ELIMINATING UNBOUNDED SEARCH IN COMPUTABLE ALGEBRA

ALEXANDER G. MELNIKOV

Abstract. Klaimullin, Melnikov and Ng [KMNa] have recently suggested a new systematic approach to algorithms in algebra which is intermediate between computationally feasible algebra [CR91, KNRS07] and abstract computable structure theory [AK00, EG00]. In this short survey we discuss some of the key results and ideas of this new topic [KMNa, KMNc, KMNb, Ala]. We also suggest several open problems.

1. intorduction

What does it mean for an infinite algebraic structure to be computable? Which algebraic structures admit an algorithmic presentation? These questions have occupied mathematicians for over a century. In the 1930's, van der Waerden [vdW30] made several algorithmic conjectures in field theory, but he was unable to verify his claims at his time due to the lack of computability-theoretic machinery. A few decades later, Turing, Kleene, Church, Markov and others established and developed the abstract theory of computable functions. Using these tools Fröhlich and Shepherdson [FS56] formally clarified the early ideas of van der Waerden. Shortly after Fröhlich and Shepherdson, Mal′cev [Mal61] and independently Rabin [Rab60] initiated a systematic development of computable structure theory [EG00, AK00]. The main objects of computable structure theory are countably infinite structures that can be computed by a Turing machine:

Definition 1.1 (Mal'cev, Rabin)**.** An algebraic structure $\mathcal{A}$ is *constructive* or *computable* if its universe is the set of natural numbers $\mathbb{N}$, and the operations and relations on $\mathcal{A}$ are (Turing) computable.

For example, every finitely generated group with algorithmically solvable Word Problem [Hig61] has a computable isomorphic copy. In both combinatorial group theory [Hig61] and computable structure theory [AK00, EG00] algorithms are allowed to be computationally inefficient. It often does not make any difference, since one of the main aims of such studies is to show that some problems – such as the Word Problem for f.g. groups – have no algorithmic solution at all, let alone a computationally feasible solution [Nov55, Boo59, Hig61]. Also, Turing computability can be viewed as a formalisation of one's constructive approach to algebra [EG00]. Strong connections with logic, and with degree theory and definability in particular, make computable structure theory technically appealing and elegant [Rog87, AK00]. Algorithmic investigations in algebra are related to other seemingly distant areas of pure mathematics such as topological group theory and model theory (e.g., [MM, Mon13]).

In contrast with (Turing) computable structure theory, *automatic* and *polynomial-time* algebra put resource bounds and other restrictions on algorithms representing algebraic structures. Automatic algebra studies algebraic structures that are presented by (typically finite) automata [KN94, KN08, KNRS07]. Automatic structures have a number of nice properties including quick computational characteristics, but automatic structures tend to be rare. For example, Tsankov [Tsa11] has showed that the group $(\mathbb{Q}, +)$ is not automatic. In spite of these difficulties, there has been enough deep work on finite automatic structures [KN94, KNRS07], most notably on finite-automatic groups [ECH+92, NT08, BS11, NS07]. Cenzer, Remmel, Downey and their co-authors developed a more relaxed approach [CR91, CR92, CDRU09, Gri90] via polynomial-time algorithms. A countably infinite algebraic structure is *polynomial-time* if there exists a coding of its domain which makes the operations and relations polynomial time computable (in the length of the input, see survey [CR91]). As we will discuss later, in many cases one can show that a (Turing) computable algebraic structure has a polynomial copy (e.g., [CDRU09, CR, Gri90]), but this phenomenon is not yet understood.

One would expect that computable structure theory and computationally feasible algebra should have significant overlaps, but it is not quite the case. Kalimullin, Melnikov and Ng [KMNa, KMNc, KMNb] have initiated a systematic development of a theory the main purpose of which is to fill this gap. (Independently, Alaev [Ala] has suggested an alternate approach.) In this brief survey we explain the key ideas and results from these recent works [KMNa, KMNc, KMNb, Ala]. An expert working in computable structure theory may find some of the key results discussed below rather counter-intuitive. Although the new topic is in its infancy, it already has technical depth and offers seemingly challenging problems, some of which will be posed below.

2. Primitive recursion and computability without delay

We open this section with two elementary examples illustrating the power of unbounded search.

Example 2.1. Clearly, there exists a (Turing) computable 2-colouring of any computable infinite tree. Simply wait for a node v to get path-connected to the root. If we put any (computable and total) restriction on the waiting time, then we will need infinitely many colours in general.

Example 2.2. Suppose $(G, \cdot)$ is a (Turing) computable group. Then the operation $a \to a^{-1}$ is also Turing computable. We simply search through the domain of G until we find a g such that $a \cdot g = e$. Note this elementary fact no longer holds if we cannot use the unbounded search.

Eliminating all unbounded loops from Turing computability gives us the notion of a primitive recursive function.

Definition 2.3 (Essentially Dedekind [Ded])**.** A function $f : \mathbb{N} \to \mathbb{N}$ is *primitive recursive* if f can be generated from the basic functions $s(x) = x + 1$, $o(x) = 0$, $I^n_m(x_1, \dots, x_n) = x_m$ by composition and the primitive recursion operator $h = \mathcal{P}(f, g)$:

$$h(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n),$$
$$h(x_1, \dots, x_n, y + 1) = g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)).$$

Remark 2.4. The restricted Church-Turing thesis for primitive recursive functions say that a function is primitive recursive iff it can be described by an algorithm that uses only bounded loops. For example, we need to eliminate all instances of WHILE ... DO, REPEAT ... UNTIL, and GOTO in a PASCAL-like language. It is easy to see that all polynomial-time functions are primitive recursive: simply put exponential bounds on all loops throughout the definition.

