Forum of Mathematics, Sigma (2018), Vol. 6, e1, 37 pages
doi:10.1017/fms.2017.29

COMPLEXITY OF SHORT GENERATING FUNCTIONS

DANNY NGUYEN and IGOR PAK

Department of Mathematics, University of California, Los Angeles, USA;
email: ldnguyen@math.ucla.edu, pak@math.ucla.edu

Received 13 March 2017; accepted 24 November 2017

Abstract

We give complexity analysis for the class of *short generating functions*. Assuming $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, we show that this class is not closed under taking many intersections, unions or projections of generating functions, in the sense that these operations can increase the bit length of coefficients of generating functions by a super-polynomial factor. We also prove that *truncated theta functions* are hard for this class.

2010 Mathematics Subject Classification: 68Q17 (primary); 68Q15 (secondary)

1. Introduction

1.1. Combinatorics and complexity of GFs.

A *short generating function* (short GF) is a rational generating function written in the form

$$(*) \qquad f(t) = \sum_{i=1}^{M} \frac{c_i t^{a_i}}{(1-t^{b_{i1}})\cdots(1-t^{b_{ik_i}})},$$

where $c_i = p_i/q_i \in \mathbb{Q}$, $a_i, b_{ij} \in \mathbb{Z}$ and $b_{ij} \neq 0$ for all i, j. The index$(f) := \max\{k_1, \ldots, k_M\}$ is the maximum number of terms in the denominators. This is always assumed to be bounded by some constant. The *length* $\ell(f)$ is defined as the total bit lengths of all constants in $(*)$. Of course, the same generating function can have many presentations as a short GF. (We also caution the reader that in general, the word *short* in 'short GF' only means that the GF is given in the form $(*)$. It does not necessarily mean the GF has polynomial length.)

In this paper we initiate the study of complexity of short GFs with bounded index and polynomial lengths. For a finite set $S \subset \mathbb{N}$, denote by $f_S(t) = \sum_{n\in S} t^n$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

the GF of S. We are interested in deciding if it is possible to write f_S as a short GF with polynomial length for a variety of sets S coming from Combinatorics, Number Theory and Discrete Geometry. Showing that some sets do not have short GFs of polynomial lengths turns out to be a surprisingly difficult problem. We are also interested in operations on short GFs and how they affect the short GFs' lengths.

Our approach is motivated by ideas from the study of integer points in convex polyhedra in fixed dimension (see Section 12.1). All such polyhedra turn out to have (multivariate) short GFs of polynomial lengths (see Definition 3.4 and Barvinok's Theorem 3.16 below). We refer to [**B2**, **B3**] for a thorough review of past and recent work on short GFs in Discrete Geometry, and to Section 12 for connections to Arithmetic Combinatorics and other areas.

1.2. Squares. Define the *truncated theta function* ϑ_r to be the GF over the squares $\leqslant 2^r$:

$$\vartheta_r(t) = \sum_{n=0}^{2^{r/2}} t^{n^2}.$$

CONJECTURE 1.1 (= Conjecture 9.1). *For every fixed $k \geqslant 1$, the truncated theta function $\vartheta_r(t)$* cannot *be written a short GF of length* poly(r) *and* index$(\vartheta_r) \leqslant k$.

The following result is the most surprising result of this paper:

THEOREM 1.2 (= Theorem 9.3). *If* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, *then Conjecture 1.1 holds.*

In other words, if each truncated theta function can be represented as a short GF of polynomial length and bounded index, then any counting problem can be solved with polynomial size circuits. See Section 12.5 for more on the complexity assumption, and Section 9 for the related results on primes.

1.3. One variable operations. Recall that we only consider GFs of finite sets. We define operations on GFs based on their supports. For example, taking the union of two GFs $f(t)$ and $g(t)$ means finding another GF $h(t)$ with $\operatorname{supp}(h) = \operatorname{supp}(f) \cup \operatorname{supp}(g)$. We can similarly define other Boolean operations.

Short GFs are known to be very versatile and useful in applications. Notably, given a bounded number of short GFs, all Boolean operations on them can be performed in polynomial time (see [**B3**, **BP**]). The result is again a short GF with polynomial length. However, when the number of short GFs is large, no such

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

polynomial time procedures are known. The following result gives a strong evidence against such possibility:

THEOREM 1.3 (= Theorem 8.1). *If* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, *then taking intersection/ union of many short GFs does not preserve polynomiality in length.*

This says taking union of many short GFs is hard structurally. It should be compared to an earlier result by Woods, which says that taking union of many short GFs is hard algorithmically, assuming $\mathsf{P} \neq \mathsf{NP}$ (see Theorem 3.21 and the following remark).

Next, define the *Minkowski sum* $f \oplus g$ of two GFs $f(t)$ and $g(t)$, to be the GF $h(t)$ with $\mathrm{supp}(h) = \mathrm{supp}(f) \oplus \mathrm{supp}(g) = \{a + b \mid a \in \mathrm{supp}(f), b \in \mathrm{supp}(g)\}$.

THEOREM 1.4 (= Theorem 8.4). *If* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, *then taking Minkowski sum of* two *short GFs does not preserve polynomiality in length.*

Giving precise formulations of these results requires some effort, see Section 8. Let us mention that in both theorems we can substitute the complexity assumptions with Conjecture 1.1. These results show strong limitations of the 'short GF technology' from a geometric point of view (see Section 12.1). Below we give further evidence of this phenomenon.

1.4. Projections. For multivariate short GFs, taking projections is a key operation. Projection is crucial for applications such as Integer Programming (see for example, [**Eis**, **Kan**, **NP2**]), and theoretical considerations such as Presburger Arithmetic (see for example, [**B2**, **NP1**, **W2**] and Section 3.2). In a crucial development, Barvinok and Woods [**BW**] showed that given a polytope P in bounded dimension, the projections of its integer points on some subspace have a short GF of polynomial length, which can also be computed in polynomial time (Theorem 3.19). This result exploited the polytopal structure of P and its convexity in a crucial way. Unfortunately, these are also the reasons that prevent their result to apply on a nongeometric level. In other words, the algorithm by Barvinok and Woods cannot produce a short GF for the projections if the input is presented only as short GF, without a polytope associated to it.

An important negative result by Woods in fact shows that given only a multivariate short GF $f(\mathbf{t})$, computing its projection is coNP-hard (see Theorem 3.21 and Remark 3.22). The following theorem is the central result of the paper. Roughly speaking, it both weakens the assumptions and strengthens the conclusions of Woods's theorem.

THEOREM 1.5 (= Corollary 7.2). *If* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$*, then taking projection of a short GF does not preserve polynomiality in length.*

This says that in general not only we cannot *compute* the projection of a short GF in polynomial time, any short GF that represents the projection must have a super-polynomial length. In other words, the barriers of using the 'short GF technology' in this case are structural rather than algorithmic.

The next result can be viewed as a refinement of the previous theorem, giving a precise characterization of complexity of projections.

THEOREM 1.6 (= Theorem 6.4). *Repeated projections of short GFs can encode every language in the nonuniform polynomial hierarchy* $\mathsf{PH/poly}$. *In fact, they form a hierarchy that coincides with* $\mathsf{PH/poly}$.

We postpone the precise formulations of these results, especially of Theorem 1.6 where the technicalities are unavoidable. Let us also mention Proposition 7.3 which can be viewed as a partial converse of Theorem 1.5 (see Section 9.3). (By itself, Conjecture 1.1 does not necessarily imply that $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, so a stronger assumption is used in Proposition 7.3.)

1.5. Paper structure. The results in this paper are largely self-contained and require little more than a few technical lemmas from [**BP**], which are all stated in Section 3 and can be treated as black boxes. We do however employ a fair amount of definitions and notations (Sections 2 and 3). We also assume the reader is familiar with basic Computational Complexity, which goes to the heart of this paper. We refer the reader to [**MM**, **Pap**] for the standard results and notation, and to [**Aar**] for a comprehensive recent survey.

Our Section 4 is the key as it describes the connection between languages and short GFs. From this point on, the reader can proceed to the development of the short GF hierarchy, culminating in the proofs of theorems 1.5 and 1.6 (Sections 5–7). Alternatively, modulo a few definitions in earlier section, the reader proceed directly to the proof of theorems 1.3 and 1.4 in Section 8. Similarly, the reader can also proceed to study complexity of squares and primes (Section 9). In Section 10 we investigate more technical questions on relative complexity of short GFs, and in Section 11 we give a proof of a technical Lemma 4.10. We conclude with final remarks and open problems in Section 12.

2. Notations

We use $\mathbb{N} = \{0, 1, 2, \ldots\}$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

All constant vectors are denoted as $\overline{a}, \overline{b}, \overline{c}, \overline{d}, \overline{n}$, and so on. The all 1 vector is also denoted by 1.
Matrixes are denoted as A, B, C, and so on.
Single variables are denoted as x, y, z, and so on; vectors of variables are denoted as $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, and so on.
We write $\mathbf{x} \leqslant \mathbf{y}$ if $x_j \leqslant y_j$ for all i.
For two tuples $\mathbf{x}$ and $\mathbf{t}$ both of length n, we denote by $\mathbf{t}^{\mathbf{x}}$ the monomial $t_1^{x_1} \ldots t_n^{x_n}$.
GF is an abbreviation for '*generating function*'.
Single-variable GFs are denoted as $f(t)$, $g(t)$, $h(t)$, $p(t)$, $q(t)$, and so on.
Multivariable GFs are denoted as $f(\mathbf{t})$, $g(\mathbf{t})$, $h(\mathbf{t})$, $p(\mathbf{t})$, $q(\mathbf{t})$, and so on.
The support of a GF $f(\mathbf{t})$ is denoted by $\operatorname{supp}(f)$.
The symbols $\neg$, $\wedge$ and $\vee$ denote negation (complement), conjunction and disjunction.
A *polyhedron* is an intersection of finitely many closed half-spaces in $\mathbb{R}^n$.
A *polytope* is a bounded polyhedron.
Polyhedra and polytopes are denoted as P, Q, R, and so on.
The function $\ell(\cdot)$ denotes the *bit length* of a number, vector, matrix, GF, or a logical formula when written in binary.
For a polyhedron Q described by a linear system $A\mathbf{x} \leqslant \overline{b}$, we denote by $\ell(Q)$ the total length $\ell(A) + \ell(\overline{b})$.

3. Polynomial time operations on short GFs

3.1. Preliminaries on short GFs. A power series $f(\mathbf{t}) = \sum \alpha_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$ is called a GF if each coefficient $\alpha_{\mathbf{x}}$ is either 0 or 1. When needed, we will write $f(\mathbf{t}) = \sum \mathbf{t}^{\mathbf{x}}$ to emphasize that f is a GF.

DEFINITION 3.1. The support of an n-variable GF $g(\mathbf{t}) = \sum \mathbf{t}^{\mathbf{x}}$ is defined as:

$$\operatorname{supp}(g) := \{\mathbf{x} \in \mathbb{Z}^n : [\mathbf{t}^{\mathbf{x}}]g(\mathbf{t}) = 1\}.$$

Here $[\mathbf{t}^{\mathbf{x}}]$ denotes the coefficient of the monomial $\mathbf{t}^{\mathbf{x}}$ in $g(\mathbf{t})$.

DEFINITION 3.2. Given a multivariable GF $f(\mathbf{t}, \mathbf{u}) = \sum \mathbf{t}^{\mathbf{x}} \mathbf{u}^{\mathbf{y}}$ with $\mathbf{x} \in \mathbb{Z}^m$, $\mathbf{y} \in \mathbb{Z}^n$, the $\mathbf{x}$-*projection* $g = \operatorname{proj}_{\mathbf{x}}(f)$ is the unique GF $g(\mathbf{t}) = \sum \mathbf{t}^{\mathbf{x}}$ with support satisfying

$$\operatorname{supp}(g) = \{\mathbf{x} \in \mathbb{Z}^m : \exists \mathbf{y} \in \mathbb{Z}^n \quad (\mathbf{x}, \mathbf{y}) \in \operatorname{supp}(f)\}.$$

If f satisfies the extra property that for every $\mathbf{x} \in \mathbb{Z}^m$ there is at most one $\mathbf{y} \in \mathbb{Z}^n$ such that $(\mathbf{x}, \mathbf{y}) \in \operatorname{supp}(f)$, then $\operatorname{proj}_{\mathbf{x}}(f)$ is called the $\mathbf{x}$-*specialization* of f, denoted by $\operatorname{spec}_{\mathbf{x}}(f)$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

DEFINITION 3.3. Consider two power series $f(\mathbf{t}) = \sum \alpha_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$ and $g(\mathbf{t}) = \sum \beta_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$. The *Hadamard product* of f and g, denoted by $f \star g$, is another GF $h(\mathbf{t}) = \sum \gamma_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$ with

$$\gamma_{\mathbf{x}} = \alpha_{\mathbf{x}} \beta_{\mathbf{x}} \text{ for every } \mathbf{x}.$$

If f and g are GFs then the above condition is equivalent to $\operatorname{supp}(h) = \operatorname{supp}(f) \cap \operatorname{supp}(g)$.

DEFINITION 3.4. For a rational function in n variables $\mathbf{t} = (t_1, \ldots, t_n)$ of the form

$$(\circledast) \qquad f(\mathbf{t}) = \sum_{i=1}^{M} \frac{c_i \mathbf{t}^{\overline{a}_i}}{(1 - \mathbf{t}^{\overline{b}_{i1}}) \cdots (1 - \mathbf{t}^{\overline{b}_{ik_i}})},$$

the length $\ell(f)$ of f is defined as

$$\ell(f) = \sum_{i} \lceil \log_2 |p_i q_i| + 1 \rceil + \sum_{i,j} \lceil \log_2 a_{ij} + 1 \rceil + \sum_{i,j,m} \lceil \log_2 b_{ijm} + 1 \rceil,$$

where $c_i = p_i/q_i \in \mathbb{Q}$, $\overline{a}_i, \overline{b}_{ij} \in \mathbb{Z}^n$, $\overline{b}_{ij} \neq 0$ and $\mathbf{t}^{\overline{a}} = t_1^{a_1} \cdots t_n^{a_n}$ if $\overline{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$.

DEFINITION 3.5. For a power series $f(\mathbf{t}) = \sum \alpha_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$ given in the form $(\circledast)$, the *index* of f is defined as

$$\operatorname{index}(f) = \max\{k_i \ : \ i = 1, \ldots, M\},$$

where k_i is the number of factors in the denominator of the ith summand.

DEFINITION 3.6. For every number of variables n and integer s, we define two classes:

$$\mathcal{GF}_{n,s} = \{\text{GFs } g(\mathbf{t}) \text{ given in the form } (\circledast) \text{ with } \operatorname{index}(g) \leqslant s\} \qquad (3.1)$$

and

$$\mathcal{GF}^*_{n,s} = \{\text{power series } g(\mathbf{t}) \text{ given in the form } (\circledast) \text{ with } \operatorname{index}(g) \leqslant s\}. \qquad (3.2)$$

Members of $\mathcal{GF}_{n,s}$ are called *short GFs*, while those of $\mathcal{GF}^*_{n,s}$ are called *short power series*.

