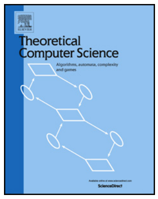
Doctopic: Algorithms, automata, complexity and games ARTICLE IN PRESS TCS:11076

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs



Algebraic structures computable without delay ☆

Iskander Kalimullin, Alexander Melnikov *, Keng Meng Ng

A R T I C L E I N F O

Article history:
Received 27 January 2016
Received in revised form 13 December 2016
Accepted 20 January 2017
Available online xxxx
Communicated by A. Kucera

Keywords:
Polynomial structures
Automatic algebra
Primitive recursion

A B S T R A C T

In this article we suggest a new systematic approach to studying algorithms on algebraic structures via primitive recursion. The approach is designed to fill the gap between abstract computable structure theory and feasible (in the sense of polynomial-time, computational or automatic) algebra.

1. Introduction

In the early 1960's, Mal'cev [33] and Rabin [41] independently gave a general definition of an algorithmically presented algebraic structure.

Definition 1.1 *(Mal'cev, Rabin).* A structure with domain ω (natural numbers) is *computable* if its operations and relations are uniformly Turing computable.

If a countably infinite $\mathcal{A}$ is isomorphic to a computable $\mathcal{B}$, then we say that $\mathcal{B}$ is a computable presentation, a computable copy, or a constructivization of $\mathcal{A}$. The notion of a computably presented structure united and extended the earlier definitions of an explicitly presented field [49] and of a "recursively presented" group with a solvable word problem [25].

Much work has been done on computable groups [27,20,10], fields [14,37,36], Boolean algebras [42,19], linear orders [11], computable aspects of model theory [23,35,2,31] and the degree-theoretic properties of algebraic structures [45,50,15]. Investigations of this sort form a field known under the names of *computable structure theory* and *effective algebra*, see books [3,13] and surveys [21]. From a purely technical point of view computable structure theory is more closely related to definability by infinitary formulae [3], $\mathbb{HF}$-definability [14], degree theory [47] and reverse mathematics [44], rather than to any actual computational applications. Nonetheless, computable structures in some natural algebraic classes tend to have computationally "feasible" presentations. We still do not have a satisfactory formal explanation of this phenomenon. Thus we have the following non-trivial question:

When does a computable algebraic structure have a feasible presentation?

☆ We are thankful to Rod Downey and Pavel Alaev for fruitful discussions. The third author is partially supported by the grant MOE-RG26/13. The second author was partially supported by Marsden Fund of New Zealand and Massey Research Fund.

* Corresponding author.
E-mail address: alexander.g.melnikov@gmail.com (A. Melnikov).

http://dx.doi.org/10.1016/j.tcs.2017.01.029

Please cite this article in press as: I. Kalimullin et al., Algebraic structures computable without delay, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.01.029

What does it mean for an infinite algebraic structure to have a feasible presentation? Different branches of effective mathematics suggest different rigorous answers to this question. For example, we could restrict ourselves to algebraic structures that are presented by finite automata [28–30]. Automatic structures have a number of nice properties, including quick computational characteristics and decidability of their first-order theories (and even beyond), but automatic structures tend to be rare. For example, a countably infinite Boolean algebra is finite-automatic iff it is isomorphic to the interval-algebra of the ordinal $\omega \cdot n$, see [30]. Although having a finite-automatic presentation of a structure is highly desirable, it is usually quite difficult to see whether a non-trivial algebraic structure has such a presentation. For instance, using deep results borrowed from additive combinatorics [16], Tsankov [48] has showed that the group of the rationals $(\mathbb{Q}, +)$ cannot be represented by a finite automaton. The result of Tsankov settled a long-standing conjecture of Khoussainov and Nerode (see e.g. [29]). Despite these difficulties, there have been a number of deep works on finite automatic structures [28,30], especially on finite-automatic groups [12,39,4,38].

Cenzer, Remmel, Downey and their co-authors developed a more relaxed and general approach, see survey [7]. More specifically, a computable presentation is *feasible*, or *polynomial time*, if the operations and relations of the structure are polynomial time computable in the length of input. Clearly this definition depends on how we represent the domain ω but we shall not discuss these subtleties here (see [7]). There is a relatively large body of research on polynomial time algebra (e.g., [7,8,6,22]), and some of these results relate computable structure theory with feasible algebra. Nonetheless, there is still a significant gap between these two topics, and deep results relating computable structure theory and polynomial time algebra are rare. In this paper we suggest a *systematic* approach designed to fill this gap.

1.1. From computable to feasible

When considering computable structures we allow algorithms to be extremely inefficient. For example, we may use an unbounded search through ω as long as we can *prove* that it will halt. More formally, our algorithms do not even have to be *primitive recursive*. Nonetheless, in several common algebraic classes we can show that every computable structure has a polynomial-time computable copy. These classes include linear orders [22], broad subclasses of Boolean algebras [5], some commutative groups [8,6], and other structures [7]. Interestingly, many known proofs of this sort (e.g., [7,8,6,22]) are essentially focused on making the operations and relations on the structure primitive recursive, and then observing that we get a polynomial-time presentation almost for free. It appears that primitive recursion plays a rather important intermediate role in such proofs. This thesis is also supported by a number of *negative* results in the literature. Indeed, to illustrate that a structure has no polynomial time computable copy, it is sometimes easiest to argue that it does not even have a copy with primitive recursive operations, see e.g. [8]. For this technical reason Cenzer and Remmel [7] came up with the following general definition.

Definition 1.2. An algebraic structure is *primitive recursive* if its domain is a primitive recursive subset of ω and the operations and relations of the structure are (uniformly) primitive recursive.

Our initial thought was that primitive recursive structures would be an excellent candidate for an intermediate class between computable structures and feasible structures. However, we very soon realized that the above definition is a bit too relaxed. In a primitive recursive structure, we may see new elements appearing in the structure extremely slowly; the principal function of the domain might not be primitive recursively bounded. This feature can be exploited to show that most computable structures have primitive recursive copies. For example, as observed by Alaev (personal communication), every computable structure whose finitely generated substructures are finite has a primitive recursive copy. Indeed, we can simply keep elements of ω out of the domain until we wait for a larger finite substructure to be revealed in the computable copy. In particular, any computable relational structure in a finite language admits a primitive recursive copy.[1] This fact strongly suggests that primitive recursive structures are not "truly" primitive recursive, i.e. they seem too close to (general) computable structures to be a good intermediate notion.

We suggest that a truly "non-delayable" computable presentation must minimally satisfy the following definition:

Definition 1.3. A countable structure is fully primitive recursive (fpr) if its domain is ω and the operations and predicates of the structure are (uniformly) primitive recursive. We also fix the convention that all finite structures are fully primitive recursive by allowing the domain to be a finite initial segment of ω.

The reader should note that the situation here is quite different from computable structures where the domain can typically be assumed an arbitrary computable subset of ω. Indeed, a fully primitive recursive structure must reveal itself without any unbounded delay. One of our main results (Theorem 3.2, to be stated) combined with the observation of Alaev discussed above imply:

Fact 1.4. There exist primitive recursive structures that have no fully primitive recursive presentation.

[1] As noted by the anonymous referee, this observation was known to Remmel and Nerode long before Alaev.