

# Capturing Relativized Complexity Classes without Order

**Anuj Dawar**

Department of Computer Science  
University of Wales, Swansea  
Swansea SA2 8PP, U.K.  
a.dawar@swansea.ac.uk

**Georg Gottlob**

Department of Information Systems  
TU Wien, Paniglgasse 16  
A-1040 Vienna, Austria  
gottlob@dbai.tuwien.ac.at

**Lauri Hella**

Department of Mathematics  
P.O. Box 4 (Yliopistonkatu 5)  
00014 University of Helsinki, Finland  
hella@cc.helsinki.fi

March 7, 1997

## Abstract

We consider the problem of obtaining logical characterisations of oracle complexity classes. In particular, we consider the complexity classes  $\text{LOGSPACE}^{NP}$  and  $\text{PTIME}^{NP}$ . For these classes, characterisations are known in terms of NP computable Lindström quantifiers which hold on ordered structures. We show that these characterisations are unlikely to extend to arbitrary (unordered) structures, since this would imply the collapse of certain exponential complexity hierarchies. We also observe, however, that  $\text{PTIME}^{NP}$  can be characterised in terms of Lindström quantifiers (not necessarily NP computable), though it remains open whether this can be done for  $\text{LOGSPACE}^{NP}$ .

## 1 Introduction

Since Fagin showed that existential second order logic captures the class NP [7], and Immerman and Vardi characterised PTIME in terms of least fixed point logic [14, 25], a large number of complexity classes have been given logical characterisations, and a tight correspondence has been established between logical expressibility and computational complexity. The results that relate logic to complexity in this way generally fall into two classes. There are those in which the correspondence holds over *all* finite structures — the paradigmatic example is Fagin’s theorem; and there are those where the correspondence holds only over structures whose logical relations include a linear order over the domain — in the style of the Immerman-Vardi result.

It has often been said that complexity classes from NP and above can be captured by a logic even in the absence of a linear order, while the classes below cannot be so captured, because the corresponding logic is too weak to construct the order, which is necessary in

order to simulate computation. Indeed, there is no class below NP for which a logic is known that captures that class over all structures. In this paper we show that the situation is a little more subtle with respect to relativized complexity classes, i.e., complexity classes defined with respect to oracles. In particular, the class  $\text{LOGSPACE}^{\text{NP}}$ , which contains NP and which has a natural logical characterisation when an order is available [22, 23, 8] appears not to be capturable without the restriction to ordered structures. Indeed, we show that the characterisations that work on ordered structures are unlikely to work in the absence of order, as this would imply the collapse of certain complexity theoretic hierarchies.

More generally, it appears that there is a trade-off between the power of the oracle and the complexity of the machine. The more powerful the oracle, the weaker we can make the basic machine while still having a capturable complexity class. With the empty oracle, neither LOGSPACE nor PTIME seems to be logically characterizable. With oracles in NP,  $\text{PTIME}^{\text{NP}}$  can be captured by a logic, but this seems unlikely for  $\text{LOGSPACE}^{\text{NP}}$ . However, we show that with  $\Sigma_2^p$  oracles, both  $\text{LOGSPACE}^{\Sigma_2^p}$  and  $\text{PTIME}^{\Sigma_2^p}$  can be captured.

## 2 Background and Notation

A signature  $\sigma = \langle R_1, \dots, R_m \rangle$  is a finite sequence of relation symbols,  $R_i$ , each with an associated arity  $n_i$ . A structure  $\mathcal{A} = \langle A, R_1^{\mathcal{A}}, \dots, R_m^{\mathcal{A}} \rangle$  over signature  $\sigma$ , consists of a universe  $A$  and relations  $R_i^{\mathcal{A}} \subseteq A^{n_i}$  interpreting the relation symbols in  $\sigma$ . Unless otherwise stated, we will assume that the universe of every structure considered is finite. We write  $|\mathcal{A}|$  to denote the universe of the structure  $\mathcal{A}$ , and  $\text{card}(S)$  for the cardinality of a set  $S$ . We will assume, in general, that the universe of  $\mathcal{A}$  is an initial segment of the natural numbers, i.e.,  $|\mathcal{A}| = n = \{0, \dots, n-1\}$  for some  $n$ . In the special case when the signature  $\sigma$  is empty, we call  $\mathcal{A}$  a pure set of size  $n$ , denoted by  $\langle n \rangle$ .

The basic equality type of a tuple  $s = \langle a_1, \dots, a_k \rangle$  in a model  $\mathcal{A}$  is the quantifier free formula  $\bigwedge_{(i,j) \in S} (x_i = x_j) \wedge \bigwedge_{(i,j) \in T} \neg(x_i = x_j)$ , where  $S = \{(i, j) \mid a_i = a_j\}$  and  $T = \{(i, j) \mid a_i \neq a_j\}$ . Note that in a pure set  $\langle n \rangle$  each tuple is described up to isomorphism by its basic equality type.

An ( $m$ -ary) *query*  $q$  (also sometimes called a global relation) is a map from structures (over some fixed signature  $\sigma$ ) to ( $m$ -ary) relations on the structures, that is closed under isomorphism. That is, if  $\langle a_1, \dots, a_m \rangle \in q(\mathcal{A})$ , and  $f$  is an isomorphism from  $\mathcal{A}$  to  $\mathcal{B}$ , then  $\langle f(a_1), \dots, f(a_m) \rangle \in q(\mathcal{B})$ . A 0-ary query is also called a *Boolean query*. When we refer to the complexity of a query  $q$ , we mean the complexity of the decision problem: given a structure  $\mathcal{A}$  and a tuple  $s$  of elements of  $\mathcal{A}$ , is it the case that  $s \in q(\mathcal{A})$ ? Here, the complexity of a query is always measured in terms of the size of the structure, i.e. the cardinality of its universe.

The  $m$ -ary query defined by a formula  $\varphi$  with free variables among  $x_1, \dots, x_m$  maps a structure  $\mathcal{A}$  to the relation  $\{s \in |\mathcal{A}|^m \mid \mathcal{A} \models \varphi[s]\}$ . We say that a query is *expressible* (or definable) in a logic  $L$  if there is some formula of  $L$  that defines it.

We write FO, LFP, *etc.* both to denote logics (i.e., sets of formulas) and the collections of queries that are expressible in the respective logics. By a class of structures, we mean a collection of structures that is closed under isomorphisms of the structures (or equivalently,

a Boolean query). We say that a logic  $L$  captures a complexity class  $\mathcal{C}$  if a query is definable in  $L$  if, and only if, it is in  $\mathcal{C}$ .

