BOUNDDED REDUCTIONS

Harry Buhrman
Edith Spaan
Leen Torenvliet

ITLI Prepublication Series
for Computation and Complexity Theory CT-90-04

University of Amsterdam
1986
86-01  The Institute of Language, Logic and Information
86-02  A Semantic Model for Integration and Modularization of Rules
86-03  Categorial Grammar and Lambda Calculus
86-04  Some Complete Logics for Branching Time, Part I: Well-founded Tense
86-05  Logical Syntax: Forward Looking Operators
86-06  Frame Representations and Discourse Representations
86-07  Unique Normal Forms for Lambda Calculus with Surjective Pairing
86-08  Polyadic quantifiers
86-09  Temporal Adverbs in the Two Track Theory of Time
86-10  Categorial Grammar and Type Theory
86-11  The Construction of Properties under Perspectives
86-12  Type Change in Semantics: The Scope of Quantification and the Coordination of Properties
86-13  Expressiveness and Completeness of an Interval Tense Logic
86-14  Year Report 1987
86-15  Going partial in Montague Grammar
86-16  Logical Constants across Varying Types
86-17  Semantic Parallels in Natural Language and Computation
86-18  Context and Inference in Dynamic Semantics
86-19  A mathematical model for the CAT framework of Eurotra
86-20  A Blisssymbolics Translation Program
86-21  Computation in Fragments of Intuitionistic Propositional Logic
86-22  Machine Models and Simulations (revised version)
86-23  Two-dimensional Modal Logics for Relation Algebras and Temporal Logics of Intervals
86-24  Language in Action
86-25  Intensional Lambek Calculi: Theory and Application
86-26  The Adequacy Problem for Sequential Propositional Logic
86-27  Peirce's Propositional Logic: From Algebra to Graphs
86-28  Dependence of Belief in Distributed Systems
86-29  Explicit Fixed Points for Interpretability Logic
86-30  Extending the Lambek Calculus with Surjective Pairing is conservative
86-31  The Axiomatization of Randomness
86-32  Elementary Inductive Definitions in HA: from Strictly Positive to Monotone
86-33  Investigations into Classical Linear Logic
86-34  Provable Fixed points in 1\alpha + 1\Omega
86-35  The Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Pres. Version)
86-36  Honest Reducibility, Completeness and Nondeterministic Complexity Classes
86-37  Finding Isomorphisms between Finite Fields
86-38  A Theory of Recursive Functionality under Recursive Functionality
86-39  A Theory of Recursive Functionality under Recursive Functionality
86-40  Adhesive Resource Bounded Computations
86-41  The Rule Language RL/I
86-42  Towards Functional Classification of Recursive Query Processing
86-43  New Foundations: a Survey of Quine's Set Theory
86-44  Index of the Heyting Nachlass
86-45  Dynamic Montague Grammar, a first sketch
86-46  The Modal Theory of Inequality
86-47  Een Relationele Semantiek voor Conceptueel Modellen: Het RL-project
86-48  SEE INSIDE BACK COVER

1988
88-01  Logic, Semantics and Philosophy of Language: The Fine-Structure of Categorial Semantics
88-02  Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
88-03  Two-dimensional Modal Logics for Relation Algebras and Temporal Logics of Intervals
88-04  Language in Action
88-05  Intensional Lambek Calculi: Theory and Application
88-06  The Adequacy Problem for Sequential Propositional Logic
88-07  Peirce's Propositional Logic: From Algebra to Graphs
88-08  Dependence of Belief in Distributed Systems
88-09  Explicit Fixed Points for Interpretability Logic
88-10  Extending the Lambek Calculus with Surjective Pairing is conservative
88-11  The Axiomatization of Randomness
88-12  Elementary Inductive Definitions in HA: from Strictly Positive to Monotone
88-13  Investigations into Classical Linear Logic
88-14  Provable Fixed points in 1\alpha + 1\Omega
88-16  Honest Reducibility, Completeness and Nondeterministic Complexity Classes
88-17  Finding Isomorphisms between Finite Fields
88-18  A Theory of Recursive Functionality under Recursive Functionality
88-19  A Theory of Recursive Functionality under Recursive Functionality
88-20  Adhesive Resource Bounded Computations
88-21  The Rule Language RL/I
88-22  Towards Functional Classification of Recursive Query Processing
88-23  New Foundations: a Survey of Quine's Set Theory
88-24  Index of the Heyting Nachlass
88-25  Dynamic Montague Grammar, a first sketch
88-26  The Modal Theory of Inequality
88-27  Een Relationele Semantiek voor Conceptueel Modellen: Het RL-project
88-28  SEE INSIDE BACK COVER

1989
89-01  Logic, Semantics and Philosophy of Language: The Fine-Structure of Categorial Semantics
89-02  Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
89-03  Two-dimensional Modal Logics for Relation Algebras and Temporal Logics of Intervals
89-04  Language in Action
89-05  Intensional Lambek Calculi: Theory and Application
88-06  The Adequacy Problem for Sequential Propositional Logic
88-07  Peirce's Propositional Logic: From Algebra to Graphs
88-08  Dependence of Belief in Distributed Systems
88-09  Explicit Fixed Points for Interpretability Logic
88-10  Extending the Lambek Calculus with Surjective Pairing is conservative
88-11  The Axiomatization of Randomness
88-12  Elementary Inductive Definitions in HA: from Strictly Positive to Monotone
88-13  Investigations into Classical Linear Logic
88-14  Provable Fixed points in 1\alpha + 1\Omega
88-16  Honest Reducibility, Completeness and Nondeterministic Complexity Classes
88-17  Finding Isomorphisms between Finite Fields
88-18  A Theory of Recursive Functionality under Recursive Functionality
88-19  A Theory of Recursive Functionality under Recursive Functionality
88-20  Adhesive Resource Bounded Computations
88-21  The Rule Language RL/I
88-22  Towards Functional Classification of Recursive Query Processing
89-03  New Foundations: a Survey of Quine's Set Theory
89-04  Index of the Heyting Nachlass
89-05  Dynamic Montague Grammar, a first sketch
89-06  The Modal Theory of Inequality
89-07  Een Relationele Semantiek voor Conceptueel Modellen: Het RL-project
89-08  SEE INSIDE BACK COVER
BOUNDDED REDUCTIONS

