A COMPARISON OF REDUCTIONS ON NONDETERMINISTIC SPACE

Harry Buhrman
Leen Torenvliet

ITLI Prepublication Series
for Computation and Complexity Theory CT-89-04

University of Amsterdam
A COMPARISON OF REDUCTIONS ON NONDETERMINISTIC SPACE

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

Received April 1989
Abstract

In this paper we investigate relations between various types of logarithmic space bounded reducibility notions on nondeterministic space bounded complexity classes. It turns out that differences exist between almost all completeness notions under different reducibility notions for almost all complexity classes considered.
1 Introduction

An important concept in structural complexity theory is the notion of resource bounded reductions. Since the first use of polynomial time bounded Turing reductions by Cook [2] and shortly thereafter the introduction of polynomial time bounded many-one reductions by Karp [6], considerable effort was put in the investigation of properties and the comparison of different reductions and corresponding completeness notions. In 1975 an extensive survey of different types of reductions—and differences between these reductions on $DEXT (= \bigcup_{n \in \mathbb{N}} \text{DTIME}(2^n))$—was given by Ladner, Lynch and Selman [7].

Berman and Hartmanis [1] introduced in 1979 "P-isomorphism" in order to investigate a structural similarity between all natural $\leq^p_m$-complete sets for $NP$. It turned out that all well-known $NP$-complete sets [3] are P-isomorphic, hence they conjectured that all $NP$-complete sets with respect to $\leq^p_m$ reductions are P-isomorphic (the Berman-Hartmanis conjecture [1]). In 1987 Watanabe [9] proved almost all possible differences between the corresponding polynomial time completeness notions on $DEXT$.

Probably because of the implications on the $P \neq NP$ question—direct results on differences between polynomial time reductions on $NP$ immediately solve the problem—the bulk of the research was devoted to time-bounded reductions and especially polynomial-time bounded reductions have received a great deal of attention. Much less is known about space-bounded reductions. Though obviously a notion at least as powerful as polynomial time bounded reductions, the logarithmic space bounded reduction has received far less attention in the past. This is the more peculiar since logarithmic space bounded reductions are useful for proving completeness results where polynomial time bounded reductions are not. Logarithmic space bounded reductions may be used for proving completeness of certain classes in $P$, thereby making the existence of an efficient parallel algorithm highly improbable [2].

We can only guess for the reason of this underexposure of logarithmic space bounded reduction to research efforts. One reason may be that the comparison of polynomial time bounded reductions was mainly directed to a setting of exponential time—a setting in which the differences both in properties and in importance are much less apparent—another may be that the observations needed to show differences between different loga-

rithmic space bounded reductions on interesting complexity classes were missing.

Recently, Immerman [4] and independently Szelepcenyi [8] proved that many nondeterministic space bounded complexity classes are closed under complementation. It turns out that this closedness under complementation is exactly the observation needed to show that many of the results derived by Watanabe for $DEXT$ go through for nondeterministic space bounded classes and logarithmic space bounded reductions. Following the lines of [9] we derive these results in the present paper.

2 Preliminaries

Let $\Sigma = \{0, 1\}$, and let $\Sigma^*$ denote the set of all words over $\Sigma$. A language is a subset of $\Sigma^*$. For any string $z$, $|z|$ denotes the length of $z$ and for any set $S$, let $|S|$ denote the cardinality of $S$. For any set $A$ the set $A^{\leq n}$ is $\{z \in A \mid |z| \leq n\}$. We assume a pairing function computable in logarithmic space from $\Sigma^* \times \Sigma^*$ to $\Sigma^*$. Let $\lambda xy. < x, y >$, be such a function.
We will use the following shorthands:

\[ \exists n : \text{for infinitely many } n. \]
\[ \forall n : \text{for all but finitely many } n. \]
We will also refer to the latter as : \( n \) for almost all \( n \).

