Logical Characterizations of $\mathrm{P}_{\mathcal{K}}$ and $\mathrm{NP}_{\mathcal{K}}$ over an Arbitrary Structure $\mathcal{K}$

O. Bournez[1], F. Cucker*[2], P. Jacobé de Naurois[1], and J.-Y. Marion[1]

[1] LORIA, 615 rue du Jardin Botanique,BP 101, 54602 Villers-lès-Nancy Cedex, Nancy, FRANCE
e-mail: {Olivier.Bournez,Paulin.De-Naurois,Jean-Yves.Marion}@loria.fr
[2] Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, HONG KONG
e-mail: macucker@math.cityu.edu.hk

Abstract. We focus on the BSS model of computation over arbitrary structures. We propose a modification of the logical characterizations of $\mathrm{P}_{\mathcal{K}}$ and $\mathrm{NP}_{\mathcal{K}}$ given by Grädel and Gurevich in order to make them applicable to any computational structure $\mathcal{K}$.

1 Introduction

In the last decades complexity theory developed in many directions to offer a broad perspective of the complexity of computations. Two directions relevant to our work are the extension of complexity theory to domains other than finite alphabets and the characterizations of complexity classes in machine-independent terms.

A seminal paper for the first direction above is the one by Blum, Schub and Smale [BSS89], where a theory of computation and complexity that allowed an ordered ring or field as alphabet for the space of admissible inputs was developed. The authors emphasized the case when the ring is the field of real numbers, $\mathbb{R}$, bringing the theory of computation into the domain of analysis, geometry and topology. Later on extended to the more abstract level of computations over arbitrary structures in [Goo94,Poi95], this BSS model, among other things, makes use of the extensively developed subject of the theory of discrete computations, initiated by the work of Turing [Tur36].

In the classical setting of computation over finite alphabets, a great amount of work has been done, along the second direction to provide machine independent characterization of the major complexity classes. Such characterizations lend further credence to the importance of the complexity classes considered, relate them to issues relevant to programming and verification, and, in general, help understand the notion of computability in a whole perspective. Several approaches for designing such characterizations have been chosen, among which one can find descriptive complexity (global methods of finite model theory).

* Partially supported by City University of Hong Kong SRG grant 7001290.

Descriptive complexity began with the work of Fagin [Fag74], who proved, in the classical setting, that the class NP can be characterized as the class of sets describable within existential second-order logic. Subsequently, Vardi and Immerman [Var82,Imm83,Imm86] used this approach to characterize P. Several other characterizations exist, for classes like LOGSACE [Gur83] or PSPACE [Mos83,GS86,Imm87,Bon88,AV89,Lei90,ASV90,AV91,Imm91]. An overview of the subject can be found in [EF95,Imm99]. These characterizations, however, applied originally only to the classical setting of computation over finite alphabets since only finite models were considered.

In [GM95], the notion of $\mathbb{R}$-structure has been introduced, and characterization of deterministic polynomial time $\mathrm{P}_{\mathbb{R}}$ and non-deterministic polynomial time $\mathrm{NP}_{\mathbb{R}}$ in terms of logics over these $\mathbb{R}$-structures were provided. These results have been later on extended in [CM99], in order to capture other complexity classes, and in [GG98] over structures other than $\mathbb{R}$.

In this paper, we extend the notions of $\mathbb{R}$-structures to $\mathcal{K}$-structures over an arbitrary computational structure $\mathcal{K}$, extend the notion of first order logic and second order logic over $\mathbb{R}$-structures of [GM95] to first and second-order logic over $\mathcal{K}$-structures. We also define a proper notion of fixed-point rule, different from [GM95], allowing to capture $\mathrm{P}_{\mathcal{K}}$, and propose a characterization of $\mathrm{NP}_{\mathcal{K}}$. Our characterizations, while only slightly different from the ones of [GG98], apply to any computational structure. In particular, when $\mathcal{K}$ is $\mathbb{Z}_2$, they coincide with the classical ones.

Indeed, in the characterizations of [GM95,CM99], some logic is hidden in basic computations over the real numbers. In particular, these characterizations assume that the set of real numbers is equipped with a sign function and a multiplication: therefore they cannot be trivially extended, say, to the additive setting or to the field of complex numbers. Similar restrictions apply to the more general characterizations of [GG98], where it is required that the underlying computational structure contains an expansion of $(\mathbb{N}, 0, 1+, -, <, \max, \min, \Sigma, \Pi)$. Here again, this does not apply to the field of complex number, or to finite structures. Our characterizations overcome these drawbacks.

2 Computing over Arbitrary Structures

This section is devoted to a brief exposition of the BSS model of computation, based on [BCSS98] and [Poi95].

2.1 Arbitrary Structures

The following notion of *structure* describes the domain of elements over which a machine computes, as well as the operations and relations that can be performed over these elements.

Definition 1. (Structure): *A* structure $\mathcal{K} = (\mathbb{K}, \{op_i\}_{i\in I}, rel_1 \ldots, rel_l, \mathbf{0}, \mathbf{1})$ *is given by some underlying set* $\mathbb{K}$*, some operators* $\{op_i\}_{i\in I}$*, and a finite number of relations* $rel_1, \ldots, rel_l$.

Constants correspond to operators of arity 0. While the index set I may be infinite, the number of operators with arity greater than 1 is assumed to be finite, that is, only symbols for constants may be infinitely many. We will not distinguish between operator and relation symbols and their corresponding interpretations as functions and relations of the same arity, respectively over the underlying set $\mathbb{K}$. We assume that the equality relation $=$ is a relation of the structure, and that there are at least two constant symbols, with different interpretations (denoted by $\mathbf{0}$ and $\mathbf{1}$ in our work) in the structure.

The main examples of structures are $\mathbb{Z}_2 = (\{0,1\}, =, \mathbf{0}, \mathbf{1})$, $\mathbb{R} = (\mathbb{R}, +, -, *, /, <, \{c \in \mathbb{R}\})$ and $\mathbb{R}_{ovs} = (\mathbb{R}, +, -, <, \{c \in \mathbb{R}\})$. When considered over $\mathbb{Z}_2$, the notions of BSS computability and complexity coincide with the one of the classical Turing model.

