ELSEVIER

Information Processing Letters 82 (2002) 321–325

Information Processing Letters

www.elsevier.com/locate/ipl

For completeness, sublogarithmic space is no space

Manindra Agrawal

Department of Computer Science, Indian Institute of Technology, Kanpur 208016, India

Received 10 December 2000; received in revised form 23 August 2001
Communicated by L.A. Hemaspaandra

Abstract

It is shown that for any class C closed under linear-time reductions, the complete sets for C under sublogarithmic reductions are also complete under 2DFA reductions, and thus are isomorphic under first-order reductions. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Isomorphisms; Completeness; Sublogarithmic reductions; Computational complexity

1. Introduction

Logarithmic space is a critical bound in space complexity. The classes above, and including, DLOG ($=$ DSPACE$(\log n)$) exhibit very different properties than those below it. For example, all non-deterministic $s(n)$ space classes are closed under complement for $s(n) = \Omega(\log n)$ [13,22] while they are not closed for $s(n)$ in $\Omega(\log\log n) \cap o(\log n)$ [24,9,17,25,10, 18]. In fact, when the space bound of a DTM is $o(\log\log n)$, the TM cannot recognize any non-regular language [19,12]. So, there exists a *gap* between the classes DSPACE$(\log\log n)$ and 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 [21, 14]). Similar gaps between time- and space-bounded complexity classes were exhibited in [23,7]. The main reason for this difference in properties is that the TMs working within sublogarithmic space cannot even record the length of the input, and thus can be 'fooled' easily. [16] 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 'blow up' 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., DTIME(n^k), NTIME(n^k), DSPACE(n^k) $(k \geqslant 1)$, E, NE, etc., there

E-mail address: manindra@iitk.ac.in (M. Agrawal).

0020-0190/01/$ – see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0020-0190(01)00296-4

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 [20]), have also been investigated [6, 8,4]. 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 $\mathrm{DSPACE}(\log\log n)$ and 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 C closed under linear-time reductions, and for any sublogarithmic space bounded reducibility, any complete set for C under the reducibility is also complete under 2DFA reducibility. The result holds for all the classes $\mathrm{DTIME}(n^k)$, $\mathrm{NTIME}(n^k)$, $\mathrm{DSPACE}(n^k)$, 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 $1^* - \{\varepsilon\}$. Similarly, 0^+ denotes the set $0^* - \{\varepsilon\}$. For a string x, $x[i]$ denotes the ith bit of x for $1 \leqslant i \leqslant |x|$.

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 $\mathrm{o}(\log n)$ 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 $\mathrm{O}(n \cdot 2^{\mathrm{o}(\log n)}) = \mathrm{o}(n^2)$. So, any such TM that always halts, works only for time $\mathrm{o}(n^2)$. 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 $\mathrm{NP}^k = \mathrm{NTIME}(n^k)$. If there is a complete set for NP under sublogarithmic space reductions, say in NP^j, then $\mathrm{NP} = \mathrm{NP}^{2j}$ violating the nondeterministic-time hierarchy theorem. Similarly one can argue about P, PSPACE, EXP, etc. However, the classes $\mathrm{P}^k = \mathrm{DTIME}(n^k)$, NP^k, $\mathrm{DSPACE}(n^k)$, 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 [20], several 'natural' problems are shown to be complete for E under 2DFA reductions.

For any resource bound r, we say that $A \leqslant_m^r B$ 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 $A \leqslant_m^{\mathrm{2dfa}} B$ when f is a 2DFA function and $A \leqslant_m^{\mathrm{sublog}} B$ when f is a sublogarithmic space function. For any class C, a set A is a $\leqslant_m^r$-hard set for C if for every set $B \in C$, $B \leqslant_m^r A$. The set A is $\leqslant_m^r$-complete for C if A is $\leqslant_m^r$-hard for C and $A \in C$. The $\leqslant_m^r$-degrees are the strongly connected components of the relation $\leqslant_m^r$. The $\leqslant_m^r$-complete degree of C is the $\leqslant_m^r$-degree of the $\leqslant_m^r$-complete sets for 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 $\leqslant_m^{\mathrm{2dfa}}$-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 $\leqslant_m^{\text{2dfa}}$-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 $\leqslant_m^{\text{sublog}}$-complete sets are also $\leqslant_m^{\text{2dfa}}$-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 [16] (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 [19].

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

Lemma 1. *Let M be a sublogarithmic space DTM. Then, there is a constant N such that for every $n \geqslant N$, for all strings z_1, z_2, w, and for every $\ell \geqslant 0$, the space used by M on the input $z_1 w^{n+\ell\cdot n!} z_2$ 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 $z_1 w^{n+\ell\cdot n!} z_2$—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 $\{c_0, c_1, \ldots, c_k\}$ for a fixed k, such that for all x, $g(x) = b_{x[1]} b_{x[2]} \cdots b_{x[|x|]} c$ where $c \in \{c_0, \ldots, c_k\}$.