Our machine model is a standard multi-tape nondeterministic \( S(n) \) space bounded Turing machine acceptor. A Turing machine may or may not be an oracle machine. The space used by the oracle tape as well as the input and output tape is not charged to the computation. We assume a standard enumeration of logarithmic-space bounded deterministic Turing machines \( M_1, M_2, \ldots \), and a universal Turing machine \( M_U \), which on input \( <i, z> \) simulates \( M_i \) on input \( z \). We use \( M^A(z) \) to denote the execution of \( M \) on input \( z \) relative to oracle set \( A \). \( Q(M, z, A) \) denotes the set of queries made by \( M^A(z) \) during its computation. For any NTM (nondeterministic Turing machine) and set \( A \), \( L(M) \) denotes the set of strings accepted by \( M \) and \( L(M, A) \) denotes the set of strings accepted by \( M \) relative to oracle set \( A \). \( L(M)(L(M, A)) \) will be called the language of \( M \) (relative to \( A \)).

In the sequel let \( S(n) \) be any space constructible function, which satisfies the following property:

\[
\liminf_{n \to \infty} \frac{\log(n)}{S(n)} = 0
\]

To obtain model-independent results we assume that all complexity classes are closed under constant factor space-overhead. By \( \text{NSPACE}(S(n)) \) we mean the complexity class specified by nondeterministic \( S(n) \) space-bounded Turing machines. We use \( \text{LOGSPACE} \) as an abbreviation for \( \text{DSPACE}(\log(n)) \).

We also consider deterministic logarithmic-space bounded Turing transducers and their standard enumeration. \( T_i \) denotes the \( i \)th logarithmic-space bounded Turing transducer and also the partial function from \( \Sigma^* \) to \( \Sigma^* \). Here we also assume a standard universal Turing transducer \( T_U \), which on input \( <i, z> \) computes \( T_i \) on input \( z \). Let \( \text{FL} \) be the set of all total functions in \( \{ T_i \}_{i \in \mathbb{N}} \).

The ordered pair \( <\alpha_1, \ldots, \alpha_k >, \alpha > \) is called a truth-table condition of norm \( k \) if \( <\alpha_1, \ldots, \alpha_k > \) is a \( k \)-tuple \( (k > 0) \) of strings in \( \Sigma^* \) and \( \alpha \) is a \( k \)-ary Boolean function [7]. The set \( \{\alpha_1, \ldots, \alpha_k \} \) 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 \( \Sigma^* \). For any tt-function \( f \) and any \( x \) in \( \Sigma^* \), \( \text{Ass}(f, x) \) denotes the associated set of \( f(x) \). For any space constructive function \( \phi \), \( f \) is called a \( \phi(n) \)-bounded truth-table function, if for every \( x \) in \( \Sigma^* \), the norm of the tt-condition \( f(x) \) is bounded by \( \phi(|x|) \). If a function \( f \) is a \( k \)-tt function for some integer \( (k > 0) \) then we call \( f \) a bounded truth-table (btt-) function. We say a tt-function is a disjunctive truth table (dtt-) function if, for any \( x \) in \( \Sigma^* \), the Boolean function of the tt-condition is disjunctive. As mentioned above, the space used by the oracle tape, is not charged to the computation, this means that the space used by the associated set is neither charged to the computation.

It is now time for some definitions.

**Definition 2.1** Let \( A_1, A_2 \subseteq \Sigma^* \).
1. $A_1$ is logspace many-one reducible to $A_2$ ($\leq_{m}^{\text{logspace}}$-reducible), if there exists a function $f$, computable in logarithmic space, such that $z \in A_1$ iff $f(x) \in A_2$.

2. $A_1$ is logspace truth-table reducible ($\leq_{tt}^{\text{logspace}}$-reducible) to $A_2$, if there exists a logarithmic space-bounded tt-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 $\langle a_1, \ldots, a_k, \alpha >$, and $\chi_{A_2}$ is the characteristic function of the set $A_2$.

3. $A_1$ is logspace Turing reducible ($\leq_{T}^{\text{logspace}}$-reducible) to $A_2$ if there exists a logarithmic space-bounded deterministic Turing machine $M$ such that $A_1 = L(M, A_2)$.

4. $A_1$ is logspace btt-reducible ($\leq_{\text{btt}}^{\text{logspace}}$-reducible) to $A_2$ if $A_1 \leq_{tt}^{\text{logspace}} A_2$ by some btt-function.

