Descriptive complexity for pictures languages

Étienne Grandjean Frédéric Olive

April 27, 2012

Abstract

This paper deals with descriptive complexity of picture languages of any dimension by syntactical fragments of existential second-order logic. Two classical classes of picture languages are studied:

- The class of *recognizable* picture languages, i.e. projections of languages defined by local constraints (or tilings): it is known as the most robust class extending the class of regular languages to any dimension;

- The class of picture languages recognized on *nondeterministic cellular automata in linear time* : cellular automata is the simplest and most natural model of parallel computation and linear time is their minimal time class allowing synchronization.

We uniformly generalize to any dimension the characterization by Giammarresi et al. [12] of the class of *recognizable* picture languages in existential monadic second-order logic.

We state several logical characterizations of the class of picture languages recognized in linear time on nondeterministic cellular automata. They are the first machine-independent characterizations of complexity classes of cellular automata.

Our characterizations are essentially deduced from normalization results we prove for first-order and existential second-order logics over pictures. They are obtained in a general and uniform framework that allows to extend them to other "regular" structures. Finally, we describe some hierarchy results that show the optimality of our logical characterizations and delineate their limits.

Keywords: Picture languages; locality and tilling; recognizability; linear time; cellular automata; logical characterizations; existential second-order logic.

Contents

Introduction

One goal of descriptive complexity is to establish logical characterizations of natural classes of problems in finite model theory. Many results in this area involve second-order logic (SO) and its restrictions, monadic second-order logic (MSO) and existential second-order logic (ESO). Indeed, there are two lines of research that roughly correspond to either of these restrictions:

a) *The formal language current:* It starts from the pioneering result by Büchi, Elgot and Trakhtenbrot [1, 7, 21] that states that the class of regular languages equals the class of languages definable in MSO, in short, REG = MSO. This line of research aims at characterizing in logic the natural classes of algebraically defined languages (sets of words) or sets of structures (trees, graphs, etc.) defined by finite state recognizability or local properties such as tilings.

b) *The computational complexity current:* It originates from another famous result, Fagin's Theorem [8], which characterizes the class NP as the class of problems definable in ESO.

For many years, both directions of research have produced plenty of results: see *e.g.* [6, 18] for descriptive complexity of formal languages and [6, 16, 13, 18] for the one of complexity classes. However, and this may be surprising, only few connections are known between those two areas of descriptive complexity. Of course, an explanation is that formal language theory has its own purposes that have little to do with complexity theory. In our opinion, the main reason is that while MSO logic *exactly fits* the fundamental notion of *recognizability*, as exemplified in the work of Courcelle [2], this logic seems *transversal* to computational complexity. We argue this is due to the intrinsic *locality* that MSO logic *inherits* from first-order logic [15, 10]. Typically, whereas MSO, or even existential MSO (EMSO), expresses some NP-complete graph problems such as 3-colourability, it cannot express some other ones such as Hamiltonicity (see [22, 3, 18] for instance) or even some tractable graph properties, such as the existence of a perfect matching in a graph. In contrast, the situation is very clear on trees as on words: MSO only captures the class of "easiest" problems; an extension of Büchi's Theorem [20] states that a tree language is MSO definable iff it is recognizable by a finite tree automaton.

Thus, Items *a)* and *b)* above seem quite separate for problems on words, trees or graphs. What about *picture languages*, that mean sets of d-pictures, i.e., d-dimensional words (or coloured grids)? First, notice the following results:

1. In a series of papers culminating in [12], Giammarresi et al. have proved that a 2-picture language is *recognizable*, i.e. is the projection of a local 2-picture language, iff it is definable in EMSO. In short: $\text{REC}^2 = \text{EMSO}$.

2. In fact, the class REC^2 contains some NP-complete problems. More generally, one observes that for each dimension $d \geq 1$, REC^d *can be defined as* the class of d-picture languages recognized by nondeterministic d-dimensional *cellular automata* in *constant time*[1].

The present paper originates from two questions about word/picture languages:

1. How can we generalize the proof of the above-mentioned theorem of Giammarresi et al. to any dimension? That is, can we establish the equality $\text{REC}^d = \text{EMSO}$ for any $d \geq 1$?

2. Can we obtain logical characterizations of time complexity classes of cellular automata[2]?

[1] That means: for such a picture language L, there is some constant integer c such that each computation stops at instant c and $p \in L$ iff it has at least one computation that stops *with each cell in an accepting state.*

[2] This originates from a question that J. Mazoyer asked us in 2000 (personal communication): give a logical characterization of the linear time complexity class of nondeterministic cellular automata.

A d-picture language is a set of d-pictures $p : [1,n]^d \to \Sigma$, i.e., d-dimensional Σ-words[3]. There are two natural manners to represent a d-picture p as a first-order structure:

- as a *pixel structure:* on the *pixel* domain $[1,n]^d$ where the sets $p^{-1}(a)$, $a \in \Sigma$, are encoded by unary relations $(U_a)_{a\in\Sigma}$ and the underlying d-dimensional grid is encoded by d successor functions (see Definition 1.2);
- as a *coordinate structure:* on the *coordinate* domain $[1,n]$ where the sets $p^{-1}(a)$ are encoded by d-ary relations $(R_a)_{a\in\Sigma}$; moreover, one uses the natural linear order of the coordinate domain $[1,n]$ and its associate successor function (see Definition 1.3).

Significantly, these two representations respectively correspond to the two above-mentioned points of view – formal language theory view *vs* computational complexity view – as illustrated by our results.

Our results

We establish two kinds of logical characterizations of d-picture languages, for all dimensions $d \geq 1$:

1. *On pixel structures:* $\mathrm{REC}^d = \mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\text{var } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$. That means a d-picture language is *recognizable iff* it is definable in monadic ESO (resp. in ESO with 1 first-order variable, or in monadic ESO with 1 universally quantified first-order variable).
2. *On coordinate structures:* $\mathrm{NLIN}^d_{\mathrm{ca}} = \mathrm{ESO}(\text{var } d+1) = \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$; that means a d-picture language is recognized by a nondeterministic d-dimensional cellular automaton in *linear time* (see *e.g.* [4]) iff it is definable in ESO with $d+1$ distinct first-order variables (resp. ESO with second-order variables of arity at most $d+1$ and a prenex first-order part of prefix $\forall^{d+1}$).

Items 1 and 2 proceed from normalization results of, respectively, first-order and ESO logics that we prove over picture languages.

Significance of our results

The *normalization equality* $\mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$ of Item 1 is a consequence of the fact that, on pixel structures (and, more generally, on structures that consist of bijective functions and unary relations), any first-order formula is equivalent to a boolean combination of *cardinality formulas* of the form: "there exists k distinct elements x such that $\psi(x)$", where ψ is a quantifier-free formula with only *one* variable. The normalization equality explicitly expresses the local feature of MSO on pictures – using only *one* first-order variable. The results of Item 1 can be regarded as an explicitation/simplification (using only*one* first-order variable) and uniformization of the proof and ideas of the main result of Giammarresi et al. [12, 11]; this allows us to generalize it to *any* dimension and, potentially, to other *regular* structures.

Intuitively, our characterization $\mathrm{NLIN}^d_{\mathrm{ca}} = \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$ of Item 2 naturally reflects a *symmetry* property of the time-space diagram of any computation of a *nondeterministic* d-dimensional cellular automaton: informally, the single first-order variable representing time *cannot be distinguished* from any of the d variables that represent the d-dimensional space; in other words, the $d+1$ variables can be permuted without this increases the expressive (or computational) power of the formula. This is the sense of the inclusion $\mathrm{ESO}(\forall^{d+1}, \text{arity } d+1) \subseteq \mathrm{NLIN}^d_{\mathrm{ca}}$ whose proof is far from trivial.

[3]More generally, the domain of a d-picture is of the "rectangular" form $[1,n_1] \times \ldots \times [1,n_d]$. For simplicity and uniformity of presentation, we have chosen to present the results of this paper in the particular case of "square" pictures of domain $[n]^d$. Fortunately, our results also hold with the same proofs for general domains $[1,n_1] \times \ldots \times [1,n_d]$.

1 Preliminaries

In the definitions below and all along the paper, we denote by Σ, Γ some finite alphabets and by d a positive integer. For any positive integer n, we set $[n] := \{1, \ldots, n\}$. We are interested in sets of pictures of any fixed dimension d.

Definition 1.1 *A d-dimensional picture or d-picture on Σ is a function $p : [n]^d \to \Sigma$ where n is a positive integer. The set $dom(p)= [n]^d$ is called the domain of picture p and its elements are called points, pixels or cells of the picture. A set of d-pictures on Σ is called a d-dimensional language, or d-language, on Σ.*

Notice that 1-pictures on Σ are nothing but nonempty words on Σ.

1.1 Pictures as model theoretic structures

Along the paper, we will often describe d-languages as sets of models of logical formulas. To allow this point of view, we must settle on an encoding of d-pictures as model theoretic structures.

For logical aspects of this paper, we refer to the usual definitions and notations in logic and finite model theory (see [6] or [18], for instance). A *signature* (or *vocabulary*) σ is a finite set of relation and function symbols each of which has a fixed arity. A (finite) *structure* S of vocabulary σ, or σ-structure, consists of a finite domain D of cardinality $n \geq 1$, and, for any symbol $s \in \sigma$, an interpretation of s over D, often denoted by s for simplicity. The tuple of the interpretations of the σ-symbols over D is called the *interpretation* of σ over D and, when no confusion results, it is also denoted σ. The *cardinality of a structure* is the cardinality of its domain. For any signature σ, we denote by STRUC(σ) the class of (finite) σ-structures. We write MODELS(Φ) the set of σ-structures which satisfy some fixed formula Φ. We will often deal with *tuples* of objects. We denote them by bold letters.

There are two natural manners to represent a picture by some logical structure: on the domain of its pixels, or on the domain of its coordinates. This gives rise to the following definitions:

Definition 1.2 *Given $p : [n]^d \to \Sigma$, we denote by $\mathsf{pixel}^d(p)$ the structure*

$$\mathsf{pixel}^d(p) = ([n]^d, (Q_s)_{s\in\Sigma}, (succ_i)_{i\in[d]}, (min_i)_{i\in[d]}, (max_i)_{i\in[d]}).$$

Here:

- *$succ_j$ is the (cyclic) successor function according to the j^{th} dimension of $[n]^d$, mapping each $(a_1, \ldots, a_d) \in [n]^d$ on $(a_1^{(j)}, \ldots, a_d^{(j)}) \in [n]^d$, where we set : $a_i^{(j)} = a_i$ for $i \neq j$ and, beside, $a_j^{(j)} = a_j + 1$ if $a_j < n$; $a_j^{(j)} = 1$ otherwise.*

  In other words, for $a \in [n]^d$, $succ_j(a)$ is the d-tuple $a^{(j)}$ obtained from a by "increasing" its j^{th} component according to the cyclic successor on $[n]$.

- *the min_i's, max_i's and Q_s's are the following unary (monadic) relations:*

$$min_i = \{a \in [n]^d : a_i = 1\}; \qquad max_i = \{a \in [n]^d : a_i = n_i\}; \qquad Q_s = \{a \in [n]^d : p(a) = s\}.$$

Definition 1.3 *Given $p : [n]^d \to \Sigma$, we denote by $\mathsf{coord}^d(p)$ the structure*

$$\mathsf{coord}^d(p) = \langle [n], (Q_s)_{s\in\Sigma}, <, succ, min, max\rangle. \qquad (1)$$

Here:

- *Each Q_s is a d-ary relation symbol interpreted as the set of cells of p labelled by an s. In other words: $Q_s = \{a \in [n]^d : p(a) = s\}$.*
- *$<$, min, max are predefined relation symbols of respective arities 2, 1, 1, that are interpreted, respectively, as the sets $\{(i, j) : 1 \leq i < j \leq n\}$, $\{1\}$ and $\{n\}$.*
- *succ is a unary function symbol interpreted as the cyclic successor. (That is: $succ(i) = i + 1$ for $i < n$ and $succ(n) = 1$.)*

For a d-language L, we set $\mathsf{pixel}^d(L) = \{\mathsf{pixel}^d(p) : p \in L\}$ and $\mathsf{coord}^d(L) = \{\mathsf{coord}^d(p) : p \in L\}$.

Remark 1.4 *Several details are irrelevant in Definitions 1.2 and 1.3, i.e. our results still hold for several variants, in particular:*

- In Definition 1.3, the fact that the linear order $<$ and the equality $=$ are allowed or not and the fact that min, max are represented by individual constants or unary relations;

- In both definitions, the fact that the successor function(s) is/are cyclic or not and is/are completed or not by predecessor(s) function(s).

At the opposite, it is essential that, in both definitions:

- The successor(s) is/are represented by function(s) *and not by (binary) relation(s);*

- The min, max are explicitly represented.

1.2 Logics under consideration

Let us now come to the logics involved in the paper. All formulas considered hereafter belong to *relational Existential Second-Order logic*. Given a signature σ, indifferently made of relational and functional symbols, a relational existential second-order formula of signature σ has the shape $\Phi \equiv \exists \mathbf{R} \varphi(\sigma, \mathbf{R})$, where $\mathbf{R} = (R_1, \dots, R_k)$ is a tuple of relational symbols and φ is a first-order formula of signature $\sigma \cup \{\mathbf{R}\}$. We denote by ESO^σ the class thus defined. We will often omit to mention σ for considerations on these logics that do not depend on the signature. Hence, ESO stands for the class of all formulas belonging to ESO^σ for some σ.

We will pay great attention to several variants of ESO. In particular, we will distinguish formulas of type $\Phi \equiv \exists \mathbf{R} \varphi(\sigma, \mathbf{R})$ according to: the number of distinct first-order variables involved in φ, the arity of the second-order symbols $R \in \mathbf{R}$, and the quantifier prefix of some prenex form of φ.

With the logic $\mathrm{ESO}^\sigma(\forall^d, \mathrm{arity}\, \ell)$, we control these three parameters: it is made of formulas of which first-order part is prenex with a universal quantifier prefix of length d, and where existentially quantified relation symbols are of arity at most ℓ. In other words, $\mathrm{ESO}^\sigma(\forall^d, \mathrm{arity}\, \ell)$ collects formulas of shape $\exists \mathbf{R} \forall \mathbf{x} \theta(\sigma, \mathbf{R}, \mathbf{x})$ where θ is quantifier free, $\mathbf{x}$ is a d-tuple of first-order variables, and $\mathbf{R}$ is a tuple of relation symbols of arity at most ℓ. Relaxing some constraints of the above definition, we set:

$$\mathrm{ESO}^\sigma(\forall^d) = \bigcup_{\ell > 0} \mathrm{ESO}^\sigma(\forall^d, \mathrm{arity}\, \ell) \text{ and } \mathrm{ESO}^\sigma(\mathrm{arity}\, \ell) = \bigcup_{d > 0} \mathrm{ESO}^\sigma(\forall^d, \mathrm{arity}\, \ell).$$

Finally, we write $\mathrm{ESO}^\sigma(\mathrm{var}\, d)$ for the class of formulas that involve at most d first-order variables, thus focusing on the sole number of distinct first-order variables (possibly quantified several times).

In the following sections, we'll prove that some logics have the same expressive power, as far as given sets of structures are concerned. When a normalization of a logic $\mathcal{L}$ into a logic $\mathcal{L}'$ is thus relativized to a specific class $\mathcal{S}$ of structures, we write: $\mathcal{L} = \mathcal{L}'$ on $\mathcal{S}$. The next definition details the meaning of this formulation.

Definition 1.5 *Given a set of structures $\mathcal{S}$ and a formula Φ, the set of models of Φ that belong to $\mathcal{S}$ is denoted by $\textsc{models}_{\mathcal{S}}(\Phi)$. Two formula Φ and Φ' are $\mathcal{S}$-equivalent if $\textsc{models}_{\mathcal{S}}(\Phi) = \textsc{models}_{\mathcal{S}}(\Phi')$. Given two logics $\mathcal{L}$ and $\mathcal{L}'$, we say that $\mathcal{L} \subseteq \mathcal{L}'$ on $\mathcal{S}$ if each $\Phi \in \mathcal{L}$ is $\mathcal{S}$-equivalent to some $\Phi' \in \mathcal{L}'$. Furthermore, we write $\mathcal{L} = \mathcal{L}'$ on $\mathcal{S}$ if both $\mathcal{L} \subseteq \mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$ hold on $\mathcal{S}$.*

In some very rare cases, we will consider the extension of ESO obtained by allowing quantification over functional symbols. The corresponding logic, ESOF, gathers all formulas of the form $\exists\mathbf{R}\exists\mathbf{f}\varphi(\sigma, \mathbf{R}, \mathbf{f})$, where $\mathbf{R}$ (resp. $\mathbf{f}$) is a tuple of relational (resp. functional) symbols and φ is any first-order formula of signature $\sigma \cup \{\mathbf{R}, \mathbf{f}\}$. The restrictions $\mathrm{ESOF}(\mathrm{var}\, d)$ and $\mathrm{ESOF}(\forall^d, \mathrm{arity}\, \ell)$ of ESOF are defined as for ESO. The expressive power of these logics is quite high. A $\sigma-$NRAM is a nondeterministic Random Access Machine that takes σ-structures as inputs in the following way: for each $s \in \sigma$ of arity ℓ and each ℓ-tuple $\mathbf{t} \in D^\ell$, a special register $[s, \mathbf{t}]$ contains the value of $s(\mathbf{t})$. Let $\mathrm{NTIME}^\sigma(n^d)$ be the class of problems on σ-structures decidable by a $\sigma-$NRAM in time $O(n^d)$ where n is the size of the domain D of structures. The following was proved in [14]:

Theorem 1.6 ([14]) *For all $d > 0$, $\mathrm{NTIME}(n^d) = \mathrm{ESOF}(\mathrm{var}\, d)$.*

In the same paper, a normalization of $\mathrm{ESOF}(\mathrm{var}\, d)$ was stated:

Proposition 1.7 ([14]) *For all $d > 0$, $\mathrm{ESOF}(\mathrm{var}\, d) = \mathrm{ESOF}(\forall^d, \mathrm{arity}\, d)$.*

2 A logical characterization of recognizable picture languages

In this section, we define the class of local (resp. recognizable) picture languages and establish the logical characterizations of the class of picture languages. In order to define a notion of locality based on subpictures we need to mark the border of each picture.

Definition 2.1 *By $\Gamma^\sharp$ we denote the alphabet $\Gamma \cup \{\sharp\}$ where $\sharp$ is a special symbol not in Γ. Let p be any d-picture of domain $[n]^d$ on Γ. The bordered d-picture of p, denoted by $p^\sharp$, is the d-picture $p^\sharp : [0, n+1]^d \to \Gamma^\sharp$ defined by $p^\sharp(a) = p(a)$ if $a \in dom(p)$; $\sharp$ otherwise. Here, "otherwise" means that a is on the border of $p^\sharp$, that is, some component a_i of a is 0 or $n+1$.*

Let us now define our notion of *local picture language* or *tilings language*. It is based on some sets of allowed patterns (called tiles) of the bordered pictures.

