* a and b, and an *instruction set* which are of the form:

- the *increment* instruction q_i $c = c + 1$; q_j means that in the state q_i the machine increases its counter c by one and moves to the state q_j;

– the *decrement* instruction q_i *if* $c \neq 0$ *then* $c = c - 1$; q_j *else* q_k means that in the state q_i the machine decreases its counter c by one and moves to the state q_j if $c > 0$, otherwise it does not change c and moves to the state q_k.

A *configuration* of a counter machine is a triple (q_i, a, b) where q_i is a state, a and b are natural numbers. $(q_i, a, b) \vdash_M (q_j, a', b')$ means that the machine moves from the configuration (q_i, a, b) to the configuration (q_j, a', b') in one step. $\vdash_M^*$ denotes reflexive and transitive closure of the relation $\vdash_M$. The machine M *halts* on an input (a, b) if $(q_0, a, b) \vdash_M^* (q_f, a', b')$ for some a' and b'. Remember that Σ_1 is the class of recursive enumerable sets and Π_1 is the class of its complements (see [14]). It is known (see [13]) that halting problem and its complement are respectively Σ_1- and Π_1-hard for two-counter machines for initial configurations of the form $(q_0, 0, 0)$.

In the proof of the next theorem we show the method of modeling a counter machine using regular languages. In the next section we will prove a stronger version of this theorem but the proof will be technically more difficult.

Theorem 2 *The theory T_2 is Σ_1-hard and Π_1-hard.*

Proof We reduce the halting problem and its complement to T_2. Let M be any two-counter machine with the set of states $Q = \{ q_0, q_1, \ldots, q_{m-1} \}$ and the set of instructions $P = \{ p_1, \ldots, p_s \}$. Let us suppose that the state q_0 is initial and the state q_1 is final. We encode the configuration (q_i, a, b) by the word of length $2^i 3^a 5^b$. In particular, the initial configuration $(q_0, 0, 0)$ is encoded by the word of length 1, i.e. by the word 0.

Now for every ℓ, $1 \leq \ell \leq s$, we construct a formula $\mathrm{Step}_\ell(x, y)$ expressing the following property:

– both languages x and y contain exactly one word each;
– if x contains the code of configuration c_1, then y contains the code of configuration c_2 such that $c_1 \vdash_M c_2$ according to the instruction p_ℓ.

Note that we do not check whether x and y indeed contain the correct code because we cannot express the property that the word length is of the form $2^i 3^a 5^b$.

We use two auxiliary formulas $\mathrm{State}_i(x)$ and $\mathrm{ChangeState}_{i,j}(x, y)$. The formula $\mathrm{State}_i(x)$ says that the configuration of code x has the state q_i:

$$(\exists z)\mathrm{Mult}_{2^i}(z, x) \wedge \neg(\exists u)\mathrm{Mult}_{2^{i+1}}(u, x).$$

The formula $\mathrm{ChangeState}_{i,j}(x, y)$ says that x and y encode two equal configurations except the state, x has the state q_i and y has the state q_j:

$$(\exists z)(\mathrm{Mult}_{2^i}(z, x) \wedge \mathrm{Mult}_{2^j}(z, y)) \wedge \neg(\exists u)\mathrm{Mult}_{2^{i+1}}(u, x).$$

If $p_\ell \in P$ is q_i $a = a + 1$; q_j, then we construct a formula $\mathrm{Step}_\ell(x, y)$ as

$$(\exists z)(\mathrm{ChangeState}_{i,j}(x, z) \wedge \mathrm{Mult}_3(z, y)).$$

If $p_\ell \in P$ is q_i *if* $a \neq 0$ *then* $a = a - 1$; q_j *else* q_k, then a formula $\mathrm{Step}_\ell(x, y)$ is

$$\big(\neg(\exists z)\mathrm{Mult}_3(z, x) \to \mathrm{ChangeState}_{i,k}(x, y)\big) \wedge \big((\exists z)\mathrm{Mult}_3(z, x) \to (\exists u)(\mathrm{Mult}_3(u, x) \wedge \mathrm{ChangeState}_{i,j}(u, y))\big).$$

For the counter b formulas $\mathrm{Step}_\ell(x, y)$ are obtained by replacing 3 with 5.