5. $A_1$ is logspace disjunctive reducible ($\leq_{d}^{\text{logspace}}$-reducible) to $A_2$ if $A_1 \leq_{tt}^{\text{logspace}} A_2$ by some dtt-function.

**Definition 2.2** Let $\leq_{r}^{\text{logspace}}$ be any of the above reductions.

1. A set $A$ is $\leq_{r}^{\text{logspace}}$-hard for some complexity class $C$ if, for all $B \in C$, $B$ is $\leq_{r}^{\text{logspace}}$-reducible to $A$.

2. A set $A$ is $\leq_{r}^{\text{logspace}}$-complete for some complexity class $C$ if $A$ is $\leq_{r}^{\text{logspace}}$-hard for $C$ and $A \in C$.

We use standard $\text{NSPACE}(S(n))$-complete sets w.r.t. $\leq_{r}^{\text{logspace}}$-reductions. [5]

**Definition 2.3** For any complexity class $C$, a set $A$ is $C$-immune if for every set $L$, $L \subseteq A$ and $\|L\| = \aleph_0$, $L \notin C$.

3 Structure of complete sets in $\text{NSPACE}$

Let $\leq_{r}^{\text{logspace}}$ be any of the reductions introduced in the previous section. We will examine if $\leq_{r}^{\text{logspace}}$-complete sets in $\text{NSPACE}(S(n))$ have an infinite subset, induced by some function $f$. We call a function $f$ length increasing if for almost all $z$: $|f(x)| > |x|$. One way to show that a set has such an infinite subset is to construct a function $f$—which is length increasing—such that $\{f(0^n) | n \in \mathbb{N}\} \subseteq A$. More formally:

**Definition 3.1** Let $M$ be a deterministic logarithmic-space bounded oracle machine and let $A$ be an oracle set such that $M$ witnesses a $\leq_{r}^{\text{logspace}}$-reduction and not for every input $x$, $Q(M, A, x) = \emptyset$.

1. We say that a function $f$ is generated by $M$ and $A$ iff, $f$ maps almost all $x \in \Sigma^*$ to some element of $Q(M, A, x)$.

2. $F_M = \{f \mid f$ is generated by a logarithmic-space bounded oracle machine which corresponds to some $\leq_{m}^{\text{logspace}}$-reduction$\}$. 

3
3. \( \mathsf{Fbtt} = \{ f \mid f \text{ is generated by a logarithmic-space bounded oracle machine which corresponds to some } \leq_{\text{btt}}^{\text{logspace-reduction}} \} \).

4. \( \mathsf{Ftt} = \{ f \mid f \text{ is generated by a logarithmic-space bounded oracle machine which corresponds to some } \leq_{\text{tt}}^{\text{logspace-reduction}} \} \).

5. \( \mathsf{FT} = \{ f \mid f \text{ is generated by a logarithmic-space bounded oracle machine which corresponds to some } \leq_{\text{tt}}^{\text{logspace-reduction}} \} \).

6. Let \( \mathsf{Fr} \) be any of the above classes. A set \( A \) has an \( \mathsf{Fr} \)-subset if there exists a function \( f \in \mathsf{Fr} \), which is total and length increasing, such that for almost all \( x \in \Sigma^* \), \( f(x) \in A \).

Remember that \( \mathsf{FL} \) denotes the set of all total functions, computable in logarithmic space, so instead of \( \mathsf{Fm} \) we use \( \mathsf{FL} \).

We will show that every \( \leq_{\text{tt}}^{\text{logspace}} \)-complete set in \( \mathsf{NSPACE}(S(n)) \) has an \( \mathsf{Fr} \)-subset. But first we need a theorem.

**Theorem 1** Let \( A \) be any set in \( \mathsf{NSPACE}(S(n)) \). There exists a set \( L_A \in \mathsf{NSPACE}(S(n)) \) \( \leq_{\text{tt}}^{\text{logspace}} A \) by \( M_i \) then for almost all \( x \) there exists a \( y \) in \( Q(M_i, A, < i, x >) \cap A \) such that \( |y| > |x| \). That is if \( L_A \) is \( \leq_{\text{tt}}^{\text{logspace}} A \) by some \( M_i \) then for almost all input \( x \), \( M_i \) queries a \( y \) to \( A \), which is larger (in length) than \( |x| \).