Definition 2.2

1. *Given a d-picture p and an integer $j \in [d]$, two cells $a = (a_i)_{i\in[d]}$ and $b = (b_i)_{i\in[d]}$ of p are j-adjacent if they have the same coordinates, except the j^{th} one for which $|a_j - b_j| = 1$.*
2. *A tile for a d-language L on Γ is a pair in $(\Gamma^\sharp)^2$.*
3. *A picture p is j-tiled by a set of tiles $\Delta \subseteq (\Gamma^\sharp)^2$ if for any two j-adjacent points $a, b \in dom(p^\sharp)$: $(p^\sharp(a), p^\sharp(b)) \in \Delta$.*
4. *Given d sets of tiles $\Delta_1, \ldots, \Delta_d \subseteq (\Gamma^\sharp)^2$, a d-picture p is tiled by $(\Delta_1, \ldots, \Delta_d)$ if p is j-tiled by Δ_j for each $j \in [d]$.*
5. *We denote by $L(\Delta_1, \ldots, \Delta_d)$ the set of d-pictures on Γ that are tiled by $(\Delta_1, \ldots, \Delta_d)$.*

6. *A d-language L on Γ is local if there exist $\Delta_1,\ldots,\Delta_d \subseteq (\Gamma^\sharp)^2$ such that $L = L(\Delta_1,\ldots,\Delta_d)$. We then say that L is $(\Delta_1,\ldots,\Delta_d)$-local, or $(\Delta_1,\ldots,\Delta_d)$-tiled.*

Definition 2.3 *A d-language L on Σ is recognizable if it is the projection (i.e. homomorphic image) of a* local *d-language over an alphabet Γ. It means there exist a surjective function $\pi : \Gamma \to \Sigma$ and a local d-language L_{loc} on Γ such that $L = \{\pi \circ p : \; p \in L_{loc}\}$. Because of the last item of Definition 2.2, one can also write: L is recognizable if there exist a surjective function $\pi : \Gamma \to \Sigma$ and d sets $\Delta_1,\ldots,\Delta_d \subseteq (\Gamma^\sharp)^2$ such that $L = \{\pi \circ p : \; p \in L(\Delta_1,\ldots,\Delta_d)\}$.We write REC^d for the class of recognizable d-languages.*

Remark 2.4 *Our notion of* locality *is weaker than the one given by Giammarresi and al. [12]. But this doesn't affect the meaning of* recognizability*, which coincides with the one used in [12]. This confirms the robustness of this latter notion.*

A characterization of recognizable languages of dimension 2 by a fragment of existential monadic second-order logic was proved by Giammarresi et al. [12]. They established:

Theorem 2.5 ([12]) *For any 2-language L: $L \in \mathrm{REC}^2 \Leftrightarrow \mathsf{pixel}^2(L) \in \mathrm{ESO}(\text{arity } 1)$.*

In this section, we come back to this result. We simplify its proof, refine the logic it involves, and generalize its scope to any dimension.

Theorem 2.6 *For any $d > 0$ and any d-language L, the following assertions are equivalent:*

1. $L \in \mathrm{REC}^d$*;*
2. $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1)$*;*
3. $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\text{arity } 1)$.

Proposition 2.9 states the equivalence 1 $\Leftrightarrow$ 2. In Proposition 2.14, we establish the normalization $\mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$ on pixel structures, from which the equivalence 2 $\Leftrightarrow$ 3 immediately follows.

2.1 A characterization of REC for pixel encoding

In order to prove Proposition 2.9, it is convenient to first normalize the sentences of $\mathrm{ESO}(\forall^1, \text{arity } 1)$. This is the role of the technical result below, which asserts that on pixel encodings, each such sentence can be rewritten in a very local form where the first-order part alludes only pairs of adjacent pixels of the bordered picture.

Lemma 2.7 *On pixel structures, any $\varphi \in \mathrm{ESO}(\forall^1, \text{arity } 1)$ is equivalent to a sentence of the form:*

$$\exists \mathbf{U} \forall x \bigwedge_{i \in [d]} \left\{ \begin{array}{rcll} min_i(x) & \to & \mathsf{m}_i(x) & \wedge \\ max_i(x) & \to & \mathsf{M}_i(x) & \wedge \\ \neg max_i(x) & \to & \Psi_i(x) & \end{array} \right\}. \tag{2}$$

Here, $\mathbf{U}$ is a list of monadic relation variables and m_i, M_i, Ψ_i are quantifier-free formulas such that

- *atoms of m_i and M_i have all the form $Q(x)$;*
- *atoms of Ψ_i have all the form $Q(x)$ or $Q(succ_i(x))$,*

where, in both cases, $Q \in \{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$.

Proof. Let be a sentence $\varphi \in \mathrm{ESO}(\forall^1, \text{arity } 1)$ of signature $pixel_{d,\Sigma}$.

Suppression of equalities: Without loss of generality, assume that φ is in negative normal form $\exists \mathbf{U} \forall x \psi$ and that each equality in ψ is of the form

$$\mathrm{succ}_{i_1}^{c_{i_1}} \ldots \mathrm{succ}_{i_k}^{c_{i_k}}(x) = x \tag{3}$$

where the k indices $i_1, \ldots, i_k$ ($k > 0$) are pairwise distinct, and the exponents $c_{i_1}, \ldots, c_{i_k}$ are positive integers. Equation (3) holds in some pixel structure $\mathsf{pixel}^d(p)$ of domain $[n_1] \times \cdots \times [n_d]$ for some x (or equivalently, for all x), iff the dimensions n_i of p satisfy the k equalities $n_{i_1} = c_{i_1}, \ldots, n_{i_k} = c_{i_k}$. So, we have to suppress any equality/inequality of the form $n_i = c$ or $n_i \neq c$, for $c \geq 1$, in φ. First, notice that the inequality $n_i \neq c$ can be rewritten as

$$n_i > c \vee \bigvee_{j \in [c-1]} n_i = j.$$

So, there remains to suppress any equality/inequality of the form $n_i = c$ or $n_i > c$, for $c \geq 1$, in φ. This can be done by introducing c new unary relation symbols denoted $\mathrm{coord}_i^{=j}(x)$, for $j \in [c]$. Intuitively, $\mathrm{coord}_i^{=j}(x)$ means: "the i^{th} coordinate of point x is j". Clearly, for any fixed $i \in [d]$ and $c > 0$, the unary relations $\mathrm{coord}_i^{=j}$ are defined by induction on $j \in [c]$ with the following formula

$$\delta_i^c = (\min_i(x) \to \mathrm{coord}_i^{=1}(x)) \ \wedge \bigwedge_{j \in [c-1]} (\neg \max_i(x) \to (\mathrm{coord}_i^{=j}(x) \to \mathrm{coord}_i^{=j+1}(\mathrm{succ}_i(x)))).$$

Using those relations, it is rather easy to see that the two formulas

$$\forall x (\max_i(x) \to \mathrm{coord}_i^{=c}(x)) \ \text{and} \ \forall x \bigwedge_{j \in [c]} (\mathrm{coord}_i^{=j}(x) \to \mathrm{coord}_i^{=j+1}(\mathrm{succ}_i(x)))$$

express the assertions $n_i = c$ and $n_i > c$, respectively. Hence, the first-order sentence $\forall x \psi(x)$ is equivalent to the $\mathrm{ESO}(\forall^1, \text{arity } 1)$-sentence:

$$\exists \mathbf{coord} \ \forall x (\delta(x) \wedge \psi'(x))$$

where **coord** denotes the list of unary relation variables $\mathrm{coord}_i^{=j}$ introduced in the required formulas $\delta_i^c(x)$, the conjunction of which is denoted $\delta(x)$, and $\psi'(x)$ is the formula $\psi(x)$ where each "subformula" $n_i = c$ (resp. $n_i > c$) is replaced by the equivalent formula $\max_i(x) \to \mathrm{coord}_i^{=c}(x)$ (resp. $\bigwedge_{j \in [c]} (\mathrm{coord}_i^{=j}(x) \to \mathrm{coord}_i^{=j+1}(\mathrm{succ}_i(x)))$).

So, our sentence φ can be assumed to be in prenex conjunctive normal form $\exists \mathbf{U} \forall x \psi$ *without equality*, that means ψ is a conjunction of clauses with literals of the form $Q(\tau(x))$ or $\neg Q(\tau(x))$ where Q belongs to the set of relations $\{(\min_i)_{i \in [d]}, (\max_i)_{i \in [d]}, (Q_s)_{s \in \Sigma}, \mathbf{U}\}$ and τ is a (possibly empty) composition of function symbols succ_i, $i \in [d]$. The idea is to introduce for each atom $Q(\tau(x))$ that occurs in ψ a new unary relation variable denoted $U_{Q,\tau}$ so that the atom $U_{Q,\tau}(x)$ is equivalent to (can replace) the atom $Q(\tau(x))$.

The $U_{Q,\tau}$'s are defined inductively by the conjunction of the following equivalences, denoted by *basic* and *succ$_i$-induct*:

- *basic*: $U_{Q,Id}(x) \to Q(x)$,
- *succ$_i$-induct*: $U_{Q,\tau \mathrm{succ}_i}(x) \to U_{Q,\tau}(\mathrm{succ}_i(x))$

from which the equivalence $U_{Q,\tau}(x) \rightarrow Q(\tau(x))$ can be deduced as wished.

Let $\delta(x)$ denote the conjunction of all the equivalences that define the $U_{Q,\tau}$'s and let $\psi'(x)$ denote the formula $\psi(x)$ where each atom $Q(\tau(x))$ is replaced by the atom $U_{Q,\tau}(x)$. Clearly, the sentence $\varphi = \exists \mathbf{U} \forall x \psi(x)$ is equivalent to the sentence

$$\varphi' = \exists \mathbf{U} \exists (U_{Q,\tau})'_s \forall x (\psi'(x) \wedge \delta(x)).$$

Now, put φ', that means $\psi'(x) \wedge \delta(x)$, in conjunctive normal form. In order to organize and transform the clauses of φ', some terminology is required about clauses:

- a clause is *x-pure* (resp. *i-cyclic*) if it only contains atoms of the form $Q(x)$ (resp. $Q(x)$ or $Q(\mathrm{succ}_i(x))$) where Q is a unary relation symbol which is neither any $\min_j$ nor any $\max_j$;
- an *i-local* clause is of the form $\neg\max_i(x) \rightarrow C(x)$ where $C(x)$ is an i-cyclic clause;
- an *i-min* (resp. *i-max*) clause is of the form $\min_i(x) \rightarrow C(x)$ (resp. $\max_i(x) \rightarrow C(x)$) where $C(x)$ is an x-pure clause.

Using those definitions, we observe that

- the clauses of the conjunctive normal form of $\psi'(x)$ are x-pure;
- the *succ$_i$-induct* clauses of $\delta(x)$ are i-cyclic;
- each *basic* implication of $\delta(x)$ of the form $U_{Q,Id}(x) \rightarrow Q(x)$ or $Q(x) \rightarrow U_{Q,Id}(x)$ is a x-pure clause except in case Q is $\min_i$ or $\max_i$ ($i \in [d]$); clearly, in this case, the implication can be rephrased in one of the following four forms:

  1. $\min_i(x) \rightarrow C(x)$,
  2. $\max_i(x) \rightarrow C(x)$,
  3. $\neg\min_i(x) \rightarrow C(x)$,
  4. $\neg\max_i(x) \rightarrow C(x)$,

  where $C(x)$ is an x-pure clause (literal). Clauses 1 and 2 are i-min and i-max clauses, respectively. Clause 4 is i-local. Clause 3 can be replaced by the i-local clause $\neg\max_i(x) \rightarrow C(\mathrm{succ}_i(x))$; this is justified by the equivalence (easily proved) of the universally quantified versions of those implications:

  $$\forall x(\neg\min_i(x) \rightarrow C(x)) \Leftrightarrow \forall x(\neg\max_i(x) \rightarrow C(\mathrm{succ}_i(x))).$$

So, we have shown how to rephrase the first-order part (that is $\psi'(x) \wedge \delta(x)$) of φ' as a conjunction of x-pure clauses, i-cyclic clauses, i-local clauses, i-min clauses and i-max clauses. In fact, all those clauses are local with the exception of i-cyclic clauses. Recall that an i-cyclic clause $C(x, \mathrm{succ}_i(x))$ only contains atoms of the two forms $Q(x)$ or $Q(\mathrm{succ}_i(x)))$ where Q is a unary relation symbol which is neither any $\min_j$ nor any $\max_j$. Its nonlocality is due to the following fact: if for a d-picture P we have $a \in \max_i$ for any pixel $a \in dom(p)$, then the pixel $\mathrm{succ}_i(a)$ is not adjacent to a in P since we have $\mathrm{succ}_i(a) \in \min_i$ by cyclicity of the function (permutation) succ_i. In order to recover locality, let us first replace each i-cyclic clause $C(x, \mathrm{succ}_i(x))$ by the equivalent conjunction of the following two clauses:

1. the i-local clause $\neg\max_i(x) \rightarrow C(x, \mathrm{succ}_i(x))$;

2. the "nonlocal" clause $\max_i(x) \to C(x, \mathrm{succ}_i(x))$.

So, there remains to get rid of the "nonlocal" clause 2. The trick consists in making available in all the points of any succ_i-cycle the value of each unary relation Q for the $\min_i$ point of this cycle. This can be done by using a new unary relation symbol $U^Q_{min,i}(x)$ defined inductively by the conjunction of the following $\min_i$ and i-local clauses

$$\min_i(x) \to (U^Q_{min,i}(x) \to Q(x))$$

$$\neg\max_i(x) \to (U^Q_{min,i}(x) \to U^Q_{min,i}(\mathrm{succ}_i(x))).$$

Clearly, for each point a of any succ_i-cycle γ, we have the constant value $U^Q_{min,i}(a) = Q(b)$ where b is the unique point in $\gamma \cap \min_i$. A new unary relation symbol $U^Q_{max,i}$ can be defined similarly for $\max_i$. This justifies that each "nonlocal" clause $\max_i(x) \to C(x, \mathrm{succ}_i(x))$ can be replaced by the x-pure clause $C'(x)$ obtained by substituting in the clause $C(x, \mathrm{succ}_i(x))$ each atom $Q(x)$ (resp. $Q(\mathrm{succ}_i(x))$) by $U^Q_{max,i}(x)$ (resp. $U^Q_{min,i}(x)$).

Let us recapitulate what we have obtained. Our initial sentence $\varphi = \exists \mathbf{U} \forall x \psi(x)$ of ESO($\forall^1$, arity 1) is logically equivalent to a sentence of the form $\varphi" = \exists \mathbf{U}' \forall x \Psi(x)$ where

- $\mathbf{U}'$ is the union of the set of ESO unary symbols of φ', that are $\mathbf{U}$ and the $U_{Q,\tau}$'s, and of the $U^Q_{min,i}$'s and $U^Q_{max,i}$'s we have just introduced;
- $\Psi(x)$ is a conjunction of x-pure clauses, i-min clauses, i-max clauses and i-local clauses.

Now, it is easy to transform the conjunction of clauses $\Psi(x)$ into the conjunction of formulas required:

$$\bigwedge_{i \in [d]} [(\min_i(x) \to \Psi_i^{min}(x)) \land (\max_i(x) \to \Psi_i^{max}(x)) \land (\neg\max_i(x) \to \Psi_i(x))].$$

More precisely, for each $i \in [d]$,

- the conjunction of the i-min clauses (resp. the i-max clauses) and the x-pure clauses of $\Psi(x)$ is trivially transformed into the required form $\min_i(x) \to \Psi_i^{min}(x)$ (resp. $\max_i(x) \to \Psi_i^{max}(x)$);
- the conjunction of the i-local clauses and the x-pure clauses of $\Psi(x)$ is similarly transformed into the required form $(\neg\max_i(x) \to \Psi_i(x))$.

This completes the proof. □

Remark 2.8 *The normal form of the formula obtained in Lemma 2.7 guarantees its local feature. In particular, notice that any successor symbol $succ_i$ can only apply to arguments assumed to be not in max_i. That means we get the same normal form if the cyclic successor functions $succ_i$, $i \in [d]$, are replaced by successor functions for which $succ_i(a) = a$ (instead of $a^{(i)}$) if $a \in max_i$.*

Proposition 2.9 *For any $d > 0$ and any d-language L on Σ: $L \in \mathrm{REC}^d \Leftrightarrow \mathsf{pixel}^d(L) \in$ ESO($\forall^1$, arity 1).*

Proof. $\boxed{\Rightarrow}$ A picture belongs to L if there exists a tiling of its domain whose projection coincides with its content. In the logic involved in the proposition, the "arity 1" corresponds to formulating the existence of the tiling, while the "$\forall^1$" is the syntactic resource needed to express that the tiling behaves as expected. Let us detail these considerations.

By Definition 2.2, there exist an alphabet Γ (which can be assumed disjoint from Σ), a surjective function $\pi : \Gamma \to \Sigma$ and d subsets $\Delta_1, \ldots, \Delta_d \subseteq (\Gamma^\sharp)^2$ such that

$$L = \{\pi \circ p' : p' \in L(\Delta_1, \ldots, \Delta_d)\} \tag{4}$$

The belonging of a picture $p' : [n]^d \to \Gamma$ to $L(\Delta_1, \ldots, \Delta_d)$ is easily expressed on $\mathsf{pixel}^d(p')$ with a first-order formula that asserts, for each dimension $i \in [d]$, that for any pixel x of p', the couple $(x, \mathrm{succ}_i(x))$ can be tiled with some element of Δ_i. Namely,

$p' \in L(\Delta_1, \ldots, \Delta_d)$ iff $\mathsf{pixel}^d(p') \models \forall x \psi_{\Delta_1,\ldots,\Delta_d}(x)$, where:

$$\psi_{\Delta_1,\ldots,\Delta_d}(x) = \bigwedge_{i \in [d]} \left\{ \begin{array}{lll} \min_i(x) \to & \displaystyle\bigvee_{(\sharp,s) \in \Delta_i} Q_s(x) & \wedge \\ \neg\max_i(x) \to & \displaystyle\bigvee_{(s,s') \in \Delta_i} (Q_s(x) \wedge Q_{s'}(\mathrm{succ}_i(x))) & \wedge \\ \max_i(x) \to & \displaystyle\bigvee_{(s,\sharp) \in \Delta_i} Q_s(x) & \end{array} \right\}.$$

Now, by (4), a picture $p : [n]^d \to \Sigma$ belongs to L iff it results from a π-renaming of a picture $p' \in L(\Delta_1, \ldots, \Delta_d)$. It means there exists a Γ-labeling of p (that is, a tuple $(Q_s)_{s \in \Gamma}$ of subsets of $[n]^d$) corresponding to a picture of $L(\Delta_1, \ldots, \Delta_d)$ (i.e. fulfilling $\forall x \psi_{\Delta_1,\ldots,\Delta_d}(x, (Q_s)_{s\in\Gamma})$) and from which the actual Σ-labeling of p (that is, the subsets $(Q_s)_{s\in\Sigma}$) is obtained *via* π. More precisely:

$p \in L$ iff $\mathsf{pixel}^d(p) \models \Theta_L$, where:

$$\Theta_L = (\exists Q_s)_{s\in\Gamma} \forall x : \psi_{\Delta_1,\ldots,\Delta_d}(x) \wedge \bigwedge_{s \in \Sigma} \left(Q_s(x) \to [\bigoplus_{s' \in \pi^{-1}(s)} Q_{s'}(x) \wedge \bigwedge_{s' \in \Gamma \setminus \pi^{-1}(s)} \neg Q_{s'}(x)] \right).$$

Here, $\bigoplus$ denotes the exclusive disjunction. Notice that since $\Sigma \cap \Gamma = \emptyset$, the tuples $(Q_s)_{s\in\Sigma}$ and $(Q_s)_{s\in\Gamma}$ are also disjoint. Since Θ_L clearly belongs to $\mathrm{ESO}(\forall^1, \text{arity } 1)$, the proof is complete.

