Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 332 (2017) 113–130

www.elsevier.com/locate/entcs

The Descriptive Complexity of Decision Problems through Logics with Relational Fixed-Point and Capturing Results [1]

Márcia Farias[2], Ana Teresa Martins[3]

Department of Computation
Federal University of Ceará
P.O. Box 12166, Fortaleza, CE, Brazil

Francicleber Ferreira[4]

Federal University of Ceará, Campus de Quixadá
63902-580, Quixadá, CE, Brazil

Abstract

In this work, we generalize the classical fixed-point logics using relations instead of operators in order to capture the notion of nondeterminism. The basic idea is that we use loops in a relation instead of fixed-points of a function, that is, X is a fixed-point of the relation $\mathcal{R}$ in case the pair (X, X) belongs to $\mathcal{R}$. We introduce the notion of initial fixed-point of an inflationary relation $\mathcal{R}$ and the associated operator **rifp**. We denote by RIFP the first-order logic with the inflationary relational fixed-point operator **rifp** and show that it captures the polynomial hierarchy using a translation to second-order logic. We also consider the fragment RIFP_1 with the restriction that the **rifp** operator can be applied at most once. We show that RIFP_1 captures the class NP and compare our logic with the nondeterministic fixed-point logic proposed by Abiteboul, Vianu and Vardi in [4].

Keywords: descriptive complexity, fixed-point logic, relational fixed-point logic, expressiveness.

1 Introduction

The characterization of the complexity classes through logics is the central theme of Descriptive Complexity [17]. Such characterizations establish that sentences in some logic define decision problems on finite structures that belong to a specific

[1] This work was partially supported by CAPES(DS), CNPq (305980/2013-0, 474821/2012-9), CAPES/CNPq (552578/2011-8), CAPES(PROCAD/NF 789/2010) and FUNCAP.

[2] Email: roberta@lia.ufc.br

[3] Email: ana@lia.ufc.br

[4] Email: fran@lia.ufc.br

http://dx.doi.org/10.1016/j.entcs.2017.04.008
1571-0661/© 2017 The Author(s). Published by Elsevier B.V.

complexity class and vice versa, that is, problems in this complexity class can be defined by sentences of the language. This kind of result builds bridges between Finite Model Theory [9] and Complexity Theory [20] and opens up the possibility to use methods and results from one area to solve problems of the other. It also enables relationships between logical resources (e.g., quantification, number of variables, higher-order quantification, fixed-point operators, etc.) and computational resources (e.g., number of processors, nondeterminism, alternation, time, space, etc.).

The relational model of databases introduced by Codd [8] made explicit the close relation between database theory and finite model theory. Databases can be seen as finite structures and query languages as logical languages [1]. Interest in logical languages has grown rapidly, since such languages, with the associated semantics, provides a strong theoretical background for database theory, especially on query processing, and a large amount of techniques from model theory can be applied. On its turn, this interest on the logical approach to databases promoted an intensive development of Finite Model Theory.

An obvious candidate for a query language is the first-order logic (FO). First-order logic was developed in the last century as a result of the efforts to provide a logical foundation to Mathematics. In this setting, infinite structures play an important role and, indeed, the methods of the model theory of first-order logic rely on the fact that structures can be infinite [7,15]. Although FO is very well suited for mathematical reasoning, its expressive power with respect to finite models is rather limited. Many problems, such as deciding whether a graph is connected or not, cannot be expressed in FO, even when restricted to finite structures. Such limitations derive from the fact that FO is not able to express inductive definitions.

This lack of expressiveness was fulfilled with the introduction of fixed-point operators to improve the expressive power of FO. Aho and Ullman [2] introduced the idea of extending the relational algebra, which is basically first-order logic in algebraic fashion, with fixed-point operators. The idea was followed by Chandra and Harel [6], giving rise to an extension of first-order logic with the least fixed-point operator, known as the least fixed-point logic LFP.

An operator is a function $F : \mathcal{P}(A) \to \mathcal{P}(A)$. When F is a monotone operator (that is, $X \subseteq Y \Rightarrow F(X) \subseteq F(Y)$), it has a least fixed-point $\mathbf{lfp}(F)$. The least fixed-point logic extends the language of first-order logic with expressions that define the least fixed-point of suitably chosen monotone operators (see the precise definition in Section 2). The least fixed-point logic is strictly more expressive than FO. For instance, graph properties like connectivity and reachability can be expressed by LFP sentences. Other fixed-point logics can be defined by changing the type of operators considered and the type of fixed-point chosen.

Given a sentence of some logic, the set of its finite models defines a decision problem. Once we use sentences in some language to express decision problems, the immediate question is: What is the complexity of such problems? This question has been answered for several logics, see [17]. First-order logic, for example, corresponds to the logarithmic-time hierarchy. One of the first results relating logical expressive power and computational complexity is the celebrated Fagin's Theorem [11]. According to Fagin's Theorem, the existential fragment of second-order logic defines exactly the problems in NP (nondeterministic polynomial time, see [20]), and it was used by Stockmeyer [21] to show that second-order logic corresponds to the polynomial hierarchy. An earlier result, due to Büchi [5], shows that the monadic second-order logic on strings defines exactly the class of regular languages. LFP has an intermediate expressive power between first-order and existential second-order logic. On the class of ordered structures, LFP corresponds to the decision problems in the class P of problems that can be solved in polynomial time [17].

Most fixed-point logics are defined in the way sketched above for the case of LFP. In particular, they use operators (functions) and their fixed-points to define inductive sets. We are interested in extending this notion considering not only operators but relations. Instead of a function $F : \mathcal{P}(A) \to \mathcal{P}(A)$ and its fixed-points, we consider a relation $\mathcal{R} \subseteq \mathcal{P}(A)\times\mathcal{P}(A)$ and its loops, that is, X such that $(X, X) \in \mathcal{R}$. Our motivation is to try to cope with nondeterminism using relations, inspired by the use of functions in fixed-point logics to formalize determinism. This generalization opens up new possibilities for the definition of fixed-point of relations and the corresponding logics. As in the case of operators used in the traditional fixed-point logics, suitable conditions on the kind of relations and fixed-points considered can be used to define several logics in different ways.

We will focus on a specific type of relation which we call inflationary (see Section 3) and can be seen as the relational counterpart of inflationary fixed-point logic [9,13]. In Section 2, we recall the fundamentals of fixed-point theory and logics. In Section 3, we define some important concepts about relational fixed-point theory. In Section 4, we present the relational inflationary fixed-point logic RIFP with its syntax and semantics, and we also show examples of how to express problems in this logic. In Section 5, we demonstrate that the RIFP logic captures the polynomial hierarchy complexity class through a translation of RIFP to second-order logic, and vice versa. In Section 6, we define a fragment of RIFP logic called RIFP_1 which contains only one application of **rifp** operator, and we show that this logic captures NP. In Section 7, we compare RIFP with NIFP, a logic defined by Abiteboul, Vianu and Vardi [4]. Finally, we conclude this work and we show future topics of research in Section 8.

2 Fixed-Point Logics and Descriptive Complexity

In this section, we review the basics concepts of first-order logic and descriptive complexity that we will use in this paper.