The formula $\mathrm{Step}(x, y)$ says that the machine M moves in one step from the configuration of code x to the configuration of code y:

$$\bigvee_{\ell=1}^{s} \mathrm{Step}_\ell(x, y).$$

The formula $\mathrm{Closed}(x)$ says: if the language x contains the code of a configuration c, then it contains the codes of all configurations reachable from c. I.e. $\mathrm{Closed}(x)$ is:

$$(\forall u)(\forall v)((u \subseteq x \wedge \mathrm{Step}(u, v)) \to v \subseteq x).$$

The formula $\mathrm{Reachable}(x)$ says that the language x contains the word 0 encoding the initial configuration $(q_0, 0, 0)$, the codes of all configurations reachable from $(q_0, 0, 0)$, and no other word, i.e. that x is the smallest closed language containing 0:

$$0 \subseteq x \wedge \mathrm{Closed}(x) \wedge (\forall y)((0 \subseteq y \wedge \mathrm{Closed}(y)) \to x \subseteq y).$$

Now we can describe the reductions. Let φ and ψ be the next formulas correspondingly:

$$(\exists x) \quad (\mathrm{Reachable}(x) \wedge (\exists y)(y \subseteq x \wedge \mathrm{State}_1(y))),$$
$$(\exists x) \quad (\mathrm{Reachable}(x) \wedge \neg(\exists y)(y \subseteq x \wedge \mathrm{State}_1(y))).$$

The formula φ says that the final configuration is reachable from $(q_0, 0, 0)$, and ψ says that it is not reachable. Then $\varphi \in T_2$ if and only if M halts on $(q_0, 0, 0)$, and $\psi \in T_2$ if and only if M does not halt on $(q_0, 0, 0)$. Therefore, the theory T_2 is both Σ_1- and Π_1-hard. □

5 Undecidability of Theory of Regular Languages with Union and the Kleene Star

In this section we study the theory T_3 of regular languages over the alphabet $\{0\}$ with the constant $0^{(0)}$ (the language $\{0\}$) and operations $+^{(2)}$ (union) and $^{*(1)}$ (the Kleene star).

The signature of T_3 contains union, therefore intersection, subtraction, complement, and other relations and functions listed in the beginning of the previous section are also definable in T_3. The constant ε denoting the language $\{\varepsilon\}$ can be defined with Kleene star as $\emptyset^*$.

The following lemma is evident.

Lemma 2 *If the relation* $\mathrm{Mult}_{k_i}(x, y)$ *is definable in* T_3 *for natural numbers* $k_1, \ldots, k_n$, *then* $\mathrm{Mult}_m(x, y)$ *is also definable for the product* $m = k_1 \ldots k_n$.

Our main technical result is the following.

Lemma 3 *The relations* $\mathrm{Mult}_k(x, y)$ *are definable in* T_3 *for* $k = 2, 3, 5, 7$.

Proof Let the relation $\mathrm{Mult}_{k_1,\dots,k_m}(x, y)$ mean that the language x contains a single word 0^p, and the language y contains exactly the words $0^{k_1 p}, \dots, 0^{k_m p}$ for some natural number $p \neq 0$. Obviously, if $\mathrm{Mult}_{k_1}(x, y), \dots, \mathrm{Mult}_{k_m}(x, y)$ are definable, then $\mathrm{Mult}_{k_1,\dots,k_m}(x, y)$ is also definable:

$$(\exists z_1)\dots(\exists z_m)(\mathrm{Mult}_{k_1}(x, z_1) \wedge \dots \wedge \mathrm{Mult}_{k_m}(x, z_m) \wedge y = z_1 + \dots + z_m).$$

The formula $\mathrm{Mult}_{2,3}(x, y)$ can be defined as

$$\mathrm{Card}_1(x) \wedge \mathrm{Card}_2(y) \wedge x^* \setminus x = y^*.$$

Indeed, let $L = \{0^p\}$ where p is some natural number, $L_1 = L^* \setminus L$. Then $L_1 = \{\varepsilon, 0^{2p}, 0^{3p}, 0^{4p}, \dots\} = \{0^{2p}, 0^{3p}\}^*$. Thus, L_1 is a Kleene star of some language containing exactly two words. On the other hand, every such language must contain both words 0^{2p} and 0^{3p} because they cannot be obtained from shorter words.

Now we define $\mathrm{Mult}_2(x, y)$ and $\mathrm{Mult}_3(x, y)$ correspondingly:

$$\mathrm{Card}_1(y) \wedge (\exists z)(\mathrm{Mult}_{2,3}(x, z) \wedge y \subseteq z) \wedge \neg(\exists u)(\mathrm{Card}_2(u) \wedge x^* \setminus (x + y) = u^*);$$

$$\mathrm{Card}_1(y) \wedge (\exists z)(\mathrm{Mult}_{2,3}(x, z) \wedge y \subseteq z) \wedge \neg\mathrm{Mult}_2(x, y).$$

If we remove 0^{3p} from L_1, then we get $L_2 = \{\varepsilon, 0^{2p}, 0^{4p}, 0^{5p}, 0^{6p}, \dots\}$. This language is a Kleene star of $\{0^{2p}, 0^{5p}\}$. Removing 0^{2p} from L_1 we get $L_3 = \{\varepsilon, 0^{3p}, 0^{4p}, 0^{5p}, 0^{6p}, \dots\}$. If L_3 is a Kleene star of some language L_3', then L_3' must contain at least three words $0^{3p}, 0^{4p}, 0^{5p}$ because neither of these words can be represented as concatenation of two shorter nonempty words from L_3.