$\boxed{\Leftarrow}$ Consider L such that $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1)$. Lemma 2.7 ensures that $\mathsf{pixel}^d(L)$ is characterized by a sentence of the form:

$$\exists \mathbf{U} \forall x \bigwedge_{i \in [d]} \left\{ \begin{array}{lll} \min_i(x) & \to & \mathsf{m}_i(x) \quad \wedge \\ \max_i(x) & \to & \mathsf{M}_i(x) \quad \wedge \\ \neg\max_i(x) & \to & \Psi_i(x) \end{array} \right\}. \tag{5}$$

Here, $\mathbf{U}$ is a list of monadic relation variables and m_i, M_i, Ψ_i are quantifier-free formulas such that atoms of m_i and M_i have all the form $Q(x)$ and atoms of Ψ_i have all the form $Q(x)$ or $Q(\mathrm{succ}_i(x))$, wih $Q \in \{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$.

We have to prove that L is the projection of some local d-language L_{loc} on some alphabet Γ, that is a $(\Delta_1, \ldots, \Delta_d)$-tiled language for some $\Delta_1, \ldots, \Delta_d \subseteq \Gamma^2$. Let $U_1, \ldots, U_k$ denote the list of (distinct) elements of the set $\{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$ of unary relation symbols of φ, so that the first ones $U_1, \ldots, U_m$ are the Q_s's (here, $\min_i$ and $\max_i$ symbols are excluded). The trick is to put each subformula $\mathsf{m}_i(x)$, $\mathsf{M}_i(x)$ and $\Psi_i(x)$ of φ into its *complete disjunctive normal form* with respect to $U_1, \ldots, U_k$. Typically, each subformula $\Psi_i(x)$ whose atoms are of the form $U_j(x)$ or $U_j(\mathrm{succ}_i(x))$, for some $j \in [k]$, is transformed into the following "complete disjunctive normal form":

$$\bigvee_{(\epsilon,\epsilon') \in \Delta_i} \left(\bigwedge_{j \in [k]} \epsilon_j U_j(x) \wedge \bigwedge_{j \in [k]} \epsilon'_j U_j(\mathrm{succ}_i(x)) \right). \tag{6}$$

Here, the following conventions are adopted:

- $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in \{0,1\}^k$ and similarly for ϵ';
- for any atom α and any bit $\epsilon_j \in \{0,1\}$, $\epsilon_j\alpha$ denotes the literal α if $\epsilon_j = 1$, the literal $\neg\alpha$ otherwise.

For $\epsilon \in \{0,1\}^k$, we denote by $\Theta_\epsilon(x)$ the "complete conjunction" $\bigwedge_{j\in[k]} \epsilon_j U_j(x)$. Intuitively, $\Theta_\epsilon(x)$ is a complete description of x and the set

$$\Gamma = \bigcup_{i\in[m]} \{0^{i-1}10^{m-i}\} \times \{0,1\}^{k-m}$$

is the set of possible colors (remember that the Q_s's that are the U_j's for $j \in [m]$ form a partition of the domain). The complete disjunctive normal form (6) of $\Psi_i(x)$ can be written into the suggestive form

$$\bigvee_{(\epsilon,\epsilon')\in\Delta_i} (\Theta_\epsilon(x) \wedge \Theta_{\epsilon'}(\mathrm{succ}_i(x))).$$

If each subformula $\mathsf{m}_i(x)$ and $\mathsf{M}_i(x)$ of φ is similarly put into complete disjunctive normal form, that is $\bigvee_{(\sharp,\epsilon)\in\Delta_i} \Theta_\epsilon(x)$ and $\bigvee_{(\epsilon,\sharp)\in\Delta_i} \Theta_\epsilon(x)$, respectively (there is no ambiguity in our implicit definition of the Δ_i's, since $\sharp \notin \Gamma$), then the above sentence (5) equivalent to φ becomes the following equivalent sentence:

$$\varphi' = \exists \mathbf{U} \forall x \bigwedge_{i\in[d]} \left\{ \begin{array}{lcll} \min_i(x) & \to & \bigvee_{(\sharp,\epsilon)\in\Delta_i} \Theta_\epsilon(x) & \wedge \\ \max_i(x) & \to & \bigvee_{(\epsilon,\sharp)\in\Delta_i} \Theta_\epsilon(x) & \wedge \\ \neg\max_i(x) & \to & \bigvee_{(\epsilon,\epsilon')\in\Delta_i} (\Theta_\epsilon(x) \wedge \Theta_{\epsilon'}(\mathrm{succ}_i(x))) & \end{array} \right\}$$

Finally, let L_{loc} denote the d-language over Γ defined by the first-order sentence φ_{loc} obtained by replacing each Θ_ϵ by the new unary relation symbol Q_ϵ in the first-order part of φ'. In other words, $\mathsf{pixel}^d(L_{\mathrm{loc}})$ is defined by the following first-order sentence:

$$\varphi_{\mathrm{loc}} = \forall x \bigwedge_{i\in[d]} \left\{ \begin{array}{lcll} \min_i(x) & \to & \bigvee_{(\sharp,\epsilon)\in\Delta_i} Q_\epsilon(x) & \wedge \\ \max_i(x) & \to & \bigvee_{(\epsilon,\sharp)\in\Delta_i} Q_\epsilon(x) & \wedge \\ \neg\max_i(x) & \to & \bigvee_{(\epsilon,\epsilon')\in\Delta_i} (Q_\epsilon(x) \wedge Q_{\epsilon'}(\mathrm{succ}_i(x))) & \end{array} \right\}$$

Hence, $L_{\mathrm{loc}} = L(\Delta_1, \dots, \Delta_d)$. That is, L_{loc} is indeed local and the corresponding sets of tiles are the Δ_i's of the previous formula. It is now easy to see that our initial d-language L is the projection of the local language L_{loc} by the projection $\pi : \Gamma \to \Sigma$ defined as follows: $\pi(\epsilon) = s$ iff $\epsilon_i = 1$ for $i \in [m]$ and U_i is Q_s. This completes the proof. □

2.2 A normalization of ESO(arity 1) on pixels

Let's now come to the last step of the proof of Theorem 2.6. A key point of this step is a quantifier elimination result for first-order logic, proved by Durand and Grandjean. Its statement needs two new definitions.

Definition 2.10 *A* bijective structure *is a finite structure of the form* $\mathcal{S} = (dom(\mathcal{S}), f_1, \dots, f_d, U_1, \dots, U_m)$, *where the* f_i*'s are unary bijective functions and the* U_i*'s are unary relations.*

Definition 2.11 *A* cardinality formula *is a first-order formula of the form* $\exists^{\geq k} x \psi(x)$*, where* $k \geq 1$ *and* $\psi(x)$ *is a quantifier-free formula with only one variable* x*. The quantifier* $\exists^{\geq k} x$ *means "there exists at least* k *elements* x*".*

The following normalization of first-order logic on bijective structures was proved in [5].

Proposition 2.12 *([5]) On bijective structures, each first-order sentence is equivalent to a boolean combination of cardinality formulas.*

Actually, this result was proved for a slightly more restrictive notion of bijective structure, but the above generalisation is straightforward. Clearly, a pixel structure is a bijective structure. Therefore:

Corollary 2.13 *On pixel structures, each first-order sentence is equivalent to a boolean combination of cardinality formulas.*

This allows to prove the following proposition.

Proposition 2.14 $\mathrm{ESO}(\text{arity } 1) \subseteq \mathrm{ESO}(\forall^1, \text{arity } 1)$ *on pixel structures, for any* $d > 0$*.*

Proof. In a pixel structure, each function symbol succ_i is interpreted as a cyclic successor, hence, as a bijective function. So, a pixel structure is a bijective structure and, by Corollary 2.13, it can be written as a boolean combination of sentences of the form $\psi^{\geq k} = \exists^{\geq k} x\, \psi(x)$ (for $k \geq 1$) where $\psi(x)$ is a quantifier-free formula with the single variable x. Therefore, it is easily seen that proving the proposition amounts to show that each sentence of the form $\psi^{\geq k}$ or $\neg\psi^{\geq k}$ can be translated in $\mathrm{ESO}(\forall^1, \text{arity } 1)$ on pixel structures. This is done as follows:

For a given sentence $\exists^{\geq k} x\, \psi(x)$, we introduce k new unary relations $U^{=0}, U^{=1}, \ldots, U^{=k-1}$ and $U^{\geq k}$, with the intended meaning:

> *A pixel* $a \in [n]^d$ *belongs to* $U^{=j}$ *(resp.* $U^{\geq k}$*) iff there are exactly* j *(resp. at least* k*) pixels* $b \in [n]^d$ *lexicographically smaller than or equal to* a *such that* $\mathsf{pixel}^d(p) \models \psi(b)$*.*

Then we have to compel these relation symbols to fit their expected interpretations, by means of a first-order formula with a single universally quantified variable. First, we demand the relations to be pairwise disjoint:

(1) $$\bigwedge_{i<j<k} \left(\neg U^{=i}(x) \vee \neg U^{=j}(x)\right) \wedge \bigwedge_{i<k} \left(\neg U^{=i}(x) \vee \neg U^{\geq k}(x)\right).$$

Then, we temporarily denote by $\leq_{\mathrm{lex}}$ the lexicographic order on $[n]^d$ inherited from the natural order on $[n]$, and by $\mathrm{succ}_{\mathrm{lex}}$, $\min_{\mathrm{lex}}$, $\max_{\mathrm{lex}}$ its associated successor function and unary relations corresponding to extremal elements. Then the sets described above can be defined inductively by the conjunction of the following six formulas:

(2) $$(\min_{\mathrm{lex}}(x) \wedge \neg\psi(x)) \rightarrow U^{=0}(x)$$

(3) $$(\min_{\mathrm{lex}}(x) \wedge \psi(x)) \rightarrow U^{=1}(x)$$

(4) $$\bigwedge_{i<k} \left(\neg\max_{\mathrm{lex}}(x) \wedge U^{=i}(x) \wedge \neg\psi(\mathrm{succ}_{\mathrm{lex}}(x))\right) \rightarrow U^{=i}(\mathrm{succ}_{\mathrm{lex}}(x))$$

(5) $\bigwedge_{i<k-1} \Big(\neg\max_{\text{lex}}(x) \wedge U^{=i}(x) \wedge \psi(\text{succ}_{\text{lex}}(x))\Big) \to U^{=i+1}(\text{succ}_{\text{lex}}(x))$

(6) $\Big(\neg\max_{\text{lex}}(x) \wedge U^{=k-1}(x) \wedge \psi(\text{succ}_{\text{lex}}(x))\Big) \to U^{\geq k}(\text{succ}_{\text{lex}}(x))$

(7) $\Big(\neg\max_{\text{lex}}(x) \wedge U^{\geq k}(x)\Big) \to U^{\geq k}(\text{succ}_{\text{lex}}(x))$.

Hence, under the hypothesis (1) $\wedge \ldots \wedge$ (7), the sentences $\psi^{\geq k}$ and $\neg\psi^{\geq k}$ are equivalent, respectively, to

$$\forall x(\max_{\text{lex}}(x) \to U^{\geq k}(x)) \text{ and } \forall x(\max_{\text{lex}}(x) \to \neg U^{\geq k}(x)).$$

To complete the proof, it remains to get rid of the symbols succ_{lex}, $\min_{\text{lex}}$ and $\max_{\text{lex}}$ that are not allowed in our language. It is done by referring to these symbols implicitly rather than explicitly. For instance, since $\text{succ}_{\text{lex}}(x) = \text{succ}_i\text{succ}_{i+1}\ldots\text{succ}_d(x)$, for each non maximal $x \in [n]^d$, i.e., distinct from $(n,\ldots,n)$, and for the smallest $i \in [d]$ such that $\bigwedge_{j>i}\max_j(x)$, each formula φ involving $\text{succ}_{\text{lex}}(x)$ actually corresponds to the conjunction:

$$\bigwedge_{i\in[d]}\left((\neg\max_i(x) \wedge \bigwedge_{i<j\leq d}\max_j(x)) \to \varphi_i\right),$$

where φ_i is obtained from φ by the substitution $\text{succ}_{\text{lex}}(x) \rightsquigarrow \text{succ}_i\text{succ}_{i+1}\ldots\text{succ}_d(x)$. Similar arguments allow to get rid of $\min_{\text{lex}}$ and $\max_{\text{lex}}$. □

Remark 2.15 *In this proof, two crucial features of a structure of type* $\mathsf{pixel}^d(p)$ *are involved:*

- *its* bijective *nature, that allows to rewrite first-order formulas as cardinality formulas with a single first-order variable;*
- *the* regularity *of its predefined arithmetics (the functions* succ_i *defined on each dimension), that endows* $\mathsf{pixel}^d(p)$ *with a grid structure: it enables us to implicitly define a linear order of the whole domain* $dom(p)$ *by means of first-order formulas with a single variable, which in turn allows to express cardinality formulas by "cumulative'" arguments, via the sets* $U^{=i}$.

Proposition 2.14 straightforwardly generalizes to the numerous structures that fulfill these two properties.

To conclude this section, let us mention that we can rather easily derive from Theorem 2.6 the following additional characterization of REC^d:

Corollary 2.16 *For any* $d > 0$ *and any* d*-language* L*,*

$$L \in \text{REC}^d \Leftrightarrow \mathsf{pixel}^d(L) \in \text{ESO}(\text{var } 1).$$

3 Towards an exact logical characterization of NLIN_{ca}

3.1 Some definitions

Besides the notion of recognizable picture language, the main concept studied in this paper is the classical notion of linear time complexity on nondeterministic cellular automata of any dimension (see [4] for instance). For simplicity of notation, we only present here the notion of *one-way* d-dimensional cellular automaton, instead of the more usual notion of *two-way* d-dimensional cellular automaton, but it is known that in the nondeterministic case, the two linear-time complexity classes so defined coincide [4]. There is some technicalities in our definition of the transition function of such an automaton here below: this is due to the need to distinguish the different positions of the pixels of a picture w.r.t. its border.

Definition 3.1 *A pixel* $\mathbf{x} = (x_1, \ldots, x_d) \in [n]^d$ *is* in position $a = (a_1, \ldots, a_d) \in \{0,1\}^d$ *in a picture* $p : [n]^d \to \Gamma$ *or in the domain* $[n]^d$ *if, for all* $i \in [d]$, $a_i = max\{b \in \{0,1\} : x_i + b \leq n\}$.

We are going to define the transition function on a pixel x of a picture p according to some "neighborhood" denoted $p_{a,x}$ (it is a sub picture of p) whose domain, denoted Dom_a, depends on the position a of the pixel in the picture.

Definition 3.2 *For each* $a = (a_1, \ldots, a_d) \in \{0,1\}^d$, *let us define the* a-domain *as* $\mathsf{Dom}_a = [0, a_1] \times \cdots \times [0, a_d]$.

The a-neighborhood *of some pixel* $x \in [n]^d$ *in position* a *in a picture* $p : [n]^d \to \Gamma$ *is the picture* $p_{a,x} : \mathsf{Dom}_a \to \Gamma$ *defined as* $p_{a,x}(b) = p(x+b)$, *where* $x + b$ *denotes the sum of the vectors* x *and* b.

We denote by $\mathsf{neighb}_a(\Gamma)$ *the set of all possible a-neighborhoods on an alphabet* Γ, *that is the set of functions* $v : \mathsf{Dom}_a \to \Gamma$.

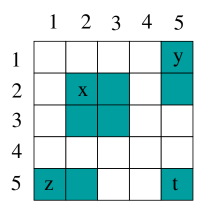

Figure 1: pixels x, y, z and t are, respectively, in position $(1,1)$, $(0,1)$, $(1,0)$, $(0,0)$. Whence their associated neighborhoods, that appear as colored cases on the figure.

Definition 3.3 *A* one-way nondeterministic d-dimensional cellular automaton *(*d-automaton*, for short) over an alphabet* Σ *is a tuple* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$, *where*

- *the finite alphabet* Γ *called the* set of states *of* $\mathcal{A}$ *includes the* input alphabet Σ *and the set* F *of* accepting states*:* $\Sigma, F \subseteq \Gamma$*;*
- δ *is the (nondeterministic)* transition function *of* $\mathcal{A}$*: it is a family of* a-transition functions $\delta = (\delta_a)_{a \in \{0,1\}^d}$ *of the form* $\delta_a : \mathsf{neighb}_a(\Gamma) \to \mathcal{P}(\Gamma)$.

Let us now define a computation.

Definition 3.4 *Let* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ *be a d-automaton and* $p, p' : [n]^d \to \Gamma$ *be two d-pictures on* Γ. *We say that* p' *is a* successor *of p for* $\mathcal{A}$, *denoted by* $p' \in \mathcal{A}(p)$, *if for each position* $a \in \{0,1\}^d$ *and each point x of position a in* $[n]^d$, $p'(x) \in \delta_a(p_{a,x})$. *The set of* j^{th}-successors *of p for* $\mathcal{A}$, *denoted by* $\mathcal{A}^j(p)$, *is defined inductively:*

$$\mathcal{A}^0(p) = \{p\} \text{ and, for } j \geq 0,\ \mathcal{A}^{j+1}(p) = \bigcup_{p' \in \mathcal{A}^j(p)} \mathcal{A}(p').$$

Definition 3.5 *A computation of a d-automaton $\mathcal{A}$ on an input d-picture $p : [n]^d \to \Gamma$ is a sequence p_1, p_2, p_3, ... of d-pictures such that $p_1 = p$ and $p_{i+1} \in \mathcal{A}(p_i)$ for each i. The picture p_i, $i \geq 1$, is called the i^{th} configuration of the computation. A computation is accepting if it is finite – it has the form $p_1, p_2, \ldots, p_k$ for some k – and the cell of minimal coordinates, $1^d = (1, \ldots, 1)$, of its last configuration is in an accepting state: $p_k(1^d) \in F$.*

Remark 3.6 *Note that the space used by a d-automaton is exactly the space (set of cells) occupied by its input d-picture.*

Definition 3.7 *Let $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ be a d-automaton and let $T : \mathbb{N} \to \mathbb{N}$ be such that $T(n) > n$. A d-picture p on Σ is accepted by $\mathcal{A}$ in time $T(n)$ if $\mathcal{A}$ admits an accepting computation of length $T(n)$ on p. That means, there exists a computation $p = p_1, p_2, \ldots, p_{T(n)} \in \mathcal{A}^{T(n)-1}(p)$ of $\mathcal{A}$ on p such that $p_{T(n)}(1^d) \in F$.*

A d-language L on Σ is accepted, or recognized, by $\mathcal{A}$ in time $T(n)$ if it is the set of d-pictures accepted by $\mathcal{A}$ in time *$T(n)$. That is $L = \{p : \exists p' \in \mathcal{A}^{T(n)-1}(p) \text{ such that } p'(1^d) \in F\}$.*

If $T(n) = cn + c'$, for some integers c, c', then L is said to be recognized in linear time and we write $L \in \mathrm{NLIN}^d_{\mathrm{ca}}$.

The time bound $T(n) > n$ of the above definition is necessary and sufficient to allow the information of any pixel of p to be communicated to the pixel of minimal coordinates, 1^d.