We follow the notation from [10,9]. A symbol-set is a set $S = \{R_1, \dots, R_l, f_1, \dots, f_k, c_1, \dots, c_m\}$ of relation, function and constant symbols. Each R_i is a relation symbol of arity a_i and f_j is a function symbol of arity r_j. An S-structure is a pair $\mathfrak{A} = (A, \sigma)$ where A is a set and σ is a map that associates each relation symbol $R \in S$ of arity k to a relation $\sigma(R) = R^{\mathfrak{A}} \subseteq A^k$, each function symbol f to a function $\sigma(f) = f^{\mathfrak{A}}$, and each constant $c \in S$ to an element $\sigma(c) = c^{\mathfrak{A}}$.

We define $STRUCT[S]$ to be the set of finite S-structures. An ordered structure is a structure in a symbol set S containing a relation symbol $<$ which is interpreted as a linear order.

An S-interpretation $\mathfrak{I}$ is a pair $(\mathfrak{A}, \beta)$ consisting of an S-structure and an assignment β which associates to each first-order variable x an element $\beta(x) \in A$, and to each relation variable X of arity k a relation $\beta(X) \subseteq A^k$.

The languages and semantics of first-order and second-order logics are defined as usual. A formula whose symbols belong to some symbol-set S is called an S-formula. We call formulas with free relation variables, but without second-order quantification as first-order formulas, since they are first-order in essence. If x is a first-order variable and $a \in A$, then $\beta\frac{a}{x}$ is the assignment defined as $\beta\frac{a}{x}(y) = a$ if $y = x$ and $\beta\frac{a}{x}(y) = \beta(y)$ otherwise. Analogously, if X is a second-order variable of arity k and $\mathbf{X} \subseteq A^k$, then $\beta\frac{\mathbf{X}}{X}(Y) = \mathbf{X}$ if $Y = X$ and $\beta\frac{\mathbf{X}}{X}(Y) = \beta(Y)$ otherwise.

Inductive sets can be defined as fixed-points of certain operators. Let A be a set and $F : \mathcal{P}(A) \to \mathcal{P}(A)$ and operator on A. The sequence of stages of F is the sequence

$$F_0(= \emptyset), F_1, \ldots$$

where $F_{i+1} = F(F_i)$ and F_i is called the i-th stage of F. An operator F is called *inductive* if its stages form an increasing chain ($F_i \subseteq F_{i+1}$). F is said *inflationary* if $X \subseteq F(X)$, for all $X \in \mathcal{P}(A)$, and it is *monotone* if $X \subseteq Y$ implies $F(X) \subseteq F(Y)$, for all $X, Y \in \mathcal{P}(A)$. We denote by F_∞ the stage F_i which is a fixed-point of F, if it exists, and $\emptyset$ otherwise. The sequence of stages of inductive (and of inflationary and monotone, since they are inductive) operators reaches a fixed-point at some stage of the sequence. If F is inductive, the *inductive fixed-point* of F is defined as $\mathbf{ind}(F) = F_\infty$. If F is inflationary, the *inflationary fixed-point* of F is defined as $\mathbf{ifp}(F) = F_\infty$. A monotone operator always has a *least fixed-point* (w.r.t. inclusion $\subseteq$) defined as $\mathbf{lfp}(F) = F_\infty$. For F an arbitrary operator, we call F_∞ the *partial fixed-point* of F and denote it by $\mathbf{pfp}(F) = F_\infty$.

Fixed-point logics are created extending the FO language with syntactical constructs that denote fixed-points of operators defined in the language. Let $\phi(X, \overline{x})$ be a formula of some language where X is a relational symbol of arity k and $\overline{x} = x_1, \ldots, x_k$. The formula ϕ defines an operator on the domain of an interpretation $\mathfrak{I}$ as

$$\phi^{\mathfrak{I}}(\mathbf{X}) = \{\overline{a} \in A^k \mid \mathfrak{I}\frac{\mathbf{X}, \overline{a}}{X, \overline{x}} \models \phi(X, \overline{x})\}.$$

A formula is *positive* on a relational variable iff this variable does not occur inside the scope of an odd number of negations (when we consider only the connectives $\wedge$ and $\vee$). If ϕ is positive on X, then $\phi^{\mathfrak{I}}$ is monotone. If ϕ is of the form $\phi = X\overline{x} \vee \phi'(X, \overline{x})$, then $\phi^{\mathfrak{I}}$ is inflationary.

The least fixed-point logic is obtained extending the FO language with the following rule:

- if X is a k-ary relational variable, $\overline{x} = x_1, \ldots, x_k$ and $\phi(X, \overline{x})$ is an LFP-formula positive on X, then $[\mathit{lfp}_{X,\overline{x}}\ \phi(X, \overline{x})](\overline{t})$ is an LFP-formula, where $\overline{t} = t_1, \ldots, t_k$ is a tuple of terms of length k.

Given an interpretation $\mathfrak{I}$, the satisfaction relation for the new formulas is defined as

$$\mathfrak{I} \models [\mathit{lfp}_{X,\overline{x}}\ \phi(X,\overline{x})](\overline{t}) \text{ iff } (t_1^{\mathfrak{I}},\dots,t_k^{\mathfrak{I}}) \in \mathbf{lfp}(\phi^{\mathfrak{I}}).$$

The inflationary fixed-point logic is defined similarly:

- if X is a k-ary relational variable, $\overline{x} = x_1,\dots,x_k$ and $\phi(X,\overline{x})$ is an IFP-formula, then $[\mathit{ifp}_{X,\overline{x}}\ \phi(X,\overline{x}) \vee X\overline{x}](\overline{t})$ is an IFP-formula, where $\overline{t} = t_1,\dots,t_k$ is a tuple of terms of length k.

The satisfaction relation is defined as:

$$\mathfrak{I} \models [\mathit{ifp}_{X,\overline{x}}\ \phi(X,\overline{x})](\overline{t}) \text{ iff } (t_1^{\mathfrak{I}},\dots,t_k^{\mathfrak{I}}) \in \mathbf{ifp}(\phi^{\mathfrak{I}}).$$

Another important fixed-point logic is the partial fixed-point logic:

- if X is a k-ary relational variable, $\overline{x} = x_1,\dots,x_k$ and $\phi(X,\overline{x})$ is a PFP-formula, then $[\mathit{pfp}_{X,\overline{x}}\ \phi(X,\overline{x})](\overline{t})$ is a PFP-formula, where $\overline{t} = t_1,\dots,t_k$ is a tuple of terms of length k.

And the corresponding satisfaction relation:

$$\mathfrak{I} \models [\mathit{pfp}_{X,\overline{x}}\ \phi(X,\overline{x})](\overline{t}) \text{ iff } (t_1^{\mathfrak{I}},\dots,t_k^{\mathfrak{I}}) \in \mathbf{pfp}(\phi^{\mathfrak{I}}).$$

LFP and IFP have the same expressive power. This result was proved for finite structures by Gurevich and Shelah [14]. Later, the restriction on the cardinality of the structures was dropped by Kreutzer [18].

We are interested in the characterization of complexity classes using logics. We can define a logic as a pair $\mathcal{L} = (L, \models)$, where L is a language and $\models$ is its satisfiability relation. An S-sentence from $\mathcal{L}$ defines a decision problem on S-structures. Let ϕ be an S-sentence. We define the class of finite models of ϕ as the class of S-structures that satisfies ϕ:

$$\mathit{Mod}(\phi) = \{\mathfrak{A} \in \mathit{STRUCT}[S] \mid \mathfrak{A} \models \phi\}.$$

The decision problem defined by a class of structures is called a *query*, following the database jargon.

