S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Vol. 21, No. 1, pp. 144–150 (2024)
https://doi.org/10.33048/semi.2024.21.???

УДК 510.52
MSC 03D15, 03D45, 16Y60

POLYNOMIAL-TIME COMPUTABLE MIN-PLUS SEMIRINGS

A.V. SELIVERSTOV

Communicated by P.P. Petrov

Abstract: We consider min-plus semirings, where the universe (or domain) is the set of all natural numbers represented in binary form, both minimum and sum are polynomial-time computable, and the computational complexity of the minimum depends only on the greatest element. Over a semiring isomorphic to the min-plus semiring of natural numbers, both truncated difference and integer division in half are polynomial-time computable with an oracle for some level of the polynomial-time hierarchy. Thus, they can be computed using polynomial space. Moreover, if two such polynomial-time computable semirings with the same universe are isomorphic to each other, then there are two primitive recursive isomorphisms between the semirings so that their composition is equal to the identity map. In this way, some previous results related to punctual categoricity can be further refined. On the other hand, our results are consistent with previous results on polynomial-time computable structures because we consider categoricity with respect to the fixed universe as well as use a non-trivial restriction on the computational complexity.

Keywords: semiring, punctual structure, polynomial time, oracle, computational complexity.

Seliverstov, A.V., Polynomial-time computable min-plus semirings.

Received January, 31, 2024, Published December, 31, 2024.

1 Introduction

The better computer technologies become, the more practical polynomial-time computability becomes. So, it is important for big data processing. This remark explains searching for new approaches to describe the class of polynomial-time computable functions. Recently, a new description is obtained by Goncharov and Nechesov [1, 2]. Of course, many authors limit themselves to obtaining some upper bound on algebraic complexity but ignore the complexity of both data representation and memory access [3, 4]. This approach is consistent with modifying of the Grzegorczyk hierarchy by Alan Cobham in the 1960s. However, polynomial-time computable algebraic structures have also attracted the attention. An algebraic structure models both coding system and memory organization. For example, a non-trivial algebraic structure is used in the hidden subgroup problem, which is closely related to public key cryptography. But the approach is free of some nerdy details of Turing machines. In the early 1990s, two seminal papers had been published by Cenzer and Remmel [5, 6]. Recently, results on finitely generated polynomial-time computable structures are obtained by Alaev [7, 8]. Although polynomial-space computable structures are less studied, they have been considered by Cenzer, Downey, Remmel, and Uddin [9].

If runtime or memory must be small, then it is important to fix the certain method of representing natural numbers. Let us use binary notations of natural numbers and denote by $\mathrm{Bin}(\omega)$ the set of all natural numbers represented in binary form.

A straightforward generalization of the concept of computable categoricity turned out to be useless for polynomial-time computable structures and isomorphisms [10, 11]. However, other variants of computable categoricity are more interesting. Roughly speaking, we found a polynomial-time computable structure $\mathfrak{N}$ such that for every isomorphic polynomial-time computable structure $\mathfrak{M} \cong \mathfrak{N}$ with the same universe, there are primitive recursive isomorphisms $f : \mathfrak{M} \to \mathfrak{N}$ and $g : \mathfrak{N} \to \mathfrak{M}$ so that $f \circ g = g \circ f = \mathrm{id}$. In contrast to the concept of weak p-categoricity introduced by Alaev [10], we use $\mathrm{Bin}(\omega)$ as the universe (or domain). Last century, under additional assumptions on a common polynomial-time computable Scott family, similar results had been obtained by Cenzer and Remmel [11].

We also consider polynomial-time computations with oracles for a level of the polynomial-time hierarchy. Of course, such computations can be done using polynomial space without any oracle.

