

For Completeness, Sublogarithmic Space is No Space

Manindra Agrawal
Department of Computer Science
Indian Institute of Technology, Kanpur
Kanpur 208016, India
email: manindra@iitk.ac.in

Abstract

It is shown that for any class \mathcal{C} closed under linear-time reductions, the complete sets for \mathcal{C} under sublogarithmic reductions are also complete under 2DFA reductions, and thus are isomorphic under first-order reductions.

Keywords: Isomorphisms; Sublogarithmic reductions; Computational Complexity.

1 Introduction

Logarithmic space is a critical bound in space complexity. For the class DLOG ($= \text{DSPACE}(\log n)$) we do not have, till now, any non-trivial upper bound, while, on the other hand, it is not too difficult to exhibit languages in DLOG — $\text{DSPACE}(o(\log n))$ [16]. The reason for this is that the TMs working within sublogarithmic space cannot even record the the length of the input, and thus can be ‘fooled’ easily. In fact, when the space bound of a DTM is $o(\log \log n)$, the TM cannot recognize any non-regular language [20, 12]. So, there exists a *gap* between the classes $\text{DSPACE}(\log \log n)$ and $\text{DSPACE}(1)$ (the class of regular sets) in the sense that an intermediate space bound does not yield a different class (this result has been generalized to even alternating TMs [22, 14]). Similar gaps between time- and space-bounded complexity classes were exhibited in [24, 7]. Because of the above-mentioned limitation of sublogarithmic TMs, it also becomes possible to prove strong separation results for sublogarithmic classes, e.g., it has been shown that the space hierarchy for any sublogarithmic bound in $\Omega(\log \log n) \cap o(\log n)$ is infinite [25, 9, 18, 26, 10, 19]. This is in direct contrast with the result that for any space bound in $\Omega(\log n)$, the space hierarchy collapses to the second level [13, 23]. [17] contains an excellent survey of the results on the sublogarithmic space classes.

Any sublogarithmic space class can be viewed as the smallest many-one degree under the corresponding sublogarithmic space reducibility (such a reducibility may not be closed under composition, and so we have to define degrees carefully—see next section). So the question naturally arises: what can we say about the structure of other degrees, especially the *complete* degrees, under sublogarithmic space reducibilities? In particular, what is the relationship between the complete degrees under various sublogarithmic space reducibilities? An immediate observation here is that for many well known classes, e.g., P, NP, PSPACE, EXP etc., while there exist complete sets under logspace reductions, there are *no* complete sets under sublogarithmic reductions. This is because all these classes are closed under logspace reductions that can ‘blowup’ the input size by a polynomial, while the sublogarithmic space functions cannot even square the input size (see next section). However, for many classes that are not closed

under such reductions, e.g., $\text{DTIME}(n^k)$, $\text{NTIME}(n^k)$, $\text{DSpace}(n^k)$ ($k \geq 1$), E, NE etc., there do exist complete sets under sublogarithmic space reductions (see next section for examples).

The complete degrees under logspace reductions have been investigated for long (see [15] for a survey) and it is known that for well-known classes, e.g. NP, the complete degree under logspace reductions contains a wide variety of languages [11]. On the other extreme, the complete degrees under reductions computed by DTMs that have no space (referred as 2DFA reductions in [21]), have also been investigated [6, 8, 4]. It can be easily shown that the complete degree of any reasonable class under 2DFA reductions is properly contained in the complete degree of the class under logspace reductions. However, there has been no work on complete degrees under sublogarithmic space reductions so far (although much work has taken place on a different class of sublogarithmic reductions: AC^0 reductions [3, 2]). The question of interest for such degrees would be to compare these with the degrees complete under logspace and 2DFA reductions. In particular, does there exist a gap—as between $\text{DSpace}(\log \log n)$ and $\text{DSpace}(1)$ —between some of these complete degrees? It is useful to note here that a similar gap is known to exist between sets complete under AC^0 and NC^0 reductions [3].

In this paper, we show that such a gap indeed exists, and in fact, extends *all the way* from complete degrees under 2DFA reductions to complete degrees under logspace reductions. In other words, we show that for any class \mathcal{C} closed under linear-time reductions, and for any sublogarithmic space bounded reducibility, any complete set for \mathcal{C} under the reducibility is also complete under 2DFA reducibility. The result holds for all the classes $\text{DTIME}(n^k)$, $\text{NTIME}(n^k)$, $\text{DSpace}(n^k)$, E, NE etc. as all these are closed under linear-time reductions.

