Strong Optimal Lower Bounds for Turing Machines that Accept Nonregular Languages*

Alberto Bertoni, Carlo Mereghetti, and Giovanni Pighizzini

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
via Comelico, 39 – 20135 Milano – Italy

Abstract. In this paper, simultaneous lower bounds on space and input head reversals for deterministic, nondeterministic and alternating Turing machines accepting *nonregular* languages are studied.
Three notions of space complexity, namely strong, middle, and weak, are considered; moreover, another notion called accept, is introduced. For all cases we obtain tight lower bounds. In particular, we prove that while in the deterministic and nondeterministic case these bounds are "strongly" optimal—in the sense that we exhibit a nonregular language over a *unary* alphabet exactly fitting them—in the alternating case optimal lower bounds for tally languages turn out to be higher than those for arbitrary languages.

1 Introduction

Gap results for space complexity of nonregular languages are well known. In the pioneering work by Stearns, Hartmanis, and Lewis [SHL65] deterministic Turing machines accepting nonregular languages by halting on any input of length n within $s(n)$ work space are studied proving that, for one–way machines, $s(n) \geq d \cdot \log n$ must hold for a suitable constant $d > 0$ and infinitely many n ($s(n) = \Omega_\infty(\log n)$ for short) while, for two–way devices, $s(n) = \Omega_\infty(\log\log n)$. Hopcroft and Ullman [HU69] generalize such lower bounds to nondeterministic machines further dropping the halting property. Optimality of these bounds is then guaranteed by several languages.

Here, a key point which is worth noting is that the $s(n)$ space bound holds for *all* the computations on inputs of length n (strong space). Different notions of space complexity have been introduced: closer to the idea of nondeterminism, Alberts evaluates $s(n)$ on the basis of the existence of a $s(n)$ space bounded accepting computation on any accepted input of length n (weak space, [Alb85]) while Szepietowski proposes an intermediate notion by computing $s(n)$ on all the computations on accepted inputs of length n (middle space, [Sze88]).

We remark that all these space notions are easily seen to be equivalent when dealing with fully space constructible bounds. Since Geffert [Gef91] shows that

* This work was supported in part by the ESPRIT Basic Research Action No. 6317: "Algebraic and Syntactic Methods in Computer Science (ASMICS 2)" and by the Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST).

there are no monotone increasing sublogarithmic fully space constructible functions, this leads us to search possible differences among above notions of space complexity by considering sublogarithmic bounds. In this sense, our paper continues the line of investigations on the complexity world below logspace (see, for instance, [Gef93, BMP94a, DH94, LR94, Sze94]) where a convenient setting to work into is certainly represented by the analysis of minimal resource requirements for nonregular languages acceptance.

We introduce a new notion of space complexity, called accept, where space bound $s(n)$ is computed on *all the accepting computations* on inputs of length n. Accept space concept turns out to be very useful especially when related to the other computational resource we will be concerned with, namely the number $i(n)$ of input head reversals.

We prove that:

1. one–way nondeterministic Turing machines accepting nonregular languages in accept $s(n)$ space must satisfy $s(n) = \Omega_\infty(\log n)$;
2. $s(n) \cdot i(n) = \Omega_\infty(\log n)$ for deterministic Turing machines using any of the four notions of space, and for nondeterministic machines as well except for weak case where $s(n) \cdot i(n) = \Omega_\infty(\log \log n)$.

We design a new algorithm that accepts the complement of Alt–Mehlhorn's language $\mathcal{L}$ [AM75] in weak nondeterministic $O(\log \log n)$ space by one input scan. This allows to prove that all the considered lower bounds are optimal with respect to either general or tally languages.

For alternating Turing machines, we obtain slightly different results. In particular, we get a $\log \log n$ lower bound for the accept case. Moreover, we show that $s(n) \cdot i(n) = \Omega_\infty(\log \log n)$ with the only exception of the strong case where $s(n) \cdot i(n) = \Omega_\infty(\log n)$.

Lower bounds for the alternating case are optimal with respect to general languages, but the study of their optimality for tally languages permits us to single out important differences from deterministic and nondeterministic cases. First of all, we show how to recognize $\mathcal{L}$ within accept $O(\log \log n)$ space on a one–way alternating machine by using the power of universal states. This proves the optimality of the corresponding lower bound, and witnesses that one–way alternation in accept $O(\log \log n)$ space is strictly more powerful than nondeterminism under the same restrictions. Subsequently, we consider middle space and show that, unlike other presented cases, the optimal space lower bound for nonregular tally languages accepted by one–way devices is *strictly higher* than that with respect to general languages. More precisely, we prove that

> one–way alternating Turing machines accepting nonregular tally languages within middle $s(n)$ space must satisfy $s(n) = \Omega_\infty(\log n)$.