## 2.1 Inductive and Infinitary Logic

Let  $\varphi$  be a first order formula in the signature  $\sigma \setminus \langle R \rangle$ , where  $R$  is  $k$ -ary. On a  $\sigma$ -structure  $\mathcal{A}$ ,  $\varphi$  defines the operator,  $\Phi_{\mathcal{A}}(R^{\mathcal{A}}) = \{s \in |\mathcal{A}|^k \mid \langle \mathcal{A}, R^{\mathcal{A}} \rangle \models \varphi[s]\}$ . If  $\varphi$  is an  $R$ -positive formula (that is, all occurrences of  $R$  in  $\varphi$  are within the scope of an even number of negations), then  $\Phi_{\mathcal{A}}$  is monotone, and has a least fixed point. This least fixed point can be obtained by iterating the operator  $\Phi_{\mathcal{A}}$  as follows:  $\varphi_{\mathcal{A}}^0 = \emptyset$ ;  $\varphi_{\mathcal{A}}^{m+1} = \Phi_{\mathcal{A}}(\varphi_{\mathcal{A}}^m)$ . The  $m^{\text{th}}$  stage of the induction determined by  $\varphi$  can be uniformly defined over all structures by a first order formula which we denote by  $\varphi^m$ . The set inductively defined by  $\varphi$  on  $\mathcal{A}$ , denoted  $\varphi_{\mathcal{A}}^{\infty}$ , is the least fixed point of the operator  $\Phi_{\mathcal{A}}$ , that is,  $\varphi_{\mathcal{A}}^{\infty} = \varphi_{\mathcal{A}}^m$ , where  $m = \|\varphi\|_{\mathcal{A}}$  is the least natural number such that  $\varphi_{\mathcal{A}}^{m+1} = \varphi_{\mathcal{A}}^m$ . Observe that, because the stages of the induction are increasing, and because there are only  $n^k$  distinct  $k$ -tuples, where  $n$  is the cardinality of  $|\mathcal{A}|$ , it must be the case that  $\|\varphi\|_{\mathcal{A}} \leq n^k$ .

We write LFP for the extension of first order logic with the **lfp** operation which uniformly determines the least fixed point of an  $R$ -positive formula. That is, for any  $R$ -positive formula  $\varphi$ , **lfp**( $R, x_1, \dots, x_k$ ) $\varphi(x_1, \dots, x_k)$  is a formula of LFP and  $\mathcal{A} \models \mathbf{lfp}(R, x_1, \dots, x_k)\varphi[s]$  if, and only if,  $s \in \varphi_{\mathcal{A}}^{\infty}$ .

Even if  $\varphi$  is not  $R$ -positive, we can define an induction, the stages of which are increasing, by iterating the inflationary operator  $\Phi'_{\mathcal{A}}$  given by  $\Phi'_{\mathcal{A}}(R^{\mathcal{A}}) = \Phi_{\mathcal{A}}(R^{\mathcal{A}}) \cup R^{\mathcal{A}}$ . We call the fixed point obtained in this way the *inflationary fixed point* of  $\varphi$ . We write IFP for the extension of first order logic with the **ifp** operation, which uniformly defines the inflationary fixed point of a formula. That is, the relational expression **ifp**( $R, x_1, \dots, x_k$ ) $\varphi$  denotes the inflationary fixed point of  $\varphi$ . Gurevich and Shelah [11] showed that on finite structures, IFP is equivalent to LFP. Immerman [14] and Vardi [25] independently showed that when we include a total ordering on the domain as part of the logical vocabulary, the language LFP expresses exactly the class of polynomial time computable properties:

**Theorem 2.1** ([14],[25]) *On ordered finite structures, LFP = PTIME.*

If we take an arbitrary formula  $\varphi$  and iterate the corresponding operator  $\Phi_{\mathcal{A}}$ , the sequence of stages may not be increasing and therefore may or may not converge to a fixed point. Define the *partial fixed point* of  $\varphi$  to be  $\varphi_{\mathcal{A}}^m$  for the least  $m$  such that  $\varphi_{\mathcal{A}}^{m+1} = \varphi_{\mathcal{A}}^m$ , if such an  $m$  exists, and empty otherwise. Because there are only  $2^{n^k}$  sets of  $k$ -tuples over a structure of size  $n$ , if such an  $m$  exists, then  $m \leq 2^{n^k}$ . We can then define another logic called PFP which extends first order logic by the partial fixed point operator **pfp**, similar to the operator **ifp**. The relational expression **pfp**( $R, x_1, \dots, x_k$ ) $\varphi$  denotes the partial fixed point of  $\varphi$ . It has been shown that on ordered structures, the logic PFP captures the complexity class PSPACE [25, 1].

Let  $L^k$  be the fragment of first order logic which consists of those formulas whose variables, both free and bound, are among  $x_1, \dots, x_k$ . Let  $L_{\infty\omega}^k$  be the closure of  $L^k$  under the operations of conjunction and disjunction applied to arbitrary (finite or infinite) sets of formulas. Let  $L_{\infty\omega}^{\omega} = \bigcup_{k \in \omega} L_{\infty\omega}^k$ . The logic  $L_{\infty\omega}^{\omega}$  was introduced by Barwise in [2]. Kolaitis and Vardi [17] showed that LFP and PFP are fragments of  $L_{\infty\omega}^{\omega}$  on the class of all finite structures.

## 2.2 Generalized Quantifiers

Let  $C$  be any collection of structures over the signature  $\sigma = \langle R_1, \dots, R_m \rangle$  (where  $R_i$  has arity  $n_i$ ) that is closed under isomorphism. We associate with  $C$  the generalized quantifier  $Q_C$ . For a logic  $L$ , define the extension  $L(Q_C)$  by closing the set of formulas of  $L$  under the following formula formation rule: if  $\varphi_1, \dots, \varphi_m$  are formulas of  $L(Q_C)$  and  $\bar{x}_1, \dots, \bar{x}_m$  are tuples of variables with the length of  $\bar{x}_i$  being  $n_i$ , then  $Q_C \bar{x}_1 \dots \bar{x}_m (\varphi_1, \dots, \varphi_m)$  is a formula of  $L(Q_C)$ . Here the quantifier  $Q_C \bar{x}_1 \dots \bar{x}_m$  binds only those occurrences of the variables among  $\bar{x}_i$  which are in  $\varphi_i$ ; all other free occurrences of variables remain free. The semantics of the quantifier is given by:  $\langle \mathcal{A}, s \rangle \models Q_C \bar{x}_1 \dots \bar{x}_m (\varphi_1(\bar{x}_1, \bar{y}_1), \dots, \varphi_m(\bar{x}_m, \bar{y}_m))$ , if and only if,  $\langle |\mathcal{A}|, \varphi_1^{\mathcal{A}}[s_1], \dots, \varphi_m^{\mathcal{A}}[s_m] \rangle \in C$ , where  $\varphi_i^{\mathcal{A}}[s_i] = \{t \in |\mathcal{A}|^{n_i} \mid \mathcal{A} \models \varphi_i[t, s_i]\}$ .

