Definability of Summation Problems for Abelian Groups and Semigroups

Faried Abu Zaid
TU Ilmenau
faried.abu-zaid@tu-ilmenau.de

Anuj Dawar
University of Cambridge
anuj.dawar@cl.cam.ac.uk

Erich Grädel
RWTH Aachen University
graedel@logic.rwth-aachen.de

Wied Pakusa
University of Oxford
wied.pakusa@cs.ox.ac.uk

***Abstract*—We study the descriptive complexity of summation problems in Abelian groups and semigroups. In general, an input to the summation problem consists of an Abelian semigroup G, explicitly represented by its multiplication table, and a subset X of G. The task is to determine the sum over all elements of X.**

Algorithmically this is a very simple problem. If the elements of X come in some order, then we can process these elements along that order and calculate the sum in a trivial way. However, what makes this fundamental problem so interesting for us is that from the viewpoint of logical definability its tractability is much more delicate. If we consider the semigroup G as an abstract structure and X as an abstract set, without a linear order and hence without a canonical way to process the elements one by one, then it is unclear how to define the sum in any logic that does not have the power to quantify over a linear order. Indeed the trivial summation algorithm cannot be expressed in any polynomial-time logic or, in fact, in any computational model which works on abstract mathematical structures in an isomorphism-invariant way without violating polynomial resource bounds.

The surprising difficulty, in terms of logical definability, of this basic mathematical problem is the reason why Ben Rossman asked, more than ten years ago, whether it can be expressed in the logic Choiceless Polynomial Time with counting (CPT). Note that, to date, CPT is one of the most powerful known candidates for a logic that might be capable of defining every polynomial-time property of finite structures.

In this paper we clarify the status of the definability for the summation problem for Abelian groups and semigroups in important polynomial-time logics. In our first main result we show that the problem can be defined in fixed-point logic with counting (FPC). Since FPC is contained in CPT this settles Rossman's question. Our proof is based on a dynamic programming approach and heavily uses the counting mechanism of FPC. In our second main result we give a matching lower bound and show that the use of counting operators cannot be avoided: the summation problem, even over Abelian groups, cannot be defined in pure fixed-point logic without counting. Our proof is based on a probabilistic argument.

I. Introduction

The **Abelian Semigroup Summation Problem (ASSP)** can be formulated as follows: Given a finite Abelian semigroup $(G, +)$ and a subset $X \subseteq G$, determine the sum $\sum X = \sum_{x \in X} x$ in G. Note that the term $\sum X$ is well-defined since the semigroup G is Abelian.

Part of this work was carried out while the second and third authors were visitors at the Simons Institute for the Theory of Computing at UC Berkeley, in the Logical Structures in Computation Programme. The fourth author was partially supported by a DFG research grant (PA 2962/1-1).

978-1-5090-3018-7/17/$31.00 ©2017 European Union

This is a very basic mathematical problem that appears in many applications in mathematics and computer science. Of course there are numerous variations and special cases of this problem. Probably the most important one is the case where G is actually an Abelian group. Further, in some applications the variation appears where the group or semigroup is infinite and fixed, but the summation is still to be done over a given finite subset of G.

Our interest in the Abelian Semigroup Summation Problem is due to the observation that it illustrates, in a mathematically very pure way, the basic differences between logics and algorithms, or between definability and complexity, that underly some of the most fundamental and exciting problems of logic in computer science.

Computationally the Abelian Semigroup Summation Problem is extremely simple. If we assume that the semigroup is given by its multiplication table, and the elements in the set X come in some order, then we can process these elements along that order and calculate the sum in a trivial way. However, if we consider the semigroup as an abstract structure and X as an abstract set, without a linear order and hence without a canonical way to process the elements of X one by one, then it is unclear how to define the sum in any logic that does not have the power to quantify over a linear order. While it is easy to see that the ASSP is not first-order definable, it has been unknown whether it is definable in more powerful logics such as least fixed-point logic (LFP), or in the finite-variable fragment of infinitary logic $\mathrm{L}^{\omega}_{\infty\omega}$. Even more interestingly, it has also been unclear whether the Abelian Semigroup Summation Problem can be dealt with by the most powerful logics that can be evaluated in polynomial time, and which currently still are candidates for logics that might capture PTIME, such as Rank Logic (FPR) or Choiceless Polynomial Time (CPT); we remark that in this paper whenever we speak of Choiceless Polynomial Time (CPT), then we refer to the variant *with counting* unless explicitly stated otherwise.

In fact the Abelian Semigroup Summation Problem has been proposed, more than ten years ago, by Ben Rossman as a problem that might be used to prove that Choiceless Polynomial Time falls short of capturing all polynomial time properties of finite structures. We quote Rossman's statement from [5]:

"This is the most basic problem I can think of that appears difficult for CPT but is obviously polynomial time. I don't even

know the answer when [the given semigroup] S is an Abelian group, or even a direct product of cyclic groups $\mathbb{Z}_2$."

We shall answer this question in this paper. It will turn out that the Abelian Semigroup Summation Problem is actually definable in fixed-point logic with counting (FPC). Further, when restricted to Abelian groups, the summation problem is also expressible in first-order solvability logic. On the other side, we shall prove that counting (or a linear-algebraic operator such as a solvability quantifier) is needed to define the Abelian Semigroup Summation Problem. Indeed, we show that the ASSP is not definable in LFP or even in Choiceless Polynomial Time without counting, not even in the case of Abelian groups, or indeed direct products of cyclic groups $\mathbb{Z}_2$.

For semigroups, this can be shown by a simple reduction from the "Even Cardinality Problem", but for groups we need more sophisticated probabilistic arguments, which may well be of independent interest.

Let us briefly recall the background concerning the quest for a logic for polynomial time and some of the logics that have been proposed in this context; for details we refer to [13], [18]. A good starting point is fixed-point logic with counting (FPC), which may be considered as the logic of reference in the search for a logic for polynomial time. Fixed-point logic with counting was introduced, somewhat informally, by Immerman. A more formal definition, based on two-sorted structures, inflationary fixed-points, and counting terms was given in [15]. Actually, FPC comes rather close to being a logic for polynomial time. It is strong enough to express most of the fundamental algorithmic techniques leading to polynomial-time procedures and it captures PTIME on many interesting classes of finite structures, including trees, planar graphs, structures of bounded tree width, and actually all classes of graphs with an excluded minor [19]. For a recent survey on FPC, see [9]. Many polynomial-time queries which had been proposed as challenging candidates for polynomial-time logics turned out to be FPC-definable. One striking example is the result of Anderson, Dawar, and Holm which shows that the size of a maximum matching in a graph is definable in FPC [2]. This was once considered as a difficult problem even for the more powerful logic Choiceless Polynomial Time [7], just as it was the case for the Abelian Semigroup Summation Problem. Indeed, our new FPC-definability result for the ASSP is a further notable example which underlines the surprising expressive power of fixed-point logic with counting.

At the moment, there are two main candidates of logics which are known to be more powerful than FPC and which might still capture polynomial time. One is the logic Choiceless Polynomial Time (CPT) which was introduced by Blass, Gurevich, and Shelah in [6], and which is based on the model of *abstract state machines*, today known as *BGS-machines*. These computational devices operate directly on relational input structures and not on string encodings such as Turing machines. Computations of BGS-machines preserve the symmetries of input structures. The consequence is that such BGS-computations have to be *choiceless* which means that one cannot arbitrarily choose elements along a computation, which is a feature used by many fundamental polynomial-time algorithms including depth-first search, Gaussian elimination etc. To compensate for this, BGS-machines can manipulate higher-order states which model *parallel executions*. Instead of arbitrarily choosing individual elements, which would break symmetries, they consider all possible choices in parallel and combine the result of the individual computations in the end. Choiceless Polynomial Time is the restriction of BGS-machines to polynomial-time resources. It is known that CPT is strictly more powerful than FPC. For example, CPT can express the CFI-query used by Cai, Fürer, and Immerman to separate FPC from polynomial time and, more generally, CPT captures polynomial time over interesting classes of structures over which FPC fails to express all polynomial-time properties [1].