Finite structures can be coded as inputs for a computation model (see [17] for details). Hence, we may ask whether a class of structures is computable or not, and what is the complexity of solving it. Thus, we may consider to which complexity class, if any, the queries defined by some logic belong.

Definition 2.1 ($\mathcal{L}$ captures $\mathcal{C}$) *Let $\mathcal{L}$ be a logic and $\mathcal{C}$ a complexity class. We say that $\mathcal{L}$* captures *$\mathcal{C}$ ($\mathcal{L} = \mathcal{C}$) iff*

- *for each sentence $\phi \in \mathcal{L}$, the class of structures $\mathit{Mod}(\phi)$ is in $\mathcal{C}$ ($\mathcal{L} \subseteq \mathcal{C}$), and,*
- *for each class of structures Q in $\mathcal{C}$ there is a sentence $\phi \in \mathcal{L}$ such that $Q = \mathit{Mod}(\phi)$ ($\mathcal{C} \subseteq \mathcal{L}$).*

Fagin showed that the existential fragment of second-order logic ($\exists$SO) captures the class of queries that can be solved in nondeterministic polynomial time (NP)

[11].

Theorem 2.2 (∃SO = NP) *∃SO captures NP.*

Restricted to ordered structures, LFP, and equivalently IFP, defines exactly the queries in PTIME ([16,22]).

Theorem 2.3 (LFP = PTIME, on ordered structures) *A class of ordered structures can be defined in LFP iff that class is in PTIME.*

The role of order is important here since LFP cannot define all queries in PTIME on unordered structures. Actually, even simple polynomial problems, as deciding whether a structure has a domain of even cardinality, cannot be defined in LFP [19]. On unordered structures, we only have the containment LFP $\subsetneq$ PTIME, but not the converse.

With respect to PFP, we have that PFP captures PSPACE, the class of decision problems that can be solved using polynomial space, on ordered structures:

Theorem 2.4 (PFP = PSPACE, on ordered structures) *A class of ordered structures can be defined in PFP iff that class is in PSPACE [3,22].*

Many other capturing results have been proved for several logics and complexity classes. See [17] for details. We have the following relations among the logics mentioned above:

$$\text{LFP} = \text{IFP} \subseteq \exists\text{SO} \subseteq \text{PFP},$$

where the last inclusion only holds for ordered structures. We want to generalize the fixed-point operators considered in the traditional fixed-point theory in order to explore new ways to define logics aiming at the characterization of complexity classes through logics in the spirit of descriptive complexity. We will introduce the relational approach in the next section.

3 Relational Fixed-Point Theory

As we have seen, the traditional fixed-point theory is based on the concept of the operator, which is essentially a function. Our proposal is to consider relations instead of functions in order to capture nondeterminism.

Given a finite domain A, let $\mathcal{R} \subseteq \mathcal{P}(A^k) \times \mathcal{P}(A^k)$ be a (second-order) relation. We say that a binary relation $\mathcal{R}$ is *total* if for all $X \in \mathcal{P}(A^k)$ there is a $Y \in \mathcal{P}(A^k)$ such that $(X, Y) \in \mathcal{R}$. A *chain* in $\mathcal{R}$ is a sequence $X_0, X_1, \ldots, X_m$ such that $X_0 = \emptyset$, $X_i \subseteq X_{i+1}$ and $(X_i, X_{i+1}) \in \mathcal{R}$.

We can define conditions on relations that resemble those used to define the types of fixed-point operators that we have seen before. We say that a relation $\mathcal{R} \subseteq \mathcal{P}(A^k) \times \mathcal{P}(A^k)$ is an **inflationary relation** if, for all $(X, Y) \in \mathcal{R}$, we have $X \subseteq Y$. Given an arbitrary relation $\mathcal{R} \subseteq \mathcal{P}(A^k) \times \mathcal{P}(A^k)$, we define an inflationary, total relation $\mathcal{R}_{\text{INF}}$ from $\mathcal{R}$ as follows:

$$\mathcal{R}_{\text{INF}} = \{(X, X \cup Y) \in \mathcal{P}(A^k)^2 \mid (X, Y) \in \mathcal{R}\} \cup \{(X, X) \in \mathcal{P}(A^k)^2 \mid \nexists Y : (X, Y) \in \mathcal{R}\}. \tag{1}$$

We say that a relation $\mathcal{R} \subseteq \mathcal{P}(A^k) \times \mathcal{P}(A^k)$ is **inductive** if any sequence $X_0, X_1, \ldots, X_m$, such that $X_0 = \emptyset$ and $(X_i, X_{i+1}) \in \mathcal{R}$, $0 \leq i \leq m-1$, is a chain in $\mathcal{R}$.

Lemma 3.1 *If $\mathcal{R}$ is inflationary then it is inductive.*

A set $X \in \mathcal{P}(A^k)$ is a **fixed-point** of the relation $\mathcal{R}$ if $\mathcal{R}(X, X)$. A fixed-point X of $\mathcal{R}$ is called **inductive** if there is a chain $X_0, X_1, \ldots, X_m$ such that $X_m = X$. We denote by $\text{IndFP}(\mathcal{R})$ the set of inductive fixed-points of $\mathcal{R}$. We say that a fixed-point X is an **initial fixed-point** of $\mathcal{R}$ if there is a chain $X_0, X_1, \ldots, X_m$ such that $X_m = X$ and no X_j, $j < m$, is a fixed-point of $\mathcal{R}$. Denote by $\text{IniFP}(\mathcal{R})$ the set of all initial fixed-points of $\mathcal{R}$.

A set X is an **inflationary fixed-point of** $\mathcal{R}$ if X is an inductive fixed-point of $\mathcal{R}_{\text{INF}}$. We say that X is an **initial inflationary fixed-point of** $\mathcal{R}$ if X is an initial fixed-point of $\mathcal{R}_{\text{INF}}$. We denote by $\mathbf{rifp}(\mathcal{R})$ the set of all initial inflationary fixed-points of $\mathcal{R}$, that is, $\mathbf{rifp}(\mathcal{R}) = \text{IniFP}(\mathcal{R}_{\text{INF}})$.

Lemma 3.2 *Let A be a finite set. If a relation $\mathcal{R} \in \mathcal{P}(A^k) \times \mathcal{P}(A^k)$ is total and inductive, then any chain $X_0 = \emptyset, X_1, \ldots, X_m$ with sufficiently large m contains a fixed-point, where $m \leq |A|^k$.*

It is easy to see that operators are particular cases of relations. Indeed, given an operator $F : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$, its graph is the relation

$$\mathcal{R}_F = \{(X, Y) \in \mathcal{P}(A^k) \times \mathcal{P}(A^k) \mid Y = F(X)\}.$$

Hence, the relational approach can be used to represent the functional one.

We will develop a logic which is able to talk about the initial inflationary fixed-points of a relation and we will see that this logic will capture some nondeterministic complexity classes.

4 The Relational Fixed-Point Logic - RIFP

In this section, we will add to first-order logic the ability to define relations that correspond to the union of initial inflationary fixed-points of defined relations. This extension will be called Relational Inflationary Fixed-Point Logic (RIFP). It will increase the expressive power beyond that of inflationary fixed-point logic.

