THE SUBLOGARITHMIC ALTERNATING SPACE WORLD∗
MACIEJ LIŚKIEWICZ† AND RÜDIGER REISCHUK‡

Abstract. This paper tries to fully characterize the properties and relationships of space classes defined by Turing machines that use less than logarithmic space – may they be deterministic, nondeterministic or alternating (DTM, NTM or ATM). We provide several examples of specific languages and show that such machines are unable to accept these languages. The basic proof method is a nontrivial extension of the $1^n \mapsto 1^n + n!$ technique to alternating TMs.

Let $l\log$ denote the logarithmic function $\log$ iterated twice, and $\Sigma_k \text{Space}(S)$, $\Pi_k \text{Space}(S)$ be the complexity classes defined by $S$–space-bounded ATMs that alternate at most $k - 1$ times and start in an existential, resp. universal state. Our first result shows that for each $k > 1$ the sets

$$\Sigma_k \text{Space}(l\log) \setminus \Pi_k \text{Space}(o(\log)) \quad \text{and} \quad \Pi_k \text{Space}(l\log) \setminus \Sigma_k \text{Space}(o(\log))$$

are both not empty. This implies that for each $S \in \Omega(l\log) \cap o(\log)$ the classes

$$\Sigma_1 \text{Space}(S) \subset \Sigma_2 \text{Space}(S) \subset \Sigma_3 \text{Space}(S) \subset \ldots$$

$$\subset \Sigma_k \text{Space}(S) \subset \Sigma_{k+1} \text{Space}(S) \subset \ldots$$

form an infinite hierarchy. Furthermore, this separation is extended to space classes defined by ATMs with a nonconstant alternation bound $A$ provided that the product $A \cdot S$ grows sublogarithmically.

These lower bounds can also be used to show that basic closure properties do not hold for such classes. We obtain that for any $S \in \Omega(l\log) \cap o(\log)$ and all $k > 1$ $\Sigma_k \text{Space}(S)$ and $\Pi_k \text{Space}(S)$ are not closed under complementation and concatenation. Moreover, $\Sigma_k \text{Space}(S)$ is not closed under intersection, and $\Pi_k \text{Space}(S)$ is not closed under union.

It is also shown that ATMs recognizing bounded languages can always be guaranteed to halt. For the class of $Z$–bounded languages with $Z \leq \exp S$ we obtain the equality $co-\Sigma_k \text{Space}(S) = \Pi_k \text{Space}(S)$.

Finally, for sublogarithmic bounded ATMs we give a separation between the weak and strong space measure, and prove a logarithmic lower space bound for the recognition of nonregular context-free languages.

Key words. space complexity, sublogarithmic complexity bounds, alternating Turing machines, halting computations, complementation of languages, complexity hierarchies, closure properties, context-free languages, bounded languages.

AMS subject classifications. 68Q05, 68Q10, 68Q25, 68Q45

1. Introduction. It is well known that if a deterministic or nondeterministic TM uses less than $l\log$ space then the machine can recognize only regular languages, and that there exist nonregular languages in $D\text{Space}(l\log)$. Therefore, let $\text{SUBLOG} := \Omega(l\log) \cap o(\log)$ denote the set of all nontrivial sublogarithmic space bounds, where $l\log$ abbreviates the twice iterated logarithmic function $n \mapsto \lfloor \log \log n \rfloor$. On the other hand, the logarithm seems to be the most dramatic bound for space complexity since most techniques used in space complexity investigations only work for bounds above this threshold. There are several important results for such space classes known, and it is an open question if they also hold for space bounds between $l\log$ and $\log$. One

∗ Most of the work was done while both authors were working at the Technische Hochschule Darmstadt, some parts were done by the first author at the University of Wrocław, and by the second author at the ICSI Berkeley, and at the Medizinische Universität zu Lübeck.
† Instytut Informatyki, Uniwersytet Wrocławski, Przesmyckiego 20, 51-151 Wrocław, Poland (liskiewicz@ii.uni.wroc.pl). The research of this author was supported by the Alexander-von-Humboldt-Stiftung and KBN grant 211979101.
‡ Institut für Theoretische Informatik, Medizinische Universität zu Lübeck, Wallstraße 40, 23560 Lübeck, Germany (reischuk@informatik.mu-luebeck.de).
of the most exciting problem of this type is whether the closure under complement for NTM
\[ NSpace(S) = co-NSpace(S) \]
shown by Immerman and Szépcecsenyi [13],[22] remains valid for sublogarithmic space bounds. If this equality were not valid for a function \( S \in \text{SUBLOG} \) then obviously \( DSpace(S) \subset NSpace(S) \).

A special situation holds for bounded languages containing only strings of a certain block structure.

**Definition 1.1.** Let \( Z : \mathbb{N} \rightarrow \mathbb{N} \) be a function. Then a language \( L \subseteq \{0,1\}^* \) is \( Z \)-bounded if each \( X \in L \) contains at most \( Z(|X|) \) zeros. \( L \) is bounded if it is \( Z \)-bounded for some constant function \( Z \).

Recently Alt, Geffert, and Mehlhorn ([2]) and independently Szepietowski ([23]) have proved that for the class of \( Z \)-bounded languages, where \( Z \) is a constant or a small growing function, the closure under complement holds, that means in this case \( NSpace(S) = co-NSpace(S) \) even for sublogarithmic bounds. Still, we conjecture that in general the above result does not hold. Towards this direction we will prove in this paper that \( \Sigma_kSpace(S) \) is not closed under complementation for any \( S \in \text{SUBLOG} \) and all \( k > 1 \).

Recall that for \( k \geq 1 \) the class \( \Sigma_kSpace(S) \) is defined as all languages that can be accepted by alternating \( S \) space-bounded TMs making at most \( k-1 \) alternations and starting in an existential state. \( \Pi_kSpace(S) \) denotes the set of languages accepted by the same kind of machines, except that they start in a universal state. By definition \( \Sigma_1Space(S) = NSpace(S) \). We will also consider ATMs with a non-constant bound \( A \) for the number of alternations. In this case, the notation \( \Sigma_ASpace(S) \) and \( \Pi_ASpace(S) \) is used.

By standard techniques it follows from Immerman-Szépcecsenyi’s theorem that for \( S \in \Omega(\log) \), and for all \( k \geq 1 \)
\[ \Sigma_1Space(S) = \Sigma_kSpace(S) = \Pi_kSpace(S). \]

Note that these techniques do not work for sublogarithmic space bounds. Recently, Chang et al. ([7]) have shown that there is a language in \( \Pi_2Space(llog) \) that does not belong to \( NSpace(o(\log)) \). Clearly, this proves that for space bounds \( S \) in \( \text{SUBLOG} \) the alternating \( S \)-space hierarchy does not collapse to the first level and that
\[ \Sigma_1Space(S) \subset \Pi_2Space(S). \]

It was left as an open problem whether the whole alternating hierarchy for sublogarithmic space is strict. Here we will prove that the problem has a positive answer.

We develop techniques to investigate properties of sublogarithmic computations and then generalize them to an inductive proof that the separation of the \( \Sigma_kSpace(S) \) and \( \Pi_kSpace(S) \) classes holds for all levels \( k \). The base case is the existence of a language that separates \( \Pi_2Space(llog) \) from \( \Sigma_2Space(o(\log)) \). Its complement separates \( \Sigma_2Space(llog) \) from \( \Pi_2Space(o(\log)) \).

Inductively we will construct a sequence of languages \( L_{\Sigma_k} \) and \( L_{\Pi_k} \) and prove that \( L_{\Sigma_k} \) can be recognized by a \( \Sigma_k \) TM with \( llog \) space, but not by any \( \Pi_k \) TM that

\footnote{In [25, p.419] it is incorrectly cited that \( DSpace(S) \subset NSpace(S) \), for \( S \in \text{SUBLOG} \), thus the problem if \( DSpace(S) = NSpace(S) \) is still open for any \( S \in \Omega(llog) \) (see Remark 6.1 in [17]).}
is \(o(\log)\)-space bounded. The corresponding claim interchanging \(\Sigma_k\) and \(\Pi_k\) holds for \(L_{\Pi_k}\). For this purpose, for infinitely many \(n\) we will explicitly pinpoint a pair of strings, one string in \(L_{\Sigma_k}\) and the other one in \(L_{\Pi_k}\), and show that any sublogarithmic space-bounded \(\Sigma_k\)TM or \(\Pi_k\)TM will make an error on at least one of these strings. Thus we obtain

**Theorem 1.2.** For all \(k > 1\) holds

\[
\Sigma_k \text{Space}(\log) \setminus \Pi_k \text{Space}(o(\log)) \neq \emptyset \quad \text{and} \\
\Pi_k \text{Space}(\log) \setminus \Sigma_k \text{Space}(o(\log)) \neq \emptyset.
\]

This result gives a complete and best possible separation for the sublogarithmic space world, except for the first level \(k = 1\). It is left open whether also \(\Sigma_1\)Space\((S) \neq \Pi_1\)Space\((S)\) for \(S \in \text{SUBLOG}\). The current techniques do not seem to be applicable to this case.

This separation implies that the sublogarithmic space hierarchy is an infinite one, contrary to the case for logarithmic or larger space bounds.

**Corollary 1.3.** For any \(S \in \text{SUBLOG}\) and all \(k \geq 1\) holds

\[
\Sigma_k \text{Space}(S) \subseteq \Sigma_{k+1} \text{Space}(S), \\
\Pi_k \text{Space}(S) \subseteq \Pi_{k+1} \text{Space}(S).
\]

Independently the existence of this strict hierarchy has been shown by von Braunm"uhl with coauthors [6]. Geffert [11] has announced similar results. (For a chronology of events see [24].)

Furthermore, we can generalize the separation to machines with an unbounded number of alternations.

**Definition 1.4.** A function \(A: \mathbb{N} \to \mathbb{N}\) is **computable in space** \(S\) if there exists a DTM that for all inputs of the form \(1^n\) writes down the binary representation of \(A(n)\) on an extra output tape using no more than \(S(n)\) work space. \(A\) is **approximable from below in space** \(S\) if there exists a function \(A'\) that is computable in space \(S\) with \(A'(n) \leq A(n)\) for all \(n \in \mathbb{N}\) and \(A'(n) = A(n)\) for infinitely many \(n \in \mathbb{N}\).

The class of bounds that are approximable from below in space \(\log\) contains functions of logarithmic and double-logarithmic growth and also polynomials of such functions. The iterated logarithm \(\log^*\) belongs to this class, too. In section 3 we will discuss a specific example of logarithmic growth.

