Institute for Language, Logic and Information

CENSUS TECHNIQUES ON RELATIVIZED SPACE CLASSES

Edith Spaan

ITLI Prepublication Series
for Computation and Complexity Theory CT-91-06

University of Amsterdam

The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas — A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem — Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens — A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh — Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem — Logical Syntax

1987
87-01 Jeroen Groenendijk, Martin Stokhof — Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch — Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer — Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem — Polyadic quantifiers
87-05 Víctor Sánchez Valencia — Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen — Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem — Categorial Grammar and Type Theory
87-08 Renate Bartsch — The Construction of Properties under Perspectives
87-09 Herman Hendriks — Type Change in Semantics: The Scope of Quantification and Coordination

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen — Algorithmic Information Theory
LP-88-02 Yde Venema — Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 — Year Report 1987
LP-88-04 Reinhard Muskens — Going partial in Montague Grammar
LP-88-05 Johan van Benthem — Logical Constants across Varying Types
LP-88-06 Johan van Benthem — Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch — Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof — Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen — A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe — A Blissymbolics Translation Program

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten — Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen — The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman — Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra — On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi — Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid — General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette — Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas — Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid — A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem — Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen — Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas — Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas — Towards implementing RL

Other prepublications:
X-88-01 Marc Jumelet — On Solovay's Completeness Theorem

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem — The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof — Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema — Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem — Language in Action
LP-89-05 Johan van Benthem — Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj — Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing — The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia — Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang — Dependency of Belief in Distributed Systems

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser — Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer — Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna — Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna — On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge — Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen — The Axiomatization of Randomness
ML-89-07 Dirk Roorda — Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda — Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid — Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas — Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas — On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet — A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree — A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. — Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi — A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet — Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet — On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel — The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas — Towards Functional Classification of Recursive Query Processing

Other Prepublications:
X-89-01 Marianne Kalsbeek — An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers — New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra — Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof — Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke — The Modal Theory of Inequality
X-89-06 Peter van Emde Boas — Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project

1990 SEE INSIDE BACK COVER

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

CENSUS TECHNIQUES ON RELATIVIZED SPACE CLASSES

Edith Spaan
Department of Mathematics and Computer Science
University of Amsterdam

ITLI Prepublication Series
for Computation and Complexity Theory
ISSN 0924-8374

Received April 1991

Census Techniques on Relativized Space Classes

Edith Spaan
Departments of Mathematics and Computer Science
University of Amsterdam
edith@fwi.uva.nl

Abstract

Recent results ([2], [4], [3], [8], [7]) have focused the attention of research in structural complexity theory to the development and applications of census techniques. In this paper we will develop several new techniques with which we can show the equality of $\text{SPACE}(s)^{\text{NTIME}(t)}$ classes in the sense of Ladner and Lynch [5] to known classes.

1 Introduction

Using census techniques, a number of time hierarchies have been shown to collapse. Hemachandra proved that the strong exponential hierarchy collapses to P^{NE} [2], and Kadin proved that, if co-NP $\subseteq \text{NP}^{\text{S}}$ for some sparse $S \in$ NP, then the polynomial hierarchy collapses to $\text{P}^{\text{NP}[\log n]}$ [4].

In [7], Schöning and Wagner prove a more general theorem, from which the above follow, as well as some new collapses, e.g. $\text{NEXP}^{\text{NE}}\!<\!Pol\!> = \text{P}^{\text{NE}}$ (where $<\!Pol\!>$ stands for a polynomial length restriction on the query length). This is essentially proved by computing, using binary search, the census function of the oracle.

Now what happens if we look at classes like $\text{NPSPACE}^{\text{NE}}$? This obviously depends on the definition of relativized space classes. If we assume that the space bound applies to the query tape, it follows from [7] that $\text{NPSPACE}^{\text{NE}} = \text{P}^{\text{NE}}$.

However, if we use the classical model of Ladner and Lynch [5], i.e. the space bound doesn't apply to the query tape, a SPACE(s) machine can ask queries of length 2^s. We know less about the structure of these classes, since Savitch's theorem [6] and the Immerman/Szelepcsényi result [3], [8] do not relativize in this setting.

