$PTIME$ and Generalized Quantifiers

Hannu Nissinen

Helsinki December 15, 2013

Master's Thesis

UNIVERSITY OF HELSINKI
Department of Mathematics and Statistics

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Mathematics and Statistics

Tekijä — Författare — Author

Hannu Nissinen

Työn nimi — Arbetets titel — Title

$PTIME$ and Generalized Quantifiers

Oppiaine — Läroämne — Subject

Mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's Thesis	December 15, 2013	53 pages

Tiivistelmä — Referat — Abstract

We consider the problem of finding a reasonable logical characterization for the complexity class $PTIME$ in the class of all finite models. We approach this problem by adding a set $\mathbf{Q}_n$ of n-ary generalized quantifiers to the infinitary finite variable logic $\mathcal{L}^{\omega}_{\infty\omega}$. More precisely, we show that it is *not* possible to characterize $PTIME$ in such a way. This result is obtained by constructing models $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ which are $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$-equivalent and by showing that there is a $PTIME$ computable boolean query q such that $q(\mathbf{A}(\mathbf{G})) \neq q(\mathbf{B}(\mathbf{G}))$ for any appropriate finite graph $\mathbf{G}$.

****

Työssä käsitellään ongelmaa tyydyttävän loogisen karakterisaation löytämiseksi vaativuusluokalle $PTIME$ kaikkien äärellisten mallien luokassa. Tätä ongelmaa lähestytään lisäämällä kaikki n-paikkaiset yleistetyt kvanttorit infinitaariseen äärellisen monen muuttujan logiikkaan $\mathcal{L}^{\omega}_{\infty\omega}$. Työssä todistetaan, ettei ole mahdollista karakterisoida vaativuusluokkaa $PTIME$ kyseisellä tavalla. Tämä tulos saavutetaan konstruoimalla mallit $\mathbf{A}(\mathbf{G})$ ja $\mathbf{B}(\mathbf{G})$, jotka ovat $\mathcal{L}^{k}_{\infty\omega}$-ekvivalentteja ja näyttämällä, että on olemassa polynomiaalisessa ajassa laskettava boolen kysely q, jolle pätee $q(\mathbf{A}(\mathbf{G})) \neq q(\mathbf{B}(\mathbf{G}))$ kaikilla sopivilla äärellisillä verkoilla $\mathbf{G}$.

Avainsanat — Nyckelord — Keywords

descriptive complexity, PTIME, generalized quantifiers, infinitary finite variable logic

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Computational complexity theory is a field of study interested in classifying computational problems with respect to time and space resources needed to solve them. It is widely agreed that $PTIME$, the class of problems solvable with a Turing machine running in polynomial time, consists of problems that are computationally feasible. However, it is surprising how little is known about the fundamental properties of $PTIME$. There is of course a vast list of problems known to be in $PTIME$, but the common features of them have remained a mystery. Put differently, giving an *exact and general description* of what kind of problems constitute $PTIME$ has turned out to be extremely difficult. This challenge leads us to consider a logician's approach to complexity theory.

Descriptive complexity theory aims to characterize complexity classes by means of mathematical logic. This approach is more abstract than the one traditionally used in complexity theory. However, this is not to be understood that one of the frameworks is better than the other - they are merely interested in somewhat different aspects of the topic. One can boldly assert that descriptive complexity theory takes a step further than the traditional approach. It aims to *describe* complexity classes instead of only studying their mutual relations. One is interested in logical forms of problems of given complexity. Consequently, a new measure for complexity emerges. A problem is as complex as is a logic needed to express the problem. Thus the concept of computational complexity reduces to the notion of logical definability. This gives an alternative way of explaining *why* a particular problem is in some class as well as *why* some classes are in some extensional relation to each other. The well-known result of Fagin [5] is a great example of such an explanation. It states that the class of problems solvable in non-deterministic polynomial time, $NPTIME$, consists exactly of the problems that can be expressed by sentences of existential second-order logic. Many similar characterizations of important complexity classes have been obtained after this pioneering result of Fagin. For instance, $LOGSPACE$ and $PSPACE$ were characterized by Immerman [9] and Vardi [15], respectively. Moreover, $PTIME$ was characterized as properties definable in least fixed-point logic, LFP, by the same researchers in the 1980's, but only in the class of ordered finite models [8, 15]. However, a general description for $PTIME$ in the class of finite models is still an open problem.

This thesis is about the problem of giving a general characterization for $PTIME$ in the class of all finite models. A possible approach is to add more expressive power to LFP. We adopt this strategy, although only indirectly. Finite variable logic with infinitary connectives, $\mathcal{L}^{\omega}_{\infty\omega}$, is known to be at least as expressive as LFP, but not powerful enough to capture $PTIME$. Thus we enrich $\mathcal{L}^{\omega}_{\infty\omega}$ with so-called *generalized quantifiers*. They behave syntactically similarly as the classical ones, but can express more complex properties such as "even number of", "at least five" etc. As opposed to traditional quantifiers, we allow them to apply to several formulas as well as bind several variables in a formula. We say that a quantifier is n-ary, if it binds at most n variables in each formula it applies to. It seems a priori possible that in the above fashion with a suitable collection of generalized quantifiers one could construct the desired logic. However, some negative results have been obtained. In [10], it was shown that the collection of all unary quantifiers does not yield a logic strong enough to capture $PTIME$. The main theorem of this thesis, originally published by Hella [7], generalizes the result in [10]. We show that for each n, it is *not* possible to capture $PTIME$ with a logic $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$, where $\mathbf{Q}_n$ denotes the class of all n-ary quantifiers.

The structure of this thesis is the following. Section 2 is devoted to defining a general concept of a logic and some important concrete logics as instances of it. We need a general and exact mathematical concept for a logic to be able to pose the question "is there a logic for $PTIME$?" in the first place. On the other hand, the concrete logics are needed to formulate the argument for the main theorem. In Section 3 we introduce generalized quantifiers by extending the concept of a quantifier until reaching the so-called Lindström quantifiers, which were originally introduced by Lindström [12]. In the next section we establish the link between computational complexity and logical definability. Consequently, we have all the tools to give an exact mathematical formulation for the question of existence of a reasonable logic to capture $PTIME$.

In Section 5 we consider the expressive power of the finite variable infinitary logics with n-ary quantifiers. We characterize $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$-equivalence in two different ways. Firstly, we show that a certain back-and-forth bijective extendability condition for partial isomorphisms between finite models $\mathcal{M}$ and $\mathcal{N}$ guarantees the preservation of $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$-formulas. We then proceed to a game-theoretic approach. We define an n-bijective k-pebble game and show that the existence of a winning strategy in this game over $\mathcal{M}$ and $\mathcal{N}$ gives the same result.

In Section 6 we dwell into the core issues of this thesis. The challenge is to construct models which are $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$-equivalent, but can be distinguished with a $PTIME$-

computable query. These models are constructed from a finite connected graph and so-called building blocks. In Section 7 we introduce the *game of k cops and a robber*. The desired equivalence is obtained with the help of this game. Afterwards, it is shown that the structures built from the blocks can be separated with a $PTIME$ computable property.

2 Logics

In this section we concentrate on various logics. We begin by introducing a general concept of a logic and call it an abstract logic. The name abstract refers to the fact that neither explicit syntactical rules for the construction of formulas nor their semantics is given. The idea is most importantly to give a set of general conditions so that all and only the concrete objects we are inclined to call logics satisfy them. On the other hand, we need this broad concept of a logic when we later pose the question about the existence of a reasonable logic describing the class of polynomial time computable queries.

2.1 Abstract Logic

The definition for an abstract logic we shall contemplate on was originally given by Kolaitis and Väänänen [10]. It is a refinement of the first formulation of an abstract logic given by Lindström [13]. Their modified definition has two essential new features. Most importantly, it allows to treat logics on any restricted classes of structures. Hence we can not only explicitly restrict our attention to the class of finite structures, but also any subclass of it. So in practise the uselfulness of this fine-grained treatment of model classes is that we can clearly express when we are for instance studying only ordered structures or arbitrary finite structures. The other new feature of the modified version is that it states the set of variables of a logic explicitly. Consequently, we can limit the number of variables naturally, when we are paying our attention to k-variable fragments of some concrete logic. Before turning to the definition of an abstract logic we point out that it is a genuine generalization of any concrete logic we are interested in. Hence all the concrete logics we study are abstract logics, differing only in the additional structural features they have. Thus we will often speak only of logics without making a distinction between abstract and concrete logics.

A *vocabulary*, denoted by greek letters $\tau, \sigma, ...$ is a set of constant symbols c_i, relation symbols R_i and function symbols f_i. Each of these three types of symbols comes with an arity, which is denoted by $ar(...)$. As usually, arity is a function from the set of symbols to natural numbers. The set of variables of a logic is denoted by $\mathcal{V}$.

The set of τ-terms is defined recursively as follows:

(i) For all $v \in \mathcal{V}$, v is a τ-term.

(ii) For all $c_i \in \tau$, c_i is a τ-term.

(iii) If $t_1, ..., t_n$ are τ-terms and f_i is an n-ary function symbol in τ, then $f_i(t_1, ..., t_n)$ is a τ-term.

Fix a vocabulary $\tau = \{c_i, .., f_i, ..., R_i, ...\}$. A *$\tau$-structure* or a *model* is a sequence $\mathcal{M} = \langle M, c_i^{\mathcal{M}}, ..., f_i^{\mathcal{M}}, ..., R_i^{\mathcal{M}}, ...\rangle$. The non-empty set M is called the *universe* or *domain* of $\mathcal{M}$. On the other hand, $c_i^{\mathcal{M}}, f_i^{\mathcal{M}}$ and $R_i^{\mathcal{M}}$ are the *interpretations* of c_i, f_i and R_i in the structure $\mathcal{M}$, respectively. Interpretations are defined in the usual way. Furthermore, we denote by $Assgn_{\mathcal{V},\mathcal{M}}$ the class of assignments of the elements of $\mathcal{V}$ in the universe of $\mathcal{M}$. Thus elements of $Assgn_{\mathcal{V},M}$ are functions from $\mathcal{V}$ to M.

Definition 2.1.1. An *abstract logic* on a model class $\mathcal{K}$ is a tuple $(\mathcal{L}, \mathcal{T}, \mathcal{K}, \mathcal{V}, \models_{\mathcal{L}})$ such that the following conditions hold:

(1) $\mathcal{T}$ is a set of vocabularies.

(2) $\mathcal{L}$ is a function such that $dom(\mathcal{L}) = \mathcal{T}$ and for all $\tau \in \mathcal{T}$, $\mathcal{L}[\tau]$ is a class. $\mathcal{L}[\tau]$ is called the class of $\mathcal{L}$-formulas of vocabulary τ.

(3) $\mathcal{K}$ is a function such that $dom(\mathcal{K}) = \mathcal{T}$ and for all $\tau \in \mathcal{T}$, $\mathcal{K}[\tau]$ is a class of τ-structures.

(4) $\mathcal{V}$ is a set of variable symbols.

(5) $\models_{\mathcal{L}}$ is a three-place relation of elements of $\mathcal{K}[\tau]$, $\mathcal{L}[\tau]$ and $Assgn_{\mathcal{V},\mathcal{M}}$, where $\tau \in \mathcal{T}$ and $\mathcal{M} \in \mathcal{K}[\tau]$.

(6) $\mathcal{L}$ is monotone i.e. if $\tau \subseteq \sigma$, then $\mathcal{L}[\tau] \subseteq \mathcal{L}[\sigma]$.

(7) If $\mathcal{M} \in \mathcal{K}[\tau]$ and $\mathcal{M} \models_{\mathcal{L},s} \varphi$, then $\varphi \in \mathcal{L}[\tau]$.

(8) Let $\mathcal{M}, \mathcal{N} \in \mathcal{K}[\tau], \varphi \in \mathcal{L}[\tau]$ and $s \in Assgn_{\mathcal{V},\mathcal{M}}$. If $\mathcal{M} \models_{\mathcal{L},s} \varphi$ and π is an isomorphism from $\mathcal{M}$ to $\mathcal{N}$, then

$$\mathcal{N} \models_{\mathcal{L},\pi(s)} \varphi.$$

(9) Let $\tau \subseteq \sigma$. If $M \in \mathcal{K}[\sigma], \varphi \in \mathcal{L}[\tau], s \in Assgn_{\mathcal{V},\mathcal{M}}$ and $\mathcal{M} \models_{\mathcal{L},s} \phi$, then

$$\mathcal{M} \restriction \tau \models_{\mathcal{L},s} \varphi.$$

(10) Let $f : \tau \to \sigma$ be a bijection preserving types and arities of symbols (i.e. mapping n-ary relation symbols to n-ary relation symbols etc.). For all $\varphi \in \mathcal{L}[\tau]$, there is $\varphi^f \in \mathcal{L}[\sigma]$ such that for all $\mathcal{M} \in \mathcal{K}[\tau]$ and all $s \in Assgn_{\mathcal{V},\mathcal{M}}$, it holds that $\mathcal{M} \models_{\mathcal{L},s} \varphi \Leftrightarrow \mathcal{M}^f \models_{\mathcal{L},s} \varphi^f$. Here $\mathcal{M}^f$ is the $\mathcal{L}[\sigma]$-structure we obtain by renaming objects of $\mathcal{M}$ according to f.

From now on we abuse our notation slightly by referring to a particular logic with the same symbol as in the first co-ordinate of the tuple in its definition. This abuse is of course made only when it is clear from the context what logic we mean.

If a triple $(\mathcal{M}, \varphi, s)$ belongs to the relation $\models_{\mathcal{L}}$, we write $\mathcal{M} \models_{\mathcal{L},s} \varphi$ and say that $\mathcal{M}$ satisfies the $\mathcal{L}$-formula φ with assingment s. Furthermore, if $\mathcal{M} \models_{\mathcal{L},s} \varphi$ for all assingnments $s \in Assgn_{\mathcal{V},\mathcal{M}}$, we write $\mathcal{M} \models_{\mathcal{L}} \varphi$ and say that $\mathcal{M}$ is a model of φ. Logical consequence is defined in the usual way: If Ψ is a set of $\mathcal{L}$-sentences, φ an $\mathcal{L}$-sentence and for all $\mathcal{M} \in \mathcal{K}[\tau]$ such that $\mathcal{M} \models_{\mathcal{L}} \Psi$, it holds also that $\mathcal{M} \models_{\mathcal{L}} \varphi$, we write $\Psi \models_{\mathcal{L}} \varphi$ and say that φ is a logical consequence of Ψ. If Ψ happens to be empty, we write just $\models_{\mathcal{L}} \varphi$ and say that φ is *valid*. Finally, we sometimes omit the subscript in $\models_{\mathcal{L}}$ when no risk of confusion arise.

Let us now fix a logic $\mathcal{L} = (\mathcal{L}, \mathcal{T}, \mathcal{K}, \mathcal{V}, \models_{\mathcal{L}})$ and make some observations on the previous definition. It has essentially two parts. Conditions from (1) to (5) define the basic notions, whereas the rest assert fundamental properties of them. The first part defines the set of $\mathcal{L}$-formulas and structures over τ, the set of variables and, most importantly, the satisfaction relation $\models_{\mathcal{L}}$.

Consider then conditions (6)-(10). The first one guarantees the syntactical property that the set of formulas of a logic is closed under expansions of vocabularies. The last four conditions from (7) to (10) are less technical and assert important metalogical properties of the truth predicate. Condition (7) states that the semantics of a logic is defined only on formulas of that particular logic. In other words, (7) rules out the possibility of asking what is the meaning of an expression that it is not even well-formed syntactically. According to the last three conditions, the truth predicate is preserved under isomorphisms, reducts and renaming. Thus truth of a formula depends only on symbols appearing in it (reduct property). Furthermore, truth does not depend on the names of objects (isomorphism property) and finally it does not depend on the names of symbols, if their interpretations are modified accordingly (renaming property).

As mentioned before, we shall in many cases be interested in restricting logics to certain classes of structures. Denote by $\mathcal{S}$ the class of all structures and by $\mathcal{F}$ the class of finite structures. By $\mathcal{L}/\mathcal{K}$ we mean the logic obtained from $\mathcal{L}$ by requiring that its models are in $\mathcal{K}$. So for example $\mathcal{L}/\mathcal{F} = (\mathcal{L}, \mathcal{T}, \mathcal{F}, \mathcal{V}, \models_{\mathcal{L}})$ is the logic $\mathcal{L}$ restricted to the class of finite structures.

2.2 General Syntactical and Semantical Rules

Both of the concrete logics that we will study are abstract logics with some additional explicit syntactical and semantical rules. So before introducing the concrete logics relevant for our later purposes, we define atomic formulas, connectives and quantifiers as general concepts. Later we define the logics themselves with the help of this extra step between abstract and concrete logics. In the next definition we assume that the logic $\mathcal{L}$ is closed under all the syntactical operations that are defined.

Definition 2.2.1. Let $\mathcal{L}$ be a logic, τ a vocabulary, $\mathcal{M}$ a τ-model and $t_1, ..., t_n$ τ-terms.