**Theorem 1.5.** For any pair of functions \(S \in \text{SUBLOG}\) and \(A > 1\) and \(A \cdot S \in o(\log)\), where \(A\) is approximable from below in space \(S\), holds:

\[
\Sigma_A \text{Space}(S) \setminus \Pi_A \text{Space}(S) \neq \emptyset, \\
\Pi_A \text{Space}(S) \setminus \Sigma_A \text{Space}(S) \neq \emptyset.
\]

**Corollary 1.6.** For any \(S\) and \(A\) as in the theorem above holds:

\[
\Sigma_A \text{Space}(S) \subseteq \Sigma_{A+1} \text{Space}(S), \\
\Pi_A \text{Space}(S) \subseteq \Pi_{A+1} \text{Space}(S).
\]

Thus one obtains for space bounds \(S \in \Omega(\log)\) and approximable functions \(A\) for example the following relations:
1. $\bigcup_{k \in \mathbb{N}} \Sigma_k \text{Space}(S) \subset \Sigma_{\log^k} \text{Space}(S)$ if $S \in o(\frac{\log}{\log^k})$.
2. $\Sigma_A \text{Space}(S) \subset \Sigma_{A+1} \text{Space}(S)$ for $A, S \in O(\log^{1/2-\epsilon})$.
3. For $k \in \mathbb{N}$ let

$$\text{ALSL}^k := A\text{AlterSpace}(\log^k, \log) .$$

Then for any $k$ holds $\text{ALSL}^k \subset \text{ALSL}^{k+1}$.

Note that for logarithmic bounds the corresponding question is still open, i.e. for any $k$ it is unknown whether

$$A\text{AlterSpace}(\log^k, \log) \subset A\text{AlterSpace}(\log^{k+1}, \log) ?$$

It is well known that for any function $S$ the complexity class $\Sigma_1 \text{Space}(S)$ is closed under union and intersection (see e.g. [25]). However, it is still an open problem whether for $S \in \text{SUBLOG}$ the class $\Sigma_1 \text{Space}(S)$ is closed under complementation. More general, for arbitrary $k$ the classes $\Sigma_k \text{Space}(S)$ are closed under union, and symmetrically the $\Pi_k \text{Space}(S)$ are closed under intersection. In [14] we have developed a technique showing that for $S \in \text{SUBLOG}$ and for $k = 2, 3, \Sigma_k \text{Space}(S)$ and $\Pi_k \text{Space}(S)$ are not closed under complementation. Furthermore, $\Sigma_k \text{Space}(S)$ is not closed under intersection, and $\Pi_k \text{Space}(S)$ not under union. Combining these ideas with the separation results above we get the same closure properties for all levels.

**Theorem 1.7.** For any $S \in \text{SUBLOG}$ and all $k > 1 \Sigma_k \text{Space}(S)$ and $\Pi_k \text{Space}(S)$ are not closed under complementation and concatenation. Moreover, $\Sigma_k \text{Space}(S)$ is not closed under intersection and $\Pi_k \text{Space}(S)$ is not closed under union.

Note that non-closure under complementation for $\Sigma_k$ and $\Pi_k$ classes is not trivially equivalent to Theorem 1.2, which says that sublogarithmic $\Sigma_k \text{Space}$ and $\Pi_k \text{Space}$ are distinct. Sublogarithmic space-bounded machines do not have a counter, which could detect an infinite path of computation. It is an interesting open problem whether $\Pi_k \text{Space}(S) = \text{co-}\Sigma_k \text{Space}(S)$ for $k = 1, 2, \ldots$ (see the discussion in [14]). Here, we obtain the following partial solution generalizing Sipser’s result on halting space-bound computation for sublogarithmic space bounded deterministic TMs [19]: For bounded languages it can be shown that there exist equivalent ATMs that always halt. This implies

**Theorem 1.8.** Let $S \in \text{SUBLOG}$ be a space bound and $Z$ be a function computable in space $S$ with $Z \leq \exp S$. Then for all $k \geq 1$ and for every $Z$–bounded language $L \subseteq \{0, 1\}^*$ holds:

$$L \in \Sigma_k \text{Space}(S) \iff T \in \Pi_k \text{Space}(S) .$$

Observe that for $S \geq \log$ the function $Z$ can grow linearly and then $Z$ does not put any restriction on the structure of the strings in $L$. Thus, this theorem gives a smooth approximation of the fact that for at least logarithmic space bounds $\Sigma_k$ and $\Pi_k$ are complementary for arbitrary languages. We conjecture that the computability of $Z$ is needed in the claim above. Furthermore, there are some indications that the theorem might not be true in general for bounds $Z$ much larger than $\exp S$.

Finally, we prove a logarithmic lower space bound for the recognition of context-free languages by ATMs. We will show that the deterministic context-free language

$$L_\# := \{1^n01^m \mid n \neq m\}$$

does not belong to $A\text{Space}(o(\log))$. It is interesting to note that this language – but not its complement – can be recognized even by a deterministic machine in weak space $\log$.
**Definition 1.9.** We say that an ATM $M$ is (strongly) $S$ space-bounded if on every input $X$ it only enters configurations that use at most $S(|X|)$ space. $M$ is weakly $S$ space-bounded if, for every input $X$ that is accepted, it has an accepting computation tree all of which configurations use at most $S(|X|)$ space. $\text{DSpace}(S)$ denotes the class of languages accepted by $S$ space-bounded DTMs and $\text{weakDSpace}(S)$ denotes the languages accepted by weakly $S$ space-bounded DTMs. A corresponding notation is used for NTMs and ATMs.

In this paper we consider only the more natural strong requirement for space complexity. For at least logarithmic space bounds the two conditions do not make a difference, while in the sublogarithmic case they obviously do. When studying the closure under complement of a language $L$ and alternating hierarchies built on this the weak measure is not appropriate. This is because for strings in $\overline{L}$ a machine for $L$ may use arbitrary much space, while a machine for $\overline{L}$ were required to be bounded.

The example above shows that with respect to the weak measure already for DTM $\text{weakDSpace}(\log)$ contains languages that do not belong to $\text{co-weakDSpace}(o(\log))$.

In [7] Chang et al. stated as an open problem whether weak and strong sublogarithmic space-bounded ATMs have the same power. Obviously, our lower space bound for recognizing $L \neq \emptyset$ by ATMs proves the following

**Theorem 1.10.** $\text{weakDSpace}(\log) \setminus \text{ASpace}(o(\log)) \neq \emptyset$.

As consequences one obtains

**Corollary 1.11.** For any $k \geq 1$ and each $S \in \text{SUBLOG}$

\[ \Sigma_k \text{Space}(S) \subset \text{weak}\Sigma_k \text{Space}(S) \quad \text{and} \quad \Pi_k \text{Space}(S) \subset \text{weak}\Pi_k \text{Space}(S). \]

**Corollary 1.12.** For each $S \in \text{SUBLOG}$

\[ \text{ASpace}(S) \subset \text{weakASpace}(S). \]

We next generalize the specific lower bound above to arbitrary deterministic context-free languages, which also improves a result for NTMs shown by Alt, Mehlhorn and Geffert [2]. Before stating the result we need the following definition (see [20] and [12]). A language $L$ is called strictly nonregular if one can find strings $u, v, w, x$ and $y$ such that $L \cap \{u\} \{v\}^* \{w\} \{x\}^* \{y\}$ is context-free, but nonregular.

**Theorem 1.13.** Let $L$ be a nonregular deterministic context-free, a strictly nonregular language, or a nonregular context-free bounded language, then

\[ L \notin \bigcup_{k \in \mathbb{N}} \Sigma_k \text{Space}(o(\log)). \]

Furthermore, for ATMs without any bound on the number of alternations it is not possible that $L$ and $\overline{L}$ both belong to $\text{ASpace}(o(\log))$.

This paper is organised as follows. In the next section the necessary technical tools for sublogarithmic space bounded ATMs will be developed. In section 3 we will define a sequence of pairs of languages indexed by the level number $k$ to prove the sublogarithmic space hierarchy. We then investigate closure properties of sublogarithmic space classes. Section 5 is devoted to the lower space bounds for context-free languages. The paper concludes with a discussion of the most interesting open problems for sublogarithmic space classes remaining.

Preliminary versions of most of these results have been presented in [14] and [15].
2. Properties of Sublogarithmic Space-Bounded ATMs. The Turing machine model we consider is equipped with a two-way read-only input tape and a single read-write work tape. Moreover the input word is stored on the input tape between end-markers $\$.

Definition 2.1. A memory state of a TM $M$ is an ordered triple $\alpha = (q, u, i)$, where $q$ is a state of $M$, $u$ a string over the work tape alphabet, and $i$ a position in $u$ (the location of the work tape head). A configuration of $M$ on an input $X$ is a pair $(\alpha, j)$ consisting of a memory state $\alpha$ and a position $j$ with $0 \leq j \leq |X| + 1$ of the input head. $j = 0$ or $j = |X| + 1$ means that this head scans the left, resp. the right end-marker. For a memory state $\alpha = (q, u, i)$ let $|\alpha|$ denote the length of the memory inscription $u$.

We may assume that for a successor $(\alpha', j')$ of a configuration $(\alpha, j)$ always holds $|\alpha'| \geq |\alpha|$. The state set of an ATM is partitioned into subsets of existential, universal, accepting, and rejecting states. We say that a configuration $((q, u, i), j)$ is existential (resp. universal, accepting, or rejecting) if $q$ has the corresponding mode. All accepting and rejecting configurations $C$ are assumed to be terminating, i.e. there are no more configurations that can be reached from $C$.

Definition 2.2. Let

$$(\alpha, i) \models^*_{M, X} (\beta, j)$$

denote the property that the ATM $M$ with $X$ on its input tape has a computation path $C_1 = (\alpha, i), C_2, \ldots, C_t = (\beta, j)$.

$$(\alpha, i) \models_{M, X} (\beta, j)$$

denotes the same fact, but with the following restriction: $t \geq 2$ and the mode of the configurations $C_2, \ldots, C_{t-1}$ is the same as that of $C_1$ (i.e. if $C_1$ is existential then all $C_l$ for $l = 2, \ldots, t - 1$ are existential, otherwise they are all universal).

$acc^k_M(\alpha, i, X)$

denotes the predicate saying that $M$ starting in configuration $(\alpha, i)$ with $X$ on its input tape accepts (i.e. has an accepting subtree), and on each computation path of that tree it makes at most $k - 1$ alternations. Let

$Space_M(\alpha, i, X)$

denote the maximum space used in configurations $M$ can reach on input $X$ starting in configuration $(\alpha, i)$ and $Space_M(X) := Space_M(\alpha_0, 0, X)$, where $(\alpha_0, 0)$ is the initial configuration of $M$. Similarly let

$Alter_M(\alpha, i, X)$

denote the maximum number of alternations $M$ can make on input $X$ starting in configuration $(\alpha, i)$ and $Alter_M(X) := Alter_M(\alpha_0, 0, X)$.