In this paper, we will look at classes $\text{DSPACE}(s)^{\text{NTIME}(t)}$ and $\text{NSPACE}(s)^{\text{NTIME}(t)}$ in the Ladner and Lynch sense. For DSPACE machines, we can still apply a variation of the census method, since a DSPACE(s) machine can query at most $2^{c \cdot s}$ strings in its entire oracle tree. For instance, we will prove that $\text{PSPACE}^{\text{NP}} = \text{P}^{\text{NE}}$. In the computation tree of an NSPACE(s) machine however, *all* queries of exponential length can occur. We will prove that $\text{NPSPACE}^{\text{NP}} = \text{NEXP}^{\text{NP}}$, since an NPSPACE machine can guess a computation of the NEXP machine and all certificates for the YES-queries in this computation. This method won't work if the oracle is too powerful. Using a new census technique, we will show that, even though in this model PSPACE = NPSPACE doesn't relativize, $\text{NPSPACE}^{\text{NE}} = \text{PSPACE}^{\text{NE}}$.

The inclusions $\mathrm{PSPACE}^{\mathrm{NP}} \subseteq \mathrm{NPSPACE}^{\mathrm{NP}}$ and $\mathrm{PSPACE}^{\mathrm{NE}} \subseteq \mathrm{EXP}^{\mathrm{NE}}$ remain open. A separation would imply that $\mathrm{P} \neq \mathrm{NP}$, and hence such a result will be hard to obtain. In the last section we will prove relativized separations for these classes, so equality of these classes is probably hard to prove as well.

2 Preliminaries

We assume that the reader is familiar with the standard Turing machine model. An *oracle* machine is a multi-tape Turing machine with an input tape, an output tape, work tapes, and a *query* tape. Oracle machines have three distinguished states QUERY, YES and NO, which are explained as follows: at some stage(s) in the computation the machine may enter the state QUERY and then goes to the state YES or goes to the state NO depending on the membership of the string currently written on the query tape in a fixed *oracle* set.

The *computation tree* of an oracle machine M on input x, is the tree of all possible computations of M in input x which is generated by assuming both YES and NO answers to the queries.

An oracle Turing machine M runs in time $t(n)$, iff for all x every path of M on input x halts in $\leq t(|x|)$ steps.

An oracle Turing machine M runs in space $s(n)$, iff for all x every path of the computation tree of machine M on input x halts, and scans no more than $s(|x|)$ cells of the work tapes. The tape cells of the input- and oracle tape are not counted. Note that M runs in time $2^{c \cdot s(n)}$, for some constant c, and therefore the length of the queries that M can ask is bounded by $2^{c \cdot s(n)}$.

Define $\mathrm{DTIME}^A(t(n))$ ($\mathrm{DSPACE}^A(s(n))$) as the class of sets that are accepted by deterministic Turing machines which run in time $t(n)$ (space ($s(n)$)), and use A as oracle set. Let $\mathrm{NTIME}^A(t(n))$ and $\mathrm{NSPACE}^A(s(n))$ be the corresponding nondeterministic classes.

In the sequel, we will consider the following classes (*Pol* is the class of all polynomial functions)

$$\begin{aligned}
\mathrm{LOGSPACE}^A &= \bigcup_{c=1}^{\infty} \mathrm{DSPACE}^A(c \cdot \log(n)) \\
\mathrm{DLIN}^A &= \bigcup_{c=1}^{\infty} \mathrm{DTIME}^A(c \cdot n) \\
\mathrm{P}^A &= \bigcup_{p \in Pol} \mathrm{DTIME}^A(p(n)) \\
\mathrm{PSPACE}^A &= \bigcup_{p \in Pol} \mathrm{DSPACE}^A(p(n)) \\
\mathrm{E}^A &= \bigcup_{c=1}^{\infty} \mathrm{DTIME}^A(2^{cn}) \\
\mathrm{EXP}^A &= \bigcup_{p \in Pol} \mathrm{DTIME}^A(2^{p(n)})
\end{aligned}$$

And their nondeterministic analogs $\mathrm{NLOGSPACE}^A$, NLIN^A, NP^A, $\mathrm{NPSPACE}^A$, NE^A and NEXP^A. For any two classes C_1, C_2, define $C_1^{C_2} := \bigcup_{A \in C_2} C_1^A$.

Since we are interested in simulating one type of machine on another machine, we will need the space and time bounds to be well-behaved.

A function f is *time-constructible*, iff there is a Turing machine which on every input of length n halts in exactly $f(n)$ steps.

A function f is *space-constructible*, iff there is a Turing machine which on every input of length n scans exactly $f(n)$ cells of the work tapes.

