Space and Reversals Complexity of Nonregular Languages

Alberto Bertoni   Carlo Mereghetti   Giovanni Pighizzini[1]

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
via Comelico 39
20135 Milano, Italy
{mereghc,pighizzi}@dsi.unimi.it

[1]Corresponding author.

Abstract

In this paper, lower bounds on *space and input head reversals* for deterministic, nondeterministic, and alternating Turing machines accepting *nonregular* languages are studied. Three notions of space, namely strong, middle, weak, are considered, and another notion, called accept, is introduced. In all cases we obtain tight lower bounds. Moreover, we prove that, while for determinism and nondeterminism such lower bounds are optimal even with respect to *unary languages*, for alternation optimal lower bounds for unary languages turn out to be strictly higher than those for languages over alphabets with two or more symbols.

Key words. Computational complexity, computational lower bounds, formal languages

AMS subject classifications. 68Q05, 68Q10, 68Q68

1 Introduction

In their foundational works, Lewis *et al.* (1965) and Stearns *et al.* (1965) first settled the following problem:

> " What is the minimal amount of space used by a Turing machine — equipped with a read–only input tape and a read/write work tape — that accepts a *nonregular* language? "

This problem turns out to be tightly related to several concepts and problems in structural complexity. Among others, it provides meaningful insights into the world of *sublogarithmic space bounded computations* (see, e.g., Geffert, 1991; Szepietowski, 1994; Liśkiewicz and Reischuk, 1996; Geffert, 199X).

Different answers have been given to the question above, depending both on the computational paradigm — e.g., determinism, nondeterminism, alternation — and on the *space notion* considered. For the sake of convenience, we collect known space lower bounds in Table 1, and refer the reader to next sections for more detailed explanations, especially concerning the notions of strong, middle, and weak space.

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

TABLE 1. Space lower bounds for one–way and two–way machines accepting nonregular languages: strong (Lewis *et al.*, 1965; Stearns *et al.*, 1965; Hopcroft and Ullman, 1969; Sudborough, 1980), middle (Szepietowski, 1988), weak (Alberts, 1985; Iwama, 1993). The table is meant to be read as follows: let r and c be two indices taking values in the sets $\{\text{One–way}, \text{Two–way}\} \times \{\text{Deterministic}, \text{Nondeterministic}, \text{Alternating}\}$ and $\{\mathsf{Strong}, \mathsf{Middle}, \mathsf{Weak}\}$, respectively. If the (r, c)th entry of the table is the function $\ell(n)$, then any r machine accepting a nonregular language in space $s(n)$ of type c must satisfy $s(n) \notin o(\ell(n))$.

All these lower bounds are proved to be *optimal.* This means that, for each statement of the form $s(n) \notin o(\ell(n))$, a nonregular language can be exhibited which is recognized in $O(\ell(n))$ space on the appropriate model of computation.

From Table 1, one immediately observes that one–way space lower bounds are, almost everywhere, strictly higher than their corresponding two–way versions. Thus, it is interesting to precisely estimate how the power of *reversing the input head* can actually lower the space complexity of nonregular languages. (A similar analysis for work tape head reversals is presented by Hong (1984).) To this purpose, we introduce the notion of Turing machine working *simultaneously* in $s(n)$ space and $i(n)$ *input head reversals*, and study lower bounds on the product $s(n) \cdot i(n)$ for accepting nonregular languages.

To approach this problem, we find it useful to introduce a new definition of space, called accept, which lays between the middle and weak notions: the space bound should be satisfied by any accepting computation (worst cost of acceptance). We are able to prove that any one–way deterministic or nondeterministic machine recognizing a nonregular language in $s(n)$ accept space, must satisfy $s(n) \notin o(\log n)$. For one–way alternation and two–way machines, we get $s(n) \notin o(\log\log n)$. Such lower limits are tight.

This is an intermediate step in the computation of lower bounds on $s(n) \cdot i(n)$ for *deterministic and nondeterministic machines.* Our next step is to reduce a $s(n)$, $i(n)$ bounded deterministic or nondeterministic machine to an equivalent *one–way* machine. By using a crossing sequence argument, we show that any deterministic or nondeterministic machine working within strong, middle, or accept (weak) $s(n)$ space and $i(n)$ input head reversals can be simulated by a *one–way* nondeterministic machine in accept (weak) space $s'(n) \in O(s(n) \cdot i(n))$.

At this point, by applying to $s'(n)$ the space lower bounds above presented for one–way accept and weak nondeterministic machines, we get the lower limits on $s(n) \cdot i(n)$ collected in Table 2.

	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. Lower bounds on the product $s(n) \cdot i(n)$ for deterministic and nondeterministic machines.

For alternation, we show, via a more direct approach, that any alternating machine accepting a nonregular language within strong $s(n)$ space and $i(n)$ input head reversals must satisfy $s(n) \cdot i(n) \notin o(\log n)$, while in middle, accept, and weak cases we get $s(n) \cdot i(n) \notin o(\log\log n)$. To complete our analysis, we give a detailed account on $s(n) \cdot i(n)$ lower bounds for the *alternating hierarchy*, i.e., for Σ_k and Π_k machines.

Even for all $s(n) \cdot i(n)$ lower bounds, we are able to guarantee optimality.

It is worth noticing that $s(n) \cdot i(n)$ lower bounds in Table 2 plus those for alternation coincide with those on space for one–way machines, collected in Table 1. This gives added evidence to the relevance of input head reversals in nonregular acceptance.

In the realm of alternation bounded machines, we obtain interesting results concerning lower bounds optimality with respect to *unary languages*, i.e., languages over a single letter alphabet. Several contributions witness the importance of studying the "unary version" of problems. For instance, Geffert (1993) proves that, by restricting to unary languages, the Theorems of Savitch and Immerman–Szelepcsényi hold true even for *sublogarithmic* space complexities. This extension turns out to be an open question when referred to general input alphabets.

Even in the study of computational lower bounds for nonregular acceptance, dissimmetries between the general and the unary case emerge. Mereghetti and Pighizzini (1995) show that one–way middle alternating machines accepting *unary* nonregular languages must satisfy the optimal lower bound $s(n) \notin o(\log n)$. This is to be compared with the optimal $s(n) \notin o(\log\log n)$ for general languages. Here, we give another evidence that unary languages are generally more difficult to accept than general languages. In fact, we prove that any one–way Π_2 machine recognizing a *unary* nonregular language in accept $s(n)$ space, must satisfy the optimal lower bound $s(n) \notin o(\log n)$. This in spite of the optimal $s(n) \notin o(\log\log n)$, for general alphabets. On the other hand, the power of reversing the input head, even only a bounded number of times, enable us to almost bridge such gaps. In fact, Geffert *et al.* (199X) exhibit nonregular unary languages that can be accepted by middle Π_1 machines satisfying $s(n) \cdot i(n) \in O(\log\log n \cdot f(n))$, for any unbounded monotone increasing recursive function $f(n)$. Roughly speaking, we can approach the $\log\log n$ lower bound for general languages as much as we like. We call this property "quasi–optimality".

For all lower bounds, we inspect unary optimality by devising efficient algorithms for recognizing the following nonregular language, introduced by Alt and Mehlhorn (1975) and by Freedman and Ladner (1975), and its complement $\mathscr{L}^c$:

$$\mathscr{L} = \{1^n \mid \text{the smallest integer not dividing } n \text{ is a power of } 2\},$$

Bertoni *et al.* (1994a) prove that $\mathscr{L}$ and $\mathscr{L}^c$ can be accepted in strong $O(\log\log n)$ space by deterministic machines performing $O\left(\frac{\log n}{\log\log n}\right)$ input head reversals. Here, we prove that $\mathscr{L}$ and $\mathscr{L}^c$ can be recognized in accept $O(\log\log n)$ space by one–way Π_2 machines, and that $\mathscr{L}^c$ is accepted in weak $O(\log\log n)$ space by a one–way nondeterministic machine. These results show that the unary optimality of $s(n) \cdot i(n)$ lower bounds can be matched by using the minimal amount $\log\log n$ of space allowed for nonregular languages.

