More on Oracles and Quantifiers

Yachin B. Pnueli*
Janos A. Makowski**

B 94–17
September 1994

Abstract

We continue the investigation of [MP94] into the relationship between classes of oracle using Turing machines and logics enhanced by Lindström quantifiers. Let $\mathcal{L}$ be a logic (FOL or SOL) and let $\mathcal{L}[K]$ be the enhancement of that logic with a Lindström quantifier for the set of structures K. We show that if for some sets of structures A, B, C we have $\mathcal{L}[A, C] \subset \mathcal{L}[B, C]$ then also $\mathcal{L}[A] \subset \mathcal{L}[B]$. This has the complexity theoretic implication that, using the appropriate oracle computation model, finding an oracle K such that $\mathbf{L}^K \subset \mathbf{P}^K$ constitutes a proof that $\mathbf{L} \subset \mathbf{P}$. Considering the case where $\mathcal{L}$ is Second Order Logic or fragments of it we use the results of Meyer and Stockmeyer about the polynomial hierarchy to show that if $\Sigma_i(\Pi_i)$ is a fragment of SOL capturing level Σ_i^P (Π_i^P) in the polynomial hierarchy, then the enhanced fragment $\Sigma_i[K](\Pi_i[K])$ (for arbitrary K) captures $(\Sigma_n^P)^K$ $((\Pi^p{}_n)^K)$ - that level relativized to an oracle for the set K. As a corollary the logic $SOL[K]$ captures the polynomial hierarchy relativized to oracle K and has a prenex normal form where all second order quantifiers appear outer most.

*Institut für Informatik, Freie Universität Berlin, Takustr. 9, D-14195, Germany, E-mail: yachin@inf.fu-berlin.de. Since Fall 1993 Minerva Fellow.
**Department of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel, E-mail: janos@cs.technion.ac.il.

1 Introduction

The purpose of this paper is to continue the investigation of [MP94] into the close relationships between the notion of enhancing a logic with a quantifier for a given set of structures K and the enhancing of a set of Turing machines which form a complexity class with an oracle to the same set K.

The study of the relationship between various logics and various complexity classes, known as *descriptive complexity* is by now well established [AV88, Fag74, Imm87, Imm89, Sto87, Ste93a, Ste93b] and more. Typically either a fragment of Second Order Logic (SOL) or an enhancement of First Order Logic (FOL) is studied and found to *capture* some well known complexity class. (A logic *captures* a complexity class if every set of structures or language recognizable by a Turing machine in the class is definable by a formula in the logic, and every set of structures definable by a formula in the logic is recognizable by a machine in the class). From the complexity theoretic point of view, the hope is that by reinterpreting the *big* open problems of complexity theory in logical terms, we gain new insight as to what are the correct terms to use and what are the right questions we should ask (or hope to answer).

Following the line of investigation which extends FOL with some Lindström quantifier, we have the well known results of Immerman [Imm87]:

- $FOL[DTC] = \mathbf{L}$ - First Order Logic enhanced by a quantifier for deterministic transitive closure - captures the complexity class of deterministic log space bounded machines.
- $FOL[TC] = \mathbf{NL}$ - FOL enhanced by a quantifier for transitive closure captures the complexity class of nondeterministic log space bounded machines.
- $FOL[ATC] = \mathbf{P}$ - FOL enhanced by a quantifier for alternating transitive closure captures the complexity class of deterministic polynomial time bounded machines.

These results lead among other things to the purely complexity theoretic results of [Imm88] that $\mathbf{NL} = Co - \mathbf{NL}$.

Recent results of the above type were published by Stewart [Ste93a, Ste93b]:

- $FOL[HAM] = \mathbf{L}^{\mathbf{NP}}$ - FOL enhanced with a quantifier for the Hamiltonicity of graphs, captures the class of log space machines using oracles in $\mathbf{NP}$.

and by the authors [MP93]:

- $FOL[HEX] = \mathbf{PSpace}$ - Where HEX is a quantifier saying that on a given graph, player 1 has a winning strategy in the game of HEX as defined in [GJ79].

In [MP94], by choosing an appropriate oracle computation model, these last two results were generalized. For arbitrary K it was shown that

- $FOL[DTC, K]$ ($FOL[TC, K]$, $FOL[ATC, K]$) capture $\mathbf{L}^K$ ($\mathbf{NL}^K$, $\mathbf{P}^K$) - the class of machines in $\mathbf{L}$ ($\mathbf{NL}$,$\mathbf{P}$) using an oracle for K.
- If, furthermore, K is such that DTC (TC, ATC) is expressible in $FOL[K]$ then $FOL[K]$ captures $\mathbf{L}^K$ ($\mathbf{NL}^K$, $\mathbf{P}^K$).

As ATC is expressible in $FOL[HEX]$ [MP93] and HEX is $\mathbf{PSpace}$–complete for $\mathbf{P}$ reductions [GJ79], it follows that this logic captures $\mathbf{P}^{\mathbf{PSpace}} = \mathbf{PSpace}$. Using Stewart's results and the fact that $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{L}^{\mathbf{NP}}$ we obtained the complexity theoretic result that $\mathbf{L}^{\mathbf{NP}} = \mathbf{NL}^{\mathbf{NP}} = \mathbf{P}^{\mathbf{NP}}$.

An important aspect of this work was that for these results to hold, one must carefully define the oracle computation model for the space bounded classes, a fact which illuminates the sensitivity of relativized–complexity results to the precise type of oracle computation model used. (See [MP94] and references therein for details of the subject and the exact model used. Aspects of this issue relevant to us are presented ahead at the end of section 3).

The equivalence of the classes $\mathbf{L}^{\mathbf{NP}} = \mathbf{NL}^{\mathbf{NP}} = \mathbf{P}^{\mathbf{NP}}$ naturally raises the question of whether such results are a coincidence dependent on the choice of the oracle and what conclusions (if any) one can draw from such results about the unrelativized classes. Specifically, does $\mathbf{D}^K \neq \mathbf{C}^K$ imply $\mathbf{D} \neq \mathbf{C}$?