2.1. Inputs of a Periodic Structure. In this section some properties of TM computations for binary inputs of the form $Z_1WW \ldots WZ_2$ will be described. Let $M$ be an ATM. Then for any integer $b \geq 0$ we define

$\mathcal{M}_b := \# \{ \alpha \mid \alpha \text{ is a memory state of } M \text{ with } |\alpha| \leq b \}$.
The following two Lemmata characterize "short" computations i.e. computations restricted to substrings $WW\ldots W$. The first one is a generalization of a result in [16].

**Lemma 2.3.** Assume that

$$X = Z_1 W^n Z_2$$

where $Z_1, W, Z_2$ are arbitrary binary strings and $n \in \mathbb{N}$. Moreover let $b$ be an integer and $(\alpha, i)$ and $(\beta, j)$ configurations with $|\alpha| \leq |\beta| \leq b$ and $|Z_1| < i, j \leq |Z_1| W^n|$. Then the following holds:

- If $M$ can go from $(\alpha, i)$ to $(\beta, j)$ without any alternation and without moving the input head out of the substring $W^n$ then $M$ can also do so such that the head never moves $M_b \cdot |W|$ or more positions to the left of $\min(i, j)$ nor to the right of $\max(i, j)$.

**Proof.** We only sketch the main idea. Assume $i \leq j$ and denote by $i_{\min}$ and $j_{\max}$ the furthest position to the left, resp. right of the input head in the computation path of $M$ that starts in $(\alpha, i)$ and ends in $(\beta, j)$. Let $M$ go from $(\alpha, i)$ to $(\beta, j)$ moving the input head $M_b \cdot |W|$ or more positions to the right of $j$, i.e. $j_{\max} - j \geq M_b \cdot |W|$. By the pigeon hole principle there exist two positions $j_1$ and $j_2$, with $j < j_1 < j_2 < j_{\max}$ and two memory states $\alpha'$ and $\alpha''$ such that $(\alpha', j_1)$ and $(\alpha', j_2)$ are the last configurations of the computation path from $(\alpha, i)$ to a configuration in which $M$ was at the position $j_{\max}$. Similarly, $(\alpha'', j_1)$ and $(\alpha'', j_2)$ are the first configurations of the computation from the position $j_{\max}$ to $(\beta, j)$. Then removing the computation paths from $(\alpha', j_1)$ to $(\alpha', j_2)$ and from $(\alpha'', j_2)$ to $(\alpha'', j_1)$ one obtains a computation that starts in $(\alpha, i)$ and ends in $(\beta, j)$ with the head never moving more than distance $j_{\max} - (j_2 - j_1) < j_{\max}$ to the right of position $j$.

**Lemma 2.4.** Let $|i - j| \geq M_b^2 (M_b + 1) \cdot |W|$. Assume that $M$ can go from configuration $(\alpha, i)$ to configuration $(\beta, j)$

- without alternating and without leaving the region between the input positions $i$ and $j$.

Then,

- there exists an integer $c \in [1..M_b]$ such that for all $d \in [1..M_b]$ there is a computation path satisfying (♠) which starts in configuration $(\alpha, i)$ and ends in $(\beta, j - d \cdot \sgn(j - i) \cdot c \cdot |W|)$, where $\sgn(z) := z/|z|$.

- Moreover, there also exists a computation path satisfying (♠) that starts in $(\alpha, i + d \cdot \sgn(j - i) \cdot c \cdot |W|)$ and ends in $(\beta, j)$.

**Proof.** In the following we will only discuss the case $i < j$ when considering the computation from configuration $(\alpha, i)$ to $(\beta, j)$. Let $|i - j| \geq M_b^2 (M_b + 1) \cdot |W|$. Define for integers $p \geq 0$ the function $h(p) := i + p \cdot |W|$ and let $t := M_b^2$. Partition integers in $[1\ldots M_b]$ into the $t$ intervals $[L_1, R_1], [L_2, R_2], \ldots, [L_t, R_t]$ of equal length $M_b$ with boundaries

$$L_s := (s - 1)(M_b + 1) + 1$$

$$R_s := L_s + M_b.$$ 

For $s \in [1\ldots t]$ consider all input positions $h(p)$ with $p \in [L_s, R_s]$ and the last configuration of $M$ (before $(\beta, j)$) that visits position $h(p)$. Among these $M_b + 1$ configurations there must exist a pair with positions $p_1 < q_1 \in [L_s, R_s]$ and identical memory states $\alpha_s$, configuration.

Let $(\gamma_1, i_1) \models_{i,j} (\gamma_2, i_2)$ denote the same property as $(\gamma_1, i_1) \models_{M,X} (\gamma_2, i_2)$, but with the restriction that $M$ going from $(\gamma_1, i_1)$ to $(\gamma_2, i_2)$ does not move the head.
to the left of \( i \) nor to the right of \( j \). Then we can write:

\[
\begin{align*}
(\alpha, i) \models_{i,j} (\alpha_1, h(p_1)) & \models_{i,j} (\alpha_1, h(q_1)) \models_{i,j} \\
(\alpha_2, h(p_2)) & \models_{i,j} (\alpha_2, h(q_2)) \models_{i,j} \\
\cdots & \\
(\alpha_t, h(p_t)) & \models_{i,j} (\alpha_t, h(q_t)) \models_{i,j} (\beta, j).
\end{align*}
\]

Since there are \( t \) pairs \( (p_s, q_s) \) and the difference between any pair is at most \( M_b \), by the pigeon hole principle there exists an integer \( c \in [1 \ldots M_b] \) and \( t/M_b = M_b \) pairs \( (p_s, q_s), (p_{s2}, q_{s2}), \ldots \) with identical difference \( c \), that means \( q_{si} - p_{si} = c \) for \( \ell = 1, 2, \ldots, M_b \). Define \( \Delta' := c \cdot |W| \).

Let \( d \) be an arbitrary integer in \([1 \ldots M_b]\) and define \( \alpha'_\ell := \alpha_{si} \) and \( i_\ell := h(p_{si}) \). Then we obtain:

\[
\begin{align*}
(\alpha, i) & \models_{i,j} (\alpha'_1, i_1) \models_{i,j} (\alpha'_1, i_1 + \Delta') \models_{i,j} \\
(\alpha'_2, i_2) & \models_{i,j} (\alpha'_2, i_2 + \Delta') \models_{i,j} \\
\cdots & \\
(\alpha'_{d'}, i_{d'}) & \models_{i,j} (\alpha'_{d'}, i_{d'} + \Delta') \models_{i,j} (\alpha'_{d'+1}, i_{d'+1}) := (\beta, j)
\end{align*}
\]

The input \( X \) contains a sequence of identical blocks \( W \) between the positions \( i \) and \( j \). For any \( \ell \in [1 \ldots d'] \), \( M \) starting in \( (\alpha'_\ell, i_\ell + \Delta') \) reaches \( (\alpha'_{\ell+1}, i_{\ell+1}) \) without moving the head to the left of \( i_\ell + \Delta' \). Therefore \( M \) making the same sequence of moves reaches \( (\alpha'_{\ell+1}, i_{\ell+1} - \ell \Delta') \) when starting in \( (\alpha'_\ell, i_\ell + (\ell - 1)\Delta') \). Thus we obtain

\[
\begin{align*}
(\alpha, i) & \models_{i,j} (\alpha'_1, i_1) \models_{i,j} (\alpha'_1, i_1 + \Delta') \models_{i,j} \\
(\alpha'_2, i_2 - \Delta') & \models_{i,j} (\beta, j - \Delta') \models_{i,j}
\end{align*}
\]

which proves that \( (\alpha, i) \models_{i,j} (\beta, j - \Delta) \) for \( \delta := d \cdot c \cdot |W| \). In a similar way, one can show that there exists a computation path that starts in configuration \( (\alpha, i + \Delta) \) and ends in \( (\beta, j) \).

In the following \( M \) will always denote an arbitrary ATM and \( S \) a space bound in \( o(\log n) \). Depending on \( M \) and \( S \), we choose a constant \( N_{M,S} \geq 2^8 \) such that for all \( n \geq N_{M,S} \)

\[
(M_{S(n)}^6 + 1)^2 < n
\]

and

\[
S(n) < \frac{1}{2} \log n - 2.
\]

**Remark.** In this section all claims following hold for any integer \( n \geq N_{M,S} \).

In [8] Geffert has shown that for sublogarithmic space bounded computations for any natural number \( \ell \) the behavior of a nondeterministic TM on input \( 1^{n+\ell n!} \) is exactly the same as on \( 1^n \). The proof is based on the so called “\( n \mapsto n + n! \) technique” developed by Stearns, Hartmanis, and Lewis in [21]. We will show that a corresponding property holds for ATMs and for all inputs of the form

\[
X = Z_1 W^n Z_2 \quad \text{and} \quad Y = Z_1 W^{n+\ell n!} Z_2.
\]
where \( Z_1, Z_2, W \) are arbitrary binary strings and \( \ell \in \mathbb{N} \).

Since in the following we will often compare computations on such an input \( X \) and a pumped version \( Y \) let us introduce a special notation for positions within these strings. If \( i \) is a position within \( X \) outside the pumped region \( W^n \), that means for the example above either in \( Z_1 \) or in \( Z_2 \), then \( i \) denotes the corresponding position within \( Y \). Thus

\[
i := \begin{cases} i & \text{if } i \leq |Z_1|, \\ i + |Y| - |X| & \text{if } i > |Z_1W^n|.
\end{cases}
\]

The main technical tools for the analysis of sublogarithmic space-bounded ATMs are stated in the following Lemmata. Here, \( X \) and \( Y \) denote strings as defined above and \( M \) an arbitrary ATM. Note that \( n \) now is not necessarily identical to the length of the input \( X \). Actually, \( X \) will in general be much larger than \( n \). But by a repeated application of the following implications we can show that any machine \( M \) still obeys a sublogarithmic bound with respect to \( n \).

**Lemma 2.5 (Pumping).** Let \( \alpha, \beta \) be memory states with \( |\alpha| \leq |\beta| \leq S(n) \), then for any \( i, j \in [0 \ldots |Z_1|] \cup [\ldots |Z_1W^n| + 1 \ldots |X| + 1] \) holds:

1. \( (\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, i) \models_{M,Y} (\beta, j) \),
2. \( (\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, i) \models_{M,Y} (\beta, j) \).

In the analysis below we will use the Pumping Lemma in the following more general form:

**Lemma 2.6.** Let \( n \) and \( m \) be integers with \( N_{M,S} \leq m \leq (n + 1)^2 \) and let \( \alpha, \beta \) be memory states with \( |\alpha| \leq |\beta| \leq S(m) \). Then for any \( i, j \in [0 \ldots |Z_1|] \cup [\ldots |Z_1W^n| + 1 \ldots |X| + 1] \) the properties 1. and 2. above hold.

These claims can be proven using the method developed in [8] and the fact that \( M_{S(n)}^{n} \leq n \).