2.2 BSS Machines

Denote by $\mathcal{K}$ a structure as above. We now define machines over $\mathcal{K}$ following the lines of [BCSS98]. A machine over $\mathcal{K}$ is roughly a Turing machine with a bi-infinite tape, where the tape cells holds elements of $\mathbb{K}$. A machine can perform operations and relations of $\mathcal{K}$ over its tape elements at unit cost. More formally,

Definition 2. (Machine): *A* machine over $\mathcal{K}$ *consists of an input space* $\mathcal{I} = \mathbb{K}^* = \bigcup_{n\in\mathbb{N}} \mathbb{K}^n$, *an output space* $\mathcal{O} = \mathbb{K}^*$, *and a register space*[3] $\mathcal{S} = \mathbb{K}_* = \{(x_i, \ldots, x_j), i \in \mathbb{Z}, j \in \mathbb{Z}, x_{i\leq k\leq j} \in \mathbb{K}\}$, *together with a connected directed graph whose nodes labelled* $0, \ldots, N$ *correspond to the set of different* instructions *of the machine. These nodes are of one of the six following types: input, output, computation, copy, branching and shift nodes.*

1. *Input nodes.* There is only one input node and it is labelled with 0. Associated with this node there is a next node $\beta(0)$, and the input map $g_I : \mathcal{I} \to \mathcal{S}$.
2. *Output nodes.* There is only one output node which is labelled with 1. It has no next nodes, once it is reached the computation halts, and the output map $g_O : \mathcal{S} \to \mathcal{O}$ places the result of the computation in the output space.
3. *Computation nodes.* Associated with a node m of this type there are a next node $\beta(m)$ and a map $g_m : \mathcal{S} \to \mathcal{S}$. The function g_m replaces the component indexed by 0 of $\mathcal{S}$ by the value $op(w_0, \ldots, w_{n-1})$ where $w_0, w_2, \ldots, w_{n-1}$ are components 0 to $n-1$ of $\mathcal{S}$ and op is some operation of the structure $\mathcal{K}$ of arity n. The other components of $\mathcal{S}$ are left unchanged. When the arity n is zero, m is a constant node.
4. *Copy nodes.* Associated with a node m of this type there are a next node $\beta(m)$ and a map $g_m : \mathcal{S} \to \mathcal{S}$. The function g_m performs one of the following actions:
   - Replace the component indexed by 0 by a copy of the component indexed by 1. This is denoted as a *copy left*.

[3] In the original paper by Blum, Shub and Smale, this is called the *state* space. We rename it *register* space to avoid confusions with the notion of 'state' in a Turing machine.

- Replace the component indexed by 0 by a copy of the component indexed by -1. This is denoted as a *copy right*.
- Exchange the component indexed by 0 and the component indexed by 1 . This is denoted as a *switch*.

5. *Branch nodes.* There are two nodes associated with a node m of this type: $\beta^+(m)$ and $\beta^-(m)$. The next node is $\beta^+(m)$ if $rel(w_0, \ldots, w_{n-1})$ is true and $\beta^-(m)$ otherwise, where $w_0, w_2, \ldots, w_{n-1}$ are components 0 to $n-1$ of $\mathcal{S}$ and rel is some relation of the structure $\mathcal{K}$ of arity n.
6. *Shift nodes.* Associated with a node m of this type there is a next node $\beta(m)$ and a map $\sigma : \mathcal{S} \to \mathcal{S}$. The σ is either a left or a right shift.

As in the classical Turing model of computation, complexity classes over an arbitrary structure $\mathcal{K}$ can be defined such as $\mathrm{P}_{\mathcal{K}}$ and $\mathrm{NP}_{\mathcal{K}}$, as well as the notions of reduction and completeness. Over the three major structures, natural complete problems exist for $\mathrm{NP}_{\mathcal{K}}$, and the question $\mathrm{P}_{\mathcal{K}} = \mathrm{NP}_{\mathcal{K}}$ remains open.

3 First-Order Logic on $\mathcal{K}$-structures

In the rest of the paper, denote by $\mathcal{K}$ a structure as above. We will consider machines over $\mathcal{K}$, as well as the complexity classes $\mathrm{P}_{\mathcal{K}}$ and $\mathrm{NP}_{\mathcal{K}}$.

3.1 Definitions

We assume that the reader is familiar with first-order mathematical logic. In order to capture computation over arbitrary structure, we extend the notion of $\mathbb{R}$-Structures from [GM95,BCSS98] to the notion of $\mathcal{K}$-Structures. Our $\mathcal{K}$-Structures are particular instances of the meta-finite structures of Grädel and Gurevich [GG98].

Definition 3. **($\mathcal{K}$-Structures)**: *Let L_s, L_f be finite vocabularies, where L_s may contain relation and function symbols with their arities, and L_f contains function symbols only, with their arities. An $\mathcal{K}$-structure of signature $\sigma = (L_s, L_f)$ is a pair $\mathcal{D} = (\mathcal{U}, \mathcal{F})$ consisting of*

(i) *a finite logical structure $\mathcal{U}$ of vocabulary L_s, called the* skeleton *of $\mathcal{D}$, whose (finite) universe A is also called the* universe *of $\mathcal{D}$, and*
(ii) *a finite set $\mathcal{F}$ of functions $f_i^{\mathcal{D}} : A^{k_i} \to \mathbb{K}$ interpreting the function symbols f_i in L_f, with the corresponding arities k_i.*

$\mathcal{K}$-Structures as above are a generalization of the concept of logical structure, over a fixed computational structure $\mathcal{K}$, in order to denote inputs to algorithms performed by BSS machines over $\mathcal{K}$. The skeleton of a $\mathcal{K}$-structure is used for describing the finite, discrete part of a structure, while the set $\mathcal{F}$ is used for describing its computational part.

Example 1. Consider for instance the real case $\mathbb{R}$, and the $\mathrm{NP}_{\mathbb{R}}$-complete problem 4FEAS: decide whether a given polynomial of degree 4 has a real zero. Inputs to this decision problem are polynomials of degree 4, which can be encoded by $\mathbb{R}$-structures of signature $(L_s, L_f) = (\emptyset, \{f_0, f_1, f_2, f_3, f_4\})$ as follows: The universe A contains one variable for each variable of the polynomial. The interpretation of the function $f_4 : A^4 \to \mathbb{R}$ of L_f gives the (real) coefficient for each monomial of degree 4, and similarly for the other coefficients with $f_3, \ldots f_0$.