Harry Buhrman
Edith Spaan
Leen Torenvliet
Department of Mathematics and Computer Science
University of Amsterdam

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

Received December 1990
Abstract

We study properties of resource- and otherwise bounded reductions and corresponding completeness notions on nondeterministic time classes which contain exponential time. As it turns out most of these reductions can be separated in the sense that their corresponding completeness notions are different. There is one notable exception. On nondeterministic exponential time 1-truth table and many-one completeness is the same notion.
1 Introduction

Efficient reducibilities and completeness are two of the central concepts of complexity theory. Since the first use of polynomial time bounded Turing reductions by Cook [4] and the introduction of polynomial time bounded many-one reductions by Karp[6], considerable effort has been put in the investigation of properties and the relative strengths of different reductions and corresponding completeness notions. In 1975 Ladner, Lynch and Selman [8] gave an extensive survey of different types of reductions and differences between these reductions on $E (= \cup_{c \in \mathbb{N}} \text{DTIME}(2^{cn}))$. However, they did not present any conclusions concerning any differences in complete sets for these various reductions. In particular they left open the question of whether these different reductions yield different complete sets. In 1987, Watanabe [10] building upon earlier work of L. Berman [1], proved almost all possible differences between the polynomial-time completeness notions on $E$ and larger deterministic time classes.

The question of differentiating between complete sets for nondeterministic time classes with respect to various bounded reductions was considered by Buhrman, Homer and Torenvliet in [2]. This paper however concentrates on differentiating on completeness notions defined by standard many-one, bounded truth-table and Turing reductions in both the polynomial time and logarithmic space case on nondeterministic time and space classes. A comparison of unbounded polynomial time and logarithmic space bounded reductions is given in [3] which involves an interesting conflict between the different interpretations of resource bounded truth table reducibilities. If defined as a bounded branching program, bounded truth table reducibilities ar as powerful as bounded Turing reductions (As can be found in [7]). If defined as bounded boolean formulae then logspace bounded truth-table reductions are identical to logspace bounded Turing reductions only if $NC_1=LOGSPACE$.

In the present paper we concentrate on the remaining open problems between notions of bounded reducibilities, and the corresponding completeness notions on $E$, $NE$, $EXP$ and $NEXP$ (and solve all of these).

- In section 3, we prove that $k$-conjunctive and $k$-disjunctive truth-table completeness are incomparable.
- In section 4, we show that many-one completeness is the same as 1-truth table completeness.
- In section 5, we give a precise relation between $k$-Turing and $m$-truth-table completeness: for $k > 1$: $k$-Turing completeness strictly contains $k$-truth-table completeness, and for $k < m < 2^k - 1$, $k$-Turing completeness and $m$-truth-table completeness are incomparable.

As all of the considered reductions are bounded by a constant number of queries, the proofs are independent of the specific model for truth-table reducibilities.

2 Preliminaries

2.1 Machines and languages

Let $\Sigma = \{0, 1\}$. Strings are elements of $\Sigma^*$, and are denoted by small letters $x, y, u, v, \ldots$. For any string $x$ the length of a string is denoted by $|x|$. Languages are subsets of $\Sigma^*$, and
are denoted by capital letters $A, B, C, S, \ldots$. For any set $S$ the cardinality of $S$ is denoted by $|S|$. We fix a pairing function $\lambda x, y. < x, y >$ computable in polynomial time from $\Sigma^* \times \Sigma^*$ to $\Sigma^*$. 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.

Oracle machines appear in the paper in two flavors: adaptive and non-adaptive. For a non-adaptive machine queries may not be interdependent, whereas an adaptive machine may compute a next query depending on the answer to previous queries.

Whenever it is obvious that a universal recognizing or transducing machine exists for a class of languages (i.e. the class is recursively presentable), we will assume an enumeration of the acceptors and/or transducers and denote this enumeration by $M_1, M_2, \ldots$. For a Turing machine $M$, $L(M)$ denotes the set of strings accepted by $M$.

### 2.2 Time classes

Let $\text{DTIME}(2^{cn})$ be the class of sets such that $A \in \text{DTIME}(2^{cn})$ iff there exists a Turing machine $M$ whose running time is bounded by $2^{cn}$ for $n \to \infty$ ($n$ is the length of the input) and $A = L(M)$. Let $\text{NTIME}(2^{cn})$ be the corresponding nondeterministic class. We define the following classes:

\[
\begin{align*}
\text{NEXP} &= \bigcup_{i=1}^{\infty} \text{NTIME}(2^{cn}) \\
\text{EXP} &= \bigcup_{i=1}^{\infty} \text{DTIME}(2^{cn}) \\
\text{NE} &= \bigcup_{c=1}^{\infty} \text{NTIME}(2^{cn}) \\
\text{E} &= \bigcup_{c=1}^{\infty} \text{DTIME}(2^{cn})
\end{align*}
\]

### 2.3 Truth tables

The ordered pair $< a_1, \ldots, a_k >, \alpha > (k > 0)$ is called a truth-table condition of norm $k$ if $< a_1, \ldots, a_k >$ is a $k$-tuple of strings, and $\alpha$ is a $k$-ary Boolean function [8]. The set $\{a_1, \ldots, a_k\}$ is called the associated set of the tt-condition. A function $f$ is a truth-table function if $f$ is total and $f(x)$ is a truth-table condition for every $x$ in $\Sigma^*$. If, for all $x$, $f(x)$ has norm less than or equal to $k$, then $f$ is called a $k$-truth-table ($k-\text{tt}$) function. We say that a tt-function $f$ is a disjunctive (conjunctive) truth-table (dtt (ctt)) function if $f$ is a truth-table condition whose Boolean function is always disjunctive (conjunctive).