Atomic $[\tau]$*-formulas.* The set of atomic $[\tau]$-formulas, $Atom[\tau]$, is defined as follows:

$$(t_1 = t_2) \in Atom[\tau],$$

$$R(t_1, ..., t_n) \in Atom[\tau], \text{ for all } R \in \tau.$$

Their semantics is given as:

$$\mathcal{M} \vDash_{\mathcal{L},s} (t_1 = t_2) \Leftrightarrow s(t_1) = s(t_2),$$

$$\mathcal{M} \vDash_{\mathcal{L},s} R(t_1, ..., t_n) \Leftrightarrow \big(s(t_1), ..., s(t_n)\big) \in R^{\mathcal{M}}.$$

Negation. The negation of $\varphi \in \mathcal{L}[\tau]$ is the formula $\neg\varphi$ and its semantics is given as:

$$\mathcal{M} \vDash_{\mathcal{L},s} \neg\varphi \Leftrightarrow \mathcal{M} \nvDash_{\mathcal{L},s} \varphi.$$

Conjunction. Let Ψ be an arbitrary collection of $\mathcal{L}[\tau]$-formulas. Then $\bigwedge_{\varphi\in\Psi} \varphi$ is a formula, and its semantics is given as

$$\mathcal{M} \vDash_{\mathcal{L},s} \bigwedge_{\varphi\in\Psi} \varphi \Leftrightarrow \mathcal{M} \vDash_{\mathcal{L},s} \varphi, \text{for all } \varphi \in \Psi.$$

Disjunction. Let Ψ be an arbitrary collection of $\mathcal{L}[\tau]$-formulas. Then $\bigvee_{\varphi\in\Psi} \varphi$ is a formula, and its semantics is given as

$$\mathcal{M} \vDash_{\mathcal{L},s} \bigvee_{\varphi\in\Psi} \varphi \Leftrightarrow \mathcal{M} \vDash_{\mathcal{L},s} \varphi, \text{for some } \varphi \in \Psi.$$

Existential quantification. If $\varphi \in \mathcal{L}$ and $x \in \mathcal{V}_{\mathcal{L}}$, then the existential quantification gives the formula $\exists x\varphi$ and its semantics is given as follows:

$$\mathcal{M} \vDash_{\mathcal{L},s} \exists x\varphi \Leftrightarrow \mathcal{M} \vDash_{\mathcal{L},s(a/x)} \varphi, \text{for some } a \in M.$$

It is an elementary fact that implication, equivalence and the universal quantifier can be defined by means of the connectives, negation and the existential quantifier. Therefore we treat them as abbreviations and let them appear in formulas. We chose to introduce conjunction and disjunction so that they allow both finite and infinite amounts of formulas. In the infinite case, they can intuitively be grasped as universal and existential quantifications with the exception that the domain of quantification is a set of formulas instead of the elements of a structure. If a connective, say conjunction, is applied to a finite set of formulas or to a set of only two formulas, we write $\bigwedge_{i \leq n} \varphi_i$ and $\varphi \wedge \psi$, respectively. We make a distinction between *free* and *bound* occurrences of variables. An occurrence is free, if it is not in the scope of a quantifier. On the other hand, an occurrence is bound if it is not free.

The set of free variables in an $\mathcal{L}[\tau]$-formula φ, denoted by *Free*(φ) is defined as follows:

(i) If φ is atomic, then all occurrences of variables in φ are free,

(ii) *Free*$(\neg\varphi)$ =*Free*(φ)

(iii) *Free*$(\varphi \wedge \psi)$ =*Free*$(\psi)\cup$*Free*(ψ)

(iv) *Free*$(\exists x_i \psi)$ =*Free*$(\psi) \setminus \{x_i\}$.

When we want to point out explicitly the free variables of a formula, we use the following notation. Let $\bar{x} = (x_1, ..., x_n)$ and $\bar{y} = (y_1, ..., y_m)$ be tuples of variables. The notation $\varphi(x_1, ..., x_n)$ means that at most the members of $\bar{x}$ have free occurrences in φ i.e. $Free(\varphi) \subseteq \{x_i : 1 \leq i \leq n\}$. Furthermore, we sometimes divide the tuple of free variables into two or more parts. In other words, if the variables of φ having free occurrences belong to $\bar{z} = (\bar{x}, \bar{y})$ we may write $\varphi(\bar{x}, \bar{y})$ instead of $\varphi(\bar{z})$. Suppose $\bar{a} = (a_1, ..., a_m)$ is a tuple of elements of a structure and $\varphi(\bar{x}, \bar{y})$ a formula. Then by $\varphi(\bar{x}, \bar{a})$ we mean a formula in which variables in $\bar{y}$ are assigned the corresponding values in $\bar{a}$. We use the notation $\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x})$, if $\mathcal{M}$ satisfies $\varphi(\bar{x})$ when the variables in $\bar{x}$ are assigned the values in $\bar{a}$. This notation is usually more convenient than writing the modifications of assignment corresponding each free variable in the subscript of the satisfaction relation.

It is often useful to think of formulas with free variables as definining relations on the domain of a structure. If $\mathcal{M}$ is a τ-structure, $\varphi(\bar{x}, \bar{y}) \in \mathcal{L}[\tau]$, then we write $\varphi^{\mathcal{M}}(\bar{x}, \bar{b}) = \{\bar{a} \in M^m : \langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x}, \bar{b})\}$ for the *relation defined by* $\varphi(\bar{x}, \bar{y})$ *with parameters* $\bar{b}$ in M. If the tuple $\bar{b}$ is empty, then we speak of the relation defined by $\varphi^{\mathcal{M}}(\bar{x})$ (without parameters).

2.3 Concrete Logics

Unlike in (infinite) model theory, first-order logic has turned out to be far too weak in expressive power for descriptive complexity theory and finite model theory in general. Hence many extensions of first-order logic have been studied in this context. In this subsection we define the concrete logics relevant for our topic. These two logics are first-order logic and the infinitary k-variable logic. More specifically, they serve as a basis for more complex logics we obtain later by introducing generalized quantifiers.

Definition 2.3.1. *First-order logic* is the logic $\mathcal{FO} = (\mathcal{FO}, \mathcal{T}, \mathcal{S}, \mathcal{V}_\infty, \models_{\mathcal{FO}})$, where

- $\mathcal{V}_\infty = \{x_i : i \in \mathbb{N}\}$.

- $\mathcal{FO}[\tau]$, the set of first-order τ-formulas, is defined as the smallest set that satisfies the following conditions:

(i) $Atom[\tau] \subseteq \mathcal{FO}[\tau]$,

(ii) If $\varphi \in \mathcal{FO}[\tau]$, then $\neg\varphi \in \mathcal{FO}[\tau]$,

(iii) If $\varphi, \psi \in \mathcal{FO}[\tau]$, then $(\varphi \wedge \psi) \in \mathcal{FO}[\tau]$,

(iv) If $\varphi \in \mathcal{FO}[\tau]$, then $\exists x_i \varphi \in \mathcal{FO}[\tau]$.

- $\models_{\mathcal{FO}}$ is defined inductively as in Definition 2.2.1.

From the definition of $\mathcal{FO}$ it follows that the number of conjunctions, disjunctions and variables in a formula of $\mathcal{FO}[\tau]$ is always finite for any vocabulary τ. To have a uniform notation among concrete logics we refer to $\mathcal{FO}$ occasionally as $\mathcal{L}_{\omega\omega}$ or $\mathcal{L}^\omega_{\omega\omega}$. The same is not true for logics in general. By contrast, if a logic allows arbitrary conjunctions and disjunctions, we write $\mathcal{L}_{\infty\omega}$. (In the last notation the ω refers to the fact that only finitely long quantifier blocks are allowed). Suppose $k \in \mathbb{N}$. By $\mathcal{L}^k$ we mean the logic obtained from $\mathcal{L}$ with the following restriction: For any $\varphi \in \mathcal{L}^k[\tau]$, *at most* k variables occur in φ. Moreover, for $\mathcal{L}$ we follow this notation and write $\mathcal{L} = \mathcal{L}^\omega = \cup_{k \in \omega} \mathcal{L}^k$.

Definition 2.3.2. The infinitary k-variable logic, denoted by $\mathcal{L}^k_{\infty\omega}$, is the logic $(\mathcal{L}^k_{\infty\omega}, \mathcal{T}, \mathcal{S}, \mathcal{V}_k, \models_{\mathcal{L}^k_{\infty\omega}})$, where

- $\mathcal{V}_k \subseteq \mathcal{V}_\infty$ such that $|\mathcal{V}_k| = k$.

- $\mathcal{L}^k_{\infty\omega}[\tau]$ is defined as follows:

(1) $Atom^k[\tau] \subseteq \mathcal{L}^k_{\infty\omega}[\tau]$, where $Atom^k[\tau] \subseteq Atom[\tau]$ is such that for all $\varphi \in Atom^k[\tau]$, the variables of φ are in $\mathcal{V}_k$,

(2) If $\varphi \in \mathcal{L}^k_{\infty\omega}[\tau]$, then $\neg\varphi \in \mathcal{L}^k_{\infty\omega}[\tau]$,

(3) If $\varphi \in \mathcal{L}^k_{\infty\omega}[\tau]$, then $\exists x_i \varphi \in \mathcal{L}^k_{\infty\omega}[\tau]$ for all $x_i \in \mathcal{V}_k$,

(4) If $\Psi \subseteq \mathcal{L}^k_{\infty\omega}[\tau]$, then $\bigwedge \Psi \in \mathcal{L}^k_{\infty\omega}[\tau]$,

(5) If $\Psi \subseteq \mathcal{L}^k_{\infty\omega}[\tau]$, then $\bigvee \Psi \in \mathcal{L}^k_{\infty\omega}[\tau]$.

- $\models_{\mathcal{L}^k_{\infty\omega}}$ is defined inductively as in Definition 2.2.1.

These are all the concrete logics relevant for us. At this point the reader may wonder why have we omitted variants of *fixed-point logics*, which are logics of great importance for descriptive complexity theory. There are a couple of reasons for this. Firstly, it is a fact proved by Kolaitis and Vardi [11] that the finite variable infinitary logic, $\mathcal{L}^\omega_{\infty\omega}$, is at least as expressive as *least fixed-point-, partial fixed-point- and inflationary fixed-point logics* (for the definitions, see for instance Chapter 8 in [4]). Hence any result concerning upper bound of expressive power of $\mathcal{L}^\omega_{\infty\omega}$ holds also for fixed-point logics. Furthermore, adding generalized quantifiers to these logics does not change the situation. Thus our main result stating that $PTIME$ cannot be captured with k-variable logic enriched with a set of n-ary generalized quantifiers, will immediately hold for fixed-point logics as well.

3 Generalized Quantifiers

3.1 On the History of Generalized Quantifiers

During the first half of the twentieth century first-order logic was *the* logic. Its model-theoretic properties like completeness, compactness and Löwenheim-Skolem property became effective tools for logicians. However, these properties on the other hand mean lack of expressive power in some respects. Many fundamental mathematical concepts, such as infinity or being countably infinite cannot be expressed due to the facts above. Having these difficulties in mind, Mostowski introduced *cardinality quantifiers* in the late 1950's [14]. Thus, in addition to classical quantifiers, he added to $\mathcal{FO}$ quantifiers like "there are infinitely many" and "there exist uncountably many". In such a way one can make minimal extensions to $\mathcal{FO}$ so that sets of different cardinalities can be separated. About ten years later, Lindström generalized this idea to so-called *Lindström quantifiers* [12]. This insightful idea led to vast research on extensions of $\mathcal{FO}$ and the line of study became known as *abstract model theory.* The purpose of this field is to study extended logics and their mutual relations as well as to give abstract characterizations for logics (cf. [1]). For example, Lindström himself proved that, among abstract logics, $\mathcal{FO}$ is the strongest logic with respect to expressive power that has the compactness property and satisfies the downward Löwenheim-Skolem theorem. Hence, any logic stronger than $\mathcal{FO}$ either does not satisfy the compactness property or can separate some infinities from each other.

So the study of generalized quantifiers began in the context of infinite models. Soon after this researchers of finite model theory and theoretical computer science realized their potential for their fields of study. However, the application of generalized quantifiers emerged from different needs compared to the infinite context. Cardinality quantifiers and most of the results on classical model theoretic properties are either meaningless or trivial, when one is interested in finite models. Finite model theory had problems of its own. First-order logic was realized to be far too weak, since it does not have any mechanism for recursion. Thus for instance connectivity of a graph is not expressible in $\mathcal{FO}$ in general. Fixpoint logics overcome this problem, but suffer from other deficits. For example queries concerning parities such as "there is an even number of elements..." are inexpressible in them [3]. In 1980's, Immerman tried to solve this problem by adding *counting quantifiers* to fixed-point logic. He conjectured, that this logic was the right one to capture all $PTIME$

properties on finite structures [8]. But only three years later, Cai et al. showed that this bold conjecture does not hold [2]. It was then a natural idea to try to enrich the expressive power of fixed-point logic with more general quantifiers. However, even fixed-point logic with all unary generalized quantifiers fails in this task (see introduction). As a matter of fact, allowing arbitrary unary quantifiers instead of only counting quantifiers does not give much. This is due to a result obtained by Kolaitis and Väänänen, that for any $k \in \mathbb{N}$, $\mathcal{L}^k_{\infty\omega}$ with all counting quantifiers is equivalent in expressive power with $\mathcal{L}^k_{\infty\omega}$ augmented with all unary quantifiers [10]. This leads one to consider n-ary generalized quantifiers, which is the topic of this thesis and therefore a natural point to end the historical considerations.

3.2 Lindström Quantifiers

We adopt a "bottom-up" approach to reach the general concept of a quantifier. Starting from the classical quantifiers, we generalize them step-by-step and eventually end up with the notion of a Lindström quantifier. To generalize the familiar quantifiers we need to somehow answer the question of what is a quantifier? A plausible answer to this question has to in part explain what is the meaning of quantifiers i.e. what they denote. We observe that the meaning of the two classical quantifiers is most often given in somewhat indirect way. Often the truth conditions for universal and existential quantifiers are given as:

$$\mathcal{M} \vDash \forall x \varphi(x, b_1 ..., b_n) \Leftrightarrow \text{ for all } a \in M, \langle M, a\rangle \vDash \varphi(x, b_1, ..., b_n),$$

$$\mathcal{M} \vDash \exists x \varphi(x, b_1 ..., b_n) \Leftrightarrow \text{ for some } a \in M, \langle M, a\rangle \vDash \varphi(x, b_1, ..., b_n).$$

For practical purposes these definitions are mostly sufficient, but they hardly describe what the quantifiers themselves denote. Recall that a formula with free variables defines a relation $\varphi^{\mathcal{M}}$ over M. In this particular case the relation is actually a set, since classical quantifiers bind only one variable. Hence we can write the previous truth definitions as

$$\mathcal{M} \vDash \forall x \varphi(x, b_1 ..., b_n) \Leftrightarrow \varphi^{\mathcal{M}}(x, b_1, ..., b_n) = M,$$

$$\mathcal{M} \vDash \exists x \varphi(x, b_1 ..., b_n) \Leftrightarrow \varphi^{\mathcal{M}}(x, b_1, ..., b_n) \neq \emptyset.$$

Thus we get natural denotations for the quantifiers. Let

$$\forall_{\mathcal{M}} = \{M\} \text{ and } \exists_{\mathcal{M}} = \{A \subseteq M : A \neq \emptyset\}.$$

Now we can write the former truth-conditions as

$$\mathcal{M} \models \forall x\varphi(x, b_1..., b_n) \Leftrightarrow \varphi^{\mathcal{M}}(x, b_1, ..., b_n) \in \forall_{\mathcal{M}}$$

$$\mathcal{M} \models \exists x\varphi(x, b_1..., b_n) \Leftrightarrow \varphi^{\mathcal{M}}(x, b_1, ..., b_n) \in \exists_{\mathcal{M}}.$$

This accomplishes the first step towards generalized quantifiers. As we have seen, the two traditional quantifiers both denote certain set of subsets of M. With this observation it becomes obvious how to generalize these notions. Let us just call *any* set of subsets of M a quantifier. More specifically, let us call such a quantifier *simple* and *unary*, since it binds *one* variable in *one* formula. We also say that such a quantifier is of *type* $\langle 1 \rangle$. The intuition behind this notion is seen later. So far we have worked in the context of some fixed structure. However, we want generalized quantifiers to satisfy an isomorphism property.

Definition 3.2.1. A *simple unary generalized quantifier* Q is a class of $\{P\}$-structures, which closed under isomorphisms, where P is a unary predicate symbol. More formally, the closure under isomorphisms means that if $\mathcal{M} = \langle M, P^{\mathcal{M}} \rangle \in Q$, $\mathcal{N} = \langle N, P^{\mathcal{N}} \rangle$ and $\mathcal{M} \cong \mathcal{N}$, then $\mathcal{N} \in Q$.

Universal and existential quantifiers are indeed simple unary quantifiers. Let $\mathcal{M}$ be a structure and $R \subseteq M$. Here are some other examples as well:

Existential quantifier: $\exists = \{\langle M, R \rangle : R \neq \emptyset\}$.

Universal quantifier: $\forall = \{\langle M, R \rangle : R = M\}$.

Counting quantifiers: $\exists_i = \{\langle M, R \rangle : |R| \geq i\}$.

Even number of: $\text{EVEN} = \{\langle M, R \rangle : |R| \text{ is even}\}$.

At least half: $\text{HALF} = \{\langle M, R \rangle : |R| \geq M/2\}$.

There are two obvious ways to generalize the quantifiers defined above. We can allow the quantifier to bind more than one variable. This approach leads to *simple n-ary quantifiers*, that is, quantifiers of type $\langle n \rangle$. They refer to sets of n-ary relations of elements of a structure.