On the other hand, the theory of punctual structures (i. e., structures computable without delay) has been developed [12, 13]. In this case, the universe is either the set of all natural numbers or an initial segment; all functions and relations are primitive recursive. Many computable structures have isomorphic punctual structures [14, 15]. Punctual structures isomorphic to a computationally categorical structure are discussed by Bazhenov and Kalimullin [16]. The restriction to be primitive recursive is non-trivial because

there are computable functions growing faster than any primitive recursive function; Gabriel Sudan found a recursive function that is not primitive recursive [17]. Unfortunately, working with primitive recursive structures, any fixed universe can be inconvenient. Thus, various primitive recursive ordered fields are discussed by Selivanov and Selivanova [18].

2 Min-plus Semirings

Let us consider the min-plus semiring $\mathfrak{N}$ of natural numbers with constant 0 and binary operations min() and + defined as usual. The operations are associative, commutative, and connected to each other with the distributive law: $\min(x+z, y+z) = \min(x, y) + z$. The constant 0 serves as the neutral element for the + operation. But the semiring $\mathfrak{N}$ does not have the neutral element $+\infty$ for the min() operation. Computations over the semiring $\mathfrak{N}$ are related to integer linear programming as well as solving other discrete optimization problems. One can obtain other tropical algebras by expanding the signature. Recently, further generalizations and examples have been discussed by Krivulin [19].

Over the set of natural numbers, the min() function defines the linear order: the relation $x \leq y$ is equivalent to the equality $x = \min(x, y)$. Therefore, one can also define the max() function that is equal to the greatest element. Let us use the max() function and the order relation, although they are not part of the signature. There is no explicit bound on their computational complexity.

Let $\ell(x)$ denote the length of $x \in \mathrm{Bin}(\omega)$. Let us set $\ell(x) = \lceil \log_2(x+1) \rceil$, in particular, $\ell(0) = 0$. Next, the length of an element means the length of its binary notation, i.e., the length of the corresponding word in $\mathrm{Bin}(\omega)$. Let poly() denote a polynomial. Thus, $\mathrm{poly}(\ell(x))$ denotes a polynomial upper bound.

Over the min-plus semiring $\mathfrak{N}$, one can calculate the minimum $\min(x, y)$ using time bounded by a polynomial in the length of the binary notation of the maximum value $\max(x, y)$. Therefore, let us further assume that in any isomorphic semiring $\mathfrak{M} \cong \mathfrak{N}$, the minimum $\min(x, y)$ can also be calculated using polynomial time, depending on the length of the element $\max(x, y)$ but regardless of the length of the sought element $\min(x, y)$ for $x \neq y$. However, such a limitation on its computational complexity is no longer trivial because the length of $\max(x, y)$ may be less than the length of $\min(x, y)$ over $\mathfrak{M}$.

The truncated difference $x \dot{-} y$ is equal to the usual difference at $y \leq x$, but vanishes at $y \geq x$. As usual, $\lfloor x \rfloor$ denotes the floor function.

3 Results

Theorem 1. *Given a semiring* $\mathfrak{M} = (\mathrm{Bin}(\omega); 0, \min, +)$ *isomorphic to the min-plus semiring of natural numbers* $\mathfrak{N}$. *For all elements* x *and* y *from* $\mathfrak{M}$, *let both sum* $x + y$ *and minimum* $\min(x, y)$ *be polynomial-time computable with an oracle for some level of the polynomial-time hierarchy so that the*

minimum $\min(x, y)$ *can be computed using* $\mathrm{poly}(\ell(\max(x, y)))$ *time depending only on the length of the greatest element* $\max(x, y)$. *For all* x *and* y, *their truncated difference* $x \dot{-} y$ *is polynomial-time computable with an oracle for some level of the polynomial-time hierarchy. For all* x, *the integer division in half* $\lfloor x/2 \rfloor$ *is also polynomial-time computable with an oracle for some level of the polynomial-time hierarchy.*

