Combinatorial Expressions and Lower Bounds

Thomas Colcombet and Amaldev Manuel

LIAFA, Université Paris-Diderot
{thomas.colcombet, amal}@liafa.univ-paris-diderot.fr

Abstract

A new paradigm, called combinatorial expressions, for computing functions expressing properties over infinite domains is introduced. The main result is a generic technique, for showing indefinability of certain functions by the expressions, which uses a result, namely Hales-Jewett theorem, from Ramsey theory. An application of the technique for proving inexpressibility results for logics on metafinite structures is given. Some extensions and normal forms are also presented.

1998 ACM Subject Classification F.1.1 Models of Computation, G.2.1 Combinatorics

Keywords and phrases expressions, lower bound, indefinability

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.249

1 Introduction

In this paper, we study the computational power of parallel devices that have an unlimited access to Boolean computations, as well as access to an infinite domain of 'data' (for instance integers or positive integers in what follows) under the restriction of a limited 'bandwidth'. This limitation formally states that only a bounded number of data can be manipulated simultaneously. This is in contrast to the operations over Boolean values that have unlimited input size.

A typical model of this form consists of finite circuits–we call them *combinatorial expressions* throughout the paper–, of bounded depth, in which gates are of two kinds: gates with unbounded domain using arbitrary operations of fan-in at most two (for instance the binary gcd, the binary sum, product, or even non-computable functions), and gates using inputs ranging over a finite domain with unrestricted fan-in (for instance disjunctions, conjunctions of unbounded fan-in, or majority gates).

We use these devices for studying problems that have sequences of data as input. Typical such problems are:

- does a sequence of positive integers (data) have a gcd of one?
- does a sequence of integers sum to zero?
- are all the integers in a sequence distinct?

The motivation of the authors in studying such devices arose from proofs of lower bounds for logics over data-words. The essence of these lower bounds can be easily captured by reduction to lower bounds over combinatorial expressions. Independently of this motivation, we believe that the objects presented here deserve a study on their own. We chose to include here a simpler application to indefinability results in metafinite model theory.

Contributions Our contributions concerning such models go in several directions.

- We introduce the model of combinatorial expressions, show some normal forms for them.
- We prove indefinability results for these expressions: Indefinability of functions (i.e., maps from tuples of data to data) using the pigeonhole principle, and indefinability of problems

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 249–261
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(i.e., maps of tuples of data to $\{0,1\}$) using reductions to problems of 'window definability' (see immediately below).

- We introduce the questions of 'window definability'. Window definable properties are properties that can be described as Boolean combinations of properties over subsets of the inputs (namely windows). Using combinatorial arguments from Ramsey theory, namely Hales-Jewett theorem, we show that some problems are window indefinable.
- We study the added expressive power when expressions are furthermore allowed to use selection gates.
- Finally, we apply these techniques for proving some indefinability results over metafinite structures (these are finite structures, in which tuples can take values in some fixed infinite domain).

Related works This work is of course is related to circuit complexity (see for instance [6]), and more precisely to families of circuits of bounded depth, since the object we are manipulating can be seen as circuits. However, the expressive power of the models that we study is very different. Indeed, not only, our combinatorial expressions can deal with data ranging over an infinite domain, but furthermore, even Boolean gates are not restricted to a simple families like Boolean connectives.

It was brought to our attention that our lower bound result is related to a result of Pascal Tesson stating that testing whether k subsets of $[n]$ form a partition requires a non-constant communication complexity in the k-party 'input on the forehead" model [7, 8]. The two results also make an analogous use of the Hales-Jewett theorem.

There are other families of machines that have been extended to infinite domains, this is in particular the subject of study of algebraic complexity (in particular [1, 5], or [9] for circuits). However, the principle of these branches of work is to see how allowing machines to have primitive capabilities to perform computations in a (infinite) field changes their expressive power. The fact that the infinite domain is equipped of an algebraic (field) structure, changes radically the expressive power, and in particular relates this branch of research to the study of polynomials.

Organisation of the paper In Section 2, we present combinatorial expressions and some motivating examples, which is then followed by a normal form theorem for the expressions and a simple indefinability result. Section 3 contains the main contribution of this paper, namely the indefinability results using the Hales-Jewett theorem. In Section 4, an extended class of expressions is presented and a normal form theorem is given for this class. Section 5 presents an application of the indefinability result to metafinite logics. In Section 6 we discuss some interesting directions for future work and conclude.

Acknowledgments The Hales-Jewett theorem was brought to our attention by Srikanth Srinivasan, and the link to multiparty communication complexity by Frederic Magniez.

2 Combinatorial expressions and normal form

The aim of this section is to introduce the objects of our study, namely *combinatorial expressions*. As usual, $\mathbb{Z}$ (*resp.* $\mathbb{N}$) is the set of (*resp.* non-negative) integers, and $[n]$ denotes the set $\{1, \ldots, n\}$.

2.1 Combinatorial expressions

Combinatorial expressions are built by composing partial maps over a *data domain* $\mathcal{D}$ which is an infinite set. Typical instances of data domains are integers ($\mathbb{Z}$), natural numbers ($\mathbb{N}$), words over a finite alphabet (A^* where A is a finite alphabet) etc. A variable X has *range* $E \subseteq \mathcal{D}$, abbreviated as $X : E$, if the set of values over which it ranges is E. We assume an infinite supply of variables for each range E. A map $f : E_1 \times \cdots \times E_k \to F$, where $E_1, \ldots, E_k, F \subseteq \mathcal{D}$, has *arity* k, *domain* $E_1 \times \cdots \times E_k$ and *range* F. The *image* of the map f is the set of values in F that f maps to, i.e. the set $\{f(a_1, \ldots, a_k) \mid a_1 \in E_1, \ldots, a_k \in E_k\}$. The expressions are built using two specific classes of functions, namely:

- *binary functions* — when $k = 2$, and,
- *finitary functions* — when each of $E_1, \ldots, E_k$ is finite.

A binary function has a bounded arity but may have an unbounded input domain, for example the usual addition on naturals $+ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is binary. On the other hand a finitary function has a finite input domain, but no restriction on the arity. An example of a finitary function is the Boolean conjunction over k inputs $\bigwedge_k : \{0,1\}^k \to \{0,1\}$. Now we formally define combinatorial expressions.