Computable structure theory and combinatorial group theory often rely on algorithms that are not even primitive recursive, let alone polynomial-time. Nonetheless, in several common algebraic classes we can show that all computable structures have polynomial-time computable copies. These classes include linear orders [Gri90], broad subclasses of Boolean algebras [CR], some commutative groups [CR92, CDRU09], and other structures [CR91]. As was noted in [KMNa], many known proofs of this sort (e.g., [CR91, CR92, CDRU09, Gri90]) are essentially focused on making the operations and relations on the structure primitive recursive, and then observing that the presentation that we obtain is in fact polynomial-time.

It appears that primitive recursion plays a rather important intermediate role in transforming (Turing) computable structures into feasible structures. This thesis is also supported by a number of negative results in the literature. Indeed, to illustrate that a structure has no polynomial time copy, it is sometimes easiest to argue that it does not even have a copy with primitive recursive operations, see e.g. [CR92]. It is thus natural to investigate into those structures that admit a presentation with primitive recursive operations:

Definition 2.5 ([KMNa])**.** A countable structure is *fully primitive recursive* (fpr) if its domain is $\mathbb{N}$ and the operations and predicates of the structure are (uniformly) primitive recursive.

(We also agree that all finite structures are fpr.) The main intuition is that we need to define more of the structure "without delay". We thus informally call fpr structures *computable without delay*.

One may ask whether the domain of a fpr A could be a primitive recursive subset of $\mathbb{N}$ as in [CR91], not the whole $\mathbb{N}$. This means that the structure could reveal its domain with an arbitrary delay. Such presentations upon a primitive recursive subset of $\mathbb{N}$ are called *primitive recursive* [CR92]. The situation here is quite different from computable structure theory:

Fact 2.6 (Follows from [Ala], see [KMNa])**.** There exist primitive recursive structures that have no fully primitive recursive presentation.

Our main goal is the complete elimination of all unbounded loops from algorithmic presentations of structures. In particular, we should not allow the domain to be revealed with an unbounded delay. Thus, a "truly primitive recursive" algebraic structure should (minimally) satisfy Def. 2.5 of a fpr structure. Note that one might impose further restrictions on fpr presentations. We will briefly discuss one such possible strengthening of Def. 2.5 in the next section.

The notion of a fpr structure is intermediate between computable structures and polynomial-time structures. It usually takes some work to produce a fpr copy of a computable structure (if it exists at all), and there are enough natural examples of computable structures that do not have a fpr copy. Similarly, not all fpr structures admit polynomial-time copies, but a good portion of our results can be extended

to polynomial-time structures. The rest of the survey is focused on the systematic development of the theory of fpr structures.

3. Existence of a fpr copy

Which computable structures admit fpr presentations?

Theorem 3.1. *In each of the following classes, every computable structure has a fully primitive recursive presentation:*

(1) Equivalence structures [CR91].
(2) Linear orders [Gri90].
(3) Torsion-free abelian groups [KMNa].
(4) Boolean algebras [KMNa].
(5) Abelian p-groups [KMNa].

Proof idea. We use various structural properties specific to the class to predict the behaviour of the structure in certain locations even before the structure has revealed itself there. For example, in (3) we first produce a computable presentation of the group with a computable maximal linearly independent set [Dob, Nur74], and then we use the maximal free subgroup upon this set as a "safe" location. The case of Boolean algebras (4) is more interesting. The proof is not uniform and it goes through several cases. In the case of an atomic algebra we use a priority construction, the old theorem of Remmel [Rem81], and tree-presentations of BA's [Gon97] to produce a Π^0_3-isomorphic fpr copy. □

The reader perhaps thinks that most common classes will have the nice property from Theorem 3.1, but this is not the case.

Theorem 3.2. *In each of the following classes, there exists a computable structure that does not admit a fpr presentation*

(1) Torsion abelian groups ([CR92]).
(2) Archimedean ordered abelian groups ([KMNa]).
(3) Undirected graphs ([KMNa]).

Proof idea. In (2) we produce an Archimedean group of rank 2 that (essentially) encodes a computable real which does not have a primitive recursive rapid approximation. The proof uses that every such group is computably categorical, so we need to diagonalise only against computable isomorphisms. The proofs of (1) and (3) are brute-force diagonalisation arguments, but (3) is a lot more subtle. In the proof we need to monitor all potential fpr copies at once and force them to reveal themselves at certain locations used for diagonalisation. □

Parts (1) and (2) of Thm. 3.2 contrast with (3) and (5) of Thm. 3.1, and (3) of Thm. 3.2 refutes the (natural) conjecture that every relational computable structure has a fpr copy. Note that Thm. 3.1 and Thm. 3.2 (combined) confirm the intermediate nature of fpr structures.

