Finite-Degree Predicates and Two-Variable First-Order Logic

Charles Paperman

University of Warsaw, Poland

Abstract

We consider two-variable first-order logic on finite words with a fixed number of quantifier alternations. We show that all languages with a neutral letter definable using the order and finite-degree predicates are also definable with the order predicate only. From this result we derive the separation of the alternation hierarchy of two-variable logic on this signature.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases First order logic, automata theory, semigroup, modular predicates

Digital Object Identifier 10.4230/LIPIcs.CSL.2015.616

1 Introduction

Finite model theory and the lower classes of circuit complexity are intricately interwoven. In the context of circuit complexity, logics are considered over finite words with *arbitrary numerical predicates*. Intuitively, we allow the use of any predicate that only depends on the size of the word. A first result from Immerman [6] provides an equivalence between languages definable by first-order logic enriched with arbitrary numerical predicates on the one hand, and languages computable by families of circuits of constant depth and polynomial size on the other. Since then, several meaningful circuit complexity classes have been shown to be equivalent to logical fragments [1, 8]. It is therefore possible to obtain deep and interesting inexpressibility results by using circuits lower bounds.

For instance, by using a famous lower bound for the *parity* language [5], Barrington, Compton, Straubing and Thérien [1] showed that the regular languages definable in first-order logic with arbitrary numerical predicates are definable with only the *regular predicates*. Relying on an algebraic description of first-order logic with regular predicates, it is possible to decide the definability of a regular language in this logic.

Conversely, it is tempting to use finite model theory methods to compute circuit lower bounds. This approach has achieved relative success for *uniform* versions of circuit complexity classes. For instance, Roy and Straubing provide a separation result for the long-standing question of the separation of **ACC** from $\mathbf{NC^1}$ in a highly uniform setting [18]. In these settings, this uniformity condition has two different interpretations:

1. In the circuit framework, it is a restriction on the complexity of the wiring of the gates.
2. In the logical framework, it is a restriction on the class of numerical predicates considered in the fragment.

In order to deal with the combinatorics of arbitrary numerical predicates, the languages with a *neutral letter* have been introduced in [2]. Formally, a language L has a neutral letter c if for any pair of words u, v, we have $ucv \in L$ if, and only if, $uv \in L$. Less formally, this letter c can be added or removed anywhere in a word without changing its membership to L. The underlying idea was that numerical predicates would be essentially useless in the presence of a neutral letter. This was made formal through the *Crane Beach conjecture*:

24th EACSL Annual Conference on Computer Science Logic (CSL 2015).
Editor: Stephan Kreutzer; pp. 616–630
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Every language with a neutral letter definable in first-order logic with arbitrary numerical predicates is definable in first-order logic with the linear order only.

Furthermore, some of the most interesting languages, such as the *parity* language, possess a neutral letter. Unfortunately, this conjecture has been disproved in the article [2] in the context of first-order logic, as long as the Bit predicate is in the signature. This result prevents the use of this approach to obtain circuit lower bounds for more expressive classes. However, for fragments of first-order logic the Crane Beach conjecture is still of interest. For instance, the Crane Beach conjecture holds for the fragment without quantifier alternation [2].

Turning to other fragments, two-variable first-order logic is a robust and well-studied class that offers a wide range of long-standing and intriguing open questions. It is not know whether the Crane Beach conjecture holds for this fragment. This question is related to a long-standing open linear lower bound for the addition function, since two-variable logic is equivalent to linear circuits of $\mathbf{AC}^0$ [8]. Therefore, if the Crane Beach conjecture holds, then the addition function is not computable by a constant-depth linear-size circuit family. This result would improve on a known lower bound for addition that states that addition is not computable by circuits of constant depth with a linear number of wires [3]. We remark that lower bound for addition has been discussed and informally mentioned several times [16, 4, 9, 8] and formally stated in the article [7, Open problem 23].

In this paper, we focus on the case of two-variable logic, which is poorly understood in this context. We first prove that languages with a neutral letter definable in two-variable logic with arbitrary numerical predicates can be defined allowing only the linear order and the following predicates:

1. The class $\mathcal{F}$ of finite-degree predicates, that is, binary predicates that are relations over integers and such that each vertex of their underlying infinite directed graph has a finite degree.
2. The predicate MSB_0 defined as follows. The predicate MSB_0 is true of x and y if the binary representation of y is obtained by zeroing the most significant bit of x. More formally
$$\mathrm{MSB}_0 = \{(x, x - 2^i) \mid x \in \mathbb{N},\ \text{and } i = \lfloor \log(x) \rfloor\}\ .$$

As an intermediate step toward a better comprehension of the Crane Beach conjecture for $\mathbf{FO}^2$, we propose to study the relationship between $<$ and $\mathcal{F}$, and present a Crane Beach result which is thus one predicate shy from showing the Crane Beach conjecture for $\mathbf{FO}^2$ over arbitrary numerical predicates.

The main result of this paper is a proof of the Crane Beach conjecture for each layer of the alternation hierarchy of the two-variable first-order logic equipped with the linear order and the finite-degree predicates.

Note that the general arbitrary numerical predicates in the statement would entail a long standing conjecture on the circuit complexity of the addition function. Thus, this result can be viewed as a uniform version of this circuit lower bound. This result immediately implies that this hierarchy is strict. This provides, to the best of our knowledge, the first example of a Crane Beach conjecture that applies to each level of an alternation hierarchy. Ramsey's Theorem for 3-hypergraphs will be our key combinatorics tool. This theorem indicates that the Crane Beach conjecture for $\mathbf{FO}^2$ hinges on the interaction between finite-degree predicates and the predicate MSB_0.

On the two-variable restriction. It is already known that the first-order logic with the "+" predicate satisfies the Crane Beach conjecture. Furthermore, the MSB_0 predicate is definable in first-order logic with the predicate "+" and the unary predicate $\{2^x \mid x \in \mathbb{N}\}$. The proof of the Crane Beach conjecture for "+" predicate can be augmented to handle this extra unary numerical predicate. Therefore, we deduce that the first-order logic with the order and the MSB_0 predicate also satisfies the Crane Beach conjecture.

The case of finite-degree predicates is more intricate. Indeed, even if this class of predicates satisfies a form of locality, it is still not known if the Crane Beach conjecture hold for $\mathbf{FO}[<, \mathcal{F}]$. This class contains numerous expressive numerical predicates as the *translated bit predicate* which is true in positions (x, y) if the $(y-x)^{\text{th}}$ bit of x is a one. The Crane Beach conjecture may holds for finite-degree predicates but the classical proof, e.g. *collapse on active domain*, seems to fail [2, 18, 11].