The second (family of) candidates are extension of logics, such as FPC, by operators from linear algebra. The motivation to consider such logics is due to the result by Atserias, Bulatov, and Dawar [3] showing that FPC cannot express the solvability of linear equation systems over finite Abelian groups. The most important such logic is Rank Logic (FPR) which is the extension of FPC by operators that compute the rank of definable matrices over finite fields [22], [23], [11], [16], [26]. Again it is known that FPR is more powerful than FPC, but we do not know much about the relation between CPT and FPR. In this paper, we will be concerned with first-order solvability logic (FOS) which is a similar extension of first-order logic (FO) by solvability quantifiers which express the solvability of linear equation systems over finite rings. FOS has been studied in [10], [26]. Note that if we would define FOS only with solvability quantifiers over finite fields, then FOS would be embeddable into FPR, because over fields the solvability of linear equation systems reduces to computing matrix ranks. However, it is open whether FPR can express the solvability problem for linear equation systems over finite *rings*, and, as a consequence, it is open whether FOS $\leq$ FPR holds or not. We shall prove that the summation problem over Abelian groups can be formulated as the solvability problem for a family of linear equation systems over finite rings. From this we obtain our FOS-definability result.

II. Logics and structures

We assume that the reader is familiar with the standard notions, ideas, and concepts from finite model theory and descriptive complexity theory, see e.g. [12], [13], [24]. In particular, we assume familiarity with the fixed-point logics LFP and IFP that extend first-order logic by least and inflationary fixed-point operators, respectively. It is known that LFP and IFP are equally expressive. Furthermore, we recall the Immerman-Vardi Theorem which says that LFP (and equivalently, IFP) captures polynomial time on finite structures that are equipped with a linear order.

Fixed-point logic with counting: Before we proceed, we recall the definition of *fixed-point logic with counting* (FPC). In a nutshell, FPC is the extension of inflationary fixed-point

logic (IFP) by *counting terms*. Formulas of FPC are evaluated over the *two-sorted extension* of an input structure $\mathfrak{A}$ by a copy of the natural numbers. Following [11] we denote by $\mathfrak{A}^{\#}$ the two-sorted extension of a τ-structure $\mathfrak{A} = (A, R_1, \dots, R_k)$ by $\mathfrak{N} = (\mathbb{N}, +, \cdot, 0, 1)$, i.e. the two-sorted structure $\mathfrak{A}^{\#} = (A, R_1, \dots, R_k, \mathbb{N}, +, \cdot, 0, 1)$ where the universe of the first sort (also referred to as *vertex sort*) is A and the universe of the second sort (also referred to as *number sort* or *counting sort*) is $\mathbb{N}$.

As usual for the two-sorted setting we have, for both, the vertex and the number sort, a collection of typed first-order variables. We agree to use Latin letters $x, y, z, \dots$ for variables which range over the vertices and Greek letters $\nu, \mu, \dots$ for variables ranging over the numbers. Similarly, for second-order variables R we allow mixed types, that is a relation symbol R of type $(k, \ell) \in \mathbb{N} \times \mathbb{N}$ stands for a relation $R \subseteq A^k \times \mathbb{N}^\ell$. Of course, already first-order logic over such two-sorted extensions is undecidable. In order to obtain a logic with polynomial-time data complexity we restrict the range of quantifiers over the number sort by fixed polynomials. More precisely, FPC-formulas can use quantifiers over the numeric sort only in the form $Q\nu \le n^q.\varphi$ where $Q \in \{\exists, \forall\}$ and where $q \ge 1$ is a fixed constant. The range of the quantifier Q is $\{0, \dots, n^q\}$ where n denotes the size of the input structure $\mathfrak{A}$. Similarly, for fixed-point operators we bound the numeric components of fixed-point variables R of type (k, ℓ) in all fixed-point definitions as follows:

$$[\text{ifp } R\vec{x}\vec{\nu} \le n^q \,.\, \varphi(\vec{x}, \vec{\nu})]\,(\vec{x}, \vec{\nu}).$$

Accordingly, when we determine the inflationary fixed-point defined by the formula above in an input structure $\mathfrak{A}$, then we only consider relations of the form $R \subseteq A^k \times \{0, \dots, n^q\}^\ell$.

The crucial elements of FPC are *counting terms* which allow to define cardinalities of sets. Starting with an arbitrary FPC-formula $\varphi(x)$ one can form a new *counting term* $s = [\#x \,.\, \varphi]$ whose value in $\mathfrak{A}$ is just the size of the set defined by φ in $\mathfrak{A}$. In particular, the term s is a *numeric term*, that is s takes its value in the number sort. More precisely, for an input structure $\mathfrak{A}$, the value $s^{\mathfrak{A}} \in \mathbb{N}$ of s in $\mathfrak{A}$ is the number of elements $a \in A$ such that $\mathfrak{A} \models \varphi(a)$. One can also allow counting terms of a more general form without increasing the expressive power of FPC. In particular, counting terms $[\#\vec{x}\vec{\mu} \le n^q \,.\, \varphi]$ over mixed tuples $\vec{x}\vec{\mu}$ of (both sorts of) variables can be simulated with unary counting terms (using variables of the base sort only) as introduced above; we refer to [25] for this translation, which makes use of the fixed-point operator of FPC, and also for more details about fixed-point logic with counting in general.

III. Definability in fixed-point logic with counting

In this section we establish our first main result and show that the summation problem in Abelian semigroups can be defined in fixed-point logic with counting (FPC). More precisely, we show that FPC can define the class of all structures $(G, +, X, y)$, where $(G, +)$ is a finite Abelian semigroup, where $X \subseteq G$, and where $y \in G$ is such that

$$\sum X = \sum_{x \in X} x = y.$$

Our FPC-procedure for the ASSP is based on a dynamic programming approach. A first idea would be to inductively define, for increasing values of $i \ge 1$, the sets $\Sigma^i \subseteq G$ consisting of all elements $g \in G$ which can be expressed as sums of elements from X of length precisely i (without repetitions of elements), that is

$$\Sigma^i = \{g \in G : g = x_1 + \cdots + x_i, x_j \in X \text{ and } x_k \neq x_\ell \text{ for } k \neq \ell\}.$$

Of course, if we could define these sets for $1 \le i \le n$ in FPC, where $n = |X|$, then we were done: the sum over all elements in X is the unique element that is contained in Σ^n.

However, maintaining the sets Σ^i alone is insufficient, since it is not clear how to define the successor stage Σ^{i+1} using only the sets Σ^j for $j \le i$. Hence we generalise this approach in a slight but crucial way. Instead of only updating the membership information about elements $g \in G$ in the sets Σ^i, we exploit the counting mechanism of FPC and inductively define the *number* of witnesses for elements $g \in \Sigma^i$, that is the number of different ways in which $g \in G$ can be expressed as an X-sum of length precisely i. Note that membership in Σ^i is a special case which corresponds to checking whether this number is zero or non-zero.

We start to make things more precise. For convenience, we assume that $(G, +)$ is an Abelian monoid, i.e. that it contains a neutral element $0 \in G$. If this is not the case, then we can easily add a fresh neutral element 0 by using a simple first-order transformation. For all $g, h, z \in G$ and $0 \le i \le n$ we define the following sets

$$T_i(g,h) = \{(x_1, \dots, x_i) \in X^i : g + x_1 + \cdots + x_i = h \text{ and } x_k \neq x_\ell \text{ for } 1 \le k \neq \ell \le i\}$$
$$R_i^{\neq z}(g,h) = \{(x_1, \dots, x_i) \in T_i(g,h) : x_j \neq z, j = 1, \dots, i\}$$

Correspondingly, we define values

$$t_i(g,h) = |T_i(g,h)|, \quad r_i^{\neq z}(g,h) = |R_i^{\neq z}(g,h)|.$$

Note that for $i = 0$ and all $g, h, z \in G$ we have

$$T_i(g,h) = R_i^{\neq z}(g,h) = \begin{cases} \{()\}, & g = h \\ \varnothing, & \text{else} \end{cases}$$

As before we let $n = |X|$. Then we observe:

Remark 1. *$\sum X = y$ if, and only if, $t_n(0, y) > 0$.*

Hence, in order to express the summation problem it suffices to define the values $t_i(g,h)$ for all $g, h \in G$ and $0 \le i \le n$. More generally, with this information we can determine which semigroup elements $g \in G$ are X-sums of length precisely $i \le n$. In fact we have $\Sigma^i = \{g \in G : t_i(0, g) > 0\}$.

The next three lemmas provide recursive definitions of the sets $T_i(g,h)$ and $R_i^{\neq z}(g,h)$ in terms of the sets $T_j(g,h)$

and $R_j^{\neq z}(g,h)$ for indices $j < i$. Although the proofs are straightforward, these three lemmas are the key in order to obtain an inductive system for the values $t_i(g,h)$ and $r_i^{\neq z}(g,h)$.