We recall the following important results from [BP] (see also [BW]):

THEOREM 3.7 [BP]. *Fix a class $\mathcal{GF}_{m,s}$. Given a short GF $f(\mathbf{t}) \in \mathcal{GF}_{m,s}$ of finite support. We can compute in time* $\operatorname{poly}(\ell(f))$ *the following:*

(1) *the norm* $N = \max\{|\mathbf{x}| : \mathbf{x} \in \operatorname{supp}(f)\}$*; (Here* $|\mathbf{x}|$ *can be any polyhedral norm on* $\mathbf{x}$*, including* $|\mathbf{x}|_\infty$ *and* $|\mathbf{x}|_1$*.)*

(2) *the cardinality* $M = |\operatorname{supp}(f)|$*, which is equal to* $f(1)$*;*

(3) *the substitution* $q(\mathbf{u}) = f(\mathbf{t}(\mathbf{u}))$*, where* $\mathbf{t}$ *is substituted by monomials in some other variables* $\mathbf{u} = (u_1, \ldots, u_n)$*. Furthermore, we have* $q(\mathbf{u}) \in \mathcal{GF}^*_{n,s}$*.*

THEOREM 3.8 [BP]. *Fix two classes* $\mathcal{GF}_{m,s_1}$ *and* $\mathcal{GF}_{m,s_2}$*. Given* $f(\mathbf{t}) \in \mathcal{GF}_{m,s_1}$ *and* $g(\mathbf{t}) \in \mathcal{GF}_{m,s_2}$ *of finite supports, we can compute in time* $\operatorname{poly}(\ell(f) + \ell(g))$ *the following:*

(1) *a short GF* $h(\mathbf{t})$ *with* $\operatorname{supp}(h) = \operatorname{supp}(f) \cap \operatorname{supp}(g)$*, that is,* $h(\mathbf{t}) = f(\mathbf{t}) \star g(\mathbf{t})$*;*

(2) *a short GF* $k(\mathbf{t})$ *with* $\operatorname{supp}(k) = \operatorname{supp}(f) \cup \operatorname{supp}(g)$*;*

(3) *a short GF* $p(\mathbf{t})$ *with* $\operatorname{supp}(p) = \operatorname{supp}(f) \backslash \operatorname{supp}(g)$*.*

Moreover, we have $h, k, p \in \mathcal{GF}_{m,s_1+s_2}$*.*

REMARK 3.9. In fact, a more general version of Theorem 3.8 part (1) was shown in [BP], which also allows taking $f \star g$ for short power series.

The following is the reason why we emphasized the bounded dimension n and index s in Definition 3.6.

PROPOSITION 3.10. *Fix* n *and* s*. Given a short power series* $f(\mathbf{t}) = \sum \beta_{\mathbf{x}} \mathbf{t}^{\mathbf{x}}$ *in* $\mathcal{GF}_{n,s}$ *and a vector* $\overline{a}_0 \in \mathbb{Z}^n$*, the coefficient* $\beta_{\overline{a}_0}$ *can be computed in time* $\operatorname{poly}(\ell(f) + \ell(\overline{a}_0))$.

Proof. We let $g(\mathbf{t}) = \mathbf{t}^{\overline{a}_0}$ and define $h(\mathbf{t}) = f(\mathbf{t}) \star g(\mathbf{t})$. Clearly, we have $h(\mathbf{t}) = \beta_{\overline{a}_0} \mathbf{t}^{\overline{a}_0}$, which implies $\beta_{\overline{a}_0} = h(1)$. Applying Theorem 3.8, we can compute $h(\mathbf{t})$ (see also Remark 3.9). By Theorem 3.7, we can compute $h(1)$. All can be done in time $\operatorname{poly}(\ell(f) + \ell(\overline{a}_0))$. □

REMARK 3.11. A similar result for n and s unbounded is unlikely to hold, considering the fact that KNAPSACK is NP-complete. An instance of KNAPSACK asks if an equation $a = \overline{b}\mathbf{x}$ is solvable, where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{N}$ are variables, and $a \in \mathbb{N}$, $\overline{b} \in \mathbb{N}^n$ are given as input. This is equivalent to checking if $[t^a] f \neq 0$, where:

$$f(t) = \frac{1}{(1 - t^{b_1}) \cdots (1 - t^{b_n})}.$$

Here n is not bounded. Note that KNAPSACK has a polynomial time algorithm if a and $\overline{b}$ are given in unary. In our case, short GFs are encoded in binary.

If f is a short GF, Proposition 3.10 allows us to decide in polynomial time whether $\overline{a}_0 \in \operatorname{supp}(f)$. Now one may ask whether is it still easy to decide if a point $\overline{a}_0$ lies in a projection of f. The answer is still positive:

PROPOSITION 3.12. *Fix m, n and s. Given a short GF $f(\mathbf{t}, \mathbf{u}) = \sum \mathbf{t}^{\mathbf{x}} \mathbf{u}^{\mathbf{y}} \in \mathcal{GF}_{m+n,s}$ of finite support and a vector $\overline{a}_0 \in \mathbb{Z}^m$, checking whether $\overline{a}_0 \in$* $\operatorname{supp}(\operatorname{proj}_{\mathbf{x}}(f))$ *can be done in time* $\operatorname{poly}(\ell(f) + \ell(\overline{a}_0))$. *Here $\mathbf{x} \in \mathbb{Z}^m$, $\mathbf{y} \in \mathbb{Z}^n$.*

Proof. Let $g(\mathbf{t}) = f(\mathbf{t}, 1)$. Clearly, we have $\overline{a}_0 \in \operatorname{supp}(\operatorname{proj}_{\mathbf{x}}(f))$ if and only if the coefficient of $\mathbf{t}^{\overline{a}_0}$ in $g(\mathbf{t})$ is nonzero. By Theorem 3.7, we can compute g in time $\operatorname{poly}(\ell(f))$. By Proposition 3.10, we can compute $[\mathbf{t}^{\overline{a}_0}]g$ in time $\operatorname{poly}(\ell(g) + \ell(\overline{a}_0)) \leqslant \operatorname{poly}(\ell(f) + \ell(\overline{a}_0))$. □

In order to further study the projections of short GFs, we need a few tools.

3.2. Presburger arithmetic and short GFs.

Presburger arithmetic (PA) is the first order theory on the integers that allows only additions and inequalities. Each *atom* (smallest term) in PA is an integer inequality of the form

$$a_1 x_1 + \cdots + a_n x_n \leqslant b,$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ are integer variables, and $a_1, \ldots, a_n, b \in \mathbb{Z}$ are integer constants. A general PA formula is formed by taking Boolean combinations (negations, conjunctions, disjunctions) of such atoms, and also applying quantifiers ($\forall/\exists$) over different variables. A sentence in PA is a formula with all variables quantified. The length $\ell(F)$ of a PA formula F is the total length of all symbols and constants in F written in binary.

EXAMPLE 3.13. Let $P, Q \subseteq \mathbb{R}^n$ be two rational polyhedra given by two systems $A_1\mathbf{x} \leqslant \overline{b}_1$ and $A_2\mathbf{x} \leqslant \overline{b}_2$. Then the set of integer points in $P \cup Q$ is described by the PA formula:

$$F = \{\mathbf{x} : A_1\mathbf{x} \leqslant \overline{b}_1 \vee A_2\mathbf{x} \leqslant \overline{b}_2\}.$$

Here we are identifying the PA formula F with the set that it defines.

EXAMPLE 3.14. The PA formula $F = \{x : \forall y (5y \geqslant x + 1) \vee (5y \leqslant x - 1)\}$ determines the set of nonmultiples of 5.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

DEFINITION 3.15. For a set $S \subseteq \mathbb{Z}^n$, denote by $\mathbf{F}(S;\mathbf{t})$ the GF

$$\mathbf{F}(S;\mathbf{t}) = \sum_{\mathbf{x}\in S} \mathbf{t}^{\mathbf{x}}.$$

PA formulas are very well-suited to capture integer points in polyhedra. The following cornerstone result by Barvinok says that integer points in a polyhedron in bounded dimension can be effectively enumerated by a short GF.

THEOREM 3.16 [B1]. *Fix n. Let $Q \subseteq \mathbb{R}^n$ be a rational polyhedron described by $A\mathbf{x} \leqslant \overline{b}$. There exists a short GF $f \in \mathcal{GF}_{n,n}$ with $\mathbf{F}(Q\cap\mathbb{Z}^n;\mathbf{t}) = f(\mathbf{t})$, which can be computed in time* poly$(\ell(Q))$. (*This implies that $\ell(f) \leqslant$* poly$(\ell(Q))$.)

We mention a useful tool about quantifier free PA formulas:

PROPOSITION 3.17 [W1, Proposition 5.2.2]. *Fix n. Let $\Phi(\mathbf{x})$ be a Boolean combination of linear inequalities in integer variables $\mathbf{x} = (x_1,\ldots,x_n)$. Then we have:*

$$\Phi(\mathbf{x}) = \text{true} \qquad \Longleftrightarrow \qquad \bigvee_{i=1}^{r} \mathbf{x} \in P_i \cap \mathbb{Z}^n,$$

where $P_1,\ldots,P_r \subseteq \mathbb{R}^n$ are disjoint polyhedra and $r \leqslant$ poly$(\ell(\Phi))$. *The system defining each P_i can be computed in time* poly$(\ell(\Phi))$.

Theorem 3.16 can be generalized to quantifier free PA formula in bounded dimension:

THEOREM 3.18 [W1, Proposition 5.3.1]. *Fix n. Let $G = \{\mathbf{x}\in\mathbb{Z}^n : \Phi(\mathbf{x})\}$ be a* PA *formula with Φ a quantifier free Boolean combination of linear inequalities in* $\mathbf{x}$. *There exists a short GF $g \in \mathcal{GF}_{n,n}$ with $\mathbf{F}(G;\mathbf{t}) = g(\mathbf{t})$, which can be computed in time* poly$(\ell(\Phi))$.

Proof. By Proposition 3.17, we can rewrite Φ as a disjoint union of polyhedra $P_1,\ldots,P_r$ with $r \leqslant$ poly$(\ell(\Phi))$. The system defining each P_i can be computed in polynomial time. Applying Theorem 3.16, we get a short GF $f_i \in \mathcal{GF}_{n,n}$ of polynomial length for each P_i. Summing up all f_i, we get a short GF $g \in \mathcal{GF}_{n,n}$ of length poly$(\ell(\Phi))$ for G. □

Next, we consider PA formulas with quantifiers. In the simplest case, F encodes the projection of integer points in a polyhedron. For this, we have:

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

THEOREM 3.19 [BW, NP2]. *Fix $m, n \in \mathbb{N}$. Let $Q \subseteq \mathbb{R}^m$ be a rational polyhedron given by a system $A\mathbf{x} \leqslant \overline{b}$, and $T : \mathbb{Z}^m \to \mathbb{Z}^n$ a linear map. Consider the* PA *formula*

$$G = \{\mathbf{y} \in \mathbb{Z}^n : \exists \mathbf{x} \in \mathbb{Z}^m (\mathbf{x} \in Q) \wedge (\mathbf{y} = T\mathbf{x})\}.$$

Then there exists a short GF g with $\mathbf{F}(G; \mathbf{t}) = g(\mathbf{t})$, which can be computed in time $\text{poly}(\ell(Q) + \ell(T))$. *Furthermore, we have $g \in \mathcal{GF}_{n,s}$, where $s = s(m)$ is a constant.*

REMARK 3.20. The above theorem was proved in [BW] for the case when P is a polytope. It was recently extended in [NP2] to all (possibly unbounded) polyhedra.

However, for general $\exists$-formulas, finding a short GF for F becomes coNP-hard:

THEOREM 3.21 [W1, Theorem 5.3.2]. *Let $\Phi(x, y)$ be a quantifier free Boolean combination of linear inequalities in x and y (singletons). Consider*

$$F = \{y \in \mathbb{Z} : \exists x \in \mathbb{Z} \Phi(x, y)\}.$$

Then computing a short GF for F is coNP*-hard.*

REMARK 3.22. By Theorem 3.18, we still can find a short GF of length $\text{poly}(\ell(\Phi))$ for $\Phi(x, y)$. So this result says that projecting a short GF is hard algorithmically. This should be compared to our Theorem 1.5, which says that projecting short GF is hard structurally. Actually, by Proposition 3.17, we can also decompose $\Phi(x, y)$ into a union of polynomially many polygons $P_i \subseteq \mathbb{R}^2$. By Theorem 3.19, the projection of integer points in each P_i on x has a short GF, which can be found in polynomial time. So taking union of these short GFs is again hard algorithmically. This should be compared to Theorem 1.3.

4. Short GFs and the class P/poly

4.1. Encoding languages in P/poly as short GFs.

For technical reasons regarding the convergence of GFs under numerical evaluation, we consider only GFs with support in $\mathbb{N}^n$ from this section onwards. Theorem 3.8 still applies to short GFs supported on $\mathbb{N}^n$.

DEFINITION 4.1. For every language $\mathcal{L} \in \{0, 1\}^*$, and every $r > 0$, we denote by $\mathcal{L}_r$ the segment

$$\mathcal{L}_r := \{\widetilde{x} \in \{0, 1\}^r : \widetilde{x} \in \mathcal{L}\}. \tag{4.1}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

For $\widetilde{x} \in \mathcal{L}_r$, let x be the corresponding integer with binary representation $\widetilde{x}$. We will also use $\mathcal{L}_r$ to denote the set of all such x with $\widetilde{x} \in \mathcal{L}_r$.

LEMMA 4.2. *For every language $\mathcal{L} \in$ P/poly, and every $r > 0$, the segment $\mathcal{L}_r$ can be characterized in* PA *as:*

$$\widetilde{x} \in \mathcal{L}_r \quad \Longleftrightarrow \quad x \in [0, 2^r) \wedge [\exists y \in [0, 2^p) \forall \mathbf{z} \in [0, 2^q)^3 : \Phi_r(x, y, \mathbf{z})], \tag{4.2}$$

where Φ_r is a quantifier free PA *expression in $x, y \in \mathbb{N}$ and $\mathbf{z} \in \mathbb{N}^3$. Moreover, we have $p, q, \ell(\Phi_r) \leqslant \mathrm{poly}_{\mathcal{L}}(r)$. (We denote by $\ell(\Phi_r)$ the total length of all symbols Φ_r, written in binary. The notation $\mathrm{poly}_{\mathcal{L}}(r)$ denotes a polynomial in r, with the polynomial degree depending on the language $\mathcal{L}$.) If in addition $\mathcal{L} \in$ P, then there is an algorithm to compute p, q and Φ_r in time $\mathrm{poly}_{\mathcal{L}}(r)$.*

Proof. By definition of the class P/poly, there is a Boolean circuit C_r such that:

$$\mathcal{L}_r = \{\widetilde{x} \in \{0, 1\}^r : C_r(\widetilde{x}) = \text{true}\}.$$

Here the circuit C_r has r input gates, and as many as $p \leqslant \mathrm{poly}_{\mathcal{L}}(r)$ noninput gates, each with in-degree at most 2. We encode the values of the noninput gates as a Boolean string $\widetilde{y} \in \{0, 1\}^p$. Let $\widetilde{x} = (x_1, \ldots, x_r)$ and $\widetilde{y} = (y_1, \ldots, y_p)$. By a standard reduction (see for example [MM, Pap]), we can encode the computation of C_r by a Boolean formula F in 3-conjunctive normal form. Explicitly, we have:

$$\mathcal{L}_r = \{\widetilde{x} \in \{0, 1\}^r : \exists \widetilde{y} \in \{0, 1\}^p \, F(\widetilde{x}, \widetilde{y}) = \text{true}\}, \tag{4.3}$$

where

$$F(\widetilde{x}, \widetilde{y}) = \bigwedge_k (a_k \vee b_k \vee c_k). \tag{4.4}$$

Here each a_k, b_k, c_k is a literal in the set $\{x_i, \neg x_i, y_j, \neg y_j : 1 \leqslant i \leqslant r, 1 \leqslant j \leqslant p\}$.