Although there are some observable patterns in the proofs of Thm. 3.2 and Thm. 3.1, it is not clear whether there is any meaningful and general enough sufficient condition for a computable structure to have (or not have) a fpr copy. We leave open:

Question 3.3. *Fix the listing $M_0, M_1, \ldots$ of all partial computable structures. What is the complexity of the index set $\{e : M_e$ has a fpr copy$\}$?*

We might also want to restrict this question to special classes such as undirected graphs.

3.1. **Strongly fpr structures.** In this subsection we briefly discuss one possible strengthening of the notion of a fpr structure. In the next section we will essentially show that almost all structures from Theorem 3.1 posses complicated fpr copies. Nonetheless, Theorem 3.1 tends to produce the most "boring" fpr copies of a structure which reveal themselves quickly only at some predictable locations. Indeed, we perhaps want our structure to be rapidly growing at every location. One way of formalising this intuition is to use the natural primitive recursive analogy of 1-decidability.

Definition 3.4. [KMNa] A fpr structure is *strongly primitive recursive* of it possesses a primitive recursive Skolem function.

That is, there exists a primitive recursive Φ such that

$$\Phi(\bar{c},\phi) = \begin{cases} -1, & \text{if } \mathcal{A} \not\models \exists x \phi(\bar{c},x), \\ y, & \text{such that } \mathcal{A} \models \phi(\bar{c},y), \end{cases}$$

where $\bar{c} \in \mathcal{A}$ and ϕ (the Gödel number of) a quantifier-free formula in the language of the structure. We note that this approach resembles the earlier notion of a *honest witness* due to Cenzer and Remmel [CR91].

Example 3.5. [KMNa] The following structures have strongly primitive recursive copies:

- The additive groups $\mathbb{Z}$ and $\mathbb{Q}$ and their direct sums.
- The countable atomless Boolean algebra.
- The order-type ω.

There exist computable structures that have no 1-decidable presentation. Such examples can be found among linear orders [Dow98] and Boolean algebras [Gon97]. Theorem 3.1 guarantees the existence of fpr presentations of these counter-examples. It follows that each of these structures has no p.r. presentation. There also exists a fpr 1-decidable structure that has no strongly primitive recursive presentation[KMNa]. (See also [Ala] for a similar approach.)

4. Uniqueness of a fpr copy

Recall that a structure is *computably categorical* or *autostable* if it has a unique computable copy up to computable isomorphism [AK00, EG00]. There has been a lot of work on computably categorical strucutres [KS99, EG00, Gon80, Smi81, LaR77, Nur74, HKS03].

Note that the inverse of a primitive recursive function does not have to be primitive recursive. Thus it is reasonable to define an isomorphism f to be *fully primitive recursive* (fpr) if f and f^{-1} are both primitive recursive. Fully primitive recursive isomorphisms preserve all properties of fpr structures at the right (primitive recursive) level.

Definition 4.1. [KMNa] A fully primitive recursive structure $\mathcal{A}$ is *fpr categorical* if it has a unique fully primitive recursive presentation up to fully primitive recursive isomorphism.

Example 4.2. [KMNa]

(1) The additive group $\mathbb{V}_p \cong \bigoplus_{i\in\omega} \mathbb{Z}_p$ is fpr-categorical

Proof. Indeed, suppose $\mathcal{A}$ and $\mathcal{B}$ are fpr presentations of $\mathbb{V}_p$. Note that given a tuple $\bar{a}$ in $\mathcal{A}$ we can primitively recursively choose a maximal $\mathbb{Z}_p$-independent sub-tuple in $\bar{a}$. We may assume that we always choose the lexicographically smallest independent tuple among all such independent sub-tuples of $\bar{a}$. We first explain how we can define a primitive recursive isomorphism $f : \mathcal{A} \to \mathcal{B}$, and then we explain how we make sure that f^{-1} is primitive recursive as well. Suppose $f : \bar{a}_s \to \bar{b}_s$ has already been defined, where $\bar{a}_s$ is the longest initial segment of $\mathcal{A}$ on which f has been defined. To define $f(a)$ on the next element a of $\mathcal{A}$ extending $\bar{a}_s$, first see whether a is dependent on $\bar{a}_s$. If yes, then suppose $a = \sum_j m_j a_j$, where the a_j range over $\bar{a}_s$ and m_j over $\mathbb{Z}_p$. In this case set $f(a) = \sum_j m_j f(a_j)$. If no, then we look through at most $p^{card(b_s)}$ first elements of $\mathcal{B}$ and choose the first found element b independent over $\bar{b}_s$. To make sure f^{-1} is primitive recursive as well, we choose the longest initial segment $\bar{b}'_s$ for which f^{-1} has been defined and repeat the procedure above but now with a and f replaced by b and f^{-1}. □

(2) The dense linear order $(\mathbb{Q}, <)$ without end points is *not* fpr-categorical

Proof. We produce fpr copies $\mathcal{A}$ and $\mathcal{B}$ and diagonalize against all pairs of primitive recursive $(p_i, p_j)_{i,j\in\omega}$, where p_j plays the role of a potential p_i^{-1}. We explain how to diagonalize against the first pair (f, g). Begin by growing increasing chains $a_0 < a_1 < a_2 < \cdots$ and $b_0 < b_1 < b_2 < \cdots$ in $\mathcal{A}$ and $\mathcal{B}$ respectively, and wait for $f(a_0)$ to halt. When we see $f(a_0)\downarrow = b_i$, keep growing $\mathcal{A}$ and $\mathcal{B}$ in the same way, but in $\mathcal{B}$ we add one additional point $b^* < b_0$. Now wait for g to converge on b^*. In order for $g = f^{-1}$ we must have $g(b^*) < a_0$ in $\mathcal{A}$, but there are currently no elements in $\mathcal{A}$ with this property, so we win against the pair (f, g).

