Institute for Language, Logic and Information

BOUNDED REDUCTIONS

Harry Buhrman
Edith Spaan
Leen Torenvliet

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

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 LP-88-01 Michiel van Lambalgen *Logic, Semantics and Philosophy of Language:* 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
ML-88-01 Jaap van Oosten *Mathematical Logic and Foundations:* 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
CT-88-01 Ming Li, Paul M.B.Vitanyi *Computation and Complexity Theory:* 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
X-88-01 Marc Jumelet *Other prepublications:* On Solovay's Completeness Theorem

1989 LP-89-01 Johan van Benthem *Logic, Semantics and Philosophy of Language:* 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
ML-89-01 Dick de Jongh, Albert Visser *Mathematical Logic and Foundations:* 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 $I\Delta_0+\Omega_1$
CT-89-01 Michiel H.M. Smid *Computation and Complexity Theory:* 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
X-89-01 Marianne Kalsbeek *Other Prepublications:* 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

BOUNDED 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 $(= \cup_{c \in I\!N} \mathrm{DTIME}\,(2^{cn}))$. 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 Σ^*, 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 Σ^*, 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 $\lambda xy.<x, y>$ computable in polynomial time from $\Sigma^* \times \Sigma^*$ to Σ^*. 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 DTIME(2^{cn}) be the class of sets such that $A \in$ 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 NTIME(2^{cn}) be the corresponding nondeterministic class. We define the following classes:

$$\begin{aligned} NEXP &= \bigcup_{i=1}^{\infty} \mathrm{NTIME}\left(2^{n^i}\right) \\ EXP &= \bigcup_{i=1}^{\infty} \mathrm{DTIME}\left(2^{n^i}\right) \\ NE &= \bigcup_{c=1}^{\infty} \mathrm{NTIME}\left(2^{cn}\right) \\ E &= \bigcup_{c=1}^{\infty} \mathrm{DTIME}\left(2^{cn}\right) \end{aligned}$$

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 α 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 Σ^*. If, for all x, $f(x)$ has norm less than or equal to k, then f is called a k-truth-table $(k - tt)$ 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 ($\leq_m^p$-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 ($\leq_{k-tt}^p$-reducible) iff there exists a polynomial-time bounded ktt-function f such that $\alpha(\chi_{A_2}(a_1), \ldots, \chi_{A_2}(a_k)) = \mathbf{true}$ iff $x \in A_1$, where $f(x)$ is $<<a_1, \ldots, a_k>, \alpha>$ and χ_{A_2} is the characteristic function of the set A_2.

3. A_1 is polynomial-time Turing reducible to A_2 ($\leq_T^p$-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 ($\leq_d^p$ ($\leq_c^p$) -reducible) to A_2, if $A_1 \leq_{tt}^p A_2$ by some dtt(ctt)-function. For $k \geq 0$, A_1 is k-disjunctive(conjunctive) reducible ($\leq_{k-d}^p(\leq_{k-c}^p)$) to A_2, if $A_1 \leq_{tt}^p A_2$ by some dtt(ctt)-function of norm k.

Let $\leq_r^p$ be any of the above reductions

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

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

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

3 Disjunctive versus Conjunctive Truth-table Reductions

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

Proof: Let K be the standard $\leq_m^p$-complete set for NE as defined above. To achieve the separation we construct a set $W \in E$ and a set $A \in NEXP$ 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 $\{b(n)\}_n$:

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

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

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

stage n:

Let $A_n' = \{<i, z>|\ z \in K$ and $b(n-1)^{n-1} < |<i, z>| \leq b(n)^n$ 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 $A_n = A_n^{'} \backslash \{x\}$. This ensures that M_n^A rejects on input $0^{b(n)}$.

end of stage n

We now show that $A \in NEXP$. 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 $b(n)^n \leq 2^{n(b(n-1)^{n-1}+1)} \leq 2^{|<i,z>|^2}$

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 $b(n'-1)^{n'-1} < |x| \leq b(n')^{n'}$. $x \in A$ iff x is not the largest query asked by $M_{n'}$ and $x \in K$. This takes deterministic time $2^{2^{|x|}} < 2^{b(n)}$.