The construction of $\mathrm{Mult}_k(x, y)$ for $k = 5, 7$ follows the same idea. The formula $\mathrm{Mult}_{5,7}(x, y)$ is

$$(\exists z)(\exists u)(\exists v)(\mathrm{Mult}_{2,3,4}(x, u) \wedge \ \mathrm{Mult}_{6,8,9}(x, v) \wedge \mathrm{Card}_5(z) \wedge x^* \setminus (x + u) = z^* \wedge y = z \setminus v).$$

Here z must be $\{0^{5p}, 0^{6p}, 9^{7p}, 0^{8p}, 0^{9p}\}$ and $y = \{0^{5p}, 0^{7p}\}$. Then we use the equality $\{\varepsilon, 0^{5p}, 0^{6p}, 0^{8p}, 0^{9p}, \dots\} = \{0^{5p}, 0^{6p}, 0^{8p}, 0^{9p}\}^*$. On the other hand, any language L_4 such that $L_4^* = \{\varepsilon, 0^{6p}, 0^{7p}, 0^{8p}, 0^{9p}, \dots\}$ must include at least $\{0^{6p}, 0^{7p}, 0^{8p}, 0^{9p}, 0^{10p}, 0^{11p}\}$:

$$\begin{aligned}
\mathrm{Mult}_5(x, y) \equiv_{T_3} \mathrm{Card}_1(y) \wedge (\exists z)(\exists u)(\mathrm{Mult}_{5,7}(x, z) \wedge \mathrm{Mult}_{2,3,4}(x, u) \wedge \\
y \subseteq z \wedge \neg(\exists v)(\mathrm{Card}_4(v) \wedge x^* \setminus (x + u + y) = v^*)), \\
\mathrm{Mult}_7(x, y) \equiv_{T_3} \mathrm{Card}_1(y) \wedge (\exists z)(\exists u)(\mathrm{Mult}_{5,7}(x, z) \wedge \mathrm{Mult}_{2,3,4}(x, u) \wedge \\
y \subseteq z \wedge (\exists v)(\mathrm{Card}_4(v) \wedge x^* \setminus (x + u + y) = v^*).
\end{aligned}$$

□

Corollary 2 *The constants* 0^2, 0^3, 0^5, 0^7 *are definable in* T_3.

Lemma 4 *The following relation* $\mathrm{Join}(x, y, z)$ *is definable in* T_3*: x, y, and z contain one word each, and if $x = \{0^p\}$ and $y = \{0^q\}$ for some p and q, then $z = \{0^{\mathrm{lcm}(p,q)}\}$, where* lcm *denotes the least common multiple.*

Proof Let us define $\mathrm{Join}(x, y, z)$ as

$$\mathrm{Card}_1(x) \wedge \mathrm{Card}_1(y) \wedge \mathrm{Card}_1(z) \wedge z^* = x^* \cap y^*.$$

If $L_1 = \{0^p\}$ and $L_2 = \{0^q\}$, then $L_1^* \cap L_2^* = \{0^{\mathrm{lcm}(p,q)\times i} : i \geq 0\}$. Therefore, $L_1^* \cap L_2^* = \{0^{\mathrm{lcm}(p,q)}\}^*$.

□

Theorem 3 *For every arithmetical formula φ one can construct a formula ψ such that φ is true in arithmetic if and only if $\psi \in T_3$.*

Proof It is known (see [15]) that two numbers can be multiplied by a counter machine with only three counters. We use next three three-counters machines:

– M_1 moves a number from the first counter into the second one:

$$(q_0, a, 0, 0) \vdash^* (q_1, 0, a, 0);$$

– M_2 computes the sum of the first two counters:

$$(q_0, a, b, 0) \vdash^* (q_1, a + b, 0, 0);$$

– M_3 computes the product of the first two counters:

$$(q_0, a, b, 0) \vdash^* (q_1, ab, 0, 0).$$

Like in the proof of Theorem 2 we can construct the formulas $\mathrm{Closed}_i(x)$ expressing the following: if $0^c \in x$ where $c = 2^j 3^a 5^b 7^d$ is the code of a configuration (q_j, a, b, d) of M_i, then x contains the codes of all reachable configurations. Concatenation was used only to construct the formulas $\mathrm{Mult}_k(x, y)$. But since each M_i has only three counters we only need the formulas $\mathrm{Mult}_k(x, y)$ where k is either power of 2 or one of 3, 5, 7. All such relations are definable in T_3 by Lemmas 2 and 3.

