The expressive power of bijections over weakly arithmetized structures

Étienne Ailloud
Département d'informatique
Université de Caen
14000 Caen, France
eailloud@etu.info.unicaen.fr

Arnaud Durand [*]
LACL, Département d'informatique
Université Paris 12
94010 Créteil, France
durand@univ-paris12.fr

September 30, 2004

Abstract

We investigate the expressive power of existential second-order formulas whose second-order quantifiers range over bijective unary functions. We show that, as long as interpretations are taken over structures with built-in linear order relation and addition function, quantifying over bijections is as expressive as quantifying over arbitrary unary functions. The originality of our result is that it remains true even if the first-order part of formula contains exactly one variable (which is universally quantified). Our result immediately provides a new characterization of non-deterministic linear time on RAMs. It also permits to derive a corollary on the Skolem normal form of first-order formulas over weakly arithmetized structures.

Keywords: Finite model theory, descriptive complexity, non-deterministic linear time.

Introduction

Descriptive complexity considers computational complexity through a logical prism. Characterizations of complexity classes have been obtained that abstract notions of time, space or hardware and provide a unified and machine-independent view of computational power. Such an approach may be useful (also) for the design of computational complexity classes: the existence of one or several clear and simple logical characterizations for a class may argue in favour of the naturalness and robustness of its definition. Particularly interesting in that context is the case of non-deterministic linear time. In a series of papers, Grandjean extensively studied that notion and proposed the class NLIN as a candidate to formalize it. Informally, a property (originally on unary functions) is decidable in NLIN if it can be decided by a non-deterministic random access machine that runs in time $O(n)$ (where n is the number of elements in the domain) while manipulating only numbers of values $O(n)$. That class appears to be very robust as it is not affected by the choice of basic operations (successor, addition, multiplication...). Its

[*] Corresponding Author

definition has been extended recently (see [GO04]) to deal with properties on any kind of input signatures (such as graphs). Let $\mathcal{S}$ be a given signature. It has been shown that NLIN coincides with the class of properties definable by formulas of the form:

$$\exists f_1 \ldots \exists f_k \; \forall x \; \phi(\mathcal{S}, 0, \overline{f}) \tag{1}$$

where each f_i is a unary function symbol and ϕ is a quantifier-free formula. This result has been refined in a lot of manners (see [GO98, Dur02]) thus providing several other logical charaterizations of non-deterministic linear time. In particular, in [Dur02], it is shown that, provided a linear order is given, NLIN can be defined by formulas of the form:

$$\exists b_1 \ldots \exists b_k \; \forall x \overline{\exists y} \; \phi(\mathcal{S}, \leq, 0, \overline{b}) \tag{2}$$

where the b_i are unary function symbols whose interpretations can be taken among bijective unary functions [1]. However, the first-order part of the formula has $\forall\exists^*$ as quantifier prefix and it seems not evident at all if it can be restricted to one universal quantifier only. The present paper addresses that question. We show that, in the presence of a linear order relation and an addition function, the first-order existential quantifiers can be removed and that NLIN is characterized by the following class of formulas:

$$\exists b_1 \ldots \exists b_k \; \forall x \; \phi(\mathcal{S}, 0, \leq, +, \overline{b}) \tag{3}$$

where the b_i are still bijections but the first-order part contains only one variable (which is universally quantified). Such a result is, a priori, surprising: considering characterization (2), there is no obvious reason that a bijective correspondence can be set between the universally quantified variable and the existentially quantified ones.

A first immediate corollary of our result is a new and concise characterization of linear time whatever the input signature is (provided it contains our “weak” arithmetic predicates). Finding such a simple characterization of NLIN helps understanding this important complexity class. Our second corollary is purely logical and concerns Skolem normal forms of first-order formulas: it holds that, in our context of weakly arithmetized finite structures, Skolem unary functions can be chosen to be bijections.

One of the main motivations of our work was to develop a descriptive approach for the study of the complexity of scheduling and sequencing problems. Basically, a scheduling problem deals with tasks that may have a length, a release time, a deadline, a dedicated processor etc... (i.e. unary functions on a universe of tasks) and looks for a compatible schedule (i.e. an injective unary function). The compatibility of the schedule may then be checked in deterministic linear time. To illustrate these points, let's consider the following NP-complete problem (referenced as problem [SS1] in [GJ79]):

Sequencing with release times and deadlines
Input: A set T of tasks and, for each task $t \in T$, a length $l(t) \in \mathbb{Z}^+$, a release time $r(t) \in \mathbb{Z}^+$ and a deadline $d(t) \in \mathbb{Z}^+$
Question: Is there a one-processor schedule for T that satisfies the release time

[1] It should be noticed that, without built-in linear order, quantifying over unary function is in general strictly more expressive than quantifying over bijections (see [DLS98]).

constraints and meets all the deadlines i.e. a one-to-one function $\sigma : T \mapsto \mathbb{Z}^+$ and a (non-cyclic) successor function $next : T \mapsto T$ such that:

$$\forall t \in T \left\{ \begin{array}{l} \sigma(next(t)) \geq \sigma(t) + l(t) \wedge \\ \sigma(t) \geq r(t) \wedge \sigma(t) + l(t) \leq d(t). \end{array} \right. \tag{4}$$