Let $\phi(X, Y)$ be an S-formula with free relation variables X and Y of arity k, and $\mathfrak{I}$ an S-interpretation. The formula $\phi(X, Y)$ defines the relation

$$\mathcal{R}^{\phi,\mathfrak{I}} = \{(V_1, V_2) \in \mathcal{P}(A^k) \times \mathcal{P}(A^k) \mid \mathfrak{I}\frac{V_1, V_2}{X, Y} \models \phi(X, Y)\}.$$

If $\phi(X, Y)$ has no free variables other than X and Y then, given a structure $\mathfrak{A}$, the relation $\mathcal{R}^{\phi,\mathfrak{I}}$ is the same for all interpretations $\mathfrak{I}$ on $\mathfrak{A}$. Thus, we write $\mathcal{R}^{\phi,\mathfrak{A}}$ instead.

The language of RIFP extends the language of first-order logic with the following rule:

- if X and Y are relation variables of arity k, $\bar{t} = t_1, \ldots, t_k$ are terms and $\phi(X, Y)$ is an S-formula, then

$$[rifp_{X,Y}\phi(X, Y)](\bar{t})$$

is an S-formula of RIFP.

The satisfaction relation is defined using the inflationary relation $\mathcal{R}^{\phi,\mathfrak{I}}_{\text{INF}}$ generated by $\mathcal{R}^{\phi,\mathfrak{I}}$. Given an interpretation $\mathfrak{I}$ we define

$$\mathfrak{I} \models [rifp_{X,Y}\phi(X, Y)](\bar{t}) \text{ iff } (t_1^{\mathfrak{I}}, \ldots, t_k^{\mathfrak{I}}) \in \bigcup \mathbf{rifp}(\mathcal{R}^{\phi,\mathfrak{I}}) = \bigcup \text{IniFP}(\mathcal{R}^{\phi,\mathfrak{I}}_{\text{INF}}).$$

4.1 Examples

In the following, we will show how to use RIFP to express some queries.

Example 4.1 (Even Cardinality) *The* EVEN *query corresponds to the class of structures whose domains have even cardinality. Note that a set A has even cardinality if there is a partition $\{A', A''\}$ of A and a bijection $f : A' \to A''$. We will use the* **rifp** *operator to construct such a bijection (actually, a binary relation which is the graph of the bijection). Let Y and X be binary relation variables and consider the following formulas:*

$$\begin{aligned}
&\mathit{FUNC}(Y) &&= \forall x \forall y \forall z [Yxy \wedge Yxz \to y = z],\\
&\mathit{INJ}(Y) &&= \forall x \forall y \forall z [Yxy \wedge Yzy \to x = z],\\
&\mathit{TOT}(Y) &&= \forall x \exists y [Yxy \vee Yyx],\\
&\mathit{DIS}(Y) &&= \forall x \neg [\exists y Yxy \wedge \exists z Yzx],
\end{aligned}$$

where $TOT(Y)$ says that each element occurs in some pair in Y.

Let $\phi(X, Y)$ be the formula

$$\phi(X, Y) = (X = \emptyset) \wedge FUNC(Y) \wedge INJ(Y) \wedge TOT(Y) \wedge DIS(Y).$$

Let $\mathfrak{A}$ be an structure and $\mathcal{R}^{\phi,\mathfrak{A}}$ be the relation defined by $\phi(X, Y)$ on $\mathfrak{A}$. A pair (B, B') belongs to $\mathcal{R}^{\phi,\mathfrak{A}}$ iff $B = \emptyset$ and B' is the graph of a bijection on a partition of the domain A of $\mathfrak{A}$. Such bijection exists if and only if $\mathfrak{A}$ has even cardinality. The relation $\mathcal{R}^{\phi,\mathfrak{A}}_{\text{INF}}$ contains $\mathcal{R}^{\phi,\mathfrak{A}}$ and the pairs (B, B) for all $B \subseteq A^2$ such that $B \neq \emptyset$. If A has odd cardinality, then any chain on $\mathcal{R}^{\phi,\mathfrak{A}}_{\text{INF}}$ has length 1 and the only initial inflationary fixed-point is $\emptyset$. If A has even cardinality, then any chain of length 2 reaches an initial inflationary fixed-point which is the graph of a bijective function, as described above. It follows that $\bigcup \mathbf{rifp}(\mathcal{R}^{\phi,\mathfrak{A}})$ is empty iff A has odd cardinality. The sentence

$$\psi_{\text{EVEN}} = \exists u \exists v [rifp_{X,Y}\phi(X, Y)](u, v)$$

is satisfied by $\mathfrak{A}$ iff $\bigcup \mathbf{rifp}(\mathcal{R}^{\phi,\mathfrak{A}})$ is nonempty. It only occurs if A has even cardinality. Hence, ψ_{EVEN} defines the EVEN *query.* □

Example 4.2 (Satisfiability) *The SAT problem consists of the set of formulas of propositional logic in conjunctive normal form (CNF) that are satisfiable. A formula α in CNF has the form*

$$\alpha = C_1 \wedge \ldots \wedge C_m,$$

where each C_i is a clause, that is, a disjunction like

$$C_i = l_1 \vee \ldots \vee l_s$$

and each l_j is a literal, that is, it is either a propositional symbol p_k, and we say that p_k occurs positive in C_i, or a negated propositional symbol $\neg p_k$, and we say that p_k occurs negative in C_i.

As we want to use formulas to express decision problems, we often have to represent problem inputs as finite structures. Let $\mathfrak{A}_\alpha$ be a structure on the symbol-set $S = \{P, N\}$ where P and N are binary relations. Let r be the number of propositional symbols that occur in α and $A = \{1, \ldots, max\{m, r\}\}$. Let $P^{\mathfrak{A}} = \{(i, j) \mid p_j \text{ occurs positive in } C_i\}$ and $N^{\mathfrak{A}} = \{(i, j) \mid p_j \text{ occurs negative in } C_i\}$. The first position of P and N represents a clause and the second one a propositional symbol. Without loss of generality, we consider that any formula has at least as many clauses as propositional symbols, which can be achieved by adding dummy clauses $(p_0 \vee \neg p_0)$ to a formula without changing its satisfiability.

We will write an RIFP-formula that is satisfied by those structures $\mathfrak{A}_\alpha$ such that α is satisfiable. We use the **rifp** *operator to construct ternary relations whose tuples represent the truth value of propositional symbols, clauses and the formula according to some valuation of propositional symbols. We use triples (a, b, c) such that a can assume the values 0, 1 or 2, according to whether it refers to a propositional symbol, a clause or a formula, respectively, b represents the index of a clause or propositional symbol, and c can be 0 or 1, for true or false, respectively. This relation will be constructed in three stages. First, we put the triples that represent a valuation. Let $PVAL(Y)$ be the formula*

$$PVAL(Y) = \forall x \forall y \forall z (Yxyz \rightarrow x = 0) \wedge \forall y ((Y0y0 \vee Y0y1) \wedge \neg (Y0y0 \wedge Y0y1)),$$

and $\phi_0(X, Y)$ be the formula

$$\phi_0(X, Y) = (X = \emptyset) \wedge PVAL(Y).$$

Two relations $\mathbf{X}, \mathbf{Y}$ satisfy $\phi_0(X, Y)$ iff $\mathbf{X} = \emptyset$ and $\mathbf{Y}$ represents a valuation, where $(0, i, 1)$ means that proposition symbol i has value true, similarly for $(0, i, 0)$. In the second step, we include those triples corresponding to the truth values of clauses. We may consider that X contains those triples that represent a valuation and were included in the first step. Consider the formula $CVAL(Y)$

$$CVAL(Y) = \forall x \forall y \forall z (Yxyz \rightarrow x = 1) \wedge \forall y ((Y1y0 \vee Y1y1) \wedge \neg (Y1y0 \wedge Y1y1)),$$

and the formula $CL(X, Y)$

$$CL(X,Y) = CVAL(Y) \wedge \forall y(Y1y1 \leftrightarrow \exists p((Pyp \wedge X0p1) \vee (Nyp \wedge X0p0))).$$