2 Preliminaries

All the strings that we consider are over $\Sigma = \{0, 1\}$. 1^+ denotes the set $1^* - \{\epsilon\}$. Similarly, 0^+ denotes the set $0^* - \{\epsilon\}$. For a string x , $x[i]$ denotes the i^{th} bit of x for $1 \leq i \leq |x|$.

Our model of computation is Turing Machines with a read-only input tape, a write-only output tape, and a read-write work tape.

A 2DFA TM is a DTM that uses no space on any input. We shall be interested in the total functions computed by these TMs. A 2DFA function is a total function computed by a 2DFA TM.

A sublogarithmic space TM is one that uses $o(\log n)$ space on every input of size n . It is straightforward to see that the number of different configurations of such TMs, on any input of size n , are $O(n \cdot 2^{o(\log n)}) = o(n^2)$. So, any such TM that always halts, works only for time $o(n^2)$. A sublogarithmic space function is a total function computed by a sublogarithmic space DTM.

We shall be interested in complete sets for different classes under sublogarithmic space DTMs. The above bound on the time of these TMs implies that several classes, e.g., P, NP do not have *any* complete set under these reductions. The reason is easy to see. Let $\text{NP}^k = \text{NTIME}(n^k)$. If there is a complete set for NP under sublogarithmic space reductions, say in NP^j , then $\text{NP} = \text{NP}^{2j}$ violating the nondeterministic-time hierarchy theorem. Similarly one can argue about P, PSPACE, EXP etc. However, the classes $\text{P}^k = \text{DTIME}(n^k)$, NP^k , $\text{DSpace}(n^k)$, E, NE etc. have complete sets under even 2DFA reductions, and thus also under sublogarithmic space reductions. An example is the set

$$\{\langle i, x, 1^t \rangle \mid \text{NTM } M_i \text{ accepts } x \text{ in } t \text{ steps}\}.$$

This set is complete for NP^1 under 2DFA reductions. In [21], several ‘natural’ problems are shown to be complete for E under 2DFA reductions.

For any resource bound r , we say that $A \leq_m^r B$ via f if the set A reduces to B via f and there is a TM computing f that works within the resource bound of r . We say $A \leq_m^{2dfa} B$ when

f is a 2DFA function and $A \leq_m^{sublog} B$ when f is a sublogarithmic space function. For any class \mathcal{C} , a set A is \leq_m^r -hard set for \mathcal{C} if for every set $B \in \mathcal{C}$, $B \leq_m^r A$. The set A is \leq_m^r -complete for \mathcal{C} if A is \leq_m^r -hard for \mathcal{C} and $A \in \mathcal{C}$. The \leq_m^r -degrees are the strongly connected components of the relation \leq_m^r . The \leq_m^r -complete degree of \mathcal{C} is the \leq_m^r -degree of the \leq_m^r -complete sets for \mathcal{C} .

3 The result

The structure of complete degrees has been investigated for long (see [15] for a survey of results on polynomial-time and logspace complete degrees). For some of the reducibility classes weaker than logspace, the structure of complete degrees under them can be completely described [5, 1, 3]. The structure of \leq_m^{2dfa} -complete degrees has also been investigated [6, 8, 4]. In [4], it was shown that for any class closed under linear-time reductions (the results there were shown for classes closed under *log-lin* reductions but their proof works for classes closed under linear-time reductions as well), the \leq_m^{2dfa} -complete sets of the class are also complete under 2DFA reductions that are one-one and size-increasing, and further, all such sets are first-order isomorphic (see [5] for a definition of first-order isomorphism) to each other. Here, we show that for any class closed under linear-time reductions, the \leq_m^{sublog} -complete sets are also \leq_m^{2dfa} -complete, and therefore, are complete under 2DFA reductions that are one-one and size-increasing as well as first-order-isomorphic to each other.

For proving our result, we shall make use of the following technical lemma proved in [17] (in fact, the lemma is proved there even for alternating sublogarithmic space TMs while we use it only for DTMs). The proof of the lemma uses essentially a generalization of the $n \mapsto n + n!$ -technique developed by Hartmanis, Sterns, and Lewis for sublogarithmic space TMs [20].

For any string w , let w^k denote the string $\underbrace{ww \cdots w}_{k \text{ times}}$.

Lemma 3.1 *Let M be a sublogarithmic space DTM. Then, there is a constant N such that for every $n \geq N$, for all strings z_1, z_2, w , and for every $l \geq 0$, the space used by M on the input $z_1 w^{n+l \cdot n!} z_2$ is the same as the space used on the input $z_1 w^n z_2$.*

The above lemma asserts that the TM M is ‘fooled’ by the string $z_1 w^{n+l \cdot n!} z_2$ —it cannot differentiate the string from $z_1 w^n z_2$.