The following formula $\mathrm{Primes}_{p_1,\ldots,p_m}(x)$ expresses that $x = \{0^n\}$ where $n = p_1^{\alpha_1} \ldots p_m^{\alpha_m}$ for some primes $p_1, \ldots, p_m \in \{2, 3, 5, 7\}$:

$$\mathrm{Primes}_{p_1,\ldots,p_m}(x) \equiv_{T_3} x = 0 \vee \big(x \neq \varepsilon \wedge \mathrm{Card}_1(x) \wedge (\forall y)((\mathrm{Card}_1(y) \wedge x \subseteq y^*) \rightarrow (y = 0 \vee y \subseteq (0^{p_1})^* \vee \cdots \vee y \subseteq (0^{p_m})^*))\big).$$

In particular, the formula $\mathrm{Primes}_p(x)$ says $x = \{0^{p^\alpha}\}$ for some α.

The formula $\mathrm{Conf}(x)$ expresses that $x = \{0^c\}$ where c encodes some configuration:

$$\mathrm{Conf}(x) \equiv_{T_3} \mathrm{Primes}_{2,3,5,7}(x).$$

The formula $\mathrm{InitConf}_1(x) \equiv_{T_3} \mathrm{Primes}_3(x)$ expresses that $x = \{0^c\}$ where c encodes an initial configuration of M_1, $\mathrm{InitConf}_2(x) \equiv_{T_3} \mathrm{InitConf}_3(x) \equiv_{T_3} \mathrm{Primes}_{3,5}(x)$ expresses that $x = \{0^c\}$ where c encodes an initial configuration of M_2 or M_3.

Analogously formulas $\mathrm{FinalConf}_i(x)$ expresses that x encodes a final configuration: $\mathrm{State}_1(x) \wedge \mathrm{Primes}_{2,5}(x)$ and $\mathrm{State}_1(x) \wedge \mathrm{Primes}_{2,3}(x)$ correspondingly.

The formula $\mathrm{Result}_i(x, y)$ expresses that x encodes an initial configuration of M_i and y encodes a corresponding final configuration:

$$\mathrm{InitConf}_i(x) \wedge \mathrm{FinalConf}_i(y) \wedge (\forall u)((x \subseteq u \wedge \mathrm{Closed}_i(u)) \to y \subseteq u).$$

We represent a natural number n by $\{\, 0^{3^n} \,\}$. The formula $\mathrm{Add}(x, y, z)$ expresses that the language z represents the sum of numbers represented by x and y:

$$\mathrm{Primes}_3(x) \wedge \mathrm{Primes}_3(y) \wedge (\exists u)(\exists v)(\exists w)(\exists t)(\mathrm{Result}_1(y, u) \wedge \mathrm{Mult}_2(v, u) \wedge \mathrm{Join}(x, v, w) \wedge \mathrm{Result}_2(w, t) \wedge \mathrm{Mult}_2(y, t)).$$

Let $x = \{\, 0^{3^a} \,\}$ and $y = \{\, 0^{3^b} \,\}$. Then $u = \{\, 0^{2\times 5^b} \,\}$, $v = \{\, 0^{5^b} \,\}$, $w = \{\, 0^{3^a 5^b} \,\}$, $t = \{\, 0^{2\times 3^{a+b}} \,\}$, and $z = \{\, 0^{3^{a+b}} \,\}$.

Using the same method we can construct the formula $\mathrm{Mult}(x, y, z)$ for multiplication:

$$\mathrm{Primes}_3(x) \wedge \mathrm{Primes}_3(y) \wedge (\exists u)(\exists v)(\exists w)(\exists t)(\mathrm{Result}_1(y, u) \wedge \mathrm{Mult}_2(v, u) \wedge \mathrm{Join}(x, v, w) \wedge \mathrm{Result}_3(w, t) \wedge \mathrm{Mult}_2(y, t)).$$

Now we can describe reduction from arithmetic to T_3. Let φ be an arbitrary arithmetical formula. We may suppose that all atomic subformulas of φ are of the forms $x + y = z$ or $x \times y = z$ where x, y, z are variables. For every arithmetical formula θ we construct its translation $\mathrm{T}(\theta)$ by induction:

- $\mathrm{T}(x + y = z)$ is $\mathrm{Add}(x, y, z)$;
- $\mathrm{T}(x \times y = z)$ is $\mathrm{Mult}(x, y, z)$;
- $\mathrm{T}(\chi \circ \theta)$ is $\mathrm{T}(\chi) \circ \mathrm{T}(\theta)$ for $\circ \in \{\wedge, \vee, \to\}$;
- $\mathrm{T}(\neg\theta)$ is $\neg\mathrm{T}(\theta)$;
- $\mathrm{T}((\exists x)\theta)$ is $(\exists x)(\mathrm{Primes}_3(x) \wedge \mathrm{T}(\theta))$;
- $\mathrm{T}((\forall x)\theta)$ is $(\forall x)(\mathrm{Primes}_3(x) \to \mathrm{T}(\theta))$.