We present now the notion of first-order logic over $\mathcal{K}$-structures. A key feature of this logic is that the quantified variable range only over the skeleton of the structures, and not over $\mathcal{K}$. Assume that V is a fixed, countable set of variables.

Definition 4. (First-Order Logic for $\mathcal{K}$-Structures): *The language of* first-order logic *for $\mathcal{K}$-structures,* $\mathrm{FO}_{\mathcal{K}}$*, contains for each signature $\sigma = (L_s, L_f)$ a set of terms and formulas. We first define terms, of which there are two kinds. When interpreted in a $\mathcal{K}$-structure $\mathcal{D} = (\mathcal{U}, \mathcal{F})$ with universe A, each term t takes values either in A, in which case we call it an* index term*, or in $\mathbb{K}$, in which case we call it a* number term*. Terms are defined inductively as follows.*

- **(i)** *The set of index terms is the closure of the set V of variables under the application of functions symbols of L_s.*
- **(ii)** *If $h_1, \ldots, h_k$ are index terms, and X is a k-ary function symbol of L_f, then $X(h_1, \ldots, h_k)$ is a number term.*
- **(iii)** *If $t_1, \ldots, t_{k_i}$ are number terms, and op_i is an operation of the computational structure $\mathcal{K}$ of arity k_i, $op_i(t_1, \ldots, t_{k_i})$ is also a number term. In particular, operations of arity 0 of $\mathcal{K}$ yield constant number terms.*

Atomic formulas *are defined as follows.*

- **(i)** *equalities $h_1 = h_2$ of index terms are atomic formulas.*
- **(ii)** *If $t_1, \ldots, t_{k_i}$ are number terms, and rel_i is a relation of the computational structure $\mathcal{K}$ of arity k_i, $rel_i(t_1, \ldots, t_{k_i})$ is an atomic formula. In particular we may consider equalities $t_1 = t_2$ of number terms.*
- **(iii)** *If $h_1, \ldots, h_k$ are index terms, and P is a k-ary relation symbol in L_s, $P(h_1, \ldots, h_k)$ is also an atomic formula.*

The set of formulas *of* $\mathrm{FO}_{\mathcal{K}}$ *is the smallest set containing all atomic formulas and which is closed under Boolean connectives, $\vee, \wedge, \neg$, and quantification $(\exists v)\psi$ and $(\forall v)\psi$.*

Definition 5. (Interpretation of First-Order Logic for $\mathcal{K}$-Structures): *We adopt the usual interpretation. The main point is that the quantifiers range over the universe of the $\mathcal{K}$-Structure: a formula $(\exists x)\psi$ where ψ contains variables $x, y_1, \ldots, y_n$, $n \geq 0$ is* `true` *at $b \in A^n$ if there exists $a \in A$ such that $\psi^{\mathcal{D}}$ is* `true`*, where x is interpreted as a and y_i is interpreted as b_i, for $i = 1, \ldots, n$. Similarly for a formula $(\forall x)\psi$ being interpreted as $\neg(\exists x)\neg\psi$. For details, see the Appendix.*

Definition 6. (Order): *Let $\sigma = (L_s, L_f)$ be a signature. A $\mathcal{K}$-structure $\mathcal{D} = (\mathcal{U}, \mathcal{F})$ of signature σ with universe A is* ordered *if there is a binary relation symbol $\leq$ in L_s whose interpretation $\leq^{\mathcal{D}} \in \mathcal{U}$ is a total order on A.*

Remark 1. Our order relation replaces the ranking function of [GM95,CM99,BCSS98,GG98]. In these papers, the ranking function is a function in L_f, defining a bijection between A and $\{0,\ldots,|A|-1\}$. This uses the property that $\mathbb{R}$ is ordered and contains at least the natural numbers. It is not applicable in our setting of arbitrary structure, where no such requirement can be made on $\mathcal{K}$.

Remark 2. It is checkable in first-order logic over $\mathcal{K}$-structures that a structure is ordered. Based on this order relation, one can also define in first-order logic a lexicographic order over A^k for any $k \in \mathbb{N}$. Therefore, we will freely use the symbol $\leq_k$ for a total order over A^k. In what follows, we will only consider ordered $\mathcal{K}$-structures. Note also that the minimal element of A^k and the maximal one with respect to this lexicographic order are definable in first-order logic. We will denote them 0 and $n^k - 1$ respectively.

Definition 7. (Fixed Point Rule)*: Fix an signature $\sigma = (L_s, L_f)$, D a relation symbol of arity $r \geq 0$ and Z a function symbol of arity r, both not contained in this signature. Let $H(D,t_1,\ldots,t_r), I_1(D,t_1,\ldots,t_r),\ldots,I_{k-1}(D,t_1,\ldots,t_r)$, $k \geq 0$ be first-order formulas over the signature $(L_s \cup \{D\}, L_f \cup \{Z\})$ with free variables $t_1,\ldots,t_r$. Let $F_1(Z,t_1,\ldots,t_r),\ldots,F_k(Z,t_1,\ldots,t_r)$ be number terms over the signature $(L_s \cup \{D\}, L_f \cup \{Z\})$ with free variables $t_1,\ldots,t_r$. We allow D to appear several times in H and the I_i's, but we do not require that its arguments are $(t_1,\ldots,t_r)$. The only restriction is that the number of free variables in H and the I_i's coincide with the arity of D. A similar remark holds for the F_i's and Z. For any $\mathcal{K}$-structure $\mathcal{D}$ of signature σ and any interpretation $\zeta : A^r \to \mathcal{K}$ of Z and $\Delta \subseteq A^r$ of D, respectively, the number terms $F_1(Z,t_1,\ldots,t_r),\ldots,F_k(Z,t_1,\ldots,t_r)$ define functions*