### 2.4 Reductions, reducibilities and completeness

Let $A_1, A_2 \subseteq \Sigma^*$. We say that:
1. $A_1$ is polynomial-time many-one reducible to $A_2$ ($\leq_p^m$-reducible) iff there exists a function $f$ computable within polynomial-time such that $x \in A_1$ iff $f(x) \in A_2$.

2. $A_1$ is polynomial-time $k$-truth-table reducible to $A_2$ ($\leq_p^{k,tt}$-reducible) iff there exists a polynomial-time bounded $ktt$-function $f$ such that $\alpha(\chi_{A_2}(a_1), \ldots, \chi_{A_2}(a_k)) = \text{true}$ iff $x \in A_1$, where $f(x)$ is $\langle a_1, \ldots, a_k \rangle$, $\alpha$ and $\chi_{A_2}$ is the characteristic function of the set $A_2$.

3. $A_1$ is polynomial-time Turing reducible to $A_2$ ($\leq_p^T$-reducible) to $A_2$ if there exists a polynomial-time bounded deterministic oracle machine such that $A_1 = L(M, A_2)$.

4. $A_1$ is polynomial-time disjunctive (conjunctive) reducible ($\leq_p^d$ ($\leq_p^c$) -reducible) to $A_2$, if $A_1 \leq_p^d A_2$ by some dtt(cctt)-function. For $k \geq 0$, $A_1$ is $k$-disjunctive (conjunctive) reducible ($\leq_p^{k,d}$ ($\leq_p^{k,c}$)) to $A_2$, if $A_1 \leq_p^{k,d} A_2$ by some dtt(cctt)-function of norm $k$.

Let $\leq_p^c$ be any of the above reductions

1. A set $A$ is $\leq_p^c$ hard for some complexity class $C$ iff for all $B \in C$, $B$ is $\leq_p^c$ reducible to $A$.

2. A set $A$ is $\leq_p^c$ complete for some complexity class $C$ iff $A$ is $\leq_p^c$ hard for $C$ and $A \in C$.

For $\text{NEXP}$ we use a standard many-one complete set $K$. $K = \{<i, x, l>| \text{ machine } i \text{ has an accepting computation on input } x \text{ within } \leq l \text{ steps}\}$. Note that this set can be recognized in $2^n$ steps and is also complete for $\text{NE}$. For $\text{EXP}$ we use $K = \{<i, x, l>| \text{ machine } i \text{ accepts } x \text{ within } l \text{ steps}\}$.

3 Disjunctive versus Conjunctive Truth-table Reductions

**Theorem 1** there exists a set $A \in \text{NEXP}$ such that $A$ is $\leq_p^{2-d}$-complete but not $\leq_p^{2-c}$-complete.

**Proof**: Let $K$ be the standard $\leq_m^p$-complete set for $\text{NE}$ as defined above. To achieve the separation we construct a set $W \in E$ and a set $A \in \text{NEXP}$ such that $W \not\leq_{2-c}^p A$ but $K \leq_{2-d}^p A$. We assume an enumeration of polynomial time 2-conjunctive truth-table reductions $M_1, M_2, \ldots$ where $M_i$ runs in time $n^i$. We need a set of elements on which to diagonalize. To do this we define a sequence of integers $\{b(n)\}_n$:

$$b(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ \frac{1}{2^{b(n-1)^{n-1} + 1}} & \text{otherwise} \end{cases}$$

We construct $A$ and $W$ in stages; $A = \bigcup_{n=0}^{\infty} A_n$

In stage 0 $A_0 = W = \emptyset$.

**stage $n$:**

Let $A'_n = \{<i, z>| z \in K \text{ and } b(n-1)^{n-1} < |<i, z>| \leq b(n)^n \text{ and } i \in \{0,1\}\}$

Simulate $M_n$ on input $0^{b(n)}$. $M_n$ queries two strings $x$ and $y$, w.l.o.g. let $x$ be the largest
(in lexicographic order) of the two. $M_n$ accepts iff $x$ and $y$ are both in the oracle set. There are two cases:

1. $|x| \leq b(n - 1)^{n-1}$
2. $b(n - 1)^{n-1} < |x| \leq b(n)^n$

In case 1, compute the answers relative to $A_{<n}$ of both $x$ and $y$ and put $0^{b(n)} \in W$ iff $M_n$ rejects. Let $A_n = A'_n$.

In case 2, put $0^{b(n)} \in W$ and let $A_n = A'_n \setminus \{x\}$. This ensures that $M_n^A$ rejects on input $0^{b(n)}$.

end of stage $n$

We now show that $A \in \text{NEXP}$. To decide $<i, z> \in A$ $(i = 0, 1)$ compute $n$ such that $b(n)^n \geq |<i, z>| > b(n - 1)^{n-1}$. Simulate machine $M_n$ on input $0^{b(n)}$ and compute $x$ and $y$. If $<i, z> = x$ reject, else accept iff $z \in K$. All this can be done in nondeterministic exponential time, since simulation of machine $M_n$ on input $0^{b(n)}$ takes time $b(n)^n \leq 2^{2^{b(n)}} \leq 2^{2^{b(n-1)^{n-1}+1}} \leq 2^{2^{b(n)}} = 2^{|<i, z>|^2}$

Next we show that $W \in E$. On input $0^{b(n)}$ simulate $M_n$ on input $0^{b(n)}$ and compute $x$ and $y$. If $|x| > b(n - 1)^{n-1}$ we accept, else we must decide membership of $x$ and $y$ to $A$. To compute if $x \in A$, determine $n' < n$ such that $b(n' - 1)^{n'-1} < |x| \leq b(n')^{n'}$. $x \in A$ if $x$ is not the largest query asked by $M_{n'}$ and $x \in K$. This takes deterministic time $2^{2^{b(n)}} < 2^{b(n)}$.