Lemma 2. *For all $g,h \in G$ and $1 \leq i \leq n$ we have*

$$T_i(g,h) = \biguplus_{x \in X} \{(x, x_2, \ldots, x_i) : (x_2, \ldots, x_i) \in R_{i-1}^{\neq x}(g+x,h)\}.$$

Lemma 3. *For all $g,h,z \in G$ and $1 \leq i \leq n$ we have*

$$R_i^{\neq z}(g,h) = T_i(g,h) \setminus \biguplus_{j=1,\ldots,i} \{(x_1, \ldots, x_i) \in T_i(g,h) : x_j = z\}.$$

Lemma 4. *For all $g,h,z \in G$, $1 \leq i \leq n$, $1 \leq j \leq i$ we have*

$$\begin{aligned} |\{(x_1, \ldots, x_i) \in T_i(g,h) : x_j = z\}| &= |R_{i-1}^{\neq z}(g+z,h)| \\ &= r_{i-1}^{\neq z}(g+z,h). \end{aligned}$$

We are now ready to give an inductive definition for the values $t_i(g,h), r_i^{\neq z}(g,h)$ for $g,h,z \in G$ and $0 \leq i \leq n$ in terms of the respective values for smaller indices.

- The case $i = 0$ is simple. For all $g,h,z \in G$ we have

$$t_0(g,h) = r_0^{\neq z}(g,h) = \begin{cases} 1 & g = h \\ 0 & \text{else.} \end{cases}$$

- For $1 \leq i \leq n$ we obtain the following equations

$$t_i(g,h) = \sum_{x \in X} r_{i-1}^{\neq x}(g+x,h) \qquad \text{(Lemma 2)}$$

$$r_i^{\neq z}(g,h) = t_i(g,h) - i \cdot r_{i-1}^{\neq z}(g+z,h) \qquad \text{(Lemma 3+4)}$$

It remains to be checked that the above system can be translated into an inductive definition for the invariants $t_i(g,h)$ and $r_i^{\neq z}(g,h)$ in fixed-point logic with counting. The first thing we observe is that the values for t_i and r_i become exponential in the size of our set X. For example, if $y = \sum X$, then we have that $t_n(0,y) = n!$ where $n = |X|$. On the other hand, a trivial upper bound for the numbers which can occur as values for t_i and r_i is n^n. Hence, we can represent the values for t_i and r_i in binary using polynomially many bits only. Let us explain how we can work with the binary encoding of natural numbers in a logical context. The standard way to do this in the logic FPC is to use the numeric sort. To be more precise, let us consider pairs $N = (\varphi(\mu), t)$ consisting of an FPC-formula φ with a free numeric variable μ and of a closed numeric term t. Then the pair N encodes in each input structure $\mathfrak{A}$ a natural number $N^{\mathfrak{A}} \in \mathbb{N}$ which is determined as

$$N^{\mathfrak{A}} = \sum_{i \leq t^{\mathfrak{A}}, \mathfrak{A} \models \varphi(i)} 2^i.$$

We further observe that, since the values $t^{\mathfrak{A}}$ of the numeric term t are polynomially bounded in the size of $\mathfrak{A}$, all natural numbers that we can represent in this way have a binary representation of polynomial length. Now, since the numeric sort is *ordered* it follows by the Immerman-Vardi Theorem that we can define in FPC *every* polynomial-time computable property and function over the numeric sort. In particular we can define in fixed-point logic with counting the basic arithmetic on natural numbers (in binary representation) that is required for the inductive construction of the values t_i and r_i according to our system above, such as the multiplication and addition of natural numbers.

The only non-trivial part that remains is the evaluation of the sum over the set X in the equation for $t_i(g,h)$. Since X is not ordered we cannot apply the Immerman-Vardi Theorem here. Instead we use a standard trick to reduce the evaluation of an *unordered sum* of natural numbers to the evaluation of an *ordered sum* of natural numbers. This reduction is FPC-definable and is crucially based on the counting mechanism of FPC. Let us recall the idea by considering our instance of the unordered sum $t_i(g,h) = \sum_{x \in X} r_x$ where $r_x = r_{i-1}^{\neq x}(g+x,h)$. First we obtain an FPC-definable linear preorder $\preceq$ on X that is defined by setting $x \preceq y$ if $r_x \leq r_y$. Let $\sim$ denote the associated equivalence relation, that is $x \sim y$ if $x \preceq y$ and $y \preceq x$. Then the equivalence classes $[x] \in X/\sim$ consist of elements which all contribute the same value to the above sum. If we denote the sizes of these classes by $|[x]|$, then we can rewrite our sum in an equivalent way as:

$$\sum_{x \in X} r_x = \sum_{[x] \in X/\sim} |[x]| \cdot r_x.$$

Since $\preceq$ induces a linear order on $X/\sim$ and since we can determine the sizes of the equivalence classes $|[x]|$ by using counting terms, this trick reduces the evaluation of the original sum over X to a sum over the ordered set $X/\sim$. At this point we can make use of the Immerman-Vardi Theorem to see that evaluating the ordered sum is possible in fixed-point logic with counting.

Theorem 5. *The summation problem in Abelian semigroups is definable in fixed-point logic with counting.*

IV. A REDUCTION TO LINEAR EQUATION SYSTEMS OVER FINITE RINGS

A particular interesting case of the summation problem over Abelian semigroups arises when we require that the underlying semigroup is an Abelian group. Recall that already for this, apparently simpler, setting of Abelian *groups* it was open whether the summation problem can be defined in Choiceless Polynomial Time which is, to date, the most promising candidate of a logic which might capture polynomial time.

In this section we show that the summation problem over Abelian groups can be reduced to linear algebra over finite rings. More precisely, we describe how to express the summation problem over Abelian groups as the solvability problem for systems of linear equations over finite cyclic rings $\mathbb{Z}_d$ where $d \in \mathbb{N}$ is a prime power. Our reduction is simple in the sense that it can be expressed in first-order logic. One consequence of this is that the summation problem over Abelian groups is definable in *first-order solvability logic* (FOS). This logic extends pure first-order logic by new operators that decide the solvability of linear equation systems over finite cyclic rings. This logic, and also many similar variants based on different linear-algebraic operators

over finite algebraic domains, have been studied intensively in recent years, see for example [11], [10], [22], [23], [26]. The reason for this is that many fundamental problems from linear algebra over finite domains turned out to be undefinable in fixed-point logic with counting [3]. This holds, in particular, for the solvability problem for linear equation systems.

A. First-order solvability logic

First-order solvability logic (FOS) extends first-order logic by *solvability quantifiers* which are logical operators capable of deciding the solvability of linear equation systems over finite cyclic rings $\mathbb{Z}_d$ where $d \in \mathbb{N}$ is a prime power. Formally, we add to the syntax of first-order logic the following formula creation rule.

- Let $\varphi_M(\vec{x},\vec{y},\vec{z}) \in$ FOS and $\psi_d(\vec{v},\vec{w},\vec{z}) \in$ FOS be formulas of first-order solvability logic where $|\vec{v}| = |\vec{w}|$. Then $\psi(\vec{z}) = \mathsf{slv}([\vec{x},\vec{y}\ \varphi_M],[\vec{v},\vec{w}\ \psi_d])$ is a formula of FOS as well.

The semantics of $\psi(\vec{z})$ over an input structure $\mathfrak{A}$ (together with an interpretation $\vec{z} \mapsto \vec{c}$ of the free variables $\vec{z}$) is defined as follows. The idea is that the formula φ_M defines a $\{0,1\}$-coefficient matrix M of a linear equation system $M \cdot x = \mathbf{1}$ over $\mathbb{Z}_d$ where $d \in \mathbb{N}$ is a prime power and this prime power d is specified by the formula ψ_d. More precisely, let $k = |\vec{v}| = |\vec{w}|$. Then the formula ψ_d defines in $(\mathfrak{A}, \vec{z} \mapsto \vec{c})$ a binary relation on k-tuples. If this relation happens to be a linear order whose length $d \in \mathbb{N}$ is a prime power, then we take this prime power d to specify the cyclic ring $\mathbb{Z}_d$. Otherwise we agree that the formula is false in $\mathfrak{A}$ (it can be shown that this case can be avoided through a syntactic criterion). Further we let $\ell = |\vec{x}|$ and $\ell' = |\vec{y}|$. Then the formula φ_M defines in $\mathfrak{A}$ the $I \times J$-matrix M_φ over $\{0,1\} \subseteq \mathbb{Z}_d$ where $I = A^\ell$ and $J = A^{\ell'}$ and where for $\vec{a} \in I$ and $\vec{b} \in J$ we have $M_\varphi(\vec{a},\vec{b}) = 1$ if, and only if, $\mathfrak{A} \models \varphi_M(\vec{a},\vec{b},\vec{c})$.