For a general requirement suppose we have built $\hat{a}_0 < \hat{a}_1 < \cdots < \hat{a}_k$ in $\mathcal{A}$ and $\hat{b}_0 < \hat{b}_1 < \cdots < \hat{b}_k$ in $\mathcal{B}$. We now wish to attack (p_i, p_j). Begin as above by growing $\hat{a}_k < a_0 < a_1 < \cdots$ and $\hat{b}_k < b_0 < b_1 < \cdots$. We wait for $p_i(\hat{a}_0)\downarrow$. When we see this, we add a new point $b^* < \hat{b}_0$ and wait for $p_j(b^*)\downarrow$. Since each pair (p_i, p_j) are total functions we can finish each pair in this way before moving on to the next pair. In between satisfying each requirement, we can extend $\mathcal{A}$ to the left and make progress towards making $\mathcal{A}$ and $\mathcal{B}$ dense. □

(3) The successor structure $\mathcal{S} = (\omega, S)$, where $S(x) = x + 1$, is *not* fpr-categorical

Proof. Build two fpr copies $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{S}$ and enumerate all potential primitive recursive isomorphisms p_e. The copy $\mathcal{B}$ is the standard copy, and $\mathcal{A}$ will be used to diagonalize against primitive recursive isomorphisms. The strategy for diagonalizing p_e is the following. Pick a fresh witness $x \in \mathcal{A}_s$ which currently has no predecessor, and wait for $p_e(x)$ to converge. While waiting we grow two independent chains, one with the least element $0_{\mathcal{A}}$ and the other with the least element x. That is, introduce distinct elements $S(0_{\mathcal{A}}), S^2(0_{\mathcal{A}}), S^3(0_{\mathcal{A}}), \cdots$ and distinct elements $S(x), S^2(x), S^3(x), \cdots$. When $p_e(x)$ converges, declare x to be $S^n(0_{\mathcal{A}})$ for the least $n > p_e(x)$. □

Even though Theorem 3.1 typically produces the most "boring" fpr presentations in each class, Theorem 4.3 below says that almost all structures in these classes have complex ("irregular", "unpredictable") fpr presentations.

Theorem 4.3 ([KMNa])**.**

(1) An equivalence structure S is fpr-categorical iff it is either of the form $F \cup E$, where F is finite and E has only classes of size 1, or S has finitely many classes at most one of which is infinite.
(2) A linear order is fpr-categorical iff it is finite.
(3) A Boolean algebra is fpr-categorical iff it is finite.
(4) An abelian p-group is fpr-categorical iff it has the form $F \oplus \mathbb{V}$, where $p\mathbb{V} = \mathbf{0}$ and F is finite.
(5) A torsion-free abelian group is fpr-categorical iff it is the trivial group $\mathbf{0}$.

Informal Discussion. In some simple cases we can appeal to Thm. 3.1 and combine it with known facts from computable structure theory. For example, suppose a Boolean algebra $\mathcal{B}$ has an atomless element and is not computably categorical. Although Theorem 3.1 is not uniform in general, it is uniformly computable for such BAs. It follows that we can produce two fpr copies of $\mathcal{B}$ that are not even computably isomorphic.

Nonetheless, some cases require a direct proof. For instance, in the case of an atomic BA we cannot appeal to Thm. 3.1 directly since the isomorphism between $\mathcal{B}$ and its fpr copy is not even $0''$ in general. Instead, in this case we need to simultaneously build two fpr copies of $\mathcal{B}$ and diagonalise against all potential fpr-isomorphisms. Although the proof is not hard, it does require some care. □

Note that Theorem 4.3 resembles the following result of Khoussainov and Nerode [KN94]: A structure is automatically categorical iff it is finite.

According to Def 4.1, every fpr-categorical structure must have a fully primitive recursive (thus, computable) copy. Theorem 4.3 suggests that fpr-categorical structures are necessarily computably categorical. Surprisingly, this is not the case:

Theorem 4.4 ([KMNa])**.** *There exists a fpr-categorical structure which is not computably categorical.*

Informal Discussion. The proof of Theorem 4.4 is quite combinatorially involved. We build the structure $\mathcal{A}$ carefully and force any fpr isomorphic copy to be fpr-isomorphic to it. The rigidity helps to make the inverse of each such isomorphism primitive recursive. Also, for this purpose we introduce a "local coordinate system" that will allow us to recognise a local part of the structure without looking through the whole structure. Finally, we combine these strategies with a diagonalisation strategy. For that, we produce a computable copy $\mathcal{B}$ of $\mathcal{A}$ and kill off all $\phi_e : \mathcal{A} \cong \mathcal{B}$. We heavily rely on the fact that we *can* delay the computation in $\mathcal{B}$. □

The structure witnessing Theorem 4.4 is rigid and is in a finite language consisting of four unary function symbols.