Remark 3.8 *The nondeterministic linear time class NLIN^d_{ca} is very robust, i.e. is not modified by many changes in the definition of the automaton or in its time bound. In particular, the constants c, c' defining the bound $T(n) = cn + c'$ can be fixed arbitrarily, provided $T(n) > n$. For example, the class NLIN^d_{ca} does not change if we take the* minimal *time $T(n) = n + 1$, called* real time*.*

3.2 A partial characterization of $\mathrm{NLIN}_{\mathrm{ca}}$

With Theorem 2.6, we stated a logical characterization of REC, the class of recognizable picture languages. The four forthcomming sections (including the present one) are devoted to the second central result of this paper: a logical characterization of $\mathrm{NLIN}_{\mathrm{ca}}$. To be precise, we will soon establish:

Theorem 3.9 *For any $d > 0$ and any d-language L,*

$$L \in \mathrm{NLIN}^d_{ca} \Leftrightarrow \mathit{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1).$$

There are many such logical characterizations of complexity classes. In most cases, the easier implication is the right-to-left one. This is not the case in the present characterization: proving that an $\mathrm{ESO}(\mathrm{var}\, d+1)$-sentence can be evaluated, over a coordinate structure taken as input, in linear time by a cellular automaton, appears to be a difficult task. This is mainly due to the "local" behaviour of cellular automata, which seems unadapted to the evaluation of an $\mathrm{ESO}(\mathrm{var}\, d+1)$-formula over a picture. Indeed, such a formula possibly connects pixels of the picture that may be arbitrarily far away from each others, and dealing with pixels that do not belong to a same neighborhood is seemingly out of the ability of our computational device.

In order to prove the right-to-left implication of Theorem 3.9, we will first have to normalize the logic under consideration, in such a way that the formulas to be evaluated are rewritten under a form that can be "handled" by a cellular automaton. We will tackle this normalization in forthcomming sections. For now, let us establish the easy part of the theorem, with Proposition 3.10 below.

Proposition 3.10 *For any $d > 0$ and any d-language L,*

$$L \in \mathrm{NLIN}_{ca}^{d} \Rightarrow \mathit{coord}^{d}(L) \in \mathrm{ESO}(\mathrm{var}\, d+1).$$

Proof. Let $L \in \mathrm{NLIN}_{\mathrm{ca}}^{d}$. By Remark 3.8, let us assume without loss of generality that L is recognized by a d-automaton $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ in time $n+1$. The sentence in $\mathrm{ESO}(\mathrm{var}\, d+1)$ that we construct is of the form $\exists (R_s)_{s\in\Gamma} \forall \mathbf{x} \forall t\, \psi(\mathbf{x}, t)$, where $\psi(\mathbf{x}, t)$ is a quantifier-free formula such that:

- ψ uses a list of exactly $d+1$ first-order variables $\mathbf{x} = (x_1, \ldots, x_d)$ and t. Intuitively, the d first ones represent the coordinates of any point in $dom(p) = [n]^d$ and the last one represents any of the first n instants $t \in [n]$ of the computation (the last instant $n+1$ is not explicitly represented);
- ψ uses, for each state $s \in \Gamma$, a relation symbol R_s of arity $d+1$. Intuitively, $R_s(a_1, \ldots, a_d, t)$ holds, for any $a = (a_1, \ldots, a_d) \in [n]^d$ and any $t \in [n]$, iff the state of cell a at instant t is s.

$\psi(\mathbf{x}, t)$ is the conjunction $\psi(\mathbf{x}, t) = \textsc{init}(\mathbf{x}, t) \wedge \textsc{step}(\mathbf{x}, t) \wedge \textsc{end}(\mathbf{x}, t)$ of three formulas whose intuitive meaning is the following.

- $\forall \mathbf{x} \forall t\, \textsc{init}(\mathbf{x}, t)$ describes the first configuration of $\mathcal{A}$, i.e. at initial instant 1, that is the input picture $p_1 = p$;
- $\forall \mathbf{x} \forall t\, \textsc{step}(\mathbf{x}, t)$ describes the computation between the instants t and $t+1$, for $t \in [n-1]$, i.e. describes the $(t+1)^{th}$ configuration p_{t+1} from the t^{th} one p_t, i.e. says $p_{t+1} \in \mathcal{A}(p_t)$;
- $\forall \mathbf{x} \forall t\, \textsc{end}(\mathbf{x}, t)$ expresses that the n^{th} configuration p_n leads to a (last) $(n+1)^{th}$ configuration $p_{n+1} \in \mathcal{A}(p_n)$ which is accepting, i.e. with an accepting state in cell 1^d: $p_{n+1}(1^d) \in F$.

Let us give explicitly these three formulas. The first one is straightforward:

$$\textsc{init}(\mathbf{x}, t) \equiv \min(t) \rightarrow \left\{ \bigwedge_{s\in\Sigma} (R_s(\mathbf{x}, t) \leftrightarrow Q_s(\mathbf{x})) \;\wedge\; \bigwedge_{s\in\Gamma\setminus\Sigma} \neg R_s(\mathbf{x}, t) \right\}$$

The second formula is

$$\textsc{step}(\mathbf{x}, t) \equiv \bigwedge_{a\in\{0,1\}^d} \bigwedge_{v\in \mathsf{neighb}_a(\Gamma)} \left\{ \begin{array}{ll} \neg\max(t) \wedge P_a(\mathbf{x}) & \wedge \\ \bigwedge_{b\in \mathsf{Dom}_a} R_{v(b)}(\mathbf{x}+b, t) & \end{array} \right\} \rightarrow \bigoplus_{s\in\delta_a(v)} R_s(\mathbf{x}, \mathrm{succ}(t)).$$

Here, $\bigoplus$ denotes the exclusive disjunction. Furthermore:

- $P_a(\mathbf{x})$ abbreviates the following formula that means that pixel $\mathbf{x}$ is in position $a = (a_1, \ldots, a_d) \in \{0,1\}^d$ in $[n]^d$: $P_a(\mathbf{x}) = \bigwedge_{i\in[d]} \sigma(a_i)\max(x_i)$, where $\sigma(1)$ is defined as $\neg$ and $\sigma(0)$ is defined as *nothing*;
- For $b = (b_1, \ldots, b_d) \in \{0,1\}^d$, $\mathbf{x}+b$ abbreviates the tuple of terms $(t_1, \ldots, t_d)$ where, for each i, the term t_i is x_i if $b_i = 0$ and is $\mathrm{succ}(x_i)$ otherwise.

It is easy to verify that the formula $\forall \mathbf{x}\, \textsc{step}(\mathbf{x}, t)$ means $p_{t+1} \in \mathcal{A}(p_t)$.

Here is the last formula.

$$\textsc{end}(\mathbf{x}, t) \equiv$$

$$\left(\max(t) \wedge \bigwedge_{i\in[d]} \min(x_i) \right) \rightarrow \left((t = x_1 \rightarrow \bigvee_{v\in N_0} R_{v(0^d)}(\mathbf{x}, t)) \wedge (t \neq x_1 \rightarrow \bigvee_{v\in N_1} \bigwedge_{b\in\{0,1\}^d} R_{v(b)}(\mathbf{x}+b, t)) \right)$$

In this formula, the sets N_0 and N_1 are defined as follows:

$N_0 = \{v \in \mathsf{neighb}_{0^d}(\Gamma) : \delta_{0^d}(v) \cap F \neq \emptyset\}$;
$N_1 = \{v \in \mathsf{neighb}_{1^d}(\Gamma) : \delta_{1^d}(v) \cap F \neq \emptyset\}$.

Let us explain the meaning of the formula $\textsc{end}(\mathbf{x}, t)$. Under the hypothesis $\max(t) \wedge \bigwedge_{i\in[d]} \min(x_i)$, the condition $t = x_1$ (resp. $t \neq x_1$) is equivalent to $n = 1$ (resp. $n > 1$). In either case, $n = 1$ or $n > 1$, the disjunction $\bigvee_{v\in N_0}$ (resp. $\bigvee_{v\in N_1}$) expresses there exists $p_{n+1} \in \mathcal{A}(p_n)$ such that $p_{n+1}(1^d) \in F$.

Hence, we have proved that for any d-picture p on Σ, the structure $\mathsf{coord}^d(p)$ satisfies the $\mathrm{ESO}(\mathrm{var}\, d+1)$-sentence $\exists (R_s)_{s\in\Gamma} \forall \mathbf{x} \forall t\, \psi(\mathbf{x}, t)$ if and only if some configuration in $\mathcal{A}^n(p)$ is accepting, i.e. $\mathcal{A}$ accepts p in time $n + 1$, or, by definition, $p \in L$. Hence, $L \in \mathrm{ESO}(\mathrm{var}\, d + 1)$, as required. □

Remark 3.11 *One notices that the obtained formula even belongs to* $\mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d + 1)$. *Actually, we prove in the next section that the logics* $\mathrm{ESO}(\mathrm{var}\, d + 1)$ *and* $\mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d + 1)$ *coincide on coordinate structures.*

The respective roles of time and space are seemingly dissymmetric in the sentence we have just constructed to express an accepting computation of a d-automaton. However, the proof and the meaning of the converse implication $L \in \mathrm{NLIN}^d_{ca} \Leftarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d + 1)$ *presented afterwards, show that in fact the d dimensions of space are – or can be made – symmetrical w.r.t. time.*

4 A first normalization of $\mathrm{ESO}(\mathrm{var}\, d)$ on coordinate structures

We now go into the normalization of $\mathrm{ESO}(\mathrm{var}\, d)$ announced after the statement of Theorem 3.9. As a first step, the present section establishes Theorem 4.1, which asserts the equivalence of the logics $\mathrm{ESO}(\mathrm{var}\, d)$ and $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ over coordinate encodings of $(d - 1)$-languages:

Theorem 4.1 $\mathrm{ESO}(\mathrm{var}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ *on* $\textsc{coord}^{d-1}$ *for any* $d > 1$. *Furthermore, this equality can be easily generalized as:* $\mathrm{ESO}(\mathrm{var}\, d') = \mathrm{ESO}(\forall^{d'}, \mathrm{arity}\, d')$ *on* $\textsc{coord}^{d-1}$ *for any* $d' \geq d > 1$.

This theorem will result from the forthcoming Proposition 4.2 and Proposition 4.4. The former states that each formula of $\mathrm{ESO}(\mathrm{var}\, d)$ can be written in such a way that its first-order part is prenex, universal, with no more than d universal quantifiers. With the latter, we rewrite each formula of $\mathrm{ESO}(\forall^d)$ in such a way that any guessed relation symbol R of the formula fulfills $\mathrm{arity}\,(R) \leq d$.

4.1 Skolemization

Proposition 4.2 *For any* $d > 0$, $\mathrm{ESO}(\mathrm{var}\, d) \subseteq \mathrm{ESO}(\forall^d)$ *on coordinate structures.*

Proof. The proof amounts to establishing that each $\mathrm{ESO}(\mathrm{var}\, d)$-formula is equivalent to an $\mathrm{ESO}(\forall^d)$-formula on any coordinate structure. Clearly, we can assume without loss of generality that the initial $\mathrm{ESO}(\mathrm{var}\, d)$-formula is first-order. So let's consider a first-order formula φ written with at most d variables. We first aim at writing φ under prenex form, *without introducing new first-order variables*. This entails introducing second-order variables existentially quantified. More precisely, the rewriting procedure is based on a depth-first traversal of the tree decomposition of φ. Each internal node of this tree corresponds to some subformula of φ of arity k – say $\theta(x_1, \ldots, x_k)$ –, and gives rise to a new relation symbol R_θ of the same arity. This relation is forced to encode the set $\{\mathbf{x} \text{ s.t. } \theta(\mathbf{x})\}$ *via* a formula $\mathrm{def}_\theta(R_\theta)$ defined as follows:

- If $\theta \equiv Qy\theta'(\mathbf{x}, y)$ where Q is a quantifier, then $\text{def}_\theta(R_\theta) \equiv \forall \mathbf{x} : R_\theta(\mathbf{x}) \leftrightarrow Qy\theta'(\mathbf{x}, y)$
- If $\theta \equiv \theta'(\mathbf{x}) \circ \theta''(\mathbf{x})$ for some connective $\circ$, then $\text{def}_\theta(R_\theta) \equiv \forall \mathbf{x} : R_\theta(\mathbf{x}) \leftrightarrow (\theta'(\mathbf{x}) \circ \theta''(\mathbf{x}))$.

If θ has no free variables, the relation symbol R_θ is choosen with arity 1 and its definition is written either $\forall x : R_\theta(x) \leftrightarrow Qy\theta(y)$ or $\forall x : R_\theta(x) \leftrightarrow (\theta' \circ \theta'')$, according to the form of θ. Here, x is any variable of φ distinct from y. Each time a node $\theta(\mathbf{x})$ has been visited, the corresponding R_θ and def_θ are generated and φ is updated by the substitution $\theta(\mathbf{x}) \rightsquigarrow R_\theta(\mathbf{x})$. Then, the procedure is run recursively on the formula so obtained.

Let us illustrate this procedure by running it on the first-order formula with three variables:

$$\varphi \equiv \exists x\,(\,\forall y \exists z\, U(x, y, z) \vee \exists y\, D(x, y)\,) \rightarrow \forall y\,(\,D(y, y) \vee \exists x\, U(x, y, x)\,)\,. \tag{7}$$

We merely display the definition formulas generated by the procedure, along with the relation symbols $R_1, \ldots, R_9$ corresponding to the nine internal nodes of φ. The successive updates of φ are implicit.

$$\begin{array}{llcl}
\text{def}_1(R_1) \equiv \forall x, y : & R_1(x, y) & \leftrightarrow & \exists z\, U(x, y, z) \\
\text{def}_2(R_2) \equiv \forall x : & R_2(x) & \leftrightarrow & \forall y\, R_1(x, y) \\
\text{def}_3(R_3) \equiv \forall x : & R_3(x) & \leftrightarrow & \exists y\, D(x, y) \\
\text{def}_4(R_4) \equiv \forall x : & R_4(x) & \leftrightarrow & (R_2(x) \vee R_3(x)) \\
\text{def}_5(R_5) \equiv \forall y : & R_5(y) & \leftrightarrow & \exists x\, R_4(x) \\
\text{def}_6(R_6) \equiv \forall y : & R_6(y) & \leftrightarrow & \exists x\, U(x, y, x) \\
\text{def}_7(R_7) \equiv \forall y : & R_7(y) & \leftrightarrow & (D(y, y) \vee R_6(y)) \\
\text{def}_8(R_8) \equiv \forall x : & R_8(x) & \leftrightarrow & \forall y\, R_7(y) \\
\text{def}_9(R_9) \equiv \forall x : & R_9(x) & \leftrightarrow & (R_5(x) \rightarrow R_8(x))
\end{array}$$

Now, our initial formula can be rewritten:

$$\varphi \equiv \exists R_1, \ldots, R_9 : \left(\bigwedge_{1 \le i \le 9} \text{def}_i(R_i) \right) \wedge \forall x\, R_9(x). \tag{8}$$

Notice that for each i, either def_i is prenex and universal, or it has the form $\forall \mathbf{u} : \alpha(\mathbf{u}) \leftrightarrow Qv\beta(\mathbf{u}, v)$. It is easily seen that this last form is equivalent to the conjunction:

$$\forall \mathbf{u} Qv(\alpha(\mathbf{u}) \rightarrow \beta(\mathbf{u}, v)) \;\wedge\; \forall \mathbf{u} Q^* v(\beta(\mathbf{u}, v) \rightarrow \alpha(\mathbf{u})),$$

where Q^* is $\forall$ if Q is $\exists$ and *vice versa*. Therefore, following Equation (8), φ can now be written as a conjunction of prenex formulas, each of which involves no more than three variables and has a quantifier prefix of the shape $\forall \mathbf{x}$ or $\forall \mathbf{x} \exists y$. In order to write this conjunction under prenex form without adding new first-order variables, we have to "replace" existential quantifiers by universal ones. Afterward φ, as a conjunction of formulas of the type $\forall x, y, z\theta$, could be written under the requisite shape. We show below how to deal with this specific formula. The general case is strictly similar.

To get rid of existential quantifiers occuring in (some of) the def_i's, we will invoke the arithmetical symbols of the signature of φ. Remember that the conjuncts that are not still universal all have the form $\forall \mathbf{x} \exists y \theta(\mathbf{x}, y)$, where $\mathbf{x}$ and y are tuples of first-order variables of respective arities k and 1. The predefined arithmetics included in coordinate signatures allow to defining, for any such conjonct, a relation of arity $k+1$ that witnesses the existence of some y fulfilling $\theta(\mathbf{x}, y)$ for a given $\mathbf{x}$. This idea is completed as follows: Let W be a new $(k+1)$-ary relation symbol associated with $\exists y \theta(\mathbf{x}, y)$. We want the assertion $\mathrm{W}(\mathbf{x}, y)$ to signify that there exists $z \le y$ such that $\theta(\mathbf{x}, z)$ holds. This interpretation is achieved thanks to the following formula:

$$\forall \mathbf{x} \forall y \left\{ \begin{array}{ll} & \min(y) \rightarrow (\mathrm{W}(\mathbf{x}, y) \leftrightarrow \theta(\mathbf{x}, y)) \\ \wedge & \mathrm{W}(\mathbf{x}, \mathrm{succ}(y)) \leftrightarrow (\theta(\mathbf{x}, \mathrm{succ}(y)) \vee \mathrm{W}(\mathbf{x}, y)) \end{array} \right\}$$

We denote by $\mathrm{W} = \mathsf{witness}(\exists y \theta)$ this last formula. When it is satisfied, the assertion $\forall \mathbf{x} \exists y \theta(\mathbf{x}, y)$ is clearly equivalent to $\forall \mathbf{x} \forall y : \max(y) \to \mathrm{W}(\mathbf{x}, y)$.

For instance, the above formula $\mathrm{def}_1(R_1)$ gives rise to the non-universal formula $\mathrm{def}_1^1(R_1) \equiv \forall x, y \exists z : R_1(x,y) \to U(x,y,z)$. This shoud be managed as follows: A ternary relation symbol W_1 is introduced (i.e., existentially quantified) and compelled to fit its intended interpretation *via* the formula: $\mathrm{W}_1 = \mathsf{witness}(\delta_1)$, where $\delta_1 \equiv \exists z : R_1(x,y) \to U(x,y,z)$. Afterwards, the formula $\mathrm{def}_1^1(R_1)$ is replaced by $\forall x, y, z : \max(z) \to \mathrm{W}_1(x,y,z)$. When this task has been achieved for each non universal formula def_i, the formula displayed in (8) becomes:

$$\exists (R_i)_{i \in I}\ \exists (\mathrm{W}_j)_{j \in J} : \left(\bigwedge_{j \in J} \mathrm{W}_j = \mathsf{witness}(\delta_j) \right) \wedge \left(\bigwedge_{i \in I} \mathrm{def}_i \right) \wedge \forall x\, R_\ell(x).$$

Here, $I = \{1, \ldots, 9\}$, $J = \{1, 3, 5, 6\}$ (the $j \in J$ correspond to formulas def_j that are non-universal), $\ell = 9$, and δ_j is the existential part of the (old) formula def_j. Clearly, this formula can be written in $\mathrm{ESO}(\forall^d)$ for $d = 3$. □

4.2 Arity vs number of first-order variables

We prove here a normalization of the logic $\mathrm{ESO}(\forall^d)$, similar to that of Proposition 1.7. This proof involves the following easy fact:

Fact 4.3 *Suppose we are given a family of functions $(f_i : X_i \to Y)_{i \in \mathcal{I}}$ and a family of relations $(R_i \subseteq X_i)_{i \in \mathcal{I}}$, indexed by the same finite set $\mathcal{I}$. The following assertions are equivalent:*

- (i) *$\forall i, j \in \mathcal{I}$, $\forall x \in X_i$, $\forall y \in X_j$: $f_i(x) = f_j(y) \Rightarrow R_i(x) = R_j(y)$;*
- (ii) *$\exists R \subseteq Y$ such that $\forall i \in \mathcal{I}$, $\forall x \in Y$: $R_i(x) = R(f_i(x))$.*

Proof. $(ii) \Rightarrow (i)$ is clear. For the converse implication, we define R on each $f_i\langle X_i \rangle$ by: $\forall x \in X_i$, $R(f_i(x)) = R_i(x)$. The hypothesis (i) guarantees the coherence of this definition. To complete it, we set $R(x) = 0$ for every $x \in Y \setminus \bigcup_{i \in \mathcal{I}} f_i\langle X_i \rangle$. This relation R clearly witnesses to condition (ii). □

Proposition 4.4 $\mathrm{ESO}(\forall^d) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d)$ *on* $\textsc{coord}^{d-1}$ *for any $d > 1$.*

Proof. Given $\Phi \in \mathrm{ESO}(\forall^d)$, we want to build a formula in $\mathrm{ESO}^\sigma(\forall^d, \text{arity } d)$ equivalent to Φ on $\textsc{coord}^{d-1}$. To fix ideas, let us assume that Φ has the very simple shape:

$$\Phi \equiv \exists R \forall x_1, \ldots, x_d \varphi(\mathbf{x}, R, \sigma), \tag{9}$$

where R is a *single* k-ary relation symbol for some $k > d$, and φ is a quantifier free formula. The formula to be built must have the form:

$$\Psi \equiv \exists \rho \forall x_1, \ldots, x_d \psi(\mathbf{x}, \rho, \sigma),$$

where ρ is a tuple of d-ary relation symbols and ψ is quantifier free.