The paper is organized as follows: Section 2 contains basic definitions, in particular, the different notions of space and the notion of Turing machine simultaneously bounded on space and input head reversals. In Section 3, we present space lower bounds for nonregular acceptance on one–way machines. We prove new lower bounds for accept space, and we study optimality with respect to languages over general alphabets. The unary optimality of one–way space lower bounds is investigated in Section 4. To this purpose, we design efficient algorithms for accepting the languages $\mathscr{L}$ and $\mathscr{L}^c$; some gap results are shown. Section 5 contains the analysis of $s(n) \cdot i(n)$ lower bounds. To this aim, type and amount of space of two–way versus one–way simulation is estimated. Again, special attention is paied on unary optimality. A "quasi–optimality" condition for $s(n) \cdot i(n)$ lower bounds for Σ_k and Π_k machines accepting unary nonregular languages is pointed out.

Some of the results contained in this paper were presented, in a preliminary form, in Bertoni *et al.* (1994), (1995).

2 Preliminary notions and results

In this section, we introduce the concept of Turing machine (deterministic, nondeterministic, and alternating) *simultaneously bounded on both space and input head reversals.* Then, we recall some facts concerning a nonregular unary language, that will be useful in the following.

Let Σ^* be the set of all strings (including the *empty string* ε) built on an alphabet Σ. We let $|x|$ be the length of the string x in Σ^*. A language L is a subset of Σ^*; if Σ is a single letter alphabet, then $L \subseteq \Sigma^*$ is said to be *unary* (or *tally*). The *complement* of L is the language $L^c = \Sigma^* \backslash L$.

We consider the standard Turing machine model (see Lewis *et al.*, 1965; Stearns *et al.*, 1965; Szepietowski, 1994) having a finite state control, a two–way read-only input tape (with input enclosed between a left and a right end marker symbol, '¢' and '\$', respectively), and a separate semi-infinite two–way read-write work tape. A *memory state*[1]

[1]Sometimes called *internal configuration* (see, e.g., Szepietowski, 1988, 1994).

of a Turing machine M is an ordered triple (q, w, i) where q is a finite control state, the string w is the nonblank content of the work tape, and the integer $1 \leq i \leq |w| + 1$ denotes the head position on the work tape. A *configuration* of M on a given input string x is an ordered pair (j, m) where the integer $0 \leq j \leq |x| + 1$ represents the input head position, while m is a memory state. The *initial configuration* is the pair $(0, (q_0, \varepsilon, 1))$ where q_0 is the *initial state*. An *accepting configuration* is any configuration containing a *final state*. Without loss of generality, we assume that M always enters accepting configurations by having its input head on the right end marker.

It is well–known that a computation of a deterministic or nondeterministic Turing machine on a given input string can be seen as a finite or infinite sequence $c_0, c_1, \ldots, c_i, \ldots$ of configurations beginning in the initial configuration c_0 and such that c_{i+1} is an *immediate successor* of c_i on the appropriate input symbol (i.e., c_{i+1} is reachable from c_i in a *move*). A computation is *accepting* whenever ending in an accepting configuration.

The *space* used in a computation is the maximum number of work tape cells taken by the configurations in that computation. An *input head reversal* in a computation is a sequence $\alpha, \beta_1, \beta_2, \ldots, \beta_s, \gamma$, with $s \geq 1$, of configurations in that computation each one being the immediate successor of the previous one, and having the form

$$\begin{array}{llll} \alpha = (j-1, m') & \beta_i = (j, m_i'') & \gamma = (j-1, m'''), & \text{or also} \\ \alpha = (j+1, m') & \beta_i = (j, m_i'') & \gamma = (j+1, m'''). & \end{array}$$

A Turing machine is said to be *one–way* whenever its computations never present input head reversals, otherwise it is *two–way*.

The reader is assumed to be familiar with the notion of *alternating Turing machine*, introduced by Chandra *et al.* (1981). We only recall that, in an alternating Turing machine, the finite control states — and hence configurations and memory states as well — are partitioned into *existential* and *universal*. A computation is described by a tree whose nodes are labelled by configurations. The root is labelled by the initial configuration. Any internal node labelled by an existential configuration e has a unique son labelled by an immediate successor of e. Any internal node labelled by a universal configuration u has a son for each immediate successor of u. A computation is accepting if the corresponding tree is finite and all its leaves are labelled by accepting configurations.

The *space* used in a computation of an alternating Turing machine is the maximum number of work tape cells taken by the configurations labelling the corresponding tree. The *number of input head reversals* used in that computation is the maximum number of input head reversals along computation paths from the root.

A Σ_k (Π_k) *machine* is an alternating Turing machine beginning its computation in an existential (universal, respectively) state, and making at most $k - 1$ switches between existential and universal states along each computation path on any input. It can be easily seen that Σ_1 machines are actually nondeterministic machines. It is stipulated that Σ_0 and Π_0 machines are *deterministic* machines. Figure 1 depicts the relations among machines with constant bounds on alternation. Arcs are directed from less to more general model; the abbreviation "Alt." on top of the hierarchy stands for alternating Turing machines without particular bounds on alternation.

Several notions of space complexity have been defined in the literature. Before quickly reviewing their definition, we recall that a given string $x \in \Sigma^*$ is said to be *accepted* by a

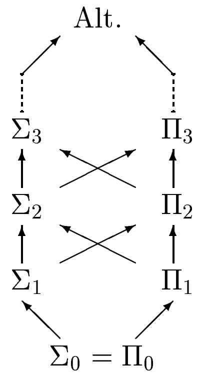

FIGURE 1. Relationships among Σ_k and Π_k machines.

Turing machine M whenever M, on input x, presents at least one accepting computation. The *language accepted by M* is the set $\{x \in \Sigma^* \mid M \text{ accepts } x\}$. Thus, a Turing machine works in $s(n)$ space which is: strong (Lewis *et al.*, 1965; Stearns *et al.*, 1965; Hopcroft and Ullman, 1969; Sudborough, 1980) if any computation on each input of length n uses no more than $s(n)$ space, middle (Szepietowski, 1988) if any computation on each accepted input of length n uses no more than $s(n)$ space, accept[2] (Hromkovič *et al.*, 1994; Bertoni *et al.*, 1995; Slobodová, 1995) if any accepting computation on each input of length n uses no more than $s(n)$ space, weak (Savitch, 1970; Alberts, 1985; Iwama, 1993) if, on each accepted input of length n, there exists at least one accepting computation requiring no more than $s(n)$ space. It is not hard to prove that these notions coincide for *fully space constructible*[3] bounds, e.g., "normal" functions above $\log n$.

In what follows, we aim to take into account space (in all its definitions above) and input head reversals requirements *at once*. Thus, we present the notion of Turing machine *simultaneously bounded on both space and input head reversals*. Precisely, we say that a given deterministic, nondeterministic, or alternating Turing machine, works in

- strong $s(n)$ space and $i(n)$ input head reversals if and only if any computation on each input of length n uses no more than $s(n)$ space and $i(n)$ input head reversals,

- middle $s(n)$ space and $i(n)$ input head reversals if and only if any computation on each accepted input of length n uses no more than $s(n)$ space and $i(n)$ input head reversals.

- accept $s(n)$ space and $i(n)$ input head reversals if and only if any accepting computation on each input of length n uses no more than $s(n)$ space and $i(n)$ input head reversals.

- weak $s(n)$ space if and only if, on each accepted input of length n, there exists at least one accepting computation requiring no more than $s(n)$ space and $i(n)$ input head reversals.

[2]The designation accept has been adopted first by Bertoni *et al.*, (1995).

[3]A function $s(n)$ is said to be *fully space constructible* whenever there exists a deterministic Turing machine which, on any input of length n, uses exactly $s(n)$ space.