Let $\phi_1(X,Y)$ be the formula

$$\phi_1(X,Y) = PVAL(X) \wedge CL(X,Y).$$

If $\mathbf{X}$ and $\mathbf{Y}$ satisfy $\phi_1(X,Y)$ then $\mathbf{X}$ corresponds to a valuation and $\mathbf{Y}$ represents the truth values of clauses according to the valuation $\mathbf{X}$. The last step calculates the truth value of the formula based on the truth values of its clauses. Let $FVAL(Y)$ be the formula

$$FVAL(Y) = ((Y200 \vee Y201) \wedge \neg(Y200 \wedge Y201)).$$

The formula $FVAL(Y)$ means that Y either have the tuple $(2,0,0)$, indicating that the input of SAT is not satisfiable, or the tuple $(2,0,1)$, otherwise. Let $FOR(X,Y)$ be the formula

$$FOR(X,Y) = Y201 \leftrightarrow (\forall y X1y1)$$

and

$$\phi_2(X,Y) = FVAL(Y) \wedge FOR(X,Y).$$

Finally, let $\phi(X,Y) = \phi_0(X,Y) \vee \phi_1(X,Y) \vee \phi_2(X,Y)$. A pair $(\mathbf{X},\mathbf{Y})$ satisfies $\phi(X,Y)$ iff it satisfies some $\phi_i(X,Y)$, $0 \le i \le 2$. In particular, $(2,0,1) \in \mathbf{Y}$ iff the formula α is satisfiable. It follows that $(2,0,1) \in \bigcup \mathbf{rifp}(\mathcal{R}^{\phi,\mathfrak{A}_\alpha})$ iff α is satisfiable. Let

$$\psi_{\mathrm{SAT}} = [\mathit{rifp}_{X,Y}\phi(X,Y)](201).$$

Then $\mathfrak{A}_\alpha \models \psi_{\mathrm{SAT}}$ iff α is satisfiable. □

We end this section with the following Theorem that relates the expressive power of RIFP and IFP.

Theorem 4.3 (IFP $\subseteq$ RIFP) *An IFP-formula has an equivalent one in RIFP.*

Proof. Let ψ be the following IFP-formula:

$$\psi := [\mathit{ifp}_{X,\overline{x}}\phi(X,\overline{x})](\bar{t}).$$

Let $\mathfrak{I}$ be an interpretation and $\phi^{\mathfrak{I}}$ be the operator defined by ϕ on $\mathfrak{I}$. Let

$$\phi_0^{\mathfrak{I}}(=\emptyset), \phi_1^{\mathfrak{I}}, \phi_2^{\mathfrak{I}}, \ldots$$

be the sequence of stages of $\phi^{\mathfrak{I}}$. We will define a formula $\theta(X,Y)$ such that the only chain in $\mathcal{R}^{\theta,\mathfrak{I}}$ leading to an initial fixed-point coincides with the stages sequence of $\phi^{\mathfrak{I}}$. Let $\theta(X,Y)$ be the formula

$$\theta(X,Y) = \forall\overline{x}(Y\overline{x} \leftrightarrow \phi(X,\overline{x})).$$

The relation $\mathcal{R}^{\theta,\mathfrak{I}}$ is exactly the graph of the operator $\phi^{\mathfrak{I}}$. Hence, any finite initial segment of the sequence of stages of $\phi^{\mathfrak{I}}$ is a chain of $\mathcal{R}^{\theta,\mathfrak{I}}$ and vice versa. Moreover, since $\phi^{\mathfrak{I}}$ is inflationary, then $\mathcal{R}^{\theta,\mathfrak{I}} = \mathcal{R}^{\theta,\mathfrak{I}}_{\mathrm{INF}}$. It follows that $\mathcal{R}^{\theta,\mathfrak{I}}_{\mathrm{INF}}$ has only one initial fixed-point and it is equal to the inflationary fixed-point of $\phi^{\mathfrak{I}}$. Thus, the formula

$$\psi' = [rifp_{X,Y}\theta(X,Y)](\bar{t})$$

is equivalent to ψ.

□

5 RIFP Captures PH

In this section, we will show that the expressive power of RIFP is equivalent to that of SO. Since SO captures the polynomial hierarchy (PH) [21], we have that RIFP also captures PH. To show that the expressive power of RIFP is the same as that of SO, we will give translations between the two languages. Consider the following translation from SO to RIFP:

$$\begin{aligned}
Tr(t_0 \equiv t_1) &:= t_0 \equiv t_1 \\
Tr(Rt_1 \dots t_k) &:= Rt_1 \dots t_k \\
Tr(Xt_1 \dots t_k) &:= Xt_1 \dots t_k \\
Tr(\neg\phi) &:= \neg Tr(\phi) \\
Tr(\phi_1 \wedge \phi_2) &:= Tr(\phi_1) \wedge Tr(\phi_2) \\
Tr(\exists x\phi) &:= \exists x Tr(\phi) \\
Tr(\exists Y\phi(Y)) &:= \exists\overline{x}[rifp_{X,Y}\psi(X,Y)]\,(\overline{x}) \vee Tr(\phi(\emptyset))
\end{aligned}$$

where ψ is the formula

$$\psi(X,Y) = (X = \emptyset \wedge Tr(\phi(Y))).$$

Lemma 5.1 *Let $\phi \in SO$. Then $\phi \equiv Tr(\phi)$ [12].*

The proof is by induction. The interesting case is the case of the second-order existential quantifier. Note that a pair $(\mathbf{V_1}, \mathbf{V_2})$ is in $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$ iff $V_1 = \emptyset$ and $\mathfrak{I}\frac{V_2}{Y} \models \phi(Y)$. We have two cases: i) $\mathfrak{I}\frac{\emptyset}{Y} \models \phi(Y)$ and in this case, $\mathfrak{I}\frac{V_2}{Y} \models Tr(\phi(\emptyset))$, or ii) $\mathfrak{I}\frac{V_2}{Y} \not\models \phi(Y)$. In this case, we have that any chain in $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$, if there is some, has size two, and any initial fixed-point is a non-empty relation V_2 that satisfies $\phi(Y)$. Conversely, any such relation V_2 is in some chain and it is an initial fixed-point. It follows that $\mathfrak{I}\frac{V_2}{Y} \models \exists\overline{x}[rifp_{X,Y}\psi(X,Y)]\,(\overline{x})$ since, by the inductive hypothesis, $Tr(\phi(Y))$ is equivalent to $\phi(Y)$. Hence the equivalence follows.

For the converse, that is, the definition of the translation from RIFP to SO, we have to be able to reconstruct the chains that give rise to the initial inflationary fixed-points using second-order logic. The key fact is the inflationary property of the relation $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$ that guarantees that any initial fixed-point can be reached through a chain of sufficiently short length.