The first result of this paper is a "logical" characterization of the cases for which this implication is true.

Theorem 1. *For A, B, C sets of structures. If $FOL[A, C] \neq FOL[B, C]$ then $FOL[A] \neq FOL[B]$.*

From which we get

Corollary 2. *If there is an oracle K such that $\mathbf{L}^K \neq \mathbf{P}^K$ ($\mathbf{L}^K \neq \mathbf{NL}^K$, $\mathbf{NL}^K \neq \mathbf{P}^K$) then $\mathbf{L} \neq \mathbf{P}$ ($\mathbf{L} \neq \mathbf{NL}$, $\mathbf{NL} \neq \mathbf{P}$).*

Note: The fact that from $\mathbf{L}^K \neq \mathbf{NL}^K$ we have $\mathbf{L} \neq \mathbf{NL}$ was already observed in [Si77, RS81, Wil86].

To the best of our knowledge, as yet, no such separating oracles were found.

On the other hand, Baker, Gill and Solovey ([BGS75]) demonstrated K such that $\mathbf{P}^K \neq \mathbf{NP}^K$. Further work along this line (see Ko [Ko89] and references therein) show oracles which separate the polynomial hierarchy at any desired level and also from $\mathbf{P}$ and $\mathbf{PSpace}$ (where $(\Sigma^P_{n+1})^K$ is defined as $\mathbf{NP}^{(\Sigma^P_n)^K}$). Contrasting such results with the above gives a strong motivation to find logics or fragments of logics which capture relativized levels of the polynomial hierarchy.

This is the second result of this paper. We define an extension of our framework in [MP94] - $SOL[K]$ - the enhancement of second order logic by a quantifier for the set K. This definition can be built up such that for any fragment Σ_i or Π_i of SOL the enhanced fragment $\Sigma_i[K]$ or $\Pi_i[K]$ is well defined (where Σ_i/Π_i contain only formulas where all SO quantifications are outer most and there are no more than i alternations of such quantifiers, starting with $\exists/\forall$). Using well known results by Fagin, Meyer and Stockmeyer [Fag74, MS72, Sto87, GJ79] which show that each fragment $\Sigma_i(\Pi_i)$ of SOL captures the appropriate $\Sigma^P_i(\Pi^P_i)$ level of the polynomial hierarchy and that the entire hierarchy ($\mathbf{PH}$) is captured by full SOL we show that

Theorem 3. *For each fragment i: Σ_i or Π_i of SOL the enhanced fragment $\Sigma_i[K]$ captures $\Sigma^P_i[K]$ and $\Pi_i[K]$ captures $\Pi^P_i[K]$.*

From which we immediately get

Corollary 4. *$\cup_\infty \Sigma_i[K]$ captures $\mathbf{PH}^K$*

and with a bit more work.

Corollary 5. *$SOL[K]$ captures $\mathbf{PH}^K$ and has a normal form where all SO quantifiers are outer most.*

One might be tempted to conclude that these results, applied to the separating oracles of Baker Gill and Solovey or Ko, would constitute a proof that $\mathbf{P} \neq \mathbf{NP}$ however this conclusion is WRONG. While we may easily prove SOL versions of theorem 1 where FOL is replaced by SOL or any Σ_i/Π_i, we do not have a "hybrid" theorem stating that $FOL[ATC, K] \subset SOL[K]$ implies $FOL[ATC] \subset SOL$. In fact we argue that in general, the opposite is actually true, i.e. that it is possible to have two logics $\mathcal{L}_1 \subseteq \mathcal{L}_2$ such that $\mathcal{L}_1[K] \nsubseteq \mathcal{L}_2[K]$ for some well chosen K.

To be more precise, one can define a notion of *natural* and *unnatural* inclusions between logics, where an inclusion is *unnatural* if the less expressive logic has more expressive syntactical constructs (as is the case when a logic formed by enhancing FOL with some generalized quantifiers is more expressive than a logic based on SOL or its fragments). Our argument is that in cases of unnatural inclusion one can utilize the difference in the expressive power of the syntactical constructs to construct a K such that $\mathcal{L}_1[K] \nsubseteq \mathcal{L}_2[K]$. This is not too surprising since in complexity theoretic terms, this is just a restatement of a theorem due to Buss (theorem 2.5 in [Bus88]) that if there is a sufficient difference between complexity classes in terms of the number and type of queries they can ask of an oracle, then there is an oracle which separates them.

The remainder of the paper formalizes the notions above as follows: Section 1 gives the basic definitions of a logic enhanced by a Lindström quantifier. Section 2 states and proves our theorem 1, its variations and its corollaries. Section 3 states and proves our theorem 3 and its corollaries. Finally in section 4 we use the notion of natural and unnatural inclusions between logics to characterize conclusions which can and cannot be drawn from our results.

We assume the reader is familiar with the basics of complexity theory as presented in [GJ79] or the excellent surveys [Sto87, Joh90], and with the basics of abstract model theory as presented in [EFT80, CK90] or in [Ebb85] of [BF85]. Throughout this work, we assume that all structures under consideration are finite and ordered (although in some cases the order requirement is superfluous) and that all sets of structures under consideration are closed to isomorphism.

Acknowledgments: We would like to thank M. Mostowski and M. Krynicki for enabling us to present a preliminary version of this paper in the Seminar on Second Order Logic (Warsaw 28.8-3.9.94) and to thank the audience at that seminar, and specifically J. Flum, W. Degen and A. Dragalin, for constructive questions and remarks.

2 Basic Definitions

We define $\mathcal{L}[K]$ - the enhancement of $\mathcal{L}$ - a logic (or a syntactic fragment of a logic) with a first order Lindström quantifier for K - a set of structures. As a basis one can think of $\mathcal{L}$ as either First Order Logic (FOL) or Second Order Logic (SOL), however the definition can be extended inductively to more than one quantifier such that $\mathcal{L}$ can itself be a logic formed by enhancing FOL or SOL with some set of Lindström quantifiers.
Note: Our definition of $\mathcal{L}[K]$ is similar to the way Ebbinghaus defines Lindström quantifiers in [Ebb85] of [BF85] (where the enhanced logic is denoted as $\mathcal{L}(Q_K)$). The modifications being that as in [MP94] we include vectorization already in the definition and that here we allow free second order variables in the $Q_K\Phi$ formula.