The substitution of d-ary symbols for R rests in the limitation of the number of first-order variables in Φ: each atomic formula involving R has the form $R(t_1, \ldots, t_k)$ where the t_i's are terms built on $x_1, \ldots, x_d$. Therefore, although R is k-ary, *in each of its occurrences* it behaves as a d-ary symbol, dealing with the sole variables $x_1, \ldots, x_d$. Hence, the key is to create a d-ary symbol for each occurrence of R in Φ or, more precisely, for each k-tuple of terms $(t_1, \ldots, t_k)$ involved in a R-atomic formula.

More formally, let us denote by $\mathsf{T}(\Phi)$ the set of terms occuring in Φ, and by $\mathsf{T}_R(\Phi)$ the set of tuples of terms involved in a R-atomic subformula of Φ. That is, each element of $\mathsf{T}_R(\Phi)$ is a k-tuple

$$\mathbf{t}(\mathbf{x}) = (t_1(x_1, \ldots, x_d), \ldots, t_k(x_1, \ldots, x_d)) \in \mathsf{T}(\Phi)^k$$

such that the formula $R(\mathbf{t}(\mathbf{x}))$ appears in Φ. For each $\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)$, consider a new d-ary relation symbol $R_{\mathbf{t}(\mathbf{x})}$. Now, consider a σ-stucture S of domain $[n]$, and denote by $\langle S, R\rangle$ some expansion of S to $\sigma \cup \{R\}$. (That is, we denote by R both the relationnal symbol and its interpretation on $[n]$.) Furthermore, fix the S-interpretation of each $R_{\mathbf{t}(\mathbf{x})}$, for $\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)$, by

$$\forall \mathbf{x} \in [n]^d : R_{\mathbf{t}(\mathbf{x})}(\mathbf{x}) = R(\mathbf{t}(\mathbf{x})), \tag{10}$$

and denote by $\mathbf{R}$ the tuple $(R_{\mathbf{t}(\mathbf{x})})_{\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)}$ thus defined. Then clearly:

$$\langle S, R\rangle \models \forall \mathbf{x} \varphi(\mathbf{x}, R, \sigma) \Leftrightarrow \langle S, \mathbf{R}\rangle \models \forall \mathbf{x} \tilde{\varphi}(\mathbf{x}, \mathbf{R}, \sigma), \tag{11}$$

where $\tilde{\varphi}$ is obtained from φ by substituing the formula $R_{\mathbf{t}(\mathbf{x})}(\mathbf{x})$ for each occurrence of the formula $R(\mathbf{t}(\mathbf{x}))$. Before continuing with this proof, let's illustrate the previous definitions with a simple example.

Example. Assume Φ is the following ESO($\forall^2$)-formula:

$$\exists R \forall x, y \varphi(x, y, R), \text{ where } \varphi \equiv R(x, y, x) \wedge \neg R(y, x, y).$$

According to the notations used so far, we have:

$$d = 2,\ k = 3,\ \mathsf{T}(\Phi) = \{x, y\} \text{ and } \mathsf{T}_R(\Phi) = \{(x, y, x), (y, x, y)\}.$$

The binary relation symbols associated to terms $\mathbf{t}(x, y) \in \mathsf{T}_R(\Phi)$ are denoted $R_{(x,y,x)}$ and $R_{(y,x,y)}$. The formula $\tilde{\varphi}$ obtained from φ, following (11), is written:

$$\tilde{\varphi} \equiv R_{(x,y,x)}(x, y) \wedge \neg R_{(y,x,y)}(x, y).$$

If, for any interpretation of R on $[n]$, we fix the interpretations of $R_{(x,y,x)}$ and $R_{(y,x,y)}$ as in (10):

$$\forall a, b < n : R_{(x,y,x)}(a, b) \Leftrightarrow R(a, b, a) \text{ and } R_{(y,x,y)}(a, b) \Leftrightarrow R(b, a, b),$$

then it is easily seen that:

$$\langle S, R\rangle \models \forall x, y \varphi(x, y, R, \sigma) \Leftrightarrow \langle S, \mathbf{R}\rangle \models \forall x, y \tilde{\varphi}(x, y, R_{(x,y,x)}, R_{(y,x,y)}, \sigma).$$

◃

Let's come back to the proof of Proposition 4.4. Equations (10) and (11) yield:

$$S \models \exists R \forall \mathbf{x} \varphi(\mathbf{x}, R, \sigma) \Rightarrow S \models \exists \mathbf{R} \forall \mathbf{x} \tilde{\varphi}(\mathbf{x}, \mathbf{R}, \sigma), \tag{12}$$

where $\mathbf{R}$ is a tuple of d-ary relation symbols indexed by $\mathsf{T}_R(\Phi)$, say $(R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}$. Unfortunately, the converse implication does not hold in general. For instance, one can check in Example 4.2 above, that the formula $\exists R_{(x,y,x)} \exists R_{(y,x,y)} \forall x \forall y \tilde{\varphi}$ has a model, while $\exists R \forall x \forall y \varphi$ doesn't have. To get the right-to-left implication in (12), we have to strengthen the hypothesis with some assertion that compels the tuple $(R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}$ to be, in some sense, the d-ary representation of some k-ary relation. All in all, we confront the following question: *Given a σ-structure S, a set $T \subset \mathsf{T}(\Phi)^k$ and a family $(R_{\mathbf{t}})_{\mathbf{t}\in T}$ of d-ary relations over the domain $[n]$ of S, what are the conditions on $(R_{\mathbf{t}})_{\mathbf{t}\in T}$ that ensure*

$$\exists R \subseteq [n]^k \text{ such that } \forall \mathbf{t} \in T, \forall \mathbf{a} \in [n]^d : \; R_{\mathbf{t}}(\mathbf{a}) = R(\mathbf{t}(\mathbf{a})) \tag{13}$$

Each k-tuple $\mathbf{t} \in T$ defines a function from $[n]^d$ to $[n]^k$, *via* the process of interpretation of terms.[4] Therefore, if we set, in the statement of Fact 4.3:

$$X_i = [n]^d \text{ for each } i, \; Y = [n]^k \text{ and } (f_i)_{i\in\mathcal{I}} = (\mathbf{t})_{\mathbf{t}\in T},$$

we get the equivalence of (13) with the following assertion:

$$\forall \mathbf{t}, \mathbf{t}' \in T, \; \forall \mathbf{a}, \mathbf{a}' \in [n]^d : \; \mathbf{t}(\mathbf{a}) = \mathbf{t}'(\mathbf{a}') \Rightarrow R_{\mathbf{t}}(\mathbf{a}) = R_{\mathbf{t}'}(\mathbf{a}'),$$

which is logically translated into the formula:

$$\bigwedge_{\mathbf{t},\mathbf{t}'\in T} \forall \mathbf{x}, \mathbf{x}' : \; \mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}') \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')), \tag{14}$$

where $\mathbf{x}$, $\mathbf{x}'$ are d-tuples of first-order variables.

In order to express condition (14) in the required formalism, it remains to reduce the number of quantifiers (remember our logic allows only d universal first-order quantifiers). Since the conjunction and the universal quantifier commute, we just have to tackle the case of a single conjunct

$$\forall \mathbf{x}, \mathbf{x}' : \; \mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}') \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')). \tag{15}$$

To process, we invoke the specificity of coordinate encodings, which has not been involved in our reasonning so far. Since succ is the only function symbol in the underlying signature σ, all the terms under consideration have the form $\mathrm{succ}^i(u)$ for some $i \in \mathbb{N}$ and some first-order variable u. Hence, the equality $\mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}')$ is actualy a conjunction of k atomic formulas of the type $\mathrm{succ}^a(x) = \mathrm{succ}^b(y)$, with $x \in \mathbf{x}$, $y \in \mathbf{x}'$ and $a, b \geq 0$. Assume, to fix ideas, that $a \geq b$. Since the successor function is not cyclic, we have:

$$\mathrm{succ}^a(x) = \mathrm{succ}^b(y) \Leftrightarrow \left(x = \mathrm{succ}^{a-b}(y) \vee \mathrm{succ}^a(x) = \mathrm{succ}^b(y) = \max\right).$$

Furthermore, $\mathrm{succ}^a(x) = \max \Leftrightarrow \bigvee_{i\leq a} x = \mathrm{pred}^i(\max)$, and hence the equality $\mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}')$ can finally be written as a conjunction of k formulas φ_ℓ, each of which involves exactly two variables and has the form

$$x = \mathrm{succ}^i(y) \vee \bigvee_{i\leq a} \bigvee_{j\leq b} \left(x = \mathrm{pred}^i(\max) \wedge y = \mathrm{pred}^j(\max)\right).$$

Writing the conjunction $\bigwedge_{\ell\leq k} \varphi_\ell$ under disjunctive normal form, we get a formula of the type: $\bigvee_{i\leq m} \bigwedge_{j\leq k} \theta_{ij}$ where each θ_{ij} has either the form $x = \mathrm{succ}^i(y)$ or $x = \mathrm{pred}^i(\max)$. Thus, we can rewrite the formula displayed in (15) as:

$$\forall \mathbf{x}, \mathbf{x}' : \left(\bigvee_{i\leq m} \bigwedge_{j\leq k} \theta_{ij} \right) \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')).$$

[4] For instance, the triple of terms $\mathbf{t} = (\mathrm{succ}^3 x, x, \mathrm{succ}^2 y)$ maps each couple $(a, b) \in [n]^2$ onto the triple $(a+3, a, b+2) \in [n]^3$, where $+$ is the addition modulo n.

Or also, as:

$$\bigwedge_{i\le m} \forall \mathbf{x},\mathbf{x}' \left(\bigwedge_{j\le k} \theta_{ij} \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')) \right). \tag{16}$$

We saw that each θ_{ij} has either the form $x = \mathrm{succ}^i(y)$ or $x = \mathrm{pred}^i(\max)$. Hence, we can eliminate one variable for each θ_{ij} by rewriting each subformula $\forall \mathbf{x},\mathbf{x}' \left(\bigwedge_{j\le k} \theta_{ij} \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')) \right)$ as $\forall \mathbf{x},\mathbf{x}' \psi_i$, where ψ_{ij} is the formula $R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')$ in which:

- each variable x such that $x = \mathrm{succ}^i(y)$ occurs in $\bigwedge_{j\le k} \theta_{ij}$ is replaced by $\mathrm{succ}^i(y)$;
- each variable x such that $x = \mathrm{pred}^i(\max)$ occurs in $\bigwedge_{j\le k} \theta_{ij}$ is replaced by $\mathrm{pred}^i(\max)$.

The reader will easily verify that this reasonning can be generalized. Thus, each conjunct of (14) can be rewritten with only d universal first-order variables. These rewritings result in a new first-order formula $k2d$-Rep, of signature $\sigma \cup \{R_{\mathbf{t}}, \mathbf{t} \in T\}$, equivalent to (14) and with quantifier prefix $\forall \mathbf{x}$, where $\mathbf{x}$ is a d-tuple of first-order variables. Clearly, the starting formula $\Phi \in \mathrm{ESO}(\forall^d, \mathrm{arity}\, k)$ considered in (9):

$$\Phi \equiv \exists R \forall x_1, \ldots, x_d \varphi(\mathbf{x}, R, \sigma)$$

is equivalent on picture encodings to the following $\Psi \in \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$:

$$\Psi \equiv \exists (R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)} : k2d\text{-Rep}((R_{\mathbf{t}})_{\mathbf{t}}, \sigma) \ \wedge \forall \mathbf{x} \tilde{\varphi}(\mathbf{x}, (R_{\mathbf{t}})_{\mathbf{t}}, \sigma),$$

where $\tilde{\varphi}$ is obtained from φ by replacing each $R(\mathbf{t}(\mathbf{x}))$ by $R_{\mathbf{t}}(\mathbf{x})$. This is the sought after formula. It remains to check that this procedure can be extended to any number of relation symbols existentially quantified in Φ. This is immediate. ◻

Theorem 4.1 immediatly proceeds from Propositions 4.2 and 4.4 above. We will remember that:

Each ESO(var d)-formula of signature σ can be writen:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(\mathrm{succ}^i(x)),\ \max(\mathrm{succ}^i(x)), \\ \mathrm{succ}^i(x) = \mathrm{succ}^j(y), \\ Q_a(\mathrm{succ}^{i_1}(x_{j_1}), \ldots, \mathrm{succ}^{i_p}(x_{j_p}), \\ R(\mathrm{succ}^{i_1}(x_{j_1}), \ldots, \mathrm{succ}^{i_p}(x_{j_p}) \end{array} \right\} \tag{17}$$

where $Q_a \in \sigma$, $R \in \mathbf{R}$ and x, y and the x_i's are all components of $\mathbf{x}$.

5 "Localization" of existentially quantified relations

This section is dedicated to the proof of the normalization of $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ on coordinate encodings of $(d-1)$-pictures, for $d \ge 2$. Our purpose is to rewrite $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$-formulas into equivalent formulas that fulfill the "sorted" or "local" property mentioned after the statement of Theorem 3.9. Before formalizing our notion of sorted formulas, let us detail what is its intended meaning. We want to deal with pixels of time-space diagram of the computation that are adjacents, that is, that are both connected and differ (by one) by at most one dimension. Two such pixels are represented by d-tuples of the form $\mathbf{x}$ and $\mathbf{x}^{(i)}$, that is:

- their components are in the same order (elsewhere they could be disconnected);

- there is at most one occurrence of succ (elsewhere, they would differ of more than one dimension).

These requirements (and a little more) are formalized in the following definition.

Definition 5.1 *Let k, d be two integers such that $d \geq k \geq 1$. A sentence over coordinate structures for k-pictures is in* $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$ *if it is of the form* $\exists \mathbf{R} \forall \mathbf{x} \psi(\mathbf{x})$ *where*

- $\mathbf{R}$ *is a list of relation symbols of arity d;*
- ψ *is a quantifier-free formula whose list of first-order variables is* $\mathbf{x} = (x_1, \ldots, x_d)$*;*
- *each atom of ψ is of one of the following forms:*
  - (i) $Q_s(x_1, \ldots, x_k)$*, for* $s \in \Sigma$*,*
  - (ii) $R(\mathbf{x})$ *or* $R(\mathbf{x}^{(i)})$ *where* $R \in \mathbf{R}$*,* $i \in [d]$*, and* $\mathbf{x}^{(i)}$ *is the tuple* $\mathbf{x}$ *where* x_i *is replaced by* $succ(x_i)$*,*
  - (iii) $min(x_i)$ *or* $max(x_i)$*, for* $i \in [d]$*.*

We prove the normalization $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$ for $(d-1)$-pictures (i.e., for $k = d-1$). In the present section, we deal with Condition (ii) of the above definition (see Proposition 5.9). In Subsection 5.1, we eliminate equalities and inequalities. At this point, we get a normalization of $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ into the so-called "half-sorted logic", denoted by $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted})$. It remains to manage the input relation symbols; this is done in Section 6, where we tackle Condition (i).

To lighten the presentation of the forthcomming results, we first introduce some notations about tuples and permutations.

Definition 5.2 *Let $n, d > 0$ and* $\mathbf{x} \in [n]^d$*.*

- *We denote by* $[\mathbf{x}]_i$ *the i^{th} component of* $\mathbf{x}$*. E.g.* $(5, 7, 2)_2 = 7$*.*
- *We say that $\mathbf{x}$ is non decreasing, and we write* $\mathbf{x}\!\uparrow$*, when* $[\mathbf{x}]_1 \leq \cdots \leq [\mathbf{x}]_d$*.*
- $\mathcal{S}(d)$ *stands for the set of permutations of* $\{1, \ldots, d\}$*. Given* $\alpha_1, \ldots, \alpha_d \in \{1, \ldots, d\}$*, we denote by* $\alpha_1 \ldots \alpha_d$ *the permutation* $\alpha \in \mathcal{S}(d)$ *that maps each i on α_i. Conversely, for* $\alpha \in \mathcal{S}(d)$ *we set* $\alpha_i := \alpha(i)$*. By* $\mathcal{T}(d)$ *we denote the set of transpositions over* $\{1, \ldots, d\}$*.*
- *If* $\alpha \in \mathcal{S}(d)$ *and* $\mathbf{x} = (x_1, \ldots, x_d)$*, we denote by* $\mathbf{x}_\alpha$ *the d-tuple* $(x_{\alpha_1}, \ldots, x_{\alpha_d})$*. It is less ambiguous to define* $\mathbf{x}_\alpha$ *by the assertion:*

  for any $i \in \{1, \ldots, d\}$, $[\mathbf{x}_\alpha]_i = [\mathbf{x}]_{\alpha(i)}$.