**Proof:** Let \( \{ M_i \}_{i \in \mathbb{N}} \) be an enumeration of logarithmic-space bounded oracle machines. We define \( L_A \) as follows:

\[
< i, x > \in L_A \iff \text{the simulation of } M_i^{A^{\leq |x|}} \text{ uses } \leq (|< i, x >|) \text{ tape cells and } \]
\[
< i, x > \notin L(A, A^{\leq |< i, x >|}).
\]

Since \( A \in \mathsf{NSPACE}(S(n)) \), there exists a nondeterministic \( S(n) \)-space bounded Turing machine \( M_A \), which accepts \( A \). Immerman [4] and Szepesváry [8] showed independently that nondeterministic space is closed under complementation. Therefore the complement of \( A (\overline{A}) \) is also in \( \mathsf{NSPACE}(S(n)) \), and is recognized by a nondeterministic \( S(n) \)-space bounded Turing machine \( M^\overline{A} \). We are first going to construct a machine that recognizes \( L_A \). Note that \( < i, x > \in L_A \) iff simulation of \( M_i \) uses more than \( S(|< i, x >|) \) tape cells or \( < i, x > \in L(M_i, A^{\leq |< i, x >|}) \). Consider the following machine \( M \):

```
input < i, x >
mark off S(|< i, x >|) tape cells.
Simulate \( M_i \) on input \( x \)
if \( M_i \) queries \( a \) then
  if \( |a| > |< i, x >| \) continue computation of \( M_i \) in the NO state \(^1\)
  else guess if \( a \) is in \( A \) or in \( \overline{A} \) and run \( M_A \) or \( M^\overline{A} \) on input \( a \).
if it rejects then REJECT
else if \( M_A \) accepts then continue \( M_i \) in the YES state
```

\(^1\) To check if \( |a| > |< i, x >| \), we use a counter keeping track of the number of symbols written on the oracle tape between two queries.
else if $M_A$ accepts then continue $M_i$ in the NO state.

accepts $M_i$ accepts or simulation of $M_i$
uses more then $S(<i,k>)$ tape cells.

end.

It is easy to see that $L(M)$ is in $NSPACE(S(n))$. Hence the complement of $L(M)$ is also in $NSPACE(S(n))$, by some machine $\overline{M}$. So $<i,k> \notin L(M)$ iff $<i,k> \in L(\overline{M})$ iff $<i,k> \in L_A$. This shows that $L_A$ is in $NSPACE(S(n))$.

Suppose for a contradiction that $M_j$ is a logarithmic-space bounded oracle machine such that $L_A = L(M_j,A)$, i.e. $L_A$ is $\leq_T^{logspace}$ reducible to $A$, via $M_j$. Let $\alpha$ be a string such that the simulation of $M_j$ uses $S(<j,k>)$ tape cells. Then there exists at least one $y$ in $Q(M_j, A,<j,k>) \cap A$ such that $|y| > |z|$.

Suppose otherwise. That is, the length of each element of $Q(M_j, A,<j,k>) \cap A$ is $\leq |z|$. Then $M_j^{<j,k>} (x) = M_j^{<j,x>} (y)$, and $<j,k> \in L(M_j, A)$. Thus $<j,k> \in L_A$ iff $<j,k> \notin L(M_j, A)$, which contradicts the fact that $L_A = L(M_j, A)$. $\Box$

Corollary 1 Every $\leq_T^{logspace}$-complete set in $NSPACE(S(n))$, has an $FT$-subset.