3 Collapses

We will first consider deterministic space classes. Recall that in our definition the space bound doesn't hold for the query tape, which enables a space(s) machine to ask queries of length $2^{s(n)}$ on inputs of length n. An application of the binary search technique in [7], leads to the following theorem:

Theorem 1 $\mathrm{DSPACE}(s(n))^{\mathrm{NTIME}(t(n))} \subseteq \mathrm{DTIME}(Pol\ (s(n)))^{\mathrm{NTIME}(Pol\ (t(2^n)))}$

For s time constructible, t at least linear.

Proof: Let M_1 be an arbitrary DSPACE($s(n)$) oracle machine, and M_2 an arbitrary NTIME($t(n)$) machine. The number of IDs of a DSPACE($s(n)$) machine on input x is bounded by $2^{c \cdot s(|x|)}$. Since the machine is deterministic, each ID can lead to at most one query, i.e. there are at most $2^{c \cdot s(|x|)}$ different queries in the computation tree of $M_1(x)$.

Define oracle set A as follows:

$$A = \{x\#k\#0^{c \cdot s(|x|)} : \text{at least } k \text{ queries in the computation tree of } M_1(x) \text{ are in } L(M_2)\}.$$

Using oracle A, we can by binary search compute the exact number of queries in the computation tree of $M_1(x)$ that are answered YES, in DTIME($Pol\ (s(n))$).

Decision procedure for A:

input $x\#k\#0^{c \cdot s(|x|)}$ ($k \leq 2^{c \cdot s(|x|)}$)
guess k different strings $q_1, \ldots, q_k$ of length $\leq 2^{c \cdot s(|x|)}$
verify that each q_i is asked in the computation tree of M_1 on input x
i.e.Guess a path of M_1 on input x
And check that q_i is asked on this path
For each q_i, guess an accepting path of M_2 on input q_i

Nondeterministic time complexity of A:

$2^{c \cdot s(\lvert x \rvert)}$	(bound on the value of k)
$\times 2^{c \cdot s(\lvert x \rvert)}$	(maximal length of an M_1 path on input x)
$\times t(2^{c \cdot s(\lvert x \rvert)})$	(time to verify that a query $\in \mathrm{L}(\mathrm{M}_2)$

Since the length of the input is larger than $c \cdot s(|x|)$, and t is at least linear, A can be accepted by an NTIME($Pol\ (t(2^n))$) bounded Turing machine.

Define oracle set B such that: If k is the exact number of different queries in the computation tree of $M_1(x)$ that are answered YES, then $x\#k\#0^{c \cdot s(|x|)} \in B$ iff $x \in \mathrm{M}_1{}^{\mathrm{L}(\mathrm{M}_2)}$. Therefore, we can recognize $\mathrm{M}_1{}^{\mathrm{L}(\mathrm{M}_2)}$ with a DTIME($Pol\ (s(n))$) machine that uses $A \oplus B$ as oracle.

Decision procedure for B:

input $x\#k\#0^{c\cdot s(|x|)}$ $(k \leq 2^{c\cdot s(|x|)})$

guess k different strings $q_1, \ldots, q_k$ of length $\leq 2^{c\cdot s(|x|)}$

verify that each q_i is really asked by M_1 on input x. as in A

and verify that each q_i is a YES string

Simulate M_1 on input x

For each query q that is asked do

  if $q = q_i$ for some i

  then proceed in the YES state

  else proceed in the NO state

ACCEPT iff the simulation accepts.

Nondeterministic time complexity of B for inputs of length n:

$Pol\ (t(2^n))$	for guessing and verifying the k YES-queries
$+2^n$	to simulate M_1 on input x.

which is in $Pol\ (t(2^n))$.

⊠

Corollary 2 $P^{NE} = PSPACE^{NP}$

$P^{NE} \subseteq PSPACE^{NP}$	by padding the queries
$PSPACE^{NP} \subseteq P^{NE}$	by theorem 1

Now we turn our attention to nondeterministic space bounded oracle machines. In the previous proof it was essential that the number of different queries in the computation tree is bounded by $2^{c\cdot s(n)}$. An NSPACE($s(n)$) machine, however, can query all strings of length $2^{s(n)}$ in its computation tree, which may lead to a double exponential number of queries in the computation tree.