  Thus, if β also belongs to $\mathcal{S}(d)$*, we get* $[(\mathbf{x}_\alpha)_\beta]_i = [\mathbf{x}_\alpha]_{\beta(i)} = [\mathbf{x}]_{\alpha\beta(i)}$*. Whence the identity:* $(\mathbf{x}_\alpha)_\beta = \mathbf{x}_{\alpha\beta}$*. In particular,* $(\mathbf{x}_\alpha)_{\alpha^{-1}} = \mathbf{x}$*.*
- *For* $\alpha \in \mathcal{S}(d)$ *and* $n > 0$*, we set* $[\alpha] = \{\mathbf{x} \in [n]^d \text{ s.t. } \mathbf{x}_\alpha\!\uparrow\}$*. In particular, denoting by* id *the identity on* $\{1, \ldots, d\}$*, we get* $\mathbf{x} \in [\mathsf{id}]$ *iff* $\mathbf{x}\!\uparrow$*. Therefore,* $\mathbf{x} \in [\alpha]$ *iff* $\mathbf{x}_\alpha \in [\mathsf{id}]$*. Clearly,* $[n]^d = \bigcup_{\alpha \in \mathcal{S}(d)} [\alpha]$*.*
- *For any i in* $\{1, \ldots, d\}$*,* $\mathbf{x}^{(i)}$ *denotes the tuple obtained from $\mathbf{x}$ by replacing its i^{th} component by its own successor. That is, if* $\mathbf{x} = (x_1, \ldots, x_d)$ *then*

  $$\mathbf{x}^{(i)} = (x_1, \ldots, x_{i-1}, succ(x_i), x_{i+1}, \ldots, x_d).$$

  As previously, the arrangement of $\mathbf{x}^{(i)}$ *according to some permutation α is denoted by* $(\mathbf{x}^{(i)})_\alpha$*.*

Example. Consider $\mathbf{x} = (5,3,7,2)$ in $[9]^4$ and $\alpha = 4213$, $\beta = 1432$ in $\mathcal{S}(4)$. Then $\mathbf{x}_\alpha = (2,3,5,7)$ is non-decreasing while $\mathbf{x}_\beta = (5,2,7,3)$ is not. Besides, $\mathbf{x}^{(3)} = (5,3,8,2)$ while $(\mathbf{x}_\alpha)^{(3)} = (2,3,6,7)$ and $(\mathbf{x}^{(3)})_\alpha = (2,3,5,8) = (\mathbf{x}_\alpha)^{(4)}$. ◁

With these notations, the request described at the beginning of the section can be rephrased as follows: we want to normalize ESO($\forall^d$, arity d)-formulas in such a way that each atomic subformula $R(t_1(\mathbf{x}),\dots,t_p(\mathbf{x}))$ built with a guessed relation symbol R has either the form $R(\mathbf{x})$ or the form $R(\mathbf{x}^{(i)})$.

Fact 5.3 *On* COORD^{d-1},[5] *any formula* $\Phi = \exists\mathbf{R}\forall\mathbf{x}\varphi(\mathbf{x},\mathbf{R},\sigma) \in \text{ESO}(\forall^d, \text{arity } d)$ *of signature* σ *can be written in such a way that:*

(a) *In each atomic subformula* $R(t_1,\dots,t_p)$ *of* φ, $Var(t_i)\cap Var(t_j) = \emptyset$ *for every* $1 \le i < j \le p$.

(b) *Each* R *in* $\mathbf{R}$ *has arity* d *exactly.*

Proof. The proof of (a) is quite immediate. We illustrate it with an example: assume that Φ involves the subformula $R(\text{succ}^2 x_1, x_2, \text{succ}\, x_1, \text{succ}^3 x_2)$ for some $R \in \sigma\cup\{\mathbf{R}\}$. Then clearly, Φ is equivalent, on picture-structures, to:

$$\exists\mathbf{R}\exists A\forall\mathbf{x} : \tilde{\varphi}(\mathbf{x},\mathbf{R},A,\sigma)\ \wedge (x_1 = \text{succ}\, x_3 \wedge x_4 = \text{succ}^3 x_2) \to (A(x_2,x_3) \leftrightarrow R(x_1,x_2,x_3,x_4))$$

where $\tilde{\varphi}$ is obtained from φ by substituting the formula $A(x_2,\text{succ}(x_1))$ to each occurrence of the atom $R(\text{succ}^2 x_1, x_2, \text{succ}\, x_1, \text{succ}^3 x_2)$.

In order to prove (b), assume for simplicity that $\mathbf{R}$ reduces to the single relational symbol R of arity $p < d$. The idea is to add $d-p$ dummy arguments to R. Clearly, Φ is equivalent to the formula:

$$\exists R:\ \tilde{\varphi} \wedge \forall\mathbf{x} \textstyle\bigwedge_{p<i\le d} R(x_1,\dots,x_i,\dots,x_d) \leftrightarrow R(x_1,\dots,\text{succ}(x_i),\dots,x_d),$$

where $\tilde{\varphi}$ is obtained from φ by replacing each atomic subformula $R(t_1,\dots,t_p)$ by $R(t_1,\dots,t_p,x_{i_1},\dots,x_{i_{d-p}})$. Here, $x_{i_1},\dots,x_{i_{d-p}}$ is the complete list of distinct variables among $\mathbf{x}$ that do not occur in $t_1,\dots,t_p$. □

Remark 5.4 *The proof of Fact 5.3 allows enhancing its statement: each atomic subformula of* $\tilde{\varphi}$ *that involves an input symbol* Q_a, *for some* $a \in \Sigma$, *has the form* $Q_a(x_{\alpha_1},\dots,x_{\alpha_{d-1}})$ *where* α *is an injection from* $\{1,\dots,d-1\}$ *into* $\{1,\dots,d\}$.

Fact 5.5 *On* COORD^{d-1}, *any formula* $\Phi = \exists\mathbf{R}\forall\mathbf{x}\varphi(\mathbf{x},\mathbf{R},\sigma) \in \text{ESO}(\forall^d, \text{arity } d)$ *of signature* σ *can be written in such a way that each atomic subformula over* $\mathbf{R}$ *of* φ *has one of the two forms:* $R(\mathbf{x}^{(i)})$ *or* $R(\mathbf{x}_\pi)$, *where* $\pi \in \mathcal{S}(d)$.

Proof. We prove the result for $d = 3$. The general case is similar. Let ℓ be the maximal value of an $i \in \mathbb{N}$ such that $\text{succ}^i(x)$ occurs in Φ, for any $x \in \mathbf{x}$. For each $R \in \mathbf{R}$, we introduce new d-ary relation symbols $R_{i,j,k}$ for every $i,j,k \le \ell$. We want to force the following interpretations of the $R_{i,j,k}$'s:

$$R_{i,j,k}(u_1,u_2,u_3) = R(\text{succ}^i u_1, \text{succ}^j u_2, \text{succ}^k u_3).$$

This is done inductively, with the formulas:

- $\forall\mathbf{x} : R_{0,0,0}(x_1,x_2,x_3) \leftrightarrow R(x_1,x_2,x_3)$

[5] We denote by COORD^{d-1} the sets of coordinate encodings of $(d-1)$-pictures.

- $\forall \mathbf{x} : \bigwedge_{i<\ell} \bigwedge_{j,k\leq\ell} (R_{i+1,j,k}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(\mathrm{succ}(x_1),x_2,x_3))$
- $\forall \mathbf{x} : \bigwedge_{j<\ell} \bigwedge_{i,k\leq\ell} (R_{i,j+1,k}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(x_1,\mathrm{succ}(x_2),x_3))$
- $\forall \mathbf{x} : \bigwedge_{k<\ell} \bigwedge_{i,j\leq\ell} (R_{i,j,k+1}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(x_1,x_2,\mathrm{succ}(x_3)))$

Factorizing the quantifications and using notations of Definition 5.2, the conjunction of these formulas can be written:

$$\forall \mathbf{x} \left\{ \begin{array}{l} R_{0,0,0}(\mathbf{x}) \leftrightarrow R(\mathbf{x}) \wedge \\ \bigwedge_{i<\ell} \bigwedge_{j,k\leq\ell} (R_{i+1,j,k}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(1)})) \wedge \\ \bigwedge_{j<\ell} \bigwedge_{i,k\leq\ell} (R_{i,j+1,k}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(2)})) \wedge \\ \bigwedge_{k<\ell} \bigwedge_{i,j\leq\ell} (R_{i,j,k+1}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(3)})) \end{array} \right\}$$

Let us denotes by $\mathsf{decomp}(R,(R_{i,j,k})_{i,j,k\leq\ell})$ this last formula. It clearly fulfills the condition of the statement. Now, consider the formula

$$\exists \mathbf{R} \exists ((R_{i,j,k})_{i,j,k\leq\ell})_{R\in\mathbf{R}} : \bigwedge_{R\in\mathbf{R}} \mathsf{decomp}(R,(R_{i,j,k})_{i,j,k\leq\ell}) \ \wedge \ \forall \mathbf{x} \tilde{\varphi}, \tag{18}$$

where $\tilde{\varphi}$ is obtained from φ by the substitutions

$$R(\mathrm{succ}^i x_{\alpha_1}, \mathrm{succ}^j x_{\alpha_2}, \mathrm{succ}^k x_{\alpha_3}) \rightsquigarrow R_{i,j,k}(x_{\alpha_1}, x_{\alpha_2}, x_{\alpha_3}).$$

Then, the formula (18) is equivalent to Φ and also fits the requirements of Fact 5.5. It is the rewriting of Φ announced. □

As a result of Fact 5.3, Remark 5.4 and Fact 5.5, each ESO($\forall^d$, arity d)-formula of signature σ has a conjonctive normal form of the shape:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(\mathrm{succ}^i(x)),\ \max(\mathrm{succ}^i(x)), \\ \mathrm{succ}^i(x) = \mathrm{succ}^j(y), \\ Q_a(\mathbf{x}_\iota),\ R(\mathbf{x}_\beta),\ R(\mathbf{x}^{(i)}) \end{array} \right\}$$

Furthermore, the trick used in the proof of Fact 5.5 also allows writing atoms involving min, max or equalities, under the form $\max(x)$, $\min(x)$ and $x = y$ for some $x, y \in \mathbf{x}$.

It remains to prove that we can get rid of the atomic formulas $R(\mathbf{x}_\beta)$, where $\beta \neq \mathsf{id}$. This part is rather technical, so we provide some preliminary explanations before stating the logical framework which allows the normalization. In order to get rid of each literal of the form $R(x_\beta)$, we will divide the set $R \subseteq [n]^d$ in $d!$ relations $R_\beta \subseteq [n]^d$, each corresponding to a given permutation β of $\{1,\ldots,d\}$.

Definition 5.6 *For $R \subseteq [n]^d$ and for each $\alpha \in \mathcal{S}(d)$, we define a d-ary relation R_α on $[n]$ by: $R_\alpha = \{\mathbf{x} \in [\mathit{id}]$ s.t. $R(\mathbf{x}_{\alpha^{-1}})\}$. Alternatively, R_α can be defined by: $R_\alpha = \{\mathbf{x}_\alpha : \mathbf{x} \in R \cap [\alpha]\}$.*

Thus, Definition 5.6 associates with each $R \subseteq [n]^d$ a family $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ of relations, each of which is entirely contained in the set $[\mathsf{id}]$. This family is intended to represent R through its $d!$ fragments according to the partition $[n]^d = \bigcup_{\alpha\in\mathcal{S}(d)}[\alpha]$. Namely, each R_α encodes the fragment $R\cap[\alpha]$ over $[\mathsf{id}]$.

Actually, $\bigcup_{\alpha\in\mathcal{S}(d)}[\alpha]$ is not really a partition, since the $[\alpha]$'s can overlap. Hence, Definition 5.6 induces some connexions between the relations R_α: if some $\mathbf{x}$ is both in $[\alpha]$ and in $[\beta]$, or equivalently, if $\mathbf{x}_\alpha = \mathbf{x}_\beta$, then $\mathbf{x}\in R\cap[\alpha]$ iff $\mathbf{x}\in R\cap[\beta]$ and hence, by Definition 5.6: $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$. We will keep in mind :

$$\forall\alpha,\beta\in\mathcal{S}(d), \forall\mathbf{x}\in[n]^d : \mathbf{x}_\alpha = \mathbf{x}_\beta \Rightarrow R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta). \tag{19}$$

The following lemma states that condition (19) ensures that the R_α's issue from a single relation R, according to Definition 5.6. Besides, a new formulation of the condition is given in Item 3 of the lemma, that will better fit our syntactical restrictions.

Lemma 5.7 *Let $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ be a family of d-ary relations on $[n]$ such that $R_\alpha \subseteq [\mathsf{id}]$ for each α. The following are equivalent:*

1. *$\exists R\subseteq[n]^d$ such that $R_\alpha = \{\mathbf{x}\in[\mathsf{id}]$ s.t. $R(\mathbf{x}_{\alpha^{-1}})\}$ for each $\alpha\in\mathcal{S}(d)$;*
2. *$\forall\alpha,\beta\in\mathcal{S}(d)$, $\forall\mathbf{x}\in[n]^d$: $\mathbf{x}_\alpha = \mathbf{x}_\beta \Rightarrow R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$;*
3. *$\forall\alpha\in\mathcal{S}(d)$, $\forall\tau\in\mathcal{T}(d)$, $\forall\mathbf{x}\in[n]^d$: $\mathbf{x} = \mathbf{x}_\tau \Rightarrow R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$.*

   (Recall $\mathcal{T}(d)$ denotes the set of transpositions over $\{1,\ldots,d\}$.)

Proof. $1\Rightarrow 2$: See Equation (19).

$2\Rightarrow 1$: For $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ fulfilling 2, consider the relation $R\subseteq[n]^d$ defined by:

$$R(\mathbf{x}) \text{ iff } R_\alpha(\mathbf{x}_\alpha) \text{ for some } \alpha \text{ such that } \mathbf{x}_\alpha\!\uparrow . \tag{20}$$

This definition is well formed, since for any $\alpha,\beta\in\mathcal{S}(d)$ and any $\mathbf{x}\in[n]^d$ such that both $\mathbf{x}_\alpha\!\uparrow$ and $\mathbf{x}_\beta\!\uparrow$ hold, we have $\mathbf{x}_\alpha = \mathbf{x}_\beta$ and thus, by 2, $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$. Now, let $\alpha\in\mathcal{S}(d)$. For any $\mathbf{x}\in[\mathsf{id}]$ we have $(\mathbf{x}_{\alpha^{-1}})_\alpha\!\uparrow$ (since $(\mathbf{x}_{\alpha^{-1}})_\alpha = \mathbf{x}$) and hence, by (20), $R_\alpha(\mathbf{x}) = R_\alpha((\mathbf{x}_{\alpha^{-1}})_\alpha) = R(\mathbf{x}_{\alpha^{-1}})$. Besides, $R_\alpha(\mathbf{x}) = 0$ for any $\mathbf{x}\notin[\mathsf{id}]$, since $R_\alpha\subseteq[\mathsf{id}]$. Thus R_α is obtained from R as required in 1.

$2\Rightarrow 3$: Let $\alpha\in\mathcal{S}(d)$, $\tau\in\mathcal{T}(d)$ and $\mathbf{x}\in[n]^d$ such that $\mathbf{x} = \mathbf{x}_\tau$. Set $\mathbf{y} = \mathbf{x}_{\alpha^{-1}}$. Then, $\mathbf{y}_\alpha = \mathbf{x} = \mathbf{x}_\tau = (\mathbf{y}_\alpha)_\tau = \mathbf{y}_{\alpha\tau}$. Therefore we get by 2: $R_\alpha(\mathbf{y}_\alpha) = R_{\alpha\tau}(\mathbf{y}_{\alpha\tau})$, and hence: $R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$.

$3\Rightarrow 2$: Let $\alpha,\beta\in\mathcal{S}(d)$ and $\mathbf{x}\in[n]^d$ such that $\mathbf{x}_\alpha = \mathbf{x}_\beta$. For $\mathbf{y} = \mathbf{x}_\alpha$, the equality $\mathbf{x}_\alpha = \mathbf{x}_\beta$ can be written $\mathbf{y} = \mathbf{y}_{\alpha^{-1}\beta}$. It means that the permutation $\alpha^{-1}\beta$ exchanges integers that index equal components of $\mathbf{y}$. It is easily seen that this property can be required for each transposition occuring in a decomposition of $\alpha^{-1}\beta$ on $\mathcal{T}(d)$. That is, there exist some transpositions $\tau_1,\ldots,\tau_k\in\mathcal{T}(d)$ such that $\alpha^{-1}\beta = \tau_1\ldots\tau_k$ and $\mathbf{y} = \mathbf{y}_{\tau_1} = \mathbf{y}_{\tau_1\tau_2} = \cdots = \mathbf{y}_{\tau_1\ldots\tau_k}$. Then, applying 3 to these successive tuples, we get: $R_\alpha(\mathbf{y}) = R_{\alpha\tau_1}(\mathbf{y}_{\tau_1}) = R_{\alpha\tau_1\tau_2}(\mathbf{y}_{\tau_1\tau_2}) = \cdots = R_{\alpha\tau_1\ldots\tau_k}(\mathbf{y}_{\tau_1\ldots\tau_k})$. Hence $R_\alpha(\mathbf{y}) = R_\beta(\mathbf{y}_{\alpha^{-1}\beta})$, that is $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$, as required. □

Lemma 5.8 *Let R and $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ be defined as in Definition 5.6. Let $\alpha,\beta\in\mathcal{S}(d)$ and $i\in\{1,\ldots,d\}$. For any $\mathbf{x}\in[\alpha]$:*

1. *$R(\mathbf{x}_\beta)$ is equivalent to $R_{\beta^{-1}\alpha}(\mathbf{x}_\alpha)$.*

2. *$R(\mathbf{x}^{(i)})$ is equivalent to:*

$$\Big(x_i = x_{\alpha_d} \wedge R_{\alpha\tau_{\alpha^{-1}(i),d}}((x_\alpha)^{(d)})\Big) \vee \bigvee_{i\leq k<d} \Big(x_i = x_{\alpha_k} < x_{\alpha_{k+1}} \wedge R_{\alpha\tau_{\alpha^{-1}(i),k}}((x_\alpha)^{(k)})\Big).$$

Proof. **1.** We have seen that $R(\mathbf{x})$ holds iff $R_\gamma(\mathbf{x}_\gamma)$ holds for *any* $\gamma \in \mathcal{S}(d)$ such that $\mathbf{x}_\gamma\uparrow$. Thus we get, for a given $\beta \in \mathcal{S}(d)$: $R(\mathbf{x}_\beta)$ iff $R_\gamma((\mathbf{x}_\beta)_\gamma)$ holds for any $\gamma \in \mathcal{S}(d)$ such that $(\mathbf{x}_\beta)_\gamma\uparrow$. That is, since $(\mathbf{x}_\beta)_\gamma = \mathbf{x}_{\beta\gamma}$:

$R(\mathbf{x}_\beta)$ iff $R_\gamma(\mathbf{x}_{\beta\gamma})$ holds for any $\gamma \in \mathcal{S}(d)$ such that $\mathbf{x}_{\beta\gamma}\uparrow$. (21)

In particular, $\mathbf{x}_{\beta\gamma}\uparrow$ iff $\mathbf{x}_{\beta\gamma} = \mathbf{x}_\alpha$, since $\mathbf{x} \in [\alpha]$. Thus, $\gamma = \beta^{-1}\alpha$ is one of the permutations such that $\mathbf{x}_{\beta\gamma}\uparrow$. Thus, replacing γ by $\beta^{-1}\alpha$ in Equation (21), we get the sought result.