Question 4.5. *Which of the common algebraic classes (such that groups, fields, integral domains, ...) contain examples of fpr-categorical but not computably categorical structures?*

5. The fpr-degrees of a structure

Note that the inverse of a primitive recursive function does not have to be primitive recursive. This feature makes the situation with fpr structures very different form computable structure theory, as it leads to a reduction:

Definition 5.1. [KMNc] Let $\mathbb{FPR}(\mathcal{A})$ be the collection of all fpr presentations of a countably infinite structure $\mathcal{A}$. For $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{FPR}(\mathcal{A})$, write $\mathcal{A}_1 \leq_{pr} \mathcal{A}_2$ if there exists a primitive recursive isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.

Clearly, $\leq_{pr}$ is reflexive and transitive. We write $A_1 \simeq_{pr} A_2$ if $A_1 \leq_{pr} A_2$ and $A_2 \leq_{pr} A_1$. In particular, we can look at the fpr-degrees of a given countably infinite structure $\mathcal{A}$.

Definition 5.2. [KMNc] The fully primitive recursive degrees of a countably infinite algebraic structure $\mathcal{A}$ is the quotient structure $\mathbf{FPR}(\mathcal{A}) = (\mathbb{FPR}(\mathcal{A}), \leq_{pr})/\simeq_{pr}$.

The fpr-degrees $\mathbf{FPR}(\mathcal{A})$ is a computability-theoretic invariant of $\mathcal{A}$ that encodes/reflects the non-primitive recursive content of the isomorphism type of $\mathcal{A}$. If $\mathcal{A}$ is fpr-categorical then $\mathbf{FPR}(\mathcal{A})$ contains a unique degree. Nonetheless, $A \simeq_{pr} B$ does not necessarily imply that there exists a fpr isomorphism $A \to B$ (it is easy to construct a counter-example).

Question 5.3. *Does* $|\mathbf{FPR}(\mathcal{A})| = \mathbf{1}$ *imply that* $\mathcal{A}$ *is fpr-categorical?*

As strange as it may sound, we still don't know the answer to the question above. It takes a quite a bit of effort to prove the rather satisfying:

Theorem 5.4 (M. and Ng, to appear)**.** *For every undirected graph* $\mathcal{G}$, $|\mathbf{FPR}(\mathcal{G})| = \mathbf{1}$ *implies that* $\mathcal{G}$ *is fpr-categorical.*

Informal discussion. In fact, we have proved more. TFAE:

(1) $|\mathbf{FPR}(\mathcal{G})| = \mathbf{1}$.
(2) $\mathcal{G}$ is fpr-categorical.
(3) Given any two f.p.r. copies $\mathcal{A} \cong \mathcal{B}$ of $\mathcal{G}$, there exist primitive recursive isomorphisms $f : \mathcal{A} \mapsto \mathcal{B}$ and $g : \mathcal{B} \mapsto \mathcal{A}$, and a primitive recursive function $t : \mathbb{N} \mapsto \mathbb{N}$ such that given $a \in \mathcal{A}$, either $Orb(a) = \{(gf)^n(a) : n \in \omega\}$ has size at most $t(a)$, or every permutation u of $Orb(a)$ can be extended to an automorphism of $\mathcal{G}$.

Note that given (3) we can run a primitive recursive back-and-forth construction to produce a fpr isomorphism between two fpr copies. First, check whether $Orb(a)$ has size $\leq t(a)$. If "yes" then match $Orb(a)$ with $Orb(f(a))$. Otherwise, if $Orb(a)$ has not yet closed after $t(a)$ steps, then do the back-and-forth on $Orb(a)$ and $Orb(f(a))$ essentially ignoring the rest of the structure. Unfortunately, the implication (1) $\to$ (2) is quite non-trivial. (It takes 3 pages just to describe the basic strategy.) □

Question 5.5. *Is there a structure* $\mathcal{A}$ *such that* $1 < |\mathbf{FPR}(\mathcal{A})| < \infty$*?*

6. A sub-hierarchy of computable categoricity

As we have seen, fpr-categoricity does not imply computable categoricity in general. Nonetheless, in common algebraic classes fpr-categorical structures tend to be computably categorical and trivial. Thus, we would like to learn more about fpr

structures and are not fpr-categorcial but are close to being fpr-categorical. In particular, for the classes mentioned in Thm 4.3 it makes sense to look at computably categorical structures from the primitive recursive point of view.

We would like to know more about fpr-degrees of computably categorical structures (see Def. 5.2). The definition below will also be of some interest and use.

Definition 6.1 ([Kri96])**.** A total computable function is *honest* if its graph is primitive recursive.

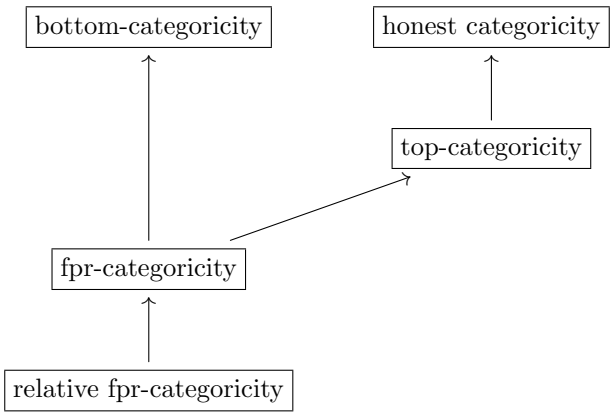
This means that we can see whether $f(x) = y$ for a given (x, y) without any unbounded delay. Thus, is some sense this is a non-deterministic version of primitive recursion. Clearly, if f is honest then so is f^{-1}, but the composition of two honest functions does not have to be honest.