Let $\mathbf{1}$ be the I-identity vector over $\mathbb{Z}_d$, that is $\mathbf{1}(\vec{a}) = 1$ for all $\vec{a} \in I$. Then M_φ and $\mathbf{1}$ determine the linear equation system $M_\varphi \cdot x = \mathbf{1}$ over $\mathbb{Z}_d$ where $x = (x_j)_{j \in J}$ is a J-vector of variables x_j which range over $\mathbb{Z}_d$. Finally, $\mathfrak{A} \models \psi(\vec{c})$ if, and only if, $M_\varphi \cdot x = \mathbf{1}$ is solvable.

At first glance it might seem that in our definition we restrict to linear equation systems of a special syntactic form. Indeed, we require that *every* linear equation in our system has the form $\sum_{j \in J} a_j \cdot x_j = 1$ with coefficients a_j from the set $\{0,1\} \subseteq \mathbb{Z}_d$. However, it is easy to show that this is not a severe restriction, because every definable linear equation system in a general form can be transformed into this kind of syntactic normal form in first-order logic (see for example Lemma 4.1 in [10]). Hence, we do not limit the expressive power of FOS by imposing these kinds of syntactic restrictions.

Another important point to observe about our definition of FOS is that we use a *uniform* solvability quantifier slv which takes the prime powers $d \in \mathbb{N}$ as part of its input. In contrast one could also annotate the prime powers to the solvability quantifiers as fixed constants, that is one could introduce separate solvability quantifiers slv_d for every prime power $d \in \mathbb{N}$ each for solving linear equation systems over $\mathbb{Z}_d$. It was recently shown that, in general, the logics based on uniform linear-algebraic operators are strictly stronger than the logics with separate operators for the domains [16]. We also expect this to be the case for first-order solvability logic with respect to the summation problem over Abelian groups.

B. Reducing the summation problem in Abelian groups to linear equation systems

To explain how our reduction works let us first fix an instance $(G,+,X)$ of our problem. Recall that in this section $(G,+)$ is an Abelian *group* rather than a semigroup. As before $X \subseteq G$ is a set of elements and we are interested in whether or not the sum over all elements in X yields some particular element $g \in G$. Without loss of generality this element $g \in G$ can always be assumed to be the neutral element $0 \in G$. Hence the problem we want to express is whether or not $\sum X = \sum_{x \in X} x = 0$.

In order to reduce the instance $(G,+,X)$ of the summation problem over Abelian groups to linear equation systems over finite rings, we first make use of the structure of the finite Abelian group $(G,+)$. Recall that by the well-known *structure theorem for finite Abelian groups* we can find group elements $g_1,\dots,g_k \in G$, $k \geq 0$, such that:

- G is the direct sum of the cyclic groups $\langle g_i \rangle$ generated by the group elements $g_i \in G$, that is $G = \langle g_1 \rangle \oplus \dots \oplus \langle g_k \rangle$.
- Each element g_i is of prime-power order, that is $|g_i| = p^\ell$ for some prime $p \in \mathbb{P}$ and some $\ell \geq 1$.

Moreover, the decomposition of G into a sum of cyclic subgroups is unique in the sense that the multiset of prime powers that occur as sizes of cyclic groups in this decomposition only depends on the group G, but not on the specific choice of generators $g_1,\dots,g_k$.

If we had access to a decomposition of G into cyclic subgroups $\langle g_1 \rangle \oplus \dots \oplus \langle g_k \rangle$ for $g_1,\dots,g_k \in G$ as above, then expressing the summation problem in G would be a simple exercise. Indeed, the following approach could be used to obtain a definition in least fixed-point logic (even without counting). To express whether $\sum X = 0$ holds in G we first project the elements in X to each of the summands $\langle g_i \rangle$ individually, and then we check for every summand $G_i = \langle g_i \rangle$ whether the sum over the projected elements is 0 in this summand G_i. This reduces the summation problem over general Abelian groups to the summation problem over cyclic groups which is much simpler. Indeed, the summation problem over *cyclic* groups can be expressed in LFP for trivial reasons. First observe that on a cyclic group it is easy to define in LFP an order: we fix a generator and then enumerate the elements of the group according to this generator one by one. In a second step, we can make use of the Immerman-Vardi Theorem which says that as soon as we have an *ordered* structure as input, we can simulate all polynomial-time algorithms in least fixed-point logic LFP.

Unfortunately, it is provably impossible to define a decomposition of a finite Abelian group $(G,+)$ into cyclic summands in any of the polynomial-time logics that we consider here (such as CPT, FPC, and so on). The reason is that in general

the choice of generators $g_1, \ldots, g_k$ is not canonical and, in fact, the number of isomorphic, but different, decompositions of G is not polynomially bounded in the size of the group G. Hence one cannot construct a decomposition of G into cyclic summands, in an isomorphism-invariant way, without violating polynomial resource bounds.

To overcome this, our main idea is as follows. Instead of trying to *construct* a decomposition of G and to evaluate the sum in each summand, which is as we just explained not possible, we rather try to *guess* one decomposition of G together with one summand in this decomposition and then verify that the sum over the projected elements from X in this summand is not 0. Of course, we have that $\sum X \neq 0$ if, and only if, we can find such a witnessing pair of a decomposition and a corresponding summand. The important point about this approach is that the "guessing step" can be formulated using a family of linear equation systems over finite cyclic rings $\mathbb{Z}_d$.

Let us make this idea more precise. We define a *witness* (showing that $\sum X \neq 0$ in G) as a pair (φ, d) consisting of a prime power $d \in \mathbb{N}$ and a (group) homomorphism $\varphi : G \to \mathbb{Z}_d$ such that $\sum \varphi(X) = \sum_{x \in X} \varphi(x) \neq 0$. Note that since φ is a homomorphism we have that $\varphi(\sum X) = \sum \varphi(X)$. Hence if we find a witness, then clearly $\sum X \neq 0$ in G. On the other hand, if $\sum X \neq 0$ in G, then we can also find a witness for this. Indeed, consider some decomposition of G as above, that is let $G = \langle g_1 \rangle \oplus \cdots \oplus \langle g_k \rangle$ for some choice of group elements $g_1, \ldots, g_k \in G$. If $\sum X \neq 0$, then we can find at least find one component $\langle g_i \rangle$ such that the (projected) sum over X in this component is not 0. Recall that $|g_i| = d$ is a prime power and, as a cyclic group, that $\langle g_i \rangle$ is isomorphic to $\mathbb{Z}_d$. Then it is easy to see that we can choose $\varphi : G \to \mathbb{Z}_d$ to be the composition of the projection of G onto the summand $\langle g_i \rangle$ and the isomorphism from $\langle g_i \rangle$ to $\mathbb{Z}_d$ so that we obtain a witness (φ, d) as claimed. We conclude that $\sum X \neq 0$ holds in G if, and only if, we can find a witness (φ, d) for this.

We next want to show that the existence of a witness (φ, d) for a prime power $d \in \mathbb{N}$ reduces to the solvability problem of a linear equation system over $\mathbb{Z}_d$. To this end we consider for every group element $g \in G$ a distinct variable x_g over the domain $\mathbb{Z}_d$. Intuitively the value of x_g should be $\varphi(g)$. Hence, the first thing we have to express is that the values of the variables x_g really define a homomorphism from G to $\mathbb{Z}_d$. This can easily be achieved by including the following set of linear equations in our system:

$$x_g + x_h = x_{g+h}, \text{ for all } g, h \in G.$$

What remains is to capture the condition $\sum \varphi(X) \neq 0$ by a linear equation. This, however, turns out to be impossible since, in general, this condition does not define an affine space. Instead we consider the following parameterised version. Let $z \in \mathbb{Z}_d, z \neq 0$. Instead of the equation $\sum \varphi(X) \neq 0$, we consider the equation $\sum \varphi(X) = z$ parameterised by z and formulate this as a linear equation in our system. To check the original condition "$\neq 0$" we then just go through all possible choices for the parameter $z \in \mathbb{Z}_d, z \neq 0$ and check the resulting systems with parameter z for solvability. To summarise, we have that a witness exists if, and only if, for *some* parameter $z \in \mathbb{Z}_d, z \neq 0$, the resulting linear equation system with respect to this parameter z is solvable. The condition $\sum \varphi(X) = z$ readily translates into the following linear equation:

$$\sum_{g \in X} x_g = z.$$

It is clear that the above described linear equation systems are definable over $(G, +, X)$ in first-order logic. Using the solvability operators over $\mathbb{Z}_d$ we can express their solvability in FOS and hence we can express in FOS whether a witnesses (φ, d) exists for some prime power $d \in \mathbb{N}$. However, in order to do this we also have *define* this prime power $d \in \mathbb{N}$ in FOS (recall that the solvability operators get the prime powers d as part of their inputs in the form of a linear order of length d).

First of all, note that we only have to consider such prime powers $d \in \mathbb{N}$ which divide the order $|G|$ of the group G. We make use of the fact that the reachability problem in undirected graphs reduces to the solvability problem for linear equations systems, see [11]. This shows that the logic FOS can simulate the logic STC which is the extension of first-order logic by a symmetric transitive closure operator, see [13], [24]. Over ordered structures, it is known that STC can express every LOGSPACE-computable property of finite structures. In particular, STC can express whether a given linear order has length $d \in \mathbb{N}$ for some prime power d. Furthermore, we claim that in STC we can define for every group element $g \in G$ a linear order whose length is the order $|g| = |\{i \cdot g : i \in \mathbb{N}\}|$ of g in G. Note that if we can show this, then we are done, because we can then check in STC whether this linear order has length $d = |g|$ for some prime power d. Moreover, for every prime power d which divides the order of G we can find a group element $g \in G$ of that order. To verify the above claim, consider for a group element $g \in G$ the graph $H_g = (G, E_g)$ with the edge relation defined by $(e, f) \in E_g :\Leftrightarrow e + g = f$. The graph H_g consist of disjoint cycles each of length $|g|$ corresponding to the different cosets of the cyclic group $\langle g \rangle$ generated by g in G. Our claim follows, since it is an easy exercise to define a linear order on a cycle in STC.

Theorem 6. *The summation problem over finite Abelian groups is definable in first-order solvability logic FOS.*

V. LOWER BOUNDS FOR THE SUMMATION PROBLEM

In this section we obtain matching lower bounds and show that counting operators, or solvability quantifiers, are really necessary in order to define the summation problem over Abelian semigroups. Actually, this is easy to show if one considers the general ASSP, that is the summation problem over Abelian *semigroups*. In fact, over semigroups, it is possible to reduce the "Even Cardinality" problem to the summation problem. From this we immediately get an undefinability result for Choiceless Polynomial Time without counting and thus, in particular, for first-order logic and least fixed-point logic. However, if we consider the summation problems over Abelian *groups*, then our inputs have more structure and our proof for the lower bound becomes much more sophisticated. In

fact, in order to obtain our lower bound for the finite-variable fragment of infinitary logic, and thus for least fixed-point logic, over Abelian groups we make use of a probabilistic argument, which can then be extended to Choiceless Polynomial Time without counting.

A. Abelian semigroups

Let us start with the simple case of the summation problem over Abelian semigroups. Recall that the Even Cardinality Problem is to determine whether the universe of a finite structure (over the empty signature) has even cardinality. It is known that the Even Cardinality Problem cannot be expressed in Choiceless Polynomial Time without counting [6], [27].

Theorem 7. *The Abelian Semigroup Summation Problem is not definable in Choiceless Polynomial Time without counting.*

Proof. In order to prove the statement it suffices to show that the Even Cardinality Problem reduces to the Abelian Semigroup Summation Problem. Let M be a finite set with at least two elements and $0 \notin M$. We define the Abelian semigroup $S(M) = (M \cup \{0\}, +)$ with $x + y := 0$ for all $x, y \in M \cup \{0\}$. Now consider the semigroup $\mathbb{Z}_2 \times S(M)$. Then

$$|M| \text{ is even } \Leftrightarrow \sum\{1\} \times M = (0,0).$$

Clearly, the mapping $M \mapsto (\mathbb{Z}_2 \times S(M), \{1\} \times M, (0,0))$ is definable in first-order logic and hence also in Choiceless Polynomial Time without counting. □

B. Abelian groups

We next prove that even in the case of finite Abelian groups, the summation problem is not definable in the infinitary logic $\mathrm{L}^{\omega}_{\infty\omega}$, and therefore also not in LFP. Recall that $\mathrm{L}^{\omega}_{\infty\omega} = \bigcup_{k<\omega} \mathrm{L}^{k}_{\infty\omega}$, where $\mathrm{L}^{k}_{\infty\omega}$ extends the k-variable fragment of first-order logic by conjunctions and disjunctions over arbitrary sets of formulae. It is well-known that, over any class of structures of bounded cardinality, and thus in particular for finite structures, LFP can be embedded into $\mathrm{L}^{\omega}_{\infty\omega}$.

Our approach is probabilistic. We first establish a limit law for $\mathrm{L}^{\omega}_{\infty\omega}$ on random relational expansions of vector spaces $(\mathbb{Z}_p)^n$. We shall then show, again by a simple probabilistic argument, that the definability of Abelian group summation problem in $\mathrm{L}^{\omega}_{\infty\omega}$ would permit to construct a sentence that would violate that limit law.

We consider the group $(\mathbb{Z}_p, +, 0)$, for some prime p, and an arbitrary finite relational vocabulary $\tau = \{X_1, \ldots, X_\ell\}$. For each $n \in \mathbb{N}$, we consider the probability spaces $S_n(\mathbb{Z}_p)$, consisting of all expansions of (the additive group of) the vector space $(\mathbb{Z}_p)^n$ by relations from τ, with the uniform probability distribution.

For every sentence ψ (in whatever logic) of vocabulary $\{+, 0\} \cup \tau$, let $\mu_n(\psi)$ denote the probability that a randomly chosen structure $\mathfrak{A} \in S_n(\mathbb{Z}_p)$ is a model of ψ. We prove the following limit law.

Theorem 8. *For every τ and for every sentence $\psi \in L^{\omega}_{\infty\omega}$ of vocabulary $\{+, 0\} \cup \tau$,*

$$\lim_{n\to\infty} \mu_n(\psi) = \frac{r}{2^\ell}, \text{ for } \ell = |\tau| \text{ and some } r \le 2^\ell.$$

Proof. Let $\delta_1, \ldots, \delta_m$ be the $m = 2^\ell$ atomic τ-types in the constant 0 (and without variables). For each j, δ_j is a conjunction over ℓ atoms or negated atoms of form $X_i(0, \ldots, 0)$, for $X_i \in \tau$. Obviously, for all $j \le m$ and all n, $\mu_n(\delta_j) = 1/m$.

For any collection $a_1, \ldots, a_k$ of elements of $(\mathbb{Z}_p)^n$ let $\mathrm{span}(a_1, \ldots, a_k)$ be the subspace generated by $a_1, \ldots, a_k$. Clearly, the size of $\mathrm{span}(a_1, \ldots, a_k)$ in $(\mathbb{Z}_p)^n$ is bounded by p^k, for any n.

Recall that an atomic k-type $t(x_1, \ldots, x_k)$ of a vocabulary σ is a maximal consistent set of atoms and negated atoms in the variables $x_1, \ldots, x_k$. In our case, $\sigma = \{+, 0\} \cup \tau$, and a k-type $t(x_1, \ldots, x_k)$ specifies the linear dependencies and independencies of $x_1, \ldots, x_k$ and the truth values of all atoms $X(y_1, \ldots, y_r)$ where $X \in \tau$, and each y_i is a $\mathbb{Z}_p$-linear combination of $x_1, \ldots, x_k$.

Definition 9. For each $j \le m$, we define AT_j to be the set of all atomic types $t(x_1, \ldots, x_k)$ of vocabulary $\{+, 0\} \cup \tau$ such that

(1) t is consistent, i.e. realisable in some $(\mathbb{Z}_p)^n$.
(2) $t \vDash \delta_j$,
(3) t implies, for each $i \le k$, that $x_i \notin \mathrm{span}(x_1, \ldots, x_{i-1})$.

We then define T_j to be the theory of all extension axioms

$$\mathrm{ext}_{s,t} := \forall \bar{x}(s(\bar{x}) \to \exists x_{k+1} t(\bar{x}, x_{k+1}))$$

where s and t are, respectively, atomic k and $k+1$-types in AT_j with $t \vDash s$.

Proposition 10. *Every extension axiom $\mathrm{ext}_{s,t} \in T_j$ has asymptotic probability one on the sequence of spaces $S_n(\mathbb{Z}_p)$.*

Proof. Let $(a_1, \ldots, a_k)$ be a realization of the atomic type $s(\bar{x}) \in \mathrm{AT}_j$ in some randomly chosen expansion $\mathfrak{A}$ of $(\mathbb{Z}_p)^n$. The type $s(\bar{x})$ fixes the truth values of all τ-atoms in the variables $x_1, \ldots, x_k$ and the constant 0, and $t(\bar{x}, x_{k+1})$ additionally fixes truth-values for the τ-atoms that contain at least one term with the variable x_{k+1}. There is a bounded number q of such atoms. Therefore, if we fix some element $b \in (\mathbb{Z}_p)^n \setminus \mathrm{span}(a_1, \ldots, a_k)$, then the probability that $\mathfrak{A} \vDash t(\bar{a}, b)$ is 2^{-q}.

The elements b that we have to explore are those outside of $\mathrm{span}(a_1, \ldots, a_k)$. Each of them fixes $|\mathrm{span}(a_1, \ldots, a_k, b) \setminus \mathrm{span}(a_1, \ldots, a_k)| \le (p-1)p^k$ new elements, so there are at least p^{n-k-1} independent choices for b. Since there are fewer than p^{nk} realizations of $s(\bar{x})$ in $\mathfrak{A}$, the probability that one of them cannot be extended to a realization of $t(\bar{x}, x_{k+1})$ is at most

$$p^{nk}(1 - 2^{-q})^{p^{n-k-1}}$$

which tends to 0 exponentially fast as n goes to infinity.

Thus, the asymptotic probability of every extension axiom $\mathrm{ext}_{s,t} \in T_j$ is one on $S_n(\mathbb{Z}_p)$. □

For every $j \leq m$, $k < \omega$, let θ_j^k be the conjunction of all extension axioms in T_j with at most k variables. Further, let $E(k,j)$ be the class of all expansions $\mathfrak{A}$ of $(\mathbb{Z}_p)^n$ (for any finite $n \geq k$) such that $\mathfrak{A} \models \delta_j \wedge \theta_j^k$.

Lemma 11. $\lim_{n\to\infty} \mu_n(\delta_j \wedge \theta_j^k) = 1/m$ *for all* $j \leq m, k < \omega$.

Proposition 12. *For every* $\psi \in L_{\infty\omega}^k$ *and every* $j \leq m$, *either* $\mathfrak{A} \models \psi$ *for all* $\mathfrak{A} \in E(k,j)$, *or* $\mathfrak{A} \models \neg\psi$ *for all* $\mathfrak{A} \in E(k,j)$.

Proof. Take any two structures $\mathfrak{A}, \mathfrak{B} \in E(k,j)$. From the fact that both structures satisfy $\delta_j \wedge \theta_j^k$ we immediately get a winning strategy for the k-pebble game on $\mathfrak{A}$ and $\mathfrak{B}$ (for background on the model comparison games for k-variable logic, see [12], [24]). Hence the two structures are $\mathrm{L}_{\infty\omega}^k$-equivalent, so it cannot be the case that ψ is true in one and false in the other. □

Given any formula $\psi \in \mathrm{L}_{\infty\omega}^k$, let $r(\psi) = |\{j \leq m : \psi \text{ is true in all } \mathfrak{A} \in E(k,j)\}|$. It follows that

$$\lim_{n\to\infty} \mu_n(\psi) = \frac{r(\psi)}{m}.$$

Hence the limit law holds for $\mathrm{L}_{\infty\omega}^\omega$. □

Theorem 13. *The Abelian group summation problem is not definable in* $L_{\infty\omega}^\omega$.

Proof. Suppose that the Abelian group summation problem is definable by a formula $\varphi(x) \in \mathrm{L}_{\infty\omega}^k$ such that for every Abelian group $(H,+,0)$, all $X \subseteq H$ and every $h \in H$,

$$(H,+,0,X) \models \varphi(h) \iff \sum X = h.$$

Consider the sentence $\psi := \exists x(\varphi(x) \wedge X(x) \wedge X(0))$, which expresses that both 0 and the sum over all elements of X are contained in X. Let $G = (\mathbb{Z}_2,+,0)$ and $H = \mathbb{Z}_2^n$. For a randomly chosen $X \subseteq H$ all elements of H have equal probability to be the sum of all elements of X. The probability that this sum is itself an element of X quickly converges to $1/2$. Thus the asymptotic probability of ψ on the spaces $S_n(\mathbb{Z}_2)$ converges to $1/4$.

However, since we use only one random relation, the denominator of the asymptotic probabilities in the limit law is 2, so $\mu_n(\psi)$ should converge to either 0,1, or $1/2$. Contradiction. □

Categoricity

A classical result about limit laws for finite random structures states that the theory of all extension axioms is ω-categorical, i.e. it has, up to isomorphism, precisely one countable model. We can prove an analogous categoricity result in our setting.

Let $\mathbb{Z}_p^*$ be the weak ω-product of $\mathbb{Z}_p$. Its elements are the functions $\boldsymbol{g} : \omega \to \mathbb{Z}_p$ such that $\boldsymbol{g}(n) = 0$ for all but finitely many n, addition is defined component-wise in the obvious way, and $\mathbf{0}$ is the constant function mapping all $n \in \omega$ to 0. The next observation says that the theories $\{\delta_j\} \cup T_j$ are categorical for expansions of $\mathbb{Z}_p^*$.

Proposition 14. *Let* $\mathfrak{A}_\omega$ *and* $\mathfrak{B}_\omega$ *be any two expansions of* $\mathbb{Z}_p^*$ *to* $\{+,0\} \cup \tau$*-structures which are both models of* $\{\delta_j\} \cup T_j$. *Then* $\mathfrak{A}_\omega$ *and* $\mathfrak{B}_\omega$ *are isomorphic.*

Proof. The universes of both $\mathfrak{A}_\omega$ and $\mathfrak{B}_\omega$ are the same as for $\mathbb{Z}_p^*$. Fix an enumeration $\boldsymbol{g}_0, \boldsymbol{g}_1, \boldsymbol{g}_2, \ldots$ of this set, and define a sequence $(f_n)_{n\in\omega}$ of partial isomorphisms from $\mathfrak{A}_\omega$ to $\mathfrak{B}_\omega$ as follows. Let $f_0 = \{(\mathbf{0},\mathbf{0})\}$. Since both $\mathfrak{A}_\omega$ and $\mathfrak{B}_\omega$ are models of δ_j, this is indeed a partial isomorphism. Suppose now that, for $k \geq 0$, p_k has already been defined, with domain $\mathrm{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$, and image $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$. Since f_k is a partial isomorphism $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$ and $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ realize the same atomic type $s(\bar{x})$.

For even k, let $\boldsymbol{a}_{k+1}$ be the first element in the enumeration $\boldsymbol{g}_0, \boldsymbol{g}_1, \boldsymbol{g}_2, \ldots$ that does not appear in the domain of p_k, and let $t(\bar{x}, x_{k+1})$ be the atomic type realized by $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k, \boldsymbol{a}_{k+1})$. Since $\mathfrak{B}_\omega \models \mathrm{ext}_{s,t}$ the tuple $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ can be extended by a suitable element $\boldsymbol{b}_{k+1}$ to a realization of $t(\bar{x}, x_{k+1})$. This defines an extension of f_k to a partial isomorphism f_{k+1} from $\mathrm{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{k+1})$ to $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k+1})$.

For odd k we proceed similarly, by choosing for $\boldsymbol{b}_{k+1}$ the first element in the enumeration of the universe that is not contained in the image of f_k. Since the appropriate extension axiom holds in $\mathfrak{A}_\omega$ the element $\boldsymbol{b}_{k+1}$ can then be matched by an element $\boldsymbol{a}_{k+1}$ to provide the extension f_{k+1}.

The union $f = \bigcup_{k\in\omega} f_k$ will then be the desired isomorphism between $\mathfrak{A}_\omega$ and $\mathfrak{B}_\omega$. □

Choiceless Polynomial Time without counting

Shelah has proved that, on finite relational structures, the properties that are decided in Choiceless Polynomial Time without counting obey a zero-one law (see [4] for a clear exposition). We can show that this proof can be adapted to prove a limit law for this logic over structures over the sequence of spaces $S_n(\mathbb{Z}_p)$, entirely analogous to that proved for $\mathrm{L}_{\infty\omega}^k$ above. We formulate the result and give an overview of the proof here, as a full proof would require more space.

Fix as above an atomic k-type s in AT_j and let T denote the number of distinct $k+1$-types $t \in \mathrm{AT}_j$ such that $t \models s$. We write strong-$\mathrm{ext}_{s,t}$ for the statement that asserts that for every k-tuple of type s, there are $p^n/2T$ distinct elements witnessing extensions of this tuple to type t. Note that this is not a fixed first-order statement as it depends on n, the number of elements in the structure. Nonetheless, a calculation similar to that in [4] establishes that strong-$\mathrm{ext}_{s,t}$ has asymptotic probability one for all s and t. Again, let Θ_j^k be the conjunction of all statements strong-$\mathrm{ext}_{s,t}$ with s a k-type in AT_j. Then, the limit law for Choiceless Polynomial Time can be formulated as follows.

Lemma 15. *If* M *is a Choiceless Polynomial Time machine (without counting), then there is an* m *such that if* $\mathfrak{A}, \mathfrak{B} \models \Theta_j^m \wedge \delta_j$ *and* M *halts with a result on both inputs* $\mathfrak{A}$ *and* $\mathfrak{B}$, *then* M *accepts* $\mathfrak{A}$ *if, and only if, it accepts* $\mathfrak{B}$.

The proof of this limit law has a number of ingredients. The key is to show that we can define a finite substructure $H(\mathfrak{A})$ of

the universe of the hereditary finite sets over $\mathfrak{A}$ which includes all sets constructed by M in the computation on input $\mathfrak{A}$ (this is standard, see [4]). We then show that, if $\mathfrak{A}$ and $\mathfrak{B}$ satisfy sufficiently many strong extensions axioms, then $H(\mathfrak{A})$ cannot be distinguished from $H(\mathfrak{B})$ in $\mathrm{L}^k_{\infty\omega}$. The key is to show that the sets constructed in the course of the computation have certain symmetries. These symmetries cannot be expressed by automorphisms of $\mathfrak{A}$, since it can be shown that a random expansion of $\mathbb{Z}_p$ does not have *any* non-trivial automorphisms. Instead we rely on partial automorphisms as in [4]. Showing that these symmetries are preserved in $H(\mathfrak{A})$ can be done along the same lines as in [4] through the construction of supported sets, though the combinatorics are different for the space $S_n(\mathbb{Z}_p)$ than they are for random graphs. As a consequence we can generalise Theorem 13.

Theorem 16. *The summation problem for Abelian groups is not definable in Choiceless Polynomial Time without counting.*

VI. Finite sums in infinite semigroups

Summation problems in groups and semigroups arise in different variations. In some important applications we deal with summations over a finite set in some fixed infinite group or semigroup. This raises the questions whether our method for defining the summation problem in fixed-point logic with counting can be adapted to this case, and moreover, whether the summation problems becomes simpler, say LFP-definable, for certain important classes of semigroups.

Question: *For which (possibly singleton) classes of finite and/or infinite Abelian semigroups does there exist a formula in FPC, or in LFP or even in FO, that defines, for each semigroup G in the class and all finite subsets X of G, the sum over X in G?*

Actually, our results in Sect. III show that an appropriately defined variant of FPC (with relativisations of quantifiers to X and bounds for numbers defined by $|X|$) works in all cases. We add a few observations about definability of summation problems in simpler logics.

Obviously, the summation problem is first-order definable for all finite classes of finite semigroups. Moreover it is first-order definable for the class of all semilattices, i.e. all commutative semigroups which are idempotent, which means that $g + g = g$ for all $g \in G$. In that case, for every set $X \subseteq G$ the sum $\sum X$ is the least upper bound wrt to the partial order defined by $g \leq g' :\leftrightarrow \exists h(g + h = g')$.

In an arbitrary monoid $(G, +, 0)$ the relation $\leq$ is always a pre-order (being reflexive and transitive), but it need not necessarily be a partial order since it is not always anti-symmetric. Clearly, if $\leq$ is actually a *linear order* then the summation problem is LFP-definable due to the Immerman-Vardi Theorem.

Another interesting case of LFP-definable summation problems arise when the summation is done over a linearly independent set X, say in a vector space over $\mathbb{Z}_2$. We can then take the obvious LFP-definition

$$[\mathrm{lfp}\, Yx\,.\, Xx \vee \exists y \exists z (Yy \wedge Xz \wedge x = y + z)](x)$$

of the subgroup that is generated by X. Since $\sum X$ is the unique element that is included in this fixed point at the last stage, it is definable due to the Stage Comparison Theorem for LFP.

The above idea can be generalised. Let G be a finite Abelian group, with a subset X, so that we can define in LFP (possibly with parameters) a set $B = \{b_1, \ldots, b_k\}$ such that

$$G \cong \langle b_1 \rangle \oplus \cdots \oplus \langle b_k \rangle.$$

Then we claim that the representation of any element $g \in G$ in terms of B is LFP-definable. In fact, if $g = \bigoplus_i z_i \cdot b_i$, for $0 \leq z_i < |b_i|$, then z_i is the unique number in $\{0, \ldots, |b_i| - 1\}$ which satisfies that $g - z_i \cdot b_i \in G_{-i} = \langle b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_k \rangle$. As we can define in LFP the subgroup G_{-i}, as we saw above, the claim follows. What this shows is that we can define in LFP the isomorphism from G to $\langle b_1 \rangle \oplus \cdots \oplus \langle b_k \rangle$. This means that we can reduce problems over G to the cyclic summands $\langle b_i \rangle$ in LFP. Since cyclic groups can be ordered in LFP, by fixing a generator, it follows from the Immerman-Vardi Theorem that every polynomial-time property of cyclic groups is LFP-definable. Hence, LFP captures all polynomial-time problems over Abelian groups in which we can define a basis B as above. In particular, we can express the summation problem.

VII. Applications in database theory

The problem of computing the sum over a set of *a priori* unknown and potentially unbounded size in a monoid appears in a number of applications. We mention here two of them that arise in database theory. The first one concerns aggregate operations in databases query languages, the second database provenance.

A. Aggregate operations

In logical terms, an aggregate operation Γ allows us to derive from a query $\varphi(\bar{x}, \bar{y})$ functions $F_\varphi(\bar{x}) := \Gamma_{\bar{y}} \varphi(\bar{x}, \bar{y})$, taking values typically in some numerical domain such as $\mathbb{N}$ or $\mathbb{Q}$. A value $F_\varphi(\bar{a})$ is obtained by collecting all tuples $\bar{b}$ such that $\varphi(\bar{a}, \bar{b})$ holds, and applying the operation Γ to this collection. Depending on the operation Γ it may be important that this collection is understood as a multiset, rather than a set, which means that duplicates are not eliminated. Standard aggregates, which are present in all commercial database systems, include count, max, min, sum, product, and statistical aggregate functions such as average or standard deviation, and for all of theses, except max and min, it is essential that we treat the collection of values $\varphi(\bar{a}, \bar{b})$ as a multiset.

Some database systems also have mechanisms for user-defined aggregate operations [28], so a bit of theory is useful to determine what an aggregate precisely is and what properties it should satisfy. Since the outcome of a query should not depend on the internal representation of the database in the system, aggregate function need to be defined on the basis of commutative and associative operations, which naturally leads to monoids.

Beyond databases, a general concept of aggregate functions based on multiset operations has also been defined in metafinite model theory [14].

B. Summation problems for multisets and monoid aggregation functions

Multisets (or bags) generalize sets by admitting multiple occurrences of elements. Given that for some applications, summation problems over multisets are more natural than over sets, it is useful to notice that, for the concerns of this paper, there is no important difference. We can easily go back and forth between summation problems for sets and for multisets.

Formally, a multiset X over a domain A is given by a function $m_X : A \to \mathbb{N}$ and can be seen as the set of pairs $\{(a,n) : a \in A, n < m(a)\} \subseteq A \times \mathbb{N}$. If $m_X(a) = 0$ then we say that a does not occur in X. The union $X \cup Y$ of two multisets X, Y over A is given by the function $m_{X\cup Y}(a) = m_X(a) + m_Y(a)$. Finite multisets can also be described by listing the elements explicitly in the form $\{\!\{a, a, b, b, b, \ldots\}\!\}$. For any set A, let $\mathrm{Mult}(A)$ be the collection of all multisets over A.

An instance of the multiset summation problem for Abelian monoids is given by a monoid $(A, +, 0)$ and a function $m_X : A \to \mathbb{N}$ (describing a multiset X), with the task of determining $\sum X = \sum_{a\in A} m_X(a) \cdot a$ in A. The set summation problem is just the special case where m_X only takes values in $\{0, 1\}$. Conversely, every multiset summation problem, given by $(A, +, 0, m : A \to \mathbb{N})$, is equivalent to the set summation problem in the monoid $A \times \mathbb{N}$ with $(a, i) + (b, j) := (a + b, 0)$ over the the set $X = \{(a, i) : a \in A, i < m(a)\}$.

Clearly $(\mathrm{Mult}(A), \cup, \varnothing)$ is a commutative monoid, called the *multiset monoid over A*. Let $\mathfrak{M} = (M, +, 0)$ be an arbitrary commutative monoid. Then every map $h : A \to M$ induces a monoid homomorphism $H : \mathrm{Mult}(A) \to \mathfrak{M}$ generated by $H(\{\!\{a\}\!\}) := h(a)$.