For the sake of brevity, all the proofs in this version of the paper are just outlined.

2 Preliminaries

We briefly recall some basic notions. For more details, the reader is referred to any of the standard introductory texts on formal languages and complexity theory (e.g., [HU79, BDG87]).

Let Σ^* be the set of all the strings on the alphabet Σ. The length of a string $x \in \Sigma^*$ is denoted by $|x|$, and the complement of a language $L \subseteq \Sigma^*$ by L^c. L is said to be *tally* (or *unary*) if and only if Σ contains only one symbol.

Throughout the paper, we consider *Turing machines* equipped with a read–only input tape containing input strings surrounded by two endmarkers, and a read/write work tape. Space complexity is measured on the only work tape hence allowing sublinear bounds. W.l.o.g. we assume that Turing machines never print the *blank* symbol onto their work tape. An *internal configuration* is a triple specifying work tape content, position of the head on the work tape, and state of the control. We recall that a *computation* of a deterministic or nondeterministic Turing machine is represented by a sequence of internal configurations while, on *alternating Turing machines*, it may be depicted by a *computation tree* [CKS81].

Let M be a deterministic, nondeterministic, or alternating Turing machine:

- M works in strong $s(n)$ space if and only if each computation on any input of length n uses at most $s(n)$ work tape cells;
- M works in middle $s(n)$ space if and only if each computation on any accepted input of length n uses at most $s(n)$ work tape cells;
- M works in accept $s(n)$ space if and only if each accepting computation on any input of length n uses at most $s(n)$ work tape cells;
- M works in weak $s(n)$ space if and only if on each accepted input of length n there exists an accepting computation using at most $s(n)$ work tape cells.

As recalled in the Introduction, all these notions are equivalent when considering overlogarithmic space complexities. Moreover, in the deterministic case, middle, accept, and weak notions coincide even for sublogarithmic bounds.

A Turing machine M is said to be *one–way* whenever its input head can never move toward left, otherwise M is *two–way*. We can formalize intermediate situations by imposing a bound $i(n)$ on the *number of input head reversals* (also *inversions*). We always compute $i(n)$ by considering those computations by which space is defined. So, for instance, we say that an alternating Turing machine M *simultaneously* works within weak $s(n)$ space and $i(n)$ input head reversals if and only if on any accepted input of length n there exists an accepting computation tree using at most $s(n)$ work tape cells and where input head direction is reversed at most $i(n)$ times on each path from the root. For technical reasons, we stipulate $i(n) = 1$ for one–way devices.

We recall that, for any two functions $f, g : \mathbf{N} \rightarrow \mathbf{N}$, $f(n) = \Omega_\infty(g(n))$ if and only if $f(n) \geq d \cdot g(n)$ for a constant $d > 0$ and infinitely many n.

3 Deterministic and Nondeterministic Turing Machines Accepting Nonregular Languages

In this section, we first study minimal space requirements for one–way and for two–way, deterministic and nondeterministic Turing machines accepting nonregular languages under the four different space notions previously introduced. In particular, we concentrate on accept space bounded models of computation. Thereafter, we extend the lower bounds on the product $s(n) \cdot i(n)$ given in [BMP94b] for the strong case to middle, accept and weak machines. Finally, we focus our attention on proving that the lower bounds presented are tight even with respect to tally languages. To this aim, we start by briefly presenting some properties of a simple nonregular tally language and of its complement.

3.1 The Languages $\mathcal{L}$ and $\mathcal{L}^c$

For any positive integer n, let $q(n)$ be the smallest positive integer not dividing n. Thus, let $\mathcal{L}$ be the language defined as

$$\mathcal{L} = \{1^n : q(n) \text{ is a power of } 2\}.$$

The language $\mathcal{L}$ was first considered by Alt and Mehlhorn in [AM75] where they proved that it is a nonregular language acceptable in strong deterministic $O(\log\log n)$ space. Such a space bound is a direct consequence of the fact that $q(n) = O(\log n)$ for any n.