Organization of the paper. Section 2 is dedicated to the necessary definitions. In Section 3 we present an Ehrenfeucht-Fraïssé game adapted to our context. We present in Section 4 our main result with immediate corollaries. The final section is dedicated to the proof.

2 Definitions

A finite word $u = u_0 \cdots u_{n-1}$ of A^* is represented by a relational structure on the set $\{0, \cdots, n-1\}$ over the vocabulary consisting of the *letter predicates* $\{\mathbf{a} \mid a \in A\}$ and of the *numerical predicates*. On the one hand, the letter predicate $\mathbf{a}$ is interpreted as the subset of all the positions labelled by the letter a. On the other hand, a numerical predicate interpretation only depends on the size n of the input word. Therefore, an interpretation of the predicate symbol $\mathbf{P}$ of arity k is a sequence $P = (P_n)_n$, where $P_n \subseteq \{0, \ldots, n-1\}^k$. Note that $\mathbf{P}$ is a syntactic object, while P is its interpretation. Furthermore a numerical predicate is said to be *uniform* if it can be seen as a relation on integers. More precisely, a numerical predicate $P = (P_n)_n$ of arity k is uniform if there exists an integer relation $Q \subseteq \mathbb{N}^k$ satisfying $Q \cap \{0, \ldots, n-1\}^k = P_n$. From now on, we do not distinguish numerical predicates from their interpretation and uniform predicates are seen as relations on integers. The class of all numerical predicates is denoted by Arb. Remark that the word *uniform* in this context is not related to the classical notion of *uniformity* in circuit complexity.

Examples

- The classical predicates $x < y$ or $x + y = z$ and $xy = z$ are numerical predicates and are uniforms.
- The predicate $x + y = \max$, where max is the last position of the word, is not uniform.

The logical formulae we consider are the first-order formulae over finite words. They are obtained with the following grammar:

$$\varphi = \mathbf{a}(x) \mid \mathbf{P}(x_1, \ldots, x_k) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists x\ \varphi\ \ .$$

Here $x, x_1, x_2, x_3, \ldots$ denote first-order variables, which are interpreted by positions in the word. The letter predicate $\mathbf{a}(x)$, is interpreted by "the letter in position x is an a," and $\mathbf{P}(x_1, \ldots, x_k)$, is interpreted by "the predicate P is true on $(x_1, \ldots, x_k)$." As usual, the Boolean connectives $\wedge$ and $\neg$ are interpreted by "and" and "not," respectively, and $\exists x$ as a first-order existential quantification. We use the standard notation $u \models \varphi$ to signify that the word u satisfies the formula φ. We also denote by $u \models \varphi(i)$ if the formula $\varphi(x)$ is *true* when its free variable is interpreted by the integer $i < |u|$. The *quantifier depth* of a formula is the maximal number of nested quantifiers.

Let $\mathcal{P}$ be a class of numerical predicates. We denote by $\mathbf{FO}[\mathcal{P}]$ the class of first-order formulae that use numerical predicates in $\mathcal{P}$. We also denote by $\mathbf{FO}^2[\mathcal{P}]$ the subclass of formulae of $\mathbf{FO}[\mathcal{P}]$ that use only two variables but allows the reuse of them. We say that a language L is definable in a fragment of logic if there exists a formula in this fragment such that L is the language of words satisfying this formula.

Example. The language $A^*aA^*bA^*cA^*$ can be described by the first-order formula

$$\exists x \; \exists y \; \exists z \; x < y < z \wedge \mathbf{a}(x) \wedge \mathbf{b}(y) \wedge \mathbf{c}(z) \ .$$

This formula uses three variables x, y and z. However, by reusing x we can rearrange it so that it uses two variables:

$$\exists x \; \mathbf{a}(x) \wedge \Big(\exists y \; x < y \wedge \mathbf{b}(y) \wedge \big(\exists x \; y < x \wedge \mathbf{c}(x)\big)\Big) \tag{1}$$

The alternation hierarchy of $\mathbf{FO}^2$ is also of interest here. To define formally the number of alternations of a formula, it is not possible to use prenex canonical normal form obtained by applying DeMorgan's laws to move negations past conjunctions, disjunctions and quantifiers. Indeed, these constructions increase the number of variables. That said, the number of alternations is still a relevant parameter that could be defined as follows: Consider the tree naturally associated to a formula, as the grammar previously exposed. For instance, formula (1) has "$\exists$" as a root and the atomic formulae as the leaf. In a two-variable first-order formula we count the maximal number of alternations between the root and the leaves once the negations have been pushed on to the leaves. A more precise definition could be found in the article [19]. We denote by $\mathbf{FO}^2_k[\mathcal{P}]$ the formulae of $\mathbf{FO}^2[\mathcal{P}]$ that have at most k quantifier alternations. The hierarchy induced by $\mathbf{FO}^2_k[<]$ is known to be strict [19] and its membership problems is decidable [12, 14]. Without loss of generality, we will always consider two-variable logic over predicates of arity at most 2.

3 Ehrenfeucht-Fraïssé game

One of the important tools for proving our main result is the Ehrenfeucht-Fraïssé game for two-variable logic. It is often used in the context of finite model theory to show certain inexpressibility results. Libkin's book [15] provides a good exposition. In this section, we present the Ehrenfeucht-Fraïssé game and briefly sketch a proof that the Crane Beach conjecture holds for $\mathbf{FO}^2_m[<, +1]$. This could be easily proved by using some algebraic descriptions of $\mathbf{FO}^2_m[<, +1]$ obtained by Kufleitner and Lauser [13] but we prove it using Ehrenfeucht-Fraïssé game as an introduction to our general result.

In the context of two-variable logic with a bounded number of alternations m and quantifier depth s, the associated Ehrenfeucht-Fraïssé game is defined as follows:

- The game is played by two players: *Spoiler* and *Duplicator*, on two relational structures. In our case, the relational structures are associated with the words u and v equipped with the letter predicates and a finite number of numerical predicates.
- The first round starts with Spoiler, who chooses either u or v and plays by putting a pebble on a position. Then Duplicator chooses the other word and puts a pebble on one of its positions.
- The subsequent rounds proceed as follows: each word is labelled by at most two pebbles. First, the two oldest pebbles are removed. Then, Spoiler plays on one structure and Duplicator on the other. If the relational structures induced by the two pairs of pebbles are not isomorphic, Spoiler wins.
- During all the game, Spoiler can change at most m times between the two words. Duplicator wins the game if he did not loose the game before the end of the s^{th} round.