Let ψ be $\mathrm{T}(\varphi)$. Then φ is true in arithmetic if and only if $\psi \in T_3$.

□

Finally, let us consider the theory T_4 of regular languages over an arbitrary alphabet $\Sigma = \{\, a_1, \ldots, a_n \,\}$ with constants $a_1^{(0)}, \ldots, a_n^{(0)}$ and operations $+^{(2)}$, $\cdot^{(2)}$, and $*^{(1)}$. The constant a_i is interpreted as the language $\{\, a_i \,\}$, other symbols have the previous meaning.

Theorem 4 *For every formula φ one can construct an arithmetical formula ψ such that $\varphi \in T_4$ if and only if ψ is true in arithmetic.*

Proof We suppose that all atomic subformulas of φ are of the forms $x + y = z$, $x \cdot y = z$, $x^* = y$, or $x = a_i$.

Let us fix some "reasonable" numeration of all DFA and let M_n be a DFA of number n. We may suppose that for every number n there exists a DFA M with the number n. There are algorithms which given DFA M_1 and M_2 construct DFA recognizing the

languages $L(M_1) \cup L(M_2)$, $L(M_1) \cdot L(M_2)$, and $L(M_1)^*$ (see [1, 8]). Then the relations $L(M_x) \cup L(M_y) = L(M_z)$, $L(M_x) \cdot L(M_y) = L(M_z)$, and $L(M_x)^* = L(M_y)$ are recursive. But every recursive relation is representable in arithmetic (see [3]). Let corresponding formulas be Union(x, y, z), Concat(x, y, z), and Star(x, y). Let n_i be the number of some DFA recognizing $\{a_i\}$. For every formula φ we construct its translation by induction:

- $\mathrm{T}(x = a_i)$ is $x = \underbrace{1 + \cdots + 1}_{n_i \text{ times}}$;
- $\mathrm{T}(x + y = z)$ is Union(x, y, z);
- $\mathrm{T}(x \cdot y = z)$ is Concat(x, y, z);
- $\mathrm{T}(x^* = y)$ is Star(x, y);
- $\mathrm{T}(\chi \circ \theta)$ is $\mathrm{T}(\chi) \circ \mathrm{T}(\theta)$ for $\circ \in \{\wedge, \vee, \to\}$;
- $\mathrm{T}(\neg\theta)$ is $\neg\mathrm{T}(\theta)$;
- $\mathrm{T}((\exists x)\theta)$ is $(\exists x)\mathrm{T}(\theta)$;
- $\mathrm{T}((\forall x)\theta)$ is $(\forall x)\mathrm{T}(\theta)$.

Let ψ be $\mathrm{T}(\varphi)$. Then $\varphi \in T_4$ if and only if ψ is true in arithmetic.

□

Corollary 3 *The theories T_3, T_4 and arithmetic are recursively isomorphic.*

Corollary 4 *The theory T_2 and arithmetic are recursively isomorphic.*

Proof It is enough to define the Kleene star with union and concatenation:

$$x^* = y \equiv_{T_4} x \subseteq y \wedge y \cdot y = y \wedge (\forall z)((x \subseteq z \wedge z \cdot z = z) \to y \subseteq z).$$

□

Corollary 5 *The theories T_2, T_3, and T_4 are undecidable.*

6 Undecidability of Theory of Regular Languages with Concatenation Only

In [4] a theory T_5 was investigated. This is the theory of regular languages over one-symbol alphabet with concatenation only. The main result of [4] is the following: the theory T_5 is algorithmically equivalent to elementary arithmetic. Indeed, the following theorem was proved:

Theorem 5 (Dudakov S.M., [4]) *Let A_e be the set of regular languages containing the empty word ε over one-symbol alphabet $\{0\}$. Then the theory T_{5e} of the algebra $\mathfrak{A}_e = (A_e, \cdot)$ is algorithmically equivalent to arithmetic.*

The crucial point of the proof is to construct a ternary relation $P(x, y, z)$ that is true if and only if

$$x = \{\varepsilon, 0^m\}, \quad z = \{\varepsilon, 0^{nm}\}, \quad y = \{\varepsilon, 0^m, 0^{2m}, 0^{3m}, \ldots, 0^{nm}\}$$

for some positive natural numbers m and n. Then the relation between z and x corresponds to divisibility of natural numbers, and concatenation of two y's corresponds to addition. It is enough to interpret addition and multiplication of natural numbers.