▶ **Definition 1.** *Combinatorial expressions* are defined inductively;

- a variable $X : E$ is a combinatorial expression with range E, and,
- if $f : E_1 \times \cdots \times E_k \to F$ is a binary or finitary function, and $t_1, \ldots, t_k$ are combinatorial expressions with ranges $E_1, \ldots, E_k$ respectively, then $f(t_1, \ldots, t_k)$ is a combinatorial expression with range F.

Let t be a combinatorial expression that contains (possibly vacuously) the variables $\bar{X} = X_1 : E_1, \ldots, X_n : E_n$. We indicate the variables of t by the notation $t(\bar{X})$. For the valuation $\bar{a} = a_1, \ldots, a_n$, $a_i \in E_i$ of the variables $\bar{X}$, the value of the expression t, denoted as $t(\bar{a})$, is defined in the obvious way; if t is a variable X_i then $t(\bar{a}) = a_i$, and if $t = f(t_1, \ldots, t_k)$ then $t(\bar{a}) = f(t_1(\bar{a}), \ldots, t_k(\bar{a}))$. Assume $F \subseteq \mathbb{N}$ is the range of the expression t. Naturally t defines a map $t : \bar{a} \to t(\bar{a})$ from the set $E_1 \times \cdots \times E_n$ to the set F. Like in the case of functions, the *image* of the term t is the set of *output* values of t, i.e. the set $\{t(\bar{a}) \mid \bar{a} \in E_1 \times \cdots \times E_n\}$. Given a map $m : \mathcal{D}^n \to \mathcal{D}$ we say the map is *realised* by an expression t if t defines the map m.

Next we introduce the notion of *depth* of an expression. For a variable X the depth is 0. For an expression $f(t_1, \ldots, t_k)$ the depth is 1 more than the maximum of depths of $t_1, \ldots, t_k$.

▶ **Definition 2** (Family of combinatorial expressions)**.** Fix a sequence $X_1 : \mathcal{D}, X_2 : \mathcal{D}, \ldots$ of variables. A family of combinatorial expressions is a sequence of expressions $(t_n)_{n\in\mathbb{N}} = t_1, t_2, \ldots$ where t_n is an expression over the variables $X_1, \ldots, X_n$.

A family of combinatorial expressions defines a map $(t_n)_{n\in\mathbb{N}} : a_1, \ldots, a_n \to t_n(a_1, \ldots, a_n)$ from $\mathcal{D}^*$ (all finite sequences over $\mathcal{D}$) to $\mathcal{D}$. The family $(t_n)_{n\in\mathbb{N}}$ is of *constant depth* if there is a $k \in \mathbb{N}$ such that each expression t_n is of depth at most k.

Given a map $m : \mathcal{D}^* \to \mathcal{D}$, we say m is *realisable* (by a constant depth family) if there is a family of combinatorial expressions (of constant depth) $(t_n)_{n\in\mathbb{N}}$ that defines m. A particular case is when the range of the map m is restricted to a set of size two, without loss of generality we assume it is $\{0,1\}$; in this case we say $(t_n)_{n\in\mathbb{N}}$ realises the *property* (or *problem*) $\{a_1, \ldots, a_n : m(a_1, \ldots, a_n) = 1\}$.

▶ **Example 3.** Some examples of combinatorial expressions and families are given below. We take the domain $\mathcal{D}$ to be the set of natural numbers $\mathbb{N}$.

1. Fix a number $k \in \mathbb{N}$. Let $m : \mathbb{N}^* \to \{0, \dots, k\}$ be the map $m : a_1, \dots, a_n \to (\sum_i a_i) \bmod k$. The map m is realised by the family $(f_n(g(X_1), \dots, g(X_n)))_{n\in\mathbb{N}}$, where $f_n : \{0, \dots, k\}^n \to \{0, \dots, k\}$ is the finitary function $f_n : a_1, \dots, a_n \to (\sum_i a_i) \bmod k$, and $g : \mathbb{N} \to \{0, \dots k\}$ is the binary function (by abuse of notation) $g : i \to i \bmod k$. This family has depth 2.
2. Let P_1 be the set of all finite sequences of non-zero naturals. This property is realised by the family $(\bigwedge_n(\mathit{zero}(X_1), \dots, \mathit{zero}(X_n)))_{n\in\mathbb{N}}$, where $\mathit{zero} : \mathbb{N} \to \{0, 1\}$ is the binary function that maps precisely all the non-zero naturals to 1, and $\bigwedge_n$ is the finitary function that defines the Boolean conjunction on n inputs. This family has depth 2.
3. Let P_2 be the property $\{a_1, \dots, a_n \in \mathbb{N}^* : a_i \neq a_j\}$. Let $\mathit{neq} : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ be the binary function that has value 1 precisely when the inputs differ. Then, the property P_2 is realised by the family (of depth 2) $(\bigwedge_{n\cdot(n-1)/2}(t_{12}, \dots, t_{1n}, t_{23} \dots, t_{2n}, \dots, t_{n-1n}))_{n\in\mathbb{N}}$ where the expression $t_{ij} = \mathit{neq}(X_i, X_j)$.
4. We claim that for any map $m : \mathbb{N}^* \to \mathbb{N}$ there is a family $(t_n)_{n\in\mathbb{N}}$ of *logarithmic depth* (i.e. the expression t_n has depth at most $\log n$) realising it. Let $p_1, p_2, \dots$ be an enumeration of the prime numbers. For each prime p_i, let p_i-$\mathit{exp} : x \to p_i^x$ be the exponential function with base p_i. Define the binary function $u_n : \mathbb{N} \to \mathbb{N}$ as

   $$u_n(x) = m(a_1, \dots, a_n) \text{ where } a_i \text{ is the exponent of } p_i \text{ in } x$$

   Finally let $\pi_n(X_1, \dots, X_n)$ be the expression (of $\log n$-depth) that computes the product of the variables $X_1, \dots, X_n$. Then, the family of expressions

   $$(u_n(\pi_n(p_1\text{-}\mathit{exp}(X_1), \dots, p_n\text{-}\mathit{exp}(X_n)))_{n\in\mathbb{N}}$$

   realises the map m.

Example 3.4 implies that the class of maps realised by families of expressions of logarithmic depth is degenerate, i.e. every map is realisable. Hence for interesting results one has to consider the class of families of sub-logarithmic depth. In the following we study the class of families of expressions of constant depth and prove that it is non-degenerate, i.e. there are maps and properties that are not realisable.