We say that Spoiler has a *winning strategy* if he has a strategy that allows him to win the game whatever Duplicator plays. The following theorem is a well-known result that could be easily adapted, for instance, from the book [15].

▶ **Theorem 1.** *A language L belongs to $\mathbf{FO}^2_m[\mathcal{P}]$ if and only if there exist predicates $P^1, \ldots, P^t \in \mathcal{P}$ and $s \in \mathbb{N}$ so that for any words $(u, v) \in L \times L^c$ Spoiler has a winning strategy for the two-pebble game with s rounds and m alternations on (u, v) over the predicates $P^1, \ldots, P^t$.*

This theorem is our main interface to logic in order to establish Crane Beach-like results. The proof method we are going to sketch is a rather classical *back-and-forth* construction. As we mention before, the next result is also a direct consequence of known algebraic characterisations of these fragments [13].

▶ **Proposition 2.** *For any m, languages with a neutral letter in $\mathbf{FO}^2_m[<, +1]$ are definable in $\mathbf{FO}^2_m[<]$.*

Sketch of proof. Let L be a language definable in $\mathbf{FO}^2_m[<, +1]$ and assume that it has a neutral letter c. Thanks to Theorem 1, there exist integers s and $k \leqslant m$ such that Spoiler has a winning strategy for the two-pebble game with s rounds and k alternations on (u, v), with $(u, v) \in L \times L^c$. We construct two words u' and v' by inserting $2s$ letters c between each position (including the beginning and the end of the words). As c is a neutral letter, we have $(u', v') \in L \times L^c$ and therefore Spoiler has a winning strategy for the two-pebble game with s rounds and k alternations. Remark that the successor relation on (u', v') is useless since the non-neutral letters are not reachable from each other in less that s rounds. Therefore one can translate the Spoiler's wining strategy on (u', v') on a wining strategy that does not use the successor relation. This wining strategy can then be translated in a wining strategy on (u, v). We then conclude thank to Theorem 1. ◀

4 Main Result

We now investigate the Crane Beach conjecture in the specific case of $\mathbf{FO}^2$ equipped with numerical predicates of finite degree. Throughout this section, we borrow from the vocabulary of graph theory in order to express properties on the structure of numerical predicates. Indeed, a binary numerical predicate can be understood as a family of graphs. Furthermore, if the predicate is uniform, it can be viewed as a single infinite graph where the set of vertices is $\mathbb{N}$. Let P be a uniform numerical predicate. The *degree* of a position k for P, denoted by $d_P(k)$, is the size of the set of all integers connected to k *via* P. More formally

$$d_P(k) = |\,\{j \mid (k, j) \in P \text{ or } (j, k) \in P\}\,| \ .$$

The notion of locality is one of the most effective tools for using the Ehrenfeucht-Fraïssé games. One way of introducing locality is to restrict the degree of the signature. A uniform binary predicate P has a *finite degree* if all positions have a finite degree. We denote by $\mathcal{F}$ the class of binary uniform finite-degree predicates.

Examples

- The predicate $kx = y$, $x^k = y, \ldots$ as well as the graph of any strictly growing function.
- The translated Bit predicate which is true in (x, y) if the $(y - x)^{\text{th}}$ bit of x is a one.

Example of nonfinite-degree predicates

- The linear ordering.
- The Bit predicate which is true of (x, y) if the y^{th} bit of x is a one.
- The MSB_0 predicate.

Predicates of finite degree do not include by definition uniform monadic predicates. However, all uniform monadic predicates can be encoded as predicates of finite degree. If P is monadic and uniform then $Q = \{(x, x) \mid x \in P\}$ is a finite-degree predicate.

The next theorem states that the Crane Beach conjecture for $\mathbf{FO}^2[\text{Arb}]$ reduces to solving the Crane Beach conjecture for the order, the MSB_0 predicate and the class of finite-degree predicates. The proof of this theorem is an adaptation of a circuit-version of a similar result [10]. Because of the lack of space, the proof of this theorem is omitted.

▶ **Theorem 3.** *Any language with neutral letter definable in* $\mathbf{FO}^2[\text{Arb}]$ *is definable in* $\mathbf{FO}^2[<, \mathcal{F}, \text{MSB}_0]$.

We believe that this last theorem does not hold without the neutral-letter hypothesis. For instance, the language $\{u\overline{u} \mid u \in A^*\}$, where $\overline{u}$ is the reversal image of u, is definable in $\mathbf{FO}^2[x + y = \max]$ but we conjecture that it is not definable by using only uniform predicates, and in particular, using predicates in the signature $[<, \mathcal{F}, \text{MSB}_0]$.

We now focus on the signature $[<, \mathcal{F}]$. To solve this problem, we will use the *locality* of the class $\mathcal{F}$. Locality is an effective tool which allows us to obtain numerous results of non-definability with the help of the Ehrenfeucht-Fraïssé games. Unfortunately, as soon as the order is present in the signature, it is no longer possible to use locality results and the absence of the order makes the fragment far less expressive. We are going to show that it is possible to add the order whilst conserving a form of locality when the other predicates are of finite degree.

▶ **Theorem 4** (Main Theorem)**.** *Let* $m \geqslant 0$. *Any language with a neutral letter definable in* $\mathbf{FO}^2_m[<, \mathcal{F}]$ *is definable in* $\mathbf{FO}^2_m[<]$.

We immediately obtain the following corollary.

▶ **Corollary 5.** *Any language with a neutral letter definable in* $\mathbf{FO}^2[<, \mathcal{F}]$ *is definable in* $\mathbf{FO}^2[<]$.

This theorem states the uselessness of finite-degree predicates for defining languages with a neutral letter in two-variable logic. More precisely, they do not even improve the logical complexity of the languages. Therefore, we immediately deduce the strictness of this hierarchy. Indeed, we mainly use the known facts that $\mathbf{FO}^2_m[<]$ is a strict hierarchy (see [19]) and that each layer is stable by inverse image of morphisms. This latter fact is a requirement for having an equational description as given in the article [12]. Then, it is sufficient to take the inverse image of a language L that separates $\mathbf{FO}^2_{m+1}[<]$ from $\mathbf{FO}^2_m[<]$ by a morphism that maps a letter which is not in the alphabet of L to the empty word.