Let $\phi(X, Y)$ be a formula with X and Y k-ary relation variables, and $\mathfrak{I}$ an interpretation whose domain has cardinality n. Let $X_0, X_1, \ldots, X_m$ be a chain in $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$ and X_m an initial fixed-point. As $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$ is inflationary, m is at most n^k (Lemma 3.2). Suppose $<$ is a $2k$-ary relation on $\mathfrak{I}$ which is an ordering on k-tuples of elements in $\mathfrak{I}$ and, for $1 \leq i \leq n^k$, let $\overline{t_i}$ the i-th k-tuple with respect to $<$. Let X' be a $2k$-ary relation defined as:

$$X' = \{(\overline{t_i}, \overline{a}) \in A^{2k} \mid \overline{a} \in X_i\}.$$

We use the first k positions of a tuple in X' to indicate the index i of a relation X_i in the chain, and the other k positions to represent a k-tuple in X_i. We can use existential quantification to check the existence of a relation like X' that witnesses the existence of a chain that reaches an initial fixed-point. We have to guarantee that successive relations in the chain are pairs in $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$. Consider the following formula:

$$\begin{aligned} \mathrm{COMPUTE}(\phi, X', \overline{t}) := {} & \exists \overline{t}' \exists X\, \exists Y (\overline{t}' = \overline{t} + 1 \wedge \\ & \phi(X, Y) \wedge \neg\phi(X, X) \wedge \\ & \forall \overline{a}(X'\overline{t}, \overline{a} \leftrightarrow X\overline{a}) \wedge \\ & \forall \overline{b}(X'\overline{t}', \overline{b} \leftrightarrow Y\overline{b} \vee X\overline{b})). \end{aligned}$$

The formula $\mathrm{COMPUTE}(\phi, X', \overline{t})$ means that $\overline{t}$ and $\overline{t}'$ are consecutive with respect to $<$, the projections of X' at $\overline{t}$ and $\overline{t}'$ form a pair in $\mathcal{R}^{\phi,\mathfrak{I}}_{\mathrm{INF}}$, and the projection at t is not a fixed-point.

The formula $\mathrm{START}(X')$ means that the relation X_1 of the chain is the projection of X' at $\overline{t_1}$ (recall that $X_0 = \emptyset$ in a chain):

$$\mathrm{START}(X') := \exists Y(\phi(\emptyset, Y) \wedge \forall \overline{a}(X'\overline{t_1}\overline{a} \leftrightarrow Y\overline{a})).$$

Finally, the formula

$$\mathrm{END}(X', \overline{t}_{\mathrm{pf}}, \overline{z}) := \exists X(\forall \overline{a}(X'\overline{t}_{fp}\overline{a} \leftrightarrow X\overline{a}) \wedge \phi(X, X)) \wedge X\overline{t}_{fp}\overline{z}$$

means that the projection of X' on $\overline{t}_{fp}$ is a fixed-point and that $\overline{z}$ belongs to this projection.

Let $Ord(<)$ be a first-order formula stating that $<$ is a linear order on k-tuples.

Consider the following translation from RIFP to SO:

$$
\begin{aligned}
Tr'(t_0 \equiv t_1) &:= t_0 \equiv t_1 \\
Tr'(Rt_1 \dots t_k) &:= Rt_1 \dots t_k \\
Tr'(Xt_1 \dots t_k) &:= Xt_1 \dots t_k \\
Tr'(\neg\phi) &:= \neg Tr'(\phi) \\
Tr'(\phi_1 \wedge \phi_2) &:= Tr'(\phi_1) \wedge Tr'(\phi_2) \\
Tr'(\exists x\phi) &:= \exists x Tr'(\phi) \\
Tr'([\mathit{rifp}_{X,Y}\phi(X,Y)](\overline{u})) &:= \exists < \exists X' \, \exists \overline{t}_{\mathrm{fp}}(Ord(<) \ \wedge \mathrm{START}(X) \wedge \\
&\quad\ \forall \overline{t}(\overline{t} < \overline{t}_{pf} \rightarrow \mathrm{COMPUTE}'(X,\overline{t})) \wedge \\
&\quad\ \mathrm{END}(X, \overline{t}_{\mathrm{pf}}, \overline{u})).
\end{aligned}
$$

where $\mathrm{COMPUTE}'(X,\overline{t})$ is obtained from $\mathrm{COMPUTE}(X,\overline{t})$ replacing ϕ with $Tr'(\phi)$.

Lemma 5.2 *Let $\phi \in$ RIFP. Then $\phi \equiv Tr'(\phi)$ [12].*

The proof is by induction and the interesting case, that of the fixed-point operator, is exactly the explanation given above.

Theorem 5.3 (RIFP = SO) *RIFP and SO have the same expressive power.*

It follows immediately from Theorem 5.3 and the fact that SO captures the polynomial hierarchy [21] that:

Corollary 5.4 (RIFP = PH) *RIFP captures the polynomial hierarchy PH.*

6 RIFP$_1$ captures NP

From Theorem 5.3, we have that RIFP is equivalent to SO. Fagin's theorem shows that the existential fragment of SO captures NP. The existential fragment of SO has formulas of the form

$$\exists X_1 \dots \exists X_q \phi$$

where ϕ is a first-order formula. There is a syntactically defined fragment of RIFP that corresponds to the existential fragment of SO. We call RIFP$_1$ the fragment of RIFP consisting of the formulas of the form

$$\exists x_1 \dots \exists x_k [\mathit{rifp}_{X,Y}\phi(X,Y)](\overline{u})$$

where $\phi(X,Y)$ is first-order.

The translations between RIFP and SO presented above do not directly give us the result we want, because they do not map the existential fragment of SO into RIFP, and vice versa. Let us consider the translation Tr from SO to RIFP. According to the translation, each existential quantifier will be replaced by an **rifp** construct. Hence, an existential formula may be mapped into an RIFP-formula with several nested **rifp** operators. To overcome this difficulty, given an existential formula with a block of existential quantifiers, we can transform it in an equivalent formula with only one existential quantifier. Then, each existential formula is equivalent to an RIFP-formula with only one **rifp** operator. However, by definition, we have:

$$Tr(\exists Y \phi(Y)) = \exists \overline{x}[\mathit{rifp}_{X,Y}\psi(X,Y)]\,(\overline{x})\ \vee Tr(\phi(\emptyset)),$$

which is not in the exact shape of an RIFP_1-formula. But we can see that $Tr(\exists Y \phi(Y))$ is equivalent to

$$\exists \overline{x}[\mathit{rifp}_{X,Y}\psi'(X,Y)]\,(\overline{x}),$$

where

$$\psi'(X,Y) = (X = \emptyset \wedge Y \neq \emptyset \wedge Tr(\phi(Y))) \vee (\phi(\emptyset) \wedge Y\overline{x}).$$

We conclude:

Lemma 6.1 ($\exists$SO $\subseteq$ RIFP$_1$) *The existential second-order logic is contained in RIFP$_1$.*