In our main result below, we shall compose two functions, one a sublogarithmic space function and the other a 2DFA function. We shall require their composition to be a 2DFA function. Clearly, this does not hold in general. So, we define the conditions under which the composition is a 2DFA function. A 2DFA function g is a *simple* 2DFA function if there exist two strings b_0 and b_1 and a set of strings $\{c_0, c_1, \dots, c_k\}$ for a fixed k , such that for all x , $g(x) = b_{x[1]} b_{x[2]} \cdots b_{x[|x|]} c$ where $c \in \{c_0, \dots, c_k\}$.

Lemma 3.2 *Let f be a sublogarithmic space function computed by the TM M , and g be a simple 2DFA function. If the TM M , on any string in the range of g , works within a constant space, then the function $f \circ g$ is a 2DFA function.*

Proof. The following TM computes $f \circ g$:

On input x , first compute the string c by simulating the 2DFA TM computing $g(x)$, where $g(x) = b_{x[1]} b_{x[2]} \cdots b_{x[|x|]} c$ for some $c \in \{c_0, \dots, c_k\}$. Now, start the simulation of M on $g(x)$ by writing the string $b_{x[1]}$ on the work tape. As and when M needs more bits of $g(x)$, look up the corresponding bit of x and write the appropriate string b_0 or b_1 (write c if M needs the last $|c|$ bits of $g(x)$) on the work tape (overwriting the earlier string). Output any bit that M outputs.

The above TM needs only a constant amount of workspace as M works on $g(x)$, for any x , within a constant space. A TM that needs a constant amount of workspace can be replaced by a TM that needs no workspace and does all the computation in its states. And so, the function $f \circ g$ is 2DFA function. \blacksquare

Theorem 3.3 *For any class \mathcal{C} closed under linear-time reductions, the \leq_m^{sublog} -hard sets for \mathcal{C} are also \leq_m^{2dfa} -hard.*

Proof. Let A be a \leq_m^{sublog} -hard set for \mathcal{C} . Let $B \in \mathcal{C}$, $B \neq \emptyset$, Σ^* . We shall exhibit a 2DFA reduction of B to A . Define a set D as accepted by the following procedure.

Input z . If z does not begin with a 1 and end with a 0, then reject. Otherwise, let $z = w_1 w_2 \cdots w_p$ such that $w_i \in 1^+ 0^+$ for $1 \leq i \leq p$. For $1 \leq i \leq p$, let $a_i = 1$ if w_i has more 1's than 0's, 0 if w_i has more 0's than 1's, ϵ otherwise. Define $x = a_1 a_2 \cdots a_p$. Accept iff $x \in B$.

The set D reduces to B via a linear-time reduction: the TM computing the reduction first scans the input to make sure that it begins with a 1 and ends with a 0. If not, then it outputs a fixed string in \bar{B} . Otherwise, it starts scanning the input from left keeping a count of the number of ones and zeroes in w_i , and outputs a_i accordingly.

So, $D \in \mathcal{C}$. Let $D \leq_m^{\text{sublog}} A$ via f and let f be computed by the sublogarithmic space TM M . We now construct a reduction, g , of B to D such that the TM M , on the range of g , requires only a constant space. Further, the function g is a simple 2DFA function. Therefore, by Lemma 3.2, function $f \circ g$ —reducing B to A —would be a 2DFA function, proving the theorem. Before giving the construction of g , we prove two claims on the behavior of the TM M on certain strings.

Let S be a space bound in $o(\log n)$ such that for every x , the space used by M on input x is bounded by $S(|x|)$. Let N be the constant for the TM M as given by Lemma 3.1. Let $u = 1^N 0^N$ and

$$R_0 = \{u^{N+l \cdot N!} \mid l \geq 0\}.$$

Our first claim follows directly from Lemma 3.1.