Clearly, these definitions are listed in increasing order of generality. Moreover, middle, accept, and weak notions are easily seen to coincide for Π_1 machines. For technical reasons, we stipulate that $i(n) = 1$ for one–way machines

We end this section by briefly reviewing some results on the complexity of accepting the unary nonregular language $\mathscr{L}$, introduced independently by Alt and Mehlhorn (1975), and by Freedman and Ladner (1975). Our interest in $\mathscr{L}$ comes from the fact that the optimality of several lower bounds in the sequel will be proved by designing efficient algorithms for recognizing $\mathscr{L}$ and its complement $\mathscr{L}^c$. For any integer $n \geq 2$, let $q(n)$ be the *smallest positive integer not dividing* n. Define

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

As shown by Alt and Mehlhorn (1975) and by Freedman and Ladner (1975), $\mathscr{L}$ can be accepted by a deterministic Turing machine within strong $O(\log\log n)$ space and $O(\log n)$ input head reversals. (The space bound is basically due to the fact that $q(n) \in O(\log n)$.) Recently, such a result has been improved by Bertoni *et al.* (1994), where it is shown that

Theorem 2.1 *The language $\mathscr{L}$ can be recognized by a deterministic Turing machine in* strong $O(\log\log n)$ *space and* $O(\frac{\log n}{\log\log n})$ *input head reversals.*

For what concerns the complement of $\mathscr{L}$, a trivial modification in the algorithm exhibited by Bertoni *et al.* (1994) for proving Theorem 2.1 shows that even $\mathscr{L}^c$ can be recognized in strong deterministic $O(\log\log n)$ space and $O(\frac{\log n}{\log\log n})$ input head reversals.

3 Optimal space lower bounds for one–way machines

In this section, we analyze space lower bounds for nonregular acceptance on deterministic, nondeterministic, and alternating Turing machines, considering the limit case of one–way machines ($i(n) = 1$, as stipulated in Section 2). In particular, we prove new lower bounds for the accept case.

Known space lower bounds for nonregular acceptance on one–way strong, middle, and weak machines, are summarized below.

- One–way deterministic and nondeterministic strong machines are considered by Lewis *et al.* (1965) and Stearns *et al.* (1965), and by Hopcroft and Ullman (1969). In these contributions, it is proved that $s(n) \notin o(\log n)$ for such devices. Sudborough (1980) extends this lower bound to one–way strong alternation.
- Weak notion is considered by Alberts (1985). One–way weak deterministic, machines must satisfy $s(n) \notin o(\log n)$, while for one–way weak nondeterminism and alternation, $s(n) \notin o(\log\log n)$ holds true.
- Szepietowski (1988) proves that one–way middle deterministic and nondeterministic machines must satisfy $s(n) \notin o(\log n)$. Moreover, since each middle machine can be seen as a weak machine, from the previous point, one can readily get $s(n) \notin o(\log\log n)$ for one–way middle alternation.

Table 3 resumes the scenario so far illustrated.

	Strong	Middle	Weak
Deterministic	$\log n$	$\log n$	$\log n$
Nondeterministic (Σ_1)	$\log n$	$\log n$	$\log \log n$
Alternating	$\log n$	$\log \log n$	$\log \log n$

TABLE 3. Space lower bounds for one–way machines accepting nonregular languages: strong (Lewis *et al.*, 1965; Stearns *et al.*, 1965; Hopcroft and Ullman, 1969; Sudborough, 1980), middle (Szepietowski, 1988), weak (Alberts, 1985).

Now, we focus on accept machines. By the results in Table 3 and trivial relationships among space notions, it is easy to see that

Theorem 3.1 *Given a one–way deterministic (alternating) Turing machine recognizing a nonregular language in* accept $s(n)$ *space, then*

$$s(n) \notin o(\log n) \qquad (s(n) \notin o(\log \log n)).$$

Let us turn to one–way accept nondeterminism. From Table 3, we have a $\log \log n$ lower bound in the weak case. This trivially gives a $\log \log n$ lower bound even for the accept case. On the other hand, the lower bound for middle machines turns out to be $\log n$. So, the question arises of whether the trivial $\log \log n$ lower bound can be raised up.

We positively solve this problem by proving the following:

Theorem 3.2 *Let M be a one–way nondeterministic Turing machine recognizing a nonregular language L in* accept $s(n)$ *space. Then*

$$s(n) \notin o(\log n).$$

Proof. Since $L \subseteq \Sigma^*$ is a nonregular language, it is easy to see that, for any given positive integer $\overline{k}$, the set of strings accepted by M in *at least* $\overline{k}$ space is nonempty. Let $\mu = a_1 a_2 \ldots a_{\overline{n}}$, with $a_i \in \Sigma$, be one of the shortest strings in such a set. So, there exists a sequence $m_0, m_1, \ldots, m_{\overline{n}}$ of memory states of M on input μ, where: m_0 is the initial memory state, m_i is reachable from m_{i-1} via a sequence of moves that "consumes" the ith symbol of μ, and $m_{\overline{n}}$ is an accepting memory state taking at least $\overline{k}$ and no more than $s(\overline{n})$ work tape cells (this latter upper bound due to accept notion).

Now, observe that if $m_i = m_j$, for some $0 \le i < j \le \overline{n}$, then even the string $x = a_1 \ldots a_i a_{j+1} \ldots a_{\overline{n}}$ would be accepted by M in at least $\overline{k}$ space (namely, via the sequence $m_0 \ldots m_i m_{j+1} \ldots m_{\overline{n}}$). But since $|x| < |\mu|$, this would contradict our choice of μ among the shortest strings accepted in at least $\overline{k}$ space.

Hence, we must require the existence of at least $\overline{n} + 1$ different memory states taking at most $s(\overline{n})$ work tape cells. By recalling that the number of memory states using no more than $s(\overline{n})$ work tape cells is bounded above by $d^{s(\overline{n})}$, for a suitable constant $d > 1$, our requirement can be formally written as $d^{s(\overline{n})} > \overline{n}$.

Let us now take an integer $\hat{k} > s(\overline{n})$, and apply the above reasoning. We get the inequality $d^{s(\hat{n})} > \hat{n}$, where $\hat{n}$ is easily seen to be greater than $\overline{n}$. In fact, if we assume, by contradiction, that $\hat{n} \le \overline{n}$ then, by considering one of the shortest strings accepted in at least $\hat{k}$ space, we would get $\hat{k} \le s(\overline{n})$. In conclusion, we can find infinitely many values of n for which $d^{s(n)} > n$ must hold true, whence our claim follows. ∎

Table 4 resumes the complete situation of space lower bounds for one–way machines that accept nonregular languages.

	Strong	Middle	Accept	Weak
Deterministic	$\log n$	$\log n$	$\log n$	$\log n$
Nondeterministic (Σ_1)	$\log n$	$\log n$	$\log n$	$\log\log n$
Π_1	$\log n$	$\log\log n$	$\log\log n$	$\log\log n$
Alternating	$\log n$	$\log\log n$	$\log\log n$	$\log\log n$

TABLE 4. Space lower bounds for one–way machines accepting nonregular languages: accept [this section].

In this and in the following tables, we choose to display also Π_1 machines — whose space lower bounds, in this case, come directly from those for general alternation. This will be useful when later we will study lower bounds for the whole alternating hierarchy.

Theorem 3.3 *All the lower bounds in Table 4 are optimal for general languages.*

Proof. Szepietowski (1988) introduces the nonregular language

$$\mathscr{A}_S = \{a^k b^m \mid m \text{ is a common multiple of all } r \leq k\},$$

and proves that it can be accepted within middle $O(\log\log n)$ space by a one–way Π_1 machine. (Basically, this is due to the fact that if $a^k b^m$ belongs to $\mathscr{A}_S$, then $m \geq d^k$, for a suitable constant $d > 1$.) This, clearly, yields the optimality of all $\log\log n$ lower bounds in Table 4, with the exception of one–way weak nondeterminism. To deal with this case, we consider the language

$$\mathscr{A}_F = \{0^n 1 0^k \mid n \neq k\},$$

presented by Freivalds (1979). It is proved that $\mathscr{A}_F$ can be accepted in weak $O(\log\log n)$ space by a one–way nondeterministic Turing machine. (The accepting algorithm uses the property that $n \neq k$ if and only if $n \not\equiv k \pmod{s}$ for some $s \leq d \cdot \log(n+k)$, with $d > 1$ a suitable constant.)