$$F^{\mathcal{D}}_{1,\zeta},\ldots,F^{\mathcal{D}}_{k,\zeta} \ : \ A^r \to \mathcal{K}$$
$$u_1,\ldots,u_r \to \begin{cases} [F_1(Z \leftarrow \zeta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}} \\ \vdots \\ [F_k(Z \leftarrow \zeta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}}, \end{cases}$$

where $[F_i(Z \leftarrow \zeta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}}$ is obtained from $F_k(Z,t_1,\ldots,t_r)$ by replacing any occurrence of Z by ζ, any occurrence of t_j by u_j, and interpreting the whole number term in $\mathcal{D}$. Also, the formulas $H(D,t_1,\ldots,t_r)$ and $I_i(D,t_1,\ldots,t_r)$, $1 \leq i \leq k-1$ define relations

$$H^{\mathcal{D}}_{\Delta}, I^{\mathcal{D}}_{i,\Delta} \subseteq A^r$$
$$u_1,\ldots,u_r \to \begin{cases} [H(D \leftarrow \Delta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}} \\ [I_1(D \leftarrow \Delta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}} \\ \vdots \\ [I_{k-1}(D \leftarrow \Delta, t_1,\ldots,t_r \leftarrow u_1,\ldots u_r)]^{\mathcal{D}}. \end{cases}$$

Let us now consider the sequence of pairs $\{\Delta^i, \zeta^i\}_{i\geq 0}$ *with* $\Delta^i \subseteq A^r$ *and* $\zeta^i : A^r \to \mathcal{K}$ *inductively defined by*

$$\Delta^0(x) = \texttt{false} \quad \textit{for all } x \in A^r$$
$$\zeta^0(x) = \mathbf{0} \quad \textit{for all } x \in A^r$$
$$\Delta^{i+1}(x) = \begin{cases} H^{\mathcal{D}}_{\Delta^i}(x) & \textit{if } \Delta^i(x) = \texttt{false} \\ \texttt{true} & \textit{otherwise} \end{cases}$$
$$\zeta^{i+1}(x) = \begin{cases} F^{\mathcal{D}}_{1,\zeta^i}(x) & \textit{if } \neg\Delta^i(x) \wedge I^{\mathcal{D}}_{1,\Delta^i}(x), \textit{else} \\ \vdots & \\ F^{\mathcal{D}}_{k-1,\zeta^i}(x) & \textit{if } \neg\Delta^i(x) \wedge I^{\mathcal{D}}_{k-1,\Delta^i}(x), \textit{else} \\ F^{\mathcal{D}}_{k,\zeta^i}(x) & \textit{if } \neg\Delta^i(x), \textit{else} \\ \zeta^i(x). & \end{cases}$$

Since $\Delta^{i+1}(x)$ *only differs from* $\Delta^i(x)$ *in case the latter is* `false`*, one has that* $\Delta^j = \Delta^{j+1}$ *for some* $j < |A|^r$*. In this case, we also have that* $\zeta^j = \zeta^{j+1}$*. We denote these fixed points by* D^∞ *and* Z^∞ *respectively and call them the* fixed points of $H(D, t_1, \ldots, t_r)$, the $F_i(Z, t_1, \ldots, t_r)$, and the $I_i(D, t_1, \ldots, t_r)$ on $\mathcal{D}$. *The* fixed point rule *is now stated as follows. If* $F_i(Z, t_1, \ldots, t_r)$, $1 \leq i \leq k$ *are number terms as previously, and* $H(D, t_1, \ldots, t_r)$, $I_i(D, t_1, \ldots, t_r)$, $1 \leq i \leq k$ *are first-order formulas as previously, then*

$$\mathbf{fp}[D(t_1, \ldots, t_r) \leftarrow H(D, t_1, \ldots, t_r)](u_1, \ldots, u_r)$$

is a first-order formula of signature (L_s, L_f)*, and*

$$\mathbf{fp}[Z(t_1, \ldots, t_r) \leftarrow F_i(Z, t_1, \ldots, t_r), I_i(D, t_1, \ldots, t_r), H(D, t_1, \ldots, t_r)](u_1, \ldots, u_r)$$

is a number term of signature (L_s, L_f)*. Their interpretations on a given* $\mathcal{K}$*-structure* $\mathcal{D}$ *are* D^∞ *and* Z^∞*, respectively.*

Remark 3. In [GM95,CM99,BCSS98], the fixed point rule allows to define only number terms. Thus, the authors need to introduce another rule, the maximization rule, which allows them to compute characteristic functions of relations of L_s as number terms, and perform logical operations such as AND, NOT, OR, by the appropriate use of multiplication, addition and a sign function. This approach, however, is not appropriate to our setting of an arbitrary structure, where no specific requirement is made on the functions and operations of the computational structure. We introduce therefore the possibility to define relations as fixed point, such as D^∞ in the definition above, which enables us to perform legally all kinds of logical operations in first-order logic. While our definition of the fixed point rule is more complicated, it makes the need for a maximization rule obsolete and enables us to prove our results over arbitrary structures, which can be specialized to be the real numbers with addition and order $\mathbb{R}_{ovs}$, or the complex numbers. Moreover, it allows us to capture also classes decided by machines having no constant node, by an appropriate specialization of the computational structure.

The fixed point rule allows us to defined an extension of the first-order logic, as follows.

Definition 8. (Fixed Point First-Order Logic for $\mathcal{K}$-Structures): *Fixed point first-order logic for $\mathcal{K}$-structures, denoted by $\mathrm{FP}_{\mathcal{K}}$, is obtained by augmenting first-order logic $\mathrm{FO}_{\mathcal{K}}$ with the fixed point rule.*

3.2 Characterizing $\mathrm{P}_{\mathcal{K}}$

It is clear that, for any signature $\sigma = (L_s, L_f)$, one can encode in polynomial time finite ordered $\mathcal{K}$-structures of signature σ as elements of $\mathbb{K}^*$ of polynomial size. For $\mathcal{D}$ an ordered $\mathcal{K}$-structure of signature σ, we will denote by $e(\mathcal{D})$ its encoding. Details can be found in the Appendix.