2.2 Normal form and definability

In the rest of the section we exhibit a normal form for the expressions. First we introduce the important notion of semantic equivalence of expressions.

▶ **Definition 4** (Equivalence of expressions)**.** Two expressions $t_1(\bar{X})$ and $t_2(\bar{X})$ over the variables $\bar{X} = X_1 : E_1, \dots, X_n : E_n$ are *equivalent* if $t_1(\bar{a}) = t_2(\bar{a})$ for all $\bar{a} = a_1, \dots, a_n$, $a_i \in E_i$.

We introduce some notation. For an expression t, we denote the range and image of t by $\mathit{range}(t)$ and $\mathit{image}(t)$ respectively. Assume $\bar{t} = t_1, \dots, t_n$ is a finite sequence of expressions. We define $\mathit{len}(\bar{t}) = n$ and $\mathit{range}(\bar{t}) = \mathit{range}(t_1) \times \dots \times \mathit{range}(t_n)$. If $\bar{s} = s_1, \dots, s_m$ and $\bar{t} = t_1, \dots, t_n$ are two sequences of expressions, then $\bar{s}, \bar{t}$ denotes the sequence $s_1, \dots, s_m, t_1, \dots, t_n$. An expression t is a *binary expression* if it consists only of binary functions.

The normal form theorem is obtained by transforming the expressions and for that we use the idea of pairing, i.e. encoding pairs of elements from $\mathcal{D}$ as an element in $\mathcal{D}$. We use the following fact from set theory.

▶ **Fact 1** (See [4], Chapter 3)**.** *If A is an infinite set, there exists an injective map from $A \times A$ to A.*

Fix an injective map π from $\mathcal{D} \times \mathcal{D}$ to $\mathcal{D}$. For elements $a_1, a_2 \in \mathcal{D}$ we let $\langle a_1, a_2 \rangle$ denote the element $\pi(a_1, a_2) \in \mathcal{D}$. Similarly for subsets $E_1, E_2 \in \mathcal{D}$ we let $\langle E_1, E_2 \rangle$ denote the set $\{\langle a_1, a_2 \rangle \mid a_1 \in E_1, a_2 \in E_2\}$.

▶ **Theorem 5** (Normal form). *For every expression $t(\bar{X})$ of depth $\ell \in \mathbb{N}$ there exists an equivalent expression of the form*

$$b(r(\bar{X}), f(\bar{s}(\bar{X}))) \tag{1}$$

where b is a binary function, f is a finitary function, $r(\bar{X})$ is a binary expression of depth at most ℓ, and $\bar{s}(\bar{X})$ is a sequence of binary expressions of depth at most ℓ. Moreover, if the image of $t(\bar{X})$ is finite then there exists an equivalent expression of the form

$$f(\bar{s}(\bar{X})) \tag{2}$$

where f is a finitary function and $\bar{s}(\bar{X})$ is a sequence of binary expressions of depth at most ℓ.

The normal form theorem allows us to formulate arguments about the expressive power of our model. We next give such an application.

▶ **Proposition 1.** *Let $m : \mathcal{D}^{2^k+1} \to \mathcal{D}$ be a map that satisfies the following property:* (†) For any index $i \in [2^k + 1]$ and any $n \in \mathbb{N}$ there exist values $a_1, \dots a_{i-1}, a_{i+1}, \dots, a_{2^k+1} \in \mathcal{D}$ such that the set $\{m\left(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_{2^k+1}\right) \mid b \in \mathcal{D}\}$ is of size at least n. *Then the map m is* not *realisable by an expression of depth at most k.*

Proof. The proof is an application of the pigeonhole principle along with the normal form theorem. Let m be a map that satisfies the Property (†).

Let $\bar{X} = X_1, \dots, X_{2^k+1}$ be the input variables with range $\mathcal{D}$. Assume there is an expression $t(\bar{X})$ of depth k realising the map m. Using the normal form theorem we obtain an equivalent expression $t'(\bar{X})$ of the form

$$b(r(\bar{X}), f(\bar{s}(\bar{X})))$$

where b is a binary, f is finitary, $r(\bar{X})$ is a binary expression of depth at most k and $\bar{s}(\bar{X})$ is a sequence of binary expressions of depth at most k. Let the image of the function f be of size n. Since the binary expression $r(\bar{X})$ has depth at most k, there is a variable X_i that is not used by the expression $r(\bar{X})$. Consider the following set of input tuples;

$$\begin{array}{ccccccc}
a_1, & \dots, & a_{i-1}, & b_1, & a_{i+1}, & \dots & a_{2^k+1} \\
a_1, & \dots, & a_{i-1}, & b_2, & a_{i+1}, & \dots & a_{2^k+1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_1, & \dots, & a_{i-1}, & b_m, & a_{i+1}, & \dots & a_{2^k+1}
\end{array}$$

where $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_{2^k+1} \in \mathcal{D}$ and $b_1, b_2, \dots \in b_m \in \mathcal{D}$, $m > n$ are such that

$$|\{m\left(a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_{2^k+1}\right) \mid i \in [m]\}| > n$$

and for each $j \neq l$ it is the case that

$$m\left(a_1, \dots, a_{i-1}, b_j, a_{i+1}, \dots, a_{2^k+1}\right) \neq m\left(a_1, \dots, a_{i-1}, b_l, a_{i+1}, \dots, a_{2^k+1}\right) .$$

Existence of $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_{2^k+1} \in \mathcal{D}$ and $b_1, b_2, \dots, b_m \in \mathcal{D}$ are guaranteed by the Property (†). The tuples differ only on the input variable X_i. Hence on all these inputs the binary expression $r(\bar{X})$ has the same output. Moreover, since f has a finite image of size n and the number of input tuples is more than n, by pigeonhole principle there exist two input tuples on which f has the same output. Let them be

$$\begin{array}{ccccccc} a_1, & \dots, & a_{i-1}, & b_{j_1}, & a_{i+1}, & \dots & a_{2^k+1}\,, \\ a_1, & \dots, & a_{i-1}, & b_{j_2}, & a_{i+1}, & \dots & a_{2^k+1}\,. \end{array}$$

It follows that on these two inputs the expressions $r(\bar{X})$ and $f(\bar{s}(\bar{X}))$ have the same output and hence the function b also has the same output. But clearly the map m differs on these inputs which is a contradiction. Hence the claim is established. ◀

▶ **Corollary 6.** *The following maps are not realised by expressions of depth at most k.*

1. $gcd : (\mathbb{N} \setminus \{0\})^{2^k+1} \to \mathbb{N} \setminus \{0\}$ *defined as* $gcd : a_1, a_2, \dots, a_{2^k+1} \to \mathrm{gcd}(a_1, a_2, \dots, a_{2^k+1})$,

2. $sum : \mathbb{Z}^{2^k+1} \to \mathbb{Z}$ *defined as* $sum : a_1, a_2, \dots, a_{2^k+1} \to a_1 + \dots + a_{2^k+1}$.