In conclusion, for what concerns the optimality of the logarithmic lower bounds in Table 4, both the languages $\mathscr{A}_S$ and $\mathscr{A}_F$ are easily seen to be recognizable in strong $O(\log n)$ space by one–way deterministic Turing machines. ∎

The results so far obtained enable us to immediately get optimal space lower bounds for one–way Σ_k and Π_k machines that accept general nonregular languages.

Theorem 3.4 *Let a one–way Σ_k, Π_k, or general alternating Turing machine accept a nonregular language in* strong, middle, accept, *or* weak *$s(n)$ space. Then, $s(n) \notin o(\ell(n))$, where $\ell(n)$ is the function written near each machine model in the diagrams in Figure 2. Such bounds are tight.*

4 Optimal space lower bounds for one–way machines: the unary case

In this section, we investigate space lower bounds for one–way machines that recognize *unary* nonregular languages. Due to the absence of any structure in the input strings, it

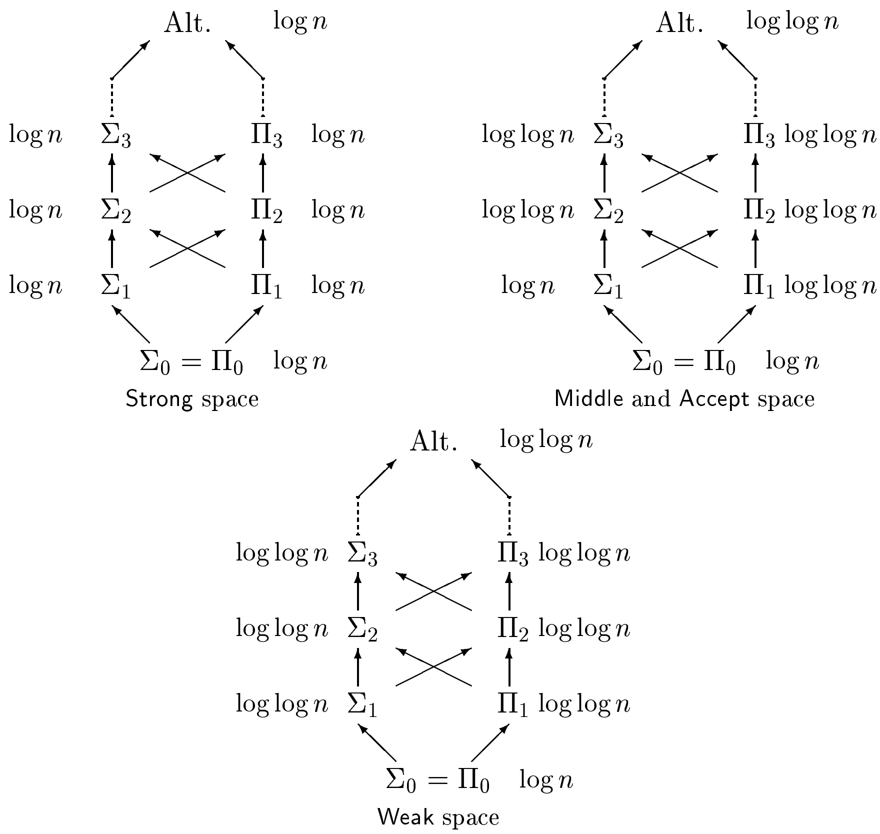

FIGURE 2. Optimal space lower bounds for one–way Σ_k and Π_k machines accepting general nonregular languages.

can be argued that unary languages are generally more "difficult" to accept than general languages. In fact, instead of Table 4, we have the situation displayed in the following

Theorem 4.1 *Let a one–way deterministic, nondeterministic, Π_1, or general alternating Turing machine accept a* unary *nonregular language in* strong*,* middle*,* accept*, or* weak *$s(n)$ space. Then, $s(n)$ must obey the lower bounds in Table 5. Such bounds are tight.*

	Strong	Middle	Accept	Weak
Deterministic	$\log n$	$\log n$	$\log n$	$\log n$
Nondeterministic (Σ_1)	$\log n$	$\log n$	$\log n$	$\log\log n$
Π_1	$\log n$	$\mathbf{log}\ n$	$\mathbf{log}\ n$	$\mathbf{log}\ n$
Alternating	$\log n$	$\mathbf{log}\ n$	$\log\log n$	$\log\log n$

TABLE 5. Optimal space lower bounds for one–way machines accepting unary nonregular languages. Bold lower bounds denote gap situations when compared with corresponding optimal sublogarithmic lower bounds in Table 4.

Proof.

(i) Mereghetti and Pighizzini (1975) prove that one–way middle alternating machines accepting unary nonregular languages must satisfy the optimal lower bound $s(n) \notin$

$o(\log n)$. This, clearly, yields the same optimal lower bound for one–way Π_1 middle machines. Furthermore, as observed in Section 2, middle, accept, and weak notions coincide for Π_1 machines. Thus, $s(n) \notin o(\log n)$ extends to one–way Π_1 accept and weak machines.

(ii) We prove the optimality of the sublogarithmic lower bounds in the weak case by showing that $\mathscr{L}^c$, the complement of the language presented at the end of Section 2, can be recognized in weak $O(\log\log n)$ space by a one–way nondeterministic Turing machine. In fact, it is easy to see that, for any $n \geq 2$, $q(n)$ is not a power of 2 if and only if there exist two positive integers s, t such that

- $2^s < t < 2^{s+1}$, $t = O(\log n)$,
- $n \equiv 0 \pmod{2^s}$ and $n \not\equiv 0 \pmod{t}$.

We remark that the limitation $t = O(\log n)$ comes from the fact that $q(n) \in O(\log n)$, as shown by Alt and Mehlhorn (1975) and by Freedman and Ladner (1975). Hence, we can construct a nondeterministic Turing machine M that accepts $\mathscr{L}^c$ by implementing 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 is not hard to see that M works in weak $O(\log\log n)$ space. Moreover, the 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.

(iii) We prove the optimality of the sublogarithmic lower bound in the accept case by showing that both $\mathscr{L}$ and its complement can be recognized in accept $O(\log\log n)$ space by one–way Σ_2 machines. We construct a Σ_2 machine M that accepts $\mathscr{L}$ by implementing the following procedure:

INPUT(1^n)
existentially guess an integer r, $r \geq 2$
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 lower bound can be obtained by observing that the only guess leading to acceptance is the one corresponding to $q(n)$, and by recalling that $q(n) \in O(\log n)$. Moreover, computing the length of the input modulo a given integer can be accomplished in one input scan. A trivial modification in the last **if**–statement shows that even $\mathscr{L}^c$ can be recognized by a one–way Σ_2 machine in accept $O(\log \log n)$ space.