Proof: Let $A$ be a $\leq_T^{logspace}$-complete set in $NSPACE(S(n))$. We now construct the set $L_A$ w.r.t. $A$ in the same way as in theorem 1. Since $A$ is $\leq_T^{logspace}$-complete for $NSPACE(S(n))$, we can now apply theorem 1: For almost all $x$ there exists a $y_x$ in $Q(M_i, A,<i,x>) \cap A$ such that $|y_x| > |x|$. We now define the following function $g$:

$$g(x) = \begin{cases} y_x & \text{if } y_x \text{ exists} \\ \text{some element of } A & \text{otherwise} \end{cases}$$

The function $g$ is total and length increasing because for almost all $x \in \Sigma^*$, $|g(x)| > |x|$, and $g(x) \in A$. Furthermore $g \in FT$, which proves the corollary. $\Box$

Corollary 2 Let $A \subseteq \Sigma^*$.

1. If $A$ is $\leq_{tt}^{logspace}$-complete for $NSPACE(S(n))$, then $A$ has an $Ftt$-subset.
2. If $A$ is $\leq_{bit}^{logspace}$-complete for $NSPACE(S(n))$, then $A$ has an $Fbtt$-subset.
3. If $A$ is $\leq_{m}^{logspace}$-complete for $NSPACE(S(n))$, then $A$ has an FL-subset.

Proof: The proof is similar to the proof of corollary 1 and is left to the reader. $\Box$

Corollary 3 Every $\leq_{m}^{logspace}$-complete set for $NSPACE(S(n))$ is not $LOGSPACE$-immune.

Proof: Let $C$ be any $\leq_{m}^{logspace}$-complete set for $NSPACE(S(n))$. Consider the set $L_C$ and a many one reduction from $L_C$ to $C$ via machine $M_C$. Applying theorem 1 it follows that for almost all $x$ $M_C$ queries a $y$ to $C$ such that $|y| > |x|$ and $y \in C$. So the set $\{Q(M_C,0^n,C)| n > n_0 \}$ for some $n_0$ large enough, is an infinite subset of $C$. Consider the following machine $M$, which accepts this subset:
input \( z \)
\[
n := |z|
\]
for all \( n' \), \( n_0 < n' < n \) do
run \( M_C \) on input \( 0^{n'} \)
   if \( Q(M_C, 0^{n'}, C) = z \) then ACCEPT
end do
REJECT

Since \( M_C \) is a logarithmic-space bounded oracle machine it is easy to see that \( M \) is also a logarithmic-space bounded machine. Furthermore \( M \) accepts if and only if \( z \in C \) and \( ||L(M)|| = \aleph_0 \). \( \Box \)

**Corollary 4** Every \( \leq_{m}^{\text{logspace}} \)-complete set for \( \text{NSPACE}(S(n)) \) has infinitely many subsets \( \{B_i\}_{i \in \mathbb{N}} \), which are \( \in \text{LOGSPACE} \) and \( ||B_i|| = \aleph_0 \).

**Proof:** Corollary 3 states that every \( \leq_{m}^{\text{logspace}} \)-complete set has an infinite subset \( \in \text{LOGSPACE} \). Let \( A \) be a \( \leq_{m}^{\text{logspace}} \)-complete set for \( \text{NSPACE}(S(n)) \) and let \( B_0 \) be such an infinite subset. Consider the set \( A_1 = A \backslash B_0 \). \( A_1 \) is \( \in \text{NSPACE}(S(n)) \) and \( A_1 \) is \( \leq_{m}^{\text{logspace}} \)-complete via the following reduction from \( A \) to \( A_1 \):

input \( z \)
   if \( z \in B_0 \) then output a fixed \( y \notin A_1 \)
   else output \( z \)
end

Now we can apply corollary 3 again on \( A_1 \). This process can be repeated infinitely often and will generate the subsets \( \{B_i\}_{i \in \mathbb{N}} \) as promised. \( \Box \)