But this proof cannot be generalized directly to arbitrary regular languages because the construction of P uses concatenation commutativity. So we need another proof when an alphabet contains more than one symbol.

Here we consider a theory T_6 of regular languages over a multi-symbol alphabet with concatenation. We demonstrate that the algebra $\mathfrak{A}_e$ is interpretable in T_6. So T_6 is algorithmically as hard as arithmetic.

Firstly we prove some definabilities.

Lemma 5 *The following is definable in T_6:*

1) the empty language $\emptyset$;
2) the set R_e of languages containing the empty word ε
3) the Kleene star operation * for languages from R_e.

Proof 1) Evidently

$$x = \emptyset \equiv_{T_6} (\forall y) x \cdot y = x.$$

For $x = \emptyset$ we have $\emptyset \cdot y = \emptyset$. If $x \neq \emptyset$, then for $y = \emptyset$ we obtain $x \cdot \emptyset = \emptyset \neq x$.

2) The set R_e can be defined as

$$R_e(x) \equiv_{T_6} (\exists y)(y \neq \emptyset \wedge x \cdot y = y).$$

If $\varepsilon \in x$, then we can assume $y = \Sigma^*$, so $y \neq \emptyset$ and $x \cdot y = y$ because

$$\Sigma^* \supseteq x \cdot \Sigma^* \supseteq \varepsilon \cdot \Sigma^* = \Sigma^*.$$

Consider the case of $\varepsilon \notin x$. If $x = \emptyset$, then $\emptyset \cdot y = \emptyset \neq y$ for any $y \neq \emptyset$. Otherwise, let w be the shortest word in x. Hence, $|w| \geq 1$. If y is any non-empty language and u is the shortest word in y, then the shortest word in the language $x \cdot y$ is of length $|u| + |w| > |u|$. So $u \notin x \cdot y$ and $x \cdot y \neq y$.

3) If $x \in R_e$, then the Kleene star for x is

$$y = x^* \equiv_{T_6} R_e(y) \wedge x \cdot y = y \wedge (\forall z)(x \cdot z = z \rightarrow y \cdot z = z).$$

Let us prove it.

Evidently $R_e(x^*)$ holds. The equality $x \cdot x^* = x^*$ holds due to the Kleene star definition. Let z be any language such that $x \cdot z = z$. Then $x^0 \cdot z = \{\varepsilon\} \cdot z = z$ and by induction we obtain $x^{n+1} \cdot z = x \cdot x^n \cdot z = x \cdot z = z$ for every natural number n. Hence,

$$x^* \cdot z = \left(\bigcup_{n\in\omega} x^n\right) \cdot z = \bigcup_{n\in\omega} (x^n \cdot z) = \bigcup_{n\in\omega} z = z.$$

Now let the language y satisfy the defining formula. We must prove $y = x^*$. We have $x^0 = \{\varepsilon\} \subseteq y$, and by induction $x^{n+1} = x \cdot x^n \subseteq x \cdot y = y$. Hence, $x^n \subseteq y$ for every natural number n and

$$x^* = \bigcup_{n\in\omega} x^n \subseteq \bigcup_{n\in\omega} y = y.$$

Thus, $x^* \subseteq y$.

Suppose that $x^* \neq y$. Then there is a word $u \in y \setminus x^*$. Let $z = x^*$. In this case the formula $x \cdot z = z$ is true. But the formula $y \cdot z = z$ is false because $u \in y \cdot z$ and $u \notin z$. The contradiction proves that the case $x^* \neq y$ is impossible and $y = x^*$. □

To continue we need to remember some well-known concatenation properties. The proofs of the following result can be found in [12], Corollary 4.1, and in [2], Theorem 2.3.5:

Theorem 6 (see [2, 12]) *1) Let $u^i = w^j$ for some words u, w and natural numbers i, j, $i + j > 0$. Then $u, w \in v^*$ for some word v.*
2) Let $w^k u^\ell = u^i w^j$ for some words u, w and natural numbers k, ℓ, i, j, and $k, \ell > 0$. Then $u, w \in v^*$ for some word v.

Now let us continue to investigate the theory of regular languages with concatenation.

Lemma 6 *In the theory T_6 the set R_* is definable. The set R_* contains all languages x such that $\varepsilon \in x$ and $x \subseteq w^*$ for some word w.*

Proof The set R_* can be defined as

$$R_*(x) \equiv_{T_6} R_e(x) \wedge (\forall y)(\forall z)\left((y \cdot x^* = x^* \wedge z \cdot x^* = x^*) \rightarrow y \cdot z = z \cdot y\right).$$

Let $x \in R_*$, i. e. $x \subseteq w^*$ for some word w. If $x = \{\varepsilon\}$, then $x^* = \{\varepsilon\}$ and the implication is true because the condition can be true for $y = z = \{\varepsilon\}$ only.