**2.2. Space and Alternation Bounds.**

**Lemma 2.7 (Small Space Bound).**

\[
\text{Space}_M(X) \leq S(n) \implies \text{Space}_M(Y) = \text{Space}_M(X).
\]

**Proof.** Let \( \text{Space}_M(X) \leq S(n) \). Assume, to the contrary, that \( \text{Space}_M(Y) \neq \text{Space}_M(X) \). We will show that \( \text{Space}_M(Y) > \text{Space}_M(X) \) cannot occur. A similar contradiction can be obtained for the case \( \text{Space}_M(Y) < \text{Space}_M(X) \).

Assume that \( \text{Space}_M(Y) > \text{Space}_M(X) \). Hence, for \( Y \) there exists a computation path \( C \) that starts in the initial configuration \( (\alpha_0, 0) \) and ends in a configuration \( (\alpha, \hat{j}) \) with \( |\alpha| = \text{Space}_M(X) \) such that from \( (\alpha, \hat{j}) \) \( M \) can reach a configuration \( (\beta, \hat{j}') \) with \( |\beta| = \text{Space}_M(X) + 1 \) in one step:

\[
(\alpha_0, 0) \models_{M,Y}^* (\alpha, \hat{j}) \models_{M,Y}^* (\beta, \hat{j}').
\]

If \( j \) fulfills the condition \( j \leq |Z_1| \) or \( j > |Z_1W^n| \) of the Pumping Lemma then one can conclude immediately:

\[
(\alpha_0, 0) \models_{M,X}^* (\alpha, j) \models_{M,X}^* (\beta, j').
\]

Otherwise, using a similar pumping argument one can show that \( M \) on input \( X \) can reach a configuration \( (\alpha, \hat{j}) \), in which the input head is located on \( W^n \) and reads the same symbol as in \( (\alpha, \hat{j}) \). Thus it can also get to memory state \( \beta \) in one more step. We get a contradiction since \( |\beta| > \text{Space}_M(X) \). \( \Box \)
LEMMA 2.8 (Small Alternation Bound).

\[ \text{Space}_M(X) \leq S(n) \text{ and } \text{Alter}_M(X) \leq \exp S(n) \implies \text{Alter}_M(Y) = \text{Alter}_M(X). \]

Proof. Let \( i \) be an integer, with \( i \in \{|Z_1|, |Z_1W^n| + 1\} \) and let \( \alpha \) be a memory state, with

\[ \text{Space}_M(\alpha, i, X) \leq S(n) \quad \text{and} \quad \text{Alter}_M(\alpha, i, X) \leq \exp S(n). \]

Assume that \( k \) is an arbitrary positive integer and let

\[ \delta_k := k \cdot M_b^2 \cdot (M_b + 1) \cdot |W|, \]

where \( b := S(n) \). We first show that for the input \( Y \) the following claim holds:

Claim 1. Let \( M \) starting in \((\alpha, i)\) alternate \( k-1 \) times and never move the input head beyond \( W^{n+\ell n!} \). Then there exists a computation of \( M \) with \( k-1 \) alternations that also starts in \((\alpha, i)\), but in which the input head is never moved farther than \( \delta_k \) positions to the right of \( i \) if \( i = |Z_1| \), resp. to the left of \( i \) if \( i = |Z_1W^n+\ell n| + 1 \).

Proof. We show this claim for \( i = |Z_1| \). The case \( i = |Z_1W^n+\ell n| + 1 \) can be treated similarly. Let us note first that for integers \( k \) such that \( \delta_k \geq n + \ell n! \) the claim holds trivially. Therefore in the proof below we consider only \( k \) with \( \delta_k < n + \ell n! \).

Let \( i' \) be the smallest integer such that \( M \) starting in \((\alpha, i)\) makes \( k-1 \) alternations with the head never moving to the left of \( i \) nor to the right of \( i' \). Assume, to the contrary that \( i' > i + \delta_k \). Therefore by the pigeon hole principle there is an interval \([L, R]\), with

\[ i + M_b^2 \cdot |W| \leq L < R \leq i' \quad \text{and} \quad R - L \geq M_b^2 \cdot (M_b + 1) \cdot |W|, \]

and a computation path \( C \) of \( k-1 \) alternations such that \( M \) with the head position in \([L, R]\) does not alternate.

Let \( C' \) be a subsequence of configurations of \( C \) of the maximal length such that all configurations of \( C' \) have the head position greater or equal to \( L \) and there is a configuration in \( C' \) with the head position \( i' \). Note that the first configuration of \( C' \) equals \((\alpha_L, L)\), for some memory state \( \alpha_L \). Moreover there is a configuration in \( C' \) with the head position \( R \). Let \((\alpha_R, R)\) denote the first such configuration.

Below we show how to cut and paste \( C' \) to obtain a computation path of the same number of alternations but with the head never reaching the position \( i' \). This yields a contradiction to the assumption that \( i' > i + \delta_k \).

Let us consider first that \( C' \) is a tail of \( C \). By Lemma 2.4 there exists a constant \( c \), with \( 1 \leq c \leq M_b \) such that \( M \) starting in \((\alpha_L, L)\) reaches \((\alpha_R, R - c \cdot |W|)\), with the head positions in \([L, R]\). If additionally \( M \) starting in \((\alpha_R, R - c \cdot |W|)\) makes the same sequence of moves as in \( C' \) when started in \((\alpha_R, R)\) then we obtain a computation for \( M \) with the same number of alternations as in \( C' \) but with the head never moving to the right of \( i' - c \cdot |W| \).

Assume now that \( C' \) is not a tail of \( C \). Then the last configuration of \( C' \) has a form \((\alpha_L', L)\), for some memory state \( \alpha_L' \). Let \((\alpha_R', R)\) be the last configuration in \( C' \) with the head position \( R \). By Lemma 2.4 there exist constants \( c_1, c_2 \), with \( 1 \leq c_1, c_2 \leq M_b \) such that \( M \) starting in \((\alpha_L, L)\) reaches \((\alpha_R, R - c_1c_2 \cdot |W|)\) and starting in \((\alpha_R', R - c_1c_2 \cdot |W|)\) reaches \((\alpha_L', L)\). It is obvious that \( M \) starting in \((\alpha_R, R - c_1c_2 \cdot |W|)\) and making the same sequence of moves as between \((\alpha_R, R)\) and \((\alpha_R', R)\) in \( C' \), reaches \((\alpha_R', R - c_1c_2 \cdot |W|)\). Hence we obtain a computation path of the
same number of alternations as in $C'$ that starts and ends also in $(\alpha_L, L)$ and $(\alpha'_L, L)$, resp. but with the head never moving to the right of $i' - c_1c_2 \cdot |W|$. □

Note that by Claim 1 and the assumption that $\text{Alter}_M(\alpha, i, X) \leq \exp S(n)$ it follows that if $M$ with $Y$ on the input tape starts in $(\alpha, i)$ and makes $k - 1$ alternations with the head never moved beyond $W^{n + \ell n!}$ then $k - 1 \leq \exp S(n)$. To see this assume the opposite. Then by Claim 1 $M$ starting in $(\alpha, i)$ makes $k - 1 = \exp S(n) + 1$ alternations such that the head is never moved farther than $\delta_k$ positions from $i$. By the assumption that $n \geq N_{M,S}$ we conclude:

$$\delta_k = (2^{S(n)} + 2) \cdot |M_b + 1| \cdot |W| \leq 2^{\frac{\log n}{2}} (M_b^3 + 1) \cdot |W| \leq n^{1/2} n^{1/2} |W| = n \cdot |W|$$

which means that $M$ can make the same computation on $X$. We obtain a contradiction since $\text{Alter}_M(\alpha, i, X) \leq \exp S(n)$. Hence our lemma follows from Claim 1 and from the following

Claim 2. For $k - 1 \leq \exp S(n)$ and for any memory state $\beta$ and for any integer $j \in \{\lfloor |Z_1|, |Z_1 n^m + 1| \}$ holds:

$M$ starting in $(\alpha, i)$ with $X$ on the input tape reaches $(\beta, j)$ with $k - 1$ alternations iff $M$ starting in $(\alpha, i)$ with the input $Y$ reaches $(\beta, j)$ with $k - 1$ alternations.

Proof. We prove the claim for $i = |Z_1| \text{ and } j = |Z_1 n^m + 1|$. In the other cases a similar proof can be used.

Assume that on input $X$ $M$ reaches $(\beta, j)$ from $(\alpha, i)$ making $k - 1$ alternations. Since

$$k \leq \exp S(n) + 1 \leq \lfloor \sqrt{n} \rfloor / 2,$$

there exist non-negative integers $n_1, n_2$ and $n_3$ with

\[(i) \quad n_1 + n_2 + n_3 = n \quad \text{and} \quad |\sqrt{n}| \leq n_2 \leq n\]

such that $M$ alternates only on the prefix $Z_1 n^{m_1}$ and on the suffix $W^{n_3 n_2}$, but not on $W^{n_2}$. By Lemma 2.6, for

$$i', j' \in [0 \ldots |Z_1 n^{m_1}|] \cup [|Z_1 n^{m_1} + n_2| + 1 \ldots |X| + 1],$$

and $m' := n$, $n' := n_2$, $\ell' := \ell n(n - 1) \ldots (n_2 + 1)$

$$(\alpha', i') \models_{M, X} (\beta', j') \iff (\alpha', i') \models_{M, Y} (\beta', j')$$

for any configurations $(\alpha', i')$ and $(\beta', j')$ that are reachable by $M$ on the computation path between $(\alpha, i)$ and $(\beta, j)$. Using this property one can easily obtain a path with $(k - 1)$ alternations for input $Y$ that starts in $(\alpha, i)$ and ends in $(\beta', j')$.

On the other hand if for integers $n_1, n_2, n_3$ fulfilling (i) there is a computation path for $M$ on $Y$ which starts in $(\alpha, i)$ and ends in $(\beta, j)$ and such that $M$ does not alternate with the head position in $[|Z_1 n^{m_1}| + 1 \ldots |Z_1 n^{m_1} + n_2 + \ell' n_2|]$ then, applying Lemma 2.6 in the same way as above, one can construct a computation path for the input $X$ which starts and ends in $(\alpha, i)$ and $(\beta, j)$, resp. and has the same number of alternations. Therefore, to complete the proof we have to show that there exists such a computation path for $Y$ if we assume that $M$ started in $(\alpha, i)$ reaches $(\beta, j)$ making $k - 1$ alternations.