4 Differences between complete sets in \( \text{NSPACE} \)

At this point we have obtained some useful properties of \( \text{NSPACE}(S(n)) \)-complete sets. For example every \( \leq_{m}^{\text{logspace}} \)-complete set in \( \text{NSPACE}(S(n)) \) has an \( FL \)-subset. Hence to construct a \( \leq^{\text{logspace}}_{2-d} \)-complete set, which is not \( \leq^{\text{logspace}}_{m} \)-complete it is sufficient to construct a set, which is \( \leq^{\text{logspace}}_{2-d} \)-complete, but has no \( FL \)-subsets. This can be done by straightforward diagonalization. The aim of the diagonalization is to put \( g(x) \), for every length increasing function \( g \in FL \) and for almost all \( x \), in the complement of \( D \), whilst on the other hand \( D \) must be \( \leq^{\text{logspace}}_{\text{btt}} \)-complete. We will construct the set \( D \) by stages. This is done with the help of a function \( b : \mathbb{N} \mapsto \mathbb{N} \). At each stage we define the set \( D_n = \{ x \in D | b(n-1) < |x| \leq b(n) \} \) and \( D = \bigcup_{n \geq 0} D_n \). This yields the following theorem.

**Theorem 2** There exists a \( \leq^{\text{logspace}}_{2-d} \)-complete set \( D \in \text{NSPACE}(S(n)) \), which is not \( \leq_{m}^{\text{logspace}} \)-complete.
Proof: Recall that \( \{T_i\}_{i \in \mathcal{N}} \) is an effective enumeration of nondeterministic logarithmic-space bounded Turing transducers. Let \( C = L(M_c) \) be a standard \( \logspace \)-complete set in \( \text{NSPACE}(S(n)) \). \( D \) is defined by the following construction:

Requirements: \( \forall f \in \text{FL}[\forall n |f(0^n)| > n \Rightarrow \exists m f(0^m) \notin D] \)

Construction:

\[ \begin{align*} 
\text{stage 0:} \\
& b(0) := 0 \\
& D_0 := \emptyset \\
\text{stage } n + 1 \\
& n' := b(n - 1) \\
& y := \begin{cases} 
T_i(0^n') & \text{if the simulation of } T_i \text{ uses } \leq S(n) \text{ tape cells} \\
\uparrow & \text{otherwise} 
\end{cases} \\
\text{(here } i \text{ is the first element of the pair } <i, j> = n.) \\
\text{if } y \neq \uparrow \text{ and } |y| > n' \text{ then} \\
& b(n) := |y| \\
& D_n := \{0z, 1x | b(n - 1) < |x| + 1 \leq b(n) \text{ and } x \in C\} \setminus \{y\} \\
\text{else} \\
& b(n) := b(n - 1) + 1 \\
& D_n := \{0z, |x| + 1 = b(n) \text{ and } x \in C\} \\
\end{align*} \]

end if construction

Consider the following machine \( M \):

Let \( a = 0 \) or \( 1 \).

input(az)

if \( M_c(x) \) rejects then reject

else Simulate stage construction until \( b(n) \geq |az| \) is reached\(^2\)

if \( b(n) = |az| \) then

if \( y = \uparrow \) and \( a = 0 \) then accept else reject

else if \( az = T_i(0^n') \) then reject else accept

end

It is clear that \( L(M) \) is in \( \text{NSPACE}(S(n)) \). Let \( g \) be a function in \( \text{FL} \) such that for almost all \( x, \ |g(x)| > |x| \). Then there exists a \( i \) such that \( T_i \) computes \( g \). Because \( T_i \) satisfies the requirements it follows that \( g(x) \) is not in \( D \) for infinitely many \( x \) and \( D \) does not have an \( \text{FL}-\)subset. Every \( \logspace \)-complete set in \( \text{NSPACE}(S(n)) \) has an \( \text{FL}-\)subset (corollary 2), so \( D \) is not \( \logspace \)-complete for \( \text{NSPACE}(S(n)) \). Since for every \( x \in C \) 1x or 0x is in \( D \), \( C \) is \( \logspace \)-reducible to \( D \) by the following reduction \( f \):

\(^2\)It is not necessary to compute \( b(n) \) completely. If during the computation of \( b(n) \), \( b(n) \) becomes bigger than \( |az| \) it is the right \( n \). Furthermore, the value of \( n \) cannot exceed \( |az| \).
\[ \forall z. \; f(z) = \langle \langle 0, x \rangle, \alpha \rangle \text{ and } \alpha(z, y) = z \lor y. \]

This proofs that \( D \) is \( \leq_{d}^{\text{logspace}} \)-complete. \( \square \)

Corollary 5

1. For any integer \( k > 1 \), there exists a \( \leq_{d}^{\text{logspace}} \)-complete set which is not \( \leq_{m}^{\text{logspace}} \)-complete for \( \text{NSPACE}(S(n)) \).