We are primarily interested in vectorized quantifiers. Given a class of structures  $C$ , let  $C_k$  be the class of all structures  $\langle A, S_1, \dots, S_m \rangle$  such that  $S_i \subseteq A^{k n_i}$  and  $\langle A^k, S_1^{(k)}, \dots, S_m^{(k)} \rangle \in C$ , where  $S_i^{(k)}$  is the relation  $S_i$  thought of as an  $n_i$ -ary relation on  $A^k$ . Then, the extension of a logic  $L$  with the set of quantifiers  $\{Q_{C_k} \mid k \in \omega\}$  is denoted  $L(\overline{Q}_C)$ .

For a set of generalized quantifiers  $\mathbf{Q}$ , we write  $L(\mathbf{Q})$  for the extension of the logic  $L$  by all the quantifiers in  $\mathbf{Q}$ . Thus, for instance,  $\text{FO}(\mathbf{Q})$  denotes the extension of first order logic by the generalized quantifiers in the set  $\mathbf{Q}$ . Note, however, that  $\text{LFP}(\mathbf{Q})$  is not well-defined for an arbitrary set  $\mathbf{Q}$  of quantifiers. This is because, in the presence of non-monotone quantifiers, positivity of a formula  $\varphi$  is no longer a guarantee for monotonicity of the corresponding operator  $\Phi_{\mathcal{A}}$ . We will avoid this problem by considering logics of the form  $\text{IFP}(\mathbf{Q})$  instead of  $\text{LFP}(\mathbf{Q})$ .

## 2.3 Some Complexity Classes

PTIME and NP denote the classes of all languages recognizable in polynomial time by a deterministic and nondeterministic Turing machine, respectively. The class ETIME (NETIME) consists of all languages recognizable by a deterministic (nondeterministic) Turing machine in time  $O(2^{kn})$ , where  $n$  is the size of the input and  $k$  is some constant. Note that ETIME is not closed under polynomial time many-one reductions (short,  $\leq_m^p$ -reductions). Therefore one often prefers to consider the more robust class EXPTIME which consists of all languages recognizable by a deterministic Turing machine in time  $O(2^{n^k})$ , where  $n$  is the size of the input and  $k$  is some constant. ETIME is a proper subset of EXPTIME. Moreover, EXPTIME is identical to the closure of ETIME under  $\leq_m^p$ -reductions. Analogously, NEXPTIME is the nondeterministic version of EXPTIME and is equal to the closure of NETIME under  $\leq_m^p$ -reductions.

LOGSPACE, LINSPEACE and PSPACE denote the classes of all languages recognizable by deterministic Turing Machines using logarithmic, linear, and polynomial workspace, respectively. Note that PSPACE is the closure of LINSPEACE under  $\leq_m^p$ -reductions. NLOGSPACE is the nondeterministic version of LOGSPACE. NLINSPEACE is the non-deterministic version of LINSPEACE. It is currently not known whether LOGSPACE = NLOGSPACE or whether LINSPEACE = NLINSPEACE. On the other hand, PSPACE coincides with nondeterministic PSPACE.

If  $C$  is a machine-based complexity class and  $D$  is any complexity class, then  $C^D$  denotes the class of all languages recognizable by a  $C$  Turing machine having access to an oracle  $A$  in  $D$ . The notion of relativization (i.e., of oracle access) is the standard notion due to Ladner and Lynch [18]. In particular, the query tape is erased after a

$\text{EBH} = \bigcup_i \text{EBH}_i, \text{ where:}$ $\text{EBH}_1 = \text{NETIME}$ $\text{EBH}_{2i} = \{A \cap B \mid A \in \text{EBH}_{2i-1} \text{ and } B \in \text{NETIME}\}$ $\text{EBH}_{2i+1} = \{A \cup B \mid A \in \text{EBH}_{2i} \text{ and } B \in \text{NETIME}\}$	$\text{EXPBH} = \bigcup_i \text{EXPBH}_i, \text{ where:}$ $\text{EXPBH}_1 = \text{NEXPTIME}$ $\text{EXPBH}_{2i} = \{A \cap B \mid A \in \text{EXPBH}_{2i-1} \text{ and } B \in \text{NEXPTIME}\}$ $\text{EXPBH}_{2i+1} = \{A \cup B \mid A \in \text{EXPBH}_{2i} \text{ and } B \in \text{NEXPTIME}\}$
--	--

Table 1: Definition of exponential Boolean hierarchies EBH and EXPBH.

$\text{EH} = \bigcup_i \Sigma_i^e, \text{ where:}$ $\Sigma_0^e = \text{ETIME}$ $\Sigma_i^e = \text{NETIME}^{\Sigma_i^p}, \text{ for } i > 0$ $\Pi_i^e = \text{co-}\Sigma_i^e$	$\text{EXPH} = \bigcup_i \Sigma_i^{\text{exp}}, \text{ where:}$ $\Sigma_0^{\text{exp}} = \text{EXPTIME}$ $\Sigma_i^{\text{exp}} = \text{NEXPTIME}^{\Sigma_i^p}, \text{ for } i > 0$ $\Pi_i^{\text{exp}} = \text{co-}\Sigma_i^{\text{exp}}$
---	--

Table 2: Definition of exponential hierarchies EH and EXPH.

query is answered; moreover, the oracle query strings of a space bounded machine are not themselves subject to the space bound.

The complement of a language  $A$  is denoted by  $\bar{A}$ . For a complexity class  $C$ ,  $\text{co-}C$  denotes the class  $\{A \mid \bar{A} \in C\}$ .

The *Boolean Hierarchy over NP* (or simply the *Boolean Hierarchy*), denoted by BH, consists of all languages that can be recognized by evaluating a Boolean combination of NP queries. More formally, BH is the union of all classes  $\text{BH}_j$  defined as follows:

$$\begin{aligned} \text{BH}_1 &= \text{NP} \\ \text{BH}_{2i} &= \{A \cap \bar{B} \mid A \in \text{BH}_{2i-1} \text{ and } B \in \text{NP}\} \\ \text{BH}_{2i+1} &= \{A \cup B \mid A \in \text{BH}_{2i} \text{ and } B \in \text{NP}\}. \end{aligned}$$

The *Polynomial Hierarchy*, denoted by PH, is the union of all classes  $\Sigma_i^p$  and  $\Pi_i^p$  for  $0 \leq i$ , where  $\Sigma_0^p = \Pi_0^p = \text{PTIME}$  and for each  $i \geq 0$ ,  $\Sigma_{i+1}^p = \text{NP}^{\Sigma_i^p}$  and  $\Pi_i^p = \text{co-}\Sigma_i^p$ .