Definition 9. (Decision Problem of Ordered $\mathcal{K}$-structures): *Let $\sigma = (L_s, L_f)$ be a signature containing a binary relation symbol $\leq$. We denote by $Struct(\sigma)$ the set of ordered $\mathcal{K}$-structures of signature σ. A decision problem of ordered $\mathcal{K}$-structures of signature σ is a subset of $Struct(\sigma)$. Modulo a polynomial encoding of ordered $\mathcal{K}$-structures, any decision problem of ordered $\mathcal{K}$-structures can be seen as a decision problem over $\mathcal{K}$ in the BSS model, and vice versa.*

This yields the following characterization of $\mathrm{P}_{\mathcal{K}}$.

Theorem 1. *Let S be a decision problem of ordered $\mathcal{K}$-structures. Then the following statements are equivalent.*

(i) $S \in \mathrm{P}_{\mathcal{K}}$.
(ii) *There exists a sentence ψ in fixed point first-order logic such that* $S = \{\mathcal{D} | \mathcal{D} \models \psi\}$.

Example 2. Recall the signature $(L_s, L_f) = (\emptyset, \{f_0, f_1, f_2, f_3, f_4\})$ given in Example 1 for encoding inputs to the 4FEAS problem. Consider $X : A \to \mathcal{K}$ whose interpretation in a $\mathcal{K}$-structure $\mathcal{D}$ of signature $(L_s, L_f \cup X)$ instantiates the variables of the polynomial. The problem 4EVAL of checking that a given polynomial of degree 4 evaluates to 0 at a given point is in $\mathrm{P}_{\mathcal{K}}$. It can be formulated as follows: There exists a sentence ϕ in fixed point first-order logic over the signature $(L_s, L_f \cup X)$ such that

$$\text{4EVAL} = \{\mathcal{D} | \mathcal{D} \models \phi\}.$$

4 Second-Order Logic on $\mathcal{K}$-structures

4.1 Definitions

Recall that $\mathcal{K}$-structures can be naturally seen as points in $\mathcal{K}^*$. A decision problem of ranked $\mathcal{K}$-structures is then a decision problem, where the (positive) inputs are encoded as $\mathcal{K}$-structures, and satisfy the natural property of being ranked.

Definition 10. (Existential Second-Order Logic For $\mathcal{K}$-Structures): *We say that ψ is an* existential second-order sentence *(of signature $\sigma = (L_s, L_f)$) if $\psi = \exists Y_1, \ldots, \exists Y_r \phi$, where ϕ is a first-order sentence in $\mathrm{FO}_{\mathcal{K}}$ of signature $(L_s, L_f \cup \{Y_1, \ldots, Y_r\})$. The function symbols $Y_1, \ldots, Y_r$ are called* function variables. *Existential second-order sentences are interpreted in $\mathcal{K}$-structures $\mathcal{D}$ of signature σ in the following way: a sentence $\mathcal{D} \models (\psi = \exists Y_1, \ldots, \exists Y_r \phi)$ if there exist functions $X_1, \ldots, X_r : A^{r_i} \to \mathbb{K}$ with r_i the arity of Y_i, such that the interpretation of ψ taking $Y_i^{\mathcal{D}}$ to be X_i yields* `true`. *The set of second-order sentences together with this interpretation constitutes* existential second-order logic *and is denoted by $\exists\mathrm{SO}_{\mathcal{K}}$.*

Existential second-order logic has at least the expressive power of fixed point first-order logic. Indeed:

Proposition 1. *For every sentence ψ of signature σ in $\mathrm{FP}_{\mathcal{K}}$ there exists a sentence $\widetilde{\psi}$ of signature σ in $\exists\mathrm{SO}_{\mathcal{K}}$ such that, for every $\mathcal{K}$-structure $\mathcal{D}$,*

$$\mathcal{D} \models \psi \text{ if and only if } \mathcal{D} \models \widetilde{\psi}.$$

4.2 Characterizing $\mathbf{NP}_{\mathcal{K}}$

Theorem 2. *Let S be a decision problem of ordered $\mathcal{K}$-structures. Then the following statements are equivalent.*

(i) $S \in \mathrm{NP}_{\mathcal{K}}$.
(ii) *There exists an existential second-order sentence ψ such that $S = \{\mathcal{D} | \mathcal{D} \models \psi\}$.*

Example 3. Recall the signature $(L_s, L_f) = (\emptyset, \{f_0, f_1, f_2, f_3, f_4\})$ given in Example 1 for encoding inputs to the 4FEAS problem. Consider $X : A \to \mathcal{K}$ whose interpretation in a $\mathcal{K}$-structure $\mathcal{D}$ instantiates the variables of the polynomial. Consider a formula ϕ in fixed point first-order logic over the signature $(L_s, L_f \cup X)$, whose interpretation for a $\mathcal{K}$-structure $\mathcal{D}$ checks whether the evaluation of the polynomial at the point $X^{\mathcal{D}} \in \mathcal{K}^{|A|}$ is 0. A instance of the result above is

$$\mathrm{4FEAS} = \{\mathcal{D} | \mathcal{D} \models \exists X \phi\}.$$

Note that by Proposition 1 ϕ can be expressed in existential second-order logic.

References

[ASV90] S. Abiteboul, S. Simon, and V. Vianu. Non-deterministic languages to express deterministic transformations. In *Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1990.

[AV89] S. Abiteboul and V. Vianu. Fixed point extensions of first-order logic and datalog-like languages. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*, pages 71–79, Washington, D.C., 1989. IEEE Computer Society Press.

[AV91] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43:62–124, 1991.

[BCSS98] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[Bon88] A. J. Bonner. Hypothetical datalog: complexity and expressibility. In *Proceedings of the International Conference on Database Theory*, volume 326 of *LNCS*, pages 144–160, Berlin, 1988.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the Amer. Math. Soc.*, 21:1–46, 1989.

[CM99] F. Cucker and K. Meer. Logics which capture complexity classes over the reals. *J. of Symb. Logic*, (64):363–390, 1999.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1995.

[Fag74] R. Fagin. Generalized first order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computation*, pages 43–73. SIAM-AMS, 1974.

[GG98] E. Grädel and Y. Gurevich. Metafinite model theory. *Information and Computation*, 140(1):26–81, 1998.