Definition 6. (Feasibility) Let $\tau = \{R_1, \ldots, R_m\}$ be some vocabulary. For R_i a relation symbol, let $\rho(R_i)$ denote its arity. Let $\Phi = \langle \phi_0, \phi_1, \ldots, \phi_m \rangle$ be formulas of $\mathcal{L}$ over a (possibly different) vocabulary σ. We say that Φ is *k–feasible for τ over σ* if the following hold:

(i) ϕ_0 has k distinguished distinct (first order) free variables.
(ii) Each ϕ_i $(m \geq i > 0)$ has $k\rho(R_i)$ distinguished distinct (first order) free variables.
(iii) It is possible for Φ to have other non–distinguished free (first and second order) variables. These do not have to be distinct among the component formulas of Φ.

Φ is *feasible for τ over σ* if it is k–feasible for τ over σ for some integer k.

Observe that if Φ is feasible for τ over σ, it can be regarded as a *logical reduction* transforming a σ structure into a τ structure. We formalize this notion as follows:

Definition 7. (The structure $\mathcal{A}_\Phi$) Let $\mathcal{A}$ (with universe A) be a σ–structure and Φ be k–feasible for τ over σ. Given a substitution $\mathcal{Z}$ for all none–distinguished free variables of Φ (FO variables to elements of A and SO variables to appropriate subsets of A), the structure $\mathcal{A}_\Phi$ is defined as follows:

(i) The universe of $\mathcal{A}_\Phi$ is the set $A_\Phi = \{\bar{a} \in A^k : \mathcal{A}, \mathcal{Z} \models \phi_0(\bar{a})\}$;
(ii) The interpretation of R_i in $\mathcal{A}_\Phi$ is the set

$$\mathcal{A}_\Phi(R_i) = \{\bar{a} \in {A_\Phi}^{\rho(R_i)} : \mathcal{A}, \mathcal{Z} \models \phi_i(\bar{a})\};$$

Note that $\mathcal{A}_\Phi$ is a τ–structure of cardinality at most $|A|^k$.

Definition 8. (The Q_K Formation Rule) Let K be a set of τ–structures and Φ be a set of σ–formulas of the logic $\mathcal{L}$ feasible for τ. The Q_K formation rule, states that

$$Q_K \Phi$$

is a formula, where all the distinguished free variables of Φ are bound. Given a substitution $\mathcal{Z}$ to the remaining free variables of Φ, $\mathcal{A}, \mathcal{Z} \models Q_K \Phi$ iff $\mathcal{A}_\Phi \in K$.

Definition 9. (The Logic $\mathcal{L}[K]$) The enhancement of a logic $\mathcal{L}$ by a quantifier for K ($\mathcal{L}[K]$) is obtained by adding the Q_K formation rule to the set of formation rules of $\mathcal{L}$ and by adding the interpretation of this rule to the set of interpretation rules of $\mathcal{L}$.

The above definition allows us to speak of $FOL[K]$, $SOL[K]$ and inductively about $FOL[K_1 \ldots K_n]$ ($SOL[K_1 \ldots K_n]$). For some of our purposes, it is convenient also to define $\mathcal{L}[K]$ where $\mathcal{L}$ is a syntactic fragment of SOL. We formalize this notion as follows:

Definition 10. (Sub logics of $SOL[K]$)

(i) $INSOL[K]$ is the restriction of $SOL[K]$ to formulas where all second order quantifiers appear outside any Q_K quantifier. (The closure of $FOL[K]$ to the formation rules of SOL).
(ii) $NFSOL[K]$ (Normal Form SOL) is the restriction of $SOL[K]$ to formulas where all second order quantifiers appear outside any other elements of the formula. (The closure of $FOL[K]$ to quantification over relation variables).
(iii) $\Sigma_i[K]$ ($\Pi_i[K]$) - is the restriction of $NFSOL[K]$ to formulas with at most $i-1$ alternations of SO quantifiers starting with $\exists$ ($\forall$). Note that all formulas with less than $i-1$ alternations are formulas of both $\Sigma_i[K]$ and $\Pi_i[K]$).

We observe that from the definitions it is not clear that for arbitrary K $NFSOL[K] = SOL[K]$ and even for K's where it is so the "transformation rules" which produce an $NFSOL[K]$ formula given an arbitrary $SOL[K]$ formula, might be K–specific. However we can show the following:

Lemma 11. (Syntactic Lemma) *Every formula in $INSOL[K]$ has an equivalent formula in $NFSOL[K]$.*

Proof: Follows by standard "normal form" techniques for SOL. Clearly any formula of $INSOL[K]$ can be formed by applying some formation rules of SOL to a set of $NFSOL[K]$ formulas. We show that after each application of a formation rule the resulting $INSOL[K]$ formula has an equivalent $NFSOL[K]$ formula. The lemma follows by induction.

The formation rules of SOL are: $\neg$, $\wedge/\vee$, First order quantification and second order quantification.

- By definition applying SO quantification leaves us with a formula in $NFSOL[K]$.
- For $\neg\phi$ the equivalent formula is obtained by "pulling" the negation inside while changing each $\exists(\forall)$ quantifier on the way to $\forall(\exists)$.
- For $\phi \wedge \psi$ ($\phi \vee \psi$) we rename relation variables to ensure that no quantified relation variable appears in both ϕ and ψ, and then "pull" out all quantifications of second order variables.
- For first order quantification observe that an $\exists x \exists R \phi$ ($\forall x \forall R \phi$) is semantically equivalent to $\exists R \exists x \phi$ ($\forall R \forall x \phi$). For $\exists x \forall R \phi$ where R is of arity n we observe that the formula $\exists \bar{R} \forall x \psi$ is semantically equivalent if $\bar{R}$ is a relation of arity $n+1$ and ψ is obtained from ϕ by replacing any appearance of $R(\mathbf{y})$ with $\bar{R}(x, \mathbf{y})$. The argument for $\forall x \exists R \phi$ is similar. Hence first order quantifiers can be "pulled" inside all second order quantifiers.