5 Proof of the main theorem

The principal ingredients are a notion of *locality*, the Ehrenfeucht-Fraïssé games and Ramsey's Theorem. For the remaining of the proof we fix $P^1, \ldots, P^t$ as predicates in $\mathcal{F}$. Our objective is to prove that for any language L with a neutral letter definable in $\mathbf{FO}^2_m[<, P^1, \ldots, P^t]$, there exists s such that for every words $u \in L$ and $v \notin L$, Spoiler has a winning strategy for the Ehrenfeucht-Fraïssé game with two pebbles, s rounds and m alternations on (u, v) and over the signature $\{<, +1\}$. The proof is decomposed as follows.

1. First, we introduce the notion of a *position's neighbourhood*.
2. Then, we define an equivalence relation between triples of disjoint neighbourhoods, which will allow us to define the different roles that these triples could play throughout the course of the game.
3. We then extract triples of so-called *well-typed* positions, with the help of Ramsey's Theorem for 3-hypergraphs.
4. Finally, we will inductively construct a winning strategy for Spoiler over the signature $\{<, +1\}$ that uses at most s rounds and m alternations. Proposition 2 allows us to conclude.

Let $E \subseteq \mathbb{N}^2$ be defined by $\{x, y\} \in E$ if, and only if, x and y are two positions connected by one of the predicates. More precisely, $\{x, y\} \in E$ if and only if

$$P^1(x, y) \vee P^1(y, x) \vee \cdots \vee P^t(x, y) \vee P^t(y, x) \ .$$

The graph $(\mathbb{N}, E)$ is the graph behind our reasoning. As each predicate is of finite degree, the graph $(\mathbb{N}, E)$ is also of finite degree. From this point on, we assume that the integer s (the number of rounds in the game) is fixed.

5.1 Definition of neighbourhood

For an integer i, the usual notion of r-neighbourhood is defined as the set of integers at distance r from i in $(\mathbb{N}, E)$. It captures the intuition that two integers with similar r-neighbourhoods cannot be distinguished in r applications of the predicates. Adding linear order to the predicates, any element between two given integers is *connected* by the order. Our specialized notion of neighbourhood thus distinguishes between the linear order and the other predicates; to this end, let us first introduce the *closure* of a finite set $F \subseteq \mathbb{N}$ as:

$$\mathrm{Cl}(F) = \{\min F, \min F + 1, \ldots, \max F\} \ .$$

Then, intuitively combining at each step the use of the predicates and that of the order, we define the 0-*neighbourhood* of $i \in \mathbb{N}$ as:

$$V(i, 0) = \mathrm{Cl}(\{i\} \cup \bigcup_{\substack{k' \leqslant i \leqslant k \\ \{k', k\} \in E}} \{k', k\}) \ .$$

and, inductively, the $(r+1)$-*neighbourhood* of $i \in \mathbb{N}$ as:

$$V(i, r+1) = \mathrm{Cl}(\bigcup_{j \in V(i,0)} V(j, r)) \ .$$

Less formally, the 0-neighbourhood of i is the set of positions j such that by moving a pebble inside this set it is possible to jump over i. We obtain immediatly that $V(i, r) \subseteq V(i, r+1)$.

▶ **Lemma 6.** *For all integers i and k, $V(i, k)$ is finite.*

We now define the function $g_s : \mathbb{N} \to \mathbb{N}$ by $g_s(i) = \min V(i, s)$.

▶ **Lemma 7.** *We have $\lim_i g_s(i) = +\infty$.*

From this we immediately deduce the following corollary, which establishes the possibility of obtaining an arbitrarily large number of neighbourhoods that do not overlap.

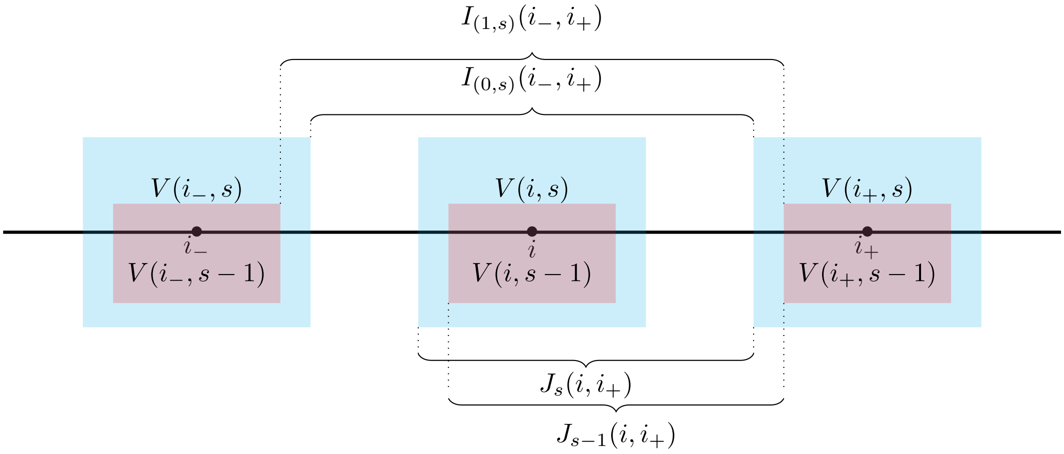


Figure 1 Neighbourhoods and segments.

▶ **Corollary 8.** *For any integer p, there exists $X \subseteq \mathbb{N}$ of size p such that for any $i, j \in X$, the s-neighbourhood of i and j are disjoint and separated by at least one integer.*

An *s-extraction* is a set of integers, such that their s-neighbourhoods are disjoint and separated by at least one integer. In short, they must be in accordance with the conditions of Corollary 8.

5.2 An equivalence relation for triples

We now introduce a notion of *similarity* for the triples of neighbourhoods taken from the Ehrenfeucht-Fraïssé two-pebble game. Let (i_-, i, i_+) be a triple of integers which is an s-extraction. More precisely, this triple satisfies that

1. $i_- < i < i_+$,
2. their s-neighbourhoods are disjoint and have at least one element between them.

According to Corollary 8, such a triple exists. We set $J_s(i, i_+)$ as the interval between the minimal position of the s-neighbourhood of i and minimal position of the s-neighbourhood of i_+. More formally,

$$J_s(i, i_+) = \{\min V(i, s), \ldots, \min V(i_+, s) - 1\} \ .$$

We also set $I_{(r,s)}(i_-, i_+)$ the interval in-between the maximal position of the $(s-r)$-neighbourhood of i_- and the minimal position of the $(s-r)$-neighbourhood of k. More formally

$$I_{(r,s)}(i_-, i_+) = \{\max V(i_-, s-r) + 1, \ldots, \min V(i_+, s-r) - 1\} \ .$$

These notations are illustrated in Figure 1.