An interesting class contained in the Polynomial Hierarchy and containing the Boolean Hierarchy is  $\text{LOGSPACE}^{\text{NP}}$ . Several different characterizations of this class exist, for an overview see [27]. In particular,  $\text{LOGSPACE}^{\text{NP}}$  is identical to the class  $\text{PTIME}^{\text{NP}[\mathcal{O}(\log n)]}$  of languages recognizable in polynomial time with a logarithmic number of queries to an oracle in NP.

The Boolean Hierarchy and the Polynomial Hierarchy have analogues at the exponential level. In particular, NETIME gives rise to the (linear) exponential Boolean hierarchy EBH and to the (linear) exponential hierarchy EH. In turn, NEXPTIME gives rise to the (full) exponential Boolean hierarchy EXPBH and to the (full) exponential hierarchy EXPH<sup>1</sup>. The exact definition of these hierarchies and their classes is given in Tables 1 and 2. Concerning the definitions of  $\Sigma_i^e$  and  $\Sigma_i^{\text{exp}}$  in Table 2, note that a  $\Sigma_{i+1}^e$  Turing machine may ask exponentially long queries to its  $\Sigma_i^p$  oracle, similarly for a  $\Sigma_{i+1}^{\text{exp}}$  Turing machine.

For each complexity class  $C \subseteq \text{PH}$  defined in this paper we define the *linear exponential version*  $E(C)$  and the *full exponential version*  $\text{Exp}(C)$  in Table 3.

The following proposition is well-known. The proof is by simple padding arguments.

<sup>1</sup>The ETIME and EXPTIME hierarchies are sometimes referred to as the *weak ETIME hierarchy* and the *weak EXPTIME Hierarchy*, respectively. They should not be confounded with the *Strong Exponential Time Hierarchy* studied in [12].

Basic class $C$	Linear exponential version $E(C)$	Full exponential version $EXP(C)$
PTIME	ETIME	EXPTIME
NP	NETIME	NEXPTIME
$\Sigma_1^P$	$\Sigma_1^E$	$\Sigma_1^{EXP}$
$\Pi_1^P$	$\Pi_1^E$	$\Pi_1^{EXP}$
PH	EH	EXPH
$BH_i$	$EBH_i$	$EXPBH_i$
BH	EBH	EXPBH
LOGSPACE	LINSPACE	PSPACE
NLOGSPACE	NLINSACE	PSPACE
$LOGSPACE^{NP}$	$LINSPACE^{NP}$	$PSPACE^{NP}$

Table 3: Exponential versions of basic classes.

**Proposition 2.2** *For each basic class  $C$  appearing in the first column of Table 3, the closure under  $\leq_m^P$ -reductions of  $E(C)$  is equal to  $Exp(C)$ .*

For a natural number  $n$ ,  $bin(n)$  denotes its standard binary encoding. If  $A$  is a language over  $\{0, 1\}^*$ , denote by  $1A$  the set of all words in  $A$  prefixed with 1. The *tally version* of  $A$  is the language  $tally(A) = \{1^n | bin(n) \in 1A\}$ .

It is well-known that there is an exponential jump in complexity if we proceed from the tally version to the binary version of a language (see [9]).

**Proposition 2.3** *Let  $C$  be any class appearing in the first column of Table 3. It holds that for each language  $A$ ,  $tally(A) \in C$  iff  $A \in E(C)$ .*

If  $C$  is a complexity class, then a  $C$  *quantifier* is a generalized quantifier (i.e., a set of structures) in  $C$ . In particular, we will deal with NP quantifiers and with NLOGSPACE quantifiers in this paper.

## 2.4 Capturing Complexity Classes

As observed earlier, Theorem 2.1 crucially depends on the presence of a linear order in the structures considered. If arbitrary structures are considered, then LFP is too weak to capture PTIME. It remains an open question whether there is some logic that captures PTIME over arbitrary structures. Similarly, it is also not known if there is any logic that captures the class LOGSPACE. Indeed, no logical characterisation is known for any complexity class below NP.

On the other hand, NP and many complexity classes above it have been shown to be captured by appropriate logics. One exception is  $LOGSPACE^{NP}$ , for which the known logical characterizations hold only for ordered structures. In particular, Stewart [22, 23] has shown that the logic  $FO(\overline{Ham})$  (i.e., first order logic extended with vectorized versions of the Hamiltonicity quantifier) captures  $LOGSPACE^{NP}$  on ordered structures. Gottlob [8] extended this result and showed that for a large number of natural complexity classes  $C$  (among which POLYLOGSPACE, all classes of the Polynomial hierarchy, and all classes of the Exponential Hierarchy), the following holds: If a set  $Q$  of quantifiers is complete for  $C$  under first order reductions, then  $FO(Q)$  captures  $LOGSPACE^C$  on ordered structures (related results can be found in [19, 6]). It was posed as an open question in [8] whether this result extends to arbitrary structures. We show in this paper that this result is unlikely to

extend to arbitrary structures, in as much as capturing  $\text{LOGSPACE}^{\text{NP}}$  by first order logic with NP quantifiers would imply the collapse of the Boolean hierarchy over  $\text{NEXPTIME}$ . (When we speak about a *collapse* of a hierarchy, we mean a collapse to some finite level, but not necessarily to the first.)

Of course, it remains difficult to prove negative results – i.e., that some complexity classes cannot be captured by any logic. Indeed, showing such a result for  $\text{LOGSPACE}^{\text{NP}}$  would separate many complexity classes (not least of all, it would separate P from NP), since (see Section 6) any complexity class containing  $\text{PTIME}^{\text{NP}}$  that is closed under compositions is captured by some logic. Moreover, it follows from results in [4] that if  $\text{LOGSPACE}^{\text{NP}}$  is captured by any logic, then it is captured by one that is an extension of first order logic by a single vectorized generalized quantifier (though not necessarily an NP quantifier).