Observe that from our proof each $INSOL[K]$ formula has an equivalent $NFSOL[K]$ one with the same number of alternations (though possibly with higher arity relation variables and a different leading quantifier). □

3 $FOL[A,C] \neq FOL[B,C]$ implies $FOL[A] \neq FOL[B]$

We are now ready to state and prove our first theorem:

Theorem 12. *Let A, B, C, K be sets of structures over possibly different vocabularies, such that K is definable in $FOL[B,C]$ but not in $FOL[A,C]$. Then B is not definable in $FOL[A]$.*

Proof: Let ϕ_K denote the $FOL[B,C]$ formula which defines the set K. Over all possible sets of structures definable in $FOL[B,C]$ but not in $FOL[A,C]$, let K be the set of structures such that ϕ_K has a minimum number of formation rules, (any $FOL[B,C]$ formula requiring less formation rules has a semantically equivalent formula in $FOL[A,C]$) where w.l.o.g. we assume $FOL[B,C]$ to have only the five formation rules $\neg$, $\vee$, $\exists x$, Q_B and Q_C.

Clearly ϕ_K cannot be a term and cannot be of the forms $\neg\psi$ and $\psi_1 \vee \psi_2$.

Claim 13. *ϕ_K cannot be of the form $\exists x \psi(x)$*

Proof: Let σ be the vocabulary of ϕ_K. Consider the formula $\psi(c)$ over the vocabulary σ' which extends σ with one constant symbol c. If the set of σ'–structures defined by $\psi(c)$ cannot be defined in $FOL[A,C]$ we contradict the minimality of ϕ_K and if it can be defined by some formula $\psi'(c) \in FOL[A,C]$ then $\exists x \psi'(x)$ defines K where $\psi'(x)$ is formed from $\psi(c)$ by substituting the variable x for the constant symbol c, thus contradicting the undefinability of K in $FOL[A,C]$. □

Claim 14. *If $\phi_K = Q\langle\phi_0, \phi_1, \ldots, \phi_m\rangle$ (where Q is either Q_B or Q_C) then for each of the ϕ_i subformulas there is an equivalent formula in $FOL[A,C]$ such that for every possible substitution of the free variables the two formulas describe the same set of structures.*

Proof: Assume the contrary, then by replacing the free variables of ϕ_i with constant symbols over an extended vocabulary, we define in $FOL[B,C]$ a set of structures not definable in $FOL[A,C]$. But ϕ_i uses less formation rules than ϕ_K - a contradiction. □

Corollary 15. *ϕ_K cannot be of the form $Q_C\Phi$.*

Claim 16. *ϕ_K is of the form $Q_B\langle\phi_0, \phi_1, \ldots, \phi_m\rangle$ where all the ϕ_i's are terms.*

Proof: That ϕ_K is of the form $\phi_K = Q_B\Phi$ follows by elimination of all other formation rules. Assume that some ϕ_i is not a term and consider the formula

$$Q_B\langle\phi_0, \phi_1, \ldots, \phi_{i-1}, R_i, \phi_{i+1}, \ldots, \phi_m\rangle \wedge \forall \mathbf{x_i} R_i(\mathbf{x_i}) \longleftrightarrow \phi_i(\mathbf{x_i})$$

over a vocabulary σ' extending σ with the relation symbol R_i of arity equal the number of distinguished free variables ϕ_i. This new formula defines a set of σ'–structures K'. If K' is definable by some $\psi \in FOL[A,C]$ then so is K: substitute the $FOL[A,C]$ equivalent of ϕ_i in place of R_i in the formula ψ. (By claim 14 such an equivalent to ϕ_i exists). Hence K' is undefinable in $FOL[A,C]$. But as

$$\forall \mathbf{x_i} R_i(\mathbf{x_i}) \longleftrightarrow \phi_i(\mathbf{x_i})$$

can be defined in $FOL[A,C]$, we conclude that the set of structures defined by

$$Q_B\langle\phi_0, \phi_1, \ldots, \phi_{i-1}, R_i, \phi_{i+1}, \ldots, \phi_m\rangle$$

cannot be defined in $FOL[A,C]$, thus contradicting the minimality of ϕ_K. □

Therefore, the set of structures defined by $Q_B\langle R_0, R_1, \ldots, R_m\rangle$ cannot be defined in $FOL[A,C]$ and hence cannot be defined in $FOL[A]$. □

Remark. Similar arguments prove the following variations of the above theory:

(i) When FOL is replaced by SOL.

(ii) When FOL is replaced by Σ_i or Π_i for arbitrary i.
(iii) When FOL is replaced by $\mathcal{L}$ where $\mathcal{L}$ is either FOL, SOL, Σ_i or Π_i enhanced by some quantifiers: If for some sets $K_1 \dots K_m$ and $L_1 \dots L_n$ we have $\mathcal{L}[K_1 \dots K_m] \neq \mathcal{L}[L_1 \dots L_n]$ then there must be some L_i not expressible in $\mathcal{L}[K_1 \dots K_m]$.
(iv) When $FOL[A, B]$ and $FOL[A, C]$ are restricted to formulas where the Q_C quantifier cannot be nested within itself.