Claim 3.3.1 *For any string $y \in R_0$, the space used by the TM M on the input y , is bounded by $S(|u^N|) = S(2N^2)$.*

Proof of Claim 3.3.1. Let $z_1 = z_2 = \epsilon$, $w = u$, $n = N$, and apply Lemma 3.1 with these parameters. \square

The above claim shows that if we make the function g map strings to the set R_0 , then $f \circ g$ becomes a 2DFA function (by Lemma 3.2). However, the set R_0 is sparse and so g has to map exponentially many strings to a single string to achieve this. As this does not appear feasible, we define a dense set for which the same property holds. Define,

$$R = \{v_1 v_2 \cdots v_{N+l \cdot N!} \mid l \geq 0 \text{ and for } 1 \leq i \leq N + l \cdot N!, v_i \in \{1^{N+N!} 0^N, 1^N 0^{N+N!}, 1^N 0^N\}\}.$$

This set is clearly dense. Now we show that,

Claim 3.3.2 *For any string y in R , the space used by the TM M on the input y , is bounded by $S(2N^2)$.*

Proof of Claim 3.3.2. Let $y \in R$ and $y = v_1 v_2 \cdots v_{N+l \cdot N!}$ as above for some $l \geq 0$. Let $y_i = v_1 v_2 \cdots v_i u^{N+l \cdot N! - i}$ for $0 \leq i \leq N + l \cdot N!$. We show that the space used by the TM M on the input y_i , for any $1 \leq i \leq N + l \cdot N!$, is bounded by $S(2N^2)$. The proof is by induction on i .

Induction hypothesis : The space used by the TM M on the input y_i is bounded by $S(2N^2)$.

Base step ($i = 0$) : Follows from Claim 3.3.1 since $y_0 \in R_0$.

Induction step : Suppose the hypothesis holds for some $i < N + l \cdot N!$. Consider $y_{i+1} = v_1 v_2 \cdots v_i v_{i+1} u^{N+l \cdot N! - i - 1}$. There are three cases: first, when $v_{i+1} = u$. In this case, the hypothesis trivially holds for $i + 1$ since $y_{i+1} = y_i$. The second case is when $v_{i+1} = 1^{N+N!} 0^N$. For this case, let $z_1 = v_1 v_2 \cdots v_i$, $w = 1$, $z_2 = 0^N u^{N+l \cdot N! - i - 1}$, $n = N$, and $l = 1$. Applying Lemma 3.1 with these parameters, we get that the TM M , on input y_{i+1} , uses at most as much space as on input y_i . Thus, by the induction hypothesis, the space used by M on input y_{i+1} is at most $S(2N^2)$. The third case is when $v_{i+1} = 1^N 0^{N+N!}$. For this case, let $z_1 = v_1 v_2 \cdots v_i 1^N$, $w = 0$, $z_2 = u^{N+l \cdot N! - i - 1}$, $n = N$, and $l = 1$. Applying Lemma 3.1 with these parameters, we again have that the space used by M on y_{i+1} is bounded by $S(2N^2)$.

Since $y_{N+l \cdot N!} = y$, the claim follows. \square

Now we construct the function g whose range is a subset of R , as promised. It is computed by the following procedure.

On input x , let $|x| = m$. Let $b_0 = 1^N 0^{N+N!}$ and $b_1 = 1^{N+N!} 0^N$. Let $c_i = (1^N 0^N)^i$ for $0 \leq i \leq k = N! - 1$ (assuming $s^0 = \epsilon$ for any string s). Scan the bits of x from left to right, and for every bit, output b_0 if the bit is zero, b_1 otherwise. While scanning, also check if $m - N$ is divisible by $N!$ (this can be done by counting the first N bits, and after that counting every $N!$ successive bits). If not, then let $r = (m - N) \pmod{N!}$ (r will be automatically computed by doing the above counting). Clearly, $1 \leq r \leq N! - 1$. Output $c_{N! - r}$.

Function g is a simple 2DFA function since $g(x) = b_{x[1]} \cdots b_{x[m]} c_i$ for some $0 \leq i \leq N! - 1$, and the TM computing $g(x)$ requires at most $O(N!)$ space—a constant. Therefore, by the above claim and Lemma 3.2, $f \circ g$ is a 2DFA function. We now show that g is a reduction of B to D . Firstly, notice that $g(x) \in R$ for every x . We also have that $g(x) = b_{x[1]} \cdots b_{x[m]} c_i$ for some $0 \leq i \leq N! - 1$. Moreover, for $1 \leq j \leq m$, $b_{x[j]} \in 1^+ 0^+$, and if $x[j] = 1$ then $b_{x[j]}$ has more 1's, otherwise more 0's. String c_i is u^i for $u = 1^N 0^N$. Therefore, it follows that the procedure accepting D given above, on input $z = g(x)$, will accept iff $x \in B$.