Let $m$ be the largest integer such that for some $n_1, n_3 \in \mathbb{N}$, with $n_1 + m + n_3 = n + \ell n!$, there is a computation path $C$ between $(\alpha, i)$ and $(\beta, j)$ of $k - 1$ alternations.
such that $M$ alternates only on the prefix $Z_1 W^{n_1}$ and suffix $W^{n_3} Z_2$. Assume to the contrary that

$$m < \lfloor \sqrt{n} \rfloor + \ell' \lfloor \sqrt{n} \rfloor!,$$

where $\ell' := \ell n(n-1) \ldots \lfloor (\sqrt{n})! \rfloor$. Then either in $W^{n_1}$ or in $W^{n_3}$ there exists a substring of the form $W^{m'}$, with $m' \geq 2 \lfloor \sqrt{n} \rfloor$, such that $M$ does not alternate on $W^{m'}$, too. W.l.o.g. let $W^{m'}$ be a substring of $W^{n_3}$. Then $W^{n_3} = W^{n_3}L_m W^{n_3}$ for some integers $n_3'$ and $n_3''$. Below it is shown that $C$ can be cut and pasted such that in the new computation path obtained $M$ does not alternate when the input head visits $W^{n_3}$. This yields a contradiction to the maximality of $m$.

Let us define the following head position bounds

$$L_1 := |Z_1 W^{n_1}|, \quad R_1 := L_1 + |W^{m}| + 1,$$
$$L_2 := |Z_1 W^{n_1+m+n_3'}|, \quad R_2 := L_2 + |W| |\sqrt{n}| + 1.$$

Not that from the assumption that $m' \geq 2 \lfloor \sqrt{n} \rfloor$ it follows that

$$(ii) \quad R_2 + |W| |\sqrt{n}| \leq |Z_1 W^{n+\ell n}|.$$

Let $C'$ be a subsequence of computations of $C$ which starts and ends with the head position in $\{L_1, R_2\}$. We claim that $C'$ can be modified to the computation path of the same number of alternations, which starts and ends in the same configurations as $C'$ and such that $M$ does not alternate with the head positions in $[L_1, R_1 + |W|]$. Only the case when $C'$ starts and ends with the head position $L_1$ and $R_2$, resp. will be described.

Let $(\alpha_1, L_1)$ be the first configuration of $C'$ and $(\beta_2, R_2)$ the last one. Moreover, let $(\beta_1, R_1)$ be the first configuration in $C'$ with the head position $R_1$ and let $(\alpha_2, L_2)$ be the last one with the head position $L_2$. Using a similar counting argument as in the proof of Lemma 2.4 one can show that

$$\exists c_1 \in [1 \ldots M_b] \ \forall d \in [1 \ldots M_b] \quad (\alpha_1, L_1) \models_{M,Y} (\beta_1, R_1 + c_1 d |W|).$$

Moreover, by Lemma 2.4 we have:

$$\exists c_2 \in [1 \ldots M_b] \ \forall d \in [1 \ldots M_b] \quad (\alpha_2, L_2 + c_2 d |W|) \models_{M,Y} (\beta_2, R_2).$$

Therefore, for $\delta := c_1 c_2 |W|$ holds:

$$(\alpha_1, L_1) \models_{M,Y} (\beta_1, R_1 + \delta),$$
$$(\alpha_2, L_2 + \delta) \models_{M,Y} (\beta_2, R_2).$$

By (ii), $M$ making the same moves as in $C'$ between $(\beta_1, R_1)$ and $(\alpha_2, L_2)$, reaches $(\alpha_2, L_2 + \delta)$ when started in $(\beta_1, R_1 + \delta)$. Hence, there is a computation path that starts in $(\alpha_1, L_1)$ ends in $(\beta_2, R_2)$ of the same number of alternations as $C'$ such that $M$ does not alternate with the head position in $[L_1, R_2 + |W|]$. This completes the proof of the claim and the lemma. \[ \square \]

2.3. Fooling ATMs by Pumping the Input.
Lemma 2.9 (1-Alternation). For any configuration \((\alpha, i)\) with
\[ i \leq |Z_1| \quad \text{or} \quad i > |Z_1 W^n| \quad \text{and} \]
\[ S_{\text{Space}}(\alpha, i, X) \leq S(n) \quad \text{and} \quad S_{\text{Space}}(\hat{\alpha}, \hat{i}, Y) \leq S(n) \text{ holds:} \]
\[ \text{acc}^2_M(\alpha, i, X) \implies \text{acc}^2_M(\hat{\alpha}, \hat{i}, Y) \quad \text{if} \quad (\alpha, i) \text{ is existential, and} \]
\[ \text{acc}^2_M(\hat{\alpha}, \hat{i}, Y) \implies \text{acc}^2_M(\alpha, i, X) \quad \text{for universal} \ (\alpha, i). \]

Proof. Assume that \((\alpha, i)\) fulfills both conditions above. First, let this configuration be existential and let \(\text{acc}^2_M(\alpha, i, X)\) be satisfied. Then there exists a universal configuration (or if \(M\) does not alternate a final accepting configuration) \((\beta_0, h)\) with \(0 \leq h \leq |X| + 1\), such that
\[ (A) \quad (\alpha, i) \models_{M,X} (\beta_0, h), \text{ and} \]
\[ (B) \quad \text{each computation path} \ C \text{ on input} \ X \text{ that starts in} \ (\beta_0, h) \text{ is finite.} \]
In addition, along each such \(C\) \(M\) does not alternate, and the final configuration of \(C\) is accepting.

We divide the string \(X\) according to \(h\) into three parts. Let \(n' := \lfloor n/2 \rfloor\). Define \(h_1 := |Z_1 W^n|\) if \(h \leq |Z_1 W^n|\), and \(h_1 := |Z_1|\), otherwise. Let \(h_2 := h_1 + |W^n| + 1\). Now let \(U\) denote the prefix of \(X\) of length \(h_1\), i.e. \(U := Z_1 W^n\) if \(h_1 = |Z_1 W^n|\) and \(U := Z_1\), otherwise. Moreover let \(V\) denote the suffix of \(X\) of length \(|X| + 1 - h_2\), i.e. if \(h_1 = |Z_1 W^n|\) then \(V := W^{n-2n'} Z_2\) else \(V := W^{n-n'} Z_2\) (note that \(V\) can be an empty word). Then, \(X = U W^n V\).

For such a partition of \(X\), the head of \(M\) in memory state \((\beta_0, h)\) is located on string $U$, if \(h \leq |Z_1 W^n|\) and on string $V$, otherwise. Let \(a := (n' + 1)(n' + 2)\ldots n\) and let \(\ell' := \ell a\). We will show that \(M\) started in \((\alpha, i)\) with \(X' := U W^{n+\ell' n!} V\) on its input tape accepts making at most one alternation. This proves the lemma since
\[ X' = U W^{n+\ell' n!} V = Z_1 W^{n+\ell' n!} Z_2 = Z_1 W^{n+\ell n!} Z_2 = Y. \]

Since \(N_{M,S} \leq n \leq (n' + 1)^2\) from Lemma 2.6 (for \(n := n'\) and \(m := n\)) and by \((A)\) it follows that
\[ (\alpha, i) \models_{M,X} (\beta_0, \hat{h}) \]
where \(\hat{h} := h\) if \(h \leq |Z_1 W^n|\) and \(\hat{h} := h + \ell' n!\) otherwise. Our lemma follows from this property and from the fact that
\[ \text{acc}^1_M(\beta_0, \hat{h}, X') \]
holds. Below we prove that this predicate is true.

Assume, to the contrary, that \(\text{acc}^1_M(\beta_0, \hat{h}, X')\) does not hold. We can distinguish two cases:
(a) \((\beta_0, \hat{h}) \models_{M,X} (\beta, t)\) for some rejecting or existential configuration \((\beta, t)\), or
(b) \(M\) starting in \((\beta_0, \hat{h})\) performs an infinite universal computation on \(X'\).

From Lemma 2.6, it follows that the memory state \(\beta\) is reachable on \(X\), too. We get a contradiction since by condition \((B)\) it must hold: if \(M\) reaches a non-universal memory state on \(X\) then it should be accepting. Therefore case \(a)\) cannot occur. Below we will prove that case \(b)\) cannot occur, too. More precisely, we will show that if \(b)\) holds then there exists an infinite universal computation path for input \(X\) which starts in \((\beta_0, \hat{h})\), also yielding a contradiction to \((B)\).
Let $C$ be an infinite universal computation path for input $X'$ that starts in $(\beta_0, \hat{h})$. From $C$ we will construct an infinite computation path for input $X$ that also starts in $(\beta_0, \hat{h})$. Let $\hat{h}_2$ denote the index of the first symbol of the string $V\$ on the input tape with input $X'$, i.e. let $\hat{h}_2 := h_2 + |W^{\ell'}n'|$. Three cases have to be distinguished.

**Case 1:** The boundary between the prefix $U$ and the string $W^{n'+\ell'n'}$ or the boundary between the string $W^{n'+\ell'n'}$ and the suffix $V$ is crossed infinitely often in $C$ (see the figure below).

![Figure 1](image1.png)

Let the boundary between the prefix $U$ and the string $W^{n'+\ell'n'}$ be crossed infinitely many times. Then there exists a memory state $\beta$ such that the configuration $(\beta, h_1)$ occurs in $C$ at least twice. From Lemma 2.6 one can conclude that

$$(\beta_0, \hat{h}) \models_{M,X} (\beta, h_1) \text{ and } (\beta, h_1) \models_{M,X} (\beta, h_1).$$

So, we obtain that $M$ starting in $(\beta_0, \hat{h})$ makes an infinite universal loop on $X$. The subcase when the boundary between the string $W^{n'+\ell'n'}$ and the suffix $V$ is crossed infinitely many times in $C$ is similar to this one.

**Case 2:** There is an initial part $C_1$ of $C$ and an infinite rest $C_2$ of $C$ such that in $C_2 M$ scans only the input to the left of $h_1$ or to the right of $\hat{h}_2$ (see the figure below).

![Figure 2](image2.png)

Let $(\beta, j)$, for $j = h_1$ or $j = \hat{h}_2$, be the last configuration of $C_1$. From the Lemma 2.6 we have that $(\beta, j)$ is reachable from $(\beta_0, \hat{h})$ on $X$, too. Let $C'_1$ denote a computation path from $(\beta_0, \hat{h})$ to $(\beta, j)$ for input $X$. Then $C'_1C_2$ is an infinite computation path for $X$.