In fact, it turns out that $NPSPACE^{NP}$ can recognize the same languages as $NEXP^{NP}$, because the NPSPACE machine can guess an accepting path of the NEXP machine and accepting paths of the NP-oracle machine on the YES queries. We will now prove a more general result of which the above is a corollary.

Theorem 3 $NTIME(s(n))^{NTIME(t(n))} \subseteq NSPACE(O(\log(t(s(n)))))^{NLIN}$
For $\log(t(s(n)))$ space constructible.

Proof: Let M_1 be an arbitrary NTIME($s(n)$) oracle machine, and M_2 an arbitrary NTIME($t(n)$) machine.

We will prove that ${M_1}^{L(M_2)}$ can be recognized by a nondeterministic space($O(\log t(s(n)))$) machine M with oracle A $\in$ NLIN.

Our NSPACE-machine M will work as follows: on input x, copy x to the oracle tape and write an arbitrary string σ of length $\leq s(|x|)$ on the oracle tape. This is the guessed accepting path of M_1 on input x. Guess $k \leq s(|x|)$: the number of YES-queries in this computation on this path. Write k arbitrary strings $\sigma_1, \ldots, \sigma_k$ of length $\leq t(s(|x|))$ on the oracle tape (separated by #). These are the guessed accepting paths of M_2 on the k Yes-queries. To ensure that the oracle has enough time, write $0^{t(s(|x|))}$ on the oracle tape. All this can be done in NSPACE($O(\log(t(s(|x|))))$), since the only thing we have to remember is the length of the query, and the value of k.

The oracle set A will be constructed in such a way that: $x\#\sigma\#\sigma_1\#\cdots\#\sigma_k \notin$ A iff σ encodes an accepting path of M_1 on input x, with exactly k YES-queries, σ_i is an accepting path of M_2 on the i-th YES-query, and the NO-queries on the M_1 path σ are really NO-queries.

Machine M accepts iff the answer to the query is NO. $x \in {M_1}^{L(M_2)}$ iff some query asked by M is not in A. Thus, $x \in {M_1}^{L(M_2)}$ iff $x \in M^A$.

It remains to prove that $A \in$ NLIN

Input $x\#\sigma\#\sigma_1\#\ldots\#\sigma_k\#0^{2^{p(|x|)}}$
if σ does not encode an accepting path of M_1 on input x
or there are not exactly k Yes-queries on this path
then ACCEPT.
else determine $q_1, \ldots, q_k$: the k Yes-queries in σ.
if σ_i does not encode an accepting path of M_2 on input q_i
then ACCEPT.
else guess a NO-query q in computation σ,
and guess a computation of M_2 on input q.
ACCEPT iff this computation accepts.

Verifying the computations $\sigma, \sigma_1, \ldots \sigma_k$ and determining the queries can be done in linear deterministic time in the length of the input.

Since the length of any query of M_1 on input x is bounded by $s(|x|)$, guessing a computation of M_2 on a NO-query can be done in nondeterministic time $t(s(|x|))$, which is linear in the length of the input. ⊠

Corollary 4 $P^{NE} = PSPACE^{NP} \subseteq EXP^{NP} \subseteq NPSPACE^{NP} = NEXP^{NP}$.

Corollary 5 $NLOGSPACE^{NP} = NP^{NP}$

Theorem 3 doesn't give any nice equivalences for classes like $NPSPACE^{NE}$. Surprisingly, it turns out that this class doesn't have more power than $PSPACE^{NE}$, as follows from the following theorem.

Theorem 6 $NSPACE(s(n))^{NTIME(t(n))} \subseteq DSPACE(O(s(n)))^{NTIME(Pol\ (t(n)))}$
For s time constructible, t at least exponential.

Proof: Let M_1 be an arbitrary NSPACE($s(n)$) oracle machine, and M_2 an arbitrary NTIME($t(n)$) machine. We assume that IDs of M_1 with an empty query tape are marked. The number of IDs of M_1 on inputs of length n is bounded by $2^{c \cdot s(n)}$, for some constant c.

Since M_1 can ask $2^{2^{c \cdot s(n)}}$ queries on inputs of length n, we can't keep track of the number of YES queries with a DSPACE($O(s(n))$) machine.

The trick we will use is that we won't look at the number of queries of M_1, but at pairs of IDs of M_1 between which a query can be generated, i.e. we look at pairs $<id, id'>$ such that id has an empty query tape, id' is in the QUERY state, and there exists a queryless path from id to id'.