Corollary 17. *Under the appropriate oracle computation model for log space bounded machines we have that: If there is an oracle K such that $\mathbf{L}^K \neq \mathbf{P}^K$ ($\mathbf{NL}^K \neq \mathbf{P}^K$, $\mathbf{L}^K \neq \mathbf{NL}^K$) then $\mathbf{L} \neq \mathbf{P}$ ($\mathbf{NL} \neq \mathbf{P}$, $\mathbf{L} \neq \mathbf{NL}$).*

Proof: By direct application of the above theorem we get that $FOL[DTC, K] \neq FOL[ATC, K]$ implies $FOL[DTC] \neq FOL[ATC]$. For any K under the appropriate oracle model we get from [MP94] that $\mathbf{L}^K \neq \mathbf{P}^K$ iff $FOL[DTC, K] \neq FOL[ATC, K]$ and by [Imm87] $\mathbf{L} \neq \mathbf{P}$ iff $FOL[DTC] \neq FOL[ATC]$. (The $\mathbf{NL}$ cases are similar). □

Because of the importance of this corollary and the delicate role played by the oracle computation model (in space bounded cases), we repeat here the details of the model, denoted in [MP94] as $u(\infty)$, for which these results hold:

(i) The oracle tape is write only and exempt from the space bound.
(ii) The machine has two distinguished states *start_query* and *answer_query*. Using these states the oracle tape can be regarded as a stack of tapes. Each time *start_query* is entered a new tape is pushed and each time *answer_query* is entered the top tape is popped and execution continues in either an *oracle_accepted* or an *oracle_rejected* state. Note that this has the effect of erasing each query as it is asked.
(iii) For non deterministic machines - nondeterministic moves are allowed only when the oracle tape stack is empty.
(iv) If $q_1, q_2 \dots q_k$ are the (partial) queries written on the stack at a given moment, then the space used by the stack is $\sum_{i=1}^{k} max(log|q_i|, 1)$.
(v) Each machine has a constant n such that the depth of the stack cannot be more than n.

The last requirement characterizes our $u(\infty)$ model. It contrasts with the unbounded model of Buss [Bus88] and the model of Wilson [Wil86] where the depth of the stack is bounded only by condition (iv) and with the models of Ladner and Lynch [LL76] and Ruzzo, Simon, and Tompa [RST84] where the stack depth is bounded to 1 (no stack).

We observe that applying variation (iv) of theorem 1 to results in [MP94] we get that corollary 17 holds also in the Ruzzo, Simon, Tompa model for the classes $\mathbf{L}$, $\mathbf{NL}$, $\mathbf{P}$ and in the Ladner, Lynch model for $\mathbf{L}$ and $\mathbf{P}$. The case of $\mathbf{L}^K \neq \mathbf{NL}^K$ implying $\mathbf{L} \neq \mathbf{NL}$ was already observed in [Wil86] for the Ruzzo, Simon, Tompa model, and in [Si77, RS81] for an oracle model where the oracle tape is subject to the space bound.

4 Capturing the Relativized Polynomial Hierarchy

Theorem 18. *(i) For each fragment $\Sigma_i(\Pi_i)$ of SOL the enhanced fragment $\Sigma_i[K](\Pi_i[K])$ captures $\Sigma_i^P[K](\Pi_i^P[K])$.*
(ii) $NFSOL[K] \equiv \cup_\infty \Sigma_i[K]$ captures $\mathbf{PH}^K$.
(iii) $SOL[K]$ also captures $\mathbf{PH}^K$ and hence is equivalent to $NFSOL[K]$ (every formula of $SOL[K]$ has a normal form where all SO quantifiers are outermost.

We prove this theorem as follows: First we prove model the checking part of (i) - each set of structures definable by a formula of the logic is recognizable by a machine of he complexity class. Then we prove the expressibility part of (i) - every set recognizable by a machine in the complexity class is definable by a formula of the logic. Thus we get part (i). Part (ii) follows as a direct corollary. To get part (iii) we show that every set of structures definable by a formula of $SOL[K]$ is recognizable by a machine in $\mathbf{PH}^K$.

Before we present the proofs we repeat the definitions of $\mathbf{PH}^K$ as they appear in [Ko89] and elsewhere:

$$(\Sigma_0^P)^K = (\Pi_0^P)^K = \mathbf{P}^K$$

$$(\Sigma_{n+1}^P)^K = \mathbf{NP}^{(\Sigma_n^P)^K} \qquad (\Pi_{n+1}^P)^K = Co - ((\Sigma_{n+1}^P)^K)$$

$$\mathbf{PH}^K = \cup_\infty (\Sigma_i^P)^K$$

4.1 Model Checking

Lemma 19. *Every formula in $FOL[K]$ has a model checker in $\mathbf{P}^K(\mathbf{L}^K)$.*

Proof: Every formula of $FOL[K]$ is also a formula of $FOL[ATC, K]$ (and $FOL[DTC, K]$). The proof follows by the model checking theorems of [MP94]. □

Theorem 20. *Every formula in $\Sigma_i[K]$ ($\Pi_i[K]$) has a model checker in $(\Sigma_i^P)^K$ $\left((\Pi_i^P)^K\right)$.*

Proof: By induction on i.

Basis ($i = 1$): For $\psi = \exists R_1 \ldots R_m \phi$ a formula in $\Sigma_1[K]$, let the model checker operate as follows: First it nondeterministically generates the relations $R_1 \ldots R_m$ and then using them as substitutions simulates the model checker of ϕ (on a vocabulary extended by $R_1 \ldots R_m$). As all relations are polynomial in the size of the input structure and the model checking itself is in $\mathbf{P}$ the whole operation is in $(\Sigma_1^P)^K = \mathbf{NP}^K$.

Induction hypothesis: For every $i \leq n$ model checking of $\Sigma_i[K]$ can be done in $(\Sigma_i^P)^K$.

Inductive step I: The set of structures satisfying a formula $\phi \in \Pi_n[K]$ is in $(\Pi_n^P)^K$.

Proof: This set is the complement of the set satisfying $\neg\phi$. But $\neg\phi$ is equivalent to a formula of $\Sigma_n[K]$ and hence has a model checker in $(\Sigma_n^P)^K$.

Inductive step II: Any formula of $\Sigma_{n+1}[K]$ has a model checker in $(\Sigma_{n+1}^P)^K$.

