Institute for Language, Logic and Information

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

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

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Roetersstraat 15
1018WB Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP 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 $DEXT$ ($= \cup_{n\in I\!N}$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 $P \overset{?}{=} NP$ 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 logarithmic space bounded reductions on interesting complexity classes were missing.

Recently, Immerman [4] and independently Szelepcsényi [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 Σ^* denote the set of all words over Σ. A language is a subset of Σ^*. For any string x, $|x|$ denotes the length of x and for any set S, let $||S||$ denote the cardinality of S. For any set A the set $A^{\leq n}$ is $\{x \in A \mid |x| \leq n\}$. We assume a pairing function computable in logarithmic space from $\Sigma^* \times \Sigma^*$ to Σ^*. Let $\lambda xy. < x, y >$, be such a function.

We will use the following shorthands:
$\overset{\infty}{\exists}\ n$: for infinitely many n.
$\overset{\infty}{\forall}\ n$: for all but finitely many n.
We will also refer to the latter as : 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 $< i, x >$ simulates M_i on input x. We use $M^A(x)$ to denote the execution of M on input x relative to oracle set A. $Q(M, x, A)$ denotes the set of queries made by $M^A(x)$ 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 $NSPACE(S(n))$we mean the complexity class specified by nondeterministic $S(n)$ space- bounded Turing machines. We use $LOGSPACE$ as an abreviation for $DSPACE(\log(n))$.

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

The ordered pair $<< a_1, \ldots, a_k >, \alpha >$ is called a truth-table condition of norm k if $< a_1, \ldots, a_k >$ is a k-tuple ($k > 0$) of strings in Σ^* and α is a k-ary Boolean function [7]. 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^*. For any tt-function$f$ and any x in Σ^*, Ass(f, x) denotes the associated set of $f(x)$. For any space constructible function ϕ, f is called a $\phi(n)$-bounded truth-table function, if for every x in Σ^*, 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 Σ^*, 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 ($\leq_m^{logspace}$-reducible), if there exists a function f, computable in logarithmic space, such that $x \in A_1$ iff $f(x) \in A_2$.*
2. *A_1 is logspace truth-table reducible ($\leq_{tt}^{logspace}$-reducible) to A_2, if there exists a logarithmic space-bounded tt-function f such that $\alpha(\chi_{A_2}(a_1)\cdots\chi_{A_2}(a_k)) =$ **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 logspace Turing reducible ($\leq_T^{logspace}$-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 ($\leq_{btt}^{logspace}$-reducible) to A_2 if $A_1 \leq_{tt}^{logspace} A_2$ by some btt-function.*
5. *A_1 is logspace disjunctive reducible ($\leq_d^{logspace}$-reducible) to A_2 if $A_1 \leq_{tt}^{logspace} A_2$ by some dtt-function.*

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

1. *A set A is $\leq_r^{logspace}$-hard for some complexity class C if, for all $B \in C$, B is $\leq_r^{logspace}$-reducible to A.*
2. *A set A is $\leq_r^{logspace}$-complete for some complexity class C if A is $\leq_r^{logspace}$- hard for C and $A \in C$.*

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

Definition 2.3 *For any complecity 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 NSPACE

Let $\leq_r^{logspace}$ be any of the reductions introduced in the previous section. We will examine if $\leq_r^{logspace}$-complete sets in $NSPACE(S(n))$ have an infinite subset, induced by some function f. We call a function f *length increasing* if for almost all x: $|f(x)| > |x|$. One way to show that a set has such an infinite subset is to construct a function f—which is length increasing—such that $\{f(0^n) \mid n \in I\!N\} \subseteq A$. 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 $\leq_r^{logspace}$-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. *$Fm = \{f \mid f$ is generated by a logarithmic-space bounded oracle machine which corresponds to some $\leq_m^{logspace}$-reduction$\}$.*

3. *Fbtt* = $\{f \mid f$ *is generated by a logarithmic-space bounded oracle machine which corresponds to some* $\le_{btt}^{logspace}$*-reduction*$\}$.
4. *Ftt* =$\{f \mid f$ *is generated by a logarithmic-space bounded oracle machine which corresponds to some* $\le_{tt}^{logspace}$*-reduction*$\}$
5. *FT* = $\{f \mid f$ *is generated by a logarithmic-space bounded oracle machine which corresponds to some* $\le_{T}^{logspace}$*-reduction*$\}$
6. *Let Fr be any of the above classes. A set A has an Fr-subset if there exists a function* $f \in Fr$*, which is total and length increasing, such that for almost all* $x \in \Sigma^*$, $f(x) \in A$.

Remember that *FL* denotes the set of all total functions, computable in logarithmic space, so instead of *Fm* we use *FL*.