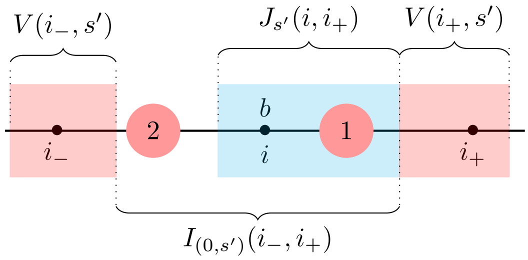
Let us take two triples (i_-, i, i_+) and (j_-, j, j_+) which form two s-extractions with $i_- < i < i_+$ and $j_- < j < j_+$. These two triples of integers are *equivalent* if two two-pebble *constrained games* are similar. We define two different notions of *constrained games* that differ only in their starting sets. These games only use two pebbles which are confined, at the r^{th} round, to the intervals

$$I_{(r,s)}(i_-, i_+) \text{ and } I_{(r,s)}(j_-, j_+) \ .$$

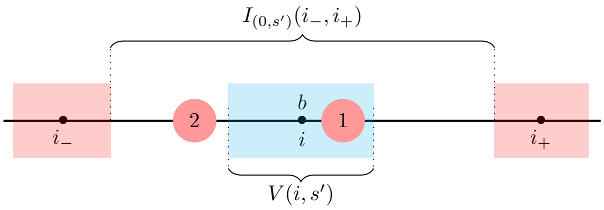
For the first game, the first pebble must be placed for both Spoiler and Duplicator in the sets $J_s(i, i_+)$ and $J_s(j, j_+)$. For the second game the first pebble is placed by Spoiler and Duplicator in the sets $V(i, s)$ and $V(j, s)$. If Duplicator wins these two games we can state that these two triples are equivalent, which we denote as $(i_-, i, i_+) \sim_s (j_-, j, j_+)$.

We now introduce formally this definition. We say that $(i_-, i, i_+) \sim_s (j_-, j, j_+)$ if for all $s' \leqslant s$ Duplicator wins the two following games. They are two-pebble games with s' rounds and s' alternation (we consider that Spoiler may alternate as much as he wishes between the two words) on the signature $\{<, P^1, \ldots, P^t\}$, and all positions are labelled by the same letter a with the exception of positions i and j which are labelled by the same letter b distinct from a. Here is the formal description of the two games:

1. For the first game, the first pebble of Spoiler and the first pebble of Duplicator are constrained to the set $J_{s'}(i, i_+)$ and $J_{s'}(j, j_+)$. At the r^{th} round, the players are constrained to choose positions in the sets $I_{(r,s')}(i_-, i_+)$ and $I_{(r,s')}(j_-, j_+)$.



2. For the second game, the Spoiler's first pebble and the first pebble of Duplicator are constrained to $V(i, s')$ and $V(j, s')$. At the r^{th} round, the players are constrained to play in the sets $I_{(r,s')}(i_-, i_+)$ and $I_{(r,s')}(j_-, j_+)$.



We say that positions $x \in I_{(r,s')}(i_-, i_+)$ and $y \in I_{(r,s')}(j_-, j_+)$ are *locally equivalent* if Duplicator can win the two restricted games when the pebbles are at these positions. The property presented in the following lemma can be deduced from the definitions and will be useful later.

▶ **Lemma 9.** *Let (i_-, i, i_+) an s-extraction. For every $0 \leqslant r \leqslant s$, we have the following*

$$J_{s-r}(i_-, i) \cup J_{s-r}(i, i_+) = V(i_-, s-r) \cup I_{(r,s)}(i_-, i_+) \ .$$

We now prove that $\sim_s$ is a finite-index equivalence relation. This is a rather classical result for this type of object in finite model theory. We remark that the equivalent classes can be seen as the sets of true formulae for each triple in a logic adapted to the two restricted games. Thus, two triples would be equivalent if they satisfy the same formulae of quantifier depth less than s. As the number of formulae is finite, we can easily deduce that $\sim_s$ equivalence relation.

▶ **Lemma 10.** *The relation $\sim_s$ is an equivalence relation of finite index.*

Ramsey's Theorem is a combinatorial result of graph theory often used in finite model theory. Here we use a version adapted to *hypergraphs*. We introduce it in the context of triples, which is a direct reformulation of the 3-*hypergraphs* variant. This theorem establishes that for every large hypergraph with coloured edges, it is possible to extract a sufficiently large monochrome sub-hypergraph. This theorem allows us to find an arbitrarily large set of triples which are all pairwise equivalent for the $\sim_s$ relation. For a set E, we denote by $\mathcal{P}_3(E)$ the set of pairwise disjoint triples of E.

▶ **Theorem 11** (Ramsey's Theorem for 3-hypergraphs [17]). *Let c be an integer. For any integer p there exists an integer n such that for any set S of size n and any function $h\colon \mathcal{P}_3(S) \to \{1, \ldots, c\}$ there exists a set $F \subseteq S$ of size p such that h is constant on $\mathcal{P}_3(F)$.*

A well-typed s-extraction is a set X that is an s-extraction and such that all the triples of X are equivalent for $\sim_s$. The following corollary is an immediate from Ramsey's Theorem, in which c is the number of s-types of triples and h is the function that associates triple with their s-type.

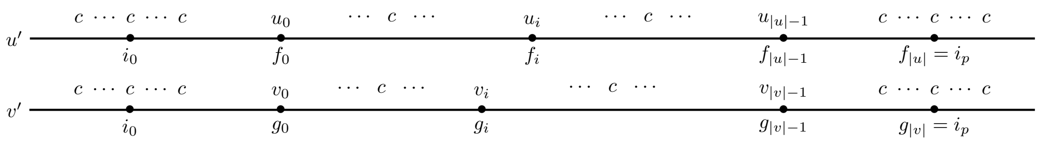
▶ **Corollary 12.** *For all integers p there exists a well-typed s-extraction of size p.*

We have now presented all of the tools necessary to present a proof of Theorem 4.

5.3 Core of the proof

Let L be a language with c as a neutral letter and definable in $\mathbf{FO}^2_m[<, P^1, \ldots, P^t]$. According to Theorem 1, there exists an integer s, such that for any words $(u, v) \in L \times L^c$, Spoiler has a winning strategy for the two-pebble game with s rounds and m alternations for the signature $\{<, P^1, \ldots, P^t\}$. Let (u, v) in $L \times L^c$ be such a pair. We now construct a strategy for Spoiler using only the order and the successor. Let $p = \max(|u|, |v|) + 1$. According to Corollary 12, there exists $X = \{i_0 < i_1 \cdots < i_p\}$, which is a well-typed s-extraction. Let $n = \max V(i_p, s)$, and let u' and v' be two words of length n and $(f_i)_{0 \leqslant i < |u|}, (g_i)_{0 \leqslant i < |v|}$ such that:

- $i_0 < f_0 < f_1 < \cdots < f_{|u|-1} < f_{|u|} = i_p$, and $i_0 < g_0 < g_1 < \cdots < g_{|v|-1} < g_{|v|} = i_p$,
- for all integers i, the positions f_i and g_i belong to X,
- $u'_{f_i} = u_i$, $v'_{g_i} = v_i$, $f_0 = g_0$ and $f_{|u|-1} = g_{|v|-1}$,
- all unassigned positions of u' and v' are labelled by the letter c.