The unary optimality of the remaining logarithmic lower bounds follows trivially. ∎

For what concerns space lower bounds for unary nonregular acceptance within the one–way alternating hierarchy, the previous theorem immediately leads to

Theorem 4.2 *Let a one–way Σ_k, Π_k, or general alternating Turing machine accept a* unary *nonregular language in* strong, middle, *or* weak *$s(n)$ space. Then, $s(n)$ must obey the lower bounds in the diagrams in Figure 3. Such bounds are tight.*

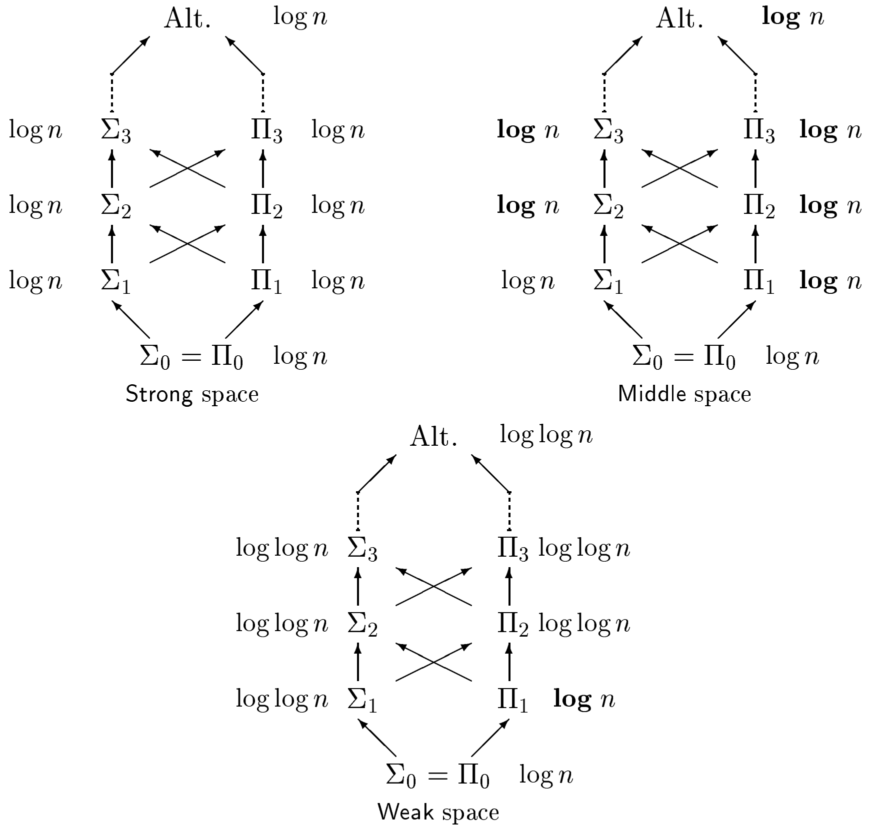

FIGURE 3. Optimal space lower bounds for one–way strong, middle, and weak Σ_k and Π_k machines accepting unary nonregular languages. Bold lower bounds correspond to gap situations when compared with corresponding optimal sublogarithmic lower bounds in Theorem 3.4.

One–way Σ_k and Π_k accept machines deserve more attention. From Theorem 4.1, we get an optimal $\log n$ space lower bound for one–way deterministic, nondeterministic, and Π_1 accept machines, while, for one–way Σ_k and Π_k accept machines, with $k \geq 2$, we have $s(n) \notin o(\log \log n)$. Theorem 4.1 (iii) shows that $s(n) \notin o(\log \log n)$ is optimal for one–way Σ_2 accept machines on unary alphabets. This, clearly, proves the unary optimality of

$s(n) \notin o(\log\log n)$ even for one–way Σ_k accept machines, with $k \geq 2$, and for one–way Π_k accept machines, with $k \geq 3$.

Thus, the only case left open is that of one–way Π_2 accept machines. For such model, we have a $\log\log n$ space lower bound which is optimal for general alphabets (see Table 4 and Theorem 3.3). What about its optimality with respect to unary languages? In the next theorem, we state a gap between space lower bounds for general and unary nonregular acceptance on one–way Π_2 accept machines. This in analogy with the case of one–way middle alternating machines, as proved by Mereghetti and Pighizzini (1995), and recalled in Theorem 4.1 (i).

Theorem 4.3 *Let M be a one–way Π_2 machine accepting a* unary *nonregular language L in* accept *$s(n)$ space. Then*

$$s(n) \notin o(\log n).$$

This bound is tight.

Proof. Let us suppose, by contradiction, that $s(n) \in o(\log n)$. Then, we show how to obtain from M an equivalent alternating automaton.

For each n, the number of different memory states taking no more than $s(n)$ work tape cells is bounded above by $d^{s(n)}$, for a constant $d > 1$. Since we are assuming that $s(n) \in o(\log n)$, and since L is nonregular, there must exist an integer n' such that $d^{s(n')} < n'$, and $1^{n'} \in L$. This allows us to prove the following

Claim *Any universal memory state reachable by M without scanning the right end marker symbol, can be reached in less than n' moves.*

Proof. Let u be a universal memory state reachable in $k \geq n'$ moves by M reading 1's on the input tape. We show that u can be reached in $k' < k$ moves as well. First, notice that, M being a Π_2 machine, any computation path leading to a given universal memory state must go only through universal memory states. So, there exists a sequence $u_0, u_1, \ldots, u_k$ of universal memory states such that: u_0 is the initial memory state, $u_k = u$, and u_{i+1} is reached from u_i by M reading a 1 on the input tape. Consider the initial segment $\mathscr{S} = u_0, u_1, \ldots, u_{n'}$. Clearly, each memory state in $\mathscr{S}$ can be reached on input $1^{n'}$. Moreover, since $1^{n'} \in L$, we get that each memory state in $\mathscr{S}$ belongs to some accepting computation tree, and hence, our machine being accept, it takes at most $s(n')$ work tape cells. Now, recall that $d^{s(n')}$ bounds the number of memory states using up to $s(n')$ work tape cells, and that $d^{s(n')} < n'$. Therefore, a pigeonhole argument on the segment $\mathscr{S} = u_0, u_1, \ldots, u_{n'}$ gives that $u_i = u_j$, for some $0 \leq i < j \leq n'$. It follows that the sequence $u_0, \ldots, u_i, u_{j+1}, \ldots, u_k = u$ leads to u in $k' = k + i - j$ moves. ∎

A simple consequence of the previous Claim, is that the set of universal memory states reachable by M on 1's of input strings is *finite*. More precisely, it does not depend on the input length, and its cardinality is bounded above by $d^{s(n')}$. In turn, E being the set of existential memory states that are immediate successors of some universal memory state reached by M without scanning the right end marker symbol, we clearly get that E is *finite* as well.

Let $M_{\tilde{m}}$ be the machine obtained by initializing M in the memory state $\tilde{m} \in E$. Notice that $M_{\tilde{m}}$ is a one–way nondeterministic machine that, by our initial assumption, uses $s(n) \in o(\log n)$ accept space. Hence, by Theorem 3.2, it can be simulated by a finite automaton $A_{\tilde{m}}$. At this point, it is not difficult to obtain from M a Π_2 automaton A whose states are the (finitely many) universal memory states reachable by M on 1's, plus the states of the automaton $A_{\tilde{m}}$, for any $\tilde{m} \in E$. Final states of A are the final states of $A_{\tilde{m}}$ plus those universal memory states from which, by scanning only the right end marker, M accepts. It is easy to verify that the resulting alternating automaton A recognizes the language L. A well–known result by Ladner *et al.* (1984) states that alternating automata exactly characterize the class of regular languages. This proves that L is regular and yields a contradiction.

The optimality of such a logarithmic lower bound follows trivially. ∎

The next theorem summarizes optimal space lower bounds for unary nonregular acceptance on one–way Σ_k and Π_k accept machines.

Theorem 4.4 *Let a one–way Σ_k, Π_k, or general alternating Turing machine accept a unary nonregular language in* accept *$s(n)$ space. Then, $s(n)$ must obey the lower bounds in the diagram in Figure 4. Such bounds are tight.*

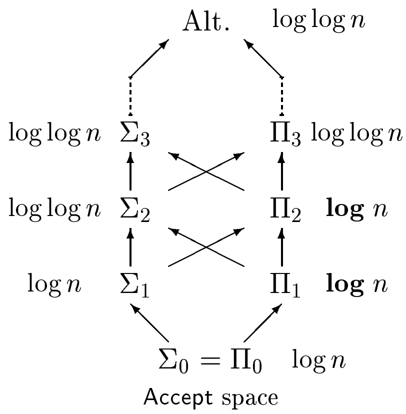

FIGURE 4. Optimal space lower bounds for one–way accept Σ_k and Π_k machines accepting unary nonregular languages. Bold lower bounds correspond to gap situations when compared with corresponding optimal sublogarithmic lower bounds in Table 3.4.