Input and guessed functions are as expected and the constraints are easily checkable in deterministic linear time.
It seems possible that a lot of natural NP-complete monoprocessor or multiprocessor scheduling problems are in fact NLIN-complete i.e. that, unlike classical NP-complete as SAT (the propositional satisfiability problem), non-deterministic algorithms that solve them really require a linear number of non-deterministic bits. Proving such a result would give precise informations on the complexity of these problems including possible non-trivial deterministic lower bounds. The main result of this paper is a preliminary step in our program: it provides a logical characterization of non-deterministic linear time that can be intrinsically viewed as an "abstract" scheduling problem. In other words, it shows that all NLIN problems can be seen as problems over unary functions (over tasks) for which a bijection (a schedule) must be found that satisfy a given set of first-order constraints expressible with only one variable, the order relation and the addition function (i.e. scheduling constraints that can be checked in deterministic linear time). However, some more work on the syntactic form of the characterization remains to be done to show the NLIN-completeness of natural scheduling problems. In particular, it would be very helpful to have a purely conjunctive first-order part (i.e. to eliminate disjunction in formulas) in the characterization in order to obtain a constraint in the style of Equation 4 (see [Oli98] for conjunctive normalization of existential second-order formulas).

The paper is organized as follows. In section 1, basic definitions are given. Then, in section 2, we establish the main results of our paper and mainly prove that unary functions and bijections have the same expressive power (for each given prefixed class of formulas) on structures with built-in linear order and addition. Then, in section 3.2, two applications of our result are given: the first concerns logical characterization of linear time, the second Skolem normal forms of formulas.

1 Definitions

A *signature* $\mathcal{S} = \{S_1, \ldots, S_l\}$ is a finite set of relation and function symbols S_j each of arity i_j (constants are 0-ary functions). It is supposed that $\mathcal{S}$ contains the equality symbol.
We suppose the reader is familiar with first-order logic. Recall that a formula $\Psi(\mathcal{S})$ is an existential second-order formula over the signature $\mathcal{S}$ if it is of the form:

$$\exists R_1 \ldots \exists R_k \ \psi(S_1, \ldots, S_l, R_1, \ldots, R_k),$$

where $\mathcal{R} = \{R_1, \ldots, R_k\}$ is a signature disjoint from $\mathcal{S}$ and $\psi(\mathcal{S}, \mathcal{R})$ is a first-order formula over the signature $\mathcal{R} \cup \mathcal{S}$.

Let $\mathcal{M} = \langle Dom; S_1^{\mathcal{M}}, \ldots, S_l^{\mathcal{M}} \rangle$ denote a finite structure over universe Dom (say $Dom = \{0, \ldots, n-1\}$) with interpretation for the symbols of $\mathcal{S}$. When no confusion is possible, we will not distinguish between a symbol and its interpretation. Let $\mathcal{M} = \langle Dom; \mathcal{S} \rangle$ be an $\mathcal{S}$-structure and $\mathcal{R}$ be a signature disjoint from $\mathcal{S}$. Structure $(\mathcal{M}, \mathcal{R})$ is an extension of $\mathcal{M}$ i.e. a $\mathcal{S} \cup \mathcal{R}$-structure $\mathcal{M}' = \langle Dom; \mathcal{S}, \mathcal{R} \rangle$.
Let $\mathbf{wa} = \{0, <, +\}$ (**wa** stands for "weak arithmetic"). Symbols of **wa** will always be interpreted respectively as the natural 0, order relation and addition function over Dom.

We are interested in a restricted class of second-order formulas with a first-order part containing only one variable which is universally quantified.

Definition 1 *Let $\mathcal{C}$ be a class of finite unary functions. Let $\exists\mathcal{C}\,\mathbf{FO}^{+<}(\forall)$ the class of model sets of second-order formulas of the form:*

$$\exists f_1 \ldots \exists f_p \; \forall x \, \phi(x, f_1, \ldots, f_p, \mathcal{S}, \mathbf{wa})$$

where ϕ is quantifier-free, $\mathcal{S}$ is a signature and the interpretation of the unary functions f_i belongs to $\mathcal{C}$.

Remark 1 *Note that we do not assume here that formulas we consider are invariant under order and addition. In other words, arithmetic predicates are part of the input and the fact a formula is true in a structure may depend on the particular choice of these predicates (as it is the case for scheduling problems). All the results of that paper will be given in that context.*
However, it is known that when restricted to properties independent of the choice of the underlying arithmetic predicates, any one variable formula over unary functions, linear order and addition function can be replaced by an equivalent formula using unary functions only (see [Gra90, GO98] for precise statements). Then, in that case, arithmetic predicates do not increase the expressive power of unary functions (but increase that of bijections).

In the rest of the paper, we will consider that $\mathcal{C}$ is either *Func* the class of all finite unary functions or *Bij* the class of finite bijections.

2 Unary functions vs. bijections

The main result of this paper asserts that as long as interpretations are taken among weakly arithmetized structures (i.e. over structures with built-in order and addition) the expressive power of unary functions and of bijections is the same. Moreover the interpretation of formulas of the one kind into formulas of the second kind can preserve the first-order quantifier structure. In its strongest form, the result can be stated as follows.