### 3 A Normal Form Result

Let  $\mathbf{Q}$  be a set of generalized quantifiers. Recall that  $L_{\infty\omega}^\omega(\mathbf{Q})$  denotes the extension of  $L_{\infty\omega}^\omega$  by the quantifiers in  $\mathbf{Q}$ . Note that if  $\mathbf{Q}$  is infinite, then a formula of this logic may contain occurrences of infinitely many different quantifiers in  $\mathbf{Q}$ . We will restrict our attention to the fragment of  $L_{\infty\omega}^\omega(\mathbf{Q})$  which consists of formulas containing only finitely many different quantifiers (but a single quantifier is allowed to have infinitely many distinct occurrences).

**Definition 3.1** *Let  $\mathbf{Q}$  be a set of quantifiers.  $L^*(\mathbf{Q})$  is the logic consisting of all formulas  $\varphi$  that belong to  $L_{\infty\omega}^\omega(\mathbf{Q}_0)$  for some finite subset  $\mathbf{Q}_0$  of  $\mathbf{Q}$ .*

Our aim is to prove that, on the class of pure sets,  $L^*(\mathbf{Q})$  collapses to a small fragment of  $\text{FO}(\mathbf{Q})$  consisting of formulas that do not involve any nesting of the quantifiers in  $\mathbf{Q}$ . The proof of this normal form result is heavily based on the analysis of  $L_{\infty\omega}^k(\mathbf{Q})$ -equivalence types that was carried out by Dawar and Hella in [5]. In fact, the collapse of  $L^*(\mathbf{Q})$  to  $\text{FO}(\mathbf{Q})$  on pure sets was already proved in [5], but without giving any explicit normal form.

**Definition 3.2** *Let  $\varphi$  be a formula of  $L^*(\mathbf{Q})$ .*

1.  $\varphi$  is a basic flat formula if it is either atomic, or of the form  $Q\bar{x}_1 \dots \bar{x}_m(\beta_1, \dots, \beta_m)$  for some  $Q \in \mathbf{Q}$  and quantifier free formulas  $\beta_1, \dots, \beta_m$ .
2.  $\varphi$  is in flat normal form if it is obtained from basic flat formulas by successive applications of Boolean operations and first order quantifications.

**Theorem 3.3** *Let  $\mathbf{Q}$  be a set of quantifiers. For any formula  $\varphi$  of  $L^*(\mathbf{Q})$  there exists a formula  $\psi$  of  $\text{FO}(\mathbf{Q})$  in flat normal form such that  $\varphi$  and  $\psi$  are equivalent on the class of pure sets.*

**Proof.** Let  $\varphi(x_1, \dots, x_l)$  be a formula of  $L^*(\mathbf{Q})$  over the empty vocabulary. Thus, there is a  $k < \omega$  and a finite  $\mathbf{Q}_0 \subseteq \mathbf{Q}$  such that  $\varphi$  belongs to  $L_{\infty\omega}^k(\mathbf{Q}_0)$ .

In [5] it was proved that each pure set<sup>2</sup>  $\langle n \rangle$  can be characterized up to  $L_{\infty\omega}^k(\mathbf{Q}_0)$ -equivalence by a sentence of the form

$$\begin{aligned} \eta_n = & \bigwedge_{1 \leq i \leq m} \exists x_1 \dots \exists x_k \psi_i \wedge \\ & \forall x_1 \dots \forall x_k \bigvee_{1 \leq i \leq m} \psi_i \wedge \\ & \bigwedge_{1 \leq j \leq r} \forall x_1 \dots \forall x_k (\varphi_j \leftrightarrow \gamma_j), \end{aligned}$$

where the formulas  $\psi_i$  are basic equality types, the formulas  $\gamma_j$  are disjunctions of basic equality types, and each of the formulas  $\varphi_j$  is of the form  $Q\bar{x}_1 \dots \bar{x}_m(\beta_1, \dots, \beta_m)$  for some  $Q \in \mathbf{Q}_0$  and quantifier free formulas  $\beta_1, \dots, \beta_m$ . That is, for every  $n' < \omega$ ,  $\langle n' \rangle \models \eta_n$ , if and only if,  $\langle n' \rangle$  and  $\langle n \rangle$  satisfy the same sentences of  $L_{\infty\omega}^k(\mathbf{Q}_0)$ . Furthermore, the  $L_{\infty\omega}^k(\mathbf{Q}_0)$ -equivalence type of each  $l$ -tuple  $t \in n^l$  can be defined by a formula of the form

$$\eta_{n,t}(x_1, \dots, x_l) = \eta_n \wedge \forall x_{l+1} \dots \forall x_k \bigvee_{1 \leq i \leq p} \psi_i,$$

where, again, the formulas  $\psi_i$  are basic equality types.

Let  $F$  be the set of all pairs  $(n, t)$  such that  $\langle n \rangle \models \varphi[t]$ . We claim now that the formula

$$\psi = \bigvee_{(n,t) \in F} \eta_{n,t}$$

is equivalent to  $\varphi$  on pure sets. Indeed, if  $\langle n \rangle \models \varphi[t]$ , then  $(n, t) \in F$ , whence  $\langle n \rangle \models \psi[t]$ . On the other hand, if  $\langle n \rangle \models \eta_{n',t}[t]$  for some  $(n', t') \in F$ , then  $\langle n' \rangle \models \varphi[t']$  and  $t$  satisfies the same  $L_{\infty\omega}^k(\mathbf{Q}_0)$ -formulas in  $\langle n \rangle$  as  $t'$  in  $\langle n' \rangle$ . In particular,  $\langle n \rangle \models \varphi[t]$ .

Clearly the formula  $\psi$  is in flat normal form. It remains to show that  $\psi$  is (equivalent to) an  $\text{FO}(\mathbf{Q})$ -formula. To see this, observe that since the set  $\mathbf{Q}_0$  is finite, there are only finitely many different formulas of the form  $\eta_{n,t}$  up to logical equivalence. Hence the infinite set  $F$  can be replaced with a finite subset  $F_0$  that contains a representative for the  $L_{\infty\omega}^k(\mathbf{Q}_0)$ -equivalence type of each pair  $(n, t) \in F$ .  $\blacksquare$

**Corollary 3.4** *For every formula  $\varphi$  of  $\text{PFP}(\mathbf{Q})$  there exists a formula  $\psi$  of  $\text{FO}(\mathbf{Q})$  in flat normal form such that  $\varphi$  and  $\psi$  are equivalent on the class of pure sets. In particular,  $\text{PFP}(\mathbf{Q})$  collapses to  $\text{FO}(\mathbf{Q})$  on pure sets.*

**Proof.** A straightforward modification of the proof that  $\text{PFP} \subseteq L_{\infty\omega}^\omega$  (see [17]) shows that  $\text{PFP}(\mathbf{Q}) \subseteq L_{\infty\omega}^\omega(\mathbf{Q})$ . Since each formula of  $\text{PFP}(\mathbf{Q})$  contains only finitely many different quantifiers, we actually get the inclusion  $\text{PFP}(\mathbf{Q}) \subseteq L^*(\mathbf{Q})$ . Hence the claim follows from Theorem 3.3.  $\blacksquare$

Note that Corollary 3.4 implies the same flat normal form also for formulas of  $\text{IFP}(\mathbf{Q})$ , since clearly  $\text{IFP}(\mathbf{Q}) \subseteq \text{PFP}(\mathbf{Q})$ .