Definition 3.2.2. A *simple n-ary generalized quantifier* is a class of $\{P\}$-structures that is closed under isomorphisms, where P is an n-ary relation symbol.

In addition to allowing a quantifier to bind n variables, we can let it apply to several formulas. Hence we end up in quantifiers of type $\langle n_1, ..., n_k \rangle$. They refer to relations of relations of elements of a structure.

Definition 3.2.3. Let $(n_1, ..., n_k) \in \mathbb{Z}_+^k$. A *Lindström quantifier* of type $\langle n_1, ..., n_k \rangle$ is a class of τ-structures that is closed under isomorphisms, where $\tau = \{R_1, ..., R_k\}$ and R_i is n_i-ary for $1 \leq i \leq k$.

We have now established the general notion of a quantifier. For convenience we will occasionally speak merely of quantifiers, when we actually mean Lindström quantifiers. Let $X, Y \subseteq M$. Here are some additional examples of quantifiers:

Härtig quantifier: $I = \{\langle M, X, Y \rangle : |X| = |Y|\}$,

Rescher quantifier: $MORE = \{\langle M, X, Y \rangle : |X| > |Y|\}$,

which are both of type $\langle 1, 1 \rangle$. An example of a quantifier of type $\langle n, n \rangle$ is the quantifier I_n, which is defined as:

$I_n = \{\langle M, X^n, Y^n \rangle : |X^n| = |Y^n|\}$.

3.3 Quantifiers and Logics

In practise we use generalized quantifiers to enrich the expressive power of a logic. We now introduce a new syntactical rule that allows us to add quantifiers to logics. This new rule can be seen as a generalization of the rule for existential quantification in Definition 2.2.1. We use the notation $\mathcal{L}(Q)$ for a logic obtained by adding a quantifier Q to some logic $\mathcal{L}$.

Definition 3.3.1. Let $\mathcal{L}$ be a logic, τ a vocabulary and $\bar{x} = (\bar{x}_1, ..., \bar{x}_k)$ a tuple of tuples of distinct variables such that $|\bar{x}_i| = n_i$ for $1 \leq i \leq k$. If Q is a generalized quantifier of type $\langle n_1, ..., n_k \rangle$ and $\varphi_1(\bar{x}_1), ..., \varphi_k(\bar{x}_k) \in \mathcal{L}(Q)[\tau]$, then $Q\bar{x}(\varphi_1(\bar{x}_1), ..., \varphi_k(\bar{x}_k)) \in \mathcal{L}(Q)[\tau]$.

We adjust the notions of free and bound variables according to this new rule. Consider the notations of the above definition and suppose $x \in \bar{x}_i$. All free occurrences of x in φ_i become bound by Q. Note, however, that x can still remain free in some $\varphi_j, j \neq i$.

Definition 3.3.2. Let $\varphi(\bar{y}) = Q\bar{x}(\varphi_1(\bar{x}_1, \bar{y}_1), ..., \varphi_k(\bar{x}_k, \bar{y}_k) \in \mathcal{L}(Q)[\tau]$. The semantics of φ is defined as:

$$\langle \mathcal{M}, \bar{b} \rangle \vDash \varphi(\bar{y}) \Leftrightarrow \langle M, \varphi_1^{\mathcal{M}}(\bar{x}_1, \bar{b}_1), ..., \varphi_k^{\mathcal{M}}(\bar{x}_k, \bar{b}_k) \rangle \in Q.$$

We can also add several quantifiers or even infinite set of quantifiers to logics. If $\mathbf{Q} = \{Q_i : i \in I\}$ is a set of quantifiers, then by $\mathcal{L}(\mathbf{Q})$ we mean the logic obtained from $\mathcal{L}$ by adding the quantifiers in $\mathbf{Q}$ to $\mathcal{L}$. We denote by $\mathbf{Q}_n$ the class of all quantifiers of arity at most n.

In general one uses quantifiers to enrich the expressive power of a logic. But how are different quantifiers related to each other? For instance, it seems intuitive that the quantifier "At least half", when added to $\mathcal{FO}$, adds more expressive power than "exactly half", since the latter can be expressed as:

Exactly half $(P) \Leftrightarrow$ (At least half $(P) \wedge$ At least half $\neg(P)$).

In such a case it seems that the other quantifier is at least as strong in expressive power as the other. Note, however, that this mutual relation of them depends on the logic, in which they are added to (above we needed the notions of negation and conjunction). The following definition gives a tool for comparing the expressive power of different quantifiers.

Definition 3.3.3. Let $\mathbf{Q}$ be a set of quantifiers and Q a quantifier with vocabulary τ. We say that *Q is $\mathcal{L}$-definable in terms of quantifiers* $\mathbf{Q}$, if there is a sentence $\varphi \in \mathcal{L}(\mathbf{Q})[\tau]$ such that for any τ-model $\mathcal{M}$ it holds that

$$\mathcal{M} \models \varphi \Leftrightarrow \mathcal{M} \in Q.$$

A more general definition of how to compare the expressive power of different logics is exposed in the next section.

4 Descriptive Complexity Theory

4.1 Identifying Logics and Complexity Classes

In this section we study the connection between logical definability and computational complexity. It is assumed that the reader has some prerequisites of elementary concepts in complexity theory. More specifically, the reader is expected to have familiarity with Turing machines and basic complexity classes $PTIME$ and $NPTIME$, as well as how to treat finite structures as inputs of Turing machines.

It is rather straightforward to represent an ordered finite structure by a string, which serves as an input for a Turing machine. However, the same is not true for a finite structure in general. In order to represent an unordered model by a string, we need to impose some ordering on it. Thus a finite model can have different representations as strings. On the other hand, a Turing machine should output the same answer regardless of the chosen representation. There is, however, a way to overcome this difficulty. Informally speaking, one requires that, although the Turing machine uses the chosen ordering, the outcome of the computation is not allowed to depend on the specific choice of the ordering. For a precise treatment of the topics considered above, we instruct to look at Chapter 7 in [4].

From now on we can treat complexity classes such as $PTIME$ as collections of queries on finite structures. We establish a link between computational queries and logics. Our goal is to identify queries as formulas and, more generally, complexity classes as logics. With this approach it becomes possible to relate the expressive power of logics to complexity classes. We begin by recognizing queries as formulas and vice versa. This trick requires a formal description of a query.

Definition 4.1.1. Let τ be a vocabulary, $\mathcal{M}$ and $\mathcal{N}$ τ-structures and $k \in \mathbb{Z}_+$. Let q be a function such that $\mathcal{M} \mapsto q(\mathcal{M})$, where $q(\mathcal{M}) \in \mathcal{P}(M^k)$. If it holds for all $\mathcal{M}, \mathcal{N}$ and $\pi : \mathcal{M} \cong \mathcal{N}$, that $\pi : \langle M, q(\mathcal{M})\rangle \cong \langle N, q(\mathcal{N})\rangle$, then we say that q is a *k-ary query* on τ-structures.

Intuitively k-ary queries are functions that take a structure $\mathcal{M}$ of some fixed vocabulary τ as argument and evaluate which k-tuples of M satisfy the property that the query asks for. The above requirement concerning the isomorphism π is there to guarantee that a query outputs the same answer for isomorphic structures. We can extend the definition of k-ary queries also to 0-ary queries. Instead of giving the k-tuples of domain of a structure that have some property, these queries ask if a given structure itself satisfies some property or not.

Definition 4.1.2. Let τ be a vocabulary and $\mathcal{M}$ and $\mathcal{N}$ τ-structures. A *boolean query* on τ-structures is a function $q : Str(\tau) \to \{0,1\}$ such that $q(\mathcal{M}) = q(\mathcal{N})$, if $\mathcal{M}$ and $\mathcal{N}$ are isomorphic.

As before, also a boolean query has to give the same answer for isomorphic structures. Of course the converse does not hold in general; two non-isomorphic structures may both satisfy the property that query is about. Boolean queries are the most interesting queries for us, since they describe properties of the structure itself as a whole. They divide classes of structures of some vocabulary τ into two parts: those which have the property that the query is about and those which do not. Therefore we can identify a boolean query on τ as $\{\mathcal{M} : q(\mathcal{M}) = 1\}$; the subclass of τ-structures whose elements satisfy the query.

Definition 4.1.3. Let $\mathcal{L}$ be a logic, τ a vocabulary and q a k-ary query on τ-structures. We say that q is *definable in* $\mathcal{L}$, if there is $\varphi(x_1, ..., x_k) \in \mathcal{L}[\tau]$ such that $q(\mathcal{M}) = \varphi^{\mathcal{M}}(x_1, ..., x_k)$ for any τ-structure $\mathcal{M}$. Moreover, a boolean query is definable in $\mathcal{L}$, if there is a sentence $\varphi \in \mathcal{L}[\tau]$ such that $q(\mathcal{M}) = 1$ exactly when $\mathcal{M} \vDash \varphi$.

With the help of the previous definition it is possible to compare a logic to a complexity class. More specifically, we can state upper and lower bounds for the expressive power of a logic. Let $\mathcal{L}$ be a logic and X some complexity class such that there is $q \in X$, which is not definable in $\mathcal{L}$. Therefore $\mathcal{L}$ is not strong enough to capture X. Similarly if a query is definable in $\mathcal{L}$, but does not belong to X, we conclude that $\mathcal{L}$ is too strong for X. There is still an obvious problem, if we want to obtain more positive result. Assume we have shown that every query of some complexity class is definable in a logic. We then know that the logic is powerful enough for the complexity class, but how to decide, if it is too powerful or not? We have to be able to convert the formulas of the logic to queries.

Suppose $\varphi(\bar{x}) \in \mathcal{L}[\tau]$ and define $q_\varphi(\mathcal{M}) = \varphi^{\mathcal{M}}(\bar{x})$. Now q_φ is a query, since by Definition 2.1.1, the satisfaction relation of a logic is invariant under isomorphisms. Thus every formula φ determines a *canonical query* q_φ. On the other hand, definition 4.1.3 gave us a way to convert a query to a formula of an appropriate logic. Therefore we are able to compare logics and complexity classes in both directions. Obviously if φ has k free variables, q_φ is k-ary, whereas sentences correspond to boolean queries.

Since formulas determine canonical queries, we are able to identify formulas as queries. In the following definitions we take advantage of this observation. Next we define a concept related to the expressive power of a logic.

Definition 4.1.4. Let $\mathcal{L}$ be a logic and $\mathcal{M}$ and $\mathcal{N}$ τ-structures. $\mathcal{M}$ and $\mathcal{N}$ are *$\mathcal{L}$-equivalent*, denoted by $\mathcal{M} \equiv_{\mathcal{L}} \mathcal{N}$, if and only if for every boolean query that is definable in $\mathcal{L}$, it holds that $q(\mathcal{M}) = q(\mathcal{N})$. Similarly, if $\bar{a} \in M^k$ and $\bar{b} \in B^k$, we write $\langle \mathcal{M}, \bar{a} \rangle \equiv_{\mathcal{L}} \langle \mathcal{N}, \bar{b} \rangle$ if and only if $\bar{a} \in q_\varphi(\mathcal{M}) \Leftrightarrow \bar{b} \in q_\varphi(\mathcal{N})$ for every $\varphi \in \mathcal{L}[\tau]$.

The previous definition gives an obvious way to recognize deficits in expressive power of a logic. Take two finite structures of the same vocabulary that are non-isomorphic. If they are, however, $\mathcal{L}$-equivalent for some logic $\mathcal{L}$, then we know that the distinctive features of the structures cannot be expressed in $\mathcal{L}$. This observation will play a key role in our main result. We can also compare the expressive powers of logics with each other.

Definition 4.1.5. Let $\mathcal{L}$ and $\mathcal{L}'$ be logics on some class of structures $\mathcal{K}$. We say that $\mathcal{L}$ is *at most as expressive as $\mathcal{L}'$ over $\mathcal{K}$*, denoted by $\mathcal{L} \leq_{\mathcal{K}} \mathcal{L}'$, if all queries q that are definable in $\mathcal{L}$ are definable in $\mathcal{L}'$. Furthermore, if $\mathcal{L} \leq_{\mathcal{K}} \mathcal{L}'$ and $\mathcal{L}' \leq_{\mathcal{K}} \mathcal{L}$, we write $\mathcal{L} \equiv_{\mathcal{K}} \mathcal{L}'$ and say that $\mathcal{L}$ and $\mathcal{L}'$ are *equivalent in expressive power over $\mathcal{K}$*.

Initially we defined generalized quantifiers as model classes, which are closed under isomorphisms. When it comes to computational complexity, generalized quantifiers can also be naturally identified as queries. Suppose Q is a quantifier of type $\langle n_1, ..., n_k \rangle$ and $\varphi(\bar{y}) = Q\bar{x}(\varphi_1(\bar{x}_1, \bar{y}_1), ..., \varphi_k(\bar{x}_k, \bar{y}_k)) \in \mathcal{L}[\tau]$. Moreover, let q_Q be the boolean query corresponding to Q i.e. given a finite model $\mathcal{M}$ and the relations defined by $\varphi_1, ..., \varphi_k$, the query q_Q evaluates whether the model formed from the relations is in Q or not. Consequently, we can extend the truth definition of formulas with quantifiers (in definition 3.3.2) to

$$\langle \mathcal{M}, \bar{b} \rangle \vDash \varphi(\bar{y}) \Leftrightarrow \langle M, \varphi_1^{\mathcal{M}}(\bar{x}_1, \bar{b}_1), ..., \varphi_k^{\mathcal{M}}(\bar{x}_k, \bar{b}_k) \rangle \in Q \Leftrightarrow q_Q(\mathcal{M}') = 1,$$

where $\mathcal{M}' = \langle M, \varphi_1^{\mathcal{M}}(\bar{x}_1, \bar{b}_1), ..., \varphi_k^{\mathcal{M}}(\bar{x}_k, \bar{b}_k) \rangle$.

We have now developed fully the framework concerning the identification of complexity classes and logics. Firstly, we observed that any formula of any logic gives rise to a canonical query. Moreover, we defined what it means that a given query is definable in a logic. Hence we can think of the expressive power of a logic as exactly the queries definable in it. This observation immediately gave us a way of comparing the expressive power of logics with each other. Furthermore, in a similar fashion we can compare logics with complexity classes, which are nothing but classes of queries. When a logic is equivalent in expressive power with some complexity class, we use a more natural vocabulary and say that the logic *captures* some particular complexity class.

4.2 Can $PTIME$ Be Captured Effectively?

Let us now concentrate on a very interesting open problem in descriptive complexity theory. It is not known whether $PTIME$ can be captured by a logic in the class $\mathcal{F}$ in a reasonable way. Such a characterization should not only be for ordered structures, but for all finite structures. Furthermore, it should be effective. By effective we mean certain restrictions on the computational hardness of satisfaction relation as well as the construction of formulas. Many weaker results have been obtained in some particular subclasses of $\mathcal{F}$. Most importantly, $PTIME$ has been characterized in the class of all ordered structures [8, 15]. However, the presence of a linear order plays a crucial part in their proofs. In the spirit of the last subsection we define the logic of $PTIME$-properties.

Definition 4.2.1. The logic of $PTIME$-properties on finite structures is the abstract logic $\mathcal{PTIME} = (\mathcal{PTIME}, \mathcal{T}, \mathcal{F}, \mathcal{V}_\infty, \models_{\mathcal{PTIME}})$, where

- $\mathcal{PTIME}[\tau]$ contains those queries q on finite τ-structures that are $PTIME$-computable.

- $\models_{\mathcal{PTIME}}$ is defined as $\mathcal{M} \models_{\mathcal{PTIME}} q \Leftrightarrow q(\mathcal{M}) = 1$.

The question of characterizing $PTIME$ without any effectiveness condition is nonsensical. We could answer this question affirmatively by stating that the logic $\mathcal{PTIME}$ succeeds in it. But this answer is of course insufficient, since we would not know anything concrete about the syntax or the semantics of the proposed logic, let alone any effective descriptions of them. These observations among others motivated Gurevich to formulate the question of capturing $PTIME$ in an exact mathematical way. In order to do so, we introduce the concept of a *Gurevich logic* [6].

Definition 4.2.2. Let τ be a finite vocabulary. We say that $\mathcal{L}$ is a *Gurevich logic*, if it satisfies the following conditions:

(i) $\mathcal{L}[\tau]$ is recursive.

(ii) There is an effective procedure, which assigns to any $\varphi \in \mathcal{L}$ a Turing machine T_φ and a polynomial P such that T_φ, given some finite structure $\mathcal{M}$ as input, computes the query $q_\varphi(\mathcal{M})$ in time $t \leq P(|M|)$.

We can now state the interesting question about finding a reasonable characterization of $PTIME$ as:

Problem 4.2.3. *Is there a Gurevich logic $\mathcal{L}$ such that $\mathcal{L} \equiv \mathcal{PTIME}$?*

An answer, no matter positive or negative, to this question would be a major result in descriptive complexity theory. Gurevich himself conjectured, that the answer is negative [6]. As a matter of fact, such negative result would imply that $PTIME \neq NPTIME$ (see [4], pp.291-292).