**Case 3:** There is an initial part $C_1$ of $C$ and an infinite rest $C_2$ of $C$ such that in $C_2 M$ scans only the string $W^{n'+\ell'n'}$ (see the figure below).
Let \((\beta, j)\), for \(j = h_1\) or \(j = \hat{h}_2\), be the last configuration of \(C_1\). Without loss of generality, assume that \(j = h_1\). Since \(C_2\) is infinite there exists \(h_1 < d < \hat{h}_2\) and a memory state \(\gamma\) such that \((\gamma, d)\) occurs on \(C_2\) at least twice. By assumption, all memory states on computation path between the two instances of \((\gamma, d)\) use at most \(S(n)\) space. Lemma 2.3 implies that there exists a computation path \(D\) such that \(D\) starts and ends in \((\gamma, d)\), and that the input head is never moved farther than \(M_{S(n)}^2 \cdot |W|\) positions to the left nor to the right of \(d\). Let \(C_2^j\) denote the part of \(C_2\) between \((\beta, j)\) and the first \((\gamma, d)\) on \(C_2\). Using Lemma 2.3 and 2.4 one can easily construct from \(C_2^j\) a computation path \(D^1\) such that

- \(D^1\) starts in \((\beta, j)\),
- \(D^1\) ends in \((\gamma, d')\), for some \(d'\) such that
  \[d' < j + M_{S(n)}^2(M_{S(n)} + 1) \cdot |W|\] and \(d' \geq \min(d, j + M_{S(n)}^2 \cdot |W|)\)
- the input head is never moved to the left of \(j\) nor to the right of
  \[j + M_{S(n)}^2(M_{S(n)} + 2) \cdot |W| \leq j + n' \cdot |W|\].

Finally, let \(C_1^j\) denote a computation path for input \(X\) starting in \((\beta_0, \hat{h})\) and ending in \((\beta, j)\). By Lemma 2.6 such a path exists. \(M\) starting in \((\beta_0, \hat{h})\) and making the same sequence of moves as in \(C_1^j D^1 D DD...\) makes an infinite universal loop on \(X\).

This completes the proof of the first implication of the lemma. Let us now assume that \(\text{acc}^2_M(\alpha, i, Y)\) holds for a universal configuration \((\alpha, i)\). If \(\text{acc}^2_M(\alpha, i, X)\) is not true then there exists an existential configuration \((\beta_0, h)\) such that: \(M\) starting in \((\alpha, i)\) and working in universal states reaches \((\beta_0, h)\) and each computation \(C\) of \(M\) on \(X\) started in \((\beta_0, h)\) is rejecting or along \(C\) \(M\) makes at least one alternation. Using the similar methods as above one can show that \(\text{acc}^2_M(\alpha, i, Y)\) does not hold, too – contradiction.

\[\Box\]

2.4. Fooling ATMs by Shifting the Input Head. In the following two lemmata we consider the influence of shifting the input head between identical copies of a fixed string \(W\). For this purpose let us denote the shift distance by \(\Delta := |W| \cdot n!\).

**Lemma 2.10 (Configuration Shift).** Let \(X = Z_1 W^{n+nl} W^n W^n Z_2\) be a binary string with \(s \geq 1\) and let \(\alpha, \beta\) be memory states with \(|\alpha| \leq |\beta| \leq S(n)\). Then, for any integer \(i\) with \(i \leq |Z_1|\) or \(i > |Z_1| W^{n+nl} W^n W^n|\) and any integers \(j, \ell \in [|Z_1 W^{n+nl}| + 1 \ldots |Z_1 W^{n+nl} W^n|]\) holds:

1. \((\alpha, i) \models_{M,X} (\beta, j)\) if and only if \((\alpha, i) \models_{M,X} (\beta, j - \Delta)\).
2. \((\alpha, j) \models_{M,X} (\beta, \ell)\) if and only if \((\alpha, j - \Delta) \models_{M,X} (\beta, \ell - \Delta)\).
3. \((\alpha, j) \models_{M,X} (\beta, i)\) if and only if \((\alpha, j - \Delta) \models_{M,X} (\beta, i)\).
First note that the conditions on $j$, $\ell$, $j - \Delta$, $\ell - \Delta$ considered are at least $n$ blocks $W$ away from the boundaries $Z_1$ and $Z_2$. Define

\[ X' := Z_1 W^n W^s W^n Z_2 \quad \text{and} \quad X'' := Z_1 W^n W^s W^{n+n!} Z_2. \]

Set \( \hat{i} := i \) if \( i \leq |Z_1| \), otherwise \( \hat{i} := i - \Delta \). Using the Pumping Lemma twice – first for the input pair \( X, X' \) and then for \( X', X'' \) – we obtain:

1. \( (\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, \hat{i}) \models_{M,X'} (\beta, j - \Delta) \)

2. \( (\alpha, j) \models_{M,X} (\beta, \ell) \iff (\alpha, j - \Delta) \models_{M,X'} (\beta, \ell - \Delta) \)

3. \( (\alpha, j) \models_{M,X} (\beta, \hat{i}) \iff (\alpha, j - \Delta) \models_{M,X'} (\beta, \hat{i}) \)

The claim of the lemma follows because \( X'' = X \).

In the inductive argument for the proof of Theorem 1.2 (Proposition 3.8 in section 3 below) we have to guarantee a certain distance of the input head from the boundaries. For this purpose we define

\[ m_{k,n} := k \cdot (n + n!) . \]

**Lemma 2.11 (Position Shift).** Let \( k \geq 2, r, s, t \) be integers with \( r, t \geq m_{k,n} \) and \( s \geq 1 \), and let \( Z_1, Z_2, W \in \{0, 1\}^* \) be arbitrary strings. Then for input \( X = Z_1 W^r W^s W^t Z_2 \) and for any configuration \((\alpha, i)\) fulfilling the requirements

1. \( |Z_1 W^r| < i \leq |Z_1 W^r W^s| \) and
2. \( \text{Space}_{M}(\alpha, i, X) \leq S(n) \) and \( \text{Space}_{M}(\alpha, i - \Delta, X) \leq S(n) \)

holds:

\[ \text{acc}_{M}^{k-1}(\alpha, i, X) \iff \text{acc}_{M}^{k-1}(\alpha, i - \Delta, X). \]

**Proof.** Let input \( X \) and configuration \((\alpha, i)\) be as above. We will only give a proof for

\[ \text{acc}_{M}^{k-1}(\alpha, i, X) \implies \text{acc}_{M}^{k-1}(\alpha, i - \Delta, X). \]

A similar argument yields the opposite implication. Let

(i) \( \text{acc}_{M}^{k-1}(\alpha, i, X) \)

be true. First we will show the following property for computations that start in \((\alpha, i - \Delta)\). Call a computation path of finite or infinite length universal if all its configurations are universal.

**Claim 1.** For a universal configuration \((\alpha, i)\) of \( M \) on \( X \) any universal computation path that starts in \((\alpha, i - \Delta)\) is finite.

**Proof.** Let us assume, to the contrary, that there exists an infinite universal computation path that starts in \((\alpha, i - \Delta)\). Hence there exists a universal configuration \((\beta, j)\) such that

\( (\alpha, i - \Delta) \models_{M,X} (\beta, j) \) and \( (\beta, j) \models_{M,X} (\beta, j) \).
If \(|Z_1 W^n| < j \leq |Z_1 W^{r+s+t-(n+1)}|\) then Lemma 2.10 (2.) implies

\[(\alpha, i) \models_{M,X} (\beta, j + \Delta) \quad \text{and} \quad (\beta, j + \Delta) \models_{M,X} (\beta, j + \Delta) .\]

This means that in \((\alpha, i)\) \(M\) starts an infinite universal computation path with \(X\) on its input tape. This yields a contradiction to \(\text{acc}^{k-1}_M(\alpha, i, X)\). On the other hand, if \(j \leq |Z_1 W^n|\) or \(j > |Z_1 W^{r+s+t-(n+1)}|\) then by Lemma 2.10 (3.)

\[(\alpha, i) \models_{M,X} (\beta, j) .\]

Since \((\beta, j) \models_{M,X} (\beta, j)\) \(M\) also generates an infinite universal computation from \((\alpha, i)\). Note that we can apply the Configuration-Shift-Lemma both to \((\alpha, i)\) and \((\beta, j)\) because by the second assumption \(|\alpha| \leq |\beta| \leq S(n)\) holds. This ends the proof of Claim 1.

First we will solve the base case \(k = 2\) and consider an existential configuration \((\alpha, i)\). Because of \(\text{acc}^1_M(\alpha, i, X)\) there exists an accepting \((\beta, j)\) with

\[(\alpha, i) \models_{M,X} (\beta, j) .\]

Using the Configuration-Shift-Lemma one can conclude that

\[(\alpha, i - \Delta) \models_{M,X} (\beta, j - \Delta) \quad \text{if} \quad |Z_1 W^{n+1}| < j \leq |Z_1 W^{r+s+t-n}|, \quad \text{and} \quad (\alpha, i - \Delta) \models_{M,X} (\beta, j) \quad \text{otherwise.}\]

Since \(\beta\) is accepting \(\text{acc}^1_M(\alpha, i - \Delta, X)\) holds.

For universal configurations \((\alpha, i)\) it will be shown that any terminating configuration \((\beta, j)\) with \((\alpha, i - \Delta) \models_{M,X} (\beta, j)\) is accepting. Together with Claim 1 this proves that \(\text{acc}^1_M(\alpha, i - \Delta, X)\) holds. Let \((\beta, j)\) with \((\alpha, i - \Delta) \models_{M,X} (\beta, j)\) be a final configuration. By Lemma 2.10

\[(\alpha, i) \models_{M,X} (\beta, j + \Delta)\]

if \(|Z_1 W^n| < j \leq |Z_1 W^{r+s+t-(n+1)}|, \text{ otherwise}

\[(\alpha, i) \models_{M,X} (\beta, j) .\]

Hence, if \(\beta\) is non-accepting then \(\text{acc}^1_M(\alpha, i, X)\) does not hold – a contradiction.

Now let \(k > 2\) and consider an existential configuration \((\alpha, i)\). Since, by assumptions, \(M\) starting in \((\alpha, i)\) with \(X\) on the input tape accepts there exists an existential computation path ending in a universal configuration \((\beta, j)\), with

(ii) \[(\alpha, i) \models_{M,X} (\beta, j) ,\]

and

(iii) \(\text{acc}^{k-2}_M(\beta, j, X) .\)

(The trivial case that \(M\) accepts without alternations could be handled as above.)

Let us divide the input \(X = Z_1 W^r W^s W^t Z_2\) into three regions \(A, B, C\) as follows:

\[A := Z_1 W^{r-(n+1)} ,\]

\[B := W^n W^n W^n W^n ,\]

\[C := W^t \cdot Z_2 .\]

According to \(j\), the input head position in configuration \((\beta, j)\), the following situations will be distinguished:

**Case 1.** The input head is located in region \(A\) or \(C\) (see Fig. 4a), i.e. \(j \leq |A|\) or \(j > |AB|\).
Fig. 4

From property (ii) and Lemma 2.10 (3.) – for $Z_1 := A$, and $Z_2 := C$ – we obtain that

$$(\alpha, i - \Delta) \models_{M,X} (\beta, j)$$

(see Fig. 4b). Therefore condition (iii) implies $\text{acc}_{M}^{k-1}(\alpha, i - \Delta, X)$.