2. For any integer \( k > 1 \), there exists a \( \leq_{k}^{\text{logspace}} \)-complete set which is not \( \leq_{m}^{\text{logspace}} \)-complete for \( \text{NSPACE}(S(n)) \).

Using the same technique we can construct a set, which is \( \leq_{d}^{\text{logspace}} \)-complete, but not \( \leq_{d}^{\text{logspace}} \)-complete.

Theorem 3 There exists a \( \leq_{d}^{\text{logspace}} \)-complete set \( D \) for \( \text{NSPACE}(S(n)) \), which is not \( \leq_{d}^{\text{logspace}} \)-complete.

Proof: Let \( \text{bin}(i) \) be the binary representation of \( i \), and \( c(i, x) = 0^m \text{bin}(i)x \), for \( x \in \Sigma^* \), \( 1 \leq i \leq |z| \) and \( |0^m \text{bin}(i)x| = 2|z| \). Let \( C \) be a standard \( \leq_{m}^{\text{logspace}} \)-complete set for \( \text{NSPACE}(S(n)) \), and \( C_x = \{ c(i, x) | 1 \leq i \leq |z| \text{ and } x \in C \} \). For every \( x \) in \( C \), we put at least one element of \( C_x \) in \( D \), so \( D \) can be \( \leq_{d}^{\text{logspace}} \)-complete. On the other hand we must ensure that, for every length increasing function \( f \) in \( \text{Fbtt} \), \( f(z) \) is not in \( C \), for almost all \( z \). Finally \( D = \bigcup_{n \geq 0} D_n \).

Requirements: \( \forall f \in \text{Fbtt} \left[ \forall \; n \mid f(0^n) \right] > n \Rightarrow \exists \; m \mid f(0^m) \notin D \]

Construction:

<table>
<thead>
<tr>
<th>stage</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b(0) := 0 )</td>
<td>( D_0 := \emptyset )</td>
</tr>
</tbody>
</table>

stage \( n + 1 \)

| \( n' := b(n - 1) \) |
| \( Y := \{ y \mid y \in \text{Ass}(T_i, 0^n') \land |y| > n' \} \) |
| \( Y := \{ y \mid y \in \text{Ass}(T_i, 0^n') \land |y| > n' \} \) |
| \( \uparrow \) |
| \( \text{if } Y \neq \emptyset \text{ and } 0 < |Y| \leq n'/2 \text{ then} \) |

| \( b(n) := \text{length of longest element of } Y \) |
| \( D_n := \{ z \mid z \in C_x \text{ and } n' < |z| \leq b(n) \} \setminus Y \) |

else

| \( b(n) := n' + 2 \) |
| \( D_n := \{ c(0, x) \mid n' < 2|x| \leq b(n) \text{ and } x \in C \} \) |
| end-if |
| end construction |

8
Using a similar approach as we did in the proof of theorem 2 it is easy to see that $D$ is in $NSPACE(S(n))$. Note that it is not necessary to store the set $Y$; it is enough to use a counter to keep track of the number elements in $Y$. Furthermore suppose $D$ has an $Fbtt$-subset. Then there exists a length increasing function $g$ in $Fbtt$, and a $k$-tt function $(k > 0)$ $f$ in FL such that $g(x) \in \text{Ass}(f, x)$ for all $x$. The number of elements in $\text{Ass}(f, x)$ is bounded by $k$. There exists a $i$ such that $T_i$ computes $f$. Because $T_i$ satisfies the requirements it follows that $g(x)$ is not in $D$ for almost all $x$, which contradicts the fact that $D$ has an $Fbtt$-subset. Therefore $D$ is not $\leq^{logspace}_{btt}$-complete. Since for ever $x \in C$ there is at least one element of $C_x$ in $D$, $C$ is $\leq^{logspace}_d$-reducible to $D$ by the following tt-function $f$:

$$\forall z. f(z) = \ll c(0, x), \ldots, c(m, x) >, \alpha >$$

$$\alpha(a_1, \ldots, a_m) = a_1 \lor \ldots \lor a_m$$

where $m = |x|$. So $D$ is $\leq^{logspace}_d$-complete.