Proof. For any $k \leq x$, the element $\min(x, k)$ is computed (with an oracle) in $\mathrm{poly}(\ell(x))$ time. Of course, $k = \min(x, k)$. Thus, its length is polynomially bounded, i. e., the inequality $\ell(k) \leq \mathrm{poly}(\ell(x))$ holds. The exact upper bound depends only on the computational complexity of the min() function over $\mathfrak{M}$. Therefore, computing the truncated difference $x \dot{-} y$ can be reduced to an exhaustive search over the set S of all elements with length at most $\mathrm{poly}(\ell(x))$. Checking whether the correct answer has been received requires to verify the equality $x = y + z$, where $\ell(z) \leq \mathrm{poly}(\ell(x))$. The upper bound for the computational complexity of such a check is a function of the form $\mathrm{poly}(\ell(x), \ell(y))$. If the equality $x = y + z$ holds, then $x \dot{-} y = z$. If the required element z does not belong to the set S, then $x \dot{-} y = 0$ regardless of y because $x \leq y$. The exhaustive search can be done using an oracle machine with an oracle for some level of the polynomial-time hierarchy. Thus, the truncated difference is polynomial-time computable with the oracle.

In the same way, computing the integer part of half $\lfloor x/2 \rfloor$ can be reduced to an exhaustive search over the set S of all elements with length at most $\mathrm{poly}(\ell(x))$. For $z \in S$, checking requires to verify whether either $z + z = x$ or $z + z + 1 = x$ holds. If one of them holds, then $\lfloor x/2 \rfloor = z$. If the required element z does not belong to the set S, then $\lfloor x/2 \rfloor = 0$. The exhaustive search can be done in polynomial-time with an oracle for some level of the polynomial-time hierarchy. □

As a consequence, let us formulate the following result, which is closer to practical use because it does not involve any oracle.

Theorem 2. *Given a semiring* $\mathfrak{M} = (\mathrm{Bin}(\omega); 0, \min, +)$ *isomorphic to the min-plus semiring of natural numbers* $\mathfrak{N}$. *For all elements* x *and* y *from* $\mathfrak{M}$, *let both sum* $x + y$ *and minimum* $\min(x, y)$ *be polynomial-time computable so that the minimum* $\min(x, y)$ *can be computed using* $\mathrm{poly}(\ell(\max(x, y)))$ *time depending only on the length of the greatest element* $\max(x, y)$. *For all* x *and* y, *their truncated difference* $x \dot{-} y$ *is computable using* $\mathrm{poly}(\ell(\max(x, y)))$ *space. For all* x, *the integer division in half* $\lfloor x/2 \rfloor$ *is also computable using polynomial space* $\mathrm{poly}(\ell(x))$.

Proof. Obviously, if there is a polynomial-time algorithm with an oracle for some level of the polynomial-time hierarchy, then the computation can be performed using polynomial space. The desired polynomial-time algorithms with an oracle for some level of the polynomial-time hierarchy exist according to Theorem 1. In fact, one can use an oracle for NP. □

Next, let us consider $(0, \min)$-structures without the addition operation. Our results are easily transferred to min-plus semirings as well as other tropical algebras.

Theorem 3. *Given a structure* $\mathfrak{M} = (\mathrm{Bin}(\omega); 0, \min)$ *isomorphic to the* $(0, \min)$*-structure* $\mathfrak{N}$ *of natural numbers with both* 0 *and* $\min()$ *defined as usual. For all elements* x *and* y *from* $\mathfrak{M}$*, let their minimum* $\min(x, y)$ *can be computed using* $\mathrm{poly}(\ell(\max(x, y)))$ *time depending only on the length of the greatest element* $\max(x, y)$*. There are both* $2^{\mathrm{poly}(\ell(x))}$*-time computable isomorphisms* $f : \mathfrak{M} \to \mathfrak{N}$ *and* $g : \mathfrak{N} \to \mathfrak{M}$ *so that* $f \circ g = g \circ f = \mathrm{id}$.

Proof. Let us apply the method from the proof of Theorem 1. For any $k \leq x$, the element $\min(x, k)$ is computed in $\mathrm{poly}(\ell(x))$ time. Thus, the inequality $\ell(k) \leq \mathrm{poly}(\ell(x))$ holds.