Consider the case of $x \neq \{\varepsilon\}$. As $\varepsilon \in x$ so x contains some non-empty word from w^* and $w \neq \varepsilon$ necessarily. Let w^k be the shortest non-empty word in x. The formula $R_e(x)$ is true trivially because $R_* \subseteq R_e$. Let $y \cdot x^* = x^*$ and $z \cdot x^* = x^*$ for some languages y and z. Then for any word $u \in y$ we have $uw^k = w^n$ for some natural number n, hence, $u = w^{n-k}$. We obtain $y \subseteq w^*$. Analogously, $z \subseteq w^*$. Therefore, $y \cdot z = z \cdot y$.

Now let the defining formula be true for some language x. Then x must contain the empty word ε. If $x = \{\varepsilon\}$, then $x \subseteq w^*$ for all w. Otherwise, let w be the shortest non-empty word in x. Thus, $w^* \subseteq x^*$. For the language $y = \{\varepsilon, w\}$ we have

$$y \cdot x^* = (\varepsilon \cdot x^*) \cup (w \cdot x^*) \subseteq x^* \cup x^* = x^*.$$

Let m be the greatest natural number such that $w = v^m$ for some word v. Evidently, $m \geq 1$ because $w = w^1$.

Let us select an arbitrary non-empty word $u \in x$ and consider the language $z = \{\varepsilon, u\}$. Analogously, we obtain $u^* \subseteq x^*$ and $z \cdot x^* = x^*$. So $y \cdot z = z \cdot y$ by the implication:

$$\{\, \varepsilon, w, u, wu \,\} = y \cdot z = z \cdot y = \{\, \varepsilon, w, u, uw \,\}.$$

The words w and u are of positive length, hence, the length of words wu and uw is greater than the lengths of ε, u, and w. This means $wu = uw$ and by Theorem 6 we have $w = v_1^i$ and $u = v_1^j$ for some word v_1 and natural numbers i and j. As $v^m = w = v_1^i$ so from Theorem 6 it follows that $v = v_0^k$ and $v_1 = v_0^\ell$ for some word v_0 and natural numbers k and ℓ. Hence, $w = v^m = v_0^{mk}$. But the number m is maximal possible, so $mk = m$ and $v_0 = v$. Thus, $u = v^{j\ell}$.

We have proved that $u \in v^*$ for any non-empty word $u \in x$. Therefore, $x \subseteq v^*$. □

Lemma 7 *Let $x \in R_*$, $x \neq \{\, \varepsilon \,\}$, $x \subseteq w^*$, and the word w be of minimal length. Then for any language $y \in R_* \cup \{\emptyset\}$ the equality $x \cdot y = y \cdot x$ holds if and only if $y \subseteq w^*$.*

Proof Let $x \cdot y = y \cdot x$. If $y = \emptyset$ or $y = \{\varepsilon\}$, then $y \subseteq w^*$ trivially.

Otherwise, let $y \subseteq u^*$ where the length of u is minimal possible. Consider any non-empty words $w^k \in x$ and $u^\ell \in y$. Then we have $w^k u^\ell \in x \cdot y$, hence, $w^k u^\ell \in y \cdot x$ and there exist natural numbers i and j such that $w^k u^\ell = u^i w^j$. By Theorem 6 we obtain $w, u \in v^*$ for some v. But w and u cannot be of the form v^m for $m > 1$. Therefore, $w = v = u$ and $y \subseteq w^*$.

The opposite claim is trivial: if $x, y \subseteq w^*$, then $x \cdot y = y \cdot x$. □

Now we can interpret the algebra $\mathfrak{A}_e$ in the theory T_6.

Theorem 7 *The theory of the algebra $\mathfrak{A}_e$ from Theorem 5 is interpretable in the theory T_6. Hence, T_6 is algorithmically as hard as the elementary arithmetic.*

Proof Let us fix any language $x_0 \in R_*$, $x_0 \neq \{\varepsilon\}$. Due to Lemma 6 we can define it: $R_*(x_0) \wedge x_0 \neq \varepsilon$. Assume that $x_0 \subseteq w^*$ and w has minimal length. Consider the set R_0 of languages such that the following formula is true:

$$R_0(x) \equiv (R_*(x) \vee x = \emptyset) \wedge x_0 \cdot x = x \cdot x_0.$$

Due to Lemma 7 such x can be only a subset of w^*. The set R_0 is a domain of our interpretation.