Now assume for a contradiction that $A$ is $\leq_{2-c}^P$-complete. Note that $0^{b(n)} \in W$ iff $M_n$ rejects. Then there must be a 2-conjunctive truth-table reduction from $W$ to $A$. Let $M_f$ be the machine witnessing this reduction. But $0^{b(j)}$ is in $W$ iff $M_f$ on input $0^{b(j)}$ rejects. This contradicts the fact that $M_f$ reduces $W$ to $A$. This proves that $A$ is not $\leq_{2-c}^P$-complete.

Finally we give the $\leq_{2-d}^P$ reduction from $K$ to $A$. Since in every step only one of the pairs $<1, x>$ or $<0, x>$ can be deleted, $x \in K$ iff $<0, x> \in A$ or $<1, x> \in A$. Therefore, the following reduction reduces $K$ to $A$:

$$g(x) = \{<0, x> \lor <1, x>\}$$

$\boxtimes$

Almost the same proof technique yields the following theorem.

**Theorem 2** there exists a set $A \in \text{NEXP}$ such that $A$ is $\leq_{2-c}^P$-complete but not $\leq_{2-d}^P$-complete.

**Proof:** The proof is almost the same as the previous one. It differs in case 2 in the diagonalization. Here we put $0^{b(n)}$ not in $W$ and add $x$ to $A'_n$. In this way we ensure that $0^{b(n)} \not\in W$ iff $M_n^A$ accepts. Note that $x \in K$ iff $<0, x> \in A$ and $<1, x> \in A$. The $\leq_{2-c}^P$-reduction from $K$ to $A$ becomes:

$$g(x) = \{<0, x> \land <1, x>\}$$

$\boxtimes$

It is easy to see that the proofs generalize to $\leq_{k-d}^P$-complete sets v.s. $\leq_{k-c}^P$-complete sets (for $k \geq 2$). The theorems solve an open problem from Watanabe [10].
Corollary 3 For all $k \geq 2$ there exists a set $A$ that is $\leq_{k-tt}^p$ -complete for NE but not $\leq_{k-d}^p$ $(\leq_{k-c}^p)$ -complete for NE.

This corollary can be strengthened. We are now able to construct a set that is $\leq_{k-tt}^p$ -complete but neither $\leq_{k-d}^p$ -complete nor $\leq_{k-c}^p$ -complete.

Corollary 4 For all $k \geq 2$ there exists a set $A$ that is $\leq_{k-tt}^p$ -complete for NE but neither $\leq_{k-d}^p$ -complete nor $\leq_{k-c}^p$ -complete for NE.

Proof: To do this we use the constructions of theorem 1 at the even stages and the constructions of theorem 2 at the odd stages $\Box$

Corollary 5 For all $k \geq 2$ there exists a set $A$ that is $\leq_{k-tt}^p$ -complete for NE but neither $\leq_{k-d}^p$ -complete nor $\leq_{k-c}^p$ -complete.

Clearly all the results in this section go through for $E$, $EXP$, and $NEXP$.

4  1-Truth-Table versus Many-One

Another question concerning reductions on $E$ and $NE$ is the following: do the notions of $\leq_{1-tt}$ and $\leq_m$ differ for complete sets. From recursion theory it is known (and easy to prove) that these two reductions are the same with respect to RE sets. Recently Homer et. al. [5] showed that these two notions are also the same for $E$ -complete sets. They left open however the question for $NE$. We solve this question here. The idea is to first prove for sets $\in NE \cap co-NE$ that are $\leq_{1-tt}^p$-reducible to a complete set are also $\leq_m^p$-reducible to this set. This can be done using a similar technique as in [5]. Once this is done we are able to reduce the general case (when this is necessary) to this special case.

Lemma 6 Let $T$ be a $\leq_{1-tt}^p$-complete set for NE. For every set $A \in NE \cap co-NE$, $A \leq_m^p T$.

Proof: We assume a standard enumeration of polynomial time 1-truth-table reductions $M_1, M_2, \ldots$ where $M_i$ runs in time $n^i$. Let $A$ be any set in $NE \cap co-NE$. Now we are going to construct a set $D \in NE$. We simulate $M_i$ on input $<i, x>$ and let $z$ be the string queried by $M_i$. Now there are 4 possible cases that can occur:

1. $M_i$ accepts iff $z$ is in the oracle set.
2. $M_i$ accepts iff $z$ is not in the oracle set.
3. $M_i$ accepts. ($M_i$ is not a 1tt reduction)
4. $M_i$ rejects. ($M_i$ is not a 1tt reduction)

In case 1 we put the pair $<i, z>$ in $D$ iff $z \in A$

In case 2 we put $<i, x>$ in $D$ iff $x \notin A$

In case 3 we put $<i, x>$ not in $D$

In case 4 we put $<i, x>$ in $D$

$D$ is in $NE$. To compute if $<i, x>$ is in $D$, simulate machine $M_i$ on input $<i, x>$ and find out in which case $M_i$ ends up. The only problem is case 2 but since $A$ is in $NE \cap co-NE$
we can compute if \( x \) is in the complement of \( A \). Since \( D \) is in \( NE \), \( D \) is 1-truth-table reducible to \( T \). Let machine \( M_h \) witness this reduction and let \( z \) be the string queried by machine \( M_h \) on input \( <h, x> \). Now we can construct the many one reduction \( f \) from \( A \) to \( T \):

\[
f(x) = z
\]