Now assume for a contradiction that A is $\leq^p_{2-c}$-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_j be the machine witnessing this reduction. But $0^{b(j)}$ is in W iff M_j on input $0^{b(j)}$ rejects. This contradicts the fact that M_j reduces W to A. This proves that A is not $\leq^p_{2-c}$-complete.

Finally we give the $\leq^p_{2-d}$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:

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

⊠

Almost the same proof technique yields the following theorem.

Theorem 2 *there exists a set $A \in NEXP$ such that A is $\leq^p_{2-c}$-complete but not $\leq^p_{2-d}$-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 $0^{b(n)} \notin W$ iff M_n^A accepts. Note that $x \in K$ iff $<0, x> \in A$ and $<1, x> \in A$. The $\leq^p_{2-c}$reduction from K to A becomes:

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

⊠

It is easy to see that the proofs generalize to $\leq^p_{k-d}$ -complete sets v.s. $\leq^p_{k-c}$ -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 $\leq^p_{k-tt}$ -complete for NE but not $\leq^p_{k-d}$ ($\leq^p_{k-c}$) -complete for NE .*

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

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

Corollary 5 *For all $k \geq 2$ there exists a set A that is $\leq^p_{k-tt}$ -complete for NE but neither $\leq^p_{d-tt}$ -complete nor $\leq^p_{c-tt}$-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 $\leq^p_{1-tt}$and $\leq^p_m$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 $\in NE \cap co\text{-}NE$ that are $\leq^p_{1-tt}$-reducible to a complete set are also $\leq^p_m$-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 $\leq^p_{1-tt}$-complete set for NE . For every set $A \in NE \cap co\text{-}NE$, $A \leq^p_m T$.*

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 $NE \cap co\text{-}NE$. 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 $<i, x>$ in D iff $x \in A$
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 $NE \cap co\text{-}NE$

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 $z \in T$. ⊠

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 $NE \cap co\text{-}NE$ and by lemma 6 they are many-one reducible to T via some other reduction. More formally:

Theorem 7 *Every $\leq^p_{1-tt}$-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.

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

CLAIM 8 *A_2 is in $NE \cap co\text{-}NE$.*

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

$x \in A_2$ iff machine M_j in case 2 and $x \in A$
$x \notin A_2$ iff machine M_j 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 z. If M_j in case 2 then x is in A iff x is in A_2. Since A_2 is in $NE \cap co\text{-}NE$ 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 $\leq_m^{rec}$ for a many-one reduction and $\leq_{1-tt}^{rec}$ 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 Σ_k be the k^{th} level of the arithmetic hierarchy as defined in [9]. For all k if A is $\leq_{1-tt}^{rec}$ -complete for Σ_k then A is $\leq_m^{rec}$ -complete for Σ_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 $P = NP \cap co\text{-}NP$however, we can prove it.

Corollary 10 *If $P = NP \cap co\text{-}NP$then every $\leq_{1-tt}^p$-complete set for NP is $\leq_m^p$-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 $B \leq_{k-T}^p A$ 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 $\leq_{k-T}^p$-complete but not $\leq_{(2^k-2)-tt}^p$ -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 $\leq_{2-T}^p$-complete but not $\leq_{2-tt}^p$-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 $\leq_m^p$-complete set for NE and let $\{b(n)\}_n$ the sequence defined in the proof of theorem 1. We will construct sets D and $W \in NEXP$ such that $W \not\leq_{2-tt}^p D$, and $K \leq_{2-T}^p D$. W and D will be constructed in stages, $D = \bigcup_{n=0}^{\infty} D_n$.