Proof. By virtue of the previous proposition, it is enough to establish the Property (†) for each of the maps m.

1. For each $i \in 2^k + 1$ and $n \in \mathbb{N}$, we define the following tuples ;

$$\begin{array}{ccccccc} 2^{n+1}, & \dots, & 2^{n+1}, & 2, & 2^{n+1}, & \dots, & 2^{n+1} \\ 2^{n+1}, & \dots, & 2^{n+1}, & 2^2, & 2^{n+1}, & \dots, & 2^{n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2^{n+1}, & \dots, & 2^{n+1}, & 2^{n+1} & 2^{n+1}, & \dots, & 2^{n+1} \\ \multicolumn{3}{c}{\underbrace{\hphantom{2^{n+1}, \quad \dots, \quad 2^{n+1},}}_{i-1 \text{ times}}} & & \multicolumn{3}{c}{\underbrace{\hphantom{2^{n+1}, \quad \dots, \quad 2^{n+1}}}_{2^k+1-i \text{ times}}} \end{array}$$

It is straightforward to check that the image of the map gcd on these tuples is of size $n + 1$.

2. As before, for each $i \in 2^k + 1$ and $n \in \mathbb{N}$, we define the following tuples ;

$$\begin{array}{ccccccc} 0, & \dots, & 0, & 1, & 0, & \dots, & 0 \\ 0, & \dots, & 0, & 2, & 0, & \dots, & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0, & \dots, & 0, & n+1 & 0, & \dots, & 0 \\ \multicolumn{3}{c}{\underbrace{\hphantom{0, \quad \dots, \quad 0,}}_{i-1 \text{ times}}} & & \multicolumn{3}{c}{\underbrace{\hphantom{0, \quad \dots, \quad 0}}_{2^k+1-i \text{ times}}} \end{array}$$

It is easily verified that the image of the map sum on these tuples is of size $n + 1$. ◀

Before concluding this section let us note that the arguments we used in Proposition 1 does not work for proving indefinability of maps with a finite image, for instance the map $P_{gcd=1} : (\mathbb{N} \setminus \{0\})^* \to \{0, 1\}$ defined as $P_{gcd=1}(a_1, a_2, \dots, a_m) = 1$ iff $gcd(a_1, \dots, a_m) = 1$. In the next section we develop advanced techniques for handling such maps.

3 Window-definability and indefinability

In this section, we provide the necessary material for showing that some problems are not expressible by combinatorial expressions of bounded depth. This is different from the indefinability result that we have seen before, Proposition 1, which was dealing with the indefinability of functions that have an infinite/unbounded image while the maps we consider in this section have an image of size 2. For this, we slightly depart from the above framework, and introduce the notion of window-definability.

In the following we use the notation A^B to denote the set of all vectors/sequences over A indexed by the set B. Let us fix a finite set of variables, $\mathcal{V} = \{X_1, \ldots, X_k\}$ ranging over $\mathcal{D}$. A *window* (over $\mathcal{V}$) is a subset of $\mathcal{V}$. Given a valuation of the variables $\bar{a} \in \mathcal{D}^{\mathcal{V}}$, its restriction to a window W is denoted $\bar{a}|_W$. Two valuations v, v' are *W-equivalent* if $v|_W = v'|_W$, i.e., indistinguishable 'through the window W'. From now, $\mathcal{W}$ designates a fixed set of windows. A problem $P \subseteq \mathcal{D}^k$ is *$\mathcal{W}$-definable* if it can be described as a Boolean combination of languages of the form

$$\{\bar{a} \in \mathcal{D}^{\mathcal{V}} \mid \bar{a}|_W \in S\} \quad \text{for some } S \subseteq \mathcal{D}^W.$$

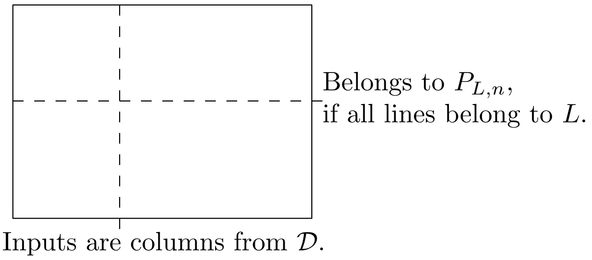
Such a Boolean combination is called a *$\mathcal{W}$-definition.* In other words, the membership to P is entirely determined given finitely many properties of the input 'seen through the windows'. Of course, if $\mathcal{V} \in \mathcal{W}$, then all problems are $\mathcal{W}$-definable. We are interested in understanding the notion of $\mathcal{W}$-definability when this is not the case (i.e. $\mathcal{V} \notin \mathcal{W}$). The *size of a $\mathcal{W}$-definition* is the number of sets of the above form it uses. A problem is *$\mathcal{W}, k$-definable* if there is a $\mathcal{W}$-definition for it of size at most k. Two problems P, R are *$\mathcal{W}, k$-separable* if there is a $\mathcal{W}, k$-definable set D such that $P \cap D = \emptyset$ and $R \subseteq D$.