[GM95] E. Grädel and K. Meer. Descriptive complexity over the real numbers. In *Proceedings of the 27th Annual ACM Symp. on the Theory of Computing*, pages 315–324, 1995.

[Goo94] J. B. Goode. Accessible telephone directories. *Journal for Symbolic Logic*, 59(1):92–105, 1994.

[GS86] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

[Gur83] Y. Gurevich. Algebras of feasible functions. In *Twenty Fourth Symposium on Foundations of Computer Science*, pages 210–214. IEEE Computer Society Press, 1983.

[Imm83] N. Immerman. languages which capture complexity classes. In *ACM Symposium on Theory of Computing*, pages 347–354, 1983.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16:760–778, 1987.

[Imm91] N. Immerman. Dspace$[n^k] = \mathrm{var}[k+1]$. In *Proc. of the 6th IEEE Symp. on Structure in Complexity Theory*, pages 334–340, 1991.

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

[Lei90] D. Leivant. Inductive definitions over finite structures. *Information and Computation*, 89:95–108, 1990.

[Mos83] Y. Moschovakis. Abstract recursion as the foundation for a theory of algorithms. In *Computation and Proof Theory (Proc. of the ASL Meeting in Aachen*, Lecture notes in Mathematics. Springer Verlag, Berlin, 1983.

[Poi95] B. Poizat. *Les Petits Cailloux*. Aléas, 1995.

[Tur36] A. M. Turing. On computable numbers, with an application to the *entscheidungsproblem*. In *Proc. London Math. Society*, volume 42 of *2*, pages 230–265, 1936.

[Var82] M. Vardi. Complexity of relational query languages. In *ACM Symposium on Theory of Computing*, pages 137–146, 1982.

Appendix

Interpretation of First-Order Logic for $\mathcal{K}$-Structures

Let $\sigma = (L_s, L_f)$ be a signature, and $\mathcal{D}$ be a $\mathcal{K}$-structure of signature σ. For any index term $h(x_1, \ldots, x_n)$ containing $n \geq 0$ variables, and any point $a \in A^n$, the interpretation of the function symbols of h in $\mathcal{D}$ extend to a natural interpretation $h^{\mathcal{D}}(a) \in A$ of h in $\mathcal{D}$ at the point a, where x_i is interpreted as a_i for $i = 1, \ldots, n$. The same holds with number terms $t(x_1, \ldots, x_n)$, the interpretation $t^{\mathcal{D}}(a)$ at a lying in $\mathbb{K}$. The interpretation of the relation symbols of L_s in $\mathcal{D}$ and the relations of $\mathcal{K}$ enable us to associate *truth values* with atomic formulas evaluated at point in A^n. Thus, if $h_1, \ldots, h_k$ are index terms and $t_1, \ldots, t_l$ are number terms containing the variables $x_1, \ldots, x_n$, and $a \in A^n$, we say that $r(h_1, \ldots, h_k)$ is `true` in $\mathcal{D}$ if $(h_1^{\mathcal{D}}, \ldots, h_k^{\mathcal{D}}) \in r^{\mathcal{D}}$ and `false` otherwise, and, for a l-ary relation rel of $\mathcal{K}$, $rel(t_1, \ldots, t_l)$ is `true` if $(t_1^{\mathcal{D}}, \ldots, t_k^{\mathcal{D}}) \in rel$ in $\mathcal{K}$, and `false` otherwise. The interpretation of the logical connectives $\vee$, $\wedge$ and $\neg$ is the usual. A formula $(\exists x)\psi$ where ψ contains variables $x, y_1, \ldots, y_n$, $n \geq 0$ is `true` at $b \in A^n$ if there exists $a \in A$ such that $\psi^{\mathcal{D}}$ is `true`, where x is interpreted as a and y_i is interpreted as b_i, for $i = 1, \ldots, n$. Similarly for a formula $(\forall x)\psi$ being interpreted as $\neg(\exists x)\neg\psi$. Proceeding recursively, we can associate with any formula ϕ with free variables $x_1, \ldots, x_n$ over σ and any $a \in A^n$ a truth value. If this value is `true` we say that $\mathcal{D}$ satisfies $\phi(a)$, and we denote this by $\mathcal{D} \models \phi(a)$. A formula with no variable is called a *sentence*. If T is a set of sentences over σ, $\mathcal{D}$ satisfies T if and only if $\mathcal{D}$ satisfies all the sentences of T. This is denoted as $\mathcal{D} \models T$. The $\mathcal{K}$-structures $\mathcal{D}$ such that $\mathcal{D} \models T$ are called the *models* of T. The sentences of T are called *axioms* and T is a *theory*.

Encoding of Ordered $\mathcal{K}$-structures

Let $\sigma = (L_s, L_f)$ be a signature containing a binary relation symbol $\leq$, and let $\mathcal{D}$ be an ordered $\mathcal{K}$-structure of signature σ. Next, replace all relations in the skeleton by the appropriate characteristic functions $\chi : A^k \to \{\mathbf{0}, \mathbf{1}\} \subseteq \mathbb{K}$. Thus, we get a structure with skeleton a set A and functions $X_1, \ldots, X_t$ of the form $X_i : A^{k_i} \to \mathbb{K}$. Each of these functions X_i can be represented by a tuple $\xi = (x_0, \ldots, x_{m_i - 1}) \in \mathbb{K}^{m_i}$ with $m_i = |A|^{k_i}$ and $x_j = X_i(\overline{a}(j))$ where $\overline{a}(j)$ is the jth tuple in A^{k_i} with respect to the lexicographic order on A^{k_i} induced by $\leq$. The concatenation

$$e(\mathcal{D}) = \xi_1.\xi_2.\ldots.\xi_t$$

of these tuples gives the encoding $e(\mathcal{D}) \in \mathbb{K}^{m_1 + \ldots + m_t}$.

Clearly, for a fixed finite signature and for any finite $\mathcal{K}$-structure $\mathcal{D}$ of size n, the size of $e(\mathcal{D})$ is bounded by some polynomial n^l where l depends only on the signature. Thus, appending zeros to $e(\mathcal{D})$ if necessary, we can also view $e(\mathcal{D}) = (x_0, \ldots, x_{l-1})$ as a single function $X_{\mathcal{D}} : A^l \to \mathbb{K}$. This means that one can encode an ordered $\mathcal{K}$-structure by a single function from the ordered set $\{0, \ldots, n^l - 1\}$ into $\mathbb{K}$. Moreover, this encoding can be performed in polynomial time.