In Theorem 4.3, all computably categorical fpr structures satisfy at least one of the properties defined below.

Definition 6.2 ([KMNc])**.** Let $\mathcal{A}$ be a fpr structure.

(1) $\mathcal{A}$ is **bottom-categorical** if $\mathbb{FPR}(\mathcal{A})$ has the $\leq_{pr}$-least element.
(2) $\mathcal{A}$ is **top-categorical** if $\mathbb{FPR}(\mathcal{A})$ has the $\leq_{pr}$-greatest element.
(3) $\mathcal{A}$ is **honestly categorical** if for each $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{FPR}(\mathcal{A})$ there exists a honest isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.
(4) $\mathcal{A}$ is **relatively fpr categorical** if there exists a pair of (oracle) primitive recursive schemata $\mathcal{P}_+$ and $\mathcal{P}_-$ such that for any isomorphic copy $\mathcal{B}$ (upon $\mathbb{N}$) of the fpr structure $\mathcal{A}$ the maps $\mathcal{P}_+^{\mathcal{B}} : \mathcal{A} \to \mathcal{B}$ and $\mathcal{P}_-^{\mathcal{B}} : \mathcal{B} \to \mathcal{A}$ are isomorphisms that are inverses of each other.

For example, every finitely generated structure in a finite functional language is bottom-categorical. Every computably categorical linear order or Boolean algebra is top-categorical. In many standard classes (including e.g. linear orders) honest categoricity is equivalent to the usual computable categoricity. All algebraically natural examples of fpr categorical structures are relatively fpr-categorical [KMNa] (see also (1) of Ex.4.2). We summarise all these notions in the diagram below.



Theorem 6.3 ([KMNc])**.** *The diagram above is proper*[1]*.*

[1]This means that all implications that are shown at the diagram above are proper. Furthermore, these implications (and their transitive closures) are the only implications that hold.

Informal Discussion. All implications are straightforward. Note that every relatively fpr-categorical structure is (relatively) computably categorical. It follows from Thm 4.4 that fpr-categoricity does not imply relative fpr-categoricity. The failures of the other missing implications are witnessed by priority constructions. □

An alternate way to compare computably categorical fpr structures uses relative computation and the primitive recursive jump $0'_{PR}$, which is defined as follows. A total function g is primitively recursively reducible to a function f ($g \leqslant_{PR} f$) if $g = \Phi^f$ for some f-primitive recursive schema Φ^f. This leads to the definitions of $g \equiv_{PR} f$ and $g <_{PR} f$ as well as the notion of primitive recursive (PR-) degree. For a total function f let $\{\Phi^f_n\}$ be the Gödel numbering of all f-primitive recursive schemata for functions with one variable. Define the primitive recursive jump f'_{PR} to be the function $f'_{PR}(n, x) = \Phi^f_n(x)$. It is easy to check that $f \leqslant_{PR} g \implies f'_{PR} \leqslant_{PR} g'_{PR}$ and $f <_{PR} f'_{PR}$.

Definition 6.4. A structure $\mathcal{A}$ is $0^{(n)}_{PR}$-categorical if it has a unique fpr-presentation, up to fully $0^{(n)}_{PR}$-isomorphism.

This is the direct analogy with the situation in computable structure theory where $0^{(n)}$-categorical structures have been studied extensively [AK00, Ash86].

Theorem 6.5 ([KMNc])**.** *For every $n > 0$ there exists a fully primitive recursive structure which is fully $0^{(n)}_{PR}$-categorical but not fully $0^{(n-1)}_{PR}$-categorical.*

Informal Discussion. In [KMNc], Kalimllin, M. and Ng have proved more than is stated in Theorem 6.5. The PR-degree any honest function can be realized as the fpr-degree of categoricity of some structure (we omit formal definitions). Remarkably, for any n, the PR-degree of $0^{(n)}_{PR}$ is honest. □

The structures witnessing Theorem 6.5 are finitely generated in a finite functional language and thus are bottom-categorical. It is unclear whether there is any interesting relationship between full $0^{(n)}_{PR}$-categoricity and any of the properties from Def. 6.2. In fact, we don't know much beyond Thm 6.3 and Thm 6.5 about the notions from Def 6.2 and full $0^{(n)}_{PR}$-categoricity.

7. Cantor's back-and-forth method revisited

As we know from Ex. 4.2 (2), $\eta = (\mathbb{Q}, <)$ has two fpr copies that are not fpr isomorphic. This means that Cantor's back-and-forth proof is no longer "effective" in the realm of computability without delay. Given the special role of back-and-forth proofs in mathematics, we would like to acquire a better understanding of this phenomenon by looking at the fpr-degree structures $\mathbf{FPR}(\mathcal{X})$ of various homogeneous $\mathcal{X}$ and comparing them (see Def. 5.2).