Proof: Any $\phi \in \Sigma_{n+1}[K]$ can be written as $\exists R_1 \ldots R_m \psi$ where ψ is in $\Pi_n[K]$. By inductive step I, given a substitution for $R_1 \ldots R_m$, the set of structures satisfying ψ is in $(\Pi_n^P)^K$. Let the model checker of ϕ guess these relations and copy them together with the original input to an oracle tape for the appropriate $\Pi_n[K]$ set. As a nondeterministic step was done and an oracle for $(\Pi_n^P)^K$ was used, this model checker is in $(\Sigma_{n+1}^P)^K$. □

Corollary 21. *Every formula in $NFSOL[K]$ has a model checker in $\mathbf{PH}^K$.*

4.2 Expressibility

The proofs below are mere variations on similar proofs in [MP94] or on Fagin's and Stockmeyer's proofs and hence are given only in outline.

Lemma 22. *$\Sigma_1[K]$ is as expressive as $\mathbf{NP}^K$.*

Proof: Let M be a machine in $\mathbf{NP}^K$, then there is a number d such that for any input of size n, M makes no more than n^d moves. Without loss of generality we can assume that the first $n^d/2$ moves of M nondeterministically generate a string of bits on a special tape which we shall call the *nondet* tape, while the remaining $n^d/2$ are deterministic, (using bits of the nondet tape to resolve any possible nondeterminism).

Using an arbitrary order relation we can encode steps of the computation of M as d–tuples over the universe of the input structure.

As n^d is also a bound on the amount of information on a tape of M, we can use a d–tuple of elements over the input structure to represent a tape position. Therefore a d–ary relation can describe the state of a tape of M if the alphabet is binary. If the alphabet is larger or if we wish to encode the head position too, a $d+1$–ary relation is necessary. Altogether a $2d+1$–ary relation can represent the evolution of a tape during the computation of M. Using 5 such relations R_{state}, R_{input}, R_{nondet}, R_{work} and R_{oracle} we can represent the complete evolution of a computation of M. (Actually R_{input}, R_{state} and R_{notdet} can be of arity $d+1$).

Claim: The answer of the oracle for K at a given computation step, is definable by a $FOL[K]$ predicate over R_{oracle} where the Q_K quantifier is outer most.

Claim: The predicate $VALID_M$ over the 5 relations saying that these relations indeed represent the evolution of the computation of M is $FOL[K]$ definable. Note that this includes saying that all 5 relations are valid encodings of their appropriate objects: For each step j each tape position has a unique and valid value, each tape has only one head position, and only one state is active at a time. It also includes saying that for step zero R_{input} encodes the input and the work and oracle tapes are empty, all heads are properly positioned and M is in its initial state. Finally, we have to say that the computation states indeed evolve according to the transition table of M. Here we use the fact that M acts deterministically as we cannot represent nondeterminism, using fixed arity relations, while insisting on each tape cell having a unique value. Also here we use the Q_K quantifier (for those transitions requiring oracle consultation).

Claim: The predicate $ACCEPTING(\mathbf{x})$ saying that a $d+1$–tuple of first order variable $\mathbf{x}$ represents a final and accepting state of the computation is first order definable.

The set of all structures accepted by a machine $M \in \mathbf{NP}^K$ can be represented then as an $\Sigma_1[K]$ formula:

$$\exists R_{input}, R_{state}, R_{nondet}, R_{work}, R_{oracle}, \exists \mathbf{x} ACCEPTING(\mathbf{x}) \wedge$$

$$VALID_M(R_{input}, R_{state}, R_{nondet}, R_{work}, R_{oracle})$$

□

Theorem 23. *For every i $\Sigma_i[K]$ is as expressive as $(\Sigma_i^P)^K$ ($\Pi_i[K]$ is as expressive as $(\Pi_i^P)^K$).*

Proof: By induction. The basis is lemma 22.

Induction hypothesis: Assume the theorem to be true for all $\Sigma_i[K]$ with $i \leq n$.

Inductive step I: $\Pi_n[K]$ is as expressive as $(\Pi_n^P)^K$.

Let S be a set in $(\Pi_n^P)^K$. By definition its complement is in $(\Sigma_n^P)^K$ and so by the inductive hypothesis it can be written as a formula in $\Sigma_n[K]$. Negating this formula and applying the syntactic lemma gives the $\Pi_n[K]$ formula defining S.

Inductive step II: $\Sigma_{n+1}[K]$ is as expressive as $(\Sigma_{n+1}^P)^K$.

A machine $M \in (\Sigma_{n+1}^P)^K$ is just an **NP** machine with access to an oracle in $(\Pi_n^P)^K$. From inductive step I such an oracle can be defined by a $\Pi_n[K]$ formula. Following arguments similar to the proof of lemma 22 the set of structures it accepts can be defined by a formula

$$\psi \equiv \exists R_{input}, R_{state}, R_{nondet}, R_{work}, R_{oracle}, \exists \mathbf{x} ACCEPTING(\mathbf{x}) \wedge$$

$$\wedge VALID_M(R_{input}, R_{state}, R_{nondet}, R_{work}, R_{oracle})$$

where the predicate $VALID_M$ now contains, instead of an $FOL[K]$ expression representing the answers of a K oracle, a $\Pi_n[K]$ expression representing the answers of a $(\Pi_n^P)^K$ oracle. Applying the syntactic lemma we get that

$$ACCEPTING(\mathbf{x}) \wedge VALID_M(R_{input}, R_{state}, R_{nondet}, R_{work}, R_{oracle})$$

has an equivalent formula in $\Pi_n[K] \cup \Sigma_n[K]$. Hence ψ has an equivalent formula in Σ_{n+1}. □

Corollary 24. *$NFSOL[K]$ is as expressive as* $\mathbf{PH}^K$

4.3 Proof of Main Results

Putting the above theorems together we proved part (i) of our theorem 18 and as an immediate corollary also part (ii). We conclude by proving part (iii):

Theorem 25. *$SOL[K] = NFSOL[K]$ and hence $SOL[K]$ captures $\mathbf{PH}^K$ and has a prenex normal form.*

Proof: as $NFSOL[K] \subseteq SOL[K]$ and $NFSOL[K]$ captures $\mathbf{PH}^K$ it remains to show that every formula in $SOL[K]$ has a model checker in $\mathbf{PH}^K$. We prove by induction.