This leads to a rather general kind of aggregate operations, the *monoid aggregates* [8]. In fact, it has been argued that "most aggregates in databases are either monoidal or can be obtained from monoidal aggregates by means of simple arithmetic operations" [21].

Definition 17. A *monoid aggregation function* H on a domain A is given by an arbitrary commutative monoid $\mathfrak{M} = (M, +, 0)$ and a function $h : A \to M$. The function H is then the homomorphism induced by h, from the multiset monoid over A into $\mathfrak{M}$. It associates with every multiset $X \in \mathrm{Mult}(X)$ the value $H(X) \in M$.

Note that the computation of the aggregation $H(X)$ corresponds to solving the summation problem over a multiset in the underlying monoid M. Indeed, $H(X) = \sum\{\!\{ha : a \in X\}\!\}$.

For such operations, the underlying monoid can be finite or infinite, but the multisets over which the sum is computed is always finite.

The standard aggregate operations present in, say, SQL, are based on monoids that are totally ordered by the monoid operation (in the sense that $x \leq y \leftrightarrow \exists z(x + z = y)$, but there are very simple other aggregates for which this is not the case. The simplest one is the parity operation.

Our results on the Abelian semigroup summation problem can be understood as saying that a database query language with the power to formulate monoidal aggregates can in fact also define the *results* of such aggregate queries, if it provides both *fixed-point recursion and counting*, but that relational recursion alone may be insufficient.

C. Database provenance.

There are several other areas in databases where the Abelian semigroup summation problem arises. Several variants of calculations in relational algebra with *annotated relations* appear for incomplete databases, probabilistic databases, bag semantics and provenance. It has been shown, for instance in [17], that these can be understood as instances of a general algorithmic approach involving semiring computations. Based on an evaluation of the atomic facts not just by Boolean truth values, but by values in an arbitrary commutative semiring, one can compute values for arbitrary queries from relational algebra or Datalog, which provide more information than just the truth or falsity of the query, and are relevant for issues such as cost, reliability, confidence, access control and others.

While the complexity of computing such values in a semiring $(K, +\cdot, 0, 1)$ may be hard, in fact #P-complete in many cases, there are simpler instances of queries that require solving the summation problem in the additive monoid $(K, +, 0)$ of the semiring. In particular, this is the case for existential projections. Given a query $\varphi(\bar{x}) := \exists\bar{y}R(\bar{x}, \bar{y})$ and a valuation V mapping atoms $R(\bar{a}, \bar{b})$ to K then the value of $\varphi(\bar{a})$ is given as the sum, in $(K, +, 0)$ over the set of all values $V(R(\bar{a}, \bar{b}))$ for arbitrary tuples $\bar{b}$.

VIII. Conclusion

We proved that the summation problem in Abelian semigroups (ASSP) is definable in fixed-point logic with counting. Furthermore, we showed that counting is really necessary: the summation problem cannot be expressed in Choiceless Polynomial Time without counting even over Abelian groups. Moreover, we saw that over Abelian groups the summation problem can be reduced to systems of linear equations over finite rings which lead to a definability result for first-order solvability logic. These results clarify the descriptive-complexity-theoretic status of the ASSP and, in particular, they answer an open question of Rossman.

Let us mention some questions for future research. First of all, our definability result for first-order solvability logic (FOS) only holds over Abelian *groups*, and it remains open whether it can be generalised to Abelian *semigroups*. At least, note that our proof techniques for the lower bounds for Choiceless Polynomial Time without counting do not work for FOS, because in FOS one can express modulo counting. Also it would be interesting to study whether for the FOS-definability result, solvability quantifiers over finite *rings* are really necessary or whether solvability quantifiers over finite *fields* suffice. This question is related to the question of whether rank operators over finite fields can express the solvability of linear equation systems over finite rings, see e.g. [10].

A second question is whether the ASSP can also be expressed in logics with data complexity in LOGSPACE, such as STC or LREC with counting [20]. Note that the ASSP can clearly be decided by a LOGSPACE-algorithm.

Finally, let us remind of our question from Section VI where we asked for classes of (infinite) semigroups over which the summation problem is definable in LFP. More generally, it would be interesting to identify classes of semigroups over which LFP can express *every* polynomial-time property of semigroups extended by unary predicates. Of course, it would also be quite interesting to study this question for stronger logics such as FPC and CPT, and also over Abelian groups. In particular, we ask whether FPC captures polynomial time over the class of all finite Abelian groups with unary predicates. Indeed, over the class of all (pure) Abelian groups $(G, +)$ one can show that FPC can express every polynomial-time property via a simple canonisation argument. However, as we saw in this paper, things already become non-trivial if we add a single unary predicate X and try to express a computationally very simple polynomial-time property, such as the summation problem, in FPC.

REFERENCES

[1] F. Abu Zaid, E. Grädel, M. Grohe, and W. Pakusa. Choiceless Polynomial Time on structures with small Abelian colour classes. In *Mathematical Foundations of Computer Science 2014*, pages 50–62. Springer, 2014.

[2] M. Anderson, A. Dawar, and B. Holm. Solving Linear Programs without Breaking Abstractions. *J. ACM*, 62(6):48:1–48:26, 2015.

[3] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

[4] A. Blass and Y. Gurevich. Strong extension axioms and Shelah's zero-one law for choiceless polynomial time. *J. Symb. Log.*, 68(1):65–131, 2003.

[5] A. Blass and Y. Gurevich. A Quick Update on the Open Problems in Blass-Gurevich-Shelah's article "On Polynomial Time Computations Over Unordered Structures". http://research.microsoft.com/en-us/um/people/gurevich/Opera/150a.pdf, 2005.

[6] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1):141–187, 1999.

[7] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.

[8] S. Cohen, Y. Sagiv, and W. Nutt. Equivalences among Aggregate Queries with Negation. *ACM Transactions on Computational Logic*, 6:328 – 360, 2005.

[9] A. Dawar. The Nature and Power of Fixed-point Logic with Counting. *ACM SIGLOG News*, pages 8–21, 2015.

[10] A. Dawar, E. Grädel, B. Holm, E. Kopczynski, and W. Pakusa. Definability of linear equation systems over groups and rings. *Logical Methods in Computer Science*, 9(4), 2013.

[11] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with Rank Operators. In *LICS '09*, pages 113–122. IEEE Computer Society, 2009.

[12] H. Ebbinghaus and J. Flum. *Finite model theory*. Springer Science, 2005.

[13] E. Grädel et al. *Finite Model Theory and Its Applications*. Springer, 2007.

[14] E. Grädel and Y. Gurevich. Metafinite Model Theory. *Information and Computation*, 140:26–81, 1998.

[15] E. Grädel and M. Otto. Inductive Definability with Counting on Finite Structures. In *Computer Science Logic, CSL 92*, volume 702 of *LNCS*, pages 231–247. Springer, 1992.

[16] E. Grädel and W. Pakusa. Rank logic is dead, long live rank logic! In *Computer Science Logic (CSL'15)*, Leibniz International Proceedings in Informatics (LIPIcs), 2015.

[17] T. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS 07*, 2007.

[18] M. Grohe. The Quest for a Logic Capturing PTIME. In *Logic in Computer Science, 2008, (LICS'08)*, pages 267–271. IEEE, 2008.

[19] M. Grohe. Fixed-Point Definability and Polynomial Time on Graph with Excluded Minors. *J. ACM*, 59(5):27:1–27:64, 2012.

[20] M. Grohe, B. Grußien, A. Hernich, and B. Laubner. L-Recursion and a new Logic for Logarithmic Space. *Logical Methods in Computer Science*, 9(1), 2012.

[21] L. Hella, L. Libkin, J. Nurmonen, and L. Wong. Logics with Aggregate Operators. *Journal of the ACM*, 48:880–907, 2001.

[22] B. Holm. *Descriptive complexity of linear algebra*. PhD thesis, University of Cambridge, 2010.

[23] B. Laubner. *The structure of graphs and new logics for the characterization of Polynomial Time*. PhD thesis, Humboldt-Universität Berlin, 2011.

[24] L. Libkin. *Elements of finite model theory*. Springer Science, 2013.

[25] M. Otto. The Expressive Power of Fixed-Point Logic with Counting. *J. Symb. Log.*, 61(1):147–176, 1996.

[26] W. Pakusa. *Linear Equation Systems and the Search for a Logical Characterisation of Polynomial Time*. PhD thesis, RWTH Aachen University, 2016.

[27] Saharon Shelah. *Choiceless Polynomial Time Logic: Inability to Express*, pages 72–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[28] H. Wang and C. Zaniolo. User-Defined Aggregates in Database Query Languages. In *DBPL 99*, pages 43–60, 2000.