Theorem 2 $\exists Bij\,\mathbf{FO}^{+<}(\forall) = \exists Func\,\mathbf{FO}^{+<}(\forall)$

One inclusion can be easily proved.

Proposition 3 $\exists Bij\,\mathbf{FO}^{+<}(\forall) \subseteq \exists Func\,\mathbf{FO}^{+<}(\forall)$

Proof. Let $\mathcal{E} \in \exists Bij\,\mathbf{FO}^{+<}(\forall)$. Then, there exists a formula $\varphi(x, b_1, \ldots, b_q, \mathcal{S})$ such that for every $\mathcal{M} = \langle Dom; \mathcal{S}, \mathbf{wa}\rangle$, $\mathcal{M} \in \mathcal{E}$ if and only if there exist permutations $b_1, \ldots, b_q$ such that $(\mathcal{M}, b_1, \ldots, b_q) \models \varphi(b_1, \ldots, b_q, \mathcal{S})$.
Let $f_1, \ldots, f_q$ and $sk_1, \ldots, sk_q$ be unary functions symbols. Let $\varphi'(f_1, \ldots, f_q, \mathcal{S})$ be the Formula obtained from φ by replacing each b_i by f_i. For each i, Skolem function sk_i is used to constrain function f_i to be bijective as follows:

$$\psi :\equiv \bigwedge_{i=1}^{q} \forall x\, f_i(sk_i(x)) = x.$$

Formula ψ states that each function f_i is surjective. It is clear that $\mathcal{E}$ is the model set of $\overline{\exists f}\,\overline{\exists sk}\ \varphi' \wedge \psi$. □

2.1 Normal forms for $\exists Func\,\mathbf{FO}^{+<}(\forall)$

In order to prove the second inclusion, we first propose a normal form for formulas which define properties in $\exists Func\,\mathbf{FO}^{+<}(\forall)$.

Proposition 4 *Let $\mathcal{S} = R_{\mathcal{S}} \cup F_{\mathcal{S}}$ be a signature with $R_{\mathcal{S}}$ (resp. $F_{\mathcal{S}}$) containing relation (resp. function) symbols only. Let P be a property on $\mathcal{S}$-structures definable in $\exists Func\,\mathbf{FO}^{+<}(\forall)$, then P can be defined by a formula of the form:*

$$\exists f_1 \ldots \exists f_p\ \forall x\, \phi(x, f_1, \ldots, f_p, \mathcal{S}, \mathbf{wa})$$

where the atoms of ϕ are of one of the three following kinds exclusively:

1. *For $R \in R_{\mathcal{S}}$: $R(t_1(x), \ldots, t_k(x))$ where each $t_i(x) \in \{x, f_1(x), \ldots, f_p(x)\}$*
2. *For $F \in F_{\mathcal{S}}$: $F(t_1(x), \ldots, t_k(x)) = f_j(x)$ where f_j and each $t_i(x) \in \{x, f_1(x), \ldots, f_p(x)\}$*
3. *$f_{i_1}(f_{i_2}(x)) \sim f_{i_3}(x)$ for $i_1, i_2, i_3 \in \{1, \ldots, p\}$ and $\sim \in \{=, \neq, <\}$ or $f_{i_1}(x) + f_{i_2}(x) = f_{i_3}(x)$*

Proof. Let $\Phi \equiv \exists f_1 \ldots \exists f_p\ \forall x\, \phi(x, f_1, \ldots, f_p, \mathcal{S}, \mathbf{wa})$ define $\mathcal{P}$ in $\exists Func\,\mathbf{FO}^{+<}(\forall)$.
Let Id be the (unary) identity function. Also, when $k = 0$ the expression $f_1 \ldots f_k(x)$ stands for x. In full generality, atoms of Φ may be of the following forms:

1. $f_{i_1} f_{i_2} \ldots f_{i_{k_1}} F(t_1(x), \ldots, t_{k_3}(x)) \sim f_{j_1} f_{j_2} \ldots f_{j_{k_2}} F'(t'_1(x), \ldots, t'_{k_4}(x))$ where $F, F' \in F_{\mathcal{S}} \cup \{+\} \cup \{Id\}$, where $\sim \in \{=, \neq, <\}$ and where the $t_i(x)$ and $t'_i(x)$ are terms over $F_{\mathcal{S}} \cup \{f_1, \ldots, f_k\} \cup \{+\} \cup \{Id\}$.
2. $R(t_1(x), \ldots, t_{k_1}(x))$ where $R \in R_{\mathcal{S}} \cup \{<\}$ and the $t_i(x)$ are terms over $F_{\mathcal{S}} \cup \{f_1, \ldots, f_k\} \cup \{+\} \cup \{Id\}$.