If  $\mathbf{Q}$  consists of NP quantifiers, then the flat normal form given in Theorem 3.3 can be further simplified.

---

<sup>2</sup>The result in [5] is formulated for complete structures over an arbitrary vocabulary. The claim for pure sets is obtained by considering the special case of the empty vocabulary.



**Corollary 3.5** *If  $\mathbf{Q}$  is a set of NP quantifiers, then, on the class of pure sets, every sentence  $FO(\mathbf{Q})$  is equivalent to a Boolean combination of NP properties.*

**Proof.** Let  $\varphi$  be a sentence of  $FO(\mathbf{Q})$ . By the proof of Theorem 3.3, on pure sets,  $\varphi$  is equivalent to a finite disjunction  $\bigvee_{n \in F} \eta_n$  of sentences of the form

$$\eta_n = \theta \wedge \bigwedge_{1 \leq j \leq r} \forall x_1 \dots \forall x_k (\varphi_j \leftrightarrow \gamma_j),$$

where  $\theta$  is a first order formula, each of the formulas  $\gamma_j$  is quantifier free, and each of the formulas  $\varphi_j$  is the result of a single application of some quantifier  $Q \in \mathbf{Q}$  to quantifier free formulas. Thus,  $\varphi$  is equivalent to a Boolean combination of first order formulas and formulas of the form

$$\forall x_1 \dots \forall x_k (\neg \varphi_j \vee \gamma_j) \wedge \forall x_1 \dots \forall x_k (\neg \gamma_j \vee \varphi_j).$$

Since each  $\varphi_j$  is NP-computable, and both NP and *co*-NP are closed under disjunctions and universal quantification, the claim follows.  $\blacksquare$

## 4 Negative Results about Generalized Quantifiers

The aim of this section is to provide evidence for the fact that over arbitrary (i.e., un-ordered) structures,  $LOGSPACE^{NP}$  cannot be captured by first order logic (or even fix-point logic) plus NP quantifiers. In particular, we show that if such a capturing result were possible, then a rather unexpected collapse of certain exponential complexity classes would occur.

For a language  $A$  over  $\{0, 1\}^*$ ,  $Pureset(A)$  denotes the set of structures arising from encoding each word of  $A$  as a pure set. More formally,

$$Pureset(A) = \{\langle n \rangle \mid 1^n \in tally(A)\}.$$

**Theorem 4.1** *If there exists a family  $\mathbf{Q}$  of NP quantifiers such that  $FO(\mathbf{Q})$  captures  $LOGSPACE^{NP}$ , then*

1.  $EBH = Linspace^{NP}$  and  $EBH$  collapses to some of its member classes; and
2.  $EXPBH = PSPACE^{NP}$  and  $EXPBH$  collapses to some of its member classes.

**Proof.** Let  $A$  be a language in  $Linspace^{NP}$ . Then, by proposition 2.3,  $tally(A)$  lies in  $LOGSPACE^{NP}$  and so, by hypothesis,  $Pureset(A)$  is expressible in  $FO(\mathbf{Q})$ . By Corollary 3.5, there exists a flat  $FO(\mathbf{Q})$  formula expressing  $Pureset(A)$ , and this formula is equivalent to a Boolean combination of NP properties. It follows that  $Pureset(A)$  and thus  $tally(A)$  is in  $BH_k$  for some constant  $k$ . Therefore, by Proposition 2.3,  $A$  is in  $EBH_k$ . It follows that  $Linspace^{NP} \subseteq EBH_k$ . Since, on the other hand,  $EBH_k \subseteq EBH \subseteq Linspace^{NP}$ , it follows that  $EBH_k = EBH = Linspace^{NP}$ . This proves 1. To see 2, recall that the closures under  $\leq_m^p$ -reductions of  $EBH_k$ ,  $EBH$ , and  $Linspace^{NP}$  are  $EXPBH_k$ ,  $EXPBH$ , and  $PSPACE^{NP}$ , respectively (Proposition 2.2). Thus it must hold that  $EXPBH_k = EXPBH = PSPACE^{NP}$ .  $\blacksquare$

The identity  $\text{EXPBH} = \text{PSPACE}^{\text{NP}}$  and the implied collapse of EXPBH would generate great surprise among complexity theorists. Most researchers dealing with these classes tend to believe that EXPBH is a proper hierarchy which is properly contained in  $\text{PSPACE}^{\text{NP}}$ . In fact, it is well known that  $\text{PSPACE}^{\text{NP}}$  coincides with the class  $\text{PTIME}^{\text{NEXPTIME}}$  of all problems solvable in polynomial time with polynomially many queries to a NEXPTIME oracle [13]. On the other hand, all problems in EXPBH can be solved in polynomial time with a *constant* number of queries to a NEXPTIME oracle. It would be rather surprising if polynomially many queries to such an oracle could be replaced by a constant number of queries.

There are interesting problems complete for  $\text{PSPACE}^{\text{NP}}$ . Here are two examples (for details see [9]):

- Let  $\Theta_2$  denote the first order closure of existential second order logic ( $\text{SO}\exists$ ). The problem of evaluating (varying)  $\Theta_2$  formulas over the fixed structure  $\langle\{0,1\}\rangle$  is complete for  $\text{PSPACE}^{\text{NP}}$ . (In other terms, the *expression complexity* of  $\Theta_2$  is  $\text{PSPACE}^{\text{NP}}$ .)
- Evaluating (varying) first order formulas with Henkin quantifiers over a fixed finite structure is  $\text{PSPACE}^{\text{NP}}$  complete.

No algorithms are known that solve those problems in polynomial time with a *constant* number of calls to a NEXPTIME oracle.