Lemma 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 $c \in \{c_0, \ldots, c_k\}$. Now, start the simulation of M on $g(x)$ by writing the string $b_{x[1]}$ on the work tape and placing the input head on the first bit of x. During the simulation, when M tries to read an adjacent block of bits of $g(x)$, move the input head to the corresponding adjacent 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 and the input head of the TM keeps track of the current block being read by M. 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. □

Theorem 3. *For any class C closed under linear-time reductions, the $\leqslant_m^{\text{sublog}}$-hard sets for C are also $\leqslant_m^{\text{2dfa}}$-hard.*

Proof. Let A be a $\leqslant_m^{\text{sublog}}$-hard set for C. Let $B \in C$, $B \neq \emptyset$, 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 $1 \leqslant i \leqslant p$. For $1 \leqslant i \leqslant p$, 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 $\overline{B}$. 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, $D \in C$. Let $D \leqslant_m^{\text{sublog}} A$ 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 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 $\mathrm{o}(\log n)$ 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 1. For the remainder of the proof, fix u to be the string $1^N 0^N$. Define

$$R_0 = \{u^{N+\ell \cdot N!} \mid \ell \geqslant 0\}.$$

Our first claim follows directly from Lemma 1.

Claim 4. *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. Let $z_1 = z_2 = \varepsilon$, $w = u$, $n = N$, and apply Lemma 1 with these parameters. □

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 2). However this is not possible since the set R_0 is sparse while the range of g needs to be a dense set. So, we now define a dense set for which the same property holds:

$$\begin{aligned} R = \bigl\{&v_1 v_2 \cdots v_{N+t\cdot N!} \mid \\ &t \geqslant 0 \text{ and for } 1 \leqslant i \leqslant N + t \cdot N!, \\ &v_i \in \{1^{N+N!}0^N, 1^N 0^{N+N!}, 1^N 0^N\}\bigr\}. \end{aligned}$$

This set is clearly dense. Now we show that

Claim 5. *For any string y in R, the space used by the TM M on the input y, is bounded by $S(2N^2)$.*

Proof. Let $y \in R$ and $y = v_1 v_2 \cdots v_{N+t\cdot N!}$ as above for some $t \geqslant 0$. Let $y_i = v_1 v_2 \cdots v_i u^{N+t\cdot N!-i}$ for $0 \leqslant i \leqslant N + t \cdot N!$. We show that the space used by the TM M on the input y_i, for any $1 \leqslant i \leqslant N + t \cdot N!$, 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 4 since $y_0 \in R_0$.

Induction step: Suppose the hypothesis holds for some $i < N + t \cdot N!$. Consider $y_{i+1} = v_1 v_2 \cdots v_i v_{i+1} \cdot u^{N+t\cdot N!-i-1}$. There are three cases:

(1) $v_{i+1} = u$. In this case, the hypothesis trivially holds for $i + 1$ since $y_{i+1} = y_i$.

(2) $v_{i+1} = 1^{N+N!}0^N$. For this case, let $z_1 = v_1 v_2 \cdots v_i$, $w = 1$, $z_2 = 0^N u^{N+t\cdot N!-i-1}$, $n = N$, and $\ell = 1$. Applying Lemma 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 $S(2N^2)$.

(3) $v_{i+1} = 1^N 0^{N+N!}$. For this case, let $z_1 = v_1 v_2 \cdots v_i 1^N$, $w = 0$, $z_2 = u^{N+t\cdot N!-i-1}$, $n = N$, and $\ell = 1$. Applying Lemma 1 with these parameters, we again have that the space used by M on y_{i+1} is bounded by $S(2N^2)$.

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

Now we define the function g whose range is a subset of R, as promised. Let

$$g(x) = b_{x[1]}, b_{x[2]} \cdots b_{x[|x|]} d_{|x|},$$

where $b_0 = 1^N 0^{N+N!}$, $b_1 = 1^{N+N!}0^N$, and $d_{|x|} = u^{N!-r}$ with $r = (|x| - N) \pmod{N!}$.

Note that $d_{|x|}$ takes values from the set $\{u^i \mid 1 \leqslant i \leqslant N!\}$, a finite set. Furthermore, function g can be computed by a 2DFA TM that scans the input from left-to-right, and outputs $b_{x[i]}$ for bit $x[i]$ for every i. While scanning, the TM computes r (this can be done by counting the first N bits, and after that counting every $N!$ successive bits), and finally outputs $d_{|x|}$ using the computed value of r. The space required by this TM is $\mathrm{O}(N!)$—a constant. Therefore, function g is a simple 2DFA function.