5 Optimal lower bounds on $s(n) \cdot i(n)$ for two–way machines

In this section, we study optimal space lower bounds for nonregular acceptance on two–way machines. First, we can easily obtain space lower bounds in the limit case of two–way machines without particular restrictions on the number of input head reversals.

Theorem 5.1 *Let a two–way Σ_k, Π_k, or general alternating Turing machine accept a nonregular language in* strong, middle, accept, *or* weak *$s(n)$ space. Then*

$$s(n) \notin o(\log\log n).$$

These bounds are tight even for unary languages.

Proof. From the literature:

- Two–way deterministic and nondeterministic machines: $s(n) \notin o(\log\log n)$ is proved for strong space by Lewis *et al.* (1965), Stearns *et al.* (1965), and Hopcroft and Ullman (1969), for middle by Szepietowski (1988), and for weak by Alberts (1985).

- Two–way alternating machines: $s(n) \notin o(\log\log n)$ is proved for strong space by Sudborough (1980), for middle by Szepietowski (1988), and for weak by Iwama (1993).

With these results in our hands, extending $s(n) \notin o(\log\log n)$ to the whole alternating hierarchy and to accept machines, can be immediately obtained by considering trivial relationships among models of computation and space notions. For what concerns the optimality with respect to unary inputs, it suffices to recall, from Theorem 2.1, that the languages $\mathscr{L}$ and $\mathscr{L}^c$ can be recognized in strong $O(\log\log n)$ space on two–way deterministic machines. ∎

By comparing the limits in the previous theorem with those obtained in Sections 3 and 4, it is clear that the possibility of reversing the input head direction turns out to be crucial in order to reduce the space complexity of nonregular languages. We precisely evaluate the importance of input head reversals by studying lower bounds on the product *space*$\times$*input head reversals.*

We start by determining lower bounds on the product $s(n) \cdot i(n)$ for deterministic and nondeterministic machines via the following reduction:

(1) given a strong, middle, or accept (weak) deterministic or nondeterministic machine working within $s(n)$ space and $i(n)$ input head reversals, simulate it with a *one–way* nondeterministic machine that uses accept (weak) space $s'(n) = O(s(n) \cdot i(n))$;

(2) apply to $s'(n)$ the space lower bound for accept (weak) one–way nondeterminism (Table 4), and get lower bounds on $s(n) \cdot i(n)$.

Thus, the core of our approach is the two–way/one–way simulation step at point (1). To attack this problem, we need a natural generalization of the notion of *crossing sequence* introduced by Hennie (1965) to deal with two–way finite state automata.

Consider the boundary between two consecutive input tape squares of a Turing machine M. Denote with m_i the memory state assumed by M when, during a computation, its input head crosses such a boundary for the ith time. We call the list $\mathscr{M} = [m_1, m_2, \ldots, m_i, \ldots]$ *crossing sequence* generated by the computation at the given boundary. Let an input tape square contain the symbol σ, and let $\mathscr{M}'$, $\mathscr{M}''$ be two crossing sequences. Then, $\mathscr{M}'$ *matches* $\mathscr{M}''$ *on* σ (write $\mathscr{M}' \rhd_\sigma \mathscr{M}''$) if there exists a computation of M generating $\mathscr{M}'$ and $\mathscr{M}''$ at the left and right, respectively, boundary of σ.

We are now ready to prove the following

Lemma 5.1 *Let M be a deterministic or nondeterministic Turing machine working in* strong, middle, *or* accept *(*weak*) $s(n)$ space, and $i(n)$ input head reversals. Then, there exists an equivalent* one–way *nondeterministic machine M' using*

$$\text{accept (weak)} \quad O(s(n) \cdot i(n)) \quad \textit{space.}$$

Proof. Without loss of generality, we assume that, when M accepts, its input head is positioned one cell right of the right end marker. It is not hard to prove the following

Claim *Given an input string $x = a_1a_2 \ldots a_n$, M presents an accepting computation on x if and only if there exists a sequence $\mathscr{M}_0, \mathscr{M}_1, \ldots, \mathscr{M}_n, \mathscr{M}_{n+1}, \mathscr{M}_{n+2}$ of crossing sequences satisfying:*

(i) *$\mathscr{M}_0$ consists only of the initial memory state of M (the* initial *crossing sequence),*

(ii) *for $0 \leq i \leq n+1$, $\mathscr{M}_i \rhd_{a_i} \mathscr{M}_{i+1}$, where we set $a_0 = ¢$ and $a_{n+1} = \$$,*

(iii) *$\mathscr{M}_{n+2}$ consists only of one accepting memory state of M (*accepting *crossing sequence).*

Hence, a nondeterministic machine M' simulating M on input $x = a_1a_2 \ldots a_n$, runs the following procedure:

```
INPUT(a_1 ... a_n)          /* input tape content: ¢a_1 ... a_n$ */
set the input head on ¢
𝒯 := the initial crossing sequence
repeat
    a := the currently scanned input symbol
    nondeterministically guess a crossing sequence 𝒯'
    if 𝒯 ▷_a 𝒯' then 𝒯 := 𝒯'
                 else reject
    move the input head one position right
until a = '$'
if 𝒯 is an accepting crossing sequence then accept
                                        else reject
```

The above Claim shows that M' and M are equivalent. Let us now inspect space and input head reversals complexity of M'.

First, suppose that M is a strong, middle, or accept machine working in $s(n)$ space and $i(n)$ input head reversals. In these three cases, it is easy to verify that each crossing sequence generated by any possible accepting computation of M on inputs of length n, consists of a list of at most $i(n)$ memory states each one using no more than $s(n)$ work tape cells. Thus, in the light of the above sketched Claim, we get that the space used by *any* accepting computation of M' is, basically, that needed for the variable $\mathscr{T}$ to store the crossing sequences of the corresponding accepting computation of M. This is also due to the fact that the test " $\mathscr{T} \rhd_a \mathscr{T}'$ " in the procedure is easily checkable with no extra space. Hence, M uses accept $O(s(n) \cdot i(n))$ space.

For what concerns input head reversals, we just notice that the test " $\mathscr{T} \rhd_a \mathscr{T}'$ " can be "locally" checked, i.e., by just knowing $\mathscr{T}$, $\mathscr{T}'$, a, and no other part of the input string. So, M' is *one–way*.

Now, suppose that M works in weak $s(n)$ space and $i(n)$ input head reversals. Then, for any accepted input of length n, there exists an accepting computation whose crossing sequences can be stored in the variable $\mathscr{T}$ by using $O(s(n) \cdot i(n))$ space. So, in this case, M' turns out to be one–way and weak $O(s(n) \cdot i(n))$ space bounded. ∎

This simulation result enables us to easily get lower bounds on the product $s(n) \cdot i(n)$ for deterministic and nondeterministic machines accepting nonregular languages.

Theorem 5.2 *Let M be a Turing machine accepting a nonregular language in $s(n)$ space of type* c*, and $i(n)$ input head reversals. If M is deterministic and* $\mathsf{c} \in \{$strong, middle, accept, weak$\}$*, or M is nondeterministic and* $\mathsf{c} \in \{$strong, middle, accept$\}$*, then*

$$s(n) \cdot i(n) \notin o(\log n).$$

If M is nondeterministic and c = weak*, then*

$$s(n) \cdot i(n) \notin o(\log \log n).$$

Proof. By Lemma 5.1, we can construct from M an equivalent one–way nondeterministic machine M' working in accept (or weak, if M is a weak machine) $O(s(n) \cdot i(n))$ space. Then, by applying to M' the lower bounds in Table 4 for one–way accept and weak nondeterminism, all the claimed lower bounds follow, except that for deterministic weak machines. In this latter case, it suffices to notice that, for deterministic machines, middle, accept, and weak notions coincide, as observed in Section 2. ∎

For alternating Turing machines, lower bounds on $s(n) \cdot i(n)$ are obtained in the following

Theorem 5.3 *Let M be an alternating Turing machine accepting a nonregular language in* strong *(*middle*,* accept*, or* weak*) $s(n)$ space, and $i(n)$ input head reversals. Then*

$$s(n) \cdot i(n) \notin o(\log n) \qquad (s(n) \cdot i(n) \notin o(\log \log n)).$$

Proof. For middle, accept, and weak cases, the result follows easily by observing, from Table 4 and Theorem 5.1, that the optimal space lower bounds for one–way and two–way alternating machines coincide.