5 Expressive Power of $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$

Ehrenfeucht-Fraïsse games remain a useful tool also in the model theory of finite structures. It is well known that the expressive power of infinitary k-variable logics $\mathcal{L}^k_{\infty\omega}$ can be characterized by certain k-pebble games. A similar characterization is possible for the corresponding logics enriched with any set of n-ary quantifiers. In [10] Kolaitis and Väänänen introduce pebble games that extend the basic game by adding extra rules for the additional quantifiers. Thus they obtain a game-theoretic way to decide whether two finite structures are $\mathcal{L}^k_{\infty\omega}(\mathbf{Q})$-equivalent. However, we will use the n-bijective k-pebble game introduced in [7]. The game is a bit more abstract, but the price is worth paying in order to cover all quantifiers in one game. We will first define back-and-forth systems of partial isomorphisms and afterwards show that a partial function preserves the truth of $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$ if and only if it belongs to this system. Then we show that the similar result can be obtained with the mentioned game-theoretic formulation.

5.1 Back-and-Forth Systems of Partial Isomorphisms

From now on we assume all structures are finite and relational i.e. their vocabularies consist of relation symbols only.

Definition 5.1.1. Let $\mathcal{L}$ be a logic and $\mathcal{M}$ and $\mathcal{N}$ τ-structures. We say that a partial function $p : M \to N$ is a *partial* $\mathcal{L}[\tau]$*-embedding* (or just $\mathcal{L}$-embedding, if τ is clear from the context), if p is injective and for any tuple $\bar{a} \in dom(p)$ and any $\varphi \in \mathcal{L}[\tau]$ it holds that

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x}) \Leftrightarrow \langle \mathcal{N}, p(\bar{a}) \rangle \vDash \varphi(\bar{x}).$$

Furthermore, we say that p is a *partial isomorphism*, if the same is true for all $\varphi \in Atom[\tau]$.

Denote by $Part^k(\mathcal{M},\mathcal{N})$ the set of all partial isomorphisms p from $\mathcal{M}$ to $\mathcal{N}$ such that $|p| \leq k$. Similarly, we write simply $Emb^k(\mathcal{M},\mathcal{N})$ for the set of all partial $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-embeddings of size at most k. In order to draw a distinction between embeddings of the two logics of interest, we write $Emb^k_{\mathcal{FO}}(\mathcal{N},\mathcal{M})$ for the set of partial $\mathcal{FO}(\mathbf{Q}_n)$-embeddings. It follows straightforwardly from the definitions that

$$Emb^k(\mathcal{M},\mathcal{N}) \subseteq Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}) \subseteq Part^k(\mathcal{M},\mathcal{N}).$$

Also the converse is actually true for the first inclusion and this result is obtained later. Intuitively this is clear, because if there was $p \in Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N}) \setminus Emb^k(\mathcal{M}, \mathcal{N})$, an $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formula that is not preserved should contain infinitary connectives. However, then by proceeding inductively we would find a subformula which is not preserved and belongs to $\mathcal{FO}^k$, a contradiction. When it comes to the latter inclusion the converse does not hold in general. We do not even need any exotic quantifiers to show it, as is seen in the next simple example.

Example 5.1.2. There are τ-structures $\mathcal{M}, \mathcal{N}$ and $k \in \mathbb{Z}_+$ such that $Part^k(\mathcal{M}, \mathcal{N}) \nsubseteq Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N})$.

Proof. Let $\tau = \{<\}$ and $\mathcal{M} = (\{0, 1, 2\}, <^{\mathcal{M}})$, where $<^{\mathcal{M}}$ is the natural ordering. Let $p \in Part^2(\mathcal{M}, \mathcal{M})$ be such that $p(0) = 0$ and $p(1) = 2$. Furthermore, let

$$\varphi(x_1, x_2) = \exists x_3(x_1 < x_3) \wedge (x_3 < x_2).$$

The formula φ gives us the desired result, since it holds that

$$\langle \mathcal{M}, 0, 1\rangle \nvDash \varphi(x_1, x_2), \text{ but } \langle \mathcal{M}, p(0), p(1)\rangle \vDash \varphi(x_1, x_2).$$

□

So the set $Part^k(\mathcal{M}, \mathcal{N})$ may contain partial isomorphisms which are not embeddings. The reason for this is the fact that the models can differ outside the domain of the partial isomorphism. Therefore the truth of a quantified formula may not be preserved, since the witnessing elements can be found outside the domain of p, where it is possible that the models look very different. We can overcome this problem by requiring that the partial isomorphisms are extendable in a suitable way to $\mathrm{dom}(\mathcal{M}) \setminus \mathrm{dom}(p)$. It is well known what this means in practise for the logic $\mathcal{L}^k_{\infty\omega}$. One can either construct a back-and-forth system of partial isomorphisms of size at most k or use a k-pebble game between structures $\mathcal{M}$ and $\mathcal{N}$. These approaches yield equivalent conditions. Our job is to generalize these ideas in order to find a criterion so that the partial isomorphisms preserve the truth also for formulas containing arbitrary n-ary quantifiers. We start with the generalization of the former approach.

Definition 5.1.3. The sequence of *canonical k-variable n-bijective back-and-forth sets*, $(I^k_m(\mathcal{M}, \mathcal{N}))_{m \in \omega}$, is defined by recursion on m as follows:

(i) $I^k_0(\mathcal{M}, \mathcal{N}) = Part^k(\mathcal{M}, \mathcal{N})$,

(ii) $I^k_{m+1}(\mathcal{M},\mathcal{N}) = \{p \in I^k_m(\mathcal{M},\mathcal{N}) :$ there is a *nice* bijection $f_p : M \to N\}$.

Let $X \subseteq Part^k(\mathcal{M},\mathcal{N})$ and $p \in X$. A bijection f_p is *nice* (for the set X), if it holds that

$$(p \restriction C) \cup (f_p \restriction D) \in X,$$

whenever $C \subseteq \mathrm{dom}(p)$, $D \subseteq M$, $|D| \leq n$ and $|C \cup D| \leq k$. Moreover, if for every $p \in X$ there is a nice bijection, we say that X satisfies the *bijective extension condition.*

The *canonical k-variable n-bijective back-and-forth system* between $\mathcal{M}$ and $\mathcal{N}$, denoted by $I^k(\mathcal{M},\mathcal{N})$, is defined as the intersection $\bigcap_{m\in\omega} I^k_m(\mathcal{M},\mathcal{N})$. The next lemma shows that the system satisfies the bijective extension condition.

Lemma 5.1.4. *$I^k(\mathcal{M},\mathcal{N})$ is the largest subset of $Part^k(\mathcal{M},\mathcal{N})$, which satisfies the bijective extension condition.*

Proof. Since $\mathcal{M}$ and $\mathcal{N}$ are finite, there are only finitely many partial isomorphisms between them. Moreover, by Definition 5.1.3

$$Part^k(\mathcal{M},\mathcal{N}) \supseteq I^k_m(\mathcal{M},\mathcal{N}) \supseteq I^k_{m+1}(\mathcal{M},\mathcal{N}),$$

for all $m \in \mathbb{N}$. Hence there is $l \in \mathbb{N}$ such that

$$I^k_l(\mathcal{M},\mathcal{N}) = I^k(\mathcal{M},\mathcal{N}),$$

because otherwise we would have an infinite descending chain of natural numbers. By the construction of the canonical back-and-forth sets, $I^k_l(\mathcal{M},\mathcal{N})$ satisfies the bijective extension condition. Furthermore, if

$$I^k(\mathcal{M},\mathcal{N}) \subset X \subseteq Part^k(\mathcal{M},\mathcal{N}),$$

then there is $p \in X$ for which there is no nice bijection and therefore X does not satisfy the bijective extension condition.

□

In the next two lemmas we prove inclusions which together show that the bijective extension condition is the right one to pick out precisely the largest subset of $Part^k(\mathcal{M},\mathcal{N})$, which has the property that its elements preserve the truth of all formulas of infinitary k-variable logics with n-ary quantifiers.

Lemma 5.1.5. *$I^k(\mathcal{M},\mathcal{N}) \subseteq Emb^k(\mathcal{M},\mathcal{N})$.*

Proof. We prove by induction on the structure of $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formulas that every $p \in I^k(\mathcal{M},\mathcal{N})$ is an $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-embedding. Let $p \in I^k(\mathcal{M},\mathcal{N})$. The initial step is clear, because if $\varphi \in Atom$ and $p \in I^k(\mathcal{M},\mathcal{N})$, then

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x}) \Leftrightarrow \langle \mathcal{N}, p(\bar{a}) \rangle \vDash \varphi(\bar{x}),$$

because $I^k(\mathcal{M},\mathcal{N}) \subseteq Part^k(\mathcal{M},\mathcal{N})$. The induction steps for negation and connectives are rather trivial. Thus only the case regarding conjunctions is treated explicitly. Let Ψ be a collection of $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formulas and suppose

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \bigwedge_{\varphi \in \Psi} \varphi.$$

By definition, $\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x})$ for any $\varphi \in \Psi$. The induction assumption implies that $\langle \mathcal{N}, p(\bar{a}) \rangle \vDash \varphi(\bar{x})$ and hence

$$\langle \mathcal{N}, \bar{a} \rangle \vDash \bigwedge_{\varphi \in \Psi} \varphi.$$

The last step concerning quantifiers is a bit more involved. Suppose $Q \in \mathbf{Q}_n$ is of type $\langle n_1, ..., n_l \rangle$ and

$$\varphi(\bar{y}) = Q\bar{x}(\varphi_1(\bar{x}_1, \bar{y}_1), ..., \varphi_l(\bar{x}_l, \bar{y}_l),$$

where $\bar{y} = (\bar{y}_1, ..., \bar{y}_l)$. Let $\bar{b} = (\bar{b}_1, ..., \bar{b}_l)$ be a tuple of elements of $\mathrm{dom}(p)$, such that the length of $\bar{b}_i$ is the same as the length of $\bar{y}_i$ for every $i, 1 \leq i \leq l$. It holds that $p \in I^k(\mathcal{M},\mathcal{N})$, which satisfies the bijective extension condition. Hence there is a nice bijection $f_p : M \to N$. Since Q is at most n-ary, for all i and $\bar{a}_i \in M^{n_i}$ it holds that $|\bar{a}_i| \leq n$. Furthermore, φ is a formula of the k-variable logic and therefore $|\bar{b}_i \cup \bar{a}_i| \leq |\bar{x}_i \cup \bar{y}_i| \leq k$, for all i. These observations guarantee that

$$(p \upharpoonright \bar{b}_i) \cup (f_p \upharpoonright \bar{a}_i) \in I^k(\mathcal{M},\mathcal{N}).$$

The induction assumption implies that for each i it holds that

$$\langle \mathcal{M}, \bar{a}_i, \bar{b}_i \rangle \vDash \varphi_i(\bar{x}_i, \bar{y}_i) \Leftrightarrow \langle \mathcal{N}, f_p(\bar{a}_i), p(\bar{b}_i) \rangle \vDash \varphi_i(\bar{x}_i, \bar{y}_i).$$

Consequently we get an isomorphism

$$f : \big(M, \varphi_1^{\mathcal{M}}(\bar{x}_1, \bar{b}_1), ..., \varphi_l^{\mathcal{M}}(\bar{x}_l, \bar{b}_l)\big) \cong \big(N, \varphi_1^{\mathcal{N}}(\bar{x}_1, p(\bar{b}_1)), ..., \varphi_l^{\mathcal{N}}(\bar{x}_l, p(\bar{b}_l))\big).$$

Since Q is closed under isomorphisms, it holds that

$$\langle \mathcal{M}, \bar{b} \rangle \vDash \varphi(\bar{y}) \text{ if and only if } \langle \mathcal{N}, p(\bar{b}) \rangle \vDash \varphi(\bar{y}).$$

□

The previous lemma tells us that all the partial isomorphisms in $I^k(\mathcal{M},\mathcal{N})$ preserve the truth of $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formulas. It also holds that any partial isomorphism preserving the truth of $\mathcal{FO}^k(\mathbf{Q}_n)$-formulas belongs to $I^k(\mathcal{M},\mathcal{N})$.

Lemma 5.1.6. *$Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}) \subseteq I^k(\mathcal{M},\mathcal{N})$.*

Proof. Recall that the canonical back-and-forth system is defined as

$$I^k(\mathcal{M},\mathcal{N}) = \bigcap_{m\in\omega} I^k_m(\mathcal{M},\mathcal{N}).$$

Hence it is enough to show that for all $m \in \mathbb{N}$ it holds that

$$Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}) \subseteq I^k_m(\mathcal{M},\mathcal{N}).$$

We prove this by induction on m. For the initial step, suppose $m = 0$. By definition $I^k_0(\mathcal{M},\mathcal{N}) = Part^k(\mathcal{M},\mathcal{N})$ and therefore any $p \in Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N})$ is also in $I^k_0(\mathcal{M},\mathcal{N})$, because every $\mathcal{FO}$-embedding is in particular a partial isomorphism. Suppose that the claim holds for m. Towards a contradiction, assume there is

$$p \in Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}) \setminus I^k_{m+1}(\mathcal{M},\mathcal{N}).$$

Let $\bar{b}$ be a tuple that contains every element of $dom(p)$ exactly once. By assumption, there is no nice bijection f_p and thus for any bijection $f : M \to N$ we can find $\bar{b}_f$, a subtuple of $\bar{b}$ and $\bar{a}_f \in M^l, l \leq n$, such that

$$(p \upharpoonright \bar{b}_f) \cup (f \upharpoonright \bar{a}_f) \notin I^k_m(\mathcal{M},\mathcal{N}).$$

By induction assumption $Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}) \subseteq I^k_m(\mathcal{M},\mathcal{N})$. Hence

$$(p \upharpoonright \bar{b}_f) \cup (f \upharpoonright \bar{a}_f) \notin Emb^k_{\mathcal{FO}}(\mathcal{M},\mathcal{N}),$$

and therefore there is an $\mathcal{FO}^k(\mathbf{Q}_n)$-formula $\varphi_f(\bar{x}_f, \bar{y}_f)$ such that

$$\langle \mathcal{M}, \bar{b}_f, \bar{a}_f \rangle \vDash \varphi_f(\bar{x}_f, \bar{y}_f),$$

but

$$\langle \mathcal{N}, p(\bar{b}_f), f(\bar{a}_f) \rangle \nvDash \varphi_f(\bar{x}_f, \bar{y}_f).$$

Since M and N are finite, we can list all the bijections from M to N as $(f_1, ..., f_r), r \in \mathbb{N}$. We know that for some $s, 1 \leq s \leq r$, it holds that $f_s = f$. This means that

$$\langle M, \varphi^{\mathcal{M}}_{f_1}(\bar{x}_{f_1}, \bar{b}_{f_1}), ..., \varphi^{\mathcal{M}}_{f_r}(\bar{x}_{f_r}, \bar{b}_{f_r}) \rangle \ncong \langle N, \varphi^{\mathcal{N}}_{f_1}(\bar{x}_{f_1}, p(\bar{b}_{f_1})), ..., \varphi^{\mathcal{N}}_{f_r}(\bar{x}_{f_1}, p(\bar{b}_{f_r})) \rangle.$$

Let Q be a quantifier containing the structure $\langle M, \varphi_{f_1}^{\mathcal{M}}(\bar{x}_{f_1}, \bar{b}_{f_1}), ..., \varphi_{f_r}^{\mathcal{M}}(\bar{x}_{f_r}, \bar{b}_{f_r})\rangle$, but not $\langle N, \varphi_{f_1}^{\mathcal{N}}(\bar{x}_{f_1}, p(\bar{b}_{f_1})), ..., \varphi_{f_r}^{\mathcal{N}}(\bar{x}_{f_1}, p(\bar{b}_{f_r}))\rangle$. For each s, it holds that

$$|\bar{x}_{f_s}| = |\bar{a}_{f_s}| \leq n.$$

We conclude that Q is n-ary, and consequently

$$\varphi(\bar{y}_{f_1}, ..., \bar{y}_{f_r}) = Q\bar{x}(\varphi_{f_1}(\bar{x}_{f_1}, \bar{y}_{f_1}), ..., \varphi_{f_r}(\bar{x}_{f_r}, \bar{y}_{f_r})$$

is an $\mathcal{FO}^k(\mathbf{Q}_n)$-formula, for which

$$\langle \mathcal{M}, \bar{b}\rangle \vDash \varphi(\bar{y}), \text{ but } \langle \mathcal{N}, p(\bar{b})\rangle \nvDash \varphi(\bar{y}),$$

which contradicts the assumption that $p \in Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N})$.

□

We are now ready to gather the results obtained so far together. We get a characterization for equivalence with respect to infinitary k-variable logics with n-ary quantifiers.

Theorem 5.1.7. *Let $\bar{a}$ be a tuple of elements of M with length at most k and $p : M \to N$ a partial function such that $dom(p) = \bar{a}$. The following conditions are equivalent:*

(i) $p \in I^k(\mathcal{M}, \mathcal{N})$

(ii) $\langle \mathcal{M}, \bar{a}\rangle \equiv_{\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)} \langle \mathcal{N}, p(\bar{a})\rangle$

(iii) $\langle \mathcal{M}, \bar{a}\rangle \equiv_{\mathcal{FO}^k(\mathbf{Q}_n)} \langle \mathcal{N}, p(\bar{a})\rangle$.