Note that by Corollary 3.4, we immediately get the following corollary to Theorem 4.1:

**Corollary 4.2** *If one of the following facts hold, then  $\text{EBH} = \text{LSPACE}^{\text{NP}}$ ,  $\text{EXPBH} = \text{PSPACE}^{\text{NP}}$ , and  $\text{EBH}$  and  $\text{EXPBH}$  both collapse to a fixed level  $k$ .*

1.  $\text{LOGSPACE}^{\text{NP}}$  is included in  $\text{IFP}(\mathbf{Q})$  or in  $\text{PFP}(\mathbf{Q})$  for some set  $\mathbf{Q}$  of NP quantifiers.
2.  $\text{PTIME}^{\text{NP}}$  is captured by  $\text{IFP}(\mathbf{Q})$  for some set  $\mathbf{Q}$  of NP quantifiers.
3.  $\text{PTIME}^{\text{NP}}$  is included in  $\text{PFP}(\mathbf{Q})$  for some set  $\mathbf{Q}$  of NP quantifiers.

It is thus unlikely that for any set  $\mathbf{Q}$  of NP quantifiers,  $\text{IFP}(\mathbf{Q})$  captures  $\text{PTIME}^{\text{NP}}$ . This is particularly interesting, because, as we will see below in this paper, the class  $\text{PTIME}^{\text{NP}}$  can be captured by an appropriate logic.

Let us conclude this section with an interesting remark concerning the collapse of the Boolean Hierarchy over NEXPTIME. By well-known result of Kadin [16] and Yap [26], the collapse of the Boolean Hierarchy BH entails the collapse of the entire Polynomial Hierarchy PH to its third level  $\Sigma_3^p$ . One may thus ask if analogous results hold also in the exponential cases, e.g., if the collapse of EXPBH would entail the collapse of the entire Exponential Hierarchy EXPH to some fixed level. Unfortunately, Kadin's proof does not carry over to the exponential case. There is evidence that proving the analogous result to Kadin's for the exponential Hierarchy would require much stronger techniques and a major complexity theoretic breakthrough. In fact, the following interesting result was recently shown by Mocas [20]:

**Proposition 4.3 ([20])** *If  $EXPTIME = NEXPTIME \Rightarrow NEXPTIME = EXPH$  then  $PH$  is properly contained in  $NEXPTIME$ .*

Note that there is currently no proof that the Polynomial Hierarchy is properly contained in  $NEXPTIME$ . Such a proof would be a major breakthrough. Many other results on  $EXPH$  are given in [12, 13, 20, 21, 9].

The premise in Mocas' Result mentions the *total* collapse of  $EXPH$ , i.e., the collapse of  $EXPH$  to its first level  $NEXPTIME$ . By applying basically the same proof argument as the one used by Mocas [20] in the proof of proposition 4.3, we can show that a similar result holds if collapses *to any level* are considered.

**Theorem 4.4** *If there is a constant  $k$  such that a collapse of  $EXPBH$  (to any level) implies a collapse of  $EXPH$  to  $\Sigma_k^{exp}$ , then  $PH$  is properly contained in  $\Sigma_k^{exp}$ .*

**Proof.** Assume a partial collapse of  $EXPBH$  implies a collapse of  $EXPH$  to  $\Sigma_k^{exp}$ . Assume  $\Sigma_k^{exp} = PH$ . Since  $\Sigma_k^{exp}$  has complete problems, then also  $PH$  must have complete problems, and thus  $PH$  collapses to some of its classes  $\Sigma_i^p$ . By the hypothesis, this entails a collapse of  $EXPH$  to  $\Sigma_k^{exp}$ . Let  $m = \max(i, k)$ . By a hierarchy theorem (see Mocas [20, 21]) it holds that  $\Sigma_m^p \neq \Sigma_m^{exp}$ , and thus  $PH \neq \Sigma_m^{exp} = \Sigma_k^{exp}$ . Contradiction. Therefore,  $\Sigma_k^{exp} \neq PH$  and thus  $PH$  is properly contained in  $\Sigma_k^{exp}$ . ■

Note that currently no level  $k$  is known such that  $PH$  is a proper subset of  $\Sigma_k^{exp}$  (though it can be seen that such a  $k$  must exist).

## 5 On Capturing PTIME Using Generalized Quantifiers.

By using similar methods as for Theorem 4.1, we show that it is very unlikely that  $PTIME$  can be expressed by extending fixed point logic with  $NLOGSPACE$  quantifiers.

**Theorem 5.1** *If  $IFP(\mathbf{Q})$  captures  $PTIME$  for a family  $\mathbf{Q}$  of  $NLOGSPACE$  quantifiers, then  $ETIME = NLINSPACE$  and  $EXPTIME = PSPACE$ .*

**Proof.** Assume the premise holds for a particular family  $\mathbf{Q}$  of  $NLOGSPACE$  quantifiers. Let  $A$  be a language in  $ETIME$ . Then  $tally(A)$  is in  $PTIME$  and so  $Pureset(A)$ , by Corollary 3.4 can be expressed by a flat  $FO(\mathbf{Q})$  formula. Since  $FO \subseteq LOGSPACE$ , such flat formulas can be evaluated in  $LOGSPACE^{NLOGSPACE} = NLOGSPACE$ . (The latter equality follows from the well-known result by Immerman and Szelepcsényi [15, 24] stating that  $NLOGSPACE$  is closed under complementation.) Thus  $tally(A)$  is in  $NLOGSPACE$  and therefore  $A$  is in  $NLINSPACE$ . Hence,  $ETIME = NLINSPACE$ . By taking the closures under  $\leq_m^p$ -reductions (see Proposition 2.2), we then also get  $EXPTIME = PSPACE$ . ■

A similar proof yields the following.

**Theorem 5.2** *If  $PTIME$  is included in  $PFP(\mathbf{Q})$  for some family  $\mathbf{Q}$  of  $NLOGSPACE$  quantifiers, then  $ETIME = NLINSPACE$  and  $EXPTIME = PSPACE$ .*

## 6 Capturing Relativized Complexity Classes

In this section we consider the question of which relativized complexity classes can be captured by some logic. To make this question precise, we can ask for which complexity classes are the isomorphism-closed properties in that class recursively indexable.

It suffices to focus on isomorphism-closed properties of graphs. To consider machines that compute graph properties, we choose the following representation of graphs as binary strings. A graph on the set of vertices  $\{0, \dots, n-1\}$  is represented by a binary string of length  $n^2$ . There is a 1 in the  $i$ th position of this binary string if, and only if, there is an edge  $(u, v)$ , where  $(u, v)$  is the  $i$ th pair in the lexicographical ordering of all pairs in  $\{0, \dots, n-1\}^2$ . Let  $\mathcal{G}$  denote the set of binary strings that encode graphs, and for any  $a, b \in \mathcal{G}$ , we write  $a \cong b$  to denote that the graphs represented by  $a$  and  $b$  are isomorphic.