Recently, this result was strengthened in [BMP94b] by showing:

Theorem 1. *The language $\mathcal{L}$ can be recognized in* strong *$O(\log\log n)$ space by a deterministic Turing machine with $O(\frac{\log n}{\log\log n})$ input head reversals.*

Let us turn to the complement of $\mathcal{L}$. A well known result in [Sip80] states that strong deterministic space complexity classes are closed under complement. However, the proof of such a result makes use of a simulation technique which is not input head reversals preserving. On the other hand, a trivial modification of the algorithm exhibited in [BMP94b] for proving Theorem 1 shows that even $\mathcal{L}^c$ can be accepted in strong deterministic $O(\log\log n)$ space and $O(\frac{\log n}{\log\log n})$ input head reversals. We now prove that we can do better on input head reversals by using weak space.

Theorem 2. *The language $\mathcal{L}^c$ can be recognized in* weak *$O(\log\log n)$ space by a one–way nondeterministic Turing machine.*

Proof. (outline) It can be shown that, for any n, $q(n)$ is not a power of 2 if and only if there exist two positive integers s, t with $2^s < t < 2^{s+1}$, $t = O(\log n)$, and such that $n \equiv 0 \pmod{2^s}$ and $n \not\equiv 0 \pmod{t}$.

Hence, we can construct a nondeterministic Turing machine M accepting $\mathcal{L}^c$ that implements the following procedure:

input(1^n)
guess an integer s, $s > 1$
guess an integer t, $2^s < t < 2^{s+1}$
if $n \equiv 0 \pmod{2^s}$ *and* $n \not\equiv 0 \pmod{t}$ **then** *accept*
else *reject*

It can be immediately seen that M works in weak $O(\log\log n)$ space. Moreover, condition $n \equiv 0 \pmod{2^s}$ *and* $n \not\equiv 0 \pmod{t}$ is easily checkable by scanning the input once while counting its length modulo 2^s and t. □

3.2 Tight Lower Bounds

We briefly recall that space lower bounds for one–way and for two–way, deterministic and nondeterministic Turing machines accepting nonregular languages are proved for the strong case in [SHL65, HU69]. Such lower bounds remain the same even when considering middle space [Sze88]. Different lower bounds for weak space are obtained in [Alb85] (see Table 1, next page).

Let us now focus on accept space. By above results and trivial relationships among space notions, it is easy to see that $s(n) = \Omega_\infty(\log n)$ and $s(n) = \Omega_\infty(\log\log n)$ for one–way deterministic and for two–way deterministic or nondeterministic machines, respectively. The only case which deserves our attention is then examined in the following:

Theorem 3. *Let M be a one–way nondeterministic Turing machine recognizing a nonregular language L in* accept *$s(n)$ space. Then $s(n) = \Omega_\infty(\log n)$.*

Proof. (outline) $L \subseteq \Sigma^*$ being nonregular, for any positive integer k we can assume $x = a_1 a_2 \ldots a_n$, with $a_i \in \Sigma$, to be one of the shortest strings accepted by M in *at least* k space. So, there exist $n+1$ internal configurations $c_0, c_1, \ldots, c_n$ of M such that c_0 is the initial internal configuration, for each $1 \leq i \leq n$, c_i can be reached from c_{i-1} by consuming the ith input symbol, and c_n is an accepting internal configuration using at least k space. Note that if $c_i = c_j$ for some $0 \leq i < j \leq n$, then even $x' = a_1 \ldots a_i a_{j+1} \ldots a_n$ is accepted by using at least k space. But $|x'| < |x|$ which clearly contradicts our assumption on x. Hence, we must require the existence of at least $n+1$ different internal configurations of M on inputs of length n, thus obtaining $2^{O(s(n))} \geq n + 1$. □

Table 1 resumes space lower bounds so far considered and obtained. Their optimality is witnessed by several languages in literature. Such languages are mainly defined over alphabets with two or more symbols. Here, we want to stress an even *stronger* optimality condition, namely that with respect to *tally* languages. To this aim, we consider the languages $\mathcal{L}$ and $\mathcal{L}^c$ studied in Section 3.1. As a consequence of Theorem 1, we can provide algorithms for accepting $\mathcal{L}$ whose space complexity exactly fits all the lower bounds for two–way machines. Furthermore, it is not difficult to construct a one–way logarithmic space bounded deterministic machine that accepts $\mathcal{L}$ by first storing in binary notation input length on its work tape. Thus, $\mathcal{L}$ is also able to fulfill all the lower bounds for one–way