Proof of Theorem 1

Let us first prove the (**i**) $\Rightarrow$ (**ii**) direction. Assume that the signature is $\sigma = (L_s, L_f)$, and that it contains a binary relation symbol $\leq$. Let $S \in \mathrm{P}_{\mathcal{K}}$, and let M be a polynomial time BSS machine over $\mathcal{K}$ deciding S. Assume that, for any ordered $\mathcal{K}$-structure $\mathcal{D}$ of size n, encoded in $\mathbb{K}^*$ by $e(\mathcal{D})$, the computation time of M on $e(\mathcal{D})$ is bounded by n^m, for some $m \in \mathbb{N}$. Assume also that the size of $e(\mathcal{D})$ is n^m. This is without loss of generality since it suffices to add some padding $\mathbf{0}$'s to the encoding $e(\mathcal{D})$ of $\mathcal{D}$. Assume also without loss of generality that the state space of M is bounded by n^m. A point in this space has coordinates $(x_0, x_1, \ldots, x_{n^m-1})$. In the following, we will use the formalism of a Turing machine, with a scanning head moving in $\mathbb{K}^m$ instead of a state space $\mathbb{K}_*$.

Consider the lexicographic order $\leq_m$ on A^m induced by the order $\leq$ on A. By Remark 2, $\leq_m$ is defined in first-order logic, and induces a bijection between A^m and the set $\{0, \ldots, n^m - 1\}$. Therefore, for any $t \in A^m$ we will define:

$$t - 1 = \max_{s \in A^m} \{s <_m t\}$$
$$t + 1 = \min_{s \in A^m} \{t <_m s\}.$$

It is clear that these elements are definable in first-order logic over ordered $\mathcal{K}$-structures, thus we will freely use these notations hereafter. Note however that, when t is minimal in A^m, we have $t-1 = t$. A similar remark holds for t maximal with $t+1 = t$. Following this notation, we will identify A^m with $\{0, \ldots, n^m - 1\}$.

Next, we assume that the number of nodes of M is bounded by 2^k, $k \in \mathbb{N}$. Note that k is fixed, independent of $\mathcal{D}$ or n. Thus, the nodes of M will be denoted as elements in A^k, with the proviso that $|A| \geq 2$. We also assume that they correspond to the first elements of A^m with respect to the lexicographic order $\leq_m$. Since their number is constant, they are all definable in first-order logic. We will therefore denote them with constants $v_{type} \in A^m$, where *type* denotes the type of the node as in Definition 2. Recall also the constant $0 \in A^m$, corresponding the the minimal element of A^m as in Remark 2.

Consider now the following relation symbol, not contained in L_s, of arity $4m$:

$$\texttt{Node}(v, t, pos, c) = \texttt{true} \text{ iff}$$
$$\text{the current node of } M \text{ at step } t \text{ is } v \text{ and}$$
$$\begin{cases} pos = 0 & \text{and the head of } M \text{ is on cell } c \\ pos = 1 & \text{and the head of } M \text{ is not on cell } c. \end{cases}$$

Consider also the following function symbol, not contained in L_f, of arity $2m$:

$$\texttt{Cell}(t, c) = \text{content of cell } c \text{ at step } t.$$

If `Cell` can be defined in $\mathrm{FP}_{\mathcal{K}}$ by a number term Z, then the implication (**i**) $\Rightarrow$ (**ii**) holds. The $\mathcal{K}$-structure $\mathcal{D}$ is accepted by M if and only if after n^m steps, the content of the first cell is $\mathbf{1}$. That is, $\texttt{Cell}(n^m - 1, 0) = \mathbf{1}$, were $n^m - 1$ is the

maximal element in A^m, as defined in Remark 2. `Node` and `Cell` can be defined inductively as follows:

$$
\begin{aligned}
\texttt{Node}(input, 0, pos, c) &\leftarrow [(pos = 0) \vee (c = 0)] \wedge [(pos = 1) \vee (c \neq 0)] \\
\texttt{Node}(v, 0, pos, c) &\leftarrow \texttt{false} \text{ if } v \neq input \\
\texttt{Node}(v, t+1, pos, c) &\leftarrow \texttt{Node}(v_{op_i}, t, pos, c) \text{ for } v_{op_i} \text{ s.t. } v = \beta(v_{op_i}) \\
&\quad \vee\ \texttt{Node}(v_{shift_l}, t, pos, c+1) \text{ for } v_{sh_l} \text{ s.t. } v = \beta(v_{shift_l}) \\
&\quad \vee\ \texttt{Node}(v_{shift_r}, t, pos, c-1) \text{ for } v_{sh_r} \text{ s.t. } v = \beta(v_{shift_r}) \\
&\quad \vee\ \texttt{Node}(v_{copy_l}, t, pos, c) \text{ for } v_{copy_l} \text{ s.t. } v = \beta(v_{copy_l}) \\
&\quad \vee\ \texttt{Node}(v_{copy_r}, t, pos, c) \text{ for } v_{copy_r} \text{ s.t. } v = \beta(v_{copy_r}) \\
&\quad \vee\ \texttt{Node}(v_{switch}, t, pos, c) \text{ for } v_{switch} \text{ s.t. } v = \beta(v_{switch}) \\
&\quad \vee\ \texttt{Node}(v_{output}, t, pos, c) \text{ for } v_{output} \text{ s.t. } v = \beta(v_{output}) \\
&\quad \vee\ \texttt{Node}(v_{rel_i}, t, 0, c) \wedge rel_i(\texttt{Cell}(t, c), \ldots, \texttt{Cell}(t, c+k_i)) \\
&\qquad \wedge pos = 0 \quad \text{for } v_{rel_i} \text{ s.t. } v = \beta^+(v_{rel_i}) \\
&\quad \vee\ \texttt{Node}(v_{rel_i}, t, 0, c) \wedge \neg rel_i(\texttt{Cell}(t, c), \ldots, \texttt{Cell}(t, c+k_i)) \\
&\qquad \wedge pos = 0 \quad \text{for } v_{rel_i} \text{ s.t. } v = \beta^-(v_{rel_i})
\end{aligned}
$$