Proof. From lemmas 5.1.5 and 5.1.6 we obtain that

$$I^k(\mathcal{M}, \mathcal{N}) \subseteq Emb^k(\mathcal{M}, \mathcal{N}) \text{ and } Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N}) \subseteq I^k(\mathcal{M}, \mathcal{N}).$$

We observe that the inclusion $Emb^k(\mathcal{M}, \mathcal{N}) \subseteq Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N})$ holds, since every $\mathcal{FO}^k[\tau]$-formula is also $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)[\tau]$-formula and thus any $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-embedding is especially an $\mathcal{FO}(\mathbf{Q}_n)$-embedding. Consequently, it holds that

$$I^k(\mathcal{M}, \mathcal{N}) = Emb^k(\mathcal{M}, \mathcal{N}) = Emb^k_{\mathcal{FO}}(\mathcal{M}, \mathcal{N}).$$

It follows immediately that the conditions (i), (ii) and (iii) are equivalent.

□

5.2 The k-Pebble n-Bijective Game $BP_n^k(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$

The previous characterization is based on particular back-and-forth sets of partial isomorphisms. We can, however, get a similar result with a game theoretic approach. It is well known that in the absence of generalized quantifiers, k-pebble games can be used to characterize equivalence with respect to k-variable logics. We next generalize those games to *n-bijective k-pebble games* and show that they are appropriate for the k-variable logics with n-ary quantifiers. Those games are based heavily on the result characterizing the equivalence of k-variable logics with generalized quantifiers using the back-and-forth systems $I^k(\mathcal{M}, \mathcal{N})$.

The game is played between two players known as player I and player II. The latter claims that two given models, $\mathcal{M}$ and $\mathcal{N}$, are equivalent whereas player I tries to refute the claim. The game starts from a partial isomorphism p between models and player II tries to find a nice bijection to extend p . Player II answers to this move by giving a subset of the domain of the partial isomorphism and a subset of M with size restrictions based on the amount of variables allowed and the maximum arity of quantifiers. Player I wins the game, if he succeeds in finding the subsets so that a new partial function determined by player's choises is not a partial isomorphism, and otherwise the game continues with player II's next move and so on. If player I does not win the game in finite number of moves, player II is the winner.

Intuitively one can think of the two players labeling elements of the models by placing pebbles on them. This is the reason for the name adopted. When player II chooses a bijection, she pairs every element of M with some element of N. Then player I chooses the two subsets to indicate which pebbles are kept on the "board" and all other pebbles are removed. If the two substructures determined by pebbles on the board are non-isomorphic, player I wins. The maximum number of pebbles held on the table corresponds to the number of variables allowed and the number of these pebbles outside the domain of the previous partial isomorphism is limited by the maximum arity of quantifiers. One can think that the number of pebbles held on the table is restricted by player I's limited ability to recall what was the pair of each element in the other structure. Complexity theoretically this intuition corresponds to restrictions in space (or memory) resources. We next define the game more rigorously.

Definition 5.2.1. The *n-bijective k-pebble game*, denoted by $BP_n^k(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$, is the following two-player game.

- The initial history of the game is the partial function p_0 determined by tuples $\bar{a} \in M^l$ and $\bar{b} \in N^l$, $l \leq k$, i.e. $p_0 = \{(a_1, b_1), ..., (a_l, b_l)\}$.

- On round $i \geq 1$, player II chooses a bijection $f_i : M \to N$ and player I answers this move by picking sets $C_i \subseteq \text{dom}(p_i)$ and $D_i \subseteq M$ such that $|C_i \cup D_i| \leq k$ and $D_i \leq n$. These choices determine a new partial function $p_i = (p_{i-1} \upharpoonright C_i) \cup (f_i \upharpoonright D_i)$.

- If p_i is not a partial isomorphism, then player I wins the game. Otherwise the game proceeds to round $i + 1$.

- If there is no $i \in \mathbb{N}$ such that player I wins the game on round i, then player II wins the game.

It is now possible to establish a link between n-bijective k-pebble game and the back-and-forth system $I^k(\mathcal{M}, \mathcal{N})$ of partial isomorphisms.

Lemma 5.2.2. *Player II has a winning strategy in* $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ *if and only if* $p_0 \in I^k(\mathcal{M}, \mathcal{N})$.

Proof. Suppose player II has a winning strategy σ in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$. On every round i, $i \in \omega$, she can find a bijection f_i according to σ such that $p_i \in Part^k(\mathcal{M}, \mathcal{N})$. Play of the game in which player II follows σ creates a sequence P_i of sets of partial isomorphisms each containing p_0, where the sets P_i are defined recursively as

$$P_0 = \{p_0\} \text{ and } P_{i+1} = P_i \cup \{p_{i+1}\}.$$

We observe that $\bigcap_{i \in \omega} P_i = \{p_0\}$. Now $p_0 \in \{p_0\}$ and $\{p_0\}$ satisfies the bijective extension condition. By lemma 5.1.4, $\{p_0\} \subseteq I^k(\mathcal{M}, \mathcal{N})$ and hence $p_0 \in I^k(\mathcal{M}, \mathcal{N})$. For the other direction, assume $p_0 \in I^k(\mathcal{M}, \mathcal{N})$. Consider the following inductively defined strategy σ of player II: At round 1, she chooses a nice bijection f_1 (which is possible, since $p_0 \in I^k(\mathcal{M}, \mathcal{N})$) and hence no matter what player I chooses for C_1 and D_1, the function p_1 belongs to $I^k(\mathcal{M}, \mathcal{N})$. At round j, player II plays similarly, which is possible by the induction assumption. Thus for all $i \in \omega$, p_i belongs to $I^k(\mathcal{M}, \mathcal{N}) \subseteq Part^k(\mathcal{M}, \mathcal{N})$ and σ is a winning strategy for player II.

□

This lemma gives straightforwardly an alternative characterization for $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-equivalence.

Theorem 5.2.3. *The following conditions are equivalent:*

(i) Player II has a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$*,*

(ii) $\langle \mathcal{M}, \bar{a} \rangle \equiv_{\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)} \langle \mathcal{N}, p(\bar{a}) \rangle$,

(iii) $\langle \mathcal{M}, \bar{a} \rangle \equiv_{\mathcal{FO}^k(\mathbf{Q}_n)} \langle \mathcal{N}, p(\bar{a}) \rangle$.

Proof. The result follows immediately from Theorem 5.1.7 and Lemma 5.2.2.

□

Note that we did not need the determinacy of the game $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ in the proof of the above theorem. However, the game is indeed determined and we need this fact later. This result would follow from the Gale-Stewart theorem, which asserts that every two-player ω-closed game is determined. (A game G is *ω-closed*, if player II wins every play of G, in which she has not lost already at some round $i \in \omega$). However, a weaker result suffices, essentially because the structures we consider are finite. This feature allows us to define a modified version of $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$, which is a game of finite length, but equivalent with the original game with respect to existences of winning strategies.

Definition 5.2.4. The *finite n-bijective k-pebble game*, $FBP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$, is defined exactly as $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$, except that the winning condition for player II is:

- Let $i < j$. If

$$(p_{i-1}, f_i, C_i, D_i) = (p_{j-1}, f_j, C_j, D_j),$$

then player II wins the game.

So player I cannot respond to player II's similar move exactly the same way he did at some point earlier. Intuitively the idea is that if player I has a winning strategy in the original game, then he has a winning strategy in which no repetetition of game positions is allowed. Moreover, since M and N are finite, there are only finitely many different quadruples (p_{i-1}, f_i, C_i, D_i). Hence $FBP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ is a game of finite length.

Lemma 5.2.5. *(i) Player I has a winning strategy in* $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ *if and only if he has a winning strategy in* $FBP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$

(ii) Player II has a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ *if and only if she has a winning strategy in* $FBP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$

Proof. Suppose I has a winning strategy σ in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$. Without loss of generality, assume that some play of the game $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$, in which σ is followed, contains a repetition of game positions (otherwise σ gives a winning strategy in $FBP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b}))$. Put differently, assume for some $i<j$ it holds that

$$(p_{i-1},f_i,C_i,D_i)=(p_{j-1},f_j,C_j,D_j).$$

But this means that the partial isomorphisms $p_i,...,p_j$ of rounds $i,...,j$ form a loop - if both players act on rounds $j+1,...,j+j-(i-1)$ as they did on rounds $i,...,j$ and so on, then the game continues infinitely long and II wins, a contradiction. Thus at some round $r>j$, σ has to tell I to act differently in order to escape from the loop. More precisely,

$$\sigma((p_{r-1},f_r))=(C_r,D_r)\neq(C_l,D_l),$$

where

$$(p_{r-1},f_r)=(p_{l-1},f_l)$$

and $i\leq l\leq j$. Indeed, it may happen that (p_{r-1},f_r,C_r,D_r) is also some repetition of already faced game position, but then I can in a similar way find his way out of this new loop. By iterating this method, player I finds a strategy, which contains no repetions and leads to some p that is not a partial isomorphism. The other direction is trivial, since a winning strategy in $FBP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b})$ is automatically a winning strategy in $BP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b})$.

(ii) Suppose player II has a winning strategy σ in $BP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b})$ i.e. she can make the game last infinitely long. There are only finitely many quadruples (p_{i-1},f_i,C_i,D_i) and hence on some round $i>j$ it holds that

$$(p_{i-1},f_i,C_i,D_i)=(p_{j-1},f_j,C_j,D_j),$$

therefore σ gives her a winning strategy in $FBP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b})$. For the converse, assume II has a winning strategy σ in $FBP^k_n(\mathcal{M},\bar{a},\mathcal{N},\bar{b})$. Whatever actions I chooses, player II can make the game last until

$$(p_{i-1},f_i,C_i,D_i)=(p_{j-1},f_j,C_j,D_j)$$

for some rounds i and j, $i<j$. In other words, she can in every situation choose bijections that induce partial isomorphisms regardless of how I acts until the game returns to some previously encountered situation. It is clear that she can make the play last infinitely long by repeating these loops over and over again.

□

Lemma 5.2.6 (Zermelo's theorem). *Every finite two-player game with perfect information is determined.*

Corollary 5.2.7. *The game $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ is determined.*

Proof. The corollary follows instantly from lemmas 5.2.5 and 5.2.6. □

Theorem 5.2.8. *Let τ be a vocabulary and $\mathcal{M}$, $\mathcal{N}$ τ-structures. Then the following conditions hold:*

(i) An l-ary query q on τ-structures is definable in $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n) \Leftrightarrow$ If $\bar{a} \in q(\mathcal{M})$ and $\bar{b} \notin q(\mathcal{N})$, then Player I has a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$,

(ii) A boolean query is definable in $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n) \Leftrightarrow$ If $q(\mathcal{M}) \neq q(\mathcal{N})$, then player I has a winning strategy in $BP^k_n(\mathcal{M}, \mathcal{N})$.

Proof. (i) Assume q is definable in $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$ and $\bar{a} \in q(\mathcal{M})$, but $\bar{b} \notin q(\mathcal{N})$. There is an $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formula $\varphi(\bar{x})$ such that

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi(\bar{x}), \text{ but } \langle \mathcal{N}, \bar{b} \rangle \nvDash \varphi(\bar{x}).$$

Therefore it is *not* the case that

$$\langle \mathcal{M}, \bar{a} \rangle \equiv_{\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)} \langle \mathcal{N}, p(\bar{a}) \rangle.$$

By theorem 5.2.3 player II does not have a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ and hence by corollary 5.2.7 player I has a winning strategy.

Suppose then I has a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$ for any τ-structures $\mathcal{M}$ and $\mathcal{N}$ and tuples $\bar{a}$ and $\bar{b}$ such that $\bar{a} \in q(\mathcal{M}), \bar{b} \notin q(\mathcal{N})$. Since II does not have a winning strategy, for all pairs of 2-tuples $(\mathcal{M}, \bar{a})$ and $(\mathcal{N}, \bar{b})$, there is an $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-formula $\varphi_{\mathcal{M},\bar{a},\mathcal{N},\bar{b}}(\bar{x})$ so that

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi_{\mathcal{M},\bar{a},\mathcal{N},\bar{b}}(\bar{x}), \text{ but } \langle \mathcal{N}, \bar{b} \rangle \nvDash \varphi_{\mathcal{M},\bar{a},\mathcal{N},\bar{b}}(\bar{x}).$$

Let $(A_i)_{i\in\omega} = ((\mathcal{M}_i, \bar{a}_i))_{i\in\omega}$ be a sequence containing up to isomorphism all the pairs $(\mathcal{M}, \bar{a})$ such that $\bar{a} \in q(\mathcal{M})$. Similarly, define $(B_j)_{j\in\omega} = ((\mathcal{N}_j, \bar{b}_j))_{i\in\omega}$ as the sequence that contains up to isomorphism all the pairs $(\mathcal{N}, \bar{b})$ such that $\bar{b} \notin q(\mathcal{N})$. Now the concatenation of them, $(A_i)_{i\in\omega} \frown (B_j)_{j\in\omega}$, is a countable list containing all such pairs that $\mathcal{M}$ is a finite τ-structure and $\bar{a} \in M^l$.

Claim.

$$\psi(\bar{x}) = \bigvee_{i\in\omega} \bigwedge_{j\in\omega} \varphi_{A_i,B_j}(\bar{x}) \text{ defines } q.$$

Proof of claim. It is enough to show that $q(\mathcal{M}) = \psi^{\mathcal{M}}(\bar{x})$ for any τ-structure $\mathcal{M}$. Suppose first that $\bar{a} \in M^l$ is in $q(\mathcal{M})$. Now $(\mathcal{M}, \bar{a})$ is isomorphic with some

$(\mathcal{M}', \bar{a}') \in (A_i)_{i\in\omega}$ and thus it holds that

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi_{A_i,B_j}(\bar{x})$$

for some $i \in \omega$ and for all $j \in \omega$. Therefore we have that $\langle \mathcal{M}, \bar{a} \rangle \vDash \psi(\bar{x})$, which means that $\bar{a} \in \psi^{\mathcal{M}}(\bar{x})$.

For the other direction, assume $\bar{a} \in \psi^{\mathcal{M}}(\bar{x})$. Thus it holds that $\langle \mathcal{M}, \bar{a} \rangle \vDash \psi(\bar{x})$, which implies that

$$\langle \mathcal{M}, \bar{a} \rangle \vDash \varphi_{A_i,B_j}(\bar{x})$$

for some $i \in \omega$ and all $j \in \omega$. By definition, $\bar{a}_i \in q(\mathcal{M}_i)$, and hence for all $j \in \omega$ it holds that

$$\langle \mathcal{M}_i, \bar{a}_i \rangle \vDash \varphi_{A_i,B_j}(\bar{x}).$$

We conclude that $\langle \mathcal{M}, \bar{a} \rangle \notin (B_i)_{i\in\omega}$, because there is no formula $\varphi_{A_i,B_j}(\bar{x})$, which would distinguish $\langle \mathcal{M}, \bar{a} \rangle$ from $\langle \mathcal{M}_i, \bar{a}_i \rangle$. This implies that $\langle \mathcal{M}, \bar{a} \rangle$ is isomorphic with some $(\mathcal{M}', \bar{a}') \in (A_i)_{i\in\omega}$, which means that $\bar{a} \in q(\mathcal{M})$ as desired.

The case $b \in N^l$ such that $\bar{b}$ is not in $q(\mathcal{N})$ is proved similarly.

(ii) The proof is similar to (i).

□

6 Construction of Models $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$

Our goal is to show that for each $n \in \mathbb{N}$, there are $PTIME$ computable boolean queries which are not definable in $\mathcal{L}^{\omega}_{\infty\omega}(\mathbf{Q}_n)$. Note that we can always find the maximum arity $n \in \omega$, if we consider a *finite* set of quantifiers. Thus the main result of the thesis will imply that in particular $PTIME$ cannot be captured with a logic of the form $\mathcal{L}^{\omega}_{\infty\omega}(\mathbf{Q})$, where $\mathbf{Q}$ is a finite set of quantifiers. To obtain the main result we will construct two non-isomorphic models which are $\mathcal{L}^{k}_{\infty\omega}(\mathbf{Q}_n)$-equivalent, but are distinquishable with a $PTIME$ query. The models used for the counterexample were originally published in [7].

Intuitively these witnessing models will be built from a large amount of basic building blocks, which are glued together with a binary relation. The idea is that the building blocks are designed so that player II can choose bijections inside the blocks quite freely and still maintain a winning strategy in $BP^k_n(\mathcal{M}, \bar{a}, \mathcal{N}, \bar{b})$. The binary relation holding between the blocks is then chosen suitably so that at least some of the winning strategies remain for II, but still so that the distinctive feature of the models can be computed in polynomial time. We start the first subsection by constructing the building blocks.

6.1 Building Blocks

There are two different kinds of building blocks, which are either models of the form $\mathcal{B}^+ = \langle B, R^+ \rangle$ or $\mathcal{B}^- = \langle B, R^- \rangle$. The domain B of these blocks is always a set having $2n+2$ elements. The relations R^+ and R^- are $n+1$-ary and they are defined in the following. We first look at the common features of all building blocks.

Definition 6.1.1. A *pre-building block* is a $\{\prec\}$-structure $\mathcal{B} = \langle B, \prec^{\mathcal{B}} \rangle$, where $B = \{c_1, ..., c_{n+1}, d_1, ..., d_{n+1}\}$ and the interpretation of the partial order $\prec$ is given as

$$x \prec^{\mathcal{B}} y \Leftrightarrow \big((x = c_i \vee x = d_i) \wedge (y = c_j \vee y = d_j)\big), \text{ where } 1 \leq i < j \leq n+1.$$

We use variables x_i (or y_i) to denote to the elements of B with index i. In other words, either $x_i = c_i$ or $x_i = d_i$. The relation $\prec$ orders the elements of $dom(\mathcal{B})$ according to the natural ordering of their indices. If we consider elements x_i and x_j with different indices $i < j$, it does not matter with respect to $\prec$ whether x_i and x_j are c's or d's. However, the only reason that $\prec$ is not a linear ordering of B is that for any index i, c_i and d_i are distinct elements, which are not comparable. With the help of this ordering we can now define the two different sorts of building blocks.