	Strong	Middle	Accept	Weak
One–way Deterministic	$\log n$	$\log n$	$\log n$	$\log n$
One–way Nondeterministic	$\log n$	$\log n$	$\log n$	$\log\log n$
Two–way Deterministic	$\log\log n$	$\log\log n$	$\log\log n$	$\log\log n$
Two–way Nondeterministic	$\log\log n$	$\log\log n$	$\log\log n$	$\log\log n$

Table 1. Space lower bounds: strong [SHL65, HU69], middle [Sze88], weak [Alb85], accept Theorem 3

devices except for the nondeterministic weak space case[2]. It seems difficult to design nondeterministic algorithms that accept $\mathcal{L}$ in weak $O(\log\log n)$ space by one input scan. However, in Section 3.1, we have provided a one–way nondeterministic machine accepting $\mathcal{L}^c$ within weak $O(\log\log n)$ space. As a matter of fact, one could make use of $\mathcal{L}^c$ for proving the optimality of all the proposed lower bounds.

Theorem 4. *All the lower bounds given in Table 1 are optimal even with respect to tally languages.*

Let us now investigate how input head reversals can explicitly concur in defining lower bounds for deterministic and nondeterministic machines recognizing nonregular languages. We show how to suitably adapt the proof given in [BMP94b] for strong bounded machines to the other space notions.

The following simulation result in part motivates the introduction of the accept space complexity notion:

Lemma 5. *Let M be a deterministic or nondeterministic Turing machine working within either* strong *or* middle *or* accept *(*weak*) $s(n)$ space, and $i(n)$ input head reversals. Then, there exists an equivalent one–way nondeterministic machine M' using* accept *(*weak*) $O(s(n)\cdot i(n))$ space.*

Proof. (outline) Any accepting computation of M on inputs of length n can be suitably represented by a sequence of $n+1$ *crossing sequences* [Hen65] of at most $i(n)+1$ internal configurations of M. Thus, M' simulates M by guessing, step by step, a crossing sequence and locally testing "compatibility" with the previous one on the currently scanned input symbol. The space needed to store each guessed crossing sequence of an accepting computation of M is clearly $O(s(n)\cdot i(n))$ whenever M is either strong or middle or accept space bounded. Hence, in these three cases, M' turns out to be a one–way nondeterministic machine using accept $O(s(n)\cdot i(n))$ space. On the other hand, was M weak space bounded, then it is easy to notice that so would do M'. □

[2] We recall that the optimality of the weak $\log\log n$ space lower bound for one–way nondeterministic machines accepting general nonregular languages was first proved in [Fre79] by means of the language $\{0^n10^k : n \neq k\}$.

Theorem 6. *Let $r \in$ {Deterministic, Nondeterministic}, $\mathsf{c} \in$ {Strong, Middle, Accept, Weak}, and $l(n)$ be the (r, c)th entry of Table 2. For any r Turing machine M accepting a nonregular language in c space $s(n)$ with $i(n)$ input head reversals, $s(n) \cdot i(n) = \Omega_\infty(l(n))$. This lower bound is optimal even with respect to tally languages.*

	Strong	Middle	Accept	Weak
Deterministic	$\log n$	$\log n$	$\log n$	$\log n$
Nondeterministic	$\log n$	$\log n$	$\log n$	$\log \log n$

Table 2. Simultaneous lower bounds on the product $s(n) \cdot i(n)$: strong [BMP94b], and middle, accept, weak Theorem 6

Proof. (outline) By Lemma 5, we can construct from M an equivalent one–way nondeterministic machine M' working in accept or weak $O(s(n) \cdot i(n))$ space. Thus, by applying to M' the lower bounds in Table 1, the desired result follows.

For what is concerned with optimality, arguments in Section 3.1 can be used to show that the tally language $\mathcal{L}^c$ fits all the lower bounds in Table 2. □

4 Alternating Turing Machines Accepting Nonregular Languages

Let us now analyze minimal resource requirements for alternating Turing machines accepting nonregular languages. Lower bounds for strong space are proved in [Sud80], for middle space in [Sze88], and for weak space in [Iwa93]. Accept space lower bounds can be easily obtained by considering the relationship with middle and weak machines (Table 3).