Since machine \( M_h \) runs in polynomial time this reduction also runs in polynomial time. Machine \( M_h \) can not end up in case 3 or 4, since this would contradict the fact the \( M_h \) is a 1-truth-table reduction from \( D \) to \( T \). The following two cases remain possible:

- Machine \( M_h \) is in case 1: \( x \in A \) iff \( <h, x> \in D \) iff \( M_h \) accepts iff \( z \in T \).
- Machine \( M_h \) is in case 2: \( x \in A \) iff \( <h, x> \notin D \) iff \( M_h \) rejects iff \( z \in T \).

So in both cases \( x \in A \) iff \( z \in T \). \( \square \)

Now for all sets in \( NE \) if a set is 1-truth-table reducible to a complete set \( T \) via say machine \( M_j \) there are strings that are accepted if the query is in \( T \). Those strings are already many-one reducible to \( T \). The other strings (i.e. the strings that get accepted by a query in the complement of \( T \)) form a set that is in \( NE \cap co-NE \) and by lemma 6 they are many-one reducible to \( T \) via some other reduction. More formally:

**Theorem 7** Every \( \leq_1^{p} \)-complete set for \( NE \) is also \( \leq^p_m \)-complete.

**Proof:** Let \( A \) be a set in \( NE \), \( T \) a 1-truth-table complete set in \( NE \) and let \( M_j \) witness the reduction from \( A \) to \( T \). On any input \( M_j \) can end up in one of the following four situations:

1. \( M_j \) queries \( z \) and accepts iff \( z \in T \)
2. \( M_j \) queries \( z \) and accepts iff \( z \notin T \)
3. \( M_j \) accepts
4. \( M_j \) rejects

We now split set \( A \) in two subsets \( A_1 \) and \( A_2 \).

\[
A_1 = \{ x \mid x \in A \text{ and } M_j \text{ is not in case 2} \} \\
A_2 = \{ x \mid x \in A \text{ and } M_j \text{ is in case 2} \}
\]

**CLAIM 8** \( A_2 \) is in \( NE \cap co-NE \).

**Proof:** We need to show that there is a \( NE \) predicate for \( A_2 \) and for the complement of \( A_2 \).

\[
x \in A_2 \text{ iff } \text{machine } M_j \text{ in case 2 and } x \in A \]  
\[
x \notin A_2 \text{ iff } \text{machine } M_j \text{ not in case 2 or } z \in T
\]
It is clear that both predicates are \( NE \).

Now we can construct the many-one reduction from \( A \) to \( T \): On input \( x \) simulate machine \( M_j \) on input \( x \). If \( M_j \) is in case 1 then output \( x \). If \( M_j \) in case 2 then \( x \) is in \( A \) iff \( x \) is in \( A_2 \). Since \( A_2 \) is in \( NE \cap co-NE \) there is by lemma 6 a many-one reduction from \( A_2 \) to \( T \) say \( g \). Now output \( g(x) \). If \( M_j \) is in case 3 output a fixed element \( t_0 \in T \) and if \( M_j \) is in case 4 output a fixed element \( t_1 \notin T \). The entire construction can be carried out in polynomial time.

The construction can be generalized to a recursion theoretic setting. We relax the time bounds and end up with recursive reductions. We now have the following equivalent reductions \( \leq^r_m \) for a many-one reduction and \( \leq^r_{1-\text{tt}} \) for a 1-truth-table reduction in exactly the same way as the above theorem was proven we can prove the following:

**Corollary 9** let \( \Sigma_k \) be the \( k \)-th level of the arithmetic hierarchy as defined in [9]. For all \( k \) if \( A \) is \( \leq^r_{1-\text{tt}} \) complete for \( \Sigma_k \) then \( A \) is \( \leq^r_m \) complete for \( \Sigma_k \).

It would be interesting to prove the same result for the class \( NP \). The problem is that the technique used in lemma 6 is not applicable for sets in \( NP \). Under the strong assumption that \( P = NP \cap co-NP \) however, we can prove it.

**Corollary 10** If \( P = NP \cap co-NP \) then every \( \leq^r_{1-\text{tt}} \) complete set for \( NP \) is \( \leq^r_m \) complete.

5 Bounded Turing versus bounded Truth-Table

We now turn our attention to bounded Turing reductions. Informally, these are Turing reductions where for any input \( x \), the number of queries asked is bounded by a constant \( k \). Note that by definition, every \( k \)-truth table reduction is a \( k \)-Turing reduction. It is well known that every \( k \)-Turing reduction can be simulated by \((2^k - 1)\)-truth-table reduction. A natural question one can ask is: “What is the relation between \( k \)-Turing reductions versus \( m \)-truth-table reductions?” In the previous section, it was proven that for nondeterministic-exponential-time complete sets: many-one = 1-truth-table = 1-Turing. In this section we prove that \( k \)-Turing reductions are more powerful than \( k \)-truth-table reductions for \( k > 1 \), and that for \( k < m < 2^k - 1 \), \( k \)-Turing and \( m \)-truth table reductions are incomparable. These results hold even for the corresponding completeness notions on \( NEXP \).

**Definition 11** Let \( Q(M, x, A) \) be the set of strings, queried in the computation of polynomial time oracle machine \( M \) with oracle \( A \) on input \( x \). We say that \( B \leq^p_{k-T} A \) if there exists a polynomial time oracle machine \( M \) such that \( B = L(M, A) \) and for all \( x \), \( |Q(M, x, A)| \leq k \).