Definition 6.1.2. Let B and $\prec$ be as in definition 6.1.1. Denote by P the set $\{d_1, ..., d_{n+1}\}$, and let R^+ and R^- be $(n+1)$-ary relation symbols. We say that $\{\prec, R^+\}$-structures $\mathcal{B}^+ = (B, R^+)$ and $\{\prec, R^-\}$-structures $\mathcal{B}^- = (B, R^-)$ are *building blocks*, where the interpretations of R^+ and R^- are

$(x_1, ..., x_{n+1}) \in R^+ \Leftrightarrow x_1 \prec ... \prec x_{n+1}$ and the cardinality of $\{i : x_i \in P\}$ is even,

$(x_1, ..., x_{n+1}) \in R^- \Leftrightarrow x_1 \prec ... \prec x_{n+1}$ and the cardinality of $\{i : x_i \in P\}$ is odd.

Next we prove a lemma that characterizes the automorphisms of $\mathcal{B}^+$ and $\mathcal{B}^-$ as well as isomorphisms between them. We denote these sets of functions by $Aut(\mathcal{B}^+)$, $Aut(\mathcal{B}^-)$ and $Isom(\mathcal{B}^+, \mathcal{B}^-)$, respectively. Note that a necessary condition for a function $f : B \to B$ to belong to any of these sets is that f is a bijection and preserves the ordering $\prec$. In the following we see that in conjunction with a property concerning the parity of $exc(f)$, we get even a sufficient condition. Here $exc(f)$ is the number of how many c's f maps to d's i.e.

$$exc(f) = |\{i \in \{1, ..., n+1\} : f(c_i) = d_i\}|.$$

Lemma 6.1.3. *Let $f : B \to B$ be a bijection preserving $\prec$. The following conditions hold:*

(i) $f \in Aut(\mathcal{B}^+) \Leftrightarrow exc(f)$ is even,

(ii) $f \in Aut(\mathcal{B}^-) \Leftrightarrow exc(f)$ is even,

(iii) $f \in Isom(\mathcal{B}^+, \mathcal{B}^-) \Leftrightarrow exc(f)$ is odd.

Proof. (i) If $f \in Aut(\mathcal{B}^+)$, then f preserves in particular the relation R^+. Clearly $(c_1, ..., c_{n+1}) \in R^+$ and thus it holds that $(f(c_1), ..., f(c_{n+1})) \in R^+$. By the definition of R^+, $|\{i : f(c_i) = d_i\}|$ is even, which implies that $exc(f)$ is even. For the converse, suppose $exc(f)$ is even. This means that for all tuples $\bar{x} = (x_1, ..., x_{n+1}) \in B^{n+1}$, the cardinalities $|\{i : x_i \in P\}|$ and $|\{i : f(x_i) \in P\}|$ are either both even or both odd. Hence $\bar{x} \in R^+ \Leftrightarrow f(\bar{x}) \in R^+$. By assumption f preserves also $\prec$ and is a bijection, which together imply that $f \in Aut(\mathcal{B}^+)$. Proofs for (ii) and (iii) are, mutatis mutandis, the same as for (i).

□

Observe that from conditions (i) and (ii) it follows that $Aut(\mathcal{B}^+) = Aut(\mathcal{B}^-)$. More importantly, the previous lemma gives us an insight on how the isomorphisms (and automorphisms) between building blocks look like. Consider any bijection f, which preserves $\prec$. In the following we see that since $exc(f)$ is all that matters, and f can be extended from an arbitrary, small enough subset of B, to either an automorphism or isomorphism of the whole structure. More specifically, any f and any set $X \subseteq B$ of size n induce an automorphism and an isomorphism, which both assign elements inside X the same values as f does.

Proposition 6.1.4. *Let $f : B \to B$ be a bijection preserving $\prec$ and $X = \{x_1, ..., x_n\} \subseteq B$. Then there exists $g \in Aut(\mathcal{B}^+)$ and $h \in Isom(\mathcal{B}^+, \mathcal{B}^-)$ such that $f \restriction X = g \restriction X = h \restriction X$.*

Proof. If $exc(f)$ is even, then choose $g = f$. By Lemma 6.1.3, it holds that $g \in Aut(\mathcal{B}^+)$. If $exc(f)$ is odd, we modify f suitably to obtain g: Since $|X| = n$, there is for some i, $1 \leq i \leq n+1$, a pair $\{c_i, d_i\} \in B \setminus X$. Now let g be such that

$$g \restriction (B \setminus \{c_i, d_i\}) = f, \text{ but } g(c_i) \neq f(c_i)$$

(which implies that also $g(d_i) \neq f(d_i)$). Now g has one more or one less points in which it changes c to d and hence $exc(g)$ is even, meaning that $g \in Aut(\mathcal{B}^+)$. The claim for h is proven in the same way, just choose $h = f$ if $exc(f)$ is odd, and do the above modification otherwise.

□

6.2 Structures A(G) and B(G)

In this subsection we continue the construction of the models that are later proved to be suitable for our purpose. We begin with a finite connected graph and add extra structure to it with the help of the building blocks. Each vertex of the original graph is replaced with a building block. Hence if the initial graph has m vertices, the structure obtained has $m(2n+2)$ elements. Furthermore, we will add some extra relations to this structure. Let us now start the first phase of the construction.

Recall that a graph $\mathbf{G} = \langle G, E^{\mathbf{G}} \rangle$ is connected, if for all $u, v \in G$ there exists a path from u to v. A *path* from u to v is a sequence $v_0 ... v_m$ (here $u = v_0$ and $v = v_m$) of elements of G such that $(v_i, v_{i+1}) \in E^{\mathbf{G}}$ for all $0 \leq i \leq m-1$. A vertex u has degree n, if $|\{v \in G : (u, v) \in E^{\mathbf{G}}\}| = n$. Furthermore, a graph is of degree n if all of its vertices have degree n. Suppose G is a graph of degree n. We associate to each vertex u a coloring h_u of its neighbours, where h_u is a bijection from $\{v \in G : (u, v) \in E^{\mathbf{G}}\}$

to $\{1, ..., n+1\}$. These colorings are independent of each other - for vertices u and v, it is possible that $h_u(v) \neq h_v(u)$.

We next define the structure obtained by replacing vertices with building blocks.

Definition 6.2.1. Let $\mathbf{G}$ be a finite connected graph of degree $n+1$, $n \geq 2$ and $S \subseteq G$. Moreover, let B be the domain of a (pre-)building block $\mathcal{B}$ and h_v some colorings of neighbours of vertices of G. The structure

$$\mathcal{C}(\mathbf{G}, S) = \langle C_G, R^{\mathcal{C}(\mathbf{G},S)}, E^{\mathcal{C}(\mathbf{G},S)} \rangle$$

is defined as follows:

- $C_G = G \times B$, where $|B| = 2n+2$.

- $R^{\mathcal{C}(\mathbf{G},S)}$ is an $(n+1)$-ary relation on C_G such that

$\big((u, x_1), ..., (u, x_{n+1})\big) \in R^{\mathcal{C}(\mathbf{G},S)} \Leftrightarrow$ either $u \notin S$ and $(x_1, ..., x_{n+1}) \in R^+$ or $u \in S$ and $(x_1, ..., x_{n+1}) \in R^-$, where R^+ and R^- are the relations of building blocks as in definition 6.1.2,

- $E^{\mathcal{C}(\mathbf{G},S)}$ is a binary relation on C_G such that

$\big((u, x_i), (v, x_j)\big) \in E^{\mathcal{C}(\mathbf{G},S)} \Leftrightarrow (u, v) \in E^{\mathbf{G}}$, $h_u(v) = i$ and $h_v(u) = j$, and either $x_i = c_i \wedge x_j = c_j$ or $x_i = d_i \wedge x_j = d_j$.

$\mathcal{C}(\mathbf{G}, S)$ is the structure promised in the beginning of this subsection. We point out that it has $(|G|(2n+2))$ elements and twice the number of edges compared to $\mathbf{G}$. More specifically, if there is an edge between u, $v \in G$ and, say $h_u(v) = 1$ and $h_v(u) = 2$, then there are correspondingly the two edges $((u, c_1), (v, c_2))$ and $((u, d_1), (v, d_2))$ in $\mathcal{C}(\mathbf{G}, S)$. The relation $R^{\mathcal{C}(\mathbf{G},S)}$ labels the building blocks replacing vertices so that for the vertices in S the building block is $\mathcal{B}^-$ and for the vertices outside S it is $\mathcal{B}^+$.

Lemma 6.2.2. *Let $\mathbf{G} = \langle G, E^{\mathbf{G}} \rangle$ be a finite connected graph of degree $n+1$ and $S, T \subseteq G$ such that $(S \setminus T) \cup (T \setminus S) = \{u, v\}$, where $u, v \in G$ and $u \neq v$. Then*

$$\mathcal{C}(\mathbf{G}, S) \cong \mathcal{C}(\mathbf{G}, T).$$

Proof. Let $\{u, v\} = (S \setminus T) \cup (T \setminus S)$. There exists a path $u = v_0, ..., v_{m+1} = v$, because $\mathbf{G}$ is connected. Note that each vertex $v_i, 1 \leq i \leq m$, belong either to both

S and T or to neither of them. This leads to the following observation. Suppose we have an isomorphism

$$f : \mathcal{C}(\mathbf{G}, S) \cong \mathcal{C}(\mathbf{G}, T)$$

such that $f(w, x) = (w, y)$. We can define for each vertex $w \in G$ a bijection $f_w : B \to B$ such that $f_w(x) = y$. By Lemma 6.1.3 it must hold that for all v_i, $exc(f_{v_i})$ is even, whereas both $exc(f_u)$ and $exc(f_v)$ are odd. This is because f has to, in particular, preserve the relation R. A natural candidate for f is hence such a function that the corresponding f_w exchanges two c and d components along the path and only one in the end vertices u and v. Now define the bijection f as

$$f(w, x) = \begin{cases} (w, d_j), & \text{if } w = v_i, x = c_j, \text{ and } j = h_{v_i}(v_{i+1}) \text{ or } j = h_{v_i}(v_{i-1}) \\ (w, c_j), & \text{if } w = v_i, x = d_j, \text{ and } j = h_{v_i}(v_{i+1}) \text{ or } j = h_{v_i}(v_{i-1}) \\ (w, x), & \text{otherwise.} \end{cases}$$

Moreover, let $f_w : B \to B$ be such that

$$f_w(x) = y \Leftrightarrow f(w, x) = (w, y)$$

for all $w \in G$ and $x \in B$. Now by the argument above, $f_{v_i} \in Aut(\mathcal{B}^+) = Aut(\mathcal{B}^-)$ for each $1 \leq i \leq m$ and $f_u, f_v \in Isom(\mathcal{B}^+, \mathcal{B}^-)$. Furthermore, f_w is identity for the rest of the vertices, and hence

$$\bar{x} \in R^{\mathcal{C}(\mathbf{G},S)} \Leftrightarrow f(\bar{x}) \in R^{\mathcal{C}(\mathbf{G},T)}$$

for all $\bar{x} \in C_G^{n+1}$. Since f preserves also the edge relation E, it holds that

$$f : \mathcal{C}(\mathbf{G}, S) \cong \mathcal{C}(\mathbf{G}, T).$$

□

We can now use the above lemma to show that for a given graph $\mathbf{G}$, there exists (up to isomorphism) only two different structures $\mathcal{C}(\mathbf{G}, S)$. Moreover, the isomorphism type of $\mathcal{C}(\mathbf{G}, S)$ is determined by the parity of S.

Lemma 6.2.3. *$\mathcal{C}(\mathbf{G}, S) \cong \mathcal{C}(\mathbf{G}, T)$ if and only if $|S|$ and $|T|$ are of the same parity.*

Proof. Consider any $S \subseteq G$ such that $|S| \geq 2$ and let $S' \subseteq S$ be such that $|(S \backslash S')| = 2$. By the previous lemma, $\mathcal{C}(\mathbf{G}, S) \cong \mathcal{C}(\mathbf{G}, (S'))$. In other words, we can erase two elements from S and preserve the isomorphism type of it. Since S is finite, by applying this trick multiple times, we eventually reach either $\mathcal{C}(\mathbf{G},\emptyset)$ or $\mathcal{C}(\mathbf{G},\{u\})$ depending on the parity of S. If in the first place $|S| < 2$, then S is already isomorphic to either $\mathcal{C}(\mathbf{G},\emptyset)$ or $\mathcal{C}(\mathbf{G},\{u\})$. We conclude that if S and T are of same parity, then $\mathcal{C}(\mathbf{G},S)$ and $\mathcal{C}(\mathbf{G},T)$ are isomorphic. For the other direction, and by the above argument, it is enough to show that $\mathcal{C}(\mathbf{G},\emptyset)$ and $\mathcal{C}(\mathbf{G},\{u\})$ are not isomorphic. Consider the reducts $\mathcal{C}(\mathbf{G},\emptyset)^*$, $\mathcal{C}(\mathbf{G},\{u\})^*$ of the models $\mathcal{C}(\mathbf{G},\emptyset)$, $\mathcal{C}(\mathbf{G},\{u\})$, in which the edge relation E is omitted, and suppose

$$f : \mathcal{C}(\mathbf{G},\{u\})^* \to \mathcal{C}(\mathbf{G},\emptyset)^*$$

is an isomorphism.

We show first that such an isomorphism cannot map two distinct elements of the same building block to elements of different building blocks. More formally, we show that there exist bijections $g : G \to G$ and $f_v : B \to B$ such that $f((v,x)) = (g(v), f_v(x))$, where $v \in G$ and $x \in B$ are arbitrary. Towards a contradiction, assume there are elements (v,x) and (v,y) such that

$$f((v,x)) = (u,x') \text{ and } f((v,y)) = (w,y'), \text{ where } u \neq w.$$

Obviously, there are two possibilities. Either x and y have the same index, in which case one of them is c element and one d element, or they have different indices. Consider the latter option first. We can always construct an $(n+1)$-tuple, which belongs to $R^{\mathcal{C}(\mathbf{G},\{u\})}$ and contains both elements (v,x) and (v,y). This is true, because we can freely choose between (v,c) and (v,d) as the element(s) to be included to the tuple, except that is has to contain (v,x) and (v,y). Since $n+1 \geq 3$, there is at least one c,d-pair with different index than x and y, and therefore we can adjust the parity of the number of d elements in the tuple in order to make it belong to $R^{\mathcal{C}(\mathbf{G},\{u\})}$. Hence the relation R cannot be preserved, if (v,x) and (v,y) are mapped into different building blocks.

Consider the alternative case then. Without loss of generality, assume $x = c_i$ and $y = d_i$. It is enough to find two different $(n+1)$-tuples $s,t \in R^{\mathcal{C}(\mathbf{G},\{u\})}$ such that s contains (v,c_i) and t contains (v,d_i), and there is at least one element (v,z), which belongs to both s and t. It is easy to construct such s and t: If $v \neq u$, let s be such that it contains only elements of the form (v,c_k) and t be similar to s, except that it contains one element (v,d_l) instead of (v,c_l), $l \neq i$. The case $v = u$ is treated similarly.

So we can now assume that $g : G \to G$ and $f_v : B \to B$ are bijections such that $f((v,x)) = (g(v), f_v(x))$. Since for all $v \neq u$ it holds that $v \notin \{u\}$, it must be the case that f_v is an automorphism of $\mathcal{B}^+$, since otherwise f is not an isomorphism. On the other hand, f_u is clearly an isomorphism $\mathcal{B}^- \to \mathcal{B}^+$, since $u \in \{u\}$, but $u \notin \emptyset$. Now by Lemma 6.1.3, $\sum_{v \in G} exc(f_v)$ is odd. Now consider the original models by expanding the models with the edge relation E. There exists some pair $((v_1,x),(v_2,y)) \in (G \times B)^2$ for which it does not hold that

$$((v_1,x),(v_2,y)) \in E^{\mathcal{C}(\mathbf{G},\{u\})} \Leftrightarrow ((v_1,x),(v_2,y)) \in E^{\mathcal{C}(\mathbf{G},\emptyset)}.$$

Consequently, f cannot be an isomorphism $C(\mathbf{G},\{u\}) \to C(\mathbf{G},\emptyset)$.

□

We conclude that for a given graph $\mathbf{G}$ there are essentially only two different structures $\mathcal{C}(\mathbf{G},S)$. It is natural to choose $C(\mathbf{G},\emptyset)$ and $C(\mathbf{G},\{u\})$ to represent the two isomorphism classes of structures $\mathcal{C}(\mathbf{G},S)$.

To make things simpler in the following, denote $C(\mathbf{G},\emptyset)$ and $C(\mathbf{G},\{u\})$ as

$$\langle A(\mathbf{G}), R^{\mathbf{A(G)}}, E^{\mathbf{A(G)}}\rangle \text{ and } \langle B(\mathbf{G}), R^{\mathbf{B(G)}}, E^{\mathbf{B(G)}}\rangle,$$

respectively. Next we add the final features to these models by adding a linear order on the set of the vertices of $\mathbf{G}$.