On the other hand, the translation Tr' from RIFP to SO when applied to formulas in RIFP_1, also does not give us an existential formula. Obviously, it can be put in the prenex form but, if we consider the last clause of the definition of Tr'

$$\begin{aligned} Tr'([\mathit{rifp}_{X,Y}\phi(X,Y)](\overline{u})) := \exists < \exists X'\, \exists \overline{t}_{\mathrm{fp}}(&Ord(<)\ \wedge \mathrm{START}(X) \wedge \\ &\forall \overline{t}(\overline{t} < \overline{t}_{pf} \to \mathrm{COMPUTE}'(X,\overline{t})) \wedge \\ &\mathrm{END}(X, \overline{t}_{\mathrm{pf}}, \overline{u})), \end{aligned}$$

the subformula $\forall \overline{t}(\overline{t} < \overline{t}_{pf} \to \mathrm{COMPUTE}'(X,\overline{t}))$ implies that we will have a universal first-order quantifier before the existential second-order quantifiers in $\mathrm{COMPUTE}'$. However, first-order quantifiers can be pushed inside the first-order part. For example, the formula $\forall x \exists X \phi$, where X has arity k, is equivalent to $\exists X'(\forall x \phi')$, where X' has arity $k+1$ and ϕ' is obtained from ϕ by replacing any atomic subformula $X\overline{t}$ with $X'x\overline{t}$. We conclude:

Lemma 6.2 (RIFP$_1$ $\subseteq$ $\exists$SO) *RIFP$_1$ is contained in the existential second-order logic.*

Theorem 6.3 (RIFP$_1$ = $\exists$SO) *RIFP$_1$ has the same expressive power as existential second-order logic.*

Therefore,

Corollary 6.4 (RIFP$_1$ = NP) *RIFP$_1$ captures NP.*

7 Comparisons between RIFP_1 and the Nondeterministic Inflationary Fixed-Point Logic

In 1997, Abiteboul, Vianu, and Vardi [4] introduced the nondeterministic inflationary fixed-point logic NIFP where two first-order operators are defined and applied in any possible order.

Let $\Phi_1, \Phi_2 : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ be two operators. They give rise to sequences of sets

$$S_0, S_1, \ldots,$$

where each $S_i \subseteq A^k$, $S_0 = \emptyset$, and $S_{i+1} = \Phi_1(S_i)$ or $S_{i+1} = \Phi_2(S_i)$. A *local fixed-point* of Φ_1, Φ_2 is a set S_m in some sequence as above such that $S_m = \Phi_1(S_m) = \Phi_2(S_m)$. Let $\mathbf{nifp}(\Phi_1, \Phi_2)$ be the set of local fixed-points of Φ_1, Φ_2. The nondeterministic inflationary fixed-point logics extends FO with formulas of the form

$$[\mathit{nifp}_{X,\overline{x}}\phi_1(X,\overline{x}), \phi_2(X,\overline{x})](\overline{t}),$$

where ϕ_1 and ϕ_2 are first-order formulas of the form $\phi(X,\overline{x}) \vee X\overline{x}$, that is, they define inflationary first-order operators. The semantics of NIFP is defined as

$$\mathfrak{I} \models [\mathit{nifp}_{X,\overline{x}}\phi_1(X,\overline{x}), \phi_2(X,\overline{x})](\overline{t}) \text{ iff } (t_1^{\mathfrak{I}}, \ldots, t_k^{\mathfrak{I}}) \in \bigcup \mathbf{nifp}(\phi_1^{\mathfrak{I}}, \phi_2^{\mathfrak{I}}).$$

In 1997, Abiteboul, Vianu, and Vardi [4] also showed that NIFP captures NP on ordered structures. Then, using Theorem 6.3 and Fagin's theorem, we have that NIFP has the same expressive power as RIFP_1 on ordered structures. In the following, we will present a translation from NIFP to RIFP_1.

Let $\Phi_1, \Phi_2 : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ be two operators. They give rise to a binary relation as follows:

$$\mathcal{R} = \{(X, \Phi_1(X)) \mid X \in \mathcal{P}(A^k)\} \cup \{(X, \Phi_2(X)) \mid X \in \mathcal{P}(A^k)\}.$$

A local fixed-point X of the pair Φ_1, Φ_2 is a fixed-point of $\mathcal{R}$, as $\Phi_1(X) = \Phi_2(X) = X$, but, if $X = \Phi_1(X) \neq \Phi_2(X)$ or $X = \Phi_2(X) \neq \Phi_1(X)$, X is a fixed-point of $\mathcal{R}$ but not a local fixed-point of Φ_1, Φ_2. We have the following cases:

i. Case 1: $S_m = \Phi_1(S_m) = \Phi_2(S_m)$ is a local fixed-point of Φ_1, Φ_2 and a fixed-point of $\mathcal{R}$;
ii. Case 2: $S_m = \Phi_1(S_m)$ and $S_m \neq \Phi_2(S_m)$ is a fixed-point of $\mathcal{R}$;
iii. Case 3: $S_m \neq \Phi_1(S_m)$ and $S_m = \Phi_2(S_m)$, similar to the previous case;
iv. Case 4: $S_m \neq \Phi_1(S_m)$ and $S_m \neq \Phi_2(S_m)$ is neither a local fixed-point of Φ_1, Φ_2 nor a fixed-point of $\mathcal{R}$.

We have to define a relation which coincides with $\mathcal{R}$, except that it does not have the fixed-points that correspond to the cases 2 and 3 above. A fixed-point of Φ_1 and Φ_2 can be verified using the formulas C_1 and C_2 below, where ϕ_1 and ϕ_2

are first-order formulas that define Φ_1 and Φ_2:

$$C_1(X) := \forall \overline{x}(X\overline{x} \leftrightarrow \phi_1(\overline{x}, X))$$

$$C_2(X) := \forall \overline{x}(X\overline{x} \leftrightarrow \phi_2(\overline{x}, X))$$

We can use these formulas to include the pair $(X, \Phi_i(X))$ if X is not a fixed-point of Φ_i, $i = 1, 2$, or if it is a fixed-point of both Φ_1 and Φ_2. Let $\psi(X, Y)$ be the following formula:

$$\begin{aligned}\psi(X, Y) := {} & [\neg C_1(X) \wedge \forall \overline{x} Y\overline{x} \ \leftrightarrow \ \phi_1(\overline{x}, X)] \vee \\ & [\neg C_2(X) \wedge \forall \overline{x} Y\overline{x} \ \leftrightarrow \ \phi_2(\overline{x}, X)] \vee \\ & [C_1(X) \wedge C_2(X) \wedge \forall \overline{x} X\overline{x} \ \leftrightarrow \ Y\overline{x}].\end{aligned}$$

A set X is an initial inflationary fixed-point of $\mathcal{R}^{\psi,\mathfrak{I}}$ iff it is a local fixed-point of Φ_1, Φ_2. The formula

$$\phi = [nifp_{S,\overline{x}}\, \phi_1(S), \phi_2(S)\,]\,(\bar{t})$$

from NIFP is equivalent to the formula

$$\psi = [rifp_{X,Y} \psi(X, Y)]\,(\bar{t})$$

from RIFP. Since ϕ_1 and ϕ_2 are first-order, then ψ is in RIFP$_1$. Hence we have proved that:

Theorem 7.1 (NIFP $\subseteq$ RIFP$_1$) *NIFP is contained in RIFP$_1$.*

In general, the converse translation from RIFP$_1$ to NIFP is not possible as the equivalence between RIFP$_1$ and NIFP only holds for ordered structures. To show that NIFP is less expressive than RIFP$_1$ we use the fact that the query Even is RIFP$_1$-definable, but it is not definable in the infinitary logic $L^{\omega}_{\infty\omega}$. We can translate NIFP to $L^{\omega}_{\infty\omega}$. Therefore, we have that RIFP$_1$ $\not\subseteq$ NIFP. We use $L^{\omega}_{\infty\omega}$ to build a formula that expresses a local nondeterministic fixed-point of a pair of first-order operators. Let us see how such formula is defined.