Basis: For ϕ containing no Q_K quantifiers, model checking can be done in $\mathbf{PH}$ [Sto87, GJ79].

Inductive step I: Assume Φ to be a set of formulas feasible for K such that each sub formula of Φ has a model checker in $\mathbf{PH}^K$ then $Q_K\Phi$ has a model checker in $\mathbf{PH}^K$.

Proof: Generating the structure $\mathcal{A}_\Phi$ requires a polynomial number of model checkings ([MP93, MP94]). As each model checking of a sub formula of Φ is in $\mathbf{PH}^K$ the whole operation is in $\mathbf{PH}^K$. A single query to the K oracle then completes the model checking.

2. Assume ϕ and ψ are formulas having model checkers in $\mathbf{PH}^K$ then every formula obtained from them via one application of an SOL formation rule also has a model checker in $\mathbf{PH}^K$.

Proof: For negation use closure to complement of $\mathbf{PH}^K$. For $\vee$ ($\wedge$) apply the model checkers of the components and accept if either (both) accept. For first order quantification iterate over all elements of the universe of the input structure (an increase in time complexity by no more than a factor of $|A|$). For existential second order quantification - nondeterministically produce the required relation and perform the model checking for ϕ using this relation as a substitution. For universal SO quantification use closure to negation. □

5 Natural and Unnatural Inclusions Between Logics

In section 3 we gave a logical characterization of cases where separation via oracles implies separation in the unrelativized case. In section 4 we showed that the enhancement of SOL with a Lindström quantifier for K captures the polynomial hierarchy relativized to K. We now reflect on what conclusions can and cannot be drawn from combining these two results.

As guiding examples, we investigate the relationships between the logic $FOL[HAM]$ and Σ_1 and the logic FOL[HEX] and full SOL. It is known that $FOL[HAM] = \mathbf{L}^{\mathbf{NP}}$ while $\Sigma_1 = \mathbf{NP}$ hence

$$\Sigma_1 \subseteq FOL[HAM]$$

Also, it is known that $FOL[HEX] = \mathbf{PSpace}$ while $SOL = \mathbf{PH}$ hence

$$SOL \subseteq FOL[HEX]$$

We now ask:

Question. Do such containment relationships relativize ? (continue to hold when both “sides” are enhanced with the same arbitrary K quantifier?)

A positive answer might seem appealing aesthetically, however it leads to some dramatic results: Consider K to be the Baker, Gill and Solovey ([BGS75]) oracle such that $\mathbf{P}^K \neq \mathbf{NP}^K$. Clearly

$$FOL[ATC, K] = \mathbf{P}^K \neq \mathbf{NP}^K = \Sigma_1[K]$$

If $\Sigma_1[K] \subseteq FOL[HAM, K]$ we can deduce $FOL[ATC, K] \subset FOL[HAM, K]$ and using our theorem 1 show that the HAM quantifier is undefinable in $FOL[ATC]$, hence $\mathbf{P} \neq \mathbf{NP}$! Similarly Yao's oracle [Yao85, Ko89] separating $\mathbf{P}^K$ from $\mathbf{PSpace}^K$ can be used to show that $SOL[K] \subseteq FOL[HEX, K]$ implies $\mathbf{P} \neq \mathbf{PSpace}$ (not as astounding as $\mathbf{P} \neq \mathbf{NP}$ but still quit dramatic).

A negative answer on the other hand does not imply any new complexity theoretic results, however it does give us some examples of logics $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L}_1 \subseteq \mathcal{L}_2$ but $\mathcal{L}_1[K] \nsubseteq \mathcal{L}_2[K]$. Specifically $\Sigma_1[K] \nsubseteq FOL[HAM, K]$ and $SOL[K] \nsubseteq FOL[HEX, K]$. Observe that in the second example both logics are even regular (both in the usual sense of [BF85] and in the ω sense of [MP94]).

As expected, given the difficulty of the **P**?**NP** question, the negative answer is the correct one. While we do not give a complete proof, we do give an outline: From [Ste93a, Ste93b] we have that

$$FOL[HAM] = FOL[DTC, HAM] = FOL[ATC, HAM]$$

From our theorem 1 we have that for any K

$$FOL[HAM, K] = FOL[DTC, HAM, K] = FOL[ATC, HAM, K]$$

However now from [MP94] we must conclude that these logics capture the complexity class $\mathbf{L}^{\mathbf{NP},K} = \mathbf{P}^{\mathbf{NP},K}$ - that is the class of machines in **P** (or even **L**) using two oracles - one for HAM and one for K.

Utilizing the fact that, powerful as they may be, machines in $\mathbf{P}^{\{K\}}$ cannot ask any of their oracles more than a polynomial number of queries, while an **NP** machine can ask, over all it possible runs, an exponential number of queries, we can use sparse sets and simple diagonalizations of the (enumerable) $\mathbf{P}^{\{K\}}$ machines to construct a set S such that membership in it depends on precisely those queries which the **P** machine never asks but which the **NP** machine can nondeterministically ask. Details are omitted as they are but minor variations on the construction in [BGS75] which shows K such that $\mathbf{NP}^K \not\subset \mathbf{P}^K$.

From a point of view of logics, this can be interpreted as follows: a $Q_K\Phi$ formula which contains free variables is in a sense a template for obtaining information about the set of structures K. If all these variables are first order (as must be the case when the logic is an enhancement of FOL), the amount of information obtained is bounded by a polynomial over the size of the input structure. On the other hand if we allow second order variables (as is the case when the logic is an enhancement of SOL) the amount of information becomes exponential.

Using this difference in the "power" of free variables in the two logics it becomes clear that regardless of how strong an "oracle" (generalized quantifier) K we use, we can always find another "oracle" S such that $SOL[S] \not\subset FOL[K, S]$.