Let $x \in [0, 2^r)$ and $y \in [0, 2^p)$ be the integers corresponding to $\widetilde{x}$ and $\widetilde{y}$, respectively. Every literal x_i corresponds to the ith digit in x being 1, and $\neg x_i$ corresponds that digit being 0. (The least significant digit in x corresponds to x_0 in $\widetilde{x}$.) In other words, x_i is true or false respectively when $\lfloor x/2^{i-1} \rfloor$ is odd or even. The same applies to y_i and y. Observe that $t = \lfloor x/2^{i-1} \rfloor$ is the only integer that satisfies $x/2^{i-1} - 1 < t \leqslant x/2^{i-1}$. Let $q = \max(r, p) \leqslant \mathrm{poly}(r)$. Each term x_i or $\neg x_i$ can be coded with an extra $\exists z$ quantifier as follows:

$$\begin{aligned} x_i \quad &\Longleftrightarrow \quad \exists z \in [0, 2^q) : \begin{Bmatrix} 2z + 1 > x/2^{i-1} - 1 \\ 2z + 1 \leqslant \quad x/2^{i-1} \end{Bmatrix}, \\ \neg x_i \quad &\Longleftrightarrow \quad \exists z \in [0, 2^q) : \begin{Bmatrix} 2z > x/2^{i-1} - 1 \\ 2z \leqslant \quad x/2^{i-1} \end{Bmatrix}. \end{aligned} \tag{4.5}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Here $\{\cdot\}$ denotes a system (conjunction) of inequalities. Analogously, each y_j or $\neg y_j$ can be coded using $\exists z$. Note that the two strict inequalities in (4.5) can be sharpened by multiplying both sides with 2^{i-1} to make all coefficients integer, and add 1 to the RHS.

Now we show how to code (4.4) using $\forall \mathbf{z}$ with $\mathbf{z} \in \mathbb{N}^3$. For each clause $(a_k \vee b_k \vee c_k)$, we consider its negation $(\neg a_k \wedge \neg b_k \wedge \neg c_k)$. Each term $\neg a_k, \neg b_k, \neg c_k$ is still one of $x_i, \neg x_i, y_i, \neg y_i$. By (4.5), we have

$$(\neg a_k \wedge \neg b_k \wedge \neg c_k) \quad \Longleftrightarrow \quad \exists \mathbf{z} \in [0, 2^q)^3 : \Phi_k(x, y, \mathbf{z}),$$

where $\mathbf{z} \in \mathbb{N}^3$, and Φ_k is a conjunction of six inequalities. Taking negation, we have:

$$\begin{aligned}(a_k \vee b_k \vee c_k) \quad &\Longleftrightarrow \quad \forall \mathbf{z} \in [0, 2^q)^3 : \neg\Phi_k(x, y, \mathbf{z}),\\ &\Longleftrightarrow \quad \forall \mathbf{z} \in [0, 2^q)^3 : \Psi_k(x, y, \mathbf{z}),\end{aligned}$$

where Ψ_k is a disjunction of six inequalities. Taking conjunction over all k in (4.4), we have:

$$F(\widetilde{x}, \widetilde{y}) \quad \Longleftrightarrow \quad \forall \mathbf{z} \in [0, 2^q)^3 : \Phi_r(x, y, \mathbf{z}), \tag{4.6}$$

where

$$\Phi_r(x, y, \mathbf{z}) = \bigwedge_k \Psi_k(x, y, \mathbf{z}). \tag{4.7}$$

Substituting (4.6) into (4.3), we have (4.2). If we assume in addition that $\mathcal{L} \in \mathsf{P}$, then the circuit C_r can be built from a Turing Machine in time $\operatorname{poly}_{\mathcal{L}}(r)$, so the expression Φ_r can also be found in time $\operatorname{poly}_{\mathcal{L}}(r)$. This completes the proof. □

DEFINITION 4.3. Given $f = \mathbf{F}(S; \mathbf{t})$, where S is a subset of a finite box $B \subset \mathbb{N}^n$. The *finite complement* $B \backslash f$ is $\mathbf{F}(B \backslash S; \mathbf{t})$.

DEFINITION 4.4. Given $f_1 = \mathbf{F}(S_1; \mathbf{t}), \ldots, f_k = \mathbf{F}(S_k; \mathbf{t})$ with $S_1, \ldots, S_k \subseteq \mathbb{N}^n$, the *intersection* $f_1 \cap \cdots \cap f_k$ is $\mathbf{F}(S_1 \cap \cdots \cap S_k; \mathbf{t})$. The *union* $f_1 \cup \cdots \cup f_k$ is $\mathbf{F}(S_1 \cup \cdots \cup S_k; \mathbf{t})$.

THEOREM 4.5. *For every language $\mathcal{L} \in \mathsf{P/poly}$ and $r > 0$, there exist a finite box B_r and short GF $f_r(t, u, \mathbf{v}) \in \mathcal{GF}_{5,5}$ with $\operatorname{supp}(f_r) \subseteq B_r$, so that*

$$\mathbf{F}(\mathcal{L}_r; t) = \operatorname{spec}_x(B_r \backslash \operatorname{proj}_{x,y}(f_r)) \tag{4.8}$$

and $\ell(B_r), \ell(f_r) \leqslant \operatorname{poly}_{\mathcal{L}}(r)$. (Here $\ell(Br)$ denotes the total bit length of all sides in B_r, written in binary.) Furthermore, there exist polynomially many short GFs $p_{r,1}, \ldots, p_{r,k_r} \in \mathcal{GF}_{2,s}$ of finite supports, each of length $\operatorname{poly}_{\mathcal{L}}(r)$, so that:

$$\operatorname{proj}_{x,y}(f_r) = p_{r,1} \cup \cdots \cup p_{r,k_r}. \tag{4.9}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Here $\mathcal{GF}_{2,s}$ is some fixed class that does not depend on $\mathcal{L}$. If we assume in addition that $\mathcal{L} \in \mathsf{P}$, then there is also an algorithm to compute B_r, f_r and each $p_{r,i}$ in time $\operatorname{poly}_{\mathcal{L}}(r)$.

Proof. For the notations proj, spec, $\cup$ and $\backslash$, we refer back to Definitions 3.2, 4.3 and 4.4. By the previous lemma, there is a PA expression Φ_r satisfying (4.2). First, define

$$\begin{aligned} B_r &= \{(x, y) : x \in [0, 2^r), y \in [0, 2^p)\}, \\ D_r &= \{(x, y, \mathbf{z}) : x \in [0, 2^r), y \in [0, 2^p), \mathbf{z} \in [0, 2^q)^3\}, \end{aligned}$$

where r, p and q are from (4.2). Define:

$$f_r(t, u, \mathbf{v}) = \sum_{\substack{(x,y,\mathbf{z}) \in D_r \\ \neg\Phi_r(x,y,\mathbf{z})}} t^x u^y \mathbf{v}^{\mathbf{z}}. \tag{4.10}$$

Recall that Φ_r is a quantifier free PA expression with length $\operatorname{poly}_{\mathcal{L}}(r)$. Applying Theorem 3.18 to $\neg\Phi_r$, we can write f_r as a short GF in $\mathcal{GF}_{5,5}$ of finite support, which has length $\ell(f_r) \leqslant \operatorname{poly}(\ell(\Phi_r)) \leqslant \operatorname{poly}_{\mathcal{L}}(r)$. For the rest of the proof, we always assume $(x, y, \mathbf{z}) \in D_r$. We will simply write $\exists \mathbf{z}$ instead of $\exists \mathbf{z} \in [0, 2^q)^3$. Projecting f_r on (x, y), we have:

$$\operatorname{proj}_{x,y}(f_r) = \sum_{(x,y):\exists\mathbf{z}\neg\Phi_r(x,y,\mathbf{z})} t^x u^y. \tag{4.11}$$

Taking the complement of $\operatorname{proj}_{x,y}(f_r)$, which lies within the box B_r, we have:

$$B_r \backslash \operatorname{proj}_{x,y}(f_r) = \sum_{(x,y):\forall\mathbf{z}\Phi_r(x,y,\mathbf{z})} t^x u^y. \tag{4.12}$$

Recall that in the proof of Lemma 4.2, the variable y describes the values of noninput gates in the circuit C_r, with input gates coming from x. Since the values of noninput gates are uniquely determined by the input gates, for every x that satisfies C_r we have a unique y. Substituting $u \leftarrow 1$, the RHS in (4.12) becomes $\mathbf{F}(\mathcal{L}_r; t)$. We obtain (4.8).

We proceed to show (4.9). Since $\neg\Phi_r$ is quantifier free with five variables, we can apply Proposition 3.17 on it and get:

$$\neg\Phi_r(x, y, \mathbf{z}) \quad \Longleftrightarrow \quad \bigvee_{i=1}^{k_r} (x, y, \mathbf{z}) \in P_{r,i} \cap \mathbb{N}^5,$$

where $P_{r,1}, \ldots, P_{r,k_r} \subseteq \mathbb{R}^5$ are disjoint polytopes (in the box D_r) and $k_r \leqslant \operatorname{poly}(\ell(\Phi_r)) \leqslant \operatorname{poly}_{\mathcal{L}}(r)$. Each polytope $P_{r,i}$ also satisfies $\ell(P_{r,i}) \leqslant \operatorname{poly}(r)$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Therefore:

$$\exists \mathbf{z} \neg \Phi_r(x, y, \mathbf{z}) \iff \bigvee_{i=1}^{k_r} \exists \mathbf{z}[(x, y, \mathbf{z}) \in P_{r,i} \cap \mathbb{N}^5]. \tag{4.13}$$

Combined with (4.11), we see that $(x, y) \in \operatorname{supp}(\operatorname{proj}_{x,y}(f_r))$ if and only if it lies in the projection of some $P_{r,i} \cap \mathbb{N}^5$. By Theorem 3.19, for each i, we can find a short GF $p_{r,i} \in \mathcal{GF}_{2,s}$ for the projection of $P_{r,i} \cap \mathbb{N}^5$. In other words, we have $p_{r,i} \in \mathcal{GF}_{2,s}$ that satisfies:

$$\operatorname{supp}(p_{r,i}) = \{(x, y) : \exists \mathbf{z}(x, y, \mathbf{z}) \in P_{r,i} \cap \mathbb{N}^5\}.$$

Here s is an absolute constant because each $P_{r,i}$ has (fixed) dimension 5. We also have $\ell(p_{r,i}) \leqslant \operatorname{poly}(\ell(P_{r,i})) \leqslant \operatorname{poly}(r)$. The union of all short GFs $p_{r,i}$ contains exactly all (x, y) satisfying (4.13). From (4.11) and (4.13), we have:

$$\operatorname{proj}_{x,y}(f_r) = p_{r,1} \cup \cdots \cup p_{r,k_r}.$$

This proves (4.9) and completes the proof. □

EXAMPLE 4.6. Since SQUARES and PRIMES are both in P, we can represent all squares or primes up to 2^r in the form (4.8), with f_r and B_r computable in time $\operatorname{poly}(r)$.

REMARK 4.7. Even though $\operatorname{spec}_x(B_r \backslash \operatorname{proj}_{x,y}(f_r))$ may seem complicated, the specialization and complement are 'inexpensive operations', which can be performed in polynomial time by Theorems 3.7 and 3.8. The main complexity resides in taking the projection of f.

REMARK 4.8. The same representation (4.8) applies to every language $\mathcal{L}$ in the complexity class $\mathsf{UP/poly}$. Such a language is characterized as follows. For every r, there is a *nondeterministic* polynomial time Turing machine that accepts only $x \in \mathcal{L}_r$, each with a unique accepting path. Given $\mathcal{L} \in \mathsf{UP/poly}$, we can obtain (4.8) by the same argument as above. In fact, (4.8) is an equivalent characterization of the class $\mathsf{UP/poly}$. Indeed, assume $\mathcal{L}_r$ can be represented as (4.8). Given f_r, for any $x \in \mathcal{L}_r$ there should be a unique certificate y such that $(x, y) \in B_r \backslash \operatorname{proj}_{x,y}(f_r)$, which is checkable in polynomial time by Proposition 3.12.

4.2. Compressing short GFs of finite supports. We describe a technical tool which will be useful later. This section can be skipped at first reading.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

DEFINITION 4.9. Consider $N = 2^r$ and a vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{N}^n$ with $x_i \in [0, N)$ for all $1 \leqslant i \leqslant d$. We define the τ_N map on $\mathbf{x}$ as:

$$\tau_N(\mathbf{x}) = x_1 + Nx_2 + \cdots + N^{n-1}x_d \in [0, N^n).$$

For an array of vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ with $\mathbf{x}_i \in [0, N)^{n_i}$, we define:

$$\tau_N(\bar{\mathbf{x}}) = (\tau_N(\mathbf{x}_1), \ldots, \tau_N(\mathbf{x}_k)) \in [0, N^{n_1}) \times \cdots \times [0, N^{n_k}).$$

Finally, for a set $S \subseteq [0, N)^{n_1} \times \cdots \times [0, N)^{n_k}$, we define $\tau_N(S) = \{\tau_N(\bar{\mathbf{x}}) : \bar{\mathbf{x}} \in S\}$.

The following technical tool allows us to reduce the number of variables in a short GF of finite support.

LEMMA 4.10. *Fix k, s and $n_1, \ldots, n_k \in \mathbb{N}$. Let $n = n_1 + \cdots + n_k$.*

(a) *Compressing. Given a short GF $g(\bar{\mathbf{t}}) = \sum \mathbf{t}_1^{\mathbf{x}_1} \ldots \mathbf{t}_k^{\mathbf{x}_k}$ of finite support in the class $\mathcal{GF}_{n,s}$, there exist an $N = 2^r$ with* supp$(g) \subseteq [0, N)^{n_1} \times \cdots \times [0, N)^{n_k}$ *and a short GF $f(\mathbf{u}) = \sum u_1^{z_1} \ldots u_k^{z_k}$ in the class $\mathcal{GF}_{k,s}$ so that*

$$\operatorname{supp}(f) = \tau_N(\operatorname{supp}(g)) \subseteq [0, N^{n_1}) \times \cdots \times [0, N^{n_k}). \tag{4.14}$$

Both f and N can be computed in time poly$(\ell(g))$ *with* $\ell(f)$, $\log N \leqslant$ poly$(\ell(g))$.

(b) *Decompressing. Conversely, given $f(\mathbf{u}) = \sum u_1^{z_1} \ldots u_k^{z_k} \in \mathcal{GF}_{k,s}$ and $N = 2^r$ such that*

$$\operatorname{supp}(f) \subseteq [0, N^{n_1}) \times \cdots \times [0, N^{n_k}),$$

there exists $g(\bar{\mathbf{t}}) = \sum \mathbf{t}_1^{\mathbf{x}_1} \ldots \mathbf{t}_k^{\mathbf{x}_k} \in \mathcal{GF}_{n,n+s}$ with supp$(g) \subseteq [0, N)^{n_1} \times \cdots \times [0, N)^{n_k}$ *which satisfies* (4.14). *The short GF g can be computed in time* poly$(\ell(f) + \log N)$.

Proof for the lemma is technical and is postponed until Section 11. We note that the compression map τ_N in Definition 4.9 is similar to that used in the polynomial identity testing algorithm of Klivans and Spielman [**KS**]. Using Lemma 4.10, we can reduce the number of variables of f_r in (4.8) down to 3.

COROLLARY 4.11. *For every language $\mathcal{L} \in$ P/poly and $r > 0$, there exist a finite box B_r and short GF $f_r(t, u, v) \in \mathcal{GF}_{3,5}$ with* supp$(f_r) \subseteq B_r$, *so that* (4.8) *holds. The rest is identical to Theorem* 4.5.

Proof. We have (4.8) with $f_r(t, u, \mathbf{v}) = \sum t^x u^y \mathbf{v}^{\mathbf{z}} \in \mathcal{GF}_{5,5}$ a short GF of finite support in five variables (t, u, v_1, v_2, v_3). Using part (a) of Lemma 4.10, we can

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

compress $\mathbf{z}$ into a single-variable w, leaving both x and y unchanged. In other words, $t^x u^y \mathbf{v}^{\mathbf{z}}$ becomes $t^x u^y v^w$. Note that $\mathrm{proj}_{x,y}$ is not affected by compression. This gives us a short GF $\widetilde{f}_r \in \mathcal{GF}_{3,5}$ with

$$\mathrm{proj}_{x,y}(\widetilde{f}_r) = \mathrm{proj}_{x,y}(f_r) \quad \text{and} \quad \ell(\widetilde{f}_r) \leqslant \mathrm{poly}(\ell(f_r)) \leqslant \mathrm{poly}(r).$$

So we can substitute $\widetilde{f}_r$ for f_r in (4.8). □

5. Short GFs and the nonuniform polynomial hierarchy

The nonuniform polynomial hierarchy $\mathsf{PH/poly}$ starts with $\mathsf{P/poly} = \boldsymbol{\Sigma}_0^{\mathsf{P}}\mathsf{/poly} = \boldsymbol{\Pi}_0^{\mathsf{P}}\mathsf{/poly}$ at the 0th level. For $k > 0$, a language $\mathcal{L}$ is in $\boldsymbol{\Sigma}_k^{\mathsf{P}}\mathsf{/poly}$ if for every $r > 0$, there is a circuit C_r of size $\mathrm{poly}_{\mathcal{L}}(r)$ so that for every string $\widetilde{x}$ of length r we have:

$$\widetilde{x} \in \mathcal{L}_r \quad \Longleftrightarrow \quad \exists \widetilde{y}_1 \forall \widetilde{y}_2 \ldots Q_k \widetilde{y}_k : C_r(x, y_1, \ldots, y_k) = 1.$$

Here $Q_1, \ldots, Q_k$ are k alternating quantifiers with $Q_1 = \exists$, and $\widetilde{y}_1, \ldots, \widetilde{y}_k$ are binary strings of length polynomial in r. For $\boldsymbol{\Pi}_k^{\mathsf{P}}\mathsf{/poly}$ the alternating quantifiers are reversed ($Q_1 = \forall$). We have the following analogue to Lemma 4.2 for each level in $\mathsf{PH/poly}$:

Lemma 5.1. *For every language $\mathcal{L} \in \boldsymbol{\Sigma}_k^{\mathsf{P}}\mathsf{/poly}$ and $r > 0$, there exists a quantifier free* PA *expression in $k+4$ variables $x \in \mathbb{N}$, $\mathbf{y} \in \mathbb{N}^k$, $\mathbf{z} \in \mathbb{N}^3$, so that $\widetilde{x} \in \mathcal{L}_r$ if and only if:*

$$x \in [0, 2^r) \wedge [Q_1 y_1 \in [0, 2^{p_1}) \ldots Q_k y_k \in [0, 2^{p_k}) Q_{k+1} \mathbf{z} \in [0, 2^q)^3 : \Phi_r(x, \mathbf{y}, \mathbf{z})]. \tag{5.1}$$

Here $Q_1, \ldots, Q_{k+1}$ are $k+1$ alternating quantifiers with $Q_1 = \exists$. Moreover, we have $p_1, \ldots, p_k, q, \ell(\Phi_r) \leqslant \mathrm{poly}_{\mathcal{L}}(r)$. For the case $\mathcal{L} \in \boldsymbol{\Pi}_k^{\mathsf{P}}\mathsf{/poly}$, the quantifiers Q_i are reversed.

Proof. For simplicity, we prove the claim for $\mathcal{L} \in \boldsymbol{\Sigma}_1^{\mathsf{P}} = \mathsf{NP/poly}$. The higher levels $\boldsymbol{\Sigma}_k^{\mathsf{P}}\mathsf{/poly}$ and $\boldsymbol{\Pi}_k^{\mathsf{P}}\mathsf{/poly}$ can be argued similarly. Since $\mathcal{L} \in \mathsf{NP/poly}$, for each r, there is a circuit C_r of size $\mathrm{poly}_{\mathcal{L}}(r)$ such that

$$\widetilde{x} \in \mathcal{L}_r \quad \Longleftrightarrow \quad \exists \widetilde{c} \in \{0, 1\}^s : C_r(\widetilde{x}, \widetilde{c}) = 1, \tag{5.2}$$

where $s \leqslant \mathrm{poly}_{\mathcal{L}}(r)$ is the certificate length. The circuit C_r also has p noninput gates with $p \leqslant \mathrm{poly}_{\mathcal{L}}(r)$. Let $p' = s + p$. Note that the certificate gates $\widetilde{c} \in \{0, 1\}^s$ and the noninput gates $\widetilde{y} \in \{0, 1\}^p$ can be coded by a single integer $y \in [0, 2^{p'})$. The argument now proceeds similarly to Lemma 4.2 with p' in place of p. □

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

REMARK 5.2. In [**Grä**, Lemma 5.2], Grädel gave a similar representation to (5.1). In his representation, each string $\widetilde{x} = (x_1, \ldots, x_r) \in \{0, 1\}^r$ is not simply mapped to its binary integer value, but to:

$$x = p_1^{x_1} \ldots p_r^{x_r} q_1^{1-x_1} \ldots q_r^{1-x_r},$$

where $p_1, \ldots, p_r, q_1, \ldots, q_r$ are the first $2r$ prime numbers.

REMARK 5.3. From this result, we see that the problem of deciding PA sentences of the form $\exists \mathbf{y} \forall \mathbf{z} \Phi(\mathbf{y}, \mathbf{z})$ is at least NP-hard. Schöning [**Sch**] showed that the problem is NP-complete even for the case $\exists y \forall z \Phi(y, z)$, that is, when both variables are singletons.

DEFINITION 5.4. Let $f = \sum \mathbf{t}^{\mathbf{x}} \mathbf{u}^{\mathbf{y}} = \mathbf{F}(S; \mathbf{t}, \mathbf{u})$, where S is a subset of a finite box $I \times J$. The *anti-projection* $\overline{\text{proj}}_{\mathbf{x}}(f)$ is $F(I; \mathbf{t}) - \text{proj}_{\mathbf{x}}(f)$, where the projection $\text{proj}_{\mathbf{x}}(f)$ is from Definition 3.2. The box $I \times J$ is always specified before taking the anti-projection.

THEOREM 5.5. *For every language $\mathcal{L} \in \mathbf{\Sigma}_k^{\mathsf{P}}/\mathsf{poly}$ and $r > 0$, there exists a short GF $f_r \in \mathcal{GF}_{k+2,k+4}$ of the form $f_r(t, u_1, \ldots, u_k, v) = \sum t^x u_1^{y_1} \ldots u_k^{y_k} v^z$ such that*

$$\mathbf{F}(\mathcal{L}_r; t) = \text{proj}_x \big(\overline{\text{proj}}_{x,y_1} (\text{proj}_{x,y_1,y_2} (\cdots (f_r) \cdots)) \big), \tag{5.3}$$

where the k alternating projections and anti-projections are taken in a finite box

$$B_r = [0, 2^r) \times [0, 2^{p_1}) \times \cdots \times [0, 2^{p_k}) \times [0, 2^q).$$

Moreover, we have $p_1, \ldots, p_k, q, \ell(f_r) \leqslant \text{poly}_{\mathcal{L}}(r)$. For $\mathcal{L} \in \mathbf{\Pi}_k^{\mathsf{P}}/\mathsf{poly}$, the projections and anti-projections are reversed.

Proof. By Lemma 5.1, we can represent $\mathcal{L}_r$ in the form (5.1). Applying the same argument in Theorem 4.5, we get $f_r(t, u_1, \ldots, u_k, \mathbf{v}) = \sum t^x u_1^{y_1} \ldots u_k^{y_k} \mathbf{v}^{\mathbf{z}} \in \mathcal{GF}_{k+4,k+4}$ that satisfy (5.3). Applying Lemma 4.10(a), we can compress the last three variables $\mathbf{v}^{\mathbf{z}} = v_1^{z_1} v_2^{z_2} v_3^{z_3}$ into just one variable v^w without affecting the projections (see the proof of Corollary 4.11). This reduces f_r to a short GF in $\mathcal{GF}_{k+2,k+4}$. □

REMARK 5.6. If in addition $L \in \mathsf{PH}$, then both Φ_r and f_r in Lemma 5.1 and Theorem 5.5 can be computed in time $\text{poly}_{\mathcal{L}}(r)$. Indeed, if $\mathcal{L} \in \mathsf{PH}$, the circuit C_r for $\mathcal{L}_r$ in Lemma 5.1's proof can be automatically generated by some polynomial time Turing Machine M. We can convert C_r to Φ_r in polynomial time, which allows us to find f_r.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

As a consequence, we obtain the following result.

COROLLARY 5.7. *Assume we are given* $a_0 \in \mathbb{N}$, *a short GF* $f(t,u,v) = \sum t^x u^y v^z \in \mathcal{GF}_{3,5}$, *and a finite box* $B \subset \mathbb{N}^3$ *with* $\operatorname{supp}(f) \subseteq B$. *Then deciding whether* $a_0 \in \operatorname{supp}(h)$ *is* NP*-complete, where* $h = \operatorname{proj}_x(\overline{\operatorname{proj}}_{x,y}(f))$. *Here the projection and anti-projection are taken within* B.

Proof. If $a_0 \in \operatorname{supp}(h)$, there exists some b_0 so that (a_0, b_0) lies in the support of $\overline{\operatorname{proj}}_{x,y}(f)$. Since $\overline{\operatorname{proj}}_{x,y}(f)$ is taken within B, which is bounded, both a_0 and b_0 must have polynomial lengths. Given such a certificate b_0, we can verify if (a_0, b_0) lies in the support of $\operatorname{proj}_{x,y}(f)$ in polynomial time, by applying Proposition 3.12. Taking a negation, we can also check whether (a_0, b_0) lies in the anti-projection $\overline{\operatorname{proj}}_{x,y}(f)$. This shows the problem is in NP.

The problem is also NP-hard. Indeed, let $\mathcal{L}$ be an NP language. Applying Theorem 5.5 for the case $\mathcal{L} \in$ NP, we have $\mathbf{F}(\mathcal{L}_r; t) = \operatorname{proj}_x(\overline{\operatorname{proj}}_{x,y}(f_r))$, where f_r is supported inside a box B_r. By Remark 5.6, we can compute f_r and B_r in polynomial time. So checking $x \in \mathcal{L}_r$ is equivalent to checking $x \in \operatorname{supp}(h_r)$, where $h_r = \operatorname{proj}_x(B_r \backslash \operatorname{proj}_{x,y}(f_r))$. □

REMARK 5.8. Compared to Proposition 3.12, we see that it is no longer easy to check for membership after taking two separate projections on a short GF.

6. A hierarchy of generating functions

We introduce a hierarchy GH of languages expressible as projections of generating functions. First, we define the lowest level $\mathsf{G} = \boldsymbol{\Sigma}_0^{\mathsf{G}} = \boldsymbol{\Pi}_0^{\mathsf{G}}$.

DEFINITION 6.1. For a language $\mathcal{L} \in \{0,1\}^*$, we say that $\mathcal{L} \in \mathsf{G}$ if there is an $s > 0$ so that for every $r > 0$, we can represent $\mathbf{F}(\mathcal{L}_r; t) = f_r(t)$ where $f_r \in \mathcal{GF}_{1,s}$ and $\ell(f_r) \leqslant \operatorname{poly}_{\mathcal{L}}(r)$. In other words, every segment $\mathcal{L}_r$ can be represented as a short GF of polynomial length in some fixed class $\mathcal{GF}_{1,s}$.

We define higher classes $\boldsymbol{\Sigma}_k^{\mathsf{G}}$ and $\boldsymbol{\Pi}_k^{\mathsf{G}}$ by taking repeated projections/anti-projections.

DEFINITION 6.2. For a language $\mathcal{L} \in \{0,1\}^*$, we say that $\mathcal{L} \in \boldsymbol{\Sigma}_k^{\mathsf{G}}$ if there is an $s > 0$ so that for every $r > 0$, we can represent:

$$\mathbf{F}(\mathcal{L}_r; t) = \operatorname{proj}_x\left(\overline{\operatorname{proj}}_{x,y_1}(\operatorname{proj}_{x,y_1,y_2}(\cdots(f_r)\cdots))\right), \tag{6.1}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

where $f_r(t, u_1, \ldots, u_k) = \sum t^x u_1^{y_1} \ldots u_k^{y_k} \in \mathcal{GF}_{k+1,s}$ is supported inside a finite box B_r, with both $\ell(B_r), \ell(f_r) \leqslant \text{poly}_{\mathcal{L}}(r)$. The k alternating projections/anti-projections are taken within B_r. The class $\boldsymbol{\Pi}_k^{\mathsf{G}}$ is defined similarly, with the projections/anti-projections in (6.1) reversed. Alternatively, $\mathcal{L} \in \boldsymbol{\Pi}_k^{\mathsf{G}}$ if and only if the complement language $\neg\mathcal{L}$ is in $\boldsymbol{\Sigma}_k^{\mathsf{G}}$.