and:

$$
\begin{aligned}
\texttt{Cell}(0, c) &\leftarrow X^{\mathcal{D}}(c) \\
\texttt{Cell}(t+1, c) &\leftarrow \begin{cases}
op_i(\texttt{Cell}(t, c), \ldots, \texttt{Cell}(t, c+k_i)) & \text{if } \texttt{Node}(v_{op_i}, t, 0, c) \\
\texttt{Cell}(t, c+1) & \text{if } \texttt{Node}(v_{copy_l}, t, 0, c) \\
\texttt{Cell}(t, c-1) & \text{if } \texttt{Node}(v_{copy_r}, t, 0, c) \\
\texttt{Cell}(t, c+1) & \text{if } \texttt{Node}(v_{switch}, t, 0, c) \\
\texttt{Cell}(t, c-1) & \text{if } \texttt{Node}(v_{switch}, t, 0, c-1) \\
\texttt{Cell}(t, c) & \text{otherwise}
\end{cases}
\end{aligned}
$$

This proves that `Cell` can be defined in $\mathrm{FP}_{\mathcal{K}}$ as the fixed point of the inductive definition above, from which (**i**) $\Rightarrow$ (**ii**) follows.

To prove (**ii**) $\Rightarrow$ (**i**), we only need to prove that formulas and number terms defined with the fixed point rule can be evaluated in polynomial time. Since the number of updates is polynomially bounded, one only needs to apply the inductive definition a polynomial number of times, which ends the proof. □

Proof of Proposition 1

It suffices to prove that the fixed point rule of Definition 7 can be expressed within existential second-order logic over $\mathcal{K}$-structures. Assume F_i, I_i, H, $1 \leq i \leq k$, Z and D are as in Definition 7. Every occurrence of $\mathbf{fp}[Z(t_1, \ldots, t_r) \leftarrow F_i(Z, t_1, \ldots, t_r), I_i(D, t_1, \ldots, t_r), H(D, t_1, \ldots, t_r)](u_1, \ldots, u_r)$ and of $\mathbf{fp}[D(t_1, \ldots, t_r) \leftarrow H(D, t_1, \ldots, t_r)](u_1, \ldots, u_r)$ in ψ will be replaced by $Z(u_1, \ldots, u_r)$ and $D(u_1, \ldots, u_r)$, respectively. Denote by ϕ the resulting formula.

For any r-ary function $F : A^r \to \mathbb{K}$, denote by $\widetilde{F}$ the $r - ary$ formula with variables $(u_1, \ldots, u_r)$ defined by $\widetilde{F}(u_1, \ldots, u_r) = (F(u_1, \ldots, u_r) = \mathbf{1})$.

Then, $\widetilde{\psi}$ is

$$
\begin{aligned}
& \exists Z \exists D \forall (u_1, \ldots, u_r) \\
& \widetilde{D}(u_1, \ldots, u_r) \Leftrightarrow H(\widetilde{D}, u_1, \ldots, u_r) \\
\wedge\ & F_1(Z, (u_1, \ldots, u_r)) = Z(u_1, \ldots, u_r) \Leftrightarrow I_1(\widetilde{D}, u_1, \ldots, u_r) \\
& \vee F_2(Z, (u_1, \ldots, u_r)) = Z(u_1, \ldots, u_r) \Leftrightarrow \begin{cases} \neg(I_1(\widetilde{D}, u_1, \ldots, u_r)) \\ \wedge \quad I_2(\widetilde{D}, u_1, \ldots, u_r) \end{cases} \\
& \vdots \\
& \vee F_{k-1}(Z, (u_1, \ldots, u_r)) = Z(u_1, \ldots, u_r) \Leftrightarrow \begin{cases} \neg(I_1(\widetilde{D}, u_1, \ldots, u_r)) \\ \vdots \\ \wedge\ \neg(I_{k-2}(\widetilde{D}, u_1, \ldots, u_r)) \\ \wedge \quad I_{k-1}(\widetilde{D}, u_1, \ldots, u_r) \end{cases} \\
& \vee F_k(Z, (u_1, \ldots, u_r)) = Z(u_1, \ldots, u_r) \Leftrightarrow \begin{cases} \neg(I_1(\widetilde{D}, u_1, \ldots, u_r)) \\ \vdots \\ \wedge\ \neg(I_{k-2}(\widetilde{D}, u_1, \ldots, u_r)) \\ \wedge\ \neg(I_{k-1}(\widetilde{D}, u_1, \ldots, u_r)) \end{cases} \\
\wedge\ & \phi.
\end{aligned}
$$

□

Proof of Theorem 2

Let $S \in \mathrm{NP}_{\mathcal{K}}$ be a problem of ordered $\mathcal{K}$-structures of signature $\sigma = (L_s, L_f)$. By definition of $\mathrm{NP}_{\mathcal{K}}$, there exists $r \in \mathbb{N}$, a function symbol Y of arity r not in L_f, and a decision problem H of ordered $\mathcal{K}$-structures of signature $(L_s, L_f \cup Y)$, such that H belongs to $\mathrm{P}_{\mathcal{K}}$ and

$$S = \{\mathcal{D} \in Struct(\sigma) | \exists Y\ (\mathcal{D}, Y) \in H\}.$$

By Theorem 1, there exists a fixed point first-order formula ϕ that describes H, and thus,

$$\mathcal{D} \in S \text{ if and only if } \mathcal{D} \models \exists Y \phi.$$

By Proposition 1, ϕ can be replaced by an equivalent second-order formula $\exists Z \varphi$ with φ a first-order formula. Then,

$$\mathcal{D} \in S \text{ if and only if } \mathcal{D} \models \exists Y \exists Z \varphi,$$

which shows (**i**) $\Rightarrow$ (**ii**). For the other direction, it suffices to guess an interpretation for the second-order quantified functions, and to check in $\mathrm{P}_{\mathcal{K}}$ that the first-order formula induced is satisfied. □