These notions are related to the above sections thanks to the following lemma:

▶ **Lemma 7.** *The problems definable by combinatorial expressions of depth k are $\mathcal{W}$-definable, where $\mathcal{W}$ is the set of all windows of size at most 2^k.*

Proof. By the normal form theorem, any expression of depth k deciding a property (since it has a finite image) is equivalent to an expression of the form $f(\bar{s}(\bar{X}))$ where f is a finitary function, and $\bar{s}(\bar{X})$ is a sequence of binary expressions of depth at most k. First of all we observe that we can assume that every binary expression in $\bar{s}(\bar{X})$ outputs a Boolean value, in other words f computes a Boolean function. Notice that there is no loss of generality here since any finitary function can be converted to a Boolean function by increasing the number of inputs. Now the claim follows by observing that each binary expression $s_i(\bar{X})$ in $\bar{s}(\bar{X})$ corresponds to a set of the form $\{\bar{a} \in \mathcal{D}^{\mathcal{V}} \mid \bar{a}|_W \in S\}$ where the window W is of size at most 2^k (namely the valuables used by the expression $s_i(\bar{X})$). ◀

We shall now head toward indefinability results. For this, we use an exact characterization of the definability for some special forms of problems: *rectangle problems.* In such problems, data are column vectors. Hence, tuples of data as input can be seen as rectangles. We are interested in relating the $\mathcal{W}$-definability of the set of rectangles such that every line belongs to some given set of valid rows L, to some simpler properties of L.


Belongs to $P_{L,n}$,
if all lines belong to L.
Inputs are columns from $\mathcal{D}$.

Formally, fix an alphabet A, and a *line property* $L \subseteq A^{\mathcal{V}}$. Then, given a positive integer n, consider the domain $\mathcal{D}_n = A^n$ (understood as 'columns'), and define the problem $P_{L,n} \subseteq \mathcal{D}_n^{\mathcal{V}}$

consisting of these rectangles such that every line belongs to L:

$$P_{L,n} = \{\bar{a} \in \mathcal{D}_n^{\mathcal{V}} \mid \pi_i(\bar{a}) \in L \quad \text{for all} \quad i \in [n]\}\ ,$$
$$\text{where} \quad \pi_i(a_1 \dots a_n) = a_i \quad \text{is extended component-wise to tuples indexed by } \mathcal{V}.$$

Our Theorem 8, just below, relates the $\mathcal{W}$-definability of these problems to the property of L being $\mathcal{W}$-closed, that we define now.

An element $\bar{a} \in A^{\mathcal{V}}$ belongs to the *$\mathcal{W}$-closure of $L \subseteq A^{\mathcal{V}}$*, denoted $\overline{L}^{\mathcal{W}}$, if for all $W \in \mathcal{W}$ there is some $\bar{b} \in L$ such that $\bar{a}$ and $\bar{b}$ are W-equivalent. The set L is *$\mathcal{W}$-closed* if it is equal to its $\mathcal{W}$-closure.

The interesting examples are more the negative ones: consider for instance $A = \{0, 1\}$, $L \subseteq A^{\mathcal{V}}$ the set of tuples that contain at least one occurrence of 1, and $\mathcal{W}$ to be $2^{\mathcal{V}} \setminus \mathcal{V}$, then L is not $\mathcal{W}$-closed since $\bar{a} = 0^{\mathcal{V}}$ does not belong to L, but for all windows $W \in \mathcal{W}$ we can define $\bar{b}$ to be 0 over W and 1 elsewhere. This $\bar{b}$ is W-equivalent to $\bar{a}$, and since it contains at least one occurrence of 1, it belongs to L.

▶ **Theorem 8.** *For all $L \subseteq A^{\mathcal{V}}$,*

- *if L is $\mathcal{W}$-closed then there is some k such that all $P_{L,n}$ has a $\mathcal{W}, k$-definition for all positive integers n, and*
- *if L is not $\mathcal{W}$-closed, then for all k, there exists n such that $P_{L,n}$ has no $\mathcal{W}, k$-definition.*

Proof of the first item. Assume that L is $\mathcal{W}$-closed, this means that:

$$L = \{\bar{a} \in A^{\mathcal{V}} \mid \bar{a}|_W \in L|_W \text{ for all } W \in \mathcal{W}\}\ , \qquad \text{where} \quad L|_W = \{\bar{a}|_W \mid \bar{a} \in L\}\ .$$

Consider now the set $R_{L,n} \subseteq (\mathcal{D}_n)^{\mathcal{V}}$ that contains $u \in (\mathcal{D}_n)^{\mathcal{V}}$ if $u|_W \in (L|_W)^n$ for all $W \in \mathcal{W}$. By definition, $R_{L,n}$ is $\mathcal{W}, |\mathcal{W}|$-definable. Let us show that $R_{L,n} = P_{L,n}$.

Consider some $\bar{a} \in (\mathcal{D}_n)^{\mathcal{V}}$. Then $\bar{a} \in P_{L,n}$ *if and only if* for all $i \in [n]$ and all $W \in \mathcal{W}$, $\pi_i(a|_W) \in L|_W$, *if and only if* for all $W \in \mathcal{W}$ and all $i \in [n]$, $\pi_i(a|_W) \in L|_W$, *if and only if* $\bar{a} \in R_{L,n}$. Hence $P_{L,n} = R_{L,n}$ is $\mathcal{W}, |\mathcal{W}|$-definable. ◀

Before being able to prove the second item, we need to introduce the deep combinatorial theorem of Hales-Jewett. In this theorem, a *combinatorial line* of B^n (for some finite set B and some positive integer n) is a set of the form $\ell = \{u[b] \mid b \in B\}$ for some $u \in (B^*x)^+B^*$, where $u[b]$ denotes u in which b has been substituted to all occurrences of x.

▶ **Theorem 9** (Hales-Jewett [3]). *Given some finite sets B and C, there is a positive integer n such that for all maps χ from B^n to C there exists a $\chi-$monochromatic combinatorial line ℓ, i.e., there is $c \in C$ such that $\chi(v) = c$ for all $v \in \ell$.*

We now use a this theorem for establishing the second item of Theorem 8. We establish in fact the following stronger lemma.

▶ **Lemma 10.** *If $\bar{a} \in \overline{L}^{\mathcal{W}} \setminus L$, then for all k, there exists n such that $P_{L,n}$ and the set $P_{L\cup\{\bar{a}\},n} \setminus P_{L,n}$ cannot be $\mathcal{W}, k$-separated.*

Hence, one cannot separate 'all lines are in L' from 'all lines are in $L \cup \{\bar{a}\}$ and there is a line equal to $\bar{a}$'. It is easy to see that Lemma 10 implies the second item of Theorem 8. Indeed, L being non $\mathcal{W}$-closed means that there exists $\bar{a} \in \overline{L}^{\mathcal{W}} \setminus L$. Assuming for the sake of contradiction that $P_{L,n}$ would be $\mathcal{W}, k$-definable would thus imply that $P_{L,n}$ and $P_{L\cup\{\bar{a}\},n} \setminus P_{L,n}$ would be $\mathcal{W}, k$-separated by $P_{L,n}$ itself. A contradiction.