DEFINITION 6.3. GH is the union of all $\boldsymbol{\Sigma}_k^{\mathsf{G}}$ and $\boldsymbol{\Pi}_k^{\mathsf{G}}$ for all $k \geqslant 0$.

We list some properties of GH:

- $\boldsymbol{\Sigma}_k^{\mathsf{G}}, \boldsymbol{\Pi}_k^{\mathsf{G}} \subseteq \boldsymbol{\Sigma}_{k+1}^{\mathsf{G}} \cap \boldsymbol{\Pi}_{k+1}^{\mathsf{G}}$ for all $k \geqslant 0$;
- $\mathsf{G}, \boldsymbol{\Sigma}_1^{\mathsf{G}}, \boldsymbol{\Pi}_1^{\mathsf{G}} \subseteq \mathsf{P/poly}$ (Propositions 3.10 and 3.12);
- $\mathsf{P/poly} \subseteq \mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}}$, the subclass of $\boldsymbol{\Sigma}_2^{\mathsf{G}}$ with only spec_x and $\overline{\text{proj}}_{x,y}$ (Theorem 4.5);
- in fact, $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}} = \mathsf{UP/poly}$ (Remark 4.8);
- $\boldsymbol{\Sigma}_k^{\mathsf{P}}/\mathsf{poly} \subseteq \boldsymbol{\Sigma}_{k+1}^{\mathsf{G}}, \boldsymbol{\Pi}_k^{\mathsf{P}}/\mathsf{poly} \subseteq \boldsymbol{\Pi}_{k+1}^{\mathsf{G}}$ for all $k \geqslant 1$ (Theorem 5.5).

The last property can actually be strengthened to:

THEOREM 6.4. *$\boldsymbol{\Sigma}_k^{\mathsf{P}}/\mathsf{poly} = \boldsymbol{\Sigma}_{k+1}^{\mathsf{G}}$ and $\boldsymbol{\Pi}_k^{\mathsf{P}}/\mathsf{poly} = \boldsymbol{\Pi}_{k+1}^{\mathsf{G}}$ for every $k \geqslant 1$. So $\mathsf{GH} = \mathsf{PH/poly}$, that is, GH is exactly the nonuniform version of PH.*

Proof. Theorem 5.5 already showed inclusion in one direction. For the other direction, assume $\mathcal{L} \in \boldsymbol{\Sigma}_{k+1}^{\mathsf{G}}$. From Definition 6.2, for every $r > 0$, we have:

$$\mathbf{F}(\mathcal{L}_r; t) = \text{proj}_x\left(\overline{\text{proj}}_{x,y_1}(\text{proj}_{x,y_1,y_2}(\cdots(f_r)\cdots))\right),$$

where f_r is a short GF of length $\text{poly}_{\mathcal{L}}(r)$ in some fixed class $\mathcal{GF}_{k+2,s}$. Here we are taking $k+1$ alternating projections and anti-projections on $f_r(x, y_1, \ldots, y_{k+1}) = \sum t^x u_1^{y_1} \ldots u_{k+1}^{y_{k+1}}$ within some finite box B_r. Note that by Proposition 3.12, we can check in polynomial time if $(x, y_1, \ldots, y_k)$ lies in the inner most projection/anti-projection. So given f_r as an advice string, we can decide if $x \in \mathcal{L}_r$ by calling a $\boldsymbol{\Sigma}_k^{\mathsf{P}}$ oracle for the remaining k projections/anti-projections. This implies $\mathcal{L} \in \boldsymbol{\Sigma}_k^{\mathsf{P}}/\mathsf{poly}$. The case $\mathcal{L} \in \boldsymbol{\Pi}_{k+1}^{\mathsf{G}}$ is similar. □

7. Short GFs have long projections

7.1. Proof of Theorem 1.5.

THEOREM 7.1. *If $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, then $\mathsf{G} \subsetneq \mathsf{P/poly}$.*

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Proof. We saw in Section 6 that $\mathsf{G} \subseteq \mathsf{P/poly}$. Now we show $\mathsf{P/poly}$ is strictly larger than G. Let $\#\mathcal{L}$ be an $\#\mathsf{P}$-complete problem (for example, #3SAT), which is outside of $\mathsf{FP/poly}$ by the assumption $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$. Associated to $\#\mathcal{L}$ is a polynomial time Turing machine M. Given $\widetilde{x} \in \{0,1\}^r$, $\#\mathcal{L}$ asks for the number of certificates $\widetilde{c} \in \{0,1\}^r$ that satisfy $M(\widetilde{x},\widetilde{c}) = 1$. Define a language:

$$\mathcal{M} = \{(\widetilde{x},\widetilde{c}) : \operatorname{length}(\widetilde{x}) = \operatorname{length}(\widetilde{c}) \text{ and } M(\widetilde{x},\widetilde{c}) = 1\}. \tag{7.1}$$

(In general, the instance $\widetilde{x}$ and certificate $\widetilde{c}$ can have different lengths. However, the Turing Machine M can always be modified to accept only $\widetilde{c}$ and $\widetilde{x}$ of equal lengths.)
Since M runs in polynomial time, we also have $\mathcal{M} \in \mathsf{P/poly}$. We show that $\mathcal{M} \notin \mathsf{G}$.

Assume the contrary, that is, $\mathcal{M} \in \mathsf{G}$. Then there is a fixed s so that for every $r > 0$, we have $\mathcal{M}_r = \operatorname{supp}(f_r)$, where $f_r \in \mathcal{GF}_{1,s}$ and $\ell(f_r) \leqslant \operatorname{poly}(r)$. Let x, $c \in [0, 2^r)$ be the integers corresponding to $\widetilde{x}, \widetilde{c} \in \{0,1\}^r$. Then the concatenated string $(\widetilde{x},\widetilde{c})$ corresponds to $x + 2^r c$. We assumed that there is an $f_{2r} \in \mathcal{GF}_{1,s}$ such that

$$\ell(f_{2r}) \leqslant \operatorname{poly}(r) \quad \text{and} \quad \sum_{(\widetilde{x},\widetilde{c}) \in \mathcal{M}_{2r}} t^{x+2^r c} = f_{2r}(t).$$

Given $\widetilde{x} \in \{0,1\}^r$, we must compute the number of $\widetilde{c} \in \{0,1\}^r$ which satisfy $(\widetilde{x},\widetilde{c}) \in \mathcal{M}_{2r}$. Define

$$g_x(t) = \sum_{0 \leqslant c < 2^r} t^{x+2^r c} = t^x \frac{1 - t^{2^{2r}}}{1 - t^{2^r}}. \tag{7.2}$$

We have $\ell(g_x) \leqslant \operatorname{poly}(r)$. We also have $f_{2r} \in \mathcal{GF}_{1,s}$ and $g_x \in \mathcal{GF}_{1,1}$. Therefore, by Theorem 3.8, the short GF $h_x = f_{2r} \star g_x$ can be computed in time $\operatorname{poly}(\ell(f_{2r}) + \ell(g_x)) \leqslant \operatorname{poly}(r)$. The number of certificates $\widetilde{c}$ for $\widetilde{x}$ is simply $h_x(1)$. This substitution can be computed in time $\operatorname{poly}(r)$ by Theorem 3.7.

To summarize, the short GF f_{2r} gives us a polynomial size circuit to solve $\#\mathcal{L}$ for all inputs $\widetilde{x} \in \{0,1\}^r$ in time $\operatorname{poly}(r)$. We conclude that $\#\mathcal{L} \in \mathsf{FP/poly}$, a contradiction. □

Now we can formulate Theorem 1.5 in precise terms:

Corollary 7.2. *If* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, *then* GH *does not collapse to its* 0*th level* G. *In other words, there is a sequence* $\{f_r\}_{r>0}$ *in some fixed class* $\mathcal{GF}_{2,s}$ *with* $\ell(f_r) \leqslant \operatorname{poly}(r)$ *so that for every* d, $\operatorname{proj}_x(f_r)$ *cannot be written as a short GF* $h_r \in \mathcal{GF}_{1,d}$ *with* $\ell(h_r) \leqslant \operatorname{poly}(r)$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Proof. Recall that $\mathsf{G} \subseteq \mathsf{P/poly} \subseteq \mathsf{GH}$ (Section 6). Now this follows from Theorem 7.1. □

7.2. A partial converse. One can ask if the above argument in the proof above can be reversed, that is, if $\#\mathsf{P} \subseteq \mathsf{FP/poly}$, does it imply that GH collapses to G? We present below a weaker result.

Recall from Section 6 that $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}}$ the subclass of $\boldsymbol{\Sigma}_2^{\mathsf{G}}$ that uses only spec_x and $\overline{\text{proj}}_{x,y}$. In other words, $\mathcal{L} \in \mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}}$ if for every $r > 0$, we have $\mathbf{F}(\mathcal{L}_r; t) = \text{spec}_x(\overline{\text{proj}}_{x,y}(f_r))$ for some f_r in some fixed class $\mathcal{GF}_{3,s}$ with $\ell(f_r) \leqslant \text{poly}_{\mathcal{L}}(r)$. We also know that $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}} = \mathsf{UP/poly}$.

PROPOSITION 7.3. *If* $\#\mathsf{P} \subseteq \mathsf{FP/poly}$, *then* GH *collapses to* $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}}$.

Proof. Since $\mathsf{GH} = \mathsf{PH/poly}$ and $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}} = \mathsf{UP/poly}$, it is equivalent to show $\mathsf{PH/poly} = \mathsf{UP/poly}$. In fact, we have a stronger collapse, namely $\mathsf{PH/poly} = \mathsf{P/poly}$. This follows easily from *Toda's theorem* (see for example, [**AB**, Section 9.3]). Indeed, by Toda's theorem , we have $\mathsf{PH} \subseteq \mathsf{P}^{\#SAT}$. Replacing the #SAT oracle by polynomial size circuits, we have $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{P/poly}} = \mathsf{P/poly}$. Taking the nonuniform version of PH, we still have $\mathsf{PH/poly} \subseteq \mathsf{P/poly}$. □

REMARK 7.4. The proposition implies that proving GH does not collapse to between its 1st and 2nd levels is at least as hard as showing $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$. However, there might still be hope of showing that GH does not collapse to its 0th level G, for example, by proving Conjecture 1.1.

REMARK 7.5. We do not claim that Proposition 7.3 is a new collapse result assuming $\#\mathsf{P} \subseteq \mathsf{FP/poly}$. Here we are only putting things in the context of short GFs. Observe that $\#\mathsf{P} \subseteq \mathsf{FP/poly}$ implies $\mathsf{NP} \subseteq \mathsf{P/poly}$. In turn, $\mathsf{NP} \subseteq \mathsf{P/poly}$ implies $\mathsf{PH} = \mathsf{S}_2^{\mathsf{P}}$ (see [**Cai**]), which is the strongest collapse currently known, assuming $\mathsf{NP} \subseteq \mathsf{P/poly}$. Note that the classical *Karp–Lipton theorem* (see for example, [**AB**, **MM**, **Pap**]), says that $\mathsf{NP} \subseteq \mathsf{P/poly}$ implies $\mathsf{PH} = \boldsymbol{\Sigma}_2^{\mathsf{P}}$, which is weaker because $\mathsf{S}_2^{\mathsf{P}} \subseteq \boldsymbol{\Sigma}_2^{\mathsf{P}} \cap \boldsymbol{\Pi}_2^{\mathsf{P}}$.

8. Intersections, unions and Minkowski sums of short GFs

8.1. Proof of Theorem 1.3. Below is the precise statement of Theorem 1.3.

THEOREM 8.1. *Assume* $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$. *Then there is an* $s > 0$ *and a family of finite subsets* $\{S_r\}_{r>0}$ *with each* $S_r = \{p_{r,1}, \ldots, p_{r,k_r}\} \subset \mathcal{GF}_{1,s}$ *so that the following hold:*

(a) *the total length of all* $p_{r,i}$ *in* S_r *is* $\mathrm{poly}(r)$;

(b) *for every fixed* d, *the intersection/union of all* $p_{r,i}$ *in* S_r *cannot be written as a short GF* $h_r \in \mathcal{GF}_{2,d}$ *with* $\ell(h_r) \leqslant \mathrm{poly}(r)$.

Proof. By Theorem 7.1, there exists a language $\mathcal{L} \in \mathsf{P/poly}$ which is outside of G. By Theorem 4.5, for every $r > 0$, we can represent:

$$\mathbf{F}(\mathcal{L}_r; t) = \mathrm{spec}_x(B_r \backslash \mathrm{proj}_{x,y}(f_r)) \quad \text{and} \quad \mathrm{proj}_{x,y}(f_r) = p_{r,1} \cup \cdots \cup p_{r,k_r},$$

where $f_r \in \mathcal{GF}_{5,5}$, $p_{r,i} \in \mathcal{GF}_{2,s}$ and $\ell(B_r)$, $\ell(f_r)$, $\sum \ell(p_{r,i}) \leqslant \mathrm{poly}(r)$. Here s is some universal constant.

Let $S_r = \{p_{r,1}, \ldots, p_{r,k_r}\}$. This family $\{S_r\}$ satisfies condition (a). We show that the union of $p_{r,i}$ cannot be written as a short GF of length $\mathrm{poly}(r)$. Indeed, assume there is d for which we can write $\mathrm{proj}_{x,y}(f_r) = p_{r,1} \cup \cdots \cup p_{r,k_r}$ as $h_r \in \mathcal{GF}_{2,d}$ with $\ell(h_r) \leqslant \mathrm{poly}(r)$. By Theorem 3.8, the complement $B_r \backslash h_r$ can be written as a short GF $g_r \in \mathcal{GF}_{2,2d}$ of length $\mathrm{poly}(r)$. Taking the specialization $\mathrm{spec}_x(g_r)$, we still have a short GF in $\mathcal{GF}_{2,2d}$ of length $\mathrm{poly}(r)$, which represents $\mathcal{L}_r$. Since this holds for all $r > 0$, we have $\mathcal{L} \in \mathsf{G}$, a contradiction. So the family $\{S_r\}$ also satisfies (b).

Note that each $p_{r,i}$ still has two variables x, y. By Lemma 4.10 part (a), we can compress each $p_{r,i}$ into a single-variable short GF $\widetilde{p}_{r,i} \in \mathcal{GF}_{1,s}$ of polynomial length. Then the new subsets $\widetilde{S}_r = \{\widetilde{p}_{r,1}, \ldots, \widetilde{p}_{r,k_r}\} \subset \mathcal{GF}_{1,s}$ still satisfy condition (a). We show they still satisfy condition (b). Indeed, note that compressing/decompression preserves intersection and union. So if $\widetilde{p}_{r,i}$ has a polynomial length union then Lemma 4.10 part (b) allows us the decompress it into a polynomial length union of $p_{r,i}$. This completes the proof for the case of union. The case of intersection follows by taking complements of $p_{r,i}$. □

8.2. Proof of Theorem 1.4.

DEFINITION 8.2. Given two GFs $a = \mathbf{F}(S_1; \mathbf{t})$ and $b = \mathbf{F}(S_2; \mathbf{t})$ with $S_1, S_2 \subseteq \mathbb{N}^n$, the *Minkowski sum* $a \oplus b$ is $\mathbf{F}(S_1 \oplus S_2; \mathbf{t})$, where $S_1 \oplus S_2$ is the usual Minkowski sum of two point sets.

EXAMPLE 8.3. Given $\overline{b} = (b_1, \ldots, b_n) \in \mathbb{N}^n$, the semigroup $\mathbb{N}\langle b_1, \ldots, b_n\rangle$ consists of all nonnegative integer combinations of the b_j's. Its generating function is given by:

$$f_{\overline{b}}(t) = \frac{1}{1 - t^{b_1}} \oplus \cdots \oplus \frac{1}{1 - t^{b_n}}.$$

Given such $\overline{b} \in \mathbb{N}^n$ and $a \in \mathbb{N}$, the KNAPSACK problem asks if $a \in \mathrm{supp}(f_{\overline{b}})$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Below is the precise statement of Theorem 1.4.

THEOREM 8.4. *Assume* #P $\not\subseteq$ FP/poly. *Then there is an* $s > 0$ *and two sequences* $\{a_r\}_{r>0}$, $\{b_r\}_{r>0} \subset \mathcal{GF}_{1,s}$ *such that:*

(a) $\ell(a_r) + \ell(b_r) \leqslant \text{poly}(r)$;

(b) *for every fixed* d, *the Minkowski sum* $a_r \oplus b_r$ *cannot be written as a short GF* h_r *in* $\mathcal{GF}_{1,d}$ *of length* $\ell(h_r) \leqslant \text{poly}(r)$.

Proof. By Theorem 8.1, there exists an $s > 0$, and for each r a subset

$$S_r = \{p_{r,1}, \ldots, p_{r,k_r}\} \subset \mathcal{GF}_{1,s} \quad \text{with} \sum \ell(p_{r,i}) \leqslant \text{poly}(r)$$

with the following property. For every fixed d, the union $h_r = p_{r,i} \cup \cdots \cup p_{r,k_r}$ cannot be written as a short GF of length poly(r) in $\mathcal{GF}_{1,d}$. Define

$$a_r(t, u) = \sum_{i=1}^{k_r} p_{r,i}(t) u^i \in \mathcal{GF}_{2,s} \tag{8.1}$$

and

$$b_r(t, u) = \sum_{i=0}^{k_r-1} t^0 u^i = \frac{1 - u^{k_r}}{1 - u} \in \mathcal{GF}_{1,1} \subset \mathcal{GF}_{2,s}. \tag{8.2}$$

Since $\sum \ell(p_{r,i}) \leqslant \text{poly}(r)$, we also have $\ell(a_r) + \ell(b_r) \leqslant \text{poly}(r)$.

Consider the terms $t^x u^{k_r}$ in the Minkowski sum $a_r \oplus b_r$. From (8.1) and (8.2), we have:

$$\{x : (x, k_r) \in \text{supp}(a_r \oplus b_r)\} = \bigcup_{i=1}^{k_r} \text{supp}(p_{r,i}) = \text{supp}(h_r).$$

In other words, we have $[u^{k_r}](a_r \oplus b_r)(t, u) = h_r(t)$. Define

$$g_r(t, u) = \sum_{x \in \mathbb{N}} t^x u^{k_r} = \frac{u^{k_r}}{1 - t}.$$

Taking the intersection of g_r with $a_r \oplus b_r$, we get:

$$[(a_r \oplus b_r) \star g_r](t, u) = u^{k_r} h_r(t). \tag{8.3}$$

Now assume there is d so that $a_r \oplus b_r$ can be written as $c_r \in \mathcal{GF}_{2,d}$ with $\ell(c_r) \leqslant \text{poly}(r)$. By Theorem 3.8, we can compute h_r by taking the Hadamard

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

product $c_r \star g_r$ and substitute $u \leftarrow 1$ in (8.3). This would imply that h_r is a short GF of length $\operatorname{poly}(r)$ in the fixed class $\mathcal{GF}_{1,d+1}$, which contradicts our first statement on h_r.

So the two sequences $\{a_r\}_{r>0}$ and $\{b_r\}_{r>0} \subset \mathcal{GF}_{2,s}$ do not have Minkowski sums of polynomial lengths. Note that each a_r and b_r still has two variables. By Lemma 4.10 part (a), we can compress a_r, b_r into single-variable short GFs $\widetilde{a}_r$, $\widetilde{b}_r \in \mathcal{GF}_{1,s}$. Note that compressing/decompression preserves Minkowski sum. So $\widetilde{a}_r \oplus \widetilde{b}_r$ does not have polynomial length, because otherwise we can decompress it to get $a_r \oplus b_r$ of polynomial length. □

9. Squares, primes, and short GFs

9.1. Short GFs and squares. Recall the definition of the class G from Section 6. We present a candidate for a language $\mathcal{L} \in \mathsf{P/poly}$ which is outside of G. Let SQUARES be the language consisting of all square numbers written in binary. Then

$$\text{SQUARES}_r = \{k^2 : k^2 < 2^r\}. \tag{9.1}$$

(Strictly speaking, some numbers in SQUARES_r have less than r digits. However, we can always pad them with enough zeros form a set of strings of the same length.)

CONJECTURE 9.1. SQUARES *is not in* G.

In other words, the conjecture says that for every fixed s, the segment SQUARES_r cannot be represented as $\operatorname{supp}(g_r)$ for a short GF $g_r \in \mathcal{GF}_{1,s}$ of length $\ell(g_r) \leqslant \operatorname{poly}(r)$. Note that this conjecture is free of complexity assumptions. If true, Conjecture 9.1 shows unconditionally that $\mathsf{G} \subsetneq \mathsf{P/poly}$, which implies $\mathsf{G} \subsetneq \mathsf{GH}$. We already know from Example 4.6 and Section 6 that $\text{SQUARES} \in \mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}} \subseteq \mathsf{GH}$. So SQUARES should be a candidate that separates G from $\mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}}$ according to this conjecture.

We begin with the following attractive result.

THEOREM 9.2. *If Conjecture* 9.1 *is false, then* INTEGER FACTORING $\in$ BPP.

Proof. We build on an argument in [B2, Section 6]. Assume there is an $s > 0$ so that for every $N = 2^r$, we can write $\mathbf{F}(\text{SQUARES}_r; t) = g_r(t)$, where $g_r(t)$ is a

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

short GF in $\mathcal{GF}_{1,s}$ with $\ell(g) \leqslant \operatorname{poly}(r)$. Consider:

$$h_r(t) = g_r(t)^4 = \left(\sum_{n^2<N} t^{n^2}\right)^4 = \sum_{k\geqslant 0} a_r(k)t^k,$$

where

$$a_r(k) = \#\left\{(n_1, n_2, n_3, n_4) : n_i^2 < N, \sum n_i^2 = k\right\}.$$

In particular, if $k < N$, then $a_r(k)$ is the number of ways to write k as a sum of four squares. Since $g_r \in \mathcal{GF}_{1,s}$, we have $h_r = g_r^4 \in \mathcal{GF}_{1,4s}$ and also $\ell(h) \leqslant \operatorname{poly}(\ell(g)) \leqslant \operatorname{poly}(r)$.

Applying Proposition 3.10, each coefficient $a_r(k)$ can be computed in time $\operatorname{poly}(r)$. By Jacobi's formula (see for example, [**HW**]), we also have:

$$a_r(k) = 8 \sum_{4\nmid d, d|k} d \quad \text{for } k < N.$$

Here d is a divisor of k which is not a multiple of 4. From this, we can compute in time $\operatorname{poly}(r)$ the sum of divisors $\sigma(k)$ for every $k < N = 2^r$. By a standard argument (see for example, [**BMS**]), given $\sigma(k)$, a factorization of k can be computed in probabilistic polynomial time (BPP). □

THEOREM 9.3. *If Conjecture 9.1 is false, then* #P $\subseteq$ FP/poly.

Proof of Theorem 9.3. In [**MA**], it is proved that the following problem is NP-complete: given $\alpha, \beta, \gamma \in \mathbb{N}$, decide whether there exists $x \in \mathbb{N}$ such that

$$0 \leqslant x \leqslant \gamma \quad \text{and} \quad x^2 \equiv \alpha (\text{mod } \beta). \tag{9.2}$$

The argument in [**MA**] actually gave bijection between the set of Boolean strings satisfying a 3SAT formula and the set of x satisfying (9.2). Here α, β and γ can be computed in polynomial time from the 3SAT formula. Since counting the number of 3SAT solutions is #P-complete, so is counting the number of solutions for (9.2).

Now assume Conjecture 9.1 fails, then SQUARES $\in$ G. This means there is an $s > 0$ so that for every $r > 0$ we can write $\mathbf{F}(\text{SQUARES}_r; t) = g_r(t)$ for some $g_r \in \mathcal{GF}_{1,s}$ with $\ell(g_r) \leqslant \operatorname{poly}(r)$. Given $\alpha, \beta, \gamma \in \mathbb{N}$, we define:

$$h(t) = \sum_{i=0}^{\gamma^2} t^i = \frac{1 - t^{\gamma^2+1}}{1-t} \quad \text{and} \quad k(t) = \sum_{x\equiv\alpha(\text{mod } \beta)} t^x = \frac{t^\alpha}{1-t^\beta}.$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Let $r = 2\lceil \log \gamma \rceil$. The number of solutions for (9.2) can be counted by taking $g_r \star h \star k$ and evaluate at $t = 1$, which are polynomial time operations by theorems 3.7 and 3.8. So the above #P-complete problem can be solved by polynomial size circuits, which are provided by the g_r for different r. This implies #P $\subseteq$ FP/poly. □

By Theorem 4.5, we can represent SQUARES_r as $\text{spec}_x(B_r \backslash \text{proj}_y(f_r))$ for some short GF f_r of length $\text{poly}(r)$. Conjecture 9.1 says that it is not possible to do so without using projections. In the domain of PA formulas, by Lemma 4.2, we can represent SQUARES_r with a $\exists\forall$-formula of length $\text{poly}(r)$. A similar question can be asked, that is, are quantifiers necessary? The following result shows that two quantifiers $\exists\forall$ are necessary in Lemma 4.2, already in the case of SQUARES.

PROPOSITION 9.4. SQUARES_r *cannot be represented by an* $\exists$*-formula of length* $\text{poly}(r)$ *in a fixed number of variables.*

Proof. By AP_k we mean a k-term arithmetic progression. It is well known that SQUARES does not contain any nontrivial AP_4. This was suggested by Fermat in 1640 and proved by Euler in 1780 (see for example, [**Wei**, page 115]). Also, the cardinality of SQUARES_r is super-polynomial in r. With these two observations, this proposition follows directly from the next theorem when $k = 4$. □

THEOREM 9.5. *For every fixed n and k, there exists a polynomial P so that the following holds. If an* $\exists$*-formula*

$$\{x : \exists \mathbf{y} \in \mathbb{Z}^n \Phi(x, \mathbf{y})\} \tag{9.3}$$

determines a set of cardinality at least $P(\ell(\Phi))$*, then it must contain a nontrivial* AP_k.

Proof. By Proposition 3.17, we know that there is a constant $c = c(n) > 0$ so that any quantifier free expression Φ in n variables describes a disjoint union of m polyhedra $P_1, \ldots, P_m \subseteq \mathbb{R}^{n+1}$ with $m < \ell(\Phi)^c$. So formula (9.3) can be rewritten as:

$$S = \left\{ x \in \mathbb{Z} : \exists \mathbf{y} \in \mathbb{Z}^n \bigvee_{i=1}^{m} (x, \mathbf{y}) \in P_i \right\}. \tag{9.4}$$

Let $\text{q}(t) = k^{n+1} t^c$. Assume that $|S| \geqslant \text{q}(\ell(\Phi)) > k^{n+1} m$. Select any $(k^{n+1} m + 1)$ different integers from S. By the pigeonhole principle, one of the polyhedra, say P_1, contains in its projection at least $k^{n+1} + 1$ of these integers. Denote those

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

integers in the projection of P_1 by $x_1, \ldots, x_s$, where $s = k^{n+1} + 1$. For every such x_i, there exists $\mathbf{y}_i \in \mathbb{Z}^n$ so that $(x_i, \mathbf{y}_i) \in P_1$. So we have:

$$(x_1, \mathbf{y}_1), \ldots, (x_s, \mathbf{y}_s) \in P_1 \cap \mathbb{Z}^{n+1}.$$

By the pigeonhole principle, two different pairs $(x_i, \mathbf{y}_i)$ and $(x_j, \mathbf{y}_j)$ have coordinates equal mod k pairwise. Since P_1 is convex, we also have

$$(\lambda x_i + (1-\lambda)x_j,\ \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j) \in P_1 \cap \mathbb{Z}^{n+1} \quad \text{where } \lambda \in \left\{\frac{1}{k}, \ldots, \frac{k-1}{k}\right\}.$$

The above points project to $\lambda x_i + (1-\lambda)x_j$. By (9.4), we get a nontrivial AP_{k+1}:

$$\left(x_i, \frac{k-1}{k}x_i + \frac{1}{k}x_j, \ldots, \frac{1}{k}x_j + \frac{k-1}{k}x_i, x_j\right),$$

a contradiction. □

REMARK 9.6. Proposition 9.4 combined with Lemma 4.2 implies that there is a sequence of formulas $\{x : \exists y\ \forall \mathbf{z}\ \Phi_r(x, y, \mathbf{z})\}$ of length $\mathrm{poly}(r)$ for which there are no equivalent formulas $\{x : \exists y\ \Psi_r(x, y)\}$ of length $\mathrm{poly}(r)$. This implies that the formulas $\{(x, y) : \forall \mathbf{z}\ \Phi_r(x, y, \mathbf{z})\}$ have no equivalent quantifier free formulas in x and y of length $\mathrm{poly}(r)$. Therefore, quantifier elimination in PA necessarily increases the length of formulas by a super-polynomial factor, even in a bounded number of variables ($x, y \in \mathbb{N}, \mathbf{z} \in \mathbb{N}^3$).

REMARK 9.7. From SQUARES, one can easily create another language $\mathcal{L} \in \mathsf{P}$ which $\mathcal{L}_r$ be represented neither by $\forall$ nor by $\exists$ formulas of length $\mathrm{poly}(r)$. For r odd, we let $\mathcal{L}$ contain all squares between 2^r and 2^{r+1}. For r even, we let $\mathcal{L}$ contain all nonsquares between 2^r and 2^{r+1}. It is clear that $\mathcal{L} \in \mathsf{P}$. The above argument shows that $\mathcal{L}_r$ cannot be represented by $\exists$-formulas of length $\mathrm{poly}(r)$ when r is odd. Under a negation, the same argument also works for $\forall$-formulas when r is even. We denote this language by SQUARES′. This will be used in Section 10.

9.2. Short GFs and arithmetic progressions.

Generalizing the above observation on sets with no arithmetic progressions, we suggest another conjecture on short GFs. Again, by AP_k we mean a *k-term arithmetic progression*.

DEFINITION 9.8. Fix $c > 0$ and $k \geqslant 3$. A short GF g is said to have the (c, k)-property if either $|\mathrm{supp}(g)| < \ell(g)^c$ or $\mathrm{supp}(g)$ contains an AP_k.

CONJECTURE 9.9. *For every s and k, there exists $c > 0$ so that every short GF $g(t) \in \mathcal{GF}_{1,s}$ has the (c, k)-property.*

PROPOSITION 9.10. *Conjecture 9.9 implies Conjecture 9.1.*

Proof. Assume Conjecture 9.9 holds but Conjecture 9.1 fails, that is, SQUARES $\in \mathsf{G}$. So there is an $s > 0$ such that SQUARES_r can be represented as $\operatorname{supp}(g_r)$ with $g_r \in \mathcal{GF}_{1,s}$ and $\ell(g_r) \leqslant \operatorname{poly}(r)$. Conjecture 9.9 applied to s and $k = 4$ gives us a $c > 0$ so that all $g \in \mathcal{GF}_{1,s}$ have the $(c, 4)$-property. We have $\operatorname{supp}(g_r) = |\text{SQUARES}_r| \gg r^c$. So if r is large enough, g_r contains an AP_4. This contradicts the fact that SQUARES is AP_4 free. □

9.3. Short GFs and primes. In a similar manner, we ask if primes can be represented by short GFs of polynomial length. Let PRIMES be the language consisting of all primes written in binary. Then

$$\text{PRIMES}_r = \{p \text{ prime} : p < 2^r\}. \tag{9.5}$$

CONJECTURE 9.11. PRIMES *is not in* G.

In other words, the conjecture says that for every fixed s, the segment PRIMES_r cannot be represented as $\operatorname{supp}(g_r)$ for a short GF $g_r \in \mathcal{GF}_{1,s}$ of length $\ell(g_r) \leqslant \operatorname{poly}(r)$. This conjecture, if true, would also show $\mathsf{G} \subsetneq \mathsf{P/poly}$ unconditionally.

PROPOSITION 9.12. *Let $\pi(n)$ be the number of primes between 1 and n. If Conjecture 9.11 is false then $\pi(n)$ can be computed by circuits of size* $\operatorname{poly}(\log n)$.

Proof. Assume Conjecture 9.11 is false, that is, there is an $s > 0$ so that for every $r > 0$ we have $\mathbf{F}(\text{PRIMES}_r; t) = g_r(t)$, where $g_r \in \mathcal{GF}_{1,s}$ and $\ell(g_r) \leqslant \operatorname{poly}(r)$. Given $n < 2^r$, we have:

$$\mathbf{F}(\text{PRIMES}_r \cap [0, n]; t) = g_r(t) \star \frac{1 - t^{n+1}}{1 - t} = h_n(t).$$

By Theorem 3.8, we can compute h_n in time $\operatorname{poly}(r)$. Substituting $t \leftarrow 1$, we get $\pi(n)$. □

REMARK 9.13. In [LO], using strong analytic tools, Lagarias and Odlyzko gave an algorithm to compute $\pi(n)$ in time $O(n^{1/2+\epsilon})$, which is exponential in $\log n$. If Conjecture 9.11 is false, then for each r, a far better $\operatorname{poly}(r)$ algorithm exists for computing $\pi(n)$ for all $n < 2^r$.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

10. Relative complexity of short GFs

In this section, we compare short GFs with PA formulas with one quantifier. We refer back to Section 3.2 for the definition of PA formulas.

10.1. PA complexity classes. The most basic PA formulas contain no quantifiers, that is, only a Boolean combination of inequalities.

DEFINITION 10.1. The class $\boldsymbol{\Sigma}_0^{\mathsf{PA}} = \boldsymbol{\Pi}_0^{\mathsf{PA}}$ consists of languages definable by quantifier free PA formulas of polynomial lengths. In other words, a language $\mathcal{L}$ is in $\boldsymbol{\Sigma}_0^{\mathsf{PA}}$ if for every $r > 0$, there is a quantifier free PA expression $\Phi_r(x)$ of length $\ell(\Phi) \leqslant \text{poly}_{\mathcal{L}}(r)$ so that:

$$x \in \mathcal{L}_r \quad \Longleftrightarrow \quad \Phi_r(x).$$

By Proposition 3.17, $\mathcal{L} \in \boldsymbol{\Sigma}_0^{\mathsf{PA}}$ if and only if every initial segment $\mathcal{L}_r$ is a union of polynomially many intervals in $\mathbb{N}$. By Theorem 3.18, we have $\boldsymbol{\Sigma}_0^{\mathsf{PA}} \subset \mathsf{G}$.

EXAMPLE 10.2. The language EVEN of even integers is not in $\boldsymbol{\Sigma}_0^{\mathsf{PA}}$. However, EVEN $\in \mathsf{G}$, because:

$$\sum_{x \in \text{EVEN}_r} t^x = t^0 + t^2 + \cdots + t^{2^r - 2} = \frac{1 - t^{2^r}}{1 - t^2}.$$

So we conclude that $\boldsymbol{\Sigma}_0^{\mathsf{PA}} \subsetneq \mathsf{G}$.

DEFINITION 10.3. The class $\boldsymbol{\Sigma}_1^{\mathsf{PA}}$ consists of languages definable by $\exists$-formulas of polynomial lengths. In other words, $\mathcal{L} \in \boldsymbol{\Sigma}_1^{\mathsf{PA}}$ if there is an n so that for every $r > 0$, we can represent

$$x \in \mathcal{L}_r \quad \Longleftrightarrow \quad \exists \mathbf{y} \in \mathbb{N}^n \, \Phi_r(x, \mathbf{y}),$$

where $\Phi_r(x, \mathbf{y})$ is a quantifier free PA expression of length $\ell(\Phi_r) = \text{poly}_{\mathcal{L}}(r)$. The class $\boldsymbol{\Pi}_1^{\mathsf{PA}}$ is defined similarly, but with $\forall$-formulas. In other words, $\mathcal{L} \in \boldsymbol{\Pi}_1^{\mathsf{PA}}$ if and only if $\neg\mathcal{L} \in \boldsymbol{\Sigma}_1^{\mathsf{PA}}$.

CONJECTURE 10.4. $\mathsf{G} \subseteq \boldsymbol{\Sigma}_1^{\mathsf{PA}} \cap \boldsymbol{\Pi}_1^{\mathsf{PA}}$.

To rephrase, this conjecture says that for every fixed s, there is an $n = n(s)$ so that every $g \in \mathcal{GF}_{1,s}$ of finite support has an $\exists$-formula representation:

$$G = \{x : \exists \mathbf{y} \in \mathbb{N}^n \, \Phi(x, \mathbf{y})\}, \quad \mathbf{F}(G; t) = g(t) \quad \text{and} \quad \ell(\Phi) \leqslant \text{poly}(\ell(g)). \tag{10.1}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

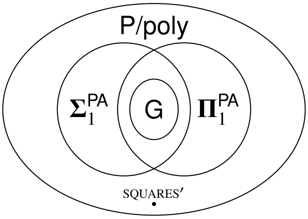


Figure 1. Short GFs versus PA formulas. Here SQUARES′ is the language defined in Remark 9.7.

Note that it would be enough to show $\mathsf{G} \subseteq \boldsymbol{\Sigma}_1^{\mathsf{PA}}$, because G is closed under taking complement of short GFs.

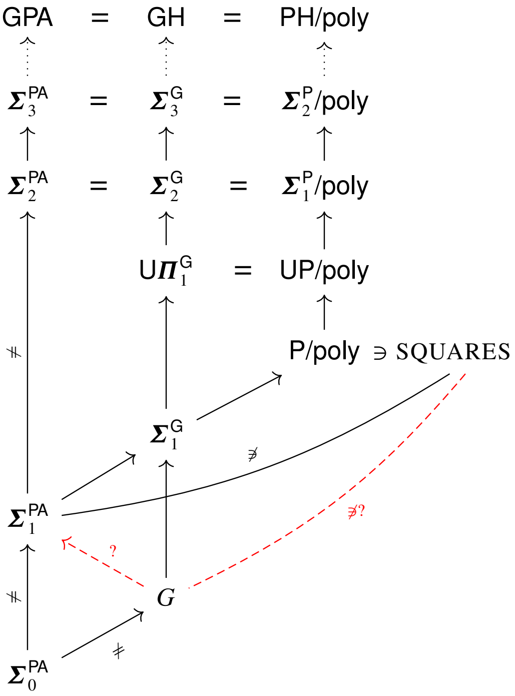
Proposition 10.5. *Conjecture 10.4 implies Conjecture 9.9, which implies Conjecture 9.1.*

Proof. Assume Conjecture 10.4 holds. Then for every fixed s, we have $n = n(s)$ for which every $g \in \mathcal{GF}_{1,s}$ has an $\exists$-formula representation (10.1). The last condition means there is a constant $d = d(s)$ such that $\ell(\Phi) < \ell(g)^d$. By Theorem 9.5, there exists $\gamma = \gamma(n, k) > 0$ so that G contains an AP$_k$ whenever $|G| > \ell(\Phi)^\gamma$. So if $|\mathrm{supp}(g)| \geqslant \ell(g)^{\gamma d}$ then $|G| = |\mathrm{supp}(g)| \geqslant \ell(g)^{\gamma d} > \ell(\Phi)^\gamma$, which implies that G contains an AP$_k$. So $c = \gamma d$ satisfies Conjecture 9.9, which should depend only on s and k. By Proposition 9.10, Conjecture 9.9 implies Conjecture 9.1. □

Figure 1 illustrates the relative relations between short GFs and PA formulas, assuming Conjecture 10.4:

One can of course define analogues of $\boldsymbol{\Sigma}_1^{\mathsf{PA}}$ and $\boldsymbol{\Pi}_1^{\mathsf{PA}}$ with more alternating quantifiers. But it turns out that $\boldsymbol{\Sigma}_{k+1}^{\mathsf{PA}} = \boldsymbol{\Sigma}_{k+1}^{\mathsf{G}} = \boldsymbol{\Sigma}_k^{\mathsf{P}}/\mathsf{poly}$ for every $k \geqslant 1$. This was implicit in Lemma 5.1 and Theorems 5.5, 6.4. For the sake of completeness, we call the hierarchy of all classes $\boldsymbol{\Sigma}_k^{\mathsf{PA}}$ and $\boldsymbol{\Pi}_k^{\mathsf{PA}}$ as GPA. Obviously $\mathsf{GPA} = \mathsf{GH} = \mathsf{PH/poly}$.

10.2. Complexity classes diagram. The following diagram summarizes various complexity classes that appeared in this paper and their relationships. An arrow $X \to Y$ indicates $X \subseteq Y$. Known strict subset relations are decorated with $\neq$. Dashed arrows and segments denote conjectural relationships.


GPA = GH = PH/poly
Σ_3^{PA} = Σ_3^{G} = Σ_2^{P}/poly
Σ_2^{PA} = Σ_2^{G} = Σ_1^{P}/poly
$\mathsf{U}\Pi_1^{\mathsf{G}}$ = UP/poly
P/poly $\ni$ SQUARES
Σ_1^{G}
Σ_1^{PA}
G
Σ_0^{PA}
$\not\leftarrow$
$\not\ni$
$\not\ni ?$
?
$\not\leftarrow$
$\not\leftarrow$

$\Sigma_{k+1}^{\mathsf{PA}} = \Sigma_{k+1}^{\mathsf{G}} = \Sigma_k^{\mathsf{P}}/\mathsf{poly}$, $k \geqslant 1$: Sections 6, 10.
$\mathsf{U}\Pi_1^{\mathsf{G}} = \mathsf{UP/poly}$: Remark 4.8.
$\Sigma_1^{\mathsf{G}} \subseteq \mathsf{P/poly}$: Proposition 3.12.

SQUARES $\overset{?}{\notin}$ G: Conjecture 9.1.
SQUARES $\notin \Sigma_1^{\mathsf{PA}}$: Proposition 9.4.
$\Sigma_0^{\mathsf{PA}} \subsetneq \Sigma_1^{\mathsf{PA}} \subsetneq \Sigma_2^{\mathsf{PA}}$: Remark 9.6.
$\Sigma_0^{\mathsf{PA}} \subsetneq \mathsf{G} \overset{?}{\subseteq} \Sigma_1^{\mathsf{PA}}$: Section 10.

11. Proof of Lemma 4.10

Let $\overline{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ be the array of multivariables of dimension $n_1, \ldots, n_k$. We first prove the result when $k = 1$, that is, when $\overline{\mathbf{x}} = \mathbf{x}_1$, $g(\overline{\mathbf{t}}) = \sum \mathbf{t}_1^{\mathbf{x}_1}$ and $f(u) = \sum u_1^{z_1}$. For convenience, we denote $\mathbf{t}_1, \mathbf{x}_1, u_1, z_1$ by $\mathbf{t}, \mathbf{x}, u$ and z, respectively. Also denote by n the dimension of the multivariable $\mathbf{x}$. So $g(\mathbf{t}) = \sum \mathbf{t}^{\mathbf{x}}$ and

$$\tau_N(\mathbf{x}) = x_1 + Nx_2 + \cdots + N^{n-1}x_n.$$

Part (a). Assume we are given $g \in \mathcal{GF}_{n,s}$. By Theorem 3.7, we can find the norm N of g in time poly$(\ell(g))$. By rounding N to the next power of 2, we still

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

have $\log N \leqslant \text{poly}(\ell(g))$ and $\text{supp}(g) \subseteq [0, N)^n$. Let $N = 2^r$. We define $f(u)$ as the specialization of $g(\mathbf{t})$ under the following substitutions:

$$t_1 \leftarrow u, \quad t_2 \leftarrow u^N, \ldots, t_n \leftarrow u^{N^{n-1}},$$

so that

$$\mathbf{t}^{\mathbf{x}} = u^{x_1 + N x_2 + \cdots + N^{n-1} x_n} = u^{\tau_N(\mathbf{x})}.$$

Clearly, we have:

$$\text{supp}(f) = \tau_N(\text{supp}(g)).$$

By Theorem 3.7, polynomial substitutions can be performed in polynomial time and gives f as a short GF in $\mathcal{GF}_{1,s}$ with $\ell(f) \leqslant \text{poly}(\ell(g))$. This proves part (a).

Part (b). Given two power series $A(\mathbf{t}) = \sum \alpha_{\mathbf{x}} \mathbf{t}^{\mathbf{x}} \in \mathcal{GF}_{n,p}$, $B(t) = \sum \beta_x t^x \in \mathcal{GF}_{1,q}$ and a linear map $\tau : \mathbb{Z}^n \to \mathbb{Z}$, we define their τ-*Hadamard product* as

$$C(\mathbf{t}) = A(\mathbf{t}) \star_\tau B(t) := \sum \alpha_{\mathbf{x}} \beta_{\tau(\mathbf{x})} \mathbf{t}^{\mathbf{x}}. \tag{11.1}$$

Now assume $f(u) = \sum u^z \in \mathcal{GF}_{1,s}$, $N = 2^r$, and $\text{supp}(f) \subseteq [0, N)^n$. From the above definition, it is clear that such a $g(\mathbf{t})$ satisfying (4.14) can be obtained as:

$$g(\mathbf{t}) = a(\mathbf{t}) \star_{\tau_N} f(t), \tag{11.2}$$

where

$$a(\mathbf{t}) = \sum_{\mathbf{x} \in [0,N)^n} \mathbf{t}^{\mathbf{x}} = \frac{1 - t_1^N}{1 - t_1} \cdots \frac{1 - t_n^N}{1 - t_n}$$

with $a \in \mathcal{GF}_{n,n}$ and $\ell(a) \leqslant \text{poly}(\log N)$.