Therefore, the computation of all elements $k \leq x$ is reduced to pairwise comparisons of those elements having bounded lengths $\ell(k) \leq \mathrm{poly}(\ell(x))$. By ordering the obtained elements, one can construct a restriction of the desired isomorphism $f : \mathfrak{M} \to \mathfrak{N}$ to an initial segment consisting of elements $k \leq x$. In particular, the image of the element x is found. This defines a unique isomorphism $f : \mathfrak{M} \to \mathfrak{N}$. Moreover, the image $f(x)$ has an upper bound $2^{\mathrm{poly}(\ell(x))}$. Thus, it is $2^{\mathrm{poly}(\ell(x))}$-time computable.

Conversely, for a natural number $y \in \mathfrak{N}$, we need to compute the element $x \in \mathfrak{M}$ so that $y = f(x)$. For $y \in \mathfrak{N}$, we calculate the restriction of an already given isomorphism $f : \mathfrak{M} \to \mathfrak{N}$ to the initial segment such that its image contains the element y. Let us prove that the length of the preimage of y is bounded by a polynomial.

There is a monotonically increasing univariate polynomial p such that for any two elements $k, z \in \mathfrak{M}$ satisfying the inequality $k \leq z$, their lengths satisfy the inequality $\ell(k) \leq p(\ell(z))$. But since we require the polynomial p to be monotonic, it may differ from the polynomial upper bound on the time complexity. Let both $x \in \mathfrak{M}$ and $y = f(x) \in \mathfrak{N}$ satisfy the inequality $\ell(x) > \ell(y)$. Since the isomorphism f is bijective, in accordance with the Dirichlet principle, there is an element $z \in \mathfrak{M}$ such that $\ell(z) \leq \ell(y)$ and $x \leq z$. Then the inequalities $\ell(x) < p(\ell(z)) < p(\ell(y))$ hold. Otherwise, the inequality $\ell(x) \leq \ell(y)$ holds. Consequently, to search for an element $x \in \mathfrak{M}$ by its image $y \in \mathfrak{N}$, it is sufficient to compute the images of all elements $k \in \mathfrak{M}$ having lengths less than $p(\ell(y))$. So, one can compute the isomorphism $g : \mathfrak{N} \to \mathfrak{M}$. □

4 Discussion

Theorem 1 does not contradict the result obtained by Alaev because we assume a non-trivial restriction on the computational complexity of the min() function. Over the min-plus semiring of natural numbers with the usual operations, both upper bounds $\mathrm{poly}(j, k)$ and $\mathrm{poly}(\max(j, k))$ do not differ significantly from each other. Nevertheless, when passing to other isomorphic

structures, the distinction between these boundaries becomes important. Thus, we showed the need for very careful definition of complexity boundaries depending on the signature.

Furthermore, over the linearly ordered ring of integers, the minimum of two elements $\min(x, y)$ cannot be calculated using space bounded by any function of the length of $\max(x, y)$. Consequently, it is impossible to transfer from natural numbers to integers without non-trivial additional restrictions.

A structure is called locally finite, when any finite set of elements generates a finite substructure. For some locally finite structure, Theorem 3 implies a property close to weak p-categoricity introduced by Alaev [10]. There is also no contradiction because we use the refined complexity upper bound as well as impose additional restrictions on the universe of structures.

Our proof of Theorem 1 essentially uses the coincidence of the universe with $\mathrm{Bin}(\omega)$. In particular, one can perform an exhaustive search over a large finite set. Such an exhaustive search is possible using a polynomial-space algorithm, but no polynomial-time algorithm has been found. Further improvement turns out to be related to the well-known problem whether the polynomial-time hierarchy collapses. Unfortunately, there is no reason to hope for a quick solution. However, in practice, the execution of many calculations is limited precisely by the memory size [20]. Thus, a polynomial-space algorithm can be useful.

The class of all functions that are polynomial-time computable with an oracle for some level of the polynomial-time hierarchy seems to be a suitable alternative for the class consisting of polynomial-time computable functions.

It is also important that the signature contains a binary function or more complicated one. Contrariwise, if the signature contains only constants, unary functions and arbitrary predicates, then our method fails.