**Theorem 12** For every \( k \) there exists a set \( D \) in \( NEXP \) that is \( \leq^p_{k-T} \) complete but not \( \leq^p_{(2^{2k-2})-\text{tt}} \) complete.

As an example of the techniques used, we first prove the degenerate case \( k = 2 \), i.e. we will construct a set \( D \in NEXP \) such that \( D \) is \( \leq^p_{2-T} \) complete but not \( \leq^p_{2-\text{tt}} \) complete.
**Proof:** Let $M_1, M_2, \ldots$, be an enumeration of the 2-truth-table reductions, where $M_i$ runs in time $n^i$. Let $K$ be the standard $\leq_{\text{P}}^n$-complete set for NE and let $\{b(n)\}_n$ the sequence defined in the proof of theorem 1. We will construct sets $D$ and $W \in \text{NEXP}$ such that $W \not\leq_{\text{P}}^2 D$, and $K \leq_{\text{P}}^2 D$. $W$ and $D$ will be constructed in stages, $D = \bigcup_{n=0}^{\infty} D_n$.

To ensure that $K \leq_{\text{P}}^2 D$, we have to exploit the fact that a 2-Turing reduction can ask 3 queries in its entire oracle tree, while a 2-truth-table reduction can ask at most 2 queries in its entire oracle tree. We will ensure that $D \subseteq \{0,1,2\} \times K$, and use the following 2-Turing reduction $M_T$ to reduce $K$ to $D$:

On input $x$, first query $<0, x>$. If the answer is YES, query $<1, x>$, and accept iff the answer is YES. If the answer to query $<0, x>$ is NO, query $<2, x>$ and accept iff the answer is YES.

For every 2-truth-table reduction, and for every $x$, there exists a copy of $x$ that is not queried. This provides enough freedom to diagonalize against the 2-truth-table reductions, while still keeping $K \leq_{\text{P}}^2 D$ by $M_T$.

In stage 0 $D_0 = W = \emptyset$

**stage $n$:**

Let $D_n = \{<i, x>| x \in K \text{ and } b(n - 1)^{n-1} < |<i, x>| \leq b(n)^n \text{ and } 0 \leq i \leq 2\}$.

Simulate $M_n$ on input $0^{b(n)}$. If $M_n$ queries strings of length $\leq b(n - 1)^{(n-1)}$ compute the answers to those strings.

Let $Q$ be the set of queries $\{0,1,2\} \times \Sigma^*$ with length $> b(n - 1)^{(n-1)}$. Let $i_0 \in \{0,1,2\}$ be a number such that $Q$ contains no string of the form $<i_0, x>$. Now we take the following action, depending on the value of $i_0$:

- $i_0 = 0$: For every $y$ occurring as second member in a pair of $Q$ do $D_n := (D_n \setminus \{<2, y>\}) \cup \{<1, y>\}$
- $i_0 = 1$: For every $y$ occurring as second member in a pair of $Q$ do $D_n := (D_n \setminus \{<2, y>\}) \cup \{<0, y>\}$
- $i_0 = 2$: For every $y$ occurring as second member in a pair of $Q$ do $D_n := (D_n \setminus \{<0, y>\}) \cup \{<1, y>\}$

Now we are able to compute if $M_n$ accepts or rejects. Put $0^{b(n)}$ in $W$ iff $M_n$ rejects on input $0^{b(n)}$.

**end of stage $n$**

We can use a similar argument as in the proof of theorem 1, to prove that $D \in \text{NEXP}$, $W \in E$ and $W$ is not $\leq_{\text{P}}^2 D$.

Our 2-Turing reduction $M_T$ accepts $x$ iff either $\{<0, x>, <1, x>\} \subseteq D$ or $<2, x> \in D$ and $<0, x> \notin D$.

We have the following possibilities for $D \cap \{<0, x>, <1, x>, <2, x>\}$

- $x \in K$: $\{<0, x>, <1, x>, <2, x>\}$ or $\{<0, x>, <1, x>\}$ or $\{<1, x>, <2, x>\}$.
- $x \notin K$: $\emptyset$ or $\{<0, x>\}$ or $\{<1, x>\}$.

Thus, $M_T$ accepts $x$ iff $x \in K$ as required. $\Box$
For this proof, it was essential that a 2-Turing reduction can ask more queries in its entire oracle tree than can a 2-truth-table reduction. Since a $k$-Turing reduction can ask $2^k - 1$ queries in its entire oracle tree, while a $2^k - 2$ truth-table reduction can ask at most $2^k - 2$ queries in its entire oracle tree, we can use a generalization of the previous construction to obtain a set $D$ that is $\leq^p_{k-1}$-complete, but not $\leq^p_{(2^k-2)-tt}$-complete, thus proving theorem 12.

**Proof:** Let $M_1, M_2, \ldots$, be an enumeration of the $2^k - 2$-truth-table reductions, where $M_i$ runs in time $n^i$. Let $K$ be a standard $\leq^p_m$-complete set for NE and let $\{b(n)\}_n$ the sequence defined in the proof of theorem 1. We construct set $D$ and $W$ in stages; $D = \bigcup_{n=0}^{\infty} D_n$. We will ensure that $D \subseteq \{0, \ldots, 2^k - 2\}$, and use the following $k$-Turing reduction $M_T$ to reduce $K$ to $D$.

On input $x$, first query $\langle 0, x \rangle$. For each query $\langle i, x \rangle$ at depth $< k$ do the following: if the answer is YES, query $\langle 2i + 1, x \rangle$, else query $\langle 2i + 2, x \rangle$. Accept iff the last query asked gets answer YES.