Here the map τ_N is from Definition 4.9. So it is enough to show that the τ-Hadamard product of two short GFs is a short GF of polynomial length. The proof follows Barvinok's argument in [**B2**] (see also [**BW**, Lemmas 3.4 and 3.6]). First, notice that the τ-Hadamard product is bilinear in $A(\mathbf{t})$ and $B(t)$. Therefore, we only need to show that $C(\mathbf{t})$ is a short GF when $A(\mathbf{t})$ and $B(\mathbf{t})$ have only one term each, that is, when:

$$A(\mathbf{t}) = \frac{\mathbf{t}^{\overline{a}}}{\prod_{i=1}^p (1 - \mathbf{t}^{\overline{b}_i})} \quad \text{and} \quad B(t) = \frac{t^c}{\prod_{j=1}^q (1 - t^{d_j})}. \tag{11.3}$$

Consider an (unbounded) polyhedron $P \subset \mathbb{R}^{p+q}$ defined as:

$$P := \left\{ \begin{matrix} \zeta_1, \ldots, \zeta_p, \xi_1, \ldots, \xi_q \geqslant 0 \\ \tau(\overline{a} + \zeta_1 \overline{b}_1 + \cdots + \zeta_p \overline{b}_p) = c + \xi_1 d_1 + \cdots + \xi_q d_q \end{matrix} \right\}. \tag{11.4}$$

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

By Theorem 3.16, we can write a short GF for $P \cap \mathbb{Z}^{p+q}$:

$$D(\mathbf{w},\mathbf{v}) := \sum_{(\boldsymbol{\zeta},\boldsymbol{\xi})\in P} \mathbf{w}^{\boldsymbol{\zeta}}\mathbf{v}^{\boldsymbol{\xi}} = \sum_{(\boldsymbol{\zeta},\boldsymbol{\xi})\in P} (w_1)^{\zeta_1}\ldots(w_p)^{\zeta_p}(v_1)^{\xi_1}\ldots(v_q)^{\xi_q}. \tag{11.5}$$

Furthermore, we have $D \in \mathcal{GF}_{p+q,p+q}$. By (11.3), the expansions of $A(\mathbf{t})$ and $B(t)$ are:

$$A(\mathbf{t}) = \sum_{\boldsymbol{\zeta}\geqslant 0} \mathbf{t}^{\overline{a}+\zeta_1\overline{b}_1+\cdots+\zeta_p\overline{b}_p} \quad \text{and} \quad B(t) = \sum_{\boldsymbol{\xi}\geqslant 0} t^{c+\xi_1 d_1+\cdots+\xi_q d_q}. \tag{11.6}$$

We substitute:

$$w_1 \leftarrow \mathbf{t}^{\overline{b}_1}, \ldots, w_p \leftarrow \mathbf{t}^{\overline{b}_p}, \quad v_1 \leftarrow 1, \ldots, v_q \leftarrow 1.$$

By (11.4)–(11.6), we get:

$$\mathbf{t}^{\overline{a}} D(\mathbf{t}^{\overline{b}_1}, \ldots, \mathbf{t}^{\overline{b}_p}, 1, \ldots, 1) = \sum_{(\boldsymbol{\zeta},\boldsymbol{\xi})\in P} \mathbf{t}^{\overline{a}+\zeta_1\overline{b}_1+\cdots+\zeta_p\overline{b}_p} = A(\mathbf{t}) \star_\tau B(t) = C(\mathbf{t}).$$

By Theorem 3.7, substitution can be done in polynomial time, and results in a short GF $C(\mathbf{t})$ of index at most $p+q$. Hence, we have $C(\mathbf{t}) \in \mathcal{GF}_{n,p+q}$ and $\ell(C) \leqslant \text{poly}(\ell(A)+\ell(B))$. Note that by taking the τ-Hadamard product, the index of C is increased to $p+q$. This pushes the index of g in (11.2) to $n+s$. So we do not get back exactly the index s for g. But $n+s$ is still a constant, and g is still a short GF in a fixed class $\mathcal{GF}_{n,n+s}$.

This completes the proof for the case $k=1$. The general case can be handled similarly.

12. Final remarks and open problems

12.1. As we mentioned in the introduction, much of this work is motivated by Barvinok's program implicit in his writing. Specifically, we were inspired by the following quote:

> 'It seems hard to prove that a particular finite, but large, set $S \subset \mathbb{Z}^d$ does not admit a short rational generating function: if a particular candidate expression for $f_S(\mathbf{x})$ is not short, one can argue that we have not searched hard enough and that there is another, better candidate'. [**B2**]

In fact, this paper originally began as a followup on [**NP1**], aiming to explain why the technology of short GFs was unable to derive the Barvinok–Woods theorem (Theorem 3.19) (see [**NP1**]). Our Theorems 1.3 and 1.5 are strong versions of this claim.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

Let us also mention [NP3, Theorem 1.1 and Corollary 1.11] which have similar setup of unions and projections of polyhedra, and give strong algorithmic extensions of Woods's theorem (Theorem 3.21).

Finally, our most recent results in [NP4] say that Presburger arithmetic with a bounded number of variables and inequalities is complete for every level in PH, which suggests an even deeper obstacle to taking unions and projections. We have yet to fully explore the implications of this result which go beyond the scope of this paper.

12.2. In notations of the introduction, a short GF $f_S(t)$ of a set $S \subset \mathbb{N}$ can be viewed as a presentation of S by an alternating sum of generalized (k-dimensional) arithmetic progressions. As such, there are many connections between short GFs and Arithmetic Combinatorics, which are yet to be explored (see [TV]). For example, when $k = 1$, taking the positive part of these arithmetic progressions corresponds to variants of Erdős's *covering systems* which received much attention in recent years (see [Guy, Hou]).

Conjecture 1.1 has an especially classical feel with its claim that squares and (generalized) arithmetic progression are incompatible. There are of course both classical and recent works on squares in arithmetic progressions, but no known results seem strong enough to apply in this case (see [BGP, Sze, Wei]).

12.3. There are two ways to think of the results in this paper. First and foremost, they provide a very strong evidence in favor of nonpolynomiality of projections and other operations with short GFs. In the opposite direction, the apparent connection to arithmetic progressions and a plethora of both analytic and combinatorial tools for working with them suggest a possibility of some lower bounds.

We would like to caution the reader. Initially we were rather optimistic about removing complexity assumptions in Theorem 1.5 by finding a direct proof of Conjecture 9.1 or some other similar lower bound. However, Proposition 7.3 and Remark 7.4 seem to suggest that this might be rather difficult. A sufficiently strong argument that shows $\mathsf{G} \subsetneq \mathsf{GH}$ could potentially show $\mathsf{UP/poly} = \mathsf{U}\boldsymbol{\Pi}_1^{\mathsf{G}} \subsetneq \mathsf{GH}$, which implies $\#\mathsf{P} \not\subseteq \mathsf{FP/poly}$, an important open problem (see Section 12.5 below).

On the other hand, the two lowest levels G and $\boldsymbol{\Sigma}_1^{\mathsf{G}}$ in GH seem to behave quite differently from higher ones. So an elementary approach to prove $\mathsf{G} \subsetneq \mathsf{GH}$ is not completely ruled out.

12.4. The idea of Section 10 is to characterize all short GFs. Roughly, Conjecture 10.4 says that every short GF is the projection of a union of

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

polynomially many polyhedra of bounded dimension. This can be viewed as a converse of the Barvinok–Woods theorem (Theorem 3.19).

Conjecture 10.4 is possibly a wishful thinking. Unfortunately, its validity is hard to judge since we have so few explicit constructions of short GFs other than projections of integer points in polyhedra. If true, Proposition 10.5 implies Conjecture 1.1 and removes the complexity assumptions from all theorems in the introduction. Moreover, it implies exponential lower bounds on the length of short GF for squares, projections and other theorems in the introduction. (In the chain of reductions, the exponential factor appears in the proof of Proposition 9.10.) These are the same bounds the *exponential time hypothesis* (ETH) implies.

12.5. It is worth comparing Theorems 9.2 and 9.3 from the computational complexity point of view. Technically speaking, these two results are not comparable. However, one is weaker than the other in the relative sense, as follows.

Recall that INTEGER FACTORING $\in \mathsf{NP} \cap \mathsf{coNP}$. While proving it to be in BPP would be a very strong result beyond the current state of art, it would not directly lead to a collapse of PH. In fact, the experts seem to be split on whether INTEGER FACTORING is in P, all the while espousing a deep-seated belief that $\mathsf{P} = \mathsf{BPP}$, thus further muddling the subject (see [**Aar**, **Gas**]). In summary, Theorem 9.2 gives a relatively weak evidence in favor of Conjecture 9.1.

On the other hand, $\#\mathsf{P}$-complete oracles are very powerful by Toda's theorem, and thus very unlikely to be in $\mathsf{FP/poly}$. As mentioned in Remark 7.5, $\#\mathsf{P} \subseteq \mathsf{FP/poly}$ would lead to a collapse of PH the second level. In other words, Theorem 9.3 gives a very strong evidence in favor of Conjecture 9.1.

Acknowledgements

We are grateful to Matthias Aschenbrenner, Sasha Barvinok, Boris Bukh, Terry Tao, Kevin Woods, Josh Zahl and the anonymous referees for many helpful remarks on the subject. We are also thankful to Joshua Grochow, Emil Jerábek for help with complexity questions. The second author was partially supported by the NSF.

References

[Aar] S. Aaronson, ‘$\mathsf{P} \overset{?}{=} \mathsf{NP}$’, in *Open Problems in Mathematics* (Springer, New York, 2016), 1–122.

[AB] S. Arora and B. Barak, *Computational Complexity: A Modern Approach* (Cambridge University Press, Cambridge, UK, 2009).

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

[BMS] E. Bach, G. Miller and J. Shallit, ‘Sum of divisors, perfect numbers and factoring’, *SIAM J. Comput.* **15** (1986), 1143–1154.

[B1] A. Barvinok, ‘A polynomial time algorithm for counting integral points in polyhedra when the fimension is fixed’, in *Proc. 34th FOCS* (IEEE, Los Alamitos, CA, 1993), 566–572.

[B2] A. Barvinok, ‘The complexity of generating functions for integer points in polyhedra and beyond’, in *Proc. ICM,* Vol. 3 (EMS, Zürich, 2006), 763–787.

[B3] A. Barvinok, *Integer Points in Polyhedra* (EMS, Zürich, 2008).

[BP] A. Barvinok and J. E. Pommersheim, ‘An algorithmic theory of lattice points in polyhedra’, in *New Perspectives in Algebraic Combinatorics* (Cambridge University Press, Cambridge, UK, 1999), 91–147.

[BW] A. Barvinok and K. Woods, ‘Short rational generating functions for lattice point problems’, *J. Amer. Math. Soc.* **16** (2003), 957–979.

[BGP] E. Bombieri, A. Granville and J. Pintz, ‘Squares in arithmetic progressions’, *Duke Math. J.* **66** (1992), 369–385.

[Cai] J.-Y. Cai, ‘$\mathsf{S}_2^{\mathsf{P}} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$’, *J. Comput. System Sci.* **73** (2007), 25–35.

[Eis] F. Eisenbrand, ‘Integer programming and algorithmic geometry of numbers’, in *50 Years of Integer Programming* (Springer, Berlin, 2010), 505–560.

[Gas] W. I. Gasarch, ‘The second $\mathsf{P} \overset{?}{=} \mathsf{NP}$ poll’, *ACM SIGACT News* **43**(2) (2012), 53–77.

[Grä] E. Grädel, ‘The complexity of subclasses of logical theories’, Dissertation, Universität Basel, 1987.

[Guy] R. K. Guy, *Unsolved Problems in Number Theory*, 3rd edn (Springer, New York, 2004).

[HW] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers* (Oxford University Press, Oxford, UK, 2008).

[Hou] B. Hough, ‘Solution of the minimum modulus problem for covering systems’, *Ann. of Math. (2)* **181** (2015), 361–382.

[Kan] R. Kannan, ‘Lattice translates of a polytope and the Frobenius problem’, *Combinatorica* **12** (1992), 161–177.

[KS] A. Klivans and D. Spielman, ‘Randomness efficient identity testing of multivariate polynomials’, in *Proc. 33rd FOCS* (ACM, New York, 2001), 216–223.

[LO] J. Lagarias and A. Odlyzko, ‘Computing $\pi(x)$: an analytic method’, *J. Algorithms* **8** (1987), 173–191.

[MA] K. Manders and L. Adleman, ‘NP-complete decision problems for binary quadratics’, *J. Comput. System Sci.* **16** (1978), 168–184.

[MM] C. Moore and S. Mertens, *The Nature of Computation* (Oxford University Press, Oxford, UK, 2011).

[NP1] D. Nguyen and I. Pak, ‘Complexity of short Presburger arithmetic’, in *Proc. 49th STOC* (ACM, New York, 2017), 812–820.

[NP2] D. Nguyen and I. Pak, ‘Enumeration of integer points in projections of unbounded polyhedra’, in *Proc. IPCO 2017*, Lecture Notes in Computer Science, 10328 (Springer, New York, 2017), 417–429.

[NP3] D. Nguyen and I. Pak, ‘The computational complexity of integer programming with alternations’, in *Proc. 32nd CCC, 2017* (LIPICS, Dagstuhl, Germany, 2017), Art. 6, 18 pp.

[NP4] D. Nguyen and I. Pak, ‘Short Presburger arithmetic is hard’, in *Proc. 58th FOCS* (IEEE, Los Alamitos, CA, 2017), 37–48.

[Pap] C. H. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1994).

[Sch] U. Schöning, ‘Complexity of Presburger arithmetic with fixed quantifier dimension’, *Theory Comput. Syst.* **30** (1997), 423–428.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press

[Sze] E. Szemerédi, 'The number of squares in an arithmetic progression', *Studia Sci. Math. Hungar.* **9**(3–4) (1974), 417.

[TV] T. Tao and V. H. Vu, *Additive Combinatorics* (Cambridge University Press, Cambridge, UK, 2006).

[Wei] A. Weil, *Number Theory. An Approach Through History* (Birkhäuser, Boston, MA, 1984).

[W1] K. Woods, 'Rational generating functions and lattice point sets', PhD Thesis, University of Michigan, 2004, 112 pp.

[W2] K. Woods, 'Presburger arithmetic, rational generating functions, and quasi-polynomials', *J. Symbolic Logic* **80** (2015), 433–449.

https://doi.org/10.1017/fms.2017.29 Published online by Cambridge University Press