An application sequence is an infinite sequence $w = w_1, w_2, \ldots$ such that $w_i \in \{1, 2\}$ for $i \in \mathbb{N}^*$, where both numbers 1 and 2 appear infinitely many times. Let $\Phi_1, \Phi_2 : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ be inflationary operators and w an application sequence. The sequence of stages of Φ_1, Φ_2 based on w is the infinite sequence $(\Phi_1, \Phi_2)^w = S_0, S_1, \ldots$ of subsets of A^k such that $S_0 = \emptyset$ and $S_{i+1} = \Phi_{w_{i+1}}(S_i)$. We call S_i the ith stage of the sequence $(\Phi_1, \Phi_2)^w$, denoted by $(\Phi_1, \Phi_2)^w_i$. A sequence of stages always reaches a local nondeterministic fixed-point of Φ_1, Φ_2 at some stage $(\Phi_1, \Phi_2)^w_m$. Let $\phi_1(X, \overline{x})$ and $\phi_2(X, \overline{x})$ be first-order formulas and let w be an application sequence. Each stage of a sequence of stages can be defined by a first-order formula. For every $i \in \mathbb{N}$, there is a first-order formula $\psi^w_i(\overline{x})$ such that for every interpretation $\mathfrak{I}$, $\psi^w_i(\overline{x})$ defines the i-th stage $(\phi^{\mathfrak{I}}_1 \phi^{\mathfrak{I}}_2)^w_i$ of the sequence of stages $(\phi^{\mathfrak{I}}_1, \phi^{\mathfrak{I}}_2)^w$ based on the application sequence w. There is a formula $\psi^w(\overline{x})$ in $L^{\omega}_{\infty\omega}$ such that, for every interpretation $\mathfrak{I}$, $\psi^w(\overline{x})$ defines the local nondeterministic fixed-point reached by the sequence of stages $(\phi^{\mathfrak{I}}_1 \phi^{\mathfrak{I}}_2)^w$ based on the application sequence w. Let $\psi^w(\overline{x})$ be defined as $\psi^w(\overline{x}) = \bigvee\{\psi_i^w(\overline{x}) \mid i \in \mathbb{N}\}$. Similarly, there is a formula $\psi^{\phi_1,\phi_2} \in L^{\omega}_{\infty\omega}$ expressing the nondeterministic fixed-point operators defined by ϕ_1 and ϕ_2. Let ψ^{ϕ_1,ϕ_2} be defined as $\psi^{\phi_1,\phi_2}(\overline{x}) = \bigvee\{\psi^w(\overline{x}) \mid \text{for every application sequence } w\}$. It follows that:

Theorem 7.2 (NIFP $\subseteq L^{\omega}_{\infty\omega}$) *Every NIFP formula has an equivalent formula in $L^{\omega}_{\infty\omega}$.*

Therefore we have:

Theorem 7.3 ($\mathbf{RIFP}_1 \not\subseteq$ NIFP) *$RIFP_1$ is not contained in NIFP.*

8 Conclusions and Future Works

We investigated first-order logic with relational fixed-point operators whose semantics differs from the semantics of classical fixed-point logics using relations instead of operations. We call it Relational Inflationary Fixed Point Logic — RIFP. Our approach generalizes that of traditional fixed-point logics adding to them expressive power to cope with nondeterminism.

We characterized the expressive power of RIFP showing that RIFP is equivalent to SO, that is, each formula of RIFP has an equivalent in SO, and vice versa. It follows that RIFP captures the polynomial hierarchy PH. We provide translations mapping formulas in one language to equivalent formulas in the other language. We also investigated the RIFP_1 fragment of formulas of the form

$$\exists\overline{z}[\mathbf{rifp}_{X,Y}\phi(X,Y)](\bar{t}),$$

where $\phi(X,Y)$ is a first-order formula. Small changes in the translations given show that RIFP_1 is equivalent to the existential fragment of SO and, hence, it captures NP, the class of problems solved in nondeterministic polynomial time.

We also compared our logic with nondeterministic inflationary logic NIFP. NIFP generalizes traditional fixed-point logics using a pair of first-order operators to construct fixed-points through sequences obtained by successively applying the operators. NIFP captures NP on ordered structures [4] and therefore has the same expressive power of RIFP_1. We showed how to construct an RIFP_1-formula equivalent to a given NIFP-formula.

The framework here presented can be used to explore other kinds of relational fixed-point logics. In the same way that new fixed-point logics are defined imposing some conditions on the operators used, we can define other relational fixed-point logics restricting the relations considered. Using this approach, several logics can be defined in a uniform way. We conclude by giving some natural next steps in this line of research, namely:

- To define the partial fixed-point version of the relational operator: The equivalence between RIFP and SO strongly depends on the fact that relations are inflationary. Removing this condition should lead to an increase of expressive power;

- To investigate the restriction of logic to monadic fragment of **RIFP** logic: The monadic fragment of RIFP is obtained restricting the **rifp** operator to formulas $\phi(X, Y)$ where X and Y are monadic relations. The translation from SO to RIFP maps monadic formulas from SO to monadic formulas in RIFP. The translation from RIFP to SO does not preserve the arity of the relations involved;
- To analyze the application of relational operators for higher-order logics beyond SO;
- To apply RIFP and its fragments to problems in the scope of Database Theory.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1995.

[2] A. V. Aho and J. D. Ullman. Universality of data retrieval languages. In *Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 110–119. ACM, 1979.

[3] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43(1):62–124, 1991.

[4] S. Abiteboul, M. Vardi, and V. Vianu. Fixpoint logics, relational machines, and computational complexity. *Journal of the ACM*, 44 - 1:30–56, 1997.

[5] J. R. Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

[6] A. K. Chandra and D. Harel. Structure and complexity of relational queries. *Foundations of Computer Science 21st Annual Symposium on*, 1980.

[7] C. C. Chang and H. J. Keisler. *Model Theory*. North-Holland, Amsterdam, 3rd edition, 1990.

[8] E. F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970.

[9] H. D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1999.

[10] H. D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Undergraduate Texts in Mathematics. Springer-Verlag, Berlin, 2nd edition, 1994.

[11] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *SLAM-AMS Proceedings*, 7:43–73, 1974.

[12] M. R. F. Farias. *Complexidade Descritiva de Lógicas de Ponto Fixo Relacionais*. PhD thesis, Federal University of Ceará, 2016.

[13] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and its applications (Vol.2)*. Heidelberg: Springer, 2007.

[14] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of pure and applied logic*, 32:265–280, 1986.

[15] W. Hodges. *Model Theory*, volume 42 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1993.

[16] N. Immerman. Relational queries computable in polynomial time. *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, 1982.

[17] N. Immerman. *Descriptive Complexity*. Springer, 1999.

[18] S. Kreutzer. *Pure and applied fixed-point logics*. PhD thesis, Bibliothek der RWTH Aachen, 2002.

[19] P. G. Kolaitis and M. Y. Vardi. On the expressive power of variable-confined logics. In *Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 348–359. IEEE, 1996.

[20] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1994.

[21] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[22] M. Y. Vardi. The complexity of relational query languages. *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 137–146, 1982.