Definition 6.2.4. Let $<$ be a binary relation symbol and $<^{\mathbf{G}}$ some fixed linear order on G with the least element $u \in G$. Structures $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ are acquired from models $C(\mathbf{G},\emptyset)$ and $C(\mathbf{G},\{u\})$ by expanding them with $<$ as:

$\mathbf{A}(\mathbf{G}) = \langle A(\mathbf{G}), R^{\mathbf{A(G)}}, E^{\mathbf{A(G)}}, <^{\mathbf{A(G)}}\rangle$,

$\mathbf{B}(\mathbf{G}) = \langle B(\mathbf{G}), R^{\mathbf{B(G)}}, E^{\mathbf{B(G)}}, <^{\mathbf{B(G)}}\rangle$.

Here $A(\mathbf{G}) = B(\mathbf{G}) = G \times B$. Moreover, the relation $<^{\mathbf{A(G)}} = <^{\mathbf{B(G)}}$ is defined with the help of $<^{\mathbf{G}}$ and $\prec$ as

$$(v,x) <^{\mathbf{A(G)}} (w,y) \Leftrightarrow v <^{\mathbf{G}} w \vee (v = w \wedge x \prec y).$$

Intuitively, the additional ordering $<^{\mathbf{A(G)}}$ is a lexigographic ordering on $G \times B$. Given two elements from it, one first compares the first co-ordinates that are totally ordered by $<^{\mathbf{G}}$. If the first components are the same, then one looks at second ones, which are partially ordered by $\prec$. Thus it is clear that $<^{\mathbf{A(G)}}$ is a partial order and the only pairwisely incomparable elements are the ones of the form (v, c_i) and (v, d_i).

7 Main Results

7.1 $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$

Structures $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ are finally the desired models that we will use in our main result. We already know that they are non-isomorphic because of Lemma 6.2.3. Next we aim to show that there is a graph $\mathbf{G}$ so that $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ are similar enough to be $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-equivalent.

A natural approach is to consider the n-bijective k-pebble game $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$. Recall that by Theorem 5.2.3 it is enough to find a winning strategy of player II in order to show the desired $\mathcal{L}^k_{\infty\omega}(\mathbf{Q}_n)$-equivalence. Thus we next focus on what kind of bijections f_i player II should play in a winning strategy. We proceed by narrowing down the set of feasible bijections for II. A minimal requirement is that the bijections f_i should be such that player I cannot win the game on round i. Say that a bijection is *good*, if this is the case. We start by looking at what conditions relations E and $<$ impose on the set of good bijections.

Let $v, w \in G$ and $x, y \in B$. Since $<$ is a linear order on G, player II should not play on any round i a bijection $f_i : A(\mathbf{G}) \to B(\mathbf{G})$ such that

$$f_i(v, x) = (w, y), \text{ for some } v \neq w.$$

In other words, f_i should be the identity function $id_{\mathbf{G}}$ with respect to the first co-ordinate. This is the case, because otherwise I can respond by choosing C_i arbitrarily and D_i as

$$D_i = \{(v, x), (w, y)\},$$

(by assumption $n \geq 2$) where $v, w \in G$ are such that $v <^{\mathbf{G}} w$, but $f_i(w) <^{\mathbf{G}} f_i(v)$. Thus, $p_i = (p_{i-1} \restriction C_i) \cup (f_i \restriction D_i)$ is not a partial isomorphism and I wins the game. Given this observation, we can define for bijections f_i, every $v \in G$ and $x \in B$ a *new bijection* $f_{i,v} : B \to B$ as the one satisfying the condition

$$f_i(v, x) = (v, f_{i,v}(x)).$$

We now know that a good bijection has to map each building block of $\mathbf{A}(\mathbf{G})$, which is labeled with a vertex of $\mathbf{G}$, onto the same building block in $\mathbf{B}(\mathbf{G})$.

Consider then the second co-ordinate of the argument of f_i. Player II has to choose a bijection, which preserves the partial order $\prec$. Hence, it is clear that a good bijection cannot change the index of any x_j, because otherwise it changes also the index of some y_l, $j \neq l$, and again, the ordering would be altered.

Since $n \geq 2$, II must indeed preserve also E on every round, because otherwise player I wins by choosing C_i arbitrarily and D_i as $D_i = \{(v,x),(w,y)\}$, where (v,x) and (w,y) are such that

$$((v,x),(w,y)) \in E^{\mathbf{A}(\mathbf{G})}, \text{ but } (f_i(v,x), f_i(w,y)) \notin E^{\mathbf{B}(\mathbf{G})}.$$

This gives rise to a question: Can a good bijection even change any c_j to d_j or the other way round. Put differently, is it possible that $f_i(v,x_j) = (v,y_j)$, where $x_j \neq y_j$? The answer is affirmative, at least if we consider only the preservation of $<$ and E, which we have shown to be necessary conditions for good bijections. Soon we will see that actually f_i has to make these exchanges in order to preserve R. Before turning the focus to the preservation of the relation R, we summarize what we know so far. Namely, f_i has to satisfy the following condition on each round i of $BP_n^k(\mathbf{A}(\mathbf{G}),(\mathbf{B}(\mathbf{G}))$:

$$\{f_i(v,c_j), f_i(v,d_j)\} = \{(v,c_j),(v,d_j)\}.$$

This leads us to the constraints that the relation R imposes on f_i's or, more precisely, when it should change x_j to y_j, $x_j \neq y_j$. Let u be the least element according to $<^{\mathbf{G}}$ (i.e. the one distincting $\mathbf{A}(\mathbf{G})$ from $\mathbf{B}(\mathbf{G})$).

Suppose $(u,c_j) \in dom(p_{i-1})$ for some $1 \leq j \leq n+1$ and consider round i in the game $BP_n^k(\mathbf{A}(\mathbf{G}),(\mathbf{B}(\mathbf{G}))$. Player II has to be careful with her move, since I can respond to her move by including (u,c_j) in C_i and choosing

$$D_i = \{(u,c_l) : 1 \leq l \leq n+1, l \neq j\}.$$

The danger for II is that then it is possible that p_i violates relation R, because it contains an $(n+1)$-tuple $((u,c_1),...,(u,c_{n+1}))$. However, player II can cope with the situation by ensuring that $f_{i,u}$ is an isomorphism $\mathcal{B}^+ \to \mathcal{B}^-$. The reason for this is the trivial fact that $u \in \{u\}$, but $u \notin \emptyset$. Luckily there is a safe bet for II - just choose f_i so that $exc(f_{i,u})$ is odd. Then by Lemma 6.1.3, $f_{i,u}$ is the desired isomorphism and, of course, does not violate the previous condition of preservation of indices of x_j's and most importantly, the relation R is preserved by f_i.

The case $(u,d_j) \in dom(p_{i-1})$ for some $1 \leq j \leq n+1$ is treated similarly. Also by a very similar argument we see that for $v \neq u$, $v \in dom(p_{i-1})$, $f_{i,v}$ must be an automorphism of $\mathcal{B}^+$.

There are no other relations in $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$), so we conclude that if player II chooses a bijection f_i according to the above constraints, then f_i is a good bijection.

However, this is not enough to give player II a winning strategy. Although player II cannot lose on a round by playing a good bijection, it may very well happen that she loses on some subsequent round due to a new position of the game. We now turn the focus on this aspect and keep in mind that every bijection player II plays has to at least be a good bijection.

7.2 A winning strategy for player II in $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$

A requirement for a victorious strategy of II in $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$ is that on each round $i \in \omega$, f_i needs to be a good bijection. We can thus state our current challenge as: given that f_i is good, is there is a uniform way for player II to choose f_{i+1} so that it is also a good bijection? A positive answer would yield a winning strategy for her. We will see that there is indeed such a strategy, if the graph $\mathbf{G}$ satisfies certain largeness condition. A more precise meaning of large in this context comes as a by-product, when we consider the question of how to define f_{i+1} by means of f_i. Instead of immediately constructing a suitable graph, we begin by reflecting on how II should play rationally. This in turn leads us to a condition the graph should satisfy, so that II can play the way she should play.

We prove by induction on i, that if f_i is a good bijection, then under certain circumstances player II can choose f_{i+1} so that it is also a good bijection. It is clear that player II cannot start with a bijection f_1 that would preserve the relation R everywhere, since in that case she would break the edge relation and lose immediately. Thus she is forced to choose f_1 so that for some $v \in G$, $v \neq u$,

$$f_{1,v} \notin Aut(\mathcal{B}^+) \text{ or } f_{1,u} \notin Isom(\mathcal{B}^+, \mathcal{B}^-).$$

Consequently, there is no better choice for II than setting $f_1 = id_{G\times B}$, so it is natural to assume player II picks it. In this case, the "*bad part*" of f_1 is $f_{i,u}$, the one corresponding to the building block labeled with u. Note also that this f_1 is clearly a good bijection. So in the initial step of the induction player II sets $f_1 = id_{G\times B}$.

For the induction step, assume f_i is a good bijection and the bad part of f_i is $f_{i,v}$, for some $i \in \mathbb{N}$ and $v \in G$. The fact that f_i is a good bijection implies that there cannot be any elements of the form (v, c_j) in $dom(p_{i-1})$, since otherwise I can choose $C_i \subseteq dom(p_{i-1})$ so that $(v, c_j) \in C_i$ and, furthermore

$$D_i = \{(v, c_l) : 1 \leq l \leq n+1, l \neq j\}$$

and win the game on round i. Thus if I chooses D_i so that it does not contain elements of the form (v, c_j), then neither does $dom(p_i)$. In this case II can choose $f_{i+1} = f_i$ and it remains a good bijection. (Note that we omitted case (v, d_j). However, it is treated similarly as (v, c_j) above.)

Hence without loss of generality we can assume there is at least one (v, c_j) in D_i. At this point player II has to be careful, since if she picks f_{i+1} such that

$$f_{i+1,v} = f_{i,v},$$

then I wins the game. More specifically, I can choose D_{i+1} so that it contains the rest n elements $(v, c_l), 1 \leq l \leq n+1$ and $l \neq j$, from block v. Then $dom(p_{i+1})$ contains an $(n+1)$-tuple $((v, c_1), ..., (v, c_{n+1}))$ from the bad part of f_{i+1} and clearly II loses.

However, II has a way out of this problem. There is at least one element (v, c_r), which is not in $dom(p_i)$ and thus cannot be chosen by I to C_{i+1}. Now by changing the c_r and d_r components with each other in $f_{i+1,v}$ player II is no more in trouble: $exc(f_{i,v})$ is of different parity than $exc(f_{i+1,v})$. This means that $f_{i+1,v}$ is not a bad part of f_{i+1}, because now

$$f_{i+1,v} \in Aut(\mathcal{B}^+) \text{ (or } f_{i+1,v} \in Isom(\mathcal{B}^+, \mathcal{B}^-), \text{ if } v = u).$$

Doing only this adjustment to f_{i+1} is not enough for II, because f_{i+1} would not preserve the edge relation E. However, II can "move" the bad part away along a path $v = v_0, ..., v_s = v'$ of $\mathbf{G}$. Player II chooses f_{i+1} so that it changes the c and d components with each other along the path. More formally, player II sets f_{i+1} to be the bijection that satisfies

$$f_{i+1}(w, x) = f_{i+1,w}(x),$$

where $f_{i+1,w}$ is defined as:

$$f_{i+1,w}(x) = \begin{cases} d_j, & \text{if } w = v_i, f_{i,w}(x) = c_j, \text{ and } j = h_{v_l}(v_{l+1}) \text{ or } j = h_{v_l}(v_{l-1}), \\ c_j, & \text{if } w = v_i, f_{i,w}(x) = d_j, \text{ and } j = h_{v_l}(v_{l+1}) \text{ or } j = h_{v_l}(v_{l-1}), \\ f_{i,w}(x), & \text{otherwise.} \end{cases}$$

Thus $f_{i+1,w}$ changes one c-element to d-element in the first and last block of the path, whereas for the rest of the blocks along the path it makes two such changes. Moreover, f_{i+1,v_l}, $0 \leq l \leq s-1$, is an automorphism for the blocks along the path and an isomorphism $\mathcal{B}^+ \to \mathcal{B}^-$ for the last block of the path, which is labeled by v'. However, there are still two problems for player II. Firstly, suppose some of the exchanged elements along the path, i.e. ones of the form (v_l, x), $1 \leq l \leq s-1$, for which $f_{i+1,v_l}(x) \neq f_{i,v_l}(x)$, is in $dom(p_i)$ and call it (v_r, x_j). Now player I can include (v_r, x_j) in C_{i+1} and (v_{r+1}, y_t) in D_{i+1}, where (v_{r+1}, y_t) is such that there is an edge between (v_r, x_j) and (v_{r+1}, y_t). Clearly p_{i+1} cannot preserve the edge relation E, since it makes a c, d-exchange for (v_{r+1}, y_t), but not for (v_r, x_j). The second problem is that some element (v', x) in the last block v' may be in $dom(p_i)$. This situation reminds one that we have already seen. The problem is that player I can include certain n elements from v' to D_{i+1} and (v', x) to C_{i+1} and thus violate the preservation of the relation R.

The considerations above lead to the largeness condition for $\mathbf{G}$. Assume the path $v = v_0, ..., v_s = v'$ has the following property: No element (v_l, x), $1 \leq l \leq s-1$, for which $f_{i+1,v_l}(x) \neq f_{i,v_l}(x)$ is in $dom(p_i)$ and no element (v', x) from the last block is in $dom(p_i)$. Put differently, neither any of the exchanged elements nor any element from the last block belong to $dom(p_i)$. Now whatever choices I makes for C_{i+1} and D_{i+1}, it holds that

$$p_{i+1} = (p_i \upharpoonright C_{i+1}) \cup (f_{i+1} \upharpoonright D_{i+1})$$

is a partial ismorphism. Hence we conlude that f_{i+1} is a good bijection.

The considerations above give us some hint of what kind of a graph $\mathbf{G}$ should be in order to allow a winning strategy for player II in $BP^k_n(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$. The graph should be large enough so that player II can always move the bad part of a bijection away along a path, for which elements of the path are not in the domain of the partial isomorphism of the previous round. Of course the problem is hence to construct an actual graph that has the desired property. We postpone this problem for a while. Instead, we try to make the problem more intuitive by introducing a new game. Then we prove that the existence of a winning strategy for player II in this new game guarantees that she has a winning strategy also in the game $BP^k_n(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$. Afterwards it will be easier to construct the desired graph $\mathbf{G}$ and show that player II has a winning strategy in this new game played on $\mathbf{G}$.

7.3 The Game of k Cops and a Robber

We introduce a new game in which all the essential parts of the n-bijective k-pebble game are included, but in a much more simpler and intuitive manner. The new game, called the game of k *cops and a robber*, is played on $\mathbf{G}$ using the graph as a board. Player II moves the robber along paths of $\mathbf{G}$ and tries to escape from the cops. Thus player II is identified as the *robber* in this new game. On the other hand, player I is the *chief police officer* trying to catch the robber with his k cops patrolling on the edges of $\mathbf{G}$. He has two different kinds of cops. There are all together k cops in his police forces, of which n cops belong to *rapid deployment forces.* Cops in this special unit are able to move onto any edge of $\mathbf{G}$ after each movement of the robber. The rest of the k cops are too busy eating doughnuts to react on robber's movement. Unfortunately, even cops of the rapid deployment forces begin to eat doughnuts and lose their ability to move immediately after they have moved to some edge of $\mathbf{G}$. However, a doughnut eating cop can still catch the robber, if he is at an edge adjacent to robber's vertex. The chief police officer can after each movement of the robber return at most n cops of the at most k cops on $\mathbf{G}$ back to police station. Due to his strict discipline, all of these cops count as members of rapid deployment forces.

Definition 7.3.1. The *game of k cops and a robber* on a finite connected graph $\mathbf{G}$, denoted by $CR^k_n(\mathbf{G})$, where n is the number of cops in rapid deployment forces, is defined as follows:

- In the initial history of the game there are no cops in play and the robber is on vertex u (the least element of $<^{\mathbf{G}}$). On round 0, player I places at most n cops on the edges of $\mathbf{G}$.

- At the beginning of round i, $i \geq 1$, there are r cops, $r \leq k$, eating doughnuts on the edges of $\mathbf{G}$. Player II may move the robber along some path P of $\mathbf{G}$ such that edges of P do not contain any cops. Afterwards, I calls s cops $(0 \leq s \leq r)$ back to the police station and places l cops, $0 \leq l \leq n$, of the rapid deployment forces on edges not already containing a cop. Moreover, I has to act so that $(r - s) + l \leq k$.

- Player I wins the game, if at the end of some round $i \in \omega$ the robber is surrounded by cops i.e. if all the edges adjacent to the vertex of the robber have a cop.

- Player II wins the game, if she does not lose it at some round $i \in \omega$.

Now what is the analogy between $CR^k_n(\mathbf{G})$ and $BP^k_n(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$? At round i, the vertex v containing the robber marks the bad part $f_{i,v}$ of f_i, whereas the movement of robber along a path corresponds to moving the bad part away from the reach of player I. On the other hand, the $r-s$ cops eating doughnuts label the corresponding vertices, which player I includes in $C_i \subseteq dom(p_{i-1})$. The freshly added l members of rapid deployment forces correspond to vertices in D_i. The restriction $(r-s)+l \leq k$ comes from the fact that $|(C_i \cup D_i)| \leq k$. Thus, if the robber can always escape from the cops, it means that player II can always find a good bijection, which moves the bad part away from the reach of player I. Based on this analogy, we get the following obvious lemma.