To ensure that $K \leq_{2-T}^p D$, 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 $K \leq_{2-T}^p D$ 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 $\in \{0,1,2\} \times \Sigma^*$ 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 $D \in NEXP$, $W \in E$ and W is not $\leq_{2-tt}^p D$.

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. ⊠

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 $\leq^p_{k-T}$-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 $<0, x>$. 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.

In stage 0 $D_0 = W = \emptyset$

stage n:

Let $D_n = \{<i, x>| \ x \in K$ and $b(n-1)^{n-1} < |<i, x>| \leq b(n)^n$ 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 $<i_0, x>$.

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 $D_n := D_n \cup \{<i, y>\}$
   if i_0 is in the right subtree of i then $D_n := D_n \setminus \{<i, y>\}$
2. if i occurs to the left of the path from the root to i_0 then $D_n := D_n \cup \{<i, y>\}$
3. if i occurs to the right of the path from the root to i_0 then $D_n := D_n \setminus \{<i, y>\}$
4. if i is in the left subtree of i_0 then $D_n := D_n \cup \{<i, y>\}$
5. if i is in the right subtree of i_0 then $D_n := D_n \setminus \{<i, y>\}$

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 $D \in NEXP$, $W \in E$ and W is not $\leq^p_{(2^k-2)-tt} D$.

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 $\leq^p_{(k+1)-tt}$-complete but not $\leq^p_{k-T}$-complete. A $\leq^p_{k-T}$ 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 $\leq^p_{k-T}$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 $\leq^p_{(k+1)-tt}$reduction.

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

Proof: We only give the proof for $k = 4$. Let K be the standard $\leq^p_m$-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:

$D'_n := \{<i,x>| \ x \in K \text{ and } b(n-1)^{(n-1)} \leq |x| \leq b(n)^n \text{ and } 0 \leq i \leq 5\}$

Simulate M_n on input $0^{b(n)}$, compute the answers to the queries that are small i.e. $< b(n-1)^{(n-1)}$. Now evaluate the branch where all the other queries receive the answer NO. Let Q' be the set of the queries that are big ($\geq b(n-1)^{(n-1)}$).

Put $0^{b(n)}$ in W iff M_n rejects

$D_n := D'_n \backslash Q'$

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:

$$g(x) = \{<0,x> \vee \ldots <4,x>\}$$

⊠

Corollary 14 *If $k < m < 2^k - 1$, then $\leq^p_{k-T}$and $\leq^p_{m-tt}$ 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

[1] Berman L. *On the structure of complete sets*. Proc. 17th IEEE conference on Foundations of Computer Science (1976) pp76-80.

[2] Buhrman H., S. Homer & L. Torenvliet. *Honest reductions, completeness and nondeterministic complexity classes*. Report CT-89-08, University of Amsterdam, Dept. of Computer Science.

[3] Buhrman H., E. Spaan & L. Torenvliet. *On Adaptive Resource Bounded Computations*. Report CT-89-08, University of Amsterdam, Dept. of Computer Science.

[4] Cook, S. A. *The complexity of theorem-proving procedures*. Proc. 3d ACM Symp. on Theory of Computing, Assoc. for Computing Machinery, New York (1971) pp151–158.

[5] Homer, S. Personal communication.

[6] Karp, R.M. *Reducibility among combinatorial problems*. Complexity of Computer Computations, R.E. Miller & J.W. Thatcher eds. Plenum N.Y. pp85–103.

[7] Ladner R.E. & N.A. Lynch. *Relativization of Questions About Log Space Computability*. Mathematical Systems Theory 10 (1976) pp19–32.

[8] Ladner, R.E., N. Lynch & A.L. Selman. *A comparison of polynomial time reducibilities*. Theoretical Computer Science 1 (1975) 103–123.

[9] Soare, R.I. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic. Springer-Verlag (1987) pp60–61.

[10] Watanabe, O. *A comparison of polynomial time completeness notions*. Theoretical Computer Science 54 (1987) pp249–265.

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
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`
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