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

Theorem 1 *Let A be any set in NSPACE*$(S(n))$. *There exists a set* $L_A \in NSPACE(S(n)) \subseteq N \times \Sigma^*$*such that: if* $\exists M_i : L_A \le_T^{logspace} A$ *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* $\le_T^{logspace} A$ *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 $\{M_i\}_{i \in N}$ be an enumeration of logarithmic-space bounded oracle machines. We define L_A as follows:

$$< i, x > \in L_A \Leftrightarrow \text{the simulation of } M_i^{A^{\le |< i,x >|}} \text{uses} \le (S(| < i, x > |)) \text{ tape cells and } < i, x > \notin L(M_i, A^{\le |< i,x >|}).$$

Since $A \in NSPACE(S(n))$, there exists a nondeterministic $S(n)$-space bounded Turing machine M_A, which accepts A. Immerman [4] and Szelepcsény [8] showed independently that nondeterministic space is closed under complementation. Therefore the complement of A ($\overline{A}$) is also in *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 $\overline{L_A}$. Note that $< i, x > \in \overline{L_A}$ iff simulation of M_i uses more than $S(| < i, x > |)$ tape cells or $< i, x > \in L(M_i, A^{\le |< i,x >|})$. 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
```

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 $M_{\overline{A}}$ accepts then continue $M_i$ in the NO state.
        ACCEPT iff $M_i$ accepts or simulation of $M_i$
        uses more then $S(| < i, x > |)$ 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 $< i, x > \notin L(M)$ iff $< i, x > \in L(\overline{M})$ iff $< i, x > \in L_A$. 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 x be a string such that the simulation of M_j uses $\leq S(| < j, x > |)$ tape cells. Then there exist at least one y in $Q(M_j, A, < j, x >) \cap A$ such that $|y| > |x|$.

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

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 proofs the corollary. ⊠

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 $\leq_{btt}^{logspace}$-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. ⊠

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 $x$
        $n := |x|$
        for all $n'$, $n_0 < n' < n$ do
                run $M_C$ on input $0^{n'}$
                **if** $Q(M_C, 0^{n'}, C) = x$ **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 $x \in C$ and $||L(M)|| = \aleph_0$. ☒

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

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

```
input $x$
        **if** $x \in B_0$ **then** output a fixed $y \notin A_1$
        **else** output $x$
**end**
```

Now we can apply corollary 3 again on A_1. This process can be repeated infinetely often and will generate the subsets $\{B_i\}_{i \in I\!N}$ as promised.☒

4 Differences between complete sets in NSPACE

At this point we have obtained some useful properties of $NSPACE(S(n))$-complete sets. For example every $\leq_m^{logspace}$-complete set in $NSPACE(S(n))$ has an FL-subset. Hence to construct a $\leq_{2-d}^{logspace}$-complete set, which is not $\leq_m^{logspace}$-complete it is sufficient to construct a set, which is $\leq_{2-d}^{logspace}$-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 $\leq_{btt}^{logspace}$-complete. We will construct the set D by stages. This is done with the help of a function $b : I\!N \mapsto I\!N$. 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 $\leq_{2-d}^{logspace}$-complete set D in $NSPACE(S(n))$, which is not $\leq_m^{logspace}$-complete.*

Proof: Recall that $\{T_i\}_{i \in I\!N}$ is an effective enumeration of nondeterministic logarithmic-space bounded Turing transducers. Let $C = L(M_c)$ be a standard $\leq_m^{logspace}$-complete set in $NSPACE(S(n))$. D is defined by the following construction:

```
Requirements : $\forall f \in FL[\overset{\infty}{\forall} n\ |f(0^n)| > n \Rightarrow \overset{\infty}{\exists}\ m\ f(0^m) \notin D]$
Construction:
    stage 0:
        $b(0) := 0$
        $D_0 := \emptyset$
    stage $n+1$
        $n' := b(n-1)$
        $y := \begin{cases} T_i(0^{n'}) & \text{if the simulation of } T_i \text{ uses} \leq S(n) \text{ tape cells} \\ \uparrow & \text{otherwise} \end{cases}$
        (here $i$ is the first element of the pair $< i, j >= n$.)
        if $y \neq \uparrow$ and $|y| > n'$ then
          $b(n) := |y|$
          $D_n := \{0x, 1x |\ b(n-1) < |x| + 1 \leq b(n) \text{ and } x \in C\} \backslash \{y\}$
        else
          $b(n) := b(n-1) + 1$
          $D_n := \{0x |\ |x| + 1 = b(n) \text{ and } x \in C\}$
        end-if
end construction
```

Consider the following machine M:

```
Let $a = 0$ or 1.
    input($ax$)
        if $M_c(x)$ rejects then reject
        else Simulate stage construction until $b(n) \geq |ax|$ is reached[2]
            if $b(n) = |ax|$ then
              if $y = \uparrow$ and $a = 0$ then accept else reject
            else if $ax = T_i(0^{n'})$ then reject else accept
    end
```

It is clear that $L(M)$ is in $NSPACE(S(n))$. Let g be a function in 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 FL-subset. Every $\leq_m^{logspace}$-complete set in $NSPACE(S(n))$has an FL-subset (corollary 2), so D is not $\leq_m^{logspace}$- complete for $NSPACE(S(n))$. Since for every $x \in C$ $1x$ or $0x$ is in D, C is $\leq_{btt}^{logspace}$-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 $|ax|$ it is the right n. Furthermore, the value of n cannot exceed $|ax|$.

$$\forall x.\ f(x) = << 0x, 1x >, \alpha > \text{ and}$$
$$\alpha(x, y) = x \vee y.$$

This proofs that D is $\leq_{2-d}^{logspace}$-complete. ⊠

Corollary 5

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

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

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

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

Requirements : $\forall f \in Fbtt[\overset{\infty}{\forall}\, n\ |f(0^n)| > n \Rightarrow \overset{\infty}{\exists}\, m f(0^m) \notin D]$
Construction :

```
stage 0
    $b(0) := 0$
    $D_0 := \emptyset$
stage $n+1$
    $n' := b(n-1)$
    $Y := \begin{cases} \{y|\ y \in \text{Ass}(T_i, 0^{n'}) \wedge |y| > n'\} & \text{if the simulation of } T_i \text{ uses } \leq S(n) \text{ tape cells} \\ \uparrow & \text{otherwise} \end{cases}$
    (here $i$ is the first element of the pair $< i, j >= n$.)
    if $Y \neq \uparrow$ and $0 < ||Y|| \leq n'/2$ then
        $b(n) :=$ length of longest element of $Y$
        $D_n := \{z|\ z \in C_x \text{ and } n' < |z| \leq b(n)\} \backslash Y$
    else
        $b(n) := n' + 2$
        $D_n := \{c(0, x)|\ n' < 2|x| \leq b(n) \text{ and } x \in C\}$
    end-if
end construction
```

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 is D not $\leq_{btt}^{logspace}$-complete. Since for ever $x \in C$ there is at least one element of C_x in D, C is $\leq_d^{logspace}$-reducible to D by the following tt-function f :

$$\forall x.f(x) = << c(0,x), \ldots, c(m,x) >, \alpha > \text{ and}$$
$$\alpha(a_1, \ldots, a_m) = a_1 \vee \ldots \vee a_m \text{ , where } m = |x|.$$

So D is $\leq_d^{logspace}$-complete.$\boxtimes$

Corollary 6 *There exists a $\leq_{tt}^{logspace}$-complete set D for $NSPACE(S(n))$,which is not $\leq_{btt}^{logspace}$-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 $\leq_T^{logspace}$-complete set and a $\leq_m^{R(n)-space}$ -complete set. We conjecture that this difference exists, i.e. there exists a set $D \in NSPACE(S(n))$, which is $\leq_m^{R(n)-space}$ -complete, but is not $\leq_T^{logspace}$-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, $\supsetneq P$, 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 polinomial Turing reductions exist. We conjecture that this is also true for logspace reductions in $NSPACE(S(n))$.

Bibliography

[1] Berman, L. & J. Hartmanis. *On isomorphisms and density of NP and other complete sets*. SIAM J. on Computing 1 (1977) 305–322.

[2] Cook, S. *An overview of computational complexity*. Communications of the ACM 26 (1983) 400–409.

[3] Garey, M. & D. Johnson. *Computer and Intractability, A guide to the theory of NP-completeness*. Freeman, San Fransisco (1979).

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

[5] Hartmanis, J. *Feasible Computations and Provable Complexity Properties*. CBMS-NSF regional conference series in applied mathematics (1978).

[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. Lynch & A.L. Selman. *A comparison of polynomial time reducibilities*. Theoretical Computer Science 1 (1975) 103–123.

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

[9] Watanabe, O. *A comparison of polynomial time completeness notions*. Theoretical Computer Science 54 (1987) 249–265.

The ITLI Prepublication Series

1986

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

1987

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

1988

Logic, Semantics and Philosophy of Language:

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

Mathematical Logic and Foundations:

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

Computation and Complexity Theory:

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

Other prepublications:

X-88-01 Marc Jumelet	On Solovay's Completeness Theorem

1989

Logic, Semantics and Philosophy of Language:

LP-89-01 Johan van Benthem	The Fine-Structure of Categorial Semantics

Mathematical Logic and Foundations:

ML-89-01 Dick de Jongh, Albert Visser	Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer	Extending the Lambda Calculus with Surjective Pairing is conservative

Computation and Complexity Theory:

CT-89-01 Michiel H.M. Smid	Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas	Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise Leen Torenvliet, Peter van Emde Boas	On Space efficient Solutions
CT-89-04 Harry Buhrman, Leen Torenvliet	A Comparison of Reductions on Nondeterministic Space

Other prepublications:

X-89-01 Marianne Kalsbeek	An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers	New Foundations. a Survey of Quine's Set Theory