Thus, $f \circ g$ is a 2DFA reduction of B to A . \blacksquare

Corollary 3.4 *For any class \mathcal{C} closed under linear-time reductions, the \leq_m^{sublog} -complete sets for \mathcal{C} are also $\leq_m^{2\text{dfa}}$ -complete.*

By a result in [4], it follows that,

Corollary 3.5 *For any class \mathcal{C} closed under linear-time reductions, the \leq_m^{sublog} -complete sets for \mathcal{C} are also complete under one-one, size-increasing, 2DFA reductions and further, are first-order-isomorphic to each other.*

The result of [4] in fact shows that the \leq_m^{2dfa} -complete sets are complete under 2DFA reductions that are very similar to simple DFA functions that we have defined. Similar to Lemma 3.2, it can be shown that the composition of a sublogarithmic space function with such a DFA function remains a sublogarithmic space function. Therefore,

Corollary 3.6 *For any class \mathcal{C} closed under linear-time reductions, sets in the \leq_m^{sublog} -complete degree of \mathcal{C} are complete for \mathcal{C} under one-one, size-increasing, 2DFA reductions and further, are first-order-isomorphic to each other.*

References

- [1] M. Agrawal. On the isomorphism problem for weak reducibilities. In *Proceedings of the Structure in Complexity Theory Conference*, pages 338–355, 1994.
- [2] M. Agrawal, E. Allender, R. Impagliazzio, T. Pitassi, and S. Rudich. Reducing the complexity of reductions. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 730–738, 1997.
- [3] M. Agrawal, E. Allender, and S. Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *J. Comput. Sys. Sci.*, 57:127–143, 1998.
- [4] M. Agrawal and S. Venkatesh. The isomorphism conjecture for 2-DFA reductions. *Intl. Jl. on Foundations of Computer Science*, 7(4):339–352, 1996.
- [5] E. Allender, J. Balcázar, and N. Immerman. A first-order isomorphism theorem. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, 1993.
- [6] E. W. Allender. Isomorphisms and 1-L reductions. *J. Comput. Sys. Sci.*, 36(6):336–350, 1988.
- [7] A. Borodin. Computational complexity and the existence of complexity gaps. *J. ACM*, 19:158–174, 1972.
- [8] K. Ganesan and S. Homer. Complete problems and strong polynomial reducibilities. *SIAM Journal on Computing*, 21:733–742, 1992.
- [9] V. Geffert. Sublogarithmic σ_2 -space is not closed under complement and other separation results. *Theoretical Informatics and Applications*, 27:349–366, 1993.
- [10] V. Geffert. A hierarchy that does not collapse: alternations in low level space. *Theoretical Informatics and Applications*, 28(5):465–512, 1994.
- [11] J. Hartmanis. On log-tape isomorphisms of complete sets. *Theoretical Computer Science*, pages 273–286, 1978.
- [12] J. Hopcroft and J. Ullman. Some results on tape-bounded computation. *J. ACM*, 16:168–177, 1969.
- [13] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17:935–938, 1988.
- [14] K. Iwama. $\text{Aspace}(o(\log \log n))$ is regular. *SIAM Journal on Computing*, 22:136–146, 1993.
- [15] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A. Selman, editor, *Complexity Theory Retrospective*, pages 108–146. Springer-Verlag, 1988.

- [16] P. Lewis, R. Stearns, and J. Hartmanis. Memory bounds for the recognition of context free and context sensitive languages. In *Proceedings of the IEEE Conference on Switching Circuit Theory and Logical Design*, pages 191–202, 1965.
- [17] M. Liśkiewicz and R. Reischuk. The complexity world below logarithmic space. In *Proceedings of the Structure in Complexity Theory Conference*, pages 64–78, 1993.
- [18] M. Liśkiewicz and R. Reischuk. Separating the lower levels of the sublogarithmic space hierarchy. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 6–27, 1993.
- [19] M. Liśkiewicz and R. Reischuk. The sublogarithmic space world. Technical Report 048-93, ICSI Berkeley, 1993.
- [20] R. Stearns, J. Hartmanis, and P. Lewis. Hierarchies of memory limited computations. In *Proceedings of the IEEE Conference on Switching Circuit Theory and Logical Design*, pages 179–190, 1965.
- [21] L. J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, Massachusetts Institute of Technology, 1974.
- [22] I. Sudborough. Efficient algorithms for path system problems and applications to alternating and time-space complexity classes. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science*, pages 62–67, 1980.
- [23] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.
- [24] B. A. Trakhtenbrot. Turing computations with logarithmic delay. *Algebra i Logika*, 3:33–48, 1964.
- [25] B. von Braunmühl. Alternation for two-way machines with sublogarithmic space. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 5–15, 1993.
- [26] B. von Braunmühl, R. Gengler, and R. Rettinger. The alternation hierarchy for sublogarithmic space is infinite. *Computational Complexity*, 3:207–230, 1993.