Proof. Assume that $P_{L,n}$ and $Q_{\bar{a},n} = P_{L\cup\{\bar{a}\},n} \setminus P_{L,n}$ are $\mathcal{W},k$-separable.

Our *first step* consists in showing that the language that separates $P_{L,n}$ from $Q_{\bar{a},n}$ can be derived from a coloring function χ of the inputs to some set C (the size of which does not depend on n). Let us assume that the $\mathcal{W},k$-separability is witnessed by a Boolean combination of the sets $R_i = \{\bar{c} \in (\mathcal{D}_n)^{\mathcal{V}} \mid \bar{c}|_{W_i} \in S_i\}$ for $i \in [k]$ where $W_i \in \mathcal{W}$ and $S_i \subseteq (\mathcal{D}_n)^{W}$. Consider now the set $C = [2]^{[k]}$ (note that it does not depend on n), and the map χ from $(\mathcal{D}_n)^{\mathcal{V}}$ to C defined by

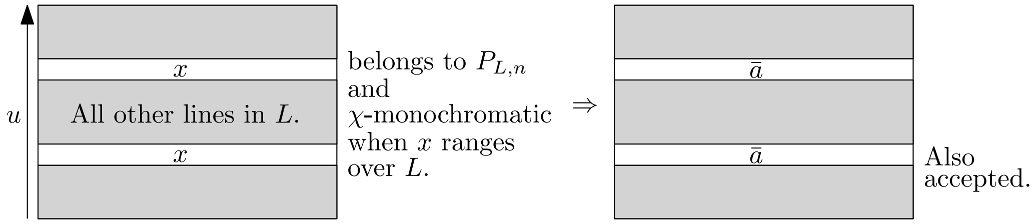
$$\text{for all } i \in [k] \text{ and } \bar{c} \in (\mathcal{D}_n)^{\mathcal{V}}, \quad \chi(\bar{c})_i = \delta_{\bar{c}|_{W_i} \in S_i} = \begin{cases} 1 & \text{if } \bar{c}|_{W_i} \in S_i\ , \\ 0 & \text{otherwise.} \end{cases}$$

This map stores all the relevant information concerning the membership in each of the R_i's. The $\mathcal{W},k$-separability means that whenever $\bar{c} \in P_{L,n}$ and $\bar{c}' \in Q_{\bar{a},n}$, $\chi(\bar{c}) \neq \chi(\bar{c}')$ $(\star)$.

We shall now lay ground for the use of Hales-Jewett. This theorem will be used on rows: a rectangle of the problem will be seen as a sequence of rows, one on top of the previous one. This is different from the input itself, since each variable accounts for one column. We will allow to use these two points of view by implicitly identifying elements from $(A^{\mathcal{V}})^n$ (sequence of rows of length $\mathcal{V}$) with elements from $(A^n)^{\mathcal{V}} = (\mathcal{D}_n)^{\mathcal{V}}$ (sequence of columns of depth n). Under this identification, we can for instance write $L^n = P_{L,n}$, since $P_{L,n}$ consists of these inputs such that every line belongs to L.

We can now apply the Hales-Jewett theorem, using $B = L$ and C as defined during the first step, thus getting a number n. The first step provides us with a coloring χ from B^n to C. Finally, the theorem of Hales-Jewett states the existence of a χ-monochromatic combinatorial line $\ell = \{u[\bar{b}] \mid \bar{b} \in L\}$ for $u \in (L \cup \{x\})^n$, of color $c \in C$.

The principle of the rest of the proof is shown is the following picture:


u
x
All other lines in L.
x
belongs to $P_{L,n}$ and χ-monochromatic when x ranges over L.
⇒
$\bar{a}$
$\bar{a}$
Also accepted.

Let us prove that $\chi(u[\bar{a}]) = c$ (This is a contradiction to $(\star)$ since $u[\bar{a}] \in Q_{\bar{b},n}$, thus completing the proof).

For all $i \in [k]$, we have:

$$\begin{aligned} \chi(u[\bar{a}])_i &= \delta_{u[\bar{a}]|_{W_i} \in S_i} && \text{(by definition of } \chi) \\ &= \delta_{u[\bar{b}]|_{W_i} \in S_i} \quad \text{for some } \bar{b} \in L && \text{(since } \bar{a} \in \overline{L}^{\mathcal{W}}) \\ &= \chi(u[\bar{b}])_i && \text{(by definition of } \chi) \\ &= c_i\ . && \text{(since } u[\bar{b}] \in \ell \text{ hand hence is mapped to } c \text{ by } \chi) \end{aligned}$$

Hence $\chi(u[\bar{a}]) = c$. ◀

We can now derive other indefinability results from Theorem 8.

▶ **Lemma 11.** *Consider a set of windows $\mathcal{W}$ such that $\mathcal{V} \notin \mathcal{W}$.*

- *For $\mathcal{D}$ being the positive integers, then the set $P_{\text{gcd}=1}$ of inputs of* gcd *one is not $\mathcal{W}$-definable.*
- *For $\mathcal{D}$ being the integers, the set $P_{\Sigma=0}$ of inputs of null sum is not $\mathcal{W}$-definable.*

Proof of first item. Let k be a positive integer. Let us show that $P_{\text{gcd}=1}$ is not $\mathcal{W},k$-definable.

Let $A = \{0,1\}$ and $L = A^{\mathcal{V}} \setminus \{0\}^{\mathcal{V}}$. We have seen that this L was not $\mathcal{W}$-closed. Hence, according to Theorem 8, there is n such that the rectangle problem $P_{L,n}$ is not $\mathcal{W},k$-definable. Let $p_1,\dots,p_n$ be n distinct prime numbers. Define the map f from $\mathcal{D}_n = \{0,1\}^n$ to positive integers by $f(a_1\dots a_n) = p_1^{a_1}p_2^{a_2}\cdots p_n^{a_n}$. Clearly, given an input tuple $\bar{a} \in (\mathcal{D}_n)^{\mathcal{V}}$, $\text{gcd}_{v\in\mathcal{V}} f(a(v))$ is 1 if and only if for all prime numbers $p = p_1,\dots,p_n$, $f(a(v))$ is not divisible by p for some $v \in \mathcal{V}$, or equivalently, if all lines in $\bar{a}$ (seen as a rectangle) contains a 0. Hence $P_{L,n} = \{\bar{a} \mid \text{gcd}_{v\in\mathcal{V}} f(a(v))\}$.