In the normalization, the addition function $+$ and the order relation $<$ are treated in the same manner as function and relation symbols of $\mathcal{S}$.
Atoms of the first kind may be split into:

$$\begin{array}{l} f_{i_1} f_{i_2} \dots f_{i_{k_1}}(F(t_1(x), \dots, t_{k_3}(x)) = g_1(x) \wedge \\ f_{j_1} f_{j_2} \dots f_{j_{k_2}}(F'(t'_1(x), \dots, t'_{k_4}(x)) = g_2(x) \wedge \\ g_1(x) \sim g_2(x) \end{array}$$

where g_1, g_2 are new unary function symbols. The last conjunct is now of the right form. The first conjunct (similarly, for the second), can again be transformed into:

$$f_{i_1} f_{i_2} \dots f_{i_{k_1}} g_3(x) = g_1(x),$$

It is not hard to see that, by introducing new quantified unary function symbols, atoms of this form can be transformed into conjunction of atoms of the form $f_{j_1} f_{j_2}(x) = f_{j_3}(x)$ i.e. with at most one composition. However, one has to add a constraint in conjunction with Φ that states:

$$\forall x \ F(t_1(x), \dots, t_{k_3}(x)) = g_3(x)$$

The left-side term of that latter formula may not be of one of the required form but, one has to remark that its compositional "depth" is strictly smaller than that of the term we start with. The process (applying the same rules) can now be continued inductively with each term $t_1(x), \dots, t_{k_3}(x)$. Finally, all generated formulas will be as required. Terms of the form $f_{i_1}(x) + f_{i_2}(x) = g_3(x)$ will appear when F is the symbol $+$.
Arguments to transform atoms of the other kind are similar. □

2.2 Definition of some basic predicates

In this section, we define predicates (first-order formulas with one free variable) that describe basic properties of the second-order bijective unary functions. They will be useful to give a high level description of the proof of the main result. Each formula below involves basic arithmetic predicates ($<$ and $+$), unary functions which are all bijective and one variable.
Let *pred*() and *succ*() be two new quantified unary function symbols. We first state that *pred*() and *succ*() are the predecessor and successor (cyclic) functions for our given linear order $<$.

$$\begin{array}{lll} \textsc{successor}(x) & :\equiv & x < \mathit{succ}(x) \ \vee \ \mathit{succ}(x) = 0 \\ \textsc{predecessor}(x) & :\equiv & \mathit{pred}(\mathit{succ}(x)) = x \end{array}$$

$\textsc{successor}(x)$ and $\textsc{predecessor}(x)$ will appear latter as sub-formulas of the constructed formula. A new abbreviation *max* defined by $max = pred(0)$ is introduced.
Again, we now introduce new function symbols and state their properties. Predicate *even*() below evaluates to true for all even elements of the domain w.r.t. natural $<$. As predicate symbols can not formally be used in our language we introduce $even(x)$ as a shortcut for $\beta(x) = x$ where β is a new bijective function. Function $div_2^0()$ (resp.

$div_2^1()$) maps each even (resp. odd) x to $\lfloor x/2 \rfloor$. As only bijections can be used, we need separate symbols for odd and even preimages. These properties of $even()$, $div_2^0()$ and $div_2^1()$ are described by the formulas $\textsc{even}(x)$ and $\textsc{half}(x)$ that follow:

$$\begin{array}{rl}
\textsc{even}(x) :\equiv & even(0) \wedge \\
& [(succ(x) < \mathrm{max} \wedge even(x)) \\
& \rightarrow (\neg even(succ(x)) \;\wedge\; even(succ^2(x)))]
\end{array}$$

$$\begin{array}{rcl}
\textsc{half}(x) & :\equiv & div_2^0(0) = 0 \;\wedge \\
& & [(even(x) \;\wedge\; succ(x) < max) \\
& & \rightarrow div_2^0(succ^2(x)) = succ(div_2^0(x))] \wedge \\
& & div_2^1(1) = 0 \;\wedge \\
& & [(odd(x) \;\wedge\; succ(x) < max) \\
& & \quad \rightarrow div_2^1(succ^2(x)) = succ(div_2^1(x))]
\end{array}$$

From $<$, it is possible to define any linear order $<_i$ by simply guessing the permutation b_i that maps successive elements for $<_i$ to successive ones for $<$ i.e. $x <_i y$ is used as shorthand for $b_i(x) < b_i(y)$. The same approach works also for addition and we set $x +_i y = z$ for $b_i(x) + b_i(y) = b_i(z)$. The following formula just defines such a permutation $b_i()$ together with its reciprocal $b_i^{-1}()$.

$$\textsc{order}_i(x) \quad :\equiv \quad b_i^{-1}(b_i(x)) = x$$

Then, every possible arithmetic predicate or function previously defined can be translated in the context of each order. In the following:

$$\begin{array}{ccl}
max_i & \text{stands for} & b_i^{-1}(max), \text{ i.e. } b_i^{-1}(pred(0)) \\
0_i & \rule{2cm}{0.4pt} & b_i^{-1}(0) \\
succ_i(x) & \rule{2cm}{0.4pt} & b_i^{-1}(succ(b_i(x))) \\
even_i(x) & \rule{2cm}{0.4pt} & even(b_i(x)) \\
div_{2i}^0(x) & \rule{2cm}{0.4pt} & b_i^{-1}(div_2^0(b_i(x))) \\
div_{2i}^1(x) & \rule{2cm}{0.4pt} & b_i^{-1}(div_2^1(b_i(x))),
\end{array}$$