**Definition 6.1** *A function  $C : \mathcal{G} \rightarrow \mathcal{G}$  is a canonical labelling function, if:*

- for any  $a \in \mathcal{G}$ ,  $a \cong C(a)$ ; and
- for any  $a, b \in \mathcal{G}$ , if  $a \cong b$ , then  $C(a) = C(b)$ .

In [10], Gurevich shows that, if there is a polynomial time computable canonical labelling function, then there is a logic that captures PTIME. This is easily generalised to the following observation:

**Proposition 6.2** *If  $C$  is a recursively presented complexity class, which contains a canonical labelling function, and is closed under compositions, then the class of isomorphism-closed properties in  $C$  is recursively indexable.*

Blass and Gurevich [3] observed that, for any polynomial time decidable equivalence relation on strings, there is a corresponding canonical element function in  $\text{PTIME}^{\text{NP}}$ . Their method, in fact, works for any equivalence relation that is decidable in NP, and hence, in particular, for the graph isomorphism problem. In the latter case, the canonical element function is just a canonical labelling function. For the sake of completeness, we sketch below a  $\text{PTIME}^{\text{NP}}$  algorithm that computes, for any  $a \in \mathcal{G}$ , the lexicographically first  $b \in \mathcal{G}$  such that  $a \cong b$ .

The oracle set is the set

$$I = \{(x, y) \mid \exists z \exists w \ x \cong z \text{ and } z = yw\}.$$

It is clear that  $I$  is in NP.

The algorithm, using the set  $I$  as oracle is now as follows:

1. **input**  $(x)$ ;
2.  $out := \varepsilon$  (the empty string);
3. **for**  $i := 1$  **to**  $n^2$  **do**:
  - 3a. write  $(x, out0)$  on the oracle tape, and query the oracle;
  - 3b. **if** oracle answers *yes* **then**  $out := out0$  **else**  $out := out1$ ;
4. **output**  $(out)$ .

Since the above algorithm is in  $\text{PTIME}^{NP}$ , the following is a direct consequence of Proposition 6.2.

**Proposition 6.3** *Any recursively presented complexity class containing  $\text{PTIME}^{NP}$  and closed under compositions is recursively indexable (and thus there is a logic capturing this class).*

It follows from this that there is a logic capturing, for example,  $\text{LOGSPACE}^{\Sigma_2^P}$ . Moreover, since this class is bounded, in the sense of [4], it follows that it is captured by a logic of the form  $\text{FO}(\overline{Q})$ . However, it remains an open question whether there is any logic capturing  $\text{LOGSPACE}^{NP}$ .

## Acknowledgment

We would like to thank Sarah Mocas for helpful discussions concerning the Boolean Hierarchy over  $\text{NEXPTIME}$  and for making her thesis and her papers available. We also thank Harry Buhrman, Lane Hemaspaandra, and Steve Homer for clarifications concerning exponential complexity classes.

## References

- [1] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43:62–124, 1991.
- [2] J. Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42:292–296, 1977.
- [3] A. Blass and Y. Gurevich. Equivalence relations, invariants, and normal forms. *SIAM Journal on Computing*, 13(4):682–689, 1984.
- [4] A. Dawar. Generalized quantifiers and logical reducibilities. *Journal of Logic and Computation*, 5(2):213–226, 1995.
- [5] A. Dawar and L. Hella. The expressive power of finitely many generalized quantifiers. *Information and Computation*, 123(2):172–184, 1995.
- [6] H.D. Ebbinghaus and J. Flum. *Finite Model Theory* Springer Verlag, 1995.
- [7] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol 7*, pages 43–73, 1974.
- [8] G. Gottlob. Relativized logspace and generalized quantifiers over ordered finite structures. *Journal of Symbolic Logic*, to appear. Short version in: *Proc. Tenth Annual Symposium on Logic in Computer Science LICS'95 (San Diego, CA, June 1995)*. IEEE Computer Society Press, pp. 65–78, 1995.
- [9] G. Gottlob, N. Leone, and H. Veith. Second order logic and the weak exponential hierarchies. In *Proc. Intl. Symp. on Mathematical Foundations of Computer Science, MFCS'95, Prague, CR*, LNCS nr. 969, pages 66–81. Springer, August/September 1995.

- [10] Y. Gurevich. Toward logic tailored for computational complexity. In M. Richter et al., editors, *Computation and Proof Theory*, pages 175–216. Springer Lecture Notes in Mathematics, 1984.
- [11] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.
- [12] L. A. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39:299–322, 1989.
- [13] E. Hemaspaandra. Census techniques collapse sparse classes. *Information Processing Letters*, 51:79–84, 1994.
- [14] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.
- [15] N. Immerman. Nondeterministic space is closed under complementation. *SIAM J. Comp.*, 17, pp. 935–939, 1988.
- [16] J. Kadin. The polynomial hierarchy collapses if the Boolean Hierarchy collapses. *SIAM J. Comput.*, 17:1263–1282, 1988.
- [17] Ph. G. Kolaitis and M. Y. Vardi. Infinitary logics and 0-1 laws. *Information and Computation*, 98(2):258–294, 1992.
- [18] R. Ladner, N. Lynch. Relativization of questions about log-space reducibility. *Math. Systems Theory*, vol. 10, pp. 19–32, 1976.
- [19] J. A. Makowsky and Y. B. Pnueli. Oracles and quantifiers. *Computer Science Logic, Refereed papers from CSL'93*, Springer LNCS vol.832, 1994, pp. 198-222.
- [20] S. Mocas. *Separating exponential time classes from polynomial time classes*. Ph.D. Thesis, Graduate School, College of Computer Science, Northeastern University, 1993.
- [21] S. Mocas. Separating classes in the exponential-time hierarchy from classes in PH. *Theoretical Computer Science*, 185:1-2, pp.221–231, 1996.
- [22] Iain A. Stewart. Logical characterization of bounded query classes I: Logspace Oracle Machines. *Fundamenta Informaticae*, vol. 18, pp. 65–92, 1993.
- [23] Iain A. Stewart. Logical characterization of bounded query classes II: Polynomial-time Oracle Machines. *Fundamenta Informaticae*, vol. 18, pp. 93–105, 1993.
- [24] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26, pp. 279–284, 1988.
- [25] M. Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on the Theory of Computing*, pages 137–146, 1982.
- [26] C. Yap. Some Consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3) pages 287–300, 1983.
- [27] Klaus Wagner. Bounded query classes. *SIAM J. Comp.*, 19(5), pp. 833–846, 1990.