If the words u and v are not of the same size, then that could give us $f_i \neq g_i$. The words u' and v' are nothing other than the words u and v after inserting neutral letters such that the non-neutral letters are on X. We also require the first and last non-neutral letters to be in the exact same positions.

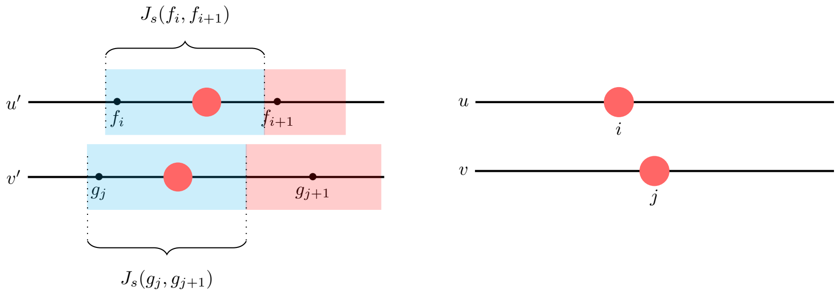
As c is a neutral letter, (u', v') is in $L \times L^c$. Therefore, Spoiler has a winning strategy for the two-pebble game over s-round and m-alternation and the signature $\{<, P^1, \ldots, P^t\}$. We now have to construct Spoiler's new strategy on (u, v). In order to do so, we simulate the game on (u', v') and construct *via* induction a winning strategy for Spoiler on (u, v). To achieve this step, we exploit a back-and-forth mechanism between the game on (u', v') and the game on (u, v). By following his winning strategy, Spoiler chooses a position on (u', v') which we translate into a position in (u, v). Duplicator then chooses a position in (u, v) which we translate on a position in (u', v'). We repeat this process until Duplicator can no longer respond in (u', v'). We must force Spoiler to play moves that are distant from one another so that his choices in (u', v') lead to a winning strategy on (u, v). If Spoiler's new pebble is in a neighbourhood different to that of the previous pebble, then by construction of the neighbourhoods, the numerical predicates, with the exception of the order predicate, do not allow for a connection between the two positions; they do not transmit *information*. In the following section we always denote by i_r (resp. j_r) the position of the pebble played at the round r on u (resp. v). Likewise, we use i'_r (resp. j'_r) for the position of the pebble at the round r on u' (resp. v').

For this construction to work, Spoiler should not win the game on (u', v') before he wins it on (u, v). This could however happen if Duplicator's choices on (u', v') are not pertinent. We avoid this situation by selecting locally equivalent positions, that is, positions where Duplicator wins the restricted games introduced in the preceding section. Thus, Spoiler cannot win by choosing moves that are close to the old pebbles. He is therefore forced to play some distant moves.

When Spoiler plays on an *extremal position* of the game on (u', v'), Duplicator can always respond at the same position on the other word. These moves therefore are of no interest in Spoiler's strategy. They are not used in the construction of the strategy of the game on (u, v). Each time Spoiler makes such a move, the game on (u, v) does not progress. More specifically, if the game has not started, the pebbles are not even placed and if the pebbles are already placed, they are not moved.

We begin by describing the game's first round, then we inductively build a strategy for the following rounds. For the first move, Spoiler's winning strategy designates a position for the game on (u', v'). Through symmetry, we assume that this is a position on u'. We therefore distinguish two cases:

1. This first move occurs within a segment of the form $J_s(f_i, f_{i+1})$ for an integer $0 \leqslant i < |u|$. In this case, we choose to play on the position i on the game on (u, v). Duplicator then responds in the game on (u, v) by playing on v at a position j. If the letter that marks j is different from the one that marks i, Duplicator loses the game immediately. Otherwise, we have to simulate Duplicator's response in the game on (u', v') by choosing a position in $J_s(g_j, g_{j+1})$ that is locally equivalent to Spoiler's first pebble. This is possible as the letters that mark f_i on u' and g_j on v' are equal, and $(f_{i-1}, f_i, f_{i+1}) \sim_s (g_{j-1}, g_j, g_{j+1})$.



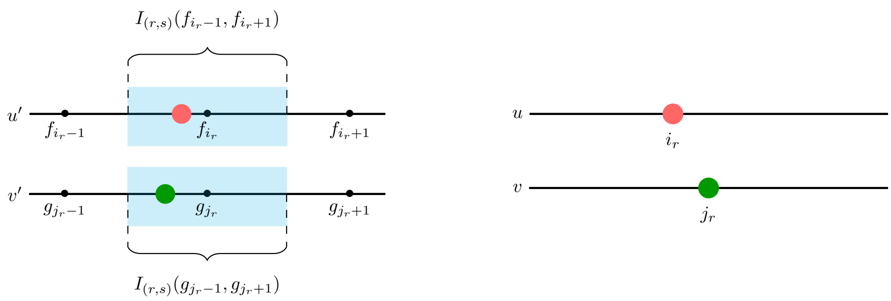
2. This first move is on an extremal position, that is smaller than $\min J_s(f_0, f_1) = \min J_s(g_0, g_1)$ or bigger than $\max J_s(f_{|u|-1}, f_{|u|}) = \max J_s(g_{|v|-1}, g_{|v|})$. In this case, the back-and-forth process is degenerate since the game on (u, v) has not started yet. It starts when Spoiler plays on a non-extremal position.
This kind of moves is not useful for Spoiler since Duplicator can only answer on the game on (u', v') by choosing the exact same position on the other word. As long as Spoiler plays on these extremal positions, it is sufficient for Duplicator to choose the exact same position. As Spoiler follows a winning strategy, he eventually plays inside a segment $J_s(f_i, f_{i+1})$ for some integers $0 \leqslant i < |u|$. Indeed, the extremal positions together with

segments $J_s(f_i, f_{i+1})$ split into a partition of all positions of the word (see Figure 1). Therefore, we can assume to be in the preceding case.