Instead, let us focus on the strong case. Notice that, M being a strong machine, the product $s(n) \cdot i(n)$ bounds any computation, accepting or not, on inputs of length n. This enables us to reason on reachability of memory states, regardless they are reached on accepted or not inputs. Hence, we can forget about the alternating accepting policy, and think of M as a nondeterministic machine.

We are now going to show that, if $s(n) \cdot i(n) \in o(\log n)$, then the number of different memory states that M can assume on *any* computation path is *finite.* We use concepts and notations introduced for Lemma 5.1.

It is easy to see that, for a suitable constant $d > 1$, no more than $d^{s(n) \cdot i(n)}$ crossing sequences can be generated by M on any input — *accepted or not* — of length n. Assume, by contradiction, that $s(n) \cdot i(n) \in o(\log n)$. Then, there exists an integer n' such that $d^{s(n) \cdot i(n)} < n$, for any $n \geq n'$.

With an approach similar to that used in the proof of Theorem 4.3, we can prove that any crossing sequence $\mathscr{M}$ generated by any computation path of M can be reached on inputs of length less than n'. It is enough to show that if $\mathscr{M}$ can be reached on a string of length $k \geq n'$, then it can be reached on an input of length less than k as well.

So, let $x = a_1 a_2 \ldots a_k$ be a string on which M reaches $\mathscr{M}$. Thus, as claimed in the proof of Lemma 5.1, there exists a sequence $\mathscr{M}_0, \mathscr{M}_1, \ldots, \mathscr{M}_k, \mathscr{M}_{k+1}$ such that:

(i) $\mathscr{M}_0$ is the initial crossing sequence,

(ii) for $0 \leq i \leq k$, $\mathscr{M}_i \rhd_{a_i} \mathscr{M}_{i+1}$, where we set $a_0 = \not{c}$,

(iii) $\mathscr{M}_{k+1} = \mathscr{M}$.

Since $k \geq n'$, by our *absurdum hypothesis* we get $d^{s(k) \cdot i(k)} < k$. Hence, a pigeonhole argument shows that $\mathscr{M}_i = \mathscr{M}_j$, for some $1 \leq i < j \leq k$. This implies that $\mathscr{M}$ can be reached even on the string $x' = a_1 \ldots a_i a_{j+1} \ldots a_k$, whose length is $k + i - j$.

In conclusion, we have shown that, for $s(n) \cdot i(n) \in o(\log n)$, there are *finitely many* (namely, no more than $d^{s(n') \cdot i(n')}$) crossing sequences generated by M on any input. Hence, *a fortiori*, the number of memory states reachable by M is *finite* as well. In other words, M can be simulated by a two–way alternating automaton, and hence, it recognizes a regular language, as stated by Ladner *et al.* (1984). This leads to a contradiction, and proves the theorem. ∎

If we collected in a table the lower bounds on the product $s(n) \cdot i(n)$ for deterministic, nondeterministic, and alternating Turing machines so far obtained, then we would get exactly the situation summarized in Table 4 for space lower bounds on one–way machines. This, together with Theorem 3.3, proves the optimality of such $s(n) \cdot i(n)$ lower bounds for general languages.

For what concerns *unary* languages, we again refer to space lower bound optimality for one–way machines given in Theorem 4.1. This yields the unary optimality of all the $s(n) \cdot i(n)$ lower bounds previously obtained, except for Π_1 middle, accept, weak machines, and alternating middle machines. For these cases, the optimality of the sublogarithmic lower bound in case of unary languages is still an open problem. The best we can prove is a "quasi–optimality" condition in the following sense:

Definition 5.1 *A lower bound $\ell(n)$ on a given computational resource is* quasi–optimal *for a class of machines $\mathscr{M}$ if, for any unbounded monotone increasing recursive function $f(n)$, there exists a language recognized by a machine $M \in \mathscr{M}$ that uses an amount $\rho(n) = O(\ell(n) \cdot f(n))$ of that computational resource.*

We then refer to the following theorem, proved by Geffert *et al.* (199X), and recall that middle, accept, and weak notions coincide for Π_1 machines, as observed in Section 2:

Theorem 5.4 *For any unbounded monotone increasing recursive function $f(n)$, there exists a unary nonregular language accepted by a* middle *Π_1 machine within $s(n)$ space and $i(n)$ input head reversals satisfying*

$$s(n) \cdot i(n) = O(\log \log n \cdot f(n)).$$

We summarize optimal lower bounds for $s(n) \cdot i(n)$ in the following

Theorem 5.5 *Let a deterministic, nondeterministic, Π_1, or general alternating machine accept a nonregular language in* strong, middle, accept, *or* weak *$s(n)$ space, and $i(n)$ input head reversals. Then, the product $s(n) \cdot i(n)$ must obey the lower bounds in the Table 6. Such bounds are tight for general languages. For unary languages, the bounds are tight except those marked with '•', which are* quasi–optimal.

	Strong	Middle	Accept	Weak
Deterministic	$\log n$	$\log n$	$\log n$	$\log n$
Nondeterministic (Σ_1)	$\log n$	$\log n$	$\log n$	$\log\log n$
Π_1	$\log n$	$\log\log n$ •	$\log\log n$ •	$\log\log n$ •
Alternating	$\log n$	$\log\log n$ •	$\log\log n$	$\log\log n$

TABLE 6. Optimal space×input head reversals lower bounds for machines accepting general and unary nonregular languages. Quasi–optimal lower bounds for unary nonregular acceptance are marked with '•'.

Before considering the alternating hierarchy, some brief comments.

As pointed out in Theorem 5.5, bounding the number of input head reversals leads to lower bounds on $s(n) \cdot i(n)$ that are analogous to those on space for one–way devices. This might give evidence of the relevance of input head reversals in nonregular acceptance. Such relevance is also stressed in the realm of unary alphabets. In fact, by allowing our devices the power of reversing the input head even only a bounded number of times, we are able to almost completely bridge (considering quasi–optimal situations in Theorem 5.5) the exponential gaps observed in Section 4 for one–way unary acceptance.

Furthermore, the power of reversing the input head enables us to prove both optimality and quasi–optimality of the lower bounds in Table 6 by using the minimal allowed amount of space:

Theorem 5.6 *The optimality of all the $s(n) \cdot i(n)$ lower bounds in Table 6 is witnessed by general nonregular languages recognizable within the* minimal *amount of space $O(\log\log n)$. For unary languages, the same space minimality property holds for both the optimality and quasi–optimality conditions.*

Proof. For logarithmic lower bounds, it is enough to consider the languages $\mathscr{L}$ and $\mathscr{L}^c$ which, according to Theorem 2.1, can be recognized by deterministic machines using strong $O(\log\log n)$ space, and performing $O\left(\frac{\log n}{\log\log n}\right)$ input head reversals.

Let us now turn to sublogarithmic lower bounds. For general alphabets, we refer to Theorem 3.3, where binary nonregular languages accepted in one–way $O(\log\log n)$ space are presented. For unary inputs, from Geffert *et al.* (199X), we learn that the languages in Theorem 5.4 witnessing quasi–optimality, are recognized in $O(\log\log n)$ space; in fact, the multiplicative term $f(n)$ is entirely due to input head reversals. Finally, in Theorem 4.1 (ii), we have shown a one–way nondeterministic machine accepting $\mathscr{L}^c$ in weak $O(\log\log n)$ space, while Theorem 4.1 (iii) states that $\mathscr{L}$ and $\mathscr{L}^c$ are recognized by one–way Σ_2 machines in accept (and hence weak) $O(\log\log n)$ space. ∎

In conclusion, from Theorems 5.5, 5.6 and trivial relationships among models of computation and space notions, we can easily obtain optimal and quasi–optimal lower bounds on the product $s(n) \cdot i(n)$ for the whole alternating hierarchy.