2.3 Expressing unary function by bijections over weakly arithmetized structures

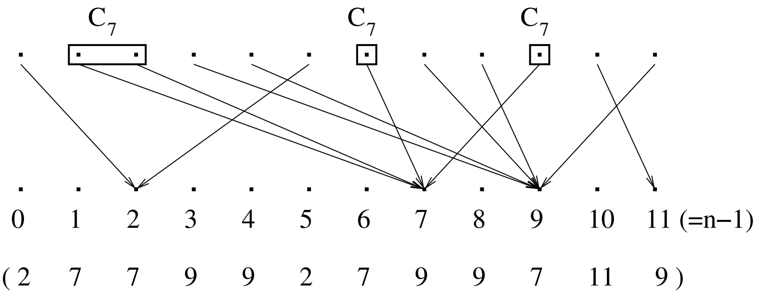
We are now ready to prove the most difficult part of our result.

Proposition 5 $\exists Func\, \mathbf{FO}^{+<}(\forall) \;\subseteq\; \exists Bij\, \mathbf{FO}^{+<}(\forall)$

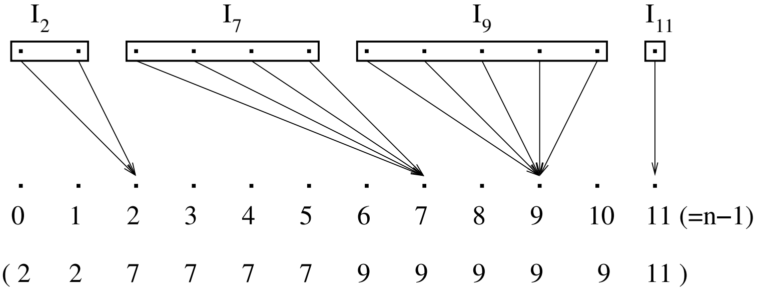
Proof. Let $\mathcal{S} = R_\mathcal{S} \cup F_\mathcal{S}$ with $R_\mathcal{S}$ (resp. $F_\mathcal{S}$) being a purely relational (resp. functional) signature. Let P be a property on $\mathcal{S}$-structures definable in $\exists Func\, \mathbf{FO}^{+<}(\forall)$ i.e. by a formula Ψ of the form:

$$\exists f_1 \ldots \exists f_p \;\forall x\, \psi(x, f_1, \ldots, f_p, \mathcal{S}, \mathbf{wa})$$

We will prove that P is in $\exists Bij\,\mathbf{FO}^{+<}(\forall)$. We first present an overview of the proof which describes how a unary function can be encoded by several bijections on ordered structures (with addition). The constraint that the encoding must be expressed by a formula with only one (universally quantified) variable complicates the construction. Let f_i be a function whose graph is described below.

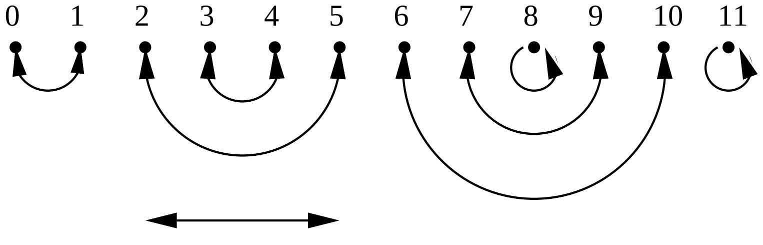


A partition of the elements of the domain Dom can be done depending on their image by f_i. We set: $x \sim_i y :\Leftrightarrow f_i(x) = f_i(y)$. Two elements of Dom are in the same equivalence class if they have the same image by f_i. Each equivalence class C_z represents the preimages of element z. A new linear order $<_i$ over Dom is constructed so that elements in the same equivalence class are contiguous. As we saw in the preceding section, any linear order $<_i$ can be obtained from $<$ by guessing a bijection b_i and setting $x <_i y$ (for $y \neq 0$) iff $b_i(x) < b_i(y)$.



($I_7 = b_i(C_7)$, e.g.)

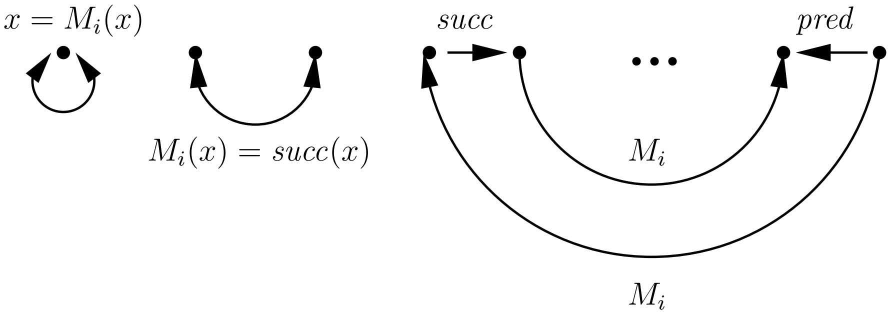
Once such a partition into intervals has been constructed, we need to group together all elements belonging to the same equivalence classes. To this aim, a new bijective "matching" M_i is introduced. In the context of our example, M_i looks like:



Properties of matching M_i are described by the following formula:

$$\begin{aligned}\textsc{matching}_i(x) :\equiv \ & M_i^2(x) = x \wedge [\, (x <_i M_i(x) \wedge M_i(x) \neq \mathit{succ}_i(x)) \\ & \rightarrow M_i(\mathit{succ}_i(x)) = \mathit{pred}_i(M_i(x)) \,],\end{aligned}$$

Formula $\textsc{matching}_i(x)$ states that, for $x <_i M_i(x)$, either $M_i(x) = x$ or $M_i(x) = \mathit{succ}_i(x)$ or $M_i(\mathit{succ}_i(x)) = \mathit{pred}_i(M_i(x))$ i.e. behaves as shown in the following picture:

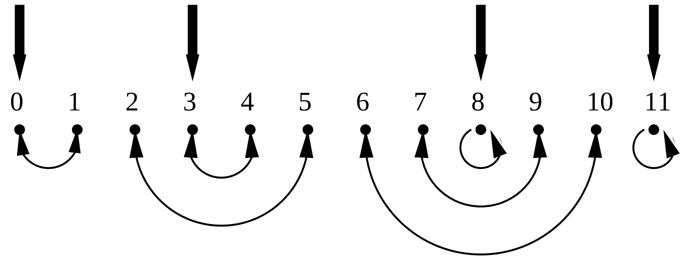


This construction is repeated for every unary function f_i. Let Φ_1 be the following formula.

$$\Phi_1 :\equiv \forall x \ \textsc{successor}(x) \wedge \textsc{predecessor}(x) \wedge \bigwedge_{i=1}^{p} \textsc{order}_i(x) \wedge \textsc{matching}_i(x),$$

Any model of Φ_1 is structured, for each i, into sequences of intervals as described above.

A nice feature of each function M_i is that each interval has an invariant. For example, it is not hard to check that for each interval I, the sum $x +_i M_i(x)$ is the same for every $x \in I$. However, as that sum can exceed the size of the domain, an alternative invariant must be found and the function $\mathit{avg}_i(x) = \left\lfloor \frac{x +_i M_i(x)}{2} \right\rfloor$ which computes the "average" on each interval will be used instead. In our example, $0, 3, 8$ and 11 are the invariants of each interval.



The existence of such an invariant per interval will make the encoding of each function f_i by permutations easy. Let $\mathcal{N}$ be such that $\mathcal{N} \models \Phi_1$. Then, for every i, by definition of M_i, the domain is structured into, say, m intervals $I_i^1, \ldots, I_i^m$ such that:

$$\forall i \forall x, y \in I_i^j \ \mathit{avg}_i(x) = \mathit{avg}_i(y) = z$$

Elements in the same intervals can then be seen as the preimages of the same element for some function f_i. Conversely, let $\mathcal{M} = \langle \mathit{Dom}; f_1, \ldots, f_k \rangle$ be some structure with

only unary function symbols. For every i, denote by $I_i(z_1), \ldots, I_i(z_m)$, the partition of Dom into sets of preimages (i.e. $\{z_1, \ldots, z_m\}$ is the image set of f_i). Then, there exists $\mathcal{N}$ such that $\mathcal{N} \models \Phi_1$ and such that, for every i, M_i realizes a partition of Dom into the same m intervals $I_i(z_1), \ldots, I_i(z_m)$ for which the $avg_i()$ function is constant.
It remains now to explicitly relate, for each i, the value of the average function on each interval and the corresponding image of f_i. More precisely, a new bijection $Val_i()$ is introduced and each f_i is encoded by:

$$f_i(x) = y \text{ if and only if } Val_i(avg_i(x)) = y.$$

Similarly, composition of functions like $f_i(f_j(x)) = f_k(x)$ is encoded by:

$$Val_i(avg_i(Val_j(avg_j(x))) = Val_k(avg_k(x))).$$

Two technical steps remain to achieve the proof. First, we have to "implement" each $avg_i()$ functions by means of bijections. Then, we have to make precise the substitution of "functional" terms by "bijective" terms into formula Ψ.

Implementation of the average function. It is not hard to see that function, $avg_i()$ satisfies the following equality:

$$avg_i(x) = \left\lfloor \frac{x +_i M_i(x)}{2} \right\rfloor = \left\lfloor \frac{x}{2} \right\rfloor +_i \left\lfloor \frac{M_i(x)}{2} \right\rfloor +_i carry$$

The value of $avg_i()$ is defined by cases depending on the parity of x and $M_i(x)$.

$$avg_i : x \mapsto \begin{cases} div^0_{2i}(x) +_i div^0_{2i}(M_i(x)) \\ \qquad\qquad\qquad\qquad \text{if } x \text{ and } M_i(x) \text{ are even} \\ (div^1_{2i}(x) +_i div^1_{2i}(M_i(x))) +_i 1 \\ \qquad\qquad\qquad\qquad \text{if } x \text{ and } M_i(x) \text{ are odd} \\ div^1_{2i}(x) +_i div^0_{2i}(M_i(x)) \\ \qquad\qquad\qquad\qquad \text{if } x \text{ is odd and } M_i(x) \text{ is even} \\ div^0_{2i}(x) +_i div^1_{2i}(M_i(x)) \\ \qquad\qquad\qquad\qquad \text{if } x \text{ is even and } M_i(x) \text{ is odd} \end{cases}$$

Such a definition requires to define “even” and “odd” predicates and the “division by two” function (what we did in section 2.2). Let Φ be the following formula:

$$\Phi \;:\equiv \Phi_1 \wedge \textsc{even}(x) \wedge \textsc{half}(x).$$

End of the proof: substitution of atoms Let Ψ be the formula that we have to transform. As we have seen, atoms of Ψ are of one of the following three kinds:

1. For $R \in R_{\mathcal{S}}$: $R(t_1(x), \ldots, t_k(x))$ where each $t_i(x) \in \{x, f_1(x), \ldots, f_p(x)\}$
2. For $F \in F_{\mathcal{S}}$: $F(t_1(x), \ldots, t_k(x)) = f_j(x)$ where f_j and each $t_i(x) \in \{x, f_1(x), \ldots, f_p(x)\}$
3. $f_{i_1}(f_{i_2}(x)) \sim f_{i_3}(x)$ for $i_1, i_2, i_3 \in \{1, \ldots, p\}$ and $\sim \in \{=, \neq, <\}$ or $f_{i_1}(x) + f_{i_2}(x) = f_{i_3}(x)$

We detail only how atoms of the form $f_{i_1}(f_{i_2}(x)) = f_{i_3}(x)$ are replaced. The other cases that do not use composition of guessed functions are similar and even simpler. It should be clear until now that each function f_i is encoded as:

$$f_i(x) = \begin{cases} \overbrace{Val_i(div^0_{2i}(x) +_i div^0_{2i}(M_i(x)))}^{f_i^{\mathbf{1}}(x)} & \\ & \text{if } x \text{ and } M_i(x) \text{ are even} \\ \overbrace{Val_i(div^0_{2i}(x) +_i div^1_{2i}(M_i(x)))}^{f_i^{\mathbf{2}}(x)} & \\ & \text{if } x \text{ is even and } M_i(x) \text{ is odd} \\ \underbrace{Val_i(div^1_{2i}(x) +_i div^0_{2i}(M_i(x)))}_{f_i^{\mathbf{3}}(x)} & \\ & \text{if } x \text{ is odd and } M_i(x) \text{ is even} \\ \underbrace{Val_i(succ_i(div^1_{2i}(x) +_i div^1_{2i}(M_i(x))))}_{f_i^{\mathbf{4}}(x)} & \\ & \text{if } x \text{ and } M_i(x) \text{ are odd} \end{cases}$$

Then, each composition $f_i(f_j(x)) = f_k(x)$ is replaced with the huge conjunction (over $4^3 = 64$ terms!) sketched below:

$$\bigwedge \begin{cases} [even_i(f_j^{\mathbf{1}}(x)) \wedge even_i(f_j^{\mathbf{1}}(x)) \wedge \\ even_j(x) \wedge even_j(M_j(x)) \wedge \\ even_k(x) \wedge even_k(M_k(x)) \\ \qquad\qquad\qquad\qquad \longrightarrow f_i^{\mathbf{1}}(f_j^{\mathbf{1}}(x)) = f_k^{\mathbf{1}}(x)] \\ [even_i(f_j^{\mathbf{1}}(x)) \wedge even_i(f_j^{\mathbf{1}}(x)) \wedge \\ even_j(x) \wedge even_j(M_j(x)) \wedge \\ even_k(x) \wedge odd_k(M_k(x)) \\ \qquad\qquad\qquad\qquad \longrightarrow f_i^{\mathbf{1}}(f_j^{\mathbf{1}}(x)) = f_k^{\mathbf{2}}(x)] \\ [even_i(f_j^{\mathbf{1}}(x)) \wedge even_i(f_j^{\mathbf{1}}(x)) \wedge \\ even_j(x) \wedge even_j(M_j(x)) \wedge \\ odd_k(x) \wedge even_k(M_k(x)) \\ \qquad\qquad\qquad\qquad \longrightarrow f_i^{\mathbf{1}}(f_j^{\mathbf{1}}(x)) = f_k^{\mathbf{3}}(x)] \\ \vdots \\ [odd_i(f_j^{\mathbf{4}}(x)) \wedge odd_i(f_j^{\mathbf{4}}(x)) \wedge \\ odd_j(x) \wedge odd_j(M_j(x)) \wedge \\ odd_k(x) \wedge odd_k(M_k(x)) \\ \qquad\qquad\qquad\qquad \longrightarrow f_i^{\mathbf{4}}(f_j^{\mathbf{4}}(x)) = f_k^{\mathbf{4}}(x)] \end{cases}$$

For the other kinds of atoms, the pre-condition of each conjunct depends on the parity of x and $M_i(t_i(x))$. Let Ψ' be the formula obtained from Ψ after the above substitution and $\Psi^t = \Phi \wedge \Psi'$. It is clear that, for every $\mathcal{M}$:

$$\mathcal{M} \models \Psi \iff \mathcal{M} \models \Psi^t$$

□

3 Two applications of the main result

3.1 Logical characterization of non-deterministic linear time

Let $\mathcal{S}$ be some signature. Let $NTIME^{\mathcal{S}}(n^d)$ be the class of $\mathcal{S}$-properties decidable by a non-deterministic Random Access Machine in time $O(n^d)$. The following theorem, due to Grandjean and Olive, gives a very precise logical description of subclasses of NP. It pursues and, in some sense, achieves a long standing line of research in that direction (see [Fag74, GO98, Imm99, Lyn92]).

Theorem 6 *[GO04] For any signature $\mathcal{S}$ and any $d > 0$, properties in $NTIME^{\mathcal{S}}(n^d)$ are exactly those definable by formulas of the form:*

$$\exists f_1 \ldots \exists f_k \, \forall^d \overline{x} \; \phi(\overline{f}, \overline{x}, \mathcal{S})$$

where the f_i are d-ary function symbols and $\overline{x}$ is a d-tuple of first-order variables (all universally quantified).

When $d = 1$ and considering properties on weakly arithmetized structures, the following corollary can be easily derived.

Corollary 7 *A property P on $S \cup \mathbf{wa}$-structure is decidable in non-deterministic linear time on a RAM if and only if it belongs to $\exists Bij \, \mathbf{FO}^{+<}(\forall)$.*

3.2 Skolemization of first-order formulas with built-in addition

A second immediate corollary of our result concerns Skolem normal forms of first-order formulas. Informally, it shows that, provided interpretation is taken over weakly arithmetized structures, formulas of first-order prefix $\forall\exists^*$ can be skolemized into formulas of first-order prefix $\forall$ by simply guessing bijections. This corollary is surprising in the sense that there is no bijective correspondance "a priori" between the universally quantified variable and the existentially quantified ones. Of course, the formula obtained after that "skolemization" process is much more complicated than the starting formula (cf. proof of proposition 5)

Corollary 8 *Let $\mathcal{S}$ be a signature and Φ be a first-order formula of the form:*

$$\forall x \overline{\exists y} \; \varphi(x, \overline{y}, \mathcal{S}, \mathbf{wa})$$

Then, there exists a second order formula Ψ of the form:

$$\exists b_1 \ldots \exists b_p \; \forall x \, \phi(x, b_1, \ldots, b_p, \mathcal{S}, \mathbf{wa})$$

where the b_i are unary function symbols whose interpretations are restricted to be bijective (the number p depends on $|\overline{y}|$) and such that for every $\mathcal{S}$-structure $\mathcal{M}$:

$$\mathcal{M} \models \Phi \iff \mathcal{M} \models \Psi.$$

4 Conclusion and perspective

In this paper, we have proven that, on structures with built-in addition and order relation, bijective unary functions are as expressive as arbitrary unary functions (in the context of existential second-order logic with fixed first-order quantifier form). We think that such a result, though interesting on its own, may be a preliminary step for a descriptive approach of the complexity of scheduling problems. In that direction, it could be interesting to obtain strong normal forms (e.g. purely conjunctive first-order part like in [Oli98]) for the class of formulas we have considered in this paper. Also, it could be interesting to know if the main result of this paper still holds when only a built-in linear order relation is allowed (i.e. without addition).

Acknowledgments. We thank two anonymous referees for their remarks and suggestions that really improve readability of the paper.

References

[Dur02] A. Durand. Binary-NP and the power of one first-order universal quantifier. *Information and Computation*, 178(1):12–22, 2002.

[DLS98] A. Durand, C. Lautemann, T. Schwentick. *Subclasses of Binary NP.* Journal of Logic and Computation, 8(2), pp.189–207, 1998.

[EF95] H.-D. Ebbinghaus, J. Flum. *Finite Model Theory.* Springer-Verlag, 1995.

[Fag74] R. Fagin. *Generalized first-order spectra with one binary predicate.* Complexity of Computation, SIAM AMS Proceedings, vol7, pp.43–73, 1974.

[GJ79] M.R. Garey, D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman and Co, 1979.

[Gra90] É. Grandjean. *First-order spectra with one variable.* J. of Computer and Systems Sciences., vol.40(2):pp.136–153, 1990.

[Gra94] É. Grandjean. *Linear time algorithms and NP-complete problems.* SIAM Journal on Computing, 23(3), pp.573–597, 1994.

[GO98] É. Grandjean, F. Olive. *Monadic logical definability of nondeterministic linear time.* Computational Complexity, 7(1), pp.54–97, 1998.

[GO04] É. Grandjean, F.Olive. *Many graph properties are easier to check than you think.* Journal of Computer and System Science, vol 68(3), pp. 546-597, 2004,

[Imm99] N. Immerman. *Descriptive Complexity.* Springer-Verlag, 1999.

[Lyn92] J.F. Lynch, *The quantifier structure of sentences that characterize nondeterministic time complexity.* Computational Complexity, pages 40–66, 1992.

[Oli98] F. Olive, *A conjunctive logical characterization of nondeterministic linear time.* Proc. 11th Annual Conference of the EACSL (CSL'97), LNCS 1414, pp. 360–372, 1998.