$x \in \mathrm{M_1}^{\mathrm{L(M_2)}}$ iff there exists a sequence of IDs $id_0, id'_0, id_1, id'_1, \ldots, id_i, id'_i, id_{i+1}$ such that:

- id_0 is the initial ID of M_1 on input x
- id'_j is in the QUERY state ($\forall j \leq i$)
- $\forall j \leq i$ one of the following situations holds
  - either id_{j+1} is the YES-successor of id'_j and there exists a queryless path from id_j to id'_j where the generated query is in $L(\mathrm{M_2})$
  - or id_{j+1} is the NO-successor of of id'_j and there exists a queryless path from id_j to id'_j where the generated query is not in $L(\mathrm{M_2})$
- There exists a queryless accepting path from id_{i+1} to an accepting ID

Our DSPACE-machine will work as follows on input x: First we compute (e.g. by binary search) the exact number of ID pairs $<id, id'>$ such that:

(*) id and id' represent space $s(|x|)$-bounded IDs
id has an empty query tape and id' is the QUERY state
and for each queryless path of $M_1(x)$ from id to id',
the answer to the generated query is YES.

$A := \{x\#k\#0^{2^{c \cdot s(|x|)}}$: there are at least k pairs of IDs that satisfy (*)$\}$.

The number of ID pairs is bounded by $2^{2c \cdot s(n)}$. Therefore, (e.g. by binary search), we can determine the exact number of pairs fulfilling (*) in DSPACE($O(s(|x|))$ with oracle A.

Decision procedure for A:

```
input x#k#0^(2^(c·s(|x|)))
Guess k different pairs of space s(|x|) IDs of M1
For each guessed pair <id, id'> do
    if      id does not have an empty query tape
            or id' is not in the query state
    then    REJECT
    else    for each queryless path of M(x) from id1 to id2 of length ≤ 2^(c·s(|x|)) do
                Determine query q
                Guess computation of M2 on input q.
                If this computation rejects, then REJECT
ACCEPT
```

Nondeterministic time needed for inputs of length n:

2^n	(maximal value of k)
$\times 2^n$	(number of paths between two IDs)
$\times t(n)$	(time to simulate M_2 on a query)

Since t is at least exponential, A can be accepted in NTIME($Pol\ (t(n))$).

Define another oracle set B:

$$B := \{x\#k\#0^{2^{c\cdot s(|x|)}} : \text{if } k \text{ is the } exact \text{ number of ID pairs fulfilling (*),} \text{ then } {M_1}^{M_2} \text{ accepts } x\}$$

Using oracle $A \oplus B$, we can recognize ${M_1}^{L(M_2)}$ in DSPACE($O(s)$).

Decision procedure for B:

Input: $x\#k\#0^{2^{c\cdot s(|x|)}}$
Guess k different pairs of space $s(|x|)$-bounded IDs of M_1
    $< ID_1, ID'_1 >, \ldots, < ID_k, ID'_k >$
And verify nondeterministically that these pairs fulfill (*) as in A
Guess a sequence of $\leq 2^{c\cdot s(|x|}$ space $s(|x|)$ bounded IDs
  $id_0, id'_0, id_1, id'_1, \ldots, id_i, id'_i, id_{i+1}$ such that:
    id_0 is the initial ID of M_1 on input x
    id'_j is in the query state
    id_{j+1} is a successor of id'_i
    there exists a queryless path from id_j to id'_j
    there exists a queryless accepting path from id_{i+1}
Now we only have to check that the queries are answered correctly:
for $j := 1$ to i do
  if $< id_j, id'_j >$ is in the list
  then the answer to the query must be YES
         if id_{j+1} is in the NO-state then REJECT
  else if id_{j+1} is in the YES-state
         then guess a queryless path from id_j to id'_j
            and guess an accepting computation of M_2 on the generated query.
ACCEPT

Nondeterministic time needed for inputs of length n:

$Pol\ (t(n))$	(for guessing and verifying the k ID pairs fulfilling (*))
$\times n$	(number of IDs in the sequence)
$\times t(n)$	(time to simulate M_2 on a query)

Since t is at least exponential, B can be accepted in NTIME($Pol\ (t(n))$). ⊠

Corollary 7 LOGSPACE^{NE} = NLOGSPACE^{NE}.