**Corollary 6** There exists a $\leq^{logspace}_{tt}$-complete set $D$ for $NSPACE(S(n))$, which is not $\leq^{logspace}_{btt}$-complete.

5 Conclusions

In the previous sections we proved that several differences exists between logspace reductions. This can be generalized in the following way:

Instead of looking at logspace reductions we can also look at $R(n)$-space reductions, where $R(n)$ is any space constructible function with the following property:

$$\liminf_{n \to \infty} \frac{R(n)}{S(n)} = \liminf_{n \to \infty} \frac{\log(n)}{R(n)} = 0$$

All the previous obtained results also go through with respect to this kind of reduction. Now the question rises whether there is a difference between a $\leq^{logspace}_D$-complete set and a $\leq^{R(n)}_m$-space -complete set. We conjecture that this difference exists, i.e. there exists a set $D \in NSPACE(S(n))$, which is $\leq^{R(n)}_m$-space -complete, but is not $\leq^{logspace}_D$-complete for $NSPACE(S(n))$.

Furthermore the results also go through in $DSPACE(S(n))$ because the construction of the set $L_A$ (theorem 1) is also possible in $DSPACE(S(n))$. Other deterministic classes are also worthwhile to investigate on differences between logspace reductions. It is not clear if these differences exist in $P$ (deterministic polynomial time), because it is not known if $P = LOGSPACE$. On the other hand for any deterministic time class, $\not\supset P$, these differences exist.

Watanabe [9] also proved that differences between disjunctive versus conjunctive polynomial time tt -reductions in $DEXT$ exist. And that differences between polynomial tt -reductions and polynomial Turing reductions exist. We conjecture that this is also true for logspace reductions in $NSPACE(S(n))$. 

9
Bibliography


The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas
86-03 Johan van Benthem
86-04 Reimand Muskens
86-05 Kenneth A. Bowen, Dick de Jongh
86-06 Johan van Benthem

1987
87-01 Jeroen Groenendijk, Martin Stokhof
87-02 Renate Bartsch
87-03 Jan Willem Klop, Roel de Vrijer
87-04 Johan van Benthem
87-05 Victor Sanchez Valencia
87-06 Eleonore Oversteegen
87-07 Johan van Benthem
87-08 Renate Bartsch
87-09 Herman Hendriks

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen
LP-88-02 Yde Venema
LP-88-03...

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten
ML-88-02 M.D.G. Swaen
ML-88-03 Dick de Jongh, Frank Veltman
ML-88-04 A.S. Troelstra
ML-88-05 A.S. Troelstra

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi
CT-88-02 Michiel H.M. Smid
CT-88-03 Michiel H.M. Smid, Mark H. Overmars
CT-88-04 Dick de Jongh, Lex Hendriks
CT-88-05 Peter van Emde Boas
CT-88-06 Michiel H.M. Smid
CT-88-07 Johan van Benthem
CT-88-08 Michiel H.M. Smid, Mark H. Overmars
CT-88-09 Theo M.V. Jansen
CT-88-10 Edith Spaan, Leen Torenvliet
CT-88-11 Sieger van Denneheuvel

Other prepublications:
X-88-01 Marc Jumelet

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser
ML-89-02 Roel de Vrijer

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid
CT-89-02 Peter van Emde Boas
CT-89-03 Ming Li, Herman Neufelise
CT-89-04 Harry Buhrman, Leen Torenvliet

Other prepublications:
X-89-01 Marianne Kalsebek
X-89-02 G. Wagemakers

The Fine-Structure of Categorial Semantics
Explicit Fixed Points for Interpretability Logic
Extending the Lambda Calculus with Surjective Pairing is conservative
Dynamic Deferred Data Structures
Machine Models and Simulations
On Space efficient Solutions
A Comparison of Reductions on Nondeterministic Space
An Orey Sentence for Predicative Arithmetic
New Foundations. A Survey of Quine's Set Theory