In stage 0 $D_0 = W = \emptyset$

**stage $n$:**

Let $D_n = \{\langle i, x \rangle | x \in K$ and $b(n - 1)^{n-1} < |\langle i, x \rangle| \leq b(n)^n$ and $0 \leq i \leq 2^k - 2\}$. Simulate $M_n$ on input $0^{b(n)}$. If $M_n$ queries strings of length $\leq b(n - 1)^{(n-1)}$ compute the answers to those strings.

Let $Q$ be the set of queries $\in \{0, \ldots, 2^k - 2\} \times \Sigma^*$ with length $> b(n - 1)^{(n-1)}$. Let $i_0 \in \{0, \ldots, 2^k - 2\}$ be such that $Q$ contains no string of the form $\langle i_0, x \rangle$.

Consider the following tree of depth $k$, where the nodes are labeled $0, \ldots, 2^k - 2$: the root has label 0, and for each node at depth $< k$ with label $i$, the left child has label $2i + 1$, and the right child label $2i + 2$.

For every $y$ that occurs as second member in a pair of $Q$ and and for every $i \in \{0, \ldots, 2^k - 2\}, i \neq i_0$, we take the following action:

1. if $i$ occurs on the path from the root to $i_0$ then if $i_0$ is in the left subtree of $i$ then $D_n := D_n \cup \{\langle i, y \rangle\}$
   if $i_0$ is in the right subtree of $i$ then $D_n := D_n \setminus \{\langle i, y \rangle\}$

2. if $i$ occurs to the left of the path from the root to $i_0$ then $D_n := D_n \cup \{\langle i, y \rangle\}$

3. if $i$ occurs to the right of the path from the root to $i_0$ then $D_n := D_n \setminus \{\langle i, y \rangle\}$

4. if $i$ is in the left subtree of $i_0$ then $D_n := D_n \cup \{\langle i, y \rangle\}$

5. if $i$ is in the right subtree of $i_0$ then $D_n := D_n \setminus \{\langle i, y \rangle\}$

Now we are able to compute if $M_n$ accepts or rejects. Put $0^{b(n)}$ in $W$ iff $M_n$ rejects on input $0^{b(n)}$.

**end of stage $n$**

We can use a similar argument as in the proof of theorem 1, to prove that $D \in \text{NEXP}$, $W \in E$ and $W$ is not $\leq^p_{(2^k-2)-tt} D$. 9
Recall that our \( k \)-Turing reduction \( M_T \) works as follows: on input \( x \), first query \(<0, x>\). For each query \(<i, x>\) at depth \(<k\) do the following: if the answer is YES, query \(<2i + 1, x>\), else query \(<2i + 2, x>\). Accept iff the last query asked gets answer YES. View this reduction as a tree of depth \( k \), where the nodes are labelled by the queries, and a YES (resp. NO) answer to a query corresponds to taking the left (resp. right) branch.

If \( x \in K \), \( M_T \) on input \( x \) takes either the leftmost path in its oracle tree, or the leftmost path through \(<i_0, x>\). In either case we accept.

If \( x \notin K \), \( M_T \) on input \( x \) takes either the rightmost path in its oracle tree, or the rightmost path through \(<i_0, x>\). In either case we reject.

Thus, \( M_T \) is a reduction from \( K \) to \( D \). ⊥

Now we will construct a set \( D \) in \( NEXP \) that is \( \leq_{(k+1)-tt}^p \)-complete but not \( \leq_{k-T}^p \)-complete. A \( \leq_{k-T}^p \) reduction can be represented as a binary tree of depth \( k \). Where every node in the tree represents a query and if the answer to the query is yes we proceed to the left branch otherwise to the right branch. The idea is to force the \( \leq_{k-T}^p \) reduction into one branch by leaving out all the queries (if possible) of that branch. Since there are only \( k \) queries on one branch there remains the freedom to code an extra pair of \( K \) into \( D \) that can be queried by a \( \leq_{(k+1)-tt}^p \) reduction.

**Theorem 13** There exists a set \( D \) in \( NEXP \) that is \( \leq_{(k+1)-tt}^p \)-complete but not \( \leq_{k-T}^p \)-complete.

**Proof:** We only give the proof for \( k = 4 \). Let \( K \) be the standard \( \leq_{n}^p \)-complete set for \( NE \) and \( \{b(n)\}_n \) the sequence defined in the proof of theorem 1. Again we use a stage construction.

**Stage \( n \):**

\[ \begin{align*}
D'_n &:= \{<i, x>| \ x \in K \text{ and } b(n - 1)^{(n-1)} \leq |x| \leq b(n)^n \text{ and } 0 \leq i \leq 5\} \\
\text{Simulate } & M_n \text{ on input } 0^{b(n)}, \text{ compute the answers to the queries that are small i.e.} \\
&\text{< } b(n - 1)^{(n-1)}. \text{ Now evaluate the branch where all the other queries receive the answer NO}. \text{ Let } Q' \text{ be the set of the queries that are big } (\geq b(n - 1)^{(n-1)}). \\
\text{Put } & 0^{b(n)} \text{ in } W \text{ iff } M_n \text{ rejects} \\
D_n &:= D'_n \setminus Q'
\end{align*} \]

**End of stage \( n \)**

Note that for every \( x : x \in K \) iff \(<i, x> \in D \) for some \( i \). The \( 5 \)-truth-table reduction from \( K \) to \( D \) becomes:

\[ g(x) = \{<0, x> \lor \ldots <4, x>\} \]

⊥

**Corollary 14** If \( k < m < 2^k - 1 \), then \( \leq_{k-T}^p \) and \( \leq_{m-tt}^p \) are incomparable with respect to complete sets for \( NEXP \).