Recall that a structure $\mathcal{X}$ is homogeneous if every isomorphism $f : F_1 \to F_2$ between any two finitely generated substructures $F_1, F_2 \subseteq \mathcal{X}$ is extendable to an automorphism of $\mathcal{X}$. See [Mac11] for a survey on homogeneous structures.

Example 7.1. The following structures are homogeneous:

- η, the dense linear order without end-points.
- $\mathcal{R}$, the Random Graph.
- $\mathcal{P} \cong \bigoplus_{i\in\omega} \mathbb{Z}_p^\infty$, the universal divisible abelian p-group.
- $\mathcal{B} \cong I(\eta)$, the countable atomless Boolean algebra.

Apart from homogeneity, there is little in common between the three homogeneous structures $\eta, \mathcal{R}, \mathcal{P}$ from Ex. 7.1. Nonetheless, they do share essentially the same back-and-forth proof of their uniqueness up to isomorphism. The (Turing) computable construction of an isomorphism for these structures has only one potentially unbounded search at every substage. Nonetheless, recently Klamullin, M. and Ng have announced the following rather counterintuitive result.

Theorem 7.2 ([KMNb])**.** *The fpr-degree structures of the dense linear order η, the random graph $\mathcal{R}$, and the universal divisible abelian p-group $\mathcal{P}$ are pairwise non-isomorphic.*

Proof idea. It follows that both $\mathbf{FPR}(\mathcal{R})$ and $\mathbf{FPR}(\mathcal{P})$ have no greatest element, while $\mathbf{FPR}(\eta)$ does. Also, $\mathbf{FPR}(\mathcal{P})$ has no maximal elements, while $\mathbf{FPR}(\mathcal{R})$ does. Some these facts require a non-trivial proof. □

Let $\mathcal{B}$ be the atomless Boolean algebra. Recall that $\mathcal{B} \cong I(\eta)$, the interval Boolean algebra of η. The reader might find the following question quite strange:

Question 7.3. *Is $\mathbf{FPR}(\mathcal{B}) \cong \mathbf{FPR}(\eta)$?*

In fact, we suspect that $\mathbf{FPR}(\mathcal{B}) \ncong \mathbf{FPR}(\eta)$.

In spite of the differences in their fpr-degree structure, there is at least one fundamental property that is common for $\mathbf{FPR}(\eta)$, $\mathbf{FPR}(\mathcal{B})$, $\mathbf{FPR}(\mathcal{P})$, and $\mathbf{FPR}(\mathcal{R})$. Kalimullin, M. and Ng have announced:

Theorem 7.4 ([KMNb])**.** *For each of the structures $\eta, \mathcal{B}, \mathcal{P}, \mathcal{R}$, the fpr-degrees are not linearly ordered under $\leq_{pr}$.*

Although the proof of the theorem above requires some work, the result is somewhat unsatisfying since we don't know the answer to the following:

Question 7.5. *Is there a structure $\mathcal{A}$ for which $\mathbf{FPR}(\mathcal{A})$ is linearly ordered and $|\mathbf{FPR}(\mathcal{A})| > 1$? Can $\mathcal{A}$ be chosen homogeneous?*

We also conjecture that the fpr-degrees of the above structures enjoy some density properties, but this is still work in progress.

References

[AK00] C. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.

[Ala] P.E. Alaev. Existence and uniqueness of structures computable in polynomial time. *Algebra and Logic, v.55 (2016), n.1, pp.72-76.*

[Ash86] C. Ash. Recursive labeling systems and stability of recursive structures in hyperarithmetical degrees. *Trans. Amer. Math. Soc.*, 298:497–514, 1986.

[Boo59] W. Boone. The word problem. *Annals of Math*, 70:207–265, 1959.

[BS11] Gábor Braun and Lutz Strüngmann. Breaking up finite automata presentable torsion-free abelian groups. *Internat. J. Algebra Comput.*, 21(8):1463–1472, 2011.

[CDRU09] Douglas Cenzer, Rodney G. Downey, Jeffrey B. Remmel, and Zia Uddin. Space complexity of abelian groups. *Arch. Math. Log.*, 48(1):115–140, 2009.

[CR] D. Cenzer and J.B. Remmel. Polynomial time versus computable boolean algebras. *Recursion Theory and Complexity, Proceedings 1997 Kazan Workshop (M. Arslanov and S. Lempp eds.), de Gruyter (1999), 15-53.*

[CR91] Douglas A. Cenzer and Jeffrey B. Remmel. Polynomial-time versus recursive models. *Ann. Pure Appl. Logic*, 54(1):17–58, 1991.

[CR92] Douglas A. Cenzer and Jeffrey B. Remmel. Polynomial-time abelian groups. *Ann. Pure Appl. Logic*, 56(1-3):313–363, 1992.

[Ded] R. Dedekind. Was sind und was sollen die zahlen? *Braunschweig: Vieweg. In [Dedekind, 1932]. Republished in 1969 by Vieweg and translated in [Dedekind, 1963].*

[Dob] V. Dobritsa. Some constructivizations of abelian groups. Siberian Journal of Mathematics, 793 vol. 24, 1983, 167–173 (in Russian).

[Dow98] R. Downey. Computability theory and linear orderings. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Stud. Logic Found. Math.*, pages 823–976. North-Holland, Amsterdam, 1998.