	Strong	Middle	Accept	Weak
One–way Alternating	$\log n$	$\log \log n$	$\log \log n$	$\log \log n$
Two–way Alternating	$\log \log n$	$\log \log n$	$\log \log n$	$\log \log n$

Table 3. Space lower bounds: strong [Sud80], middle [Sze88], weak [Iwa93]

By also taking into account input head reversals, we get the lower bounds summarized in Table 4, and proved in the following:

Theorem 7. *Let M be an alternating Turing machine accepting a nonregular language in strong (middle or accept or weak) space $s(n)$ with $i(n)$ input head reversals. Then, $s(n) \cdot i(n) = \Omega_\infty(\log n)$ $(s(n) \cdot i(n) = \Omega_\infty(\log \log n))$.*

	Strong	Middle	Accept	Weak
Alternating	$\log n$	$\log\log n$	$\log\log n$	$\log\log n$

Table 4. Simultaneous lower bounds on the product $s(n) \cdot i(n)$: Theorem 7

Proof. (outline) For the strong case, we can "forget" the fact that M is an alternating machine, since computational resources are measured without distinguishing between accepting and nonaccepting computations. Thus, we can apply the technique used in [BMP94b] for the corresponding nondeterministic case. The other results follow by observing that the lower bounds for the corresponding one–way and two–way devices coincide. □

Let us now investigate the optimality of the bounds in Tables 3 and 4. We begin by observing that the lower bounds for strong and weak cases coincide with the corresponding ones for nondeterministic machines, whence their optimality (even for tally languages) comes immediately out. The same argument works for two–way machines using middle and accept space. So, to complete the optimality analysis of Table 3, it remains to consider middle and accept space lower bounds for one–way alternating machines.

For the accept case, from Table 1 we first notice that no one–way nondeterministic machine recognizing a nonregular language in accept $O(\log\log n)$ space can exist. So, in order to show the optimality of the $\log\log n$ lower bound, we actually have to use the power of universal branching as in the following:

Theorem 8. *The language $\mathcal{L}$ can be recognized in* accept $O(\log\log n)$ *space by a one–way alternating Turing machine.*

Proof. (outline) We can construct an alternating Turing machine M accepting $\mathcal{L}$ that implements the following procedure:

input(1^n)
existentially guess an integer r, $r > 2$
universally generate all the integers s, $2 \leq s \leq r$
if $s < r$ **then**
  if $n \equiv 0 \pmod s$ **then** *accept*
    else *reject*
else
  if $n \not\equiv 0 \pmod s$ *and* s is a power of 2 **then** *accept*
    else *reject*

The universal branching checks whether a guessed integer r is the smallest not dividing n and whether r is a power of 2. M accepts whenever both these conditions are satisfied. Accept $O(\log\log n)$ space bound can be obtained by observing that the only guess leading to acceptance is the one corresponding to $q(n)$, and recalling that $q(n) = O(\log n)$. Further, computing the length of the input modulo a given integer can be accomplished in one input scan. □

Let us now turn to the middle model. The optimality of the $\log\log n$ space lower bound for one–way alternating machines was first proved in [Sze88] by means of the nonregular language

$$\mathcal{A} = \{a^k b^m : m \text{ is a common multiple of all } r \leq k\}.$$

The language $\mathcal{A}$ is defined over a two letter alphabet, and this is fundamental for efficiently computing membership since the algorithm proposed in [Sze88] is crucially based on the internal structure of the strings in $\mathcal{A}$. Up to now, the existence of a unary—and hence structureless—language matching the considered lower bound has been an open question.

Here, we solve this problem by giving a negative answer. In fact, we obtain an higher optimal lower bound for tally languages.

Theorem 9. *Let M be a one–way alternating Turing machine accepting a nonregular tally language L in* middle *$s(n)$ space. Then, $s(n) = \Omega_\infty(\log n)$. Moreover, this bound is tight.*

Proof. (outline) By contradiction, assume $s(n) = o(\log n)$. So, we can find an integer n' such that $2^{s(n')} < n'$ (*) and the string $1^{n'}$ belongs to L. Let c be an internal configuration reachable by M in $m \geq n'$ moves. Then, there exists a sequence $c_0, c_1, \ldots, c_{n'}, \ldots, c_m$ where c_0 is the initial internal configuration, $c_m = c$, and c_{i+1} is an immediate successor of c_i for each $0 \leq i \leq m-1$. L being unary, internal configurations $c_0, c_1, \ldots, c_{n'}$ can be reached on input $1^{n'}$, and hence, by the middle space bound, use at most $s(n')$ space. From the inequality (*), there must exist $0 \leq i < j \leq n'$ such that $c_i = c_j$ thus implying that c is even reachable in less than m moves. By iterating this argument, we get that any configuration reachable by M can be reached within less then n' moves. This implies that L is regular whence a contradiction follows.