In the same way, one can consider polynomial-time computable $(0, \dot{-}, +)$-structures, where the truncated difference $x \dot{-} y$ is computed in polynomial time $\mathrm{poly}(\ell(\max(x, y)))$ depending only on the length of the greatest element $\max(x, y)$. Of course, over the structure, the linear order can be defined. The inequality $x \leq y$ is equivalent to equality $x \dot{-} y = 0$.

References

[1] S. Goncharov, A. Nechesov, *Solution of the problem $P = L$*, Mathematics, **10**:1 (2022), 113.

[2] S. Goncharov, A. Nechesov, *Polynomial analogue of Gandy's fixed point theorem*, Mathematics, **9**:17 (2021), 2102.

[3] E. Neumann, A. Pauly, *A topological view on algebraic computation models*, Journal of Complexity, **44** (2018), 1–22.

[4] I.V. Latkin, A.V. Seliverstov, *On computations over ordered rings*, Siberian Electronic Mathematical Reports, **19**:2 (2022), 1054–1076.

[5] D. Cenzer, J. Remmel, *Polynomial-time versus recursive models*, Annals of Pure and Applied Logic, **54**:1 (1991), 17–58.

[6] D. Cenzer, J. Remmel, *Polynomial-time Abelian groups*, Annals of Pure and Applied Logic, **56** (1992), 313–363.
[7] P.E. Alaev, *Polynomially computable structures with finitely many generators*, Algebra and Logic, **59**:3 (2020), 266–272.
[8] P.E. Alaev, *Finitely generated structures computable in polynomial time*, Siberian Mathematical Journal, **63**:5 (2022), 801–818.
[9] D. Cenzer, R.G. Downey, J.B. Remmel, Z. Uddin, *Space complexity of Abelian groups*, Archive for Mathematical Logic, **48** (2009), 115–140.
[10] P.E. Alaev, *Structures computable in polynomial time. I*, Algebra and Logic, **55**:6 (2017), 421–435.
[11] D. Cenzer, J.B. Remmel, *Complexity and categoricity*, Information and Computation, **140**:1 (1998), 2–25.
[12] I. Kalimullin, A. Melnikov, A. Montalban, *Punctual definability on structures*, Annals of Pure and Applied Logic, **172** (2021), 102987.
[13] N. Greenberg, M. Harrison-Trainor, A. Melnikov, D. Turetsky, *Non-density in punctual computability*, Annals of Pure and Applied Logic, **172** (2021), 102985
[14] I. Kalimullin, A. Melnikov, K.M. Ng, *Algebraic structures computable without delay*, Theoretical Computer Science, **674** (2017), 73–98.
[15] M.V. Zubkov, I.Sh. Kalimullin, A.G. Mel'nikov, A.N. Frolov, *Punctual copies of algebraic structures*, Siberian Mathematical Journal, **60**:6 (2019), 993–1002.
[16] N.A. Bazhenov, I.Sh. Kalimullin, *Punctual categoricity spectra of computably categorical structures*, Algebra and Logic, **60**:3 (2021), 223–228.
[17] C. Calude, S. Marcus, I. Tevy, *The first example of a recursive function which is not primitive recursive*, Historia Mathematica, **6**:4 (1979), 380–384.
[18] V. Selivanov, S. Selivanova, *Primitive recursive ordered fields and some applications*, Computability, **12** (2023), 71–99.
[19] *N.K. Krivulin*, *On the solution of a two-sided vector equation in tropical algebra*, Vestnik St. Petersburg University, Mathematics, **56** (2023), 172–181.
[20] O.A. Zverkov, A.V. Seliverstov, *Effective lower bounds on the matrix rank and their applications*, Programming and Computer Software, **49**:5 (2023), 441–447.

Alexandr Vladislavovich Seliverstov
Institute for Information Transmission Problems
of the Russian Academy of Sciences,
Bolshoy Karetny per., 19, build.1,
127051, Moscow, Russia
Email address: slvstv@iitp.ru