Thus, assume that $P_{\text{gcd}=1}$ would be $\mathcal{W},k$-definable, then the problem $P_{L,n}$ would also be $\mathcal{W},k$-definable. This is a contradiction. ◀

Proof of second item. The approach is similar. Let us assume without loss of generality that $|\mathcal{V}| \geq 2$. Let k be a positive integer. Let us show that $P_{\Sigma=0}$ is not $\mathcal{W},k$-definable.

Let $A = \{0,1,-1\}$ and L be the tuples $\bar{b} \in A^{\mathcal{V}}$ that sum to 0. This set is not $\mathcal{W}$-closed. Indeed, consider a tuple $\bar{a}$ that consists only of 0's but for one occurrence of 1. For all windows $W \in \mathcal{W}$, either the occurrence of 1 does not occur in W, and $\bar{a}$ is W-equivalent to the null tuple which belongs to L, or there is some occurrence of 0 that occurs outside W, and by switching this 0 into a -1 yields once more a W-equivalent tuple in L. Hence, according to Theorem 8, for n sufficiently large, $P_{L,n}$ is not $\mathcal{W},k$-definable.

Let us consider now some $\lambda > |\mathcal{V}|$, and the map from $\mathcal{D}_n = A^n$ to integers defined by $f(a_1\dots a_n) = \sum_{i\in[n]} \lambda^i a_i$. Thanks to the choice of a sufficiently large λ, for all inputs $\bar{a} \in (\mathcal{D}_n)^{\mathcal{V}}$, $\sum_{v\in\mathcal{V}} f(a(v)) = 0$ if and only if all rows in $\bar{a}$ sum to 0, i.e., if and only if $\bar{a} \in P_{L,n}$.

Thus, assume that $P_{\Sigma=0}$ would be $\mathcal{W},k$-definable, then the problem $P_{L,n}$ would also be $\mathcal{W},k$-definable. This is a contradiction. ◀

▶ **Corollary 12.** *The problems of null sum and of* gcd *one over more that 2^k inputs are not recognizable by combinatorial expressions of depth at most k.*

4 Selection functions

So far in our expressions we allowed only functions with bounded domain and unbounded arity, or bounded arity and unbounded domain. It is natural to ask if the class of expressions can be extended without being degenerate (i.e. not accepting all maps from $\mathcal{D}^*$ to $\mathcal{D}$). In this section we present the class of selection functions that are similar to the *multiplexer gates* in digital circuits. Intuitively a selection function takes m values $a_1,\dots,a_m \in \mathcal{D}$ and a number i in $[m]$ as input and outputs the value a_i, i.e. it selects the ith input. To keep the discussion simple, let us assume that $\mathcal{D}$ contains the natural numbers $\mathbb{N}$.

▶ **Definition 13.** Formally, a selection function sel_m of arity $m \in \mathbb{N}$ is a function of the form

$$sel_m : \left(\prod_{i\in[m]} E_i\right) \times [m] \to \bigcup_{i\in[m]} E_i$$

where each $E_i \subseteq \mathcal{D}$ such that $sel_m(a_1,\dots,a_m,j) = a_j$ for $a_1 \in E_1,\dots,a_m \in E_m, j \in [m]$.

We define the *extended class of combinatorial expressions* inductively as follows: every combinatorial expression belongs to the extended class. Further more, if $sel_m : \left(\prod_{i\in[m]} E_i\right) \times [m] \to \bigcup_{i\in[m]} E_i$ is a selection function and $t_1,\dots,t_m,t$ are expressions in the extended class with ranges $E_1, \dots, E_m, [m]$ respectively then $sel_m(t_1, \dots, t_m, t)$ is an expression with range $\bigcup_{i \in [m]} E_i$ that belongs to the extended class of terms.

In the following we prove that the extended class of expressions have the same expressive power. First we prove a normal form theorem for the extended class.

▶ **Theorem 14** (Normal form theorem for extended class of expressions). *For every expression $t(\bar{X})$ of depth $\ell \in \mathbb{N}$ there exists an equivalent expression of the form*

$$sel_m(\bar{r}(\bar{X}), f(\bar{s}(\bar{X})))$$

where $m \in \mathbb{N}$, $\bar{r}(\bar{X})$ is a sequence of binary expressions of depth at most ℓ, f is a finitary function, and $\bar{s}(\bar{X})$ is a sequence of binary expressions of depth at most ℓ. Moreover if the image of $t(\bar{X})$ is finite then there exists an equivalent expression of the form

$$f(\bar{s}(\bar{X}))$$

where f is a finitary function and $\bar{s}$ is a sequence of binary expressions of depth at most ℓ.

An immediate consequence of the above result is that for defining functions from $\mathcal{D}^*$ to $\mathcal{D}$ with finite image, selection functions are useless. This situation is not different in general also. From Example 4 it follows that,

▶ Remark. The function $sel_{2^k+1} : \mathcal{D}^{2^k+1} \times [2^k + 1] \to \mathcal{D}$ is definable by a combinatorial expression (that does not use selection functions) of depth $k + 1$.

However selection functions add succinctness as the following propositions shows.

▶ **Proposition 2.** *The function $sel_{2^k+1} : \mathcal{D}^{2^k+1} \times [2^k + 1] \to \mathcal{D}$ is not definable by a combinatorial expression of depth k.*

Proof. The proof is very close to the proof of Proposition 1. Assume there is a combinatorial expression $t(X_1, \dots, X_{2^k+1}, y)$ that defines the function sel_{2^k+1}. By the normal form theorem we transform t into an equivalent expression of the form

$$b(r(X_1, \dots, X_{2^k+1}, y), f(\bar{s}(X_1, \dots, X_{2^k+1}, y)))$$

where b is binary, f is finitary, and r and $\bar{s}$ are binary expressions of depth k. Since the expression r has depth k, there is a variable X_i that is not present in r. Choose and element $a \in \mathcal{D}$ and consider the inputs $S = \{(x_1, \dots, x_{2^k+1}, i) \in \mathcal{D}^{2^k+2} \mid x_i \in \mathcal{D}, \forall j \neq i, x_j = a\}$. Choose inputs $\bar{u}, \bar{v} \in S$ such that $\bar{u} \neq \bar{v}$ and $f(\bar{s}(\bar{u})) = f(\bar{s}(\bar{v}))$. Such inputs $\bar{u}$ and $\bar{v}$ exist by pigeonhole principle (since S is infinite while the image of f is finite). Observe that $b(r(\bar{u}), f(\bar{s}(\bar{u}))) = b(r(\bar{v}), f(\bar{s}(\bar{v})))$ contradicting the fact that $sel_{2^k+1}(\bar{u}) \neq sel_{2^k+1}(\bar{v})$. Hence the claim is proved. ◀