For the optimality, it is enough to consider the language $\mathcal{L}$. □

Theorem 9 completes optimality analysis of the bounds in Table 3 by remarking the only difference, so far obtained, on minimal resource requirements between tally and general nonregular languages acceptance.

Further, the optimality of the bounds in Table 4 can be immediately drawn from the corresponding one–way case except for middle machines accepting nonregular tally languages. We conjecture that the difference pointed out in Theorem 9 holds even in this case, and we leave, as an open problem, to prove or disprove that $\log\log n$ is an optimal lower bound on the product middle space$\times$input head reversals for alternating Turing machines accepting nonregular tally languages.

Acknowledgment. The authors wish to thank anonymous referees for useful comments and remarks. In particular, observations made by a referee helped in simplifying the proof of Theorem 2.

References

[Alb85] M. Alberts. Space complexity of alternating Turing machines. In *Fundamentals of Computation Theory, Proceedings*, Lecture Notes in Computer Science 199, pages 1–7. Springer Verlag, 1985.

[AM75] H. Alt and K. Mehlhorn. A language over a one symbol alphabet requiring only $O(\log\log n)$ space. *SIGACT news*, 7:31–33, 1975.

[BDG87] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. EATCS Monographs on Theoretical Computer Science 11. Springer Verlag, 1987.

[BMP94a] A. Bertoni, C. Mereghetti, and G. Pighizzini. On languages accepted with simultaneous complexity bounds and their ranking problem. In *Mathematical Foundations of Computer Science 1994, Proceedings*, Lecture Notes in Computer Science 841, pages 245–255. Springer Verlag, 1994.

[BMP94b] A. Bertoni, C. Mereghetti, and G. Pighizzini. An optimal lower bound for nonregular languages. *Information Processing Letters*, 50:289–292, 1994. Corrigendum. *ibid.*, 52:339, 1994.

[CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.

[DH94] C. Damm and M. Holzer. Inductive counting below LOGSPACE. In *Mathematical Foundations of Computer Science 1994, Proceedings*, Lecture Notes in Computer Science 841, pages 276–285. Springer Verlag, 1994.

[Fre79] R. Freivalds. On time complexity of deterministic and nondeterministic Turing machines. *Latvijskij Matematiceskij Eshegodnik*, 23:158–165, 1979. (In Russian).

[Gef91] V. Geffert. Nondeterministic computations in sublogarithmic space and space constructibility. *SIAM J. Computing*, 20:484–498, 1991.

[Gef93] V. Geffert. Tally version of the Savitch and Immerman–Szelepcsényi theorems for sublogarithmic space. *SIAM J. Computing*, 22:102–113, 1993.

[Hen65] F. Hennie. One–tape, off–line Turing machine computations. *Information and Control*, 8:553–578, 1965.

[HU69] J. Hopcroft and J. Ullman. Some results on tape–bounded Turing machines. *Journal of the ACM*, 16:168–177, 1969.

[HU79] J. Hopcroft and J. Ullman. *Introduction to automata theory, languages, and computations.* Addison–Wesley, Reading, MA, 1979.

[Iwa93] K. Iwama. ASPACE($o(\log\log n)$) is regular. *SIAM J. Computing*, 22:136–146, 1993.

[LR94] M. Liśkiewicz and R. Reischuk. The complexity world below logarithmic space. In *Structure in Complexity Theory, Proceedings*, pages 64–78, 1994.

[SHL65] R. Stearns, J. Hartmanis, and P. Lewis. Hierarchies of memory limited computations. In *IEEE Conf. Record on Switching Circuit Theory and Logical Design*, pages 179–190, 1965.

[Sip80] M. Sipser. Halting space–bounded computations. *Theoretical Computer Science*, 10:335–338, 1980.

[Sud80] I. Sudborough. Efficient algorithms for path system problems and applications to alternating and time–space complexity classes. In *Proc. 21st IEEE Symposium on Foundations of Computer Science*, pages 62–73, 1980.

[Sze88] A. Szepietowski. Remarks on languages acceptable in $\log\log n$ space. *Information Processing Letters*, 27:201–203, 1988.

[Sze94] A. Szepietowski. *Turing Machines with Sublogarithmic Space*. Lecture Notes in Computer Science 843. Springer Verlag, 1994.