Combining theorem 1 with theorem 6, we obtain the following analog of theorem 1 for nondeterministic space classes.

Corollary 8 $\mathrm{NSPACE}(s(n))^{\mathrm{NTIME}(t(n))} \subseteq \mathrm{DTIME}(Pol\ (s(n)))^{\mathrm{NTIME}(Pol\ (t(2^n)))}$

For s time constructible, t at least exponential.

Corollary 9 $\mathrm{P}^{\mathrm{NTIME}(2^{2^{Pol}})} = \mathrm{PSPACE}^{\mathrm{NE}} = \mathrm{NPSPACE}^{\mathrm{NE}} \subseteq \mathrm{EXP}^{\mathrm{NE}} = \mathrm{NEXP}^{\mathrm{NE}}$

$\subseteq$:	By padding the queries
$\mathrm{NPSPACE}^{\mathrm{NE}} \subseteq \mathrm{P}^{\mathrm{NTIME}(2^{2^{Pol}})}$:	By corollary 8
$\mathrm{NEXP}^{\mathrm{NE}} \subseteq \mathrm{EXP}^{\mathrm{NE}}$:	Follows from [7]

4 Relativized Separations

We'll now focus on the following relations between classes as proved in corollary 4 and 9 of the previous section:

1. $\mathrm{P}^{\mathrm{NE}} = \mathrm{PSPACE}^{\mathrm{NP}} \subseteq \mathrm{EXP}^{\mathrm{NP}} \subseteq \mathrm{NPSPACE}^{\mathrm{NP}} = \mathrm{NEXP}^{\mathrm{NP}}$
2. $\mathrm{P}^{\mathrm{NTIME}(2^{2^{Pol}})} = \mathrm{PSPACE}^{\mathrm{NE}} = \mathrm{NPSPACE}^{\mathrm{NE}} \subseteq \mathrm{EXP}^{\mathrm{NE}} = \mathrm{NEXP}^{\mathrm{NE}}$

We can't hope to prove any of the inclusions to be strict, since that would imply that $\mathrm{P} \neq \mathrm{NP}$.

$$\mathrm{P} = \mathrm{NP} \Rightarrow \begin{cases} \mathrm{EXP}^{\mathrm{NP}} = \mathrm{EXP}^{\mathrm{P}} = \mathrm{EXP} \subseteq \mathrm{P}^{\mathrm{NE}} \\ \mathrm{EXP}^{\mathrm{NE}} = \mathrm{EXP}^{\mathrm{E}} = \mathrm{DTIME}(2^{2^{Pol}}) \subseteq \mathrm{P}^{\mathrm{NTIME}(2^{2^{Pol}})} \\ \mathrm{NEXP}^{\mathrm{NP}} = \mathrm{NEXP}^{\mathrm{P}} = \mathrm{NEXP} \subseteq \mathrm{EXP}^{\mathrm{NP}} \end{cases}$$

We will construct for each inclusion an oracle such that the inclusion becomes strict.

Theorem 10 *There exists a set A such that:* $\mathrm{EXP}^{\mathrm{NP}^A} \neq \mathrm{NEXP}^{\mathrm{NP}^A}$.

Using a similar construction as in [1], where an oracle is constructed such that $\mathrm{P}^{\mathrm{NP}^A} \neq \mathrm{NP}^{\mathrm{NP}^A}$, we can diagonalize such that:

$$L_A := \{0^n : \exists y(|y| = 2^n \wedge \forall z(|z| = |y| \rightarrow yz \in A))\} \notin \mathrm{EXP}^{\mathrm{NP}^A}$$

Theorem 11 *There exists a set A such that* $\mathrm{P}^{\mathrm{NE}^A} \neq \mathrm{EXP}^{\mathrm{NP}^A}$.

We will use a similar construction as in [4], where it is shown that for any function $f \in o(\log)$ there exists a relativized world such that: $\mathrm{P}^{\mathrm{NP}[f(n)]} \neq \mathrm{P}^{\mathrm{NP}[\log n]}$.

For every oracle A, let L_A be the following language:

$$L_A := \{0^n : \text{the lexicographically largest string } y \text{ of length } 2^n \text{ such that } \exists z(|z| = 2^n \text{ and } yz \in A), \text{ is odd}\}$$

For every oracle A, $L_A \in \mathrm{EXP}^{\mathrm{NP}^A}$:

NP^{A} machine: input y
guess y', z such that $|y| = |y'| = |z|$ and $y' \geq y$
Accept if and only if $y'z \in A$.