Theorem 5.7 *Let a Σ_k, Π_k, or general alternating Turing machine accept a nonregular language in* strong, middle, accept, *or* weak *$s(n)$ space, and $i(n)$ input head reversals. Then, $s(n) \cdot i(n)$ must obey the lower bounds in the diagrams in Figure 5. Such bounds are tight for general languages. For unary languages, the bounds are tight except those marked with '•' which are quasi–optimal. Again, optimality and quasi–optimality conditions can be met by using the minimal amount $O(\log\log n)$ of space.*

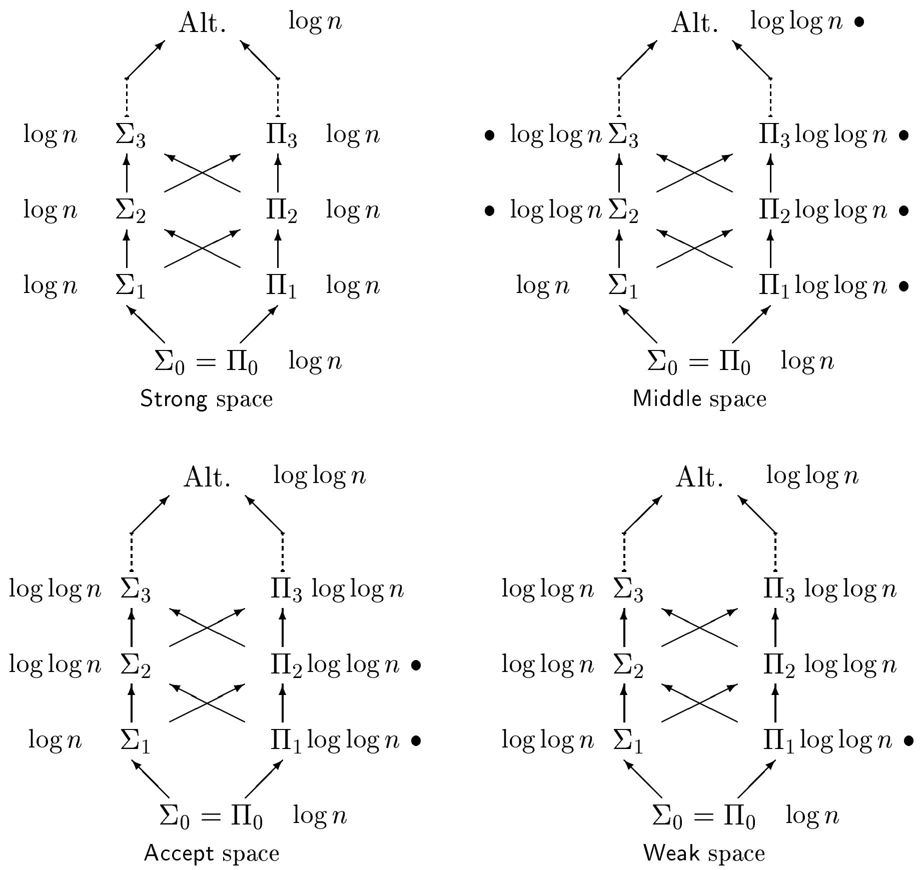

FIGURE 5. Space×input head reversals lower bounds for Σ_k and Π_k machines accepting general and unary nonregular languages. These bounds are tight for general nonregular languages. Quasi–optimal lower bounds for unary nonregular acceptance are marked with '•'.

References

ALBERTS, M. (1985), Space complexity of alternating Turing machines. In *Fundamentals of Computation Theory, Proceedings*, Lecture Notes in Computer Science, Vol. 199, pp. 1–7. Springer Verlag.

ALT, H., AND MEHLHORN, K. (1975), A language over a one symbol alphabet requiring only $O(\log\log n)$ space. *SIGACT news*, **7**, 31–33, 1975.

BERTONI, A., MEREGHETTI, C., AND PIGHIZZINI, G. (1994), On languages accepted with simultaneous complexity bounds and their ranking problem. In *Mathematical Foundations of Computer Science 1994, Proceedings*, Lecture Notes in Computer Science, Vol. 841, pp. 245–255. Springer Verlag.

BERTONI, A., MEREGHETTI, C., AND PIGHIZZINI, G. (1994a), An optimal lower bound for nonregular languages. *Information Processing Letters*, **50**, 289–292. Corrigendum. *ibid.*, **52**, 339.

BERTONI, A., MEREGHETTI, C., AND PIGHIZZINI, G. (1995), Strong optimal lower bounds for Turing machines that accept nonregular languages. In *Mathematical*

Foundations of Computer Science 1995, Proceedings, Lecture Notes in Computer Science, Vol. 969, pp. 309–318. Springer Verlag.

CHANDRA, A., KOZEN, D., AND STOCKMEYER, L. (1981), Alternation. *Journal of the ACM*, **28**, 114–133.

FREEDMAN, A., AND LADNER, R. (1975), Space bounds for processing contentless inputs. *Journal of Computer and System Sciences*, **11**, 118–128.

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

GEFFERT, V. (1991), Nondeterministic computations in sublogarithmic space and space constructibility. *SIAM J. Computing*, **20**, 484–498.

GEFFERT, V. (1993), Tally version of the Savitch and Immerman–Szelepcsényi theorems for sublogarithmic space. *SIAM J. Computing*, **22**, 102–113.

GEFFERT, V. (199X), Bridging across the $\log(n)$ space frontier. *Information and Computation.* To be published.

GEFFERT, V., MEREGHETTI, C., AND PIGHIZZINI, G. (199X), Sublogarithmic Bounds on Space and Reversals. *SIAM J. Computing.* To be published.

HENNIE, F. (1965), One–tape, off–line Turing machine computations. *Information and Control*, **8**, 553–578.

HONG, J.-W. (1984), A tradeoff theorem for space and reversal. *Theoretical Computer Science*, **32**, 221–224.

HOPCROFT, J., AND ULLMAN, J. (1969), Some results on tape–bounded Turing machines. *Journal of the ACM*, **16**, 168–177.

HROMKOVIČ, J., ROVAN, B., AND SLOBODOVÁ, A. (1994), Deterministic versus nondeterministic space in terms of synchronized alternating machines. *Theoretical Computer Science*, **132**, 319–336.

IWAMA, K. (1993), ASPACE($o(\log\log n)$) is regular. *SIAM J. Computing*, **22**, 136–146.

LADNER, R., LIPTON, R., AND STOCKMEYER, L. (1984), Alternating pushdown and stack automata. *SIAM J. Computing*, **13**, 135–155.

LEWIS, P., STEARNS, R., AND HARTMANIS, J. (1965), Memory bounds for the recognition of context free and context sensitive languages. In *IEEE Conf. Record on Switching Circuit Theory and Logical Design*, pp. 191–202.

LIŚKIEWICZ, M., AND REISCHUK, R. (1996), The sublogarithmic alternating space world. *SIAM J. Computing*, **25**, 828–861.

MEREGHETTI, C., AND PIGHIZZINI, G. (1995), A remark on middle space bounded alternating Turing machines. *Information Processing Letters*, **56**, 229–232.

SAVITCH, W. (1970), Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, **4**, 177–192.

SLOBODOVÁ, A. (1995), On the power of one–way globally deterministic synchronized alternating Turing machines and multihead automata. *International Journal of Foundations of Computer Science*, **6**, 431–446.

STEARNS, R., HARTMANIS, J., AND LEWIS, P. (1965), Hierarchies of memory limited computations. In *IEEE Conf. Record on Switching Circuit Theory and Logical Design*, pp. 179–190.

SUDBOROUGH, I. (1980), 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*, pp. 62–73.

SZEPIETOWSKI, A. (1988), Remarks on languages acceptable in $\log\log n$ space. *Information Processing Letters*, **27**, 201–203.

SZEPIETOWSKI, A. (1994), *Turing Machines with Sublogarithmic Space*. Lecture Notes in Computer Science, Vol. 843. Springer Verlag.