Lemma 7.3.2. *If player II has a winning strategy in $CR_n^k(\mathbf{G})$, then she has a winning strategy in $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$.*

Proof. Suppose player II has a winning strategy in $CR_n^k(\mathbf{G})$. Thus on any round $i \in \omega$, the robber can escape from the cops. Now suppose that the robber is on vertex v at the beginning of round i. Since she can escape from the cops, there is a path of $\mathbf{G}$, which does not contain any cops and leads to a vertex v'. By assumption, robber has an escape route also from v' on round $i+1$ i.e. after I has moved at most n cops and robber being on vertex v'. Without loss of generality, we may assume that on round i of $BP_n^k(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$ the bad part of f_i is $f_{i,v}$ and f_i is a good bijection. Now II chooses f_{i+1} as in the illustration of winning strategy for her and the path related to the definition of f_{i+1} is the escape path of the robber on round i. Consequently, she moves the bad part to v'. Since robber has an escape route from v', the bijection f_{i+1} (with bad part $f_{i+1,v}$) is also a good bijection.

□

7.4 Results

We begin by showing that for each $k > n$, there is a finite connected graph $\mathbf{G}$ of degree $n+1$ so that structures $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ are $\mathcal{L}_{\infty\omega}^k(\mathbf{Q}_n)$-equivalent. This result is achieved with the help of the game of k cops and a robber.

Lemma 7.4.1. *For each $k > n$, there is a finite connected graph of degree $n+1$ such that the robber has a winning strategy in $CR_n^k(\mathbf{G})$.*

Proof. We construct the graph $\mathbf{G}$ from a collection $\mathbf{H} = \langle H_i, E^{\mathbf{H}_i} \rangle_{0 \leq i \leq m}$ of smaller graphs, which are assumed to be mutually distinct. Assume that for each i, $\mathbf{H}_i$ is a connected graph of degree n, has $m \geq 2k+2$ elements and remains connected, if less than n of its edges are removed. More specifically, let

$$H_i = \{v_{ij} : 0 \leq j \leq m, j \neq i\}.$$

Now let $\mathbf{G} = \langle G, E^{\mathbf{G}} \rangle$ be defined as follows:

$$G = \bigcup_{0 \leq i \leq m} H_i,$$

and

$$E^{\mathbf{G}} = \bigcup_{0 \leq i \leq m} E^{\mathbf{H}_i} \cup \{(v_{ij}, v_{ji}) : 0 \leq i, j \leq m, i \neq j\}.$$

Hence $\mathbf{G}$ is graph consisting of graphs $\mathbf{H}_i$ with all of their own edges and in addition a single edge (v_{ij}, v_{ji}) connecting each $\mathbf{H}_i$ to $\mathbf{H}_j$, $i \neq j$. Note that $\mathbf{G}$ is a connected graph of degree $n+1$ and has $m(m+1)$ elements (There are $m+1$ graphs in $\mathbf{H}$ each having m elements). We show that player II has a winning strategy in $CR_n^k(\mathbf{G})$. Let us say that a vertex v_{ij} is *safe* for the robber, if there are no cops adjacent to any vertex in H_i or H_j. By the following combinatorial argument we see that there is always a safe vertex for robber, because there are only k cops and $m(m+1)$ vertices.

Player I can place a cop between essentially two different kinds of vertices. A cop can be placed between two vertices in the same block H_i or between vertices v_{ij} and v_{ji} of blocks H_i and H_j. In the former case, a cop can make (at most, if some of the vertices are already in control of another cop) $2m$ vertices unsafe for the robber; all the m vertices in H_i and all the m vertices of the form v_{ji} in the rest m blocks H_j, $j \neq i$. However, the latter option is a better strategy for I (to be verified later). By placing a cop between v_{ij} and v_{ji}, it makes at most

$$2m + 2(m-1)$$

vertices unsafe - all the $2m$ vertices in H_i and H_j as well as $2(m-1)$ vertices of the form v_{ri} and v_{rj} in the rest

$$(m+1) - 2 = m - 1$$

blocks H_r, $r \neq i, j$. It is easy to calculate how many vertices I can make unsafe for the robber with k cops by following this better strategy. As seen above, the first cop on edge (v_{ij}, v_{ji}) makes

$$2m + 2(m-1)$$

vertices unsafe. Now it is clear that the best choice for the edge of the second cop is (v_{rs}, v_{sr}), where i, j, r and s are all mutually different. This makes $2(m-2)$ vertices of the blocks H_r and H_s unsafe, since two vertices of both of the blocks were already made unsafe by the first cop. Moreover, the second cop makes vertices of the form

v_{pr} and v_{ps} unsafe, and there are now $2(m-3)$ such vertices, which are not already made unsafe by the first cop. Eventually, by placing cops like this, all the k cops make

$$\sum_{l=1}^{k} 2(m-2(l-1)) + 2(m-(2l-1)) = \sum_{l=1}^{k} 4m - 8l + 6$$

vertices unsafe. Given the restrictions for k and m, it holds that for each l, the l^{th} term in the sum is greater than $2m$, which verifies that the chosen strategy for I is indeed better than the strategy of placing cops "inside" blocks. Finally, an easy calculation shows that there is always a safe vertex for the robber. The number of unsafe vertices is given as:

$$\sum_{l=1}^{k} 4m-8l+6 = 4km+6k-\frac{8k(k+1)}{2} \leq -4(\frac{m}{2}-1)^2+(\frac{m}{2}-1)(4m+2) = m^2+m-6,$$

since $k \leq \frac{m}{2} - 1$. By assumption, there are $m^2 + m$ vertices in $\mathbf{G}$. Thus there are always six safe vertices for the robber.

Naturally the winning strategy for II is to put the robber on a safe vertex, say v_{jl} at the beginning of round i. Player I has only n cops in his rapid deployment forces. This implies that after moving them, either (a) the chief police officer still has less than n cops on the edges of H_j or otherwise (b) there are no cops adjacent to vertices of H_l . Recall that there is a safe vertex for the robber in any game position. Without loss of generality, assume v_{rs} is a safe vertex in this new position.

Consider the alternative (a) first. By assumption, H_j is a graph of degree n, which remains connected when less than n of its edges are removed. Hence there is a cop-free path for the robber to any vertex inside the block H_j. In order to reach the new safe vertex, the robber acts as follows. First she moves to vertex $v_{jr} \in H_j$. Since $v_{rs} \in H_r$ is safe, by definition, there are no cops adjacent to vertices of H_r. Hence there is no cop on edge (v_{jr}, v_{rj}). Therefore she can use this edge as an escape route leading to the block H_r. There are no cops between vertices of H_r, and thus she can freely move from v_{rj} to the vertex v_{rs}.

Assume then that the option (b) holds i.e. after the movement of rapid deployment forces, there are no cops adjacent to vertices of H_l. Thus, in particular, edges (v_{jl}, v_{lj}) and (v_{lr}, v_{rl}) do not contain cops. But this gives the robber an escape route from H_j to H_l, and finally, from H_l to H_r. As above, there are no cops between vertices of H_r, and therefore she can move to the vertex v_{rs}.

We conclude that in any case, after the action of rapid deployment forces on round i, II can move the robber to a new safe vertex on round $i+1$.

□

The previous lemma gives us the other part of the main result. However, we still need to show that structures $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ differ from each other in some respect, which is computable within a reasonable time constraint.

Lemma 7.4.2. *There is a PTIME computable boolean query q such that $q(\mathbf{A}(\mathbf{G})) \neq q(\mathbf{B}(\mathbf{G}))$ for any finite connected graph $\mathbf{G}$ of degree $n+1$.*

Proof. Structures $\mathbf{A}(\mathbf{G})$ and $\mathbf{B}(\mathbf{G})$ are non-isomorphic. Hence it is enough to show that a boolean query q, for which

$$q(\mathcal{M}) = 1 \Leftrightarrow \mathcal{M} \cong \mathbf{A}(\mathbf{G})$$

for any finite connected graph of degree $(n+1)$ as well as some choice of ordering $<^{\mathcal{M}}$ and functions h_u enumerating neighbours of vertices, is computable in $PTIME$. Suppose we are given a finite structure $\mathcal{M}$. First of all one has to check, if $\mathcal{M}$ is isomorphic with $\mathbf{A}(\mathbf{G})$ or $\mathbf{B}(\mathbf{G})$ for some appropriate $\mathbf{G}$. In other words, one checks whether $\mathcal{M}$ is even a structure $\mathcal{C}(\mathbf{G}, S)$ for some $\mathbf{G}$ and $S \subseteq G$. This can be done in polynomial time with respect to $|dom(\mathcal{M})|$. Of course, if $\mathcal{M}$ fails this test, we assign $q(\mathcal{M}) = 0$. Otherwise $\mathcal{M}$ is isomorphic to either $\mathbf{A}(\mathbf{G})$ or $\mathbf{B}(\mathbf{G})$, so we may assume $\mathcal{M} = \mathbf{A}(\mathbf{G})$ or $\mathcal{M} = \mathbf{B}(\mathbf{G})$.

Let $U \subseteq G \times B$ be such that for all $v \in G$ and $1 \leq i \leq n+1$, it contains one and only one element of the form (v, x_i), where $x_i = c_i$ or $x_i = d_i$. Moreover, suppose membership of U is preserved by the edge relation E i.e. if

$$((v, x_i), (v', y_j)) \in E^{\mathcal{M}},$$

then

$$\big((v, x_i) \in U \Leftrightarrow (v', y_j) \in U\big).$$

The construction of such U takes only a polynomial time. The set U can be constructed for example by starting from the vertex u and arbitrarily choosing elements (u, x_i) to be included in U. Then for any neighbour v of u, it is determined whether (v, c_i) or (v, d_i) is put to U , where $i = h_u(v)$. Then one moves to the next vertex according to the ordering $<^{\mathcal{M}}$ and again, arbitrarily chooses those elements, which are not already in U. By continuing recursively, eventually one ends up with the set U. Clearly this can be done in a way respecting the time constraint, since the arbitrary choices one makes in the construction do not have any essential effect on the outcome.

Now for each $v \in G$, U determines an $(n+1)$-tuple $\bar{x}_v = (x_1, ..., x_{n+1}) \in B^{n+1}$, such that

$$x_1 \prec ... \prec x_{n+1} \text{ and } ((v, x_1), ..., (v, x_{n+1})) \in U.$$

So U contains some $(n+1)$-tuple from each building block and the tuple has either c_i or d_i component from each index $1 \leq i \leq n+1$. Furthermore, the fact that U preserves the edge relation leads to the following observation. Every time we added an element of the form (v, c) to U, another element (w, c) became automatically included to U. Hence U contains an even number of elements of the form (v, c_i). Similarly, U contains an even number of elements of the form (v, d_i). Let $\bar{x}_V$ denote a tuple, which is formed by concatenating all the tuples $\bar{x}_v$, $v \in G$. Now by the above it holds that the number of c's and the number of d's in $\bar{x}_V$ are both even.

Define S to be the set that contains all the vertices $v \in G$ such that $\bar{x}_v \notin R^{\mathcal{M}}$. This set S can be constructed from the set U in polynomial time.

Recall that $\mathcal{M} = \mathbf{A}(\mathbf{G})$ or $\mathcal{M} = \mathbf{B}(\mathbf{G})$ and assume first that $\mathcal{M} = \mathbf{A}(\mathbf{G})$. For all $v \in G$, it holds that $v \notin \emptyset$, and thus by definition, $\bar{x}_v \notin R^{\mathcal{M}} \Leftrightarrow \bar{x}_v \in R^-$. Hence S contains all such $v \in G$ that $\bar{x}_v \in R^-$. Moreover, $\bar{x}_v \in R^-$ if and only if there is an odd number of d elements in $\bar{x}_v$. This means that for each $\bar{x}_v$, for which $v \notin S$, there is an even number of d elements in $\bar{x}_v$. Now if erase all such tuples $\bar{x}_v$ that $v \notin S$ from the tuple $\bar{x}_V$, we get the tuple, which contains all such $\bar{x}_v$ that $v \in S$. Now consider the overall number of d elements in tuples $\bar{x}_v, v \in S$. We subtracted an even number from an even number, and hence there is altogether an even number of d elements in the tuples $\bar{x}_v, v \in S$. Any $\bar{x}_v, v \in S$, itself contains an odd number of d elements, and therefore $|S|$ has to be even.

Assume then that $\mathcal{M} = \mathbf{B}(\mathbf{G})$. In this case S contains all such $v \in G, v \neq u$, that $\bar{x}_v \in R^-$. The vertice u belongs to S, if $\bar{x}_u \in R^+$, and otherwise $u \in G \setminus S$. Assume first that u is in S. By the same argument as above, there is altogether an even number of d elements in those $\bar{x}_v$ that $v \in S$. Since $u \in S$, there is an even number of d elements in $\bar{x}_u$. Thus there is an even number of vertices v, $v \neq u$, in S. By assumption, also u is in S, and therefore $|S|$ is odd. Suppose then that u is not in S. Now $\bar{x}_u \in R^-$, which means that $\bar{x}_u$ contains an odd number of d elements. Now erase from $\bar{x}_V$ those tuples $\bar{x}_v$, $v \neq u$, for which $v \notin S$. In this tuple obtained, there is an even number of d elements. Then erase $\bar{x}_u$ from this tuple. By assumption on u, we are then left with a tuple, which contains an odd number of d elements. But this means that $|S|$ is odd.

Therefore we have that

$$q(\mathcal{M}) = 1 \Leftrightarrow |S| \text{ is even.}$$

This accomplishes the proof, since the parity of $|S|$ can be decided in polynomial time.

□

We have now obtained all the necessary results to conclude this thesis with the main theorem.

Theorem 7.4.3. *PTIME cannot be captured with* $\mathcal{L}^{\omega}_{\infty\omega}(\mathbf{Q}_n)$.

Proof. It is enough to show that for each $n \in \mathbb{N}$ there is a *PTIME* computable boolean query, which cannot be defined in the logic $\mathcal{L}^{\omega}_{\infty\omega}(\mathbf{Q}_n)$. By Lemma 7.4.1, for each n and $k > n$, there is a finite connected graph $\mathbf{G}$ of degree $n+1$ such that player II has a winning strategy in $CR^k_n(\mathbf{G})$. Applying Lemma 7.3.2, this means that player II has a winning strategy in $BP^k_n(\mathbf{A}(\mathbf{G}), (\mathbf{B}(\mathbf{G}))$. On the other hand, by Lemma 7.4.2, there is a *PTIME* computable boolean query q such that $q(\mathbf{A}(\mathbf{G})) \neq q(\mathbf{B}(\mathbf{G}))$, and by Theorem 5.2.8, q is not definable in $\mathcal{L}^{\omega}_{\infty\omega}(\mathbf{Q}_n)$.

□

References

1 J. Barwise, and S. Feferman, editors, *Model-Theoretic logics*, Springer-Verlag Berlin (1985)

2 J. Cai, M. Fürer, and N. Immerman, An Optimal Lower Bound on the Number of Variables for Graph Identification, in Proceedings 30^{th} IEEE Symposium on Foundations of Computer Science, pp. 612-617 (1989)

3 A. Chandra, and D. Harel, Structure and Complexity of Relational Queries, in *Journal of Computer and System Sciences* **25**, pp.99-128 (1982).

4 H.-D. Ebbinghaus, and J. Flum, *Finite Model Theory*, 2^{nd} edition, Springer-Verlag Berlin/Heidelberg/New York, (1999)

5 R. Fagin, Generalized First-order Spectra and Polynomial-time Recognizable Sets, in R.M. Karp, Ed., in *Complexity of Computation*, SIAM-AMS Proceedings, Vol 7 (1974).

6 Y. Gurevich, Logic and the Challenge of Computer Science, in *Trends in Theoretical Computer Science*, pp.1-57 E.Börger, editor, Computer Science pres, New York (1988)

7 L. Hella, Logical Hierarchies in PTIME, in *Information and Computation* **129**, pp. 1-19 (1996).

8 N. Immerman, Relational Queries Computable in Polynomial Time, in *Information and Control* **68**, pp. 86-104 (1986)

9 N. Immerman, Languages which Capture Complexity Classes, in *Journal on computing* SIAM **16**, pp. 760-778 (1987).

10 Ph. G. Kolaitis, and J. Väänänen, Generalized quantifiers and pebble games on finite structures, in *Annals of Pure and Applied Logic* **74** pp. 23-75 (1995).

11 Ph. G. Kolaitis, and M. Vardi, Infinitary logics and 0-1 laws, in *Information and Computation* **98**, pp. 258-294 (1992).

12 P. Lindström, First-order Predicate Logic with Generalized Quantifiers, in *Theoria* **32**, pp. 186-195 (1966).

13 P. Lindström, On extensions of Elementary Logic, in *Theoria* **35**, pp.1-11 (1969).

14 A. Mostowski, On a Generalization of Quantifiers, in *Fundamenta mathematicae* **44**, pp.12-36 (1957).

15 M. Vardi, The Complexity of Relational Query Languages, in *Proceedings,* 14^{th} ACM Symposium on Theory of computing, pp. 137-146 (1982).