EXP machine: binary search on strings of length 2^n.
To determine the lexicographically largest string y of length 2^n such that: $\exists z$ such that $|z| = 2^n$ and $yz \in A$
Accept if and only if last bit of y is 1.

We will construct A such that $L_A \notin \mathrm{P}^{\mathrm{NE}^A}$. Note that $\mathrm{P}^{\mathrm{NE}^A} = \mathrm{P}^{\mathrm{K}(A)}$, where $\mathrm{K}(A)$ is the standard many-one complete set for NE^{A}.

$$\mathrm{K}(A) := \{< M, x, k > \mid M \text{ is a nondeterministic machine that accepts } x \text{ with oracle } A \text{ in at most } k \text{ steps}\}$$

Let $M_1, M_2, \ldots$ be a recursive enumeration of P oracle machines. Our construction will ensure that for each machine M_i, there exists an integer n_i such that:

$$M_i^{\mathrm{K}(A)} \text{ accepts } 0^{n_i} \text{ iff } 0^{n_i} \notin L_A$$

stage 0: $A := \emptyset$

stage i: Diagonalizing against the i-th machine M_i. Let $p(n)$ be a polynomial such that $p(n)$ bounds the running time of M_i on inputs of length n. The number of queries in the computation tree is bounded by $2^{p(n)}$. Choose n_i such that:

1. None of the previous diagonalization steps deal with strings of length $\geq 2^{n_i}$.
2. $2^{2^{n_i}} > 2^{2p(n_i)}$

During stage n_i, we will only add strings of length $2 \cdot 2^{n_i}$ to A. At any iteration of the loop σ_{pref} will be the lexicographically largest string of length 2^{n_i}, that occurs as a prefix of a string of length $2 \cdot 2^{n_i}$ in A. In the construction, some strings will be reserved for the complement. We will keep these strings in set A_c.

```
A_c := ∅
Consider the tree of M_i on input 0^{n_i}.
All queries in the tree are unmarked.
σ_pref := 0^{2^{n_i}}.
A := A ∪ {0^{2·2^{n_i}}}
LOOP: For each unmarked query q =< M, x, k > in the computation tree of M_i do
          if      q ∈ K(A)
          then    Take some accepting path of M^A on input x
                  *Length of path ≤ k ≤ q ≤ 2^{p(n_i)} *
                  For each NO-query q' on this path do
                      A_c := A_c ∪ {q'}
                  Mark query q
       If      M_i^{K(A)}(0^{n_i}) accepts and σ_pref is even
               or M_i^{K(A)}(0^{n_i}) rejects and σ_pref is odd
       then    We are done (0^{n_i} is a counterexample for M_i)
       else    σ_pref := lexicographic successor of σ_pref of length 2^{n_i}
               Choose some string τ of length 2 · 2^{n_i}, such that
                   σ_pref is a prefix of τ and τ ∉ A_c
               A := A ∪ {τ}
               Go back to LOOP.
```

We only add strings to A_c if we mark some query of M_i, and for each query, we add at most $2^{p(n_i)}$ strings to A_c. Since we never unmark queries and the number of queries in the computation tree of M_i is bounded by $2^{p(n_i)}$, we add at most $2^{2p(n_i)}$ strings to A_c during stage n_i. Since $2^{2^{n_i}} > 2^{2p(n_i)}$, for any prefix σ of length 2^{n_i}, there must exist a string τ of length $2 \cdot 2^{n_i}$ such that σ is a prefix of τ, and τ is never added to A_c.

Now we only have to prove that the number of iterations of the loop is bounded by $2^{2^{n_i}}$. Suppose we are at some iteration of the loop, and we do not exit the loop. Suppose the value of σ_{pref} at this iteration is σ. Then, the path of M_i is accepting iff σ is odd. Suppose that at the next iteration of the loop no new queries of M_i are marked. Then the path of M_i in this iteration is the same as in the previous iteration of the loop. But then M_i accepts iff σ is odd iff the lexicographic successor of σ is even, and we exit the loop. Therefore, the number of iterations of the loop is bounded by the number of queries in the computation tree of M_i. Since the number of queries is bounded by $2^{p(n_i)}$, the number of iterations of the loop is certainly bounded by $2^{2^{n_i}}$ as required.