Now, we have an isomorphism f between the algebra $\mathfrak{A}_e$ and the algebra $(R_0, \cdot)$:

$$f(u) = \{\, w^m : 0^m \in u \,\}, \quad f(u_1 \cdot u_2) = f(u_1) \cdot f(u_2).$$

So, for each formula ψ for $\mathfrak{A}_e$ we can construct its translation $\mathrm{T}(\psi)$ for T_6 by induction:

– all atomic formulas remain unchanged: $\mathrm{T}(t = s)$ is $t = s$ for any terms t and s;
– $\mathrm{T}(\psi_1 \circ \psi_2)$ is $\mathrm{T}(\psi_1) \circ \mathrm{T}(\psi_2)$ for $\circ \in \{\, \wedge, \vee, \to \,\}$ and $\mathrm{T}(\neg\psi)$ is $\neg\mathrm{T}(\psi)$;

– all quantifiers become R_0-bounded: $\mathrm{T}((\exists x)\psi)$ is $(\exists x)(R_0(x) \wedge \mathrm{T}(\psi))$ and $\mathrm{T}((\forall x)\psi)$ is $(\forall x)(R_0(x) \to \mathrm{T}(\psi))$.

Entire closed formula φ becomes

$$\varphi^* \equiv (\exists x_0)(R_*(x_0) \wedge x_0 \neq \varepsilon \wedge \mathrm{T}(\varphi)).$$

Therefore, φ belongs the theory of $\mathfrak{A}_e$ if and only if φ^* belongs T_6.

□

The theory T_2 is a conservative extension of T_6, hence, T_6 cannot be more complex than T_2. Using Corollary 4 we obtain

Corollary 6 *The theory T_6 and arithmetic are recursively isomorphic.*

7 Conclusion

We have considered classical operations on languages: union, concatenation and the Kleene star. We have proved that the theory of regular language is decidable with the Kleene star only but undecidable with union and other operations and with concatenation only.

Some interesting problems remain open.

– What is time and space complexity of the theory of regular languages with union only?
– Are there other natural operations on regular languages with decidable theories?
– Investigate decidability of other language classes with different operations.
– In particular, find non-trivial classes and operations with decidable theories.
– What is the set of all constants definable in T_3?

Acknowledgments This work is funded by Russian Foundation for Basic Research (RFBR), project number 20-01-00435.

References

1. Aho, A.V., Ullman, J.D.: The theory of parsing, translation and compiling. Volume 1: parsing. Englewood cliffs, NJ: Prentice-Hall inc (1972)
2. Allouche, J., Shallit, J.: Automatic sequences: theory, applications, generalizations. Cambridge University Press, Cambridge (2003)
3. Boolos, G.S., Burgess, J.P., Jeffrey, R.C.: Computability and Logic, 5th edn. Cambridge University Press, New York (2007)
4. Dudakov, S.M.: On undecidability of concatenation theory for one-symbol languages. Lobachevskii J. Math. **41**(2), 168–175 (2020)
5. Dudakov, S., Karlov, B.: On decidability of regular languages theories. In: Van Bevern, R., Kucherov, G. (eds.) Computer Science — Theory and Applications. CSR 2019. Lecture notes in computer science, vol. 11532, pp. 119–130 (2019)
6. Grzegorczyk, A.: Undecidability without arithmetization. Stud. Logica. **79**(2), 163–230 (2005)

7. Grzegorczyk, A., Zdanowski, K.: Undecidability and Concatenation. In: Ehrenfeucht, A., Marek, V.W., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 72–91. IOS Press, Amsterdam (2008)
8. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, 3rd edn. Pearson, Harlow, Essex (2013)
9. Karlov, B.N.: On regular languages theory with Kleene star operation (in Russian). In: Proceedings of International Conference Algebra and Mathematical Logic: Theory and Applications, pp. 117–119 (2019)
10. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Shannon, C., McCarthy, J. (eds.) Automata Studies, pp. 3–42. Princeton University Press, Princeton (1951)
11. Koppelberg, S.: HandBook of Boolean Algebras, Volume I. North Holland P.C (1989)
12. Lyndon, R.C., Schützenberger, M.P.: The equation $a^M = b^N c^P$ in a free group. Michigan Math. J. **9**(4), 289–298 (1962)
13. Minsky, M.L.: Computation: finite and infinite machines. Englewood cliffs, NJ: Prentice-Hall inc. (1967)
14. Rogers, H.: Theory of Recursive Functions and Effective Computability. McGraw-Hill Education, New York (1967)
15. Schroeppel, R.: A Two-Counter Machine Cannot Calculate 2^N. Technical Report, vol. 257. Massachusetts Institute of Technology, A. I. Laboratory (1973)
16. Švejdar, V.: On interpretability in the theory of concatenation. Notre dame J. Formal Logic **50**(1), 87–95 (2009)
17. Visser, A.: Growing commas. A study of sequentiality and concatenation. Notre dame J. Formal Logic **50**(1), 61–85 (2009)