As before the results also go through for \( NE, E \) and \( EXP \).
6 Conclusions

In the previous sections we proved that almost reductions on $NE$, $E$, $EXP$ and $NEXP$ are incomparable except those where inclusion is trivial. As a consequence the extended Berman Hartmanis conjecture for those reductions fails. It follows that for example the degree of 2-truth-table complete sets are not p-isomorphic. An interesting step would be to disprove the extended conjecture for the degree of many-one complete sets. Perhaps the techniques discussed here could lead towards results in that direction.

The proof of the non-separation of many-one and 1-truth-table reductions fails for $NP$. The problem is that it is not known if the universal polynomial time function is computable in $NP$. For all well behaved classes that contain the universal polynomial time function, this non-separation result is true.

One area of great interest would be to separate the various polynomial time reductions on classes between $P$ and $PSPACE$, and in particular to do this for $NP$.

All the previous obtained results go through with respect to logspace reductions for nondeterministic and deterministic space classes that contain that universal logspace function. Interesting would be to prove similar result for $NLOGSPACE$.

Acknowledgements We would like to thank Steven Homer and Peter van Emde Boas for fruitful discussions.
Bibliography


The ITLI Prepublication Series

1990

Logic, Semantics and Philosophy of Language
LP-90-01 Jaap van der Does
LP-90-02 Jeroen Groenendijk, Martin Stokhof
LP-90-03 Renate Bartsch
LP-90-04 Aarne Ranta
LP-90-05 Patrick Blackburn
LP-90-06 Gemmaro Cianciola
LP-90-07 Gemmaro Cianciola
LP-90-08 Herman Hendriks
LP-90-09 Paul Dekker
LP-90-10 Theo M.V. Janssen
LP-90-11 Johan van Benthem
LP-90-12 Serge Lapierre
LP-90-13 Zhisheng Huang
LP-90-14 Jeroen Groenendijk, Martin Stokhof
LP-90-15 Maarten de Rijke
LP-90-16 Zhisheng Huang, Karen Kwast

Mathematical Logic and Foundations
ML-90-01 Harold Schellinx
ML-90-02 Jaap van Oosten
ML-90-03 Yde Venema
ML-90-04 Maarten de Rijke
ML-90-05 Domenico Zambella
ML-90-06 Jaap van Oosten
ML-90-07 Maarten de Rijke
ML-90-08 Harold Schellinx
ML-90-09 Dick de Jongh, Duccio Pianigiani
ML-90-10 Michiel van Lambalgen
ML-90-11 Paul C. Gilmore

Computation and Complexity Theory
CT-90-01 John Tromp, Peter van Emde Boas
CT-90-02 Sieger van Denneheuvel
CT-90-03 Gerard R. Renardel de Lavalette
CT-90-04 Harry Buhrman, Edith Spaan
CT-90-05 Sieger van Denneheuvel, Karen Kwast
CT-90-06 Michel Smid, Peter van Emde Boas
CT-90-07 Kees Doets

Other Prepublications
X-90-01 A.S. Troelstra
X-90-02 Maarten de Rijke
X-90-03 L.D. Beklemishev
X-90-04
X-90-05 Valent Shchepet
X-90-06 Lenat, Valentin Goranko, Solomon Passy
X-90-07 V.Yu. Shavrukov
X-90-08 L.D. Beklemishev
X-90-09 V.Yu. Shavrukov
X-90-10 Sieger van Denneheuvel
X-90-11 Alessandra Carbone
X-90-12 Maarten de Rijke
X-90-13 K.N. Ignatiev
X-90-14 L.A. Chagrov
X-90-15 A.S. Troelstra

A Generalized Quantifier Logic for Naked Infinitives
Dynamic Montague Grammar
Concept Formation and Concept Composition
Intuitionistic Categorial Grammar
Nominal Tense Logic
The Variability of Impersonal Subjects
Anaphora and Dynamic Logic
Flexible Montague Grammar
The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
Models for Discourse Markers
General Dynamics
A Functional Partial Semantics for Intensional Logic
Logics for Belief Dependence
Two Theories of Dynamic Semantics
The Modal Logic of Inequality
Awareness, Negation and Logical Omniscience
Isomorphisms and Non-Isomorphisms of Graph Models
A Semantical Proof of De Jongh's Theorem
Relational Games
Unary Interpretability Logic
Sequences with Simple Initial Segments
Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
A Note on the Interpretability Logic of Finitely Axiomatized Theories
Some Syntactical Observations on Linear Logic
Solution of a Problem of David Guaspari
Randomness in Set Theory
The Consistency of an Extended NaSet
Associative Storage Modification Machines
A Normal Form for PCSI Expressions
Generalized Kolmogorov Complexity
in Relativized Separations
Bounded Reductions
Efficient Normalization of Database and Constraint Expressions
Dynamic Data Structures on Multiple Storage Media, a Tutorial
Greatest Fixed Points of Logic Programs
Remarks on Intuitionism and the Philosophy of Mathematics,
Revised Version
Some Chapters on Interpretability Logic
On the Complexity of Arithmetical Interpretations of Modal Formulae
Annual Report 1989
Using the Universal Modality: Gains and Questions
The Lindenbaum Fixed Point Algebra is Undecidable
Provability Logics for Natural Turing Progressions of Arithmetical
Theories
On Rosser's Provability Predicate
An Overview of the Rule Language RL/1
Provable Fixed points in \( \mathcal{I}\o+\mathcal{O} \), revised version
Bi-Unary Interpretability Logic
Dzhaparidze's Polymodal Logic: Arithmetical Completeness,
Fixed Point Property, Craig's Property
Undecidable Problems in Correspondence Theory
Lectures on Linear Logic

X-90-01 A.S. Troelstra