Theorem 12 *There exists a set A such that:* $\mathrm{P}^{\mathrm{NTIME}^A(2^{2^{Pol}})} \neq \mathrm{EXP}^{\mathrm{NE}^A}$

Use the same construction as in the previous theorem, this time using the following language L_A.

$$L_A := \{0^n : \text{ the lexicographically largest string } y \text{ of length } 2^n \text{ such that } \exists z(|z| = 2^{2^n} \text{ and } yz \in A), \text{ is odd}\}$$

Acknowledgements I would like to thank Lane Hemachandra, Leen Torenvliet, Peter van Emde Boas and Rineke Verbrugge for fruitful discussions and proof reading.

References

[1] Heller, H., *Relativized polynomial hierarchies extending two levels*, Mathematical Systems Theory 17 (1984), 71–84.

[2] Hemachandra L.A., *The strong exponential hierarchy collapses*, Proc 19th STOC conference (1987), 110–122.

[3] Immerman, N. *Nondeterministic space is closed under complementation.* SIAM J. on Computing 17 (1988) pp. 935–938.

[4] Kadin, J., $\mathrm{P}^{\mathrm{NP}[\log n]}$ *and sparse Turing-complete sets for NP*, Proc. of 2nd conference on Structure in Complexity Theory (1987), 33–40.

[5] Ladner, R.E., Lynch, N.A., *Relativization of questions about Logspace computability.* Mathematical Systems Theory 10 (1976) 19–32.

[6] Savitch W.J. *Relationships between nondeterministic and deterministic tape complexities.* J. Computer and System Sciences 4 (1970) 177–192.

[7] Schöning, U., Wagner, K.W., *Collapsing oracle hierarchies, census functions and logarithmically many queries*, Proc. 5th Symp. on Theoretical Aspects of Computer Science (1988) 91–98.

[8] Szelepcsényi, R. *The method of forcing for nondeterministic automata.* Bulletin of the EATCS 33 (1987) pp. 96–100.

The ITLI Prepublication Series

1990

Logic, Semantics and Philosophy of Language

LP-90-01 Jaap van der Does	A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof	Dynamic Montague Grammar
LP-90-03 Renate Bartsch	Concept Formation and Concept Composition
LP-90-04 Aarne Ranta	Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn	Nominal Tense Logic
LP-90-06 Gennaro Chierchia	The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia	Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks	Flexible Montague Grammar
LP-90-09 Paul Dekker	The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen	Models for Discourse Markers
LP-90-11 Johan van Benthem	General Dynamics
LP-90-12 Serge Lapierre	A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang	Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof	Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke	The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast	Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker	Existential Disclosure, Implicit Arguments in Dynamic Semantics

Mathematical Logic and Foundations

ML-90-01 Harold Schellinx	Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten	A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema	Relational Games
ML-90-04 Maarten de Rijke	Unary Interpretability Logic
ML-90-05 Domenico Zambella	Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten	Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke	A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx	Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani	Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen	Randomness in Set Theory
ML-90-11 Paul C. Gilmore	The Consistency of an Extended NaDSet

Computation and Complexity Theory

CT-90-01 John Tromp, Peter van Emde Boas	Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette	A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar	Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet	Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast	Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas	Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets	Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas	Physiological Modelling using RL
CT-90-09 Roel de Vrijer	Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting

Other Prepublications

X-90-01 A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke	Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev	On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04	Annual Report 1989
X-90-05 Valentin Shehtman	Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy	Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov	The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev	Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov	On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas	An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone	Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke	Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev	Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova	Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra	Lectures on Linear Logic

1991

Mathematical Logic and Foundations

ML-91-01 Yde Venema	Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge	On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella	On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx	Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra	History of Constructivism in the Twentieth Century

Computation and Complexity Theory

CT-91-01 Ming Li, Paul M.B. Vitanyi	Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitanyi	How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitanyi	Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast	Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast	Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan	Census Techniques on Relativized Space Classes

Other Prepublications

X-91-01 Alexander Chagrov, Michael Zakharyaschev	The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev	On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov	Subalgebras of Diagonizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev	Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem	Temporal Logic
X-91-06	Annual Report 1990
X-91-07 A.S. Troelstra	Lectures on Linear Logic, Errata and Supplement