By the above claim, the TM computing f works within constant space on any string in the range of function g, and therefore, by Lemma 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 $g(x) = b_{x[1]} \cdots b_{x[|x|]} u^i$ for some $1 \leqslant i \leqslant N!$. Moreover, for $1 \leqslant j \leqslant |x|$, $b_{x[j]} \in 1^+0^+$, and if $x[j] = 1$ then $b_{x[j]}$ has more ones, otherwise more zeroes. String u, whose (possibly) multiple copies are appended at the end has equal number of ones and zeroes. 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. □

Corollary 6. *For any class C closed under linear-time reductions, the $\leqslant_m^{\text{sublog}}$-complete sets for C are also $\leqslant_m^{\text{2dfa}}$-complete.*

By a result in [4], it follows that

Corollary 7. *For any class C closed under linear-time reductions, the $\leqslant_m^{\text{sublog}}$-complete sets for 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 $\leqslant_m^{\text{2dfa}}$-complete sets are complete under 2DFA reductions that are very similar to simple DFA functions that we have defined. Similar to Lemma 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 8. *For any class C closed under linear-time reductions, sets in the $\leqslant_m^{\text{sublog}}$-complete degree of C are complete for 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, 1994, pp. 338–355.

[2] M. Agrawal, E. Allender, R. Impagliazzio, T. Pitassi, S. Rudich, Reducing the complexity of reductions, in: Proceedings of Annual ACM Symposium on the Theory of Computing, 1997, pp. 730–738.

[3] M. Agrawal, E. Allender, S. Rudich, Reductions in circuit complexity: An isomorphism theorem and a gap theorem, J. Comput. System Sci. 57 (1998) 127–143.

[4] M. Agrawal, S. Venkatesh, The isomorphism conjecture for 2-DFA reductions, Internat. J. on Foundations Comput. Sci. 7 (4) (1996) 339–352.

[5] E. Allender, J. Balcázar, 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. System Sci. 36 (6) (1988) 336–350.

[7] A. Borodin, Computational complexity and the existence of complexity gaps, J. ACM 19 (1972) 158–174.

[8] K. Ganesan, S. Homer, Complete problems and strong polynomial reducibilities, SIAM J. Comput. 21 (1992) 733–742.

[9] V. Geffert, Sublogarithmic σ_2-space is not closed under complement and other separation results, Theoret. Inform. Appl. 27 (1993) 349–366.

[10] V. Geffert, A hierarchy that does not collapse: alternations in low level space, Theoret. Inform. Appl. 28 (5) (1994) 465–512.

[11] J. Hartmanis, On log-tape isomorphisms of complete sets, Theoret. Comput. Sci. 7 (1978) 273–286.

[12] J. Hopcroft, J. Ullman, Some results on tape-bounded computation, J. ACM 16 (1969) 168–177.

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

[14] K. Iwama, Aspace $(o(\log\log n))$ is regular, SIAM J. Comput. 22 (1993) 136–146.

[15] S. Kurtz, S. Mahaney, J. Royer, The structure of complete degrees, in: A. Selman (Ed.), Complexity Theory Retrospective, Springer, Berlin, 1988, pp. 108–146.

[16] M. Liśkiewicz, R. Reischuk, The complexity world below logarithmic space, in: Proceedings of the Structure in Complexity Theory Conference, 1993, pp. 64–78.

[17] M. Liśkiewicz, R. Reischuk, Separating the lower levels of the sublogarithmic space hierarchy, in: Proceedings of the Symposium on Theoretical Aspects of Computer Science, 1993, pp. 6–27.

[18] M. Liśkiewicz, R. Reischuk, The sublogarithmic space world, Technical Report 048-93, ICSI, Berkeley, CA, 1993.

[19] R. Stearns, J. Hartmanis, P. Lewis, Hierarchies of memory limited computations, in: Proceedings of the IEEE Conference on Switching Circuit Theory and Logical Design, 1965, pp. 179–190.

[20] L.J. Stockmeyer, The complexity of decision problems in automata theory and logic, PhD Thesis, MIT, Cambridge, MA, 1974.

[21] 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, 1980, pp. 62–67.

[22] R. Szelepcsényi, The method of forced enumeration for nondeterministic automata, Acta Inform. 26 (1988) 279–284.

[23] B.A. Trakhtenbrot, Turing computations with logarithmic delay, Algebra i Logika 3 (1964) 33–48.

[24] B. von Braunmühl, Alternation for two-way machines with sublogarithmic space, in: Proceedings of the Symposium on Theoretical Aspects of Computer Science, 1993, pp. 5–15.

[25] B. von Braunmühl, R. Gengler, R. Rettinger, The alternation hierarchy for sublogarithmic space is infinite, Comput. Complexity 3 (1993) 207–230.