We now explain how to construct a winning strategy for Spoiler on (u, v) for the next rounds. We construct it inductively. We now assume to have played $1 \leqslant r < s$ rounds and that the pebbles of the preceding round are on positions i_r on u (resp. j_r on v) as well as i'_r on u' (resp. j'_r on v'). It is Spoiler's turn to play. By induction, we assume the following properties to be satisfied:

- If positions i'_r and j'_r belong to $I_{(r,s)}(f_{i_r-1}, f_{i_r+1})$ and to $I_{(r,s)}(g_{j_r-1}, g_{j_r+1})$ then they are locally equivalent for at least one of the two constrained games at $(s-r)$-rounds (see Figure 2). The first constrained game corresponds to the second case, and the second constrained game corresponds to the third case.
- If this latter condition is not satisfied, then both pebbles have the exact same value, which is an extremal position on (u', v'). More precisely, $i'_r = j'_r$ and either

$$i'_r < \min J_{s-r}(f_0, f_1) = \min J_{s-r}(g_0, g_1) \text{ or}$$
$$i'_r > \max J_{s-r}(f_{|u|-1}, f_{|u|}) = \max J_{s-r}(g_{|v|-1}, g_{|v|}) \ .$$



- We assume the configuration of the game on (u', v') to be winnable for Spoiler: he has a winning strategy in less than $(s-r)$ rounds.

We are going to distinguish two cases. Either Duplicator is going to answer on Spoiler's latest move in the game on (u, v) or Spoiler wins the game. Since we seek a winning strategy for Spoiler, we assume that Duplicator successfully answers on (u, v). If this is true, then we are going to find an adequate answer for Duplicator in the game on (u', v'). Since Spoiler has a winning strategy for this latter game, Duplicator eventually loses the game on (u', v') and therefore the game on (u, v). We remark that the number of alternations of the new Spoiler's winning strategy on (u, v) is at most the one of his strategy on (u', v'). This concludes the proof.

Nevertheless, it remains to be explained how we construct the position of Spoiler on (u, v) and how to deduce from a correct answer for Duplicator on (u, v), a correct answer for Duplicator on (u', v').

We use the Spoiler's winning strategy on (u', v') to construct a new move for Spoiler on (u, v). Without loss of generality, we assume that this move is on u' and we denote by i'_{r+1} its position. We now distinguish four cases that only depend on the value of i'_{r+1} (see Figure 2). Indeed, the segment $\{0, \ldots, n-1\}$ is split into four parts that correspond to the four following cases:

1. The first case corresponds to segments of the form $J_{s-r-1}(f_k, f_{k+1})$ for $k \neq i_r$ and $k \neq i_{r-1}$. It includes almost all the positions of $\{0, \ldots, n\}$ except extremal positions and a *hole* around positions i'_r and i'_{r-1}.

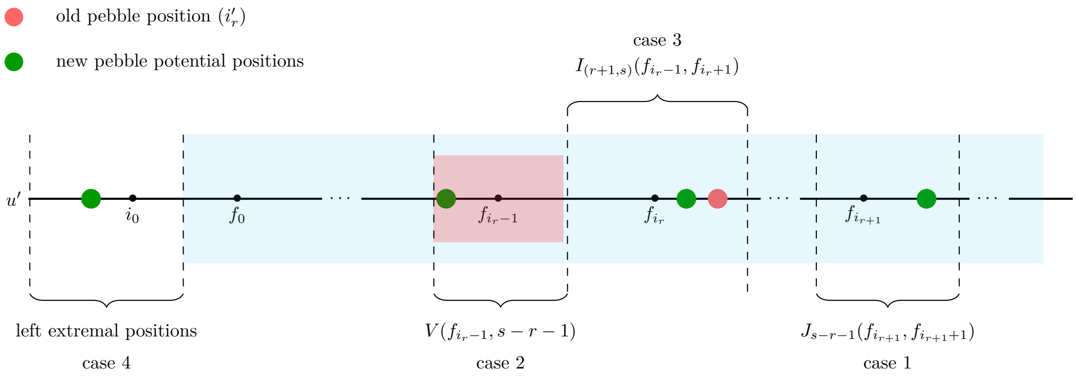


Figure 2 The four cases to deal with.

2. The second case corresponds to the *truncated* segment to the left of the previous pebble on u'. This is the initial segment of the second constrained game for this position. More precisely it is the segment $V(f_{i_r-1}, s-r-1)$.
3. The third case corresponds to the allowed positions for the constrained game around i'_r. More precisely, it is the segment $I_{(r+1,s)}(f_{i_r-1}, f_{i_r+1})$.
4. The last case corresponds to the extremal positions. They are the positions that are not handled by the other cases. They are either at the beginning or at the end of the word.

The four cases deal with all the positions since the segments of the form $J_{s-r-1}(f_k, f_{k+1})$ and the extremal positions form a partition of all the positions. Furthermore, by Lemma 9, we have

$$J_{s-r-1}(f_{i_r-1}, f_{i_r}) \cup J_{s-r-1}(f_{i_r}, f_{i_r+1}) = V(f_{i_r-1}, s-r-1) \cup I_{(r+1,s)}(f_{i_r-1}, f_{i_r+1}) \ .$$

We now construct the back-and-forth strategy for each of the four cases:

1. There exists an integer k different from i_r and $i_r - 1$ such that the position i'_{r+1} belongs to $J_{s-r-1}(f_k, f_{k+1})$. It is then sufficient for Spoiler to choose $i_{r+1} = k$ on u as its next move for the game on (u, v). We remark that all the predicates other than the linear order are evaluated to *false* between i'_r and i'_{r+1}. We assume Duplicator to be able to answer correctly at a position j_{r+1}. We now choose a position j'_{r+1} on v' in the set $J_{s-r-1}(g_{j_{r+1}}, g_{j_{r+1}} + 1)$ such that positions i'_{r+1} and j'_{r+1} are locally equivalent for the first constrained game. This is possible since positions $f_{i_{r+1}}$ and $g_{j_{r+1}}$ are labelled by the same letter and because

$$(f_{i_{r+1}-1}, f_{i_{r+1}}, f_{i_{r+1}+1}) \sim_s (g_{j_{r+1}-1}, g_{j_{r+1}}, g_{j_{r+1}+1}) \ .$$

We remark that all predicates except for the linear order are evaluated as *false* between j'_r and j'_{r+1}. Furthermore, the value of the order predicate between i'_r and i'_{r+1} is exactly the same as between i_r and i_{r+1} which is also the same as between j_r and j_{r+1} and between j'_r and j'_{r+1}. Since the letters labelling positions i_{r+1} on u and j_{r+1} on v are the same, we deduce that position j'_{r+1} is correct for Duplicator. Consequently, the new configuration satisfies the induction hypothesis.
2. We assume that i'_{r+1} belongs to $V(f_{i_r-1}, s-r-1)$. In this case, we choose $i_{r+1} = i_r - 1$, meaning that Spoiler plays on the position just to the left of i_r. Since the successor relation is in the signature, Duplicator is also forced to play at the position immediately to the left. Here the very same arguments that in case 1 allow us to build a position j'_{r+1}

so that the new configuration satisfies the induction hypothesis hold. The only difference, is that this time we are using the second constrained game, not the first.