5 Application in metafinite logics

In this section we describe an application of our indefinability results, namely to prove inexpressibility results for logics on metafinite structures. *Metafinite model theory* was initiated by Grädel and Gurevich [2] in order 'to extend the approach and methods of finite model theory beyond finite structures'. A *metafinite structure* $\mathfrak{M}$ is a triple $\langle \mathfrak{A}, \mathfrak{B}, \rho \rangle$ where $\mathfrak{A}$ is a finite first order structure, $\mathfrak{B}$ is a first order structure (typically infinite) and ρ is a weight function from the domain of $\mathfrak{A}$ to the domain $\mathfrak{B}$ (the original definition allows a finite number of weight functions of different arity). For an example consider the structure

$\mathfrak{M} = \langle \mathfrak{A}, \mathfrak{B}, \rho \rangle$ where $\mathfrak{A} = ([n], \leq)$ is a finite linear order, $\mathfrak{B} = (\mathbb{N}, +, \times)$ is the natural numbers with arithmetic, and ρ is a map from $[n]$ to $\mathbb{N}$. In short, $\mathfrak{M}$ represents a sequence of natural numbers. Such kind of *weighted* structures arise naturally in several areas of computer science. An important such case concerns databases, where some data naturally range over infinite/unbounded domains. When considering logics (for instance first order logic) expressing properties over metafinite structures, quantifications are assumed to only range over the finite structure $\mathfrak{A}$, but the formulas can access the structure $\mathfrak{B}$ via the functions ρ and the use of the relations in $\mathfrak{B}$. In [2] several theorems from finite model theory are lifted to the case of metafinite models.

We now show that our indefinability results can be used to derive indefinability results in this context. We consider structures of the following form; $\mathfrak{M} = \langle \mathfrak{A}, \mathfrak{B}, \rho \rangle$ where $\mathfrak{A} = ([n], \leq)$ is a finite linear order, $\mathfrak{B}$ is the natural numbers $\mathbb{N}$ with all possible relations and functions of all arity (denoted as $\mathbb{N}^*$), and ρ is a map from the positions in $\mathfrak{A}$ to $\mathbb{N}$. We consider the *monadic second order logic* on these structures which has the following syntax and semantics: we have first order variables $x, y, \ldots$ and set variables $X, Y, \ldots$ that range over positions and sets of positions in the structure $\mathfrak{A}$ respectively. When x is a variable then $\rho(x)$ is a term over the structure $\mathfrak{B}$, and if $t_1, \ldots, t_k$ are terms over the structure $\mathfrak{B}$ and f is a function in $\mathfrak{B}$ of arity k then $f(t_1, \ldots, t_k)$ is a term over the structure $\mathfrak{B}$. If R is a relation of arity k in $\mathfrak{B}$ and $t_1, \ldots, t_k$ are terms over $\mathfrak{B}$, then $R(t_1, \ldots, t_k)$ is an atomic formula of the logic. The only other atomic formulas are of the form $x \leq y$. The rest of the formulas of the logic are defined inductively: $\varphi_1 \vee \varphi_2$, $\neg\varphi_1$, $\exists x.\ \varphi_1$, $\exists X.\varphi_1$ are formulas when φ_1, φ_2 are formulas. The semantics of terms and formulas are defined in the obvious way (see [2], Definition 3.1).

Using our inexpressibility result we prove the following theorem:

▶ **Theorem 15.** *The set of all structures* $\mathfrak{M} = \langle ([n], \leq), \mathbb{N}^*, \rho \rangle$ *that satisfy the property*

$$\gcd(\rho(1), \ldots, \rho(n)) = 1 \tag{3}$$

is not definable in monadic second order logic.

Before concluding, we note the following. The first remark is that by similar arguments we can also prove indefinability of the property $\sum(\rho(1), \ldots, \rho(n)) = 0$ in monadic second order logic. Secondly, since our construction of the expression depends only on the fact that the quantification over the structure $\mathfrak{A}$ is finite, the theorem also holds for any finite structure $\mathfrak{A}$, i.e. any finite signature not necessarily the signature $(\leq)$, and any logical formalism where the quantification over $\mathfrak{A}$ is finite – in particular higher order logics.

6 Conclusion

In this work we introduced a formalism of expressions that take inputs from an infinite domain. The expressions are shown to have equivalent expressions in a normal form which allows to prove indefinability results using regularity lemmas from combinatorics. We point out some interesting avenues for further exploration. Firstly in this paper we have placed no restriction on the size of the circuits. But it seems that some of the results in Section 3 point towards the possibility of finer indefinability results that take into account the size of the expressions. Secondly we have defined our domain $\mathcal{D}$ to be infinite. It is also an interesting to investigate expressive power of families of expressions when the data domain is unbounded yet finite, and grows asymptotically with the input size.

References

1 Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

2 Erich Grädel and Yuri Gurevich. Metafinite model theory. *Information and Computation*, 140(1):26 – 81, 1998.

3 R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*. A Wiley-Interscience publication. Wiley, 1990.

4 T. Jech. *Set Theory: The Third Millennium Edition, Revised and Expanded*. Springer Monographs in Mathematics. Springer, 2003.

5 Pascal Koiran. A weak version of the blum, shub, and smale model. *Journal of Computer and System Sciences*, 54(1):177 – 189, 1997.

6 Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1994.

7 Pascal Tesson. *Computational Complexity Questions Related to Finite Monoids and Semigroups*. PhD thesis, School of Computer Science, McGill University, Montreal, 2003.

8 Pascal Tesson. An application of the hales-jewett theorem to multiparty communication complexity. Extract from the PhD Thesis, 2004.

9 L. G. Valiant. Completeness classes in algebra. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.