This argument can be generalized by defining a notion *natural* and *unnatural* inclusion relationships between logics. An inclusion is natural if any syntactical construct of the less expressive logic has an equivalent construct in the more expressive logic. Examples of natural inclusions include $FOL[ATC] \subseteq \Sigma_1$ and $\Sigma_1 \subseteq \Sigma_2$. Examples of unnatural inclusions include $\Sigma_1 \subseteq FOL[HAM]$, $SOL \subseteq FOL[HEX]$ and $\Sigma_2 \subseteq \Sigma_1[HAM]$ (observe that $\Sigma_1[HAM]$ captures $\mathbf{NP}^{\mathbf{NP}}$ and thus has the same expressive power as Σ_2).

While for $\mathcal{L}_1 \subseteq \mathcal{L}_2$ via natural inclusions, for all K, we have $\mathcal{L}_1[K] \subseteq \mathcal{L}_2[K]$, when the inclusion is unnatural we can expect to find sets K such that $\mathcal{L}_1[K] \nsubseteq \mathcal{L}_2[K]$ although the details of constructing such sets may be technically difficult.

We conclude by returning once again to the complexity theoretic point of view and observing that similar arguments were already made by Buss in [Bus88]. There, the equivalent of the notion of a natural inclusion is an inclusion via general simulation of machines in one class by machines in the other. For this case, Buss argues that inclusion relationships should relativize (his *relativization thesis*). For the case of unnatural inclusions (not specifically mentioned in his paper), one can apply his theorem (theorem 2.5 in [Bus88]) stating that if there is a sufficient difference between complexity classes in terms of the number and type of queries they can ask of an oracle, then there is an oracle which separates them.

References

[AV88] S. Abiteboul and V. Vianu. Fixedpoint extensions of first order logic and Datalog-like languages. In *Proc. 4'th IEEE Symp. on Logic in Computer Science*, pp. 71-79, 1988.

[BF85] J. Barwise and S. Feferman, editors. *Model-Theoretic Logics.* Perspectives in Mathematical Logic. Springer Verlag, 1985.

[BGS75] T.Baker, J.Gill and R. Solovay. Relativization of the **P**?**NP** question. In SIAM J. comp. Vol. 4 pages 431-442, 1975.

[Bus88] J.F. Buss. Alternations and space–bounded computations. *Journal of Computer and System Sciences*, 36:351–378, 1988.

[CK90] C.C. Chang and H.J. Keisler. *Model Theory.* Studies in Logic, vol 73. North–Holland, 3rd edition, 1990.

[Ebb85] H.D. Ebbinghaus. Extended logics: The general framework. In *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 2. Springer Verlag, 1985.

[EFT80] H.D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic.* Undergraduate Texts in Mathematics. Springer-Verlag, 1980.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computation*, volume 7 of *American Mathematical Society Proc*, pages 27–41. Society for Industrial and Applied Mathematics, 1974.

[GJ79] M.G. Garey and D.S. Johnson. *Computers and Intractability.* Mathematical Series. W.H. Freeman and Company, 1979.

[Imm87] aN. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16(4):760–778, Aug 1987.

[Imm88] N. Immerman. Nondeterministic space is closed under complement. *SIAM Journal on Computing*, 17:935–938, 1988.

[Imm89] N. Immerman. Expressibility and parallel complexity. *SIAM Journal on Computing*, 18:625–638, 1989.

[Joh90] D.S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 1, chapter 2. Elsevier Science Publishers, 1990.

[Ko89] K.I.Ko. Relativized Polynomial Time Hierarchies Having exactly k levels. SIAM j. Comp. Vol 18 pages 392-408, 1989.

[LL76] R.E. Ladner and N. Lynch. Relativization of questions about log–space reducibility. *Mathematical Systems Theory*, 10:19–32, 1976.

[MS72] A.R.Meyer and L.J.Stockmeyer. The Equivalence problem for regular expressions with squaring requires exponential time. Proc. 13'th Ann. Symp. on Switching and Automata Theory, IEEE, Long Beach CA, pages 125-129.

[MP93] J.A. Makowsky and Y.B. Pnueli. Computable quantifiers and logics over finite structures. To appear in 'Quantifiers: Generalizations, extensions and variants of elementary logic', M. Krynicki, M. Mostowski and L.W. Szczerba eds., Kluwer Academic Publishers, 1993.

[MP94] J.A. Makowsky and Y.B. Pnueli. Oracles and Quantifiers LNCS 832 pp. 189-222, Springer 1994.

[RS81] C.W. Rackoff and J.I. Seiferas. Limitations on Separating Nondeterministic Complexity Classes. *SIAM Journal of Computing*, 10:742–745, 1981.

[RST84] W.L. Ruzzo, J. Simon, and M. Tompa. Space bounded hierarchies and probabilistic computations. *Journal of Computer and System Sciences*, 28:216–230, 1984.

[Si77] J. Simon. On Some Subrecursive Reducibilities. *Ph.D. Dissertation*, Standford Univ. March 1977.

[Ste93a] I.A. Stewart. Logical characterizations of bounded query classes I: Logspace oracle machines. *Fundamenta Informaticae*, 18:65–92, 1993.

[Ste93b] I.A. Stewart. Logical characterizations of bounded query classes II: Polynomial–time oracle machines. *Fundamenta Informaticae*, 18:93–105, 1993.

[Sto87] L. Stockmeyer. Classifying the computational complexity of problems. *Journal of Symbolic Logic*, 52(1):1–43, 1987.

[Wil86] C.B. Wilson. Parallel computation and the NC hierarchy relativized. In *Structure in Complexity Theory*, volume 223 of *Lecture Notes in Computer Science*, pages 362–382. Springer Verlag, 1986.

[Yao85] A. Yao Separating the polynomial-time hierarchy by oracles. In *Proc. 26'th IEEE Symp. on Foundations of Comp. Science*, pages 1-10. 1985.

This article was processed using the LaTeX macro package with LLNCS style