3. If i'_{r+1} belongs to $I_{(r+1,s)}(f_{i_r-1}, f_{i_r+1})$, then according to the induction hypothesis, Duplicator has a position j'_{r+1} in the set $I_{(r+1,s)}(g_{j_r-1}, g_{j_r+1})$ which is locally equivalent to i'_{r+1}. By choosing this position and by setting $i_{r+1} = i_r$ and $j_{r+1} = j_r$, we obtain a new configuration that satisfies the induction hypothesis. We remark that in this case, the game configuration on (u, v) does not change.
4. The last case is the one which i'_{r+1} does not satisfy any of the preceding case. By construction, the positions of the words are split into segments $J_{s-r}(f_k, f_{k+1})$ (resp. $J_{s-r}(g_k, g_{k+1})$) and the extremal positions. Therefore, if the integer i_{r+1} is not treated by the other cases, then this position has to be extremal. That is to say

$$i'_{r+1} < \min J_{s-r-1}(f_0, f_1) = \min J_{s-r-1}(g_0, g_1)$$

or

$$i'_{r+1} > \max J_{s-r-1}(f_{|u|-1}, f_{|u|}) = \max J_{s-r-1}(g_{|v|-1}, g_{|v|}) \ .$$

We choose $j'_{r+1} = i'_{r+1}$ for Duplicator on v', as well as $i_{r+1} = i_r$ and $j_{r+1} = j_r$. Therefore the game on (u, v) does not evolve and the new configuration satisfies the induction hypothesis. We remark that it is possible for i'_{r+1} to be an extremal position but be handled by one of the preceding cases. For instance, if i_r belongs to $J_{s-r}(f_0, f_1)$ and if

$$i_{r+1} \in I_{(r+1,s)}(i_0, f_1) \cap \{0, \ldots, \min J_{s-r-1}(f_0, f_1)\} \ ,$$

then Duplicator follows the first constrained game and it is therefore possible that $i_{r+1} \neq j_{r+1}$. In this particular case, since i'_{r+1} and j'_{r+1} are locally equivalent, the configuration still satisfies the induction hypothesis.

As all the cases are treated, we have proved that as long as Duplicator answers correctly on (u, v), it is possible for him to answer correctly on (u', v'). Since Spoiler follows a winning strategy on (u', v'), Duplicator will eventually not be able to answer on (u, v). This concludes the proof.

Acknowledgement. I am grateful to Olivier Carton for all of his help and support, without which this paper would not have been possible. I also thank Michaël Cadilhac, Amy Hadfield, and the anonymous reviewers for their contribution in improving a lot the final version of this paper.

References

1 David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *J. Comput. System Sci.*, 44(3):478–499, 1992.

2 David A. Mix Barrington, Neil Immerman, Clemens Lautemann, Nicole Schweikardt, and Denis Thérien. First-order expressibility of languages with neutral letters or: The Crane Beach conjecture. *J. Comput. System Sci.*, 70(2):101–127, 2005.

3 AshokK. Chandra, Steven Fortune, and Richard Lipton. Lower bounds for constant depth circuits for prefix problems. In Josep Diaz, editor, *Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 109–117. Springer Berlin Heidelberg, 1983.

4 Shiva Chaudhuri and Jaikumar Radhakrishnan. Deterministic restrictions in circuit complexity. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 30–36. ACM, New York, 1996.

5 Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

6 Neil Immerman. Languages that capture complexity classes. *SIAM J. Comput.*, 16(4):760–778, 1987.

7 Michal Koucký. Circuit complexity of regular languages. *Theory Comput. Syst.*, 45(4):865–879, 2009.

8 Michal Koucký, Clemens Lautemann, Sebastian Poloczek, and Denis Therien. Circuit lower bounds via ehrenfeucht-fraisse games. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, CCC'06, pages 190–201, Washington, DC, USA, 2006. IEEE Computer Society.

9 Michal Koucký, Pavel Pudlák, and Denis Thérien. Bounded-depth circuits: Separating wires from gates. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, STOC'05, pages 257–265, New York, NY, USA, 2005. ACM.

10 Andreas Krebs and Pierre Mckenzie. Private communication.

11 Andreas Krebs and A. V. Sreejith. Non-definability of languages by generalized first-order formulas over $(\mathbb{N},+)$. In *Proceedings of the 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, LICS'12, pages 451–460, Washington, DC, USA, 2012. IEEE Computer Society.

12 Andreas Krebs and Howard Straubing. An effective characterization of the alternation hierarchy in two-variable logic. In *Proceedings of the 32th international conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'12)*, volume 18 of *LIPIcs – Leibniz International Proceedings in Informatics*, pages 86–98, Dagstuhl, Germany, 2012. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

13 Manfred Kufleitner and Alexander Lauser. Quantifier alternation in two-variable first-order logic with successor is decidable. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science*, STACS'13, pages 305–316. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013.

14 Manfred Kufleitner and Pascal Weil. The FO^2 alternation hierarchy is decidable. In *Computer science logic 2012*, volume 16 of *LIPIcs – Leibniz International Proceedings in Informatics*, pages 426–439. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012.

15 Leonid Libkin. *Elements of Finite Model Theory.* Springer, 2004.

16 Prabhakar Ragde and Avi Wigderson. Linear-size constant-depth polylog-threshold circuits. *Inform. Process. Lett.*, 39(3):143–146, 1991.

17 Frank P. Ramsey. On a Problem of Formal Logic. *Proc. London Math. Soc.*, S2-30(1):264, 1930.

18 Amitabha Roy and Howard Straubing. Definability of languages by generalized first-order formulas over $(\mathbb{N},+)$. *SIAM J. Comput.*, 37(2):502–521 (electronic), 2007.

19 Philipp Weis and Neil Immerman. Structure theorem and strict alternation hierarchy for FO^2 on words. *Log. Methods Comput. Sci.*, 5(3):3:4, 23, 2009.