[ECH+92] David B. A. Epstein, James W. Cannon, Derek F. Holt, Silvio V. F. Levy, Michael S. Paterson, and William P. Thurston. *Word processing in groups*. Jones and Bartlett Publishers, Boston, MA, 1992.

[EG00] Y. Ershov and S. Goncharov. *Constructive models*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 2000.

[Fre66] G. A. Freiman. *Nachala strukturnoi teorii slozheniya mnozhestv*. Kazan. Gosudarstv. Ped. Inst; Elabuž. Gosudarstv. Ped. Inst., Kazan, 1966.

[FS56] A. Fröhlich and J. Shepherdson. Effective procedures in field theory. *Philos. Trans. Roy. Soc. London. Ser. A.*, 248:407–432, 1956.

[Gon80] S. Goncharov. The problem of the number of nonautoequivalent constructivizations. *Algebra i Logika*, 19(6):621–639, 745, 1980.

[Gon97] S. Goncharov. *Countable Boolean algebras and decidability*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 1997.

[Gri90] Serge Grigorieff. Every recursive linear ordering has a copy in dtime-space(n, log(n)). *J. Symb. Log.*, 55(1):260–276, 1990.

[Hig61] G. Higman. Subgroups of finitely presented groups. *Proc. Roy. Soc. Ser. A*, 262:455–475, 1961.

[HKS03] Denis R. Hirschfeldt, Bakhadyr Khoussainov, and Richard A. Shore. A computably categorical structure whose expansion by a constant has infinite computable dimension. *J. Symbolic Logic*, 68(4):1199–1241, 2003.

[KMNa] I. Kalimullin, A. Melnikov, and K.M. Ng. Algebraic structures computable without delay. *Submitted*.

[KMNb] I. Kalimullin, A. Melnikov, and K.M. Ng. Cantor's back-and-forth method and computability without delay. *(To appear)*.

[KMNc] I. Kalimullin, A. Melnikov, and K.M. Ng. The diversity of categoricity without delay. *Algebra i Logika (to appear)*.

[KN94] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, pages 367–392, 1994.

[KN08] Bakhadyr Khoussainov and Anil Nerode. Open questions in the theory of automatic structures. *Bulletin of the EATCS*, 94:181–204, 2008.

[KNRS07] Bakhadyr Khoussainov, André Nies, Sasha Rubin, and Frank Stephan. Automatic structures: richness and limitations. *Log. Methods Comput. Sci.*, 3(2):2:2, 18, 2007.

[Kri96] L. Kristiansen. *Papers on Subrecursion Theory, Dr Scient Thesis, Research report 217*. PhD thesis, University of Oslo, 1996.

[KS99] Bakhadyr Khoussainov and Richard A. Shore. Effective model theory: the number of models and their complexity. In *Models and computability (Leeds, 1997)*, volume 259 of *London Math. Soc. Lecture Note Ser.*, pages 193–239. Cambridge Univ. Press, Cambridge, 1999.

[LaR77] P. LaRoche. Recursively presented boolean algebras. *Notices AMS*, 24:552–553, 1977.

[Mac11] Dugald Macpherson. A survey of homogeneous structures. *Discrete Math.*, 311(15):1599–1634, 2011.

[Mal61] A. Mal′cev. Constructive algebras. I. *Uspehi Mat. Nauk*, 16(3 (99)):3–60, 1961.

[MM] A. Melnikov and A. Montalban. Computable polish group actions. *To appear*.

[Mon13] Antonio Montalbán. A computability theoretic equivalent to Vaught's conjecture. *Adv. Math.*, 235:56–73, 2013.

[Nov55] P. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov*, 44:1–143, 1955.

[NS07] André Nies and Pavel Semukhin. Finite automata presentable abelian groups. In *Logical foundations of computer science*, volume 4514 of *Lecture Notes in Comput. Sci.*, pages 422–436. Springer, Berlin, 2007.

[NT08] André Nies and Richard M. Thomas. FA-presentable groups and rings. *J. Algebra*, 320(2):569–585, 2008.

[Nur74] A. Nurtazin. *Computable classes and algebraic criteria of autostability*. Summary of Scientific Schools, Math. Inst. SB USSRAS, Novosibirsk, 1974.

[Rab60] M. Rabin. Computable algebra, general theory and theory of computable fields. *Trans. Amer. Math. Soc.*, 95:341–360, 1960.

[Rem81] Jeffrey B. Remmel. Recursive boolean algebras with recursive atoms. *J. Symb. Log.*, 46(3):595–616, 1981.

[Rog87] H. Rogers. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, second edition, 1987.

[Smi81] R. Smith. Two theorems on autostability in p-groups. In *Logic Year 1979–80 (Proc. Seminars and Conf. Math. Logic, Univ. Connecticut, Storrs, Conn., 1979/80)*, volume 859 of *Lecture Notes in Math.*, pages 302–311. Springer, Berlin, 1981.

[Tsa11] Todor Tsankov. The additive group of the rationals does not have an automatic presentation. *J. Symbolic Logic*, 76(4):1341–1351, 2011.

[vdW30] B. van der Waerden. Eine Bemerkung über die Unzerlegbarkeit von Polynomen. *Math. Ann.*, 102(1):738–739, 1930.