**Case 2.** The input head in $(\beta, j)$ visits region $B$ (see Fig. 5), i.e. $|A| < j \leq |AB|$.

Fig. 5
In this case using property (ii) and Lemma 2.10 (2.) – for $Z'_1 := Z_1 W^{r-2(n+n!)}$, $Z'_2 := W^{r-2n}Z_2$, and $s' := n! + n + s + n$ – we conclude that

$$(\alpha, i - \Delta) \models_{M,X} (\beta, j - \Delta).$$

Now apply the induction hypothesis for $k - 1$ with parameters $r' := r - (n + n!)$, $s'$ and $t' := t - n$ to configuration $(\beta, j)$. By definition of the parameters $m_{k,n}$ the requirements 1. and 2. are fulfilled. Therefore (iii) implies

$$\text{acc}^{k-2}_M(\beta, j - \Delta, X),$$

and hence $\text{acc}^{k-1}_M(\alpha, i - \Delta, X)$. This completes the proof for existential configurations.

For a universal $(\alpha, i)$, similar to the case $k = 2$, it will be shown that for any final or existential configuration $(\beta, j)$ that ends a universal computation path

$$(\alpha, i - \Delta) \models_{M,X} (\beta, j)$$

implies $\text{acc}^{k-2}_M(\beta, j, X)$.

Remember that because of Claim 1 only finite paths have to be considered. Let $(\beta, j)$ be such a configuration. Divide the input $X$ into three regions $A, B, C$ as above. Depending on which region is visited by the input head in configuration $(\beta, j)$, two cases are considered. If the input head is in region $A$ or $C$ (as in Fig. 6a) then from Lemma 2.10 (3.) we obtain that $(\alpha, i) \models_{M,X} (\beta, j)$. $\text{acc}^{k-1}_M(\alpha, i, X)$ thus implies $\text{acc}^{k-2}_M(\beta, j, X)$.

![Diagram](image)

Fig. 6

Otherwise the input head is located in $B$, i.e. $|A| < j \leq |AB|$ (see Fig. 6a). By Lemma 2.10 (2.), one can deduce that $(\alpha, i) \models_{M,X} (\beta, j + \Delta)$, which implies $\text{acc}^{k-2}_M(\beta, j + \Delta, X)$. Using the induction hypothesis for configuration $(\beta, j + \Delta)$ and for $k - 1$ with $r' := r - n$, $s' := n + s + n + n!$ and $t' := t - (n + n!)$ we obtain $\text{acc}^{k-2}_M(\beta, j, X)$, which completes the proof.  

\[ \square \]
2.5. Halting Computations for ATMs. Let $S$ and $Z$ be functions such that $Z$ is computable in space $S$ and $Z \leq \exp S$. We say that a binary string $X$ is $Z$–bounded if it contains at most $Z(|X|)$ zeros.

**Lemma 2.12.** For every $S$–space-bounded ATM $M$ there exists an ATM $M'$, which is also $S$–space-bounded, such that for all $Z$–bounded strings $X$ holds:

- $M'$ accepts $X$ iff $M$ accepts $X$,
- $\text{Alter}_{M'}(X) \leq \text{Alter}_M(X)$,
- if $\text{Alter}_M(X) < \infty$ then every computation path of $M'$ on $X$ is finite.

**Proof.** Let $M$ be an ATM and let $X$ be a $Z$–bounded input. In the proof below, $M_b$ denotes the number of memory states of $M$ as defined in Section 2.1.

Let a **crossing** be any transition of $M$ from a configuration, in which it reads an input symbol $a$ to a configuration reading an input symbol $b \neq a$, where $a, b \in \{0, 1\} \cup \{\$\}$. A sequence $C = C_u, C_{u+1}, \ldots, C_v$ of consecutive configurations of a computation path on $X$ is a **long turn** if $C$ does not contain alternations, nor crossings, if in $C_u$ and $C_v$ the input head is at the same position $i$ for some $1 \leq i \leq |X|$, and within $C$

- either the input head visits position $i + M_b^2$, but never moves to the left of $i$,
- or it visits position $i - M_b^2$, but never moves to the right of $i$,

where $b$ is the amount of space used in $C_v$.

On the other hand, a sequence $C$ without alternations or crossings is a **long hop** if the positions $i$ and $j$ of the input head in $C_u$, resp. $C_v$ are at least at a distance $M_b^2 + 1$ apart and within $C$ the input head never leaves the region between these two positions.

Now we are ready to describe the behaviour of the machine $M'$. It first computes the value $Z(|X|)$, which by assumption can be done in space $S(|X|)$, and then simulates $M$ step by step. Let $b_t$ be the amount of work space used by $M$ by its $t$–th step.

After having simulated step $t$ of $M$ the machine $M'$ stops and rejects iff

a1) $M$ rejects at this step, or
a2) $M$ has just finished a long turn that contains only existential configurations, or
a3) since its last alternation $M$ has executed $2(Z(|X|) + 1) \cdot M_{b_t} + 1$ many crossings, or
a4) within the last $2M_{b_t}^3 + 1$ steps $M$ has not made any progress, that means performed an alternation, a crossing, a long turn or a long hop.

$M'$ stops and accepts iff

b1) $M$ accepts, or
b2) $M$ has just finished a long turn that contains only universal configurations.

To check these conditions one counter for the number of crossings, one counter for the number of steps since the last progress and a sliding window for the most recent further part distance to the right or left, which can also be realized by counters, suffice.

The length of all counters is bounded by $O(S(|X|))$. Thus, $M'$ is $O(S)$–space bounded.

It is obvious that $\text{Alter}_{M'}(X) \leq \text{Alter}_M(X)$. To see that all computations of $M'$ are finite, first notice that if $M$ does not make progress infinitely often $M'$ will stop the simulation eventually. Assume that $M'$ does not stop on some path. If $\text{Alter}_M(X) < \infty$ this cannot be due to alternations nor to crossings of $M$ since there is also a finite bound set by $M'$. Thus it remains the case that $M$ within one block
of identical input symbols performs infinitely many steps without an alternation. $M'$ would stop if $M$ makes a long turn, thus $M$ has to make an unbounded number of long hops. After a long hop to one side it cannot make a long hop to the other side, because this would result in a long turn. Thus, $M$ eventually has to reach the boundary of this block and performs a crossing, a contradiction.

From Lemma 2.3 follows that $M'$ accepts the same set of $Z$-bounded strings as $M$. In case a2) there is a shorter turn that brings $M$ into a configuration identical to $C_v$. Thus, if $M$ has an accepting subtree for configuration $C_u$ then it still has after chopping of that $C_v$ which is reached by the long turn. The dual argument holds in case b2). Observe that in case a3) $M$ must have gone through a loop and one can stop the simulation. This is because there are at most $2(Z(|X|)+1)$ different positions on the input tape (counting both directions) to perform a crossing on a $Z$-bounded string $X$. Hence, at some position a memory state must repeat. A similar argument holds in case a4) for the at most $M'_2$ many input positions that can be visited without performing a long turn or hop.

Using this lemma we can show the following theorem that extends Sipser's space-bounded halting result to alternating TMs.

**Theorem 2.13.** Let $S, A, Z$ be bounds with $A < \infty$ and $Z \leq \exp S$ computable in space $S$. Then for every $S$-space-bounded $\Sigma_A$ TM $M$ there exists a $\Sigma_A$ TM $M'$ of space complexity $S$ such that for all inputs $X$

- $M'$ accepts $X$ iff $M$ accepts $X$ and $X$ is $Z$-bounded, and
- every computation path of $M'$ on $X$ is finite.

The identity of $\Sigma_k$ and $\text{co-}\Pi_k$ for $Z$-bounded languages (Theorem 1.8) now follows easily.

### 3. Hierarchies.

In this section the separation results will be proved.

#### 3.1. Technical Preliminaries.

As a specific example of a function that can be computed in sublogarithmic space consider the following function from [3]

$$ F(n) := \min\{k \in \mathbb{N} \mid k \text{ does not divide } n\} . $$

It is easy to see that $F \in O(\log)$. Thus, on input $1^n$ a TM can simply try all candidate $k = 2, 3, \ldots$ by counting the input length $\text{mod} k$ until the first nondivisor is found. Using the binary representation this requires at most $\log F(n) \leq \log n + O(1)$ space.

Obviously, $F$ takes constant values like 2 or 3 infinitely often. We want to show that also the logarithmic upper bound is achieved infinitely often. This would imply that there exists another function $G$ of logarithmic growth that can be approximated from below in space $\log n$. Let $p_1 < p_2 < \ldots$ be the standard enumeration of primes and define

$$ \Phi(k) := \prod_{p_i \leq k} p_i^{\lfloor \log_{p_i} k \rfloor}, $$

$$ \Phi^{-1}(n) := \min\{k \mid \Phi(k) \geq n\}, $$

$$ G(n) := \min\{\ell \mid \ell > \Phi^{-1}(n) \text{ and } \ell \text{ is a prime power}\} . $$

The following properties can easily be derived.

1. $\Phi^{-1}(\Phi(k)) = k$ and $\Phi(\Phi^{-1}(n)) \geq n$.
2. $F(\Phi(k)) = G(\Phi(k))$, since any $\ell \leq k$ divides $\Phi(k)$ and the first nondivisor in the sequence $k+1, k+2, \ldots$ must be a prime power.
3. $F(n) \leq G(n)$ for all $n$, which can be seen as follows: Let $k = \Phi^{-1}(n)$. Since we have already considered the case $n = \Phi(k)$ due to property 1. we may assume $n < \Phi(k)$. By definition of $\Phi$ there must exist a prime power $p^{|\log_{p_i} k|}$ that is not a divisor of $n$. Thus, $F(n) \leq k < G(n)$.

4. $\Phi(k) = e^{k(1+o(1))}$.

The prime number theorem implies

$$\prod_{p_i \leq k} p_i = e^{k(1+o(1))}.$$ 

Thus, $\Phi(k) \geq e^{k(1+o(1))}$. On the other hand,

$$\Phi(k) \leq \prod_{p_i \leq \sqrt{k}} p_i^{\log_{p_i} k} \cdot \prod_{\sqrt{k} < p_i \leq k} p_i \leq \prod_{p_i \leq \sqrt{k}} k \cdot \prod_{p_i \leq k} p_i \leq e^{(\sqrt{k} \ln k + k)(1+o(1))}.$$ 

Hence $\Phi(k) \leq e^{k(1+o(1))}$.

5. $\Phi^{-1}(n) = \ln n (1 + o(1))$.

6. $G(n) = \Phi^{-1}(n)(1+o(1)) = \ln n (1 + o(1))$, since any interval of the form $[\Phi^{-1}(n), \Phi^{-1}(n) \cdot (1+o(1))]$ is guaranteed to contain a prime.