2. From $\mathbf{x}^{(i)} = (x_1, \ldots, \mathrm{succ}(x_i), \ldots, x_d)$ we get:

$$(\mathbf{x}^{(i)})_\alpha = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}},\ \mathrm{succ}(x_i),\ x_{\alpha_{j+1}}, \ldots, x_{\alpha_d})$$

where $j = \alpha^{-1}(i)$. Since $x_{\alpha_1} \leq \cdots \leq x_{\alpha_d}$, the above tuple $(\mathbf{x}^{(i)})_\alpha$ is almost increasingly ordered. More precisely, there exists $k \in \{1, \ldots, d\}$ such that:

$$x_{\alpha_1} \leq \cdots \leq x_{\alpha_{j-1}} \leq x_i = x_{\alpha_{j+1}} = \cdots = x_{\alpha_k} \leq x_{\alpha_{k+1}} \leq \cdots \leq x_{\alpha_d},$$

where $j = \alpha^{-1}(i)$. Clearly, the largest such k is characterized by: $(k = d)$ or ($k < d$ and $x_i = x_{\alpha_k} < x_{\alpha_{k+1}}$). Or equivalently, by:

$(x_i = x_{\alpha_d})$ or ($k < d$ and $x_i = x_{\alpha_k} < x_{\alpha_{k+1}}$). (22)

If we denote by $\tau_{j,k}$ the transposition over $\{1, \ldots, d\}$ which permutes j and k, the definition of k yields that the tuple

$$(x^{(i)})_{\alpha\tau_{j,k}} = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}}, \boxed{x_{\alpha_k}}, x_{\alpha_{j+1}}, \ldots, x_{\alpha_{k-1}}, \boxed{\mathrm{succ}(x_i)}, x_{\alpha_{k+1}}, \ldots, x_{\alpha_d})$$

is non decreasing. Hence, $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{j,k}}((x^{(i)})_{\alpha\tau_{j,k}})$. Besides, since $x_{\alpha_k} = x_i$, the tuple $(x^{(i)})_{\alpha\tau_{j,k}}$ above can also be written:

$$(x^{(i)})_{\alpha\tau_{j,k}} = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}}, \boxed{x_i}, x_{\alpha_{j+1}}, \ldots, x_{\alpha_{k-1}}, \boxed{\mathrm{succ}(x_{\alpha_k})}, x_{\alpha_{k+1}}, \ldots, x_{\alpha_d}).$$

That is: $(x^{(i)})_{\alpha\tau_{j,k}} = (x_\alpha)^{(k)}$. Therefore: $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{j,k}}((x_\alpha)^{(k)})$. Reminding that $j = \alpha^{-1}(i)$, we can finally state: there exists a sole $k \in \{i, \ldots, d\}$ defined by (22), and for this k we have: $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{\alpha^{-1}(i),k}}((x_\alpha)^{(k)})$. The conclusion easily proceeds. □

Proposition 5.9 *For $d > 1$, $\mathrm{ESO}(\forall^d, \text{arity } d) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted})$ on COORD^{d-1}.*

Proof. To simplify, assume we want to translate in $\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted})$ some $\mathrm{ESO}(\forall^d, \text{arity } d)$-formula of the very simple shape: $\Phi \equiv \exists R \forall \mathbf{x} \varphi(\mathbf{x}, R)$, where R is a (single) d-ary relation symbol, $\mathbf{x}$ is a d-tuple of first-order variables, and φ is a quantifier-free formula. Since the sets $[\alpha]$, $\alpha \in \mathcal{S}(d)$, cover the domain $[n]$, we obtain an equivalent rewriting of Φ with the following artificial relativization: $\Phi \equiv \exists R \forall \mathbf{x} \bigwedge_{\alpha \in \mathcal{S}(d)} (\mathbf{x} \in [\alpha] \to \varphi)$. Furthermore, all atomic subformulas of φ built on R can be assumed of the form $R(\mathbf{x}_\beta)$ or $R(\mathbf{x}^{(i)})$, thanks to Fact 5.5.

To get rid of these literals, we substitute to R a tuple of relations $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ that encode R on the sets $[\alpha]$. Recall we proved in Lemma 5.7 that this substitution is legal as soon as $R_\alpha \subseteq [\mathrm{id}]$ and $R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$ for all

$\alpha \in \mathcal{S}(d)$, $\tau \in \mathcal{T}(d)$ and every $\mathbf{x} \in [n]^d$ such that $\mathbf{x}_\tau = \mathbf{x}$. Then, Lemma 5.8 gives the translation of R-atomic formulas into formulas expressed in term of the R_α's. All in all, we get the equivalence of the initial formula Φ to the following:

$$\exists (R_\alpha)_{\alpha\in\mathcal{S}(d)} \left\{ \begin{array}{l} \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} (R_\alpha(\mathbf{x}) \to \mathbf{x} \in [\mathsf{id}])\wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \bigwedge_{\tau\in\mathcal{T}(d)} (\mathbf{x}_\tau = \mathbf{x} \to (R_\alpha(\mathbf{x}) \leftrightarrow R_{\alpha\tau}(\mathbf{x})))\wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \left(\mathbf{x} \in [\alpha] \to \varphi_\alpha(\mathbf{x}, (R_\gamma)_{\gamma\in\mathcal{S}(d)})\right) \end{array} \right\} \tag{23}$$

where each φ_α is obtained from φ by the substitutions:

- $R(\mathbf{x}_\beta) \rightsquigarrow R_{\beta^{-1}\alpha}(x_\alpha)$
- $R(\mathbf{x}^{(i)}) \rightsquigarrow \left\{ \begin{array}{l} \left(x_i = x_{\alpha_d} \wedge R_{\alpha\tau_{\alpha^{-1}(i),d}}((x_\alpha)^{(d)})\right) \vee \\ \bigvee_{i\le k<d} \left(x_i = x_{\alpha_k} < x_{\alpha_{k+1}} \wedge R_{\alpha\tau_{\alpha^{-1}(i),k}}((x_\alpha)^{(k)})\right) \end{array} \right\}$

The first two conjuncts of (23) ensure that the family $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ encodes a relation R (see Lemma 5.7) ; the third interprets assertions of the form $R(\mathbf{x}_\beta)$ and $R(\mathbf{x}^{(i)})$ according to the modalities described in Lemma 5.8. Because of permutability of the conjunction and the universal quantifier, this third conjunct can be rewritten:

$$\bigwedge_{\alpha\in\mathcal{S}(d)} \forall \mathbf{x} : \mathbf{x} \in [\alpha] \to \varphi_\alpha(\mathbf{x}, (R_\gamma)_{\gamma\in\mathcal{S}(d)}) \tag{24}$$

For a fixed conjunct in (24), i.e. for a fixed α, all atomic subformulas of φ_α built on the R_γ's have the form $R_\gamma(\mathbf{x}_\alpha)$ or $R_\gamma((\mathbf{x}_\alpha)^{(i)})$ for some $\gamma \in \mathcal{S}(d)$ and some $i \in \{1, \ldots, d\}$. Hence, the substitution of variables $\mathbf{x}/\mathbf{x}_\alpha$ allows to write such a conjunct as: $\forall \mathbf{x} : \mathbf{x} \in [\mathsf{id}] \to \tilde{\varphi}_\alpha$ where $\tilde{\varphi}_\alpha \equiv \varphi_\alpha(\mathbf{x}/\mathbf{x}_\alpha)$ only involves (R_γ)-subformulas of the form $R_\gamma(\mathbf{x})$ or $R_\gamma(\mathbf{x}^{(k)})$ for some $\gamma \in \mathcal{S}(d)$ and $k \in \{1, \ldots, d\}$. Finally, the initial formula Φ is proved equivalent to:

$$\exists (R_\alpha)_{\alpha\in\mathcal{S}(d)} \left\{ \begin{array}{l} \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} (R_\alpha(\mathbf{x}) \to \mathbf{x} \in [\mathsf{id}])\wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \bigwedge_{\tau\in\mathcal{T}(d)} (\mathbf{x}_\tau = \mathbf{x} \to (R_\alpha(\mathbf{x}) \leftrightarrow R_{\alpha\tau}(\mathbf{x})))\wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \left(\mathbf{x} \in [\mathsf{id}] \to \tilde{\varphi}_\alpha(\mathbf{x}, (R_\gamma)_{\gamma\in\mathcal{S}(d)})\right) \end{array} \right\}$$

that fulfills the requirement of Proposition 5.9. □

5.1 Getting rid of arithmetic

Does mean introducing new second order variables of arity 2, we can assume that our formula $\Phi \in \mathrm{ESO}^\sigma(\forall^d, \text{arity } d)$ – where each relation symbol of arity $d \le 2$ occurs in normalised (i.e. sorted) form – involves no comparison (equality, inequality) relation. We obtain that in two successive steps.

First, if Φ involves inequalities $<$ and $>$, then it is equivalent to the following formula Φ' without inequalities (but with equalities) and two new binary relation symbols $\prec$ and $\succ$:

$$\Phi' \equiv \exists \prec\ \exists \succ : \tilde{\Phi} \wedge \forall x_1, x_2 \left\{ \begin{array}{rcll} (x_1 = x_2 \vee x_1 \prec x_2) & \to & (\neg\max(x_2) \to x_1 \prec \mathrm{succ}(x_2)) & \wedge \\ (x_1 = x_2 \vee x_1 \succ x_2) & \to & (\neg\max(x_1) \to \mathrm{succ}(x_1) \succ x_2) & \wedge \\ x_1 \prec x_2 & \to & (\neg(x_1 \succ x_2) \wedge \neg(x_1 = x_2)) & \wedge \\ x_1 \succ x_2 & \to & \neg(x_1 = x_2) & \end{array} \right\}$$

where $\tilde{\Phi}$ is obtained from Φ by the substitutions $u < v \rightsquigarrow u \prec v$ (resp. $u > v \rightsquigarrow u \succ v$). This is justified as follows: the first two conjuncts of the subformula $\forall x_1 \forall x_2 \{\cdots\}$ express that $x_1 < x_2 \Rightarrow x_1 \prec x_2$ (resp. $x_1 > x_2 \Rightarrow x_1 \succ x_2$). The third and fourth conjuncts express that the three relations $\prec, \succ$ and $=$ are totally disjoint. That implies that $\prec$ and $\succ$ have their exact meaning. Notice also that the occurrences of $\prec$ and $\succ$ preserve the sorted property.

Secondly, if Φ involves equalities (without any inequality), it is equivalent to the following formula Φ', written without the symbol "=" but with the new binary symbol $\approx$:

$$\Phi' \equiv \exists \approx :\ \tilde{\Phi} \ \wedge\ \forall x_1, x_2 \Psi$$

where $\tilde{\Phi}$ is obtained from Φ by replacing each equality $u = v$ by $u \approx v$, and Ψ is the conjunction of the following formulas:

- $\min(x_1) \to (\min(x_2) \leftrightarrow x_1 \approx x_2)$;
- $\min(x_2) \to (\min(x_1) \leftrightarrow x_1 \approx x_2)$;
- $(\neg\max(x_1) \wedge \neg\max(x_2)) \to (x_1 \approx x_2 \leftrightarrow \mathrm{succ}(x_1) \approx \mathrm{succ}(x_2))$.

Notice that this transformation preserves sorted property. The results obtained until now can be recapitulated as follows:

> On $(d-1)$-pictures, each $\mathrm{ESO}^{\sigma}(\forall^d, \text{arity } d)$-formula can be written under the form:
>
> $$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(x),\ \max(x), \\ Q(\mathbf{x}_\iota),\ R(\mathbf{x}),\ R(\mathbf{x}^{(i)}) \end{array} \right\} \qquad (25)$$
>
> Here, $\mathbf{R}$ (resp. $\mathbf{Q}$) is a list of relation symbols of arity d (resp. $d-1$), $x \in \mathbf{x}$, $Q \in (Q_s)_{s \in \Sigma}$, $R \in \mathbf{R}$, $\iota \in \mathcal{I}(d)$, $i \in [d]$.

6 Localization of input relations

Let's come back to the normalization of the formula Φ of Equation (28). For simplicity, assume that it has the restricted form $\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \varphi(\mathbf{x}, \mathbf{R}, Q, \sigma)$, with only one relation symbol Q of arity $d-1$.

We aim at defining a tuple of relations $(Q_\alpha)_{\alpha \in \mathcal{I}(d)}$, in such a way that $Q_\alpha(\mathbf{x}) = Q(\mathbf{x}_\alpha)$ for each $\mathbf{x}$. Clearly, such relations will allow to write Φ under the desired form, by replacing each subformula $Q(\mathbf{x}_\alpha)$ by the sorted translation: $Q_\alpha(\mathbf{x})$. The difficulty is to express this definition with our syntactical restrictions, that is, without involving any $\mathbf{x}_\alpha$ with $\alpha \neq \mathsf{id}$.

Notice that the strategy used in Section 5 to "sort" atomic subformulas $R(\mathbf{x}_\alpha)$ build on any *existentially quantified* d-ary relation R is no more avalaible, since it means suppressing R in favour of some new existentially quantified relations. Of course, we can't operate like this with the *input* relation Q. To give an hint of the method used in this section, let us consider an easy example.

An easy case Consider the case where $d = 2$. We deal with two first-order variables x and y and we only accept atoms of the form $Q(x)$, $R(x,y)$, $R(\mathrm{succ}(x), y)$ and $R(x, \mathrm{succ}(y))$ for any input unary relation Q and any guessed binary relation R. How can we tackle occurences of some atom $Q(y)$ in the formula ? A natural idea is to define a new binary relation Q_2 in such a way that $Q_2(x,y) = Q(y)$ holds for any x,y. (We denote it Q_2 to refer both to Q and to the projection of (x,y) on its *second* component.) Hence, we set:

$$Q_2 = \{(x,y) : Q(y)\}.$$

Thus, any atom $Q(y)$ could be replaced by the sorted atom $Q_2(x,y)$. But the logical definition of Q_2 with our syntactical constraints compels to introduce an additional binary relation T that will be used as a buffer to *transport* the information $Q(y)$ into the expression $Q_2(x,y)$. We set

$$T = \{(x,y) : Q(x+y)\}.$$

Clearly, T is inductively defined from Q by the assertions $T(x,0) = Q(x)$ and $T(x+1,y) = T(x,y+1)$. Besides, Q_2 is defined from T by $Q_2(0,y) = T(0,y)$ and $Q_2(x,y) = Q_2(x+1,y)$. All these assertions can be rephrazed in our logical framework, with the following formulas:

$$\forall x,y \left\{ \begin{array}{rcl} \min(y) & \to & (T(x,y) \leftrightarrow Q(x)) \ \wedge \\ (\neg\max(x) \wedge \neg\max(y)) & \to & T(\mathrm{succ}(x),y) \leftrightarrow T(x,\mathrm{succ}(y)) \end{array} \right\}$$

$$\forall x,y \left\{ \begin{array}{l} \min(x) \to (Q_2(x,y) \leftrightarrow T(x,y)) \ \wedge \\ Q_2(x,y) \leftrightarrow Q_2(\mathrm{succ}(x),y) \end{array} \right\}$$

Now, it remains to insert this defining formulas in the initial formula Φ to be normalized, and to replace each occurrence of $Q(y)$ by $Q_2(x,y)$. Of course, such a construction has to be carried on for each input unary relation Q.

The general case Given $i, j \in \{1,\ldots,d\}$ and $\alpha \in \mathcal{S}(d)$, we denote by (ij) the transposition that exchanges i and j, and by $\alpha(ij)$ the composition of α and (ij). It is well-known that each permutation α can be written as a product of transpositions, $\alpha = (u_1v_1)(u_2v_2)\ldots(u_pv_p)$. It is easily seen that this product can be chosen in such a way that $v_i = u_{i+1}$ for any i. This is because if some sequence $(ab)(cd)$ with $b \neq c$ occurs, it can be replaced by $(ca)(ab)(bc)(cd)$, and a well chosen iteration of such rewritings yields the desired decomposition. This can be further refined, by fixing at d one element of the first transposition involved in the decomposition and by prohibiting useless sequence as $(ab)(ba)$. Finally, each $\alpha \in \mathcal{S}(d)$ can be written

$$\alpha = (du_1)(u_1u_2)\ldots(u_{k-2}u_{k-1})(u_{k-1}u_k), \tag{26}$$

where u_i, u_{i+1} and u_{i+2} are pairwise distinct elements of $\{1,\ldots,d\}$ for any i. We call *alternated factorization of α* such a decomposition.

A permutation α admits of several alternated factorizations, and we want to single out one of them for each α, in order to allow an inductive reasoning build on this particular decomposition. There is no canonical way to perform this task. In the following lemma, we roughly describe one possible choice, that implicitely refers to the graph $\mathcal{G}_d$ on domain $\mathcal{S}(d)$ whose edges correspond to those pairs of permutations (α,β) such that $\beta = \alpha(ij)$ for some $i, j \in \{1,\ldots,d\}$.

Lemma 6.1 *There exists an oriented tree $\mathcal{T}_d$ covering $\mathcal{S}(d)$ that is rooted at* id *and such that each $\mathcal{T}_d$-path starting at* id*, say* id$\alpha_1\ldots\alpha_p$*, corresponds to an alternated factorization of α_p.*

Proof. Trees $\mathcal{T}_d$ for $d \geq 2$ are defined inductively. For $d = 2$, there is a unique such tree: $(12) \to (21) = (12)(21)$. So, assume we are given $\mathcal{T}_{d-1}$ and carry on the construction of $\mathcal{T}_d$ as follows:

(a) First, view each permutation $\alpha = \alpha_1 \dots \alpha_{d-1} \in \mathcal{S}(d-1)$ as a permutation of $\{2, \dots, d\}$ by renaming α_i as $\alpha_i + 1$. That is, replace $\alpha = \alpha_1 \dots \alpha_{d-1}$ by $\alpha^+ = (\alpha_1 + 1) \dots (\alpha_{d-1} + 1)$.

(b) Then, replace each such α^+ by $\beta = \beta_1 \dots \beta_d \in \mathcal{S}(d)$ defined by: $\beta_1 = 1$ and $\beta_i = [\alpha^+]_{i-1} = \alpha_{i-1} + 1$ for $i > 1$. Thus, $\mathcal{T}_{d-1}$ now covers the set of permutations $\beta \in \mathcal{S}(d)$ that fulfill $[\beta]_1 = 1$.

(c) For each node β thus obtained, create a new node labelled by the composition of β with the transposition $(1d)$ – that is by the permutation $\beta(1d)$ – and create an edge $\beta \to \beta(1d)$.

(d) Finally, link each such node $\beta(1d)$ to $d-2$ new nodes $\beta(1d)(di)$, for $i = 2, \dots, d-1$.

In Fig 2, we display the steps of the construction of $\mathcal{T}_4$ from $\mathcal{T}_2$. Letters (a), ..., (d) in the figure refer to the above items. The correction of the method on this example is clear. We leave it to the reader to verify that it generalises to any d. □

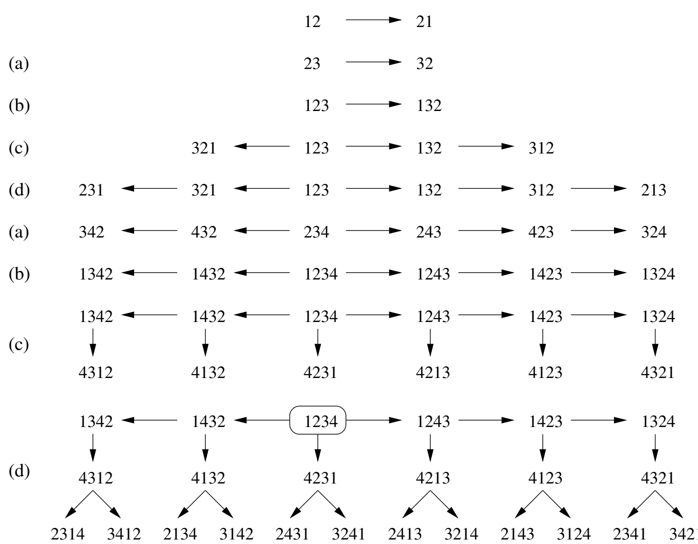

Figure 2: Construction of $\mathcal{T}_4$ from $\mathcal{T}_2$. The result is an oriented tree rooted at id, spanning $\mathcal{S}(4)$, whose all pathes from id are alternated.

This lemma allows us to choose, for each $\alpha \in \mathcal{S}(d)$, *one* alternated factorization of α: it is the decomposition $(di_1)(i_1i_2)\dots(i_{k-1}i_k)$ corresponding to the unique path from id to α in $\mathcal{T}_d$. We denote by $\mathsf{id}.di_1.i_1i_2.\dots.i_{k-1}i_k$ this particular factorization. And when this path until α can be continued *in* $\mathcal{T}_d$ to some permutation $\alpha(i_ki_{k+1})$, we denote by $\alpha.i_ki_{k+1}$ this last permutation. For instance, in the example displayed in Fig 2, we can write $2143 = 4123.13$ and $3124 = 4123.14$ while $4321 = 4123(24)$ *cannot be written* 4123.24. Notice furthermore that the integers i_k and i_{k+1} are ordered in the notation $\alpha.i_ki_{k+1}$ (unlike in the notation $\alpha(i_ki_{k+1})$): we place in first position the integer i_k involved in the last transposition leading to α (with $i_k = d$ if $\alpha = \mathsf{id}$). All in all, the reader is invited to keep in mind the numerous presuppositions attached to the notation $\alpha.uv$: the statement $\beta = \alpha.uv$ means:

- $\beta = \alpha(uv)$;
- (α,β) is an edge of $\mathcal{T}_d$;
- either $\alpha = \mathsf{id}$ and $u = d$, or $\alpha = \gamma.tu$ for some $\gamma \in \mathcal{S}(d)$ and some $t \neq u$ in $\{1,\ldots,d\}$.

Let us now come to a straightforward remark connecting d-tuples to alternated factorizations. First recall that for $\mathbf{x} = (x_1,\ldots,x_d)$, $i \in \{1,\ldots,d\}$ and $\alpha,\beta \in \mathcal{S}(d)$, we denoted by $[\mathbf{x}]_i$ the i^{th} component of $\mathbf{x}$, we defined $\mathbf{x}_\alpha$ as the d-tuple $(x_{\alpha_1},\ldots,x_{\alpha_d})$, and we noticed that $(\mathbf{x}_\alpha)_\beta = \mathbf{x}_{\alpha\beta}$. (See Definition 5.2.)

Fact 6.2 *For any $\mathbf{x} \in [n]^d$, $\alpha \in \mathcal{S}(d)$, and $i,j \in \{1,\ldots,d\}$:*

(a) $\mathbf{x}_{\alpha.ij} = (\mathbf{x}_\alpha)_{(ij)}$.

(b) $[\mathbf{x}_{\alpha.ij}]_j = [\mathbf{x}]_d$.

Proof. (*a*) From $\alpha.ij = \alpha(ij)$ and $\mathbf{x}_{\alpha\beta} = (\mathbf{x}_\alpha)_\beta$. (*b*) By induction on α: if $\alpha = \mathsf{id}$, then necessarily $i = d$ and $[\mathbf{x}_{\mathsf{id}.dj}]_j = [\mathbf{x}]_d$ clearly holds. Otherwise, $[\mathbf{x}_{\alpha.ij}]_j = [(\mathbf{x}_\alpha)_{ij}]_j = [\mathbf{x}_\alpha]_i = [\mathbf{x}]_d$, by induction hypothesis. ◻

Given a d-tuple $\mathbf{x} = (x_1,\ldots,x_d)$ of first-order variables, we denote by $\mathbf{x}^-$ the $(d-1)$-tuple obtained from $\mathbf{x}$ by erasing its last component. That is,

$$(x_1,\ldots,x_{d-1},x_d)^- := (x_1,\ldots,x_{d-1}).$$

In particular, for $\alpha \in \mathcal{S}(d)$, we denote by $\mathbf{x}_\alpha^-$ the $(d-1)$-tuple $(x_{\alpha_1},\ldots,x_{\alpha_{d-1}})$. Each $(d-1)$-tuple build upon the d variables $x_1,\ldots,x_d$ can clearly be written $\mathbf{x}_\alpha^-$ for some $\alpha \in \mathcal{S}(d)$. Therefore, the occurence of a non-sorted atom in Φ has the form $Q(\mathbf{x}_\alpha^-)$ for some $Q \in (Q_a)_{a\in\Sigma}$ and some $\alpha \in \mathcal{S}(d)$, and the purpose of this section amounts to rewrite each such occurence $Q(\mathbf{x}_\alpha^-)$ as $Q'(\mathbf{x})$ for some well chosen relation Q'.

Definition 6.3 *Given a $(d-1)$-ary relation Q and two family of d-ary relations, $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$ and $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$, we say that $(T_\alpha,Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ is a d-simulation of Q if the following axioms hold, for any $\alpha \in \mathcal{S}(d)$ and any $i,j \leq d$ such that $\alpha.ij$ is defined, and for any d-tuple $\mathbf{x}$ of variables:*

(A_1) $T_{\mathsf{id}}(\mathbf{x}) = Q(\mathbf{x}^-)$ *if* $[\mathbf{x}]_d = 0$.

(A_2) $T_{\alpha.ij}(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x}_{(ij)})$.

(A_3) $T_\alpha(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x})$ *if* $[\mathbf{x}]_i = 0$.

(A_4) $Q_{\mathsf{id}}(\mathbf{x}) = Q(\mathbf{x}^-)$.

(A_5) $Q_{\alpha.ij}(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x})$ *if* $[\mathbf{x}]_j = 0$.

(A_6) $Q_{\alpha.ij}(\mathbf{x})$ *doesn't depend on* $[\mathbf{x}]_j$.

Lemma 6.4 *Let $(T_\alpha,Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ be a d-simulation of some $(d-1)$-ary predicate Q. For any $\mathbf{x} \in [n]^d$ and $\alpha \in \mathcal{S}(d)$: $Q_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-)$.*

Proof. Let us first prove that for any $\mathbf{x} \in [n]^d$ and $\alpha \in \mathcal{S}(d)$:

$$[\mathbf{x}]_d = 0 \Rightarrow T_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-). \tag{27}$$

We procced by induction on α. If $\alpha = \mathsf{id}$, (27) follows from (A_1). Given a non-identique permutation $\alpha.ij$, we have:

$$\begin{aligned} T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) &= T_{\alpha.ij}((\mathbf{x}_\alpha)_{(ij)}) \text{ by Fact 6.2-(a).} \\ &= T_{\alpha.ij}(\mathbf{x}_\alpha) \text{ by } (\mathsf{A}_2). \end{aligned}$$

But $[\mathbf{x}_\alpha]_i = [(\mathbf{x}_\alpha)_{(ij)}]_j = [\mathbf{x}_{\alpha.ij}]_j$ and hence, by Fact 6.2-(b): $[\mathbf{x}_\alpha]_i = [\mathbf{x}]_d = 0$. Therefore we can resume the above sequence of equalities with:

$$\begin{aligned} T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) &= T_\alpha(\mathbf{x}_\alpha) \text{ by } (\mathsf{A}_3) \text{ since } [\mathbf{x}_\alpha]_i = 0. \\ &= Q(\mathbf{x}^-) \text{ by induction hypothesis.} \end{aligned}$$

This completes the proof of (27)

Let us now prove the equality $Q_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-)$. If $\alpha = \mathsf{id}$, the result comes from (A_4). For a non-identique permutation $\alpha.ij$, we have to prove $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q(\mathbf{x}^-)$ for any tuple $\mathbf{x} \in [n]^d$. First notice that we can restrict, without loss of generality, to the case where $[\mathbf{x}]_d = 0$. Indeed, we can otherwise consider the tuple $\mathbf{y}$ obtained from $\mathbf{x}$ by setting $[\mathbf{x}]_d$ to 0. (That is, $\mathbf{y}$ only differs from $\mathbf{x}$ by its d^{th} component, which is null.) Clearly, $\mathbf{y}^- = \mathbf{x}^-$. Besides, the j^{th} component of $\mathbf{x}_{\alpha.ij}$ is $[\mathbf{x}]_d$, from Fact 6.2-(b). Similarly, the j^{th} component of $\mathbf{y}_{\alpha.ij}$ is $[\mathbf{y}]_d$. Hence, the tuples $\mathbf{x}_{\alpha.ij}$ and $\mathbf{y}_{\alpha.ij}$ coincide on each component of rank distinct from j. Therefore $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q_{\alpha.ij}(\mathbf{y}_{\alpha.ij})$ by (A_6) and we get: $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q(\mathbf{x}^-)$ iff $Q_{\alpha.ij}(\mathbf{y}_{\alpha.ij}) = Q(\mathbf{y}^-)$. Thus, we can assume $[\mathbf{x}]_d = 0$. It follows $[\mathbf{x}_{\alpha.ij}]_j = [\mathbf{x}]_d = 0$, by Fact 6.2-(b), and hence:

$$\begin{aligned} Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) &= T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) \text{ by } (\mathsf{A}_5)\text{, since } [\mathbf{x}_{\alpha.ij}]_j = 0. \\ &= Q(\mathbf{x}^-) \text{ by (27), since } [\mathbf{x}]_d = 0. \end{aligned}$$

The proof is complete. □

Lemma 6.5 *Let Q be a $(d-1)$-ary relation and $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$, $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ be two tuple of d-ary relations satisfying, for each d-tuple $\mathbf{x} = (x_1, \dots, x_d)$:*

(F_1) $min(x_d) \to (T_{\mathsf{id}}(\mathbf{x}) \leftrightarrow Q(\mathbf{x}^-))$.

(F_2) $\big(\neg max(x_i) \wedge \neg max(x_j)\big) \to \big(T_{\alpha.ij}(\mathbf{x}^{(i)}) \leftrightarrow T_{\alpha.ij}(\mathbf{x}^{(j)})\big)$.

(F_3) $min(x_i) \to \big(T_\alpha(\mathbf{x}) \leftrightarrow T_{\alpha.ij}(\mathbf{x})\big)$.

(F_4) $Q_{\mathsf{id}}(\mathbf{x}) \leftrightarrow Q(\mathbf{x}^-)$.

(F_5) $min(x_j) \to \big(Q_{\alpha.ij}(\mathbf{x}) \leftrightarrow T_{\alpha.ij}(\mathbf{x})\big)$.

(F_6) $Q_{\alpha.ij}(\mathbf{x}) \leftrightarrow Q_{\alpha.ij}(\mathbf{x}^{(j)})$.

Then $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ is a d-simulation of Q. Furthermore, each Q admits such a d-simulation fulfilling (F_1)...(F_6)

Proof. Clearly, the formula (F_i) is a mere transcription of the axiom (A_i) for each $i \neq 2$. We have to prove that (F_2) implies (A_2). Formula (F_2) yields that $T_{\alpha.ij}(\mathbf{x})$ has the same value on tuples of the form

$$\mathbf{x} = (\mathbf{u}, x + c, \mathbf{v}, y - c, \mathbf{w}),$$

for any $c \leq \min\{n-1-x, y\}$, where $x+c$ (resp. $y-c$) is the component of rank i (resp. j) of $\mathbf{x}$. (That is: $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$ and $\mathbf{w} \in [n]^{d-j}$.) In other words, the value of $T_{\alpha.ij}(\mathbf{x})$ depends on $[\mathbf{x}]_i + [\mathbf{x}]_j$ rather than on the precise values of these two components. As a consequence, for any $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$, $\mathbf{w} \in [n]^{d-j}$ and any $x, y \in [n]$:

$$T_{\alpha.ij}(\mathbf{u}, x, \mathbf{v}, y, \mathbf{w}) = T_{\alpha.ij}(\mathbf{u}, y, \mathbf{v}, x, \mathbf{w}).$$

This is Axiom (A_2).

It remains to prove that such relations $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$ and $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ exist for every Q. To see this, assume we are given a $(d-1)$-ary relation Q and define the Q_α's and the T_α's as follows:

- $Q_\alpha(\mathbf{x}) = Q(\mathbf{x}^-_{\alpha^{-1}})$ for any $\mathbf{x} \in [n]^d$;
- $T_{\mathsf{id}}(\mathbf{x}) = Q_{id}(\mathbf{x})$ for any $\mathbf{x} \in [n]^d$;
- $T_{\alpha.ij}(\mathbf{u}, x, \mathbf{v}, y, \mathbf{w}) = Q_{\alpha.ij}(\mathbf{u}, x+y, \mathbf{v}, 0, \mathbf{w})$ for any $x, y \in [n]$, $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$ and $\mathbf{w} \in [n]^{d-j}$.

We leave it to the reader to check that the sequence $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ satisfies the formulas (F_1), ..., (F_6) (and hence, is a d-simulation of Q). □

Proposition 6.6 *For any $d > 0$, $\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted}) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$ on $\textsc{coord}^{d-1}$.*

Theorem 4.1, Proposition 5.9 and Proposition 6.6 can now be collected in the following result:

Theorem 6.7 $\mathrm{ESO}(\text{var } d) = \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$ *on* $\textsc{coord}^{d-1}$ *for any* $d > 1$.

Remark 6.8 *All in all, our normalization process can be summarized as follows:*

> *On $(d-1)$-pictures, each $\mathrm{ESO}(\text{var } d)$-formula can be written under the form:*
>
> $$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} min(x_i),\ max(x_i), \\ Q(x_1, \dots, x_{d-1}),\ R(\mathbf{x}),\ R(\mathbf{x}^{(i)}) \end{array} \right\} \quad (28)$$
>
> *Here, $\mathbf{R}$ (resp. $\mathbf{Q}$) is a list of relation symbols of arity d (resp. $d-1$), $\mathbf{x} = (x_1, \dots, x_d)$, $i \in [d]$, $Q \in (Q_s)_{s\in\Sigma}$, $R \in \mathbf{R}$. (Recall that for $\mathbf{x} = (x_1, \dots, x_d)$, $\mathbf{x}^{(i)}$ is meant to denote the tuple $(x_1, \dots, x_{i-1}, succ(x_i), x_{i+1}, \dots, x_d)$.) Moreover, we can assume that the successor function only applies to arguments that are not maximal, or alternatively, that $succ(n) = n$, that means that the interpretation of the successor function symbol is the* noncyclic *successor* instead of *the cyclic one.*

Theorem 3.9 straightforwardly proceeds from Theorem 6.7 completed by Remark 6.8, Proposition 3.10, and from the proposition below:

Proposition 6.9 *For any $d > 0$ and any d-language L,*

$$\mathit{coord}^d(L) \in \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1, \text{sorted}) \Rightarrow L \in \mathrm{NLIN}^d_{ca}.$$

Proof. Assume $\mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d+1, \mathrm{sorted})$. Without loss of generality and according to Remark 6.8, we can assume that $\mathsf{coord}^d(L)$ is defined by an $\mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d+1, \mathrm{sorted})$ sentence of the form $\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \forall t\, \psi(\mathbf{x}, t)$ where $\mathbf{x} = (x_1, \ldots, x_d)$ and

$$\psi(\mathbf{x}, t) \equiv \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(x_i),\ \max(x_i),\ \min(t),\ \max(t), \\ Q(\mathbf{x}),\ R(\mathbf{x}, t),\ R(\mathbf{x}^{(i)}, t),\ R(\mathbf{x}, \mathrm{succ}(t)) \end{array} \right\}.$$

for $i \in [d]$, $Q \in (Q_s)_{s \in \Sigma}$ and $R \in \mathbf{R}$. Moreover, the succ symbol is interpreted as the *noncyclic* successor function.

The key point is that sentence Φ can be checked in $O(n)$ steps (for an input picture of domain $[n]^d$) by a *local and parallel nondeterministic* process. More precisely, it is easy to construct a d-automaton which uses the following informal but intuitive algorithm to check whether $\mathsf{coord}^d(p) \models \Phi$, for any picture $p : [n]^d \to \Sigma$:

> *For $t = 1, 2, \ldots, n$, check in parallel whether the truth value of atoms $R(a, t)$ (for all $a \in [n]^d$) are compatible:*
>
> - *with each other;*
> - *with the "previous" values of atoms $R(a, t-1)$ (when $t > 1$);*
> - *with the values of the input atoms $Q(a)$.*
>
> *If the answer is "yes" then accept, otherwise reject.*

The process is correct because at each moment the cellular automaton at cell $a \in [n]^d$ has only to consider the fixed number of information bits that the point $a = (a_1, \ldots, a_d)$ and its d *neighbor* points $a^{(i)} = (a_1, \ldots, a_{i-1}, \mathrm{succ}(a_i), a_{i+1}, \ldots, a_d)$ hold. Each of the n iterations of the loop (for t from 1 to n) is performed in constant time, hence the total time is $O(n)$. □

Conclusion

Finally, notice that Theorem 3.9 characterizing the *linear time* complexity class of nondeterministic *cellular automata* is very similar to the following result by the authors [14] about time complexity $O(n^d)$ of nondeterministic RAM's (for any $d \geq 1$):

$$\mathrm{NTIME}_{\mathrm{ram}}(n^d) = \mathrm{ESOF}(\mathrm{var}\, d) = \mathrm{ESOF}(\forall^d, \mathrm{arity}\, d).$$

The main difference is that this last result involves the existential second-order logic with *functions* (ESOF) instead of or in addition to *relations*, and holds in all kinds of structures without restriction: pictures, structures of any arity, etc. It is interesting and maybe surprising to notice that, in those results, the *time degree d* of a RAM computation plays the same role as the *dimension $d+1$* of the time-space diagram of a linear time bounded computation for a d-dimensional cellular automaton.

Both results confirm the robustness of the time complexity classes $\mathrm{NTIME}_{\mathrm{ram}}(n^d)$ and $\mathrm{NLIN}^d_{\mathrm{ca}}$. They stress the significance of the RAM's as a sequential model, and of the cellular automata as a parallel model.

References

[1] J. R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.

[2] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*. Cambridge University Press, to appear in 2012.

[3] M. de Rougemont. Second-order and inductive definability on finite structures. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 33:47–63, 1987.

[4] M. Delorme and J. Mazoyer. *Cellular automata: A parallel model*. Springer, 373 pages, Mathematics and Its Applications, 1998.

[5] A. Durand and E. Grandjean. First-order queries on structures of bounded degree are computable with constant delay. *ACM Trans. Comput. Log.*, 8(4), 2007.

[6] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[7] C.C. Elgot. Decision problem of finite automata design and related arithmetics. *Trans. Amer. Math. Society*, 98:21–51, 1961.

[8] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings*, pages 43–73, 1974.

[9] J. Flum, E. Grädel, and T. Wilke, editors. *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, volume 2 of *Texts in Logic and Games*. Amsterdam University Press, 2008.

[10] H. Gaifman. On local and nonlocal properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North Holland, 1982.

[11] D. Giammarresi and A. Restivo. *Two-dimensional languages, in: Handbook of Theoretical Computer Science*, volume 3- Beyond words, chapter 4, pages 215–267. Springer-Verlag New York, 1997.

[12] D. Giammarresi, A. Restivo, S. Seibert, and W. Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Information and Computation*, 125(1):32 – 45, 1996.

[13] E. Grädel, Ph. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S.Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. Springer, 2007.

[14] E. Grandjean and F. Olive. Graph properties checkable in linear time in the number of vertices. *Journal of Computer and System Sciences*, 68:546–597, 2004.

[15] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*, pages 132–145. North Holland, 1965.

[16] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1999.

[17] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. Internal Report IT-94-264, Laboratoire d'Informatique Fondamentale de Lille, Université de Lille, France, 1994.

[18] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[19] O. Matz and N. Schweikardt. Expressive power of monadic logics on words, trees, pictures, and graphs. In Flum et al. [9], pages 531–552.

[20] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to felgota decision problem of Second-Order logic. *Mathematical Systems Theory*, 2(1):57–81, March 1968.

[21] B.A. Trakhtenbrot. Finite automata and the logic of one-place predicates. *Siberian Math. J.*, 3:103–131, 1962. English translation in: AMS Transl. 59 (1966) 23–55.

[22] G. Turán. On the definability of properties of finite graphs. *Discrete Mathematics*, 49:291–302, 1984.