Hence, the function $G$ is of logarithmic growth and approximated from below by $F$.

Let $\mathcal{F}$ be an infinite subset of the natural numbers with the following property:

$$(\bullet) \quad n \in \mathcal{F} \quad \implies \quad n + n! \notin \mathcal{F}.$$ 

Using the function $F$ we can give a simple example for such a set $\mathcal{F}$ (compare [7]):

$$\mathcal{F} := \{n > 2 | \forall \ell \in [3 \ldots n-1] \quad F(\ell) < F(n)\}.$$ 

The following property of $\mathcal{F}$ will be needed in the lower bound proofs.

**Lemma 3.1.**

1. Every interval of the form $[m, m^3]$ with $m \geq 3$ contains an element of $\mathcal{F}$.

2. For any integer $n > 2$ holds $n + n! \notin \mathcal{F}$.

**Proof.** Since the function $F$ is not bounded the set $\mathcal{F}$ is infinite. More specific, $\mathcal{F}$ contains all numbers of the form $\Phi(p_k)$ because $F(\Phi(p_k)) > p_k$ and for all $n < \Phi(p_k)$ by the same argument as in 3. above $F(\Phi(p_k)) \leq p_k$. The first claim can be shown by estimating the density of the sequence $(\Phi(p_k))_{k=1,2,\ldots}$. Since $p_{k+1} \leq 2p_k$ for all $k$ we get

$$\Phi(p_{k+1}) = \prod_{p_i \leq p_{k+1}} p_i^{\log_{p_i} p_{k+1}} \leq \prod_{p_i \leq p_{k+1}} p_i^{1+\log_{p_i} p_k} = \Phi(p_k) \cdot \prod_{p_i \leq p_{k+1}} p_i \leq \Phi(p_k)^3.$$ 

2.) follows easily from the equation $F(n) = F(n + n!)$. To see this equality note that any divisor of $n$ divides $n + n!$, too. Hence $F(n) \leq F(n + n!)$. On the other hand from the definition of $F$ we know that

$$F(n) \text{ does not divide } n$$

and, since $F(n) \leq n$, that

$$F(n) \text{ divides } n!.$$ 

Therefore $F(n)$ does not divide $n + n!$, which means that $F(n + n!) \leq F(n)$. 

\[\square\]
3.2. ATMs with a Constant Number of Alternations. With the help of sets \( F \) as defined above we construct a sequence of languages that separate the different levels of the alternation hierarchy for sublogarithmic space-bounded ATMs.

**Definition 3.2.** For an infinite subset \( F \) of the natural numbers let \( L_F \) be the language over the single letter alphabet \( \{1\} \) given by \( 1^n \in L_F \) iff \( n \in F \). Assume that \( F \) has property (\( \blacklozenge \)) and that \( L_F \in \Pi_2 \text{Space}(\log) \) and \( \overline{L}_F \in \Sigma_2 \text{Space}(\log) \). Then we define \( L_2 := \{1\}^+ \), and for \( k \geq 3 \) \( L_k := (L_{k-1} \{0\})^+ \). Furthermore,

\[
L_{\Pi 2} := L_F \quad \text{and} \quad L_{\Sigma 2} := \{1\}^+ \cap \overline{L}_F ,
\]

\[
L_{\Sigma k} := \{ w_1 0 w_2 0 \ldots 0 w_p 0 \mid p \in \mathbb{N}, w_i \in L_{k-1} \text{ and } \exists i \in [1 \ldots p] \ w_i \in L_{\Pi (k-1)} \} ,
\]

\[
L_{\Pi k} := \{ w_1 0 w_2 0 \ldots 0 w_p 0 \mid p \in \mathbb{N}, w_i \in L_{k-1} \text{ and } \forall i \in [1 \ldots p] \ w_i \in L_{\Sigma (k-1)} \} .
\]

Note that \( L_{\Sigma 2} \) and \( L_{\Pi 2} \) are just complementary. For larger \( k \) the corresponding languages are “almost” complementary, that means if restricting to strings with a syntactically correct division into subwords by the 0-blocks (more formally \( L_{\Pi k} = L_k \cap L_{\Sigma k} \)).

**Lemma 3.3.** For the specific \( F \) defined above with the help of the function \( F \) holds

\[
L_F \in \Pi_2 \text{Space}(\log) \quad \text{and} \quad \overline{L}_F \in \Sigma_2 \text{Space}(\log) .
\]

**Proof.** We describe \( \log \) space-bounded \( \Pi_2 \) TMs \( M_{\Pi} \) and \( \Sigma_2 \) TMs \( M_{\Sigma} \) that recognize the language \( L_F \), resp. the complement of \( L_F \). The machine \( M_{\Pi} \) verifies the condition \( \forall \ell \in [3 \ldots n - 1] \ F(\ell) < F(n) \) as follows:

- deterministically it computes \( F(n) \) and writes down the binary representation of \( F(n) \) on the tape;
- universally it guesses an integer \( \ell \in [3 \ldots n - 1] \):
  - it moves its input head to the right and stops \( \ell \) positions from the right end of the string \( 1^n \);
- existentially it guesses an integer \( k \in [1 \ldots F(n) - 1] \) and then moving the input head to the right, checks deterministically whether \( k \) divides \( \ell \).

\( M_{\Pi} \) accepts if \( k \) does not divide \( \ell \).

The complementary machine \( M_{\Sigma} \) writes down on the work tape \( F(n) \) in binary and tests whether

\[
\exists \ell \in [3 \ldots n - 1] \quad \forall k \in [1 \ldots F(n) - 1] \quad k \text{ divides } \ell .
\]

Similarly as in \( M_{\Pi} \) the input head position represents the integer \( \ell \). The integer \( k \) is stored in binary on the work tape. It is obvious that \( M_{\Pi} \) recognizes \( L_F \) and that \( M_{\Sigma} \) recognizes \( \overline{L}_F \) in space \( O(\log) \). \( \square \)

Thus languages \( L_F \) as assumed in Definition 3.2 exist. For the base case of the following inductive separation we also need the property that \( L_F \notin \Sigma_2 \text{Space}(o(\log)) \) and symmetrically that \( \overline{L}_F \notin \Pi_2 \text{Space}(o(\log)) \). This has been shown for the example above explicitly in [14]. Below we will give a general argument showing that this property simply follows from the condition \( n \in F \) and \( n + n \notin F \).

**Lemma 3.4.** For any \( k \geq 2 \) holds

\[
L_{\Sigma k} \in \Sigma_k \text{Space}(\log) ,
\]

\[
L_{\Pi k} \in \Pi_k \text{Space}(\log) .
\]
The proof of these properties is simple using the fact that \( L_\Sigma \in \Pi_2 \text{Space}(l\log) \) and \( L_\Sigma \in \Sigma_2 \text{Space}(l\log) \). The separation now follows from the following

**Theorem 3.5.** For any \( k \geq 2 \) holds

\[
L_{\Sigma k} \notin \Pi_k \text{Space}(o(\log)) , \\
L_{\Pi k} \notin \Sigma_k \text{Space}(o(\log)) .
\]

We will define specific inputs that belong to \( L_{\Sigma k} \) and \( L_{\Pi k} \) and show that any sublogarithmic space-bounded machine cannot work correctly on both inputs.

Let \( L = L_\Sigma \) be fixed. Recall that infinitely many \( n \in \mathbb{N} \) exist with \( n \in \mathcal{F} \), \( 1^n \in L \) and \( 1^{n+1!} \notin L \).

**Definition 3.6.** For \( n \in \mathcal{F} \) define words

\[
W_{\Sigma \Sigma 2}^n := 1^{n+n!} \quad \text{and} \quad W_{\Pi \Pi 2}^n := 1^n ,
\]

and for \( k \geq 3 \)

\[
W_{\Sigma \Sigma k}^n := \left[ W_{\Sigma \Sigma k-1}^n \ 0 \right]^{m_{k,n}} W_{\Pi \Pi k-1}^n 0 \left[ W_{\Sigma \Sigma k-1}^n \ 0 \right]^{m_{k,n}} , \\
W_{\Pi \Pi k}^n := \left[ W_{\Sigma \Sigma k-1}^n \ 0 \right]^{m_{k,n}} W_{\Pi \Pi k-1}^n 0 \left[ W_{\Sigma \Sigma k-1}^n \ 0 \right]^{m_{k,n}} ,
\]

where the \( m_{k,n} \) are the parameters already used in the Position-Shift-Lemma.

From the definition follows easily

**Lemma 3.7.** For \( k \geq 2 \) and every \( n \in \mathcal{F} \)

\[
W_{\Sigma \Sigma k}^n \in L_{\Sigma k} \quad \text{and} \quad W_{\Sigma k}^n \notin L_{\Pi k} , \\
W_{\Pi \Pi k}^n \in L_{\Pi k} \quad \text{and} \quad W_{\Pi k}^n \notin L_{\Sigma k} .
\]

Let \( k \geq 2 \) and \( S \in \text{SUBLOG} \) be a space bound. We will prove Theorem 3.5 by showing that if a \( \Sigma_k \) TM \( M \) accepts \( L_{\Pi k} \) in space \( S \) then for sufficiently large \( n \in \mathcal{F} \) \( M \) accepts \( W_{\Sigma k}^n \), too. Similarly, if a \( \Pi_k \) TM \( M \) accepts \( L_{\Sigma k} \) in space \( S \) then for large \( n \in \mathcal{F} \) it accepts \( W_{\Pi k}^n \) and hence makes a mistake. Recall that \( \mathcal{N}_{M,S} \) denotes the constant defined for \( M \) and \( S \) in Section 2.

**Proposition 3.8.** Let \( S \in o(\log) \) and \( M \) be an ATM. Then for any \( k \geq 2 \), for all \( n \geq \mathcal{N}_{M,S} \), for all strings \( U, V \in \{0,1\}^* \), and for any configuration \((\alpha, i)\) with

1. \( i \leq |U| \) or \( i > |U| W_{\Pi k}^n \) and
2. \( \text{Space}_M(\alpha, i, U W_{\Pi k}^n V) \leq S(n) \) and \( \text{Space}_M(\alpha, i, U W_{\Sigma k}^n V) \leq S(n) \)

holds:

\[
\text{acc}^k_M(\alpha, i, U W_{\Pi k}^n V) \implies \text{acc}^k_M(\alpha, \hat{i}, U W_{\Sigma k}^n V) \quad \text{if } (\alpha, i) \text{ is existential}, \\
\text{acc}^k_M(\alpha, i, U W_{\Pi k}^n V) \implies \text{acc}^k_M(\alpha, i, U W_{\Pi k}^n V) \quad \text{if } (\alpha, i) \text{ is universal}.
\]

**Proof.** Remember that \( \hat{i} \) was defined as

\[
\hat{i} = \begin{cases} 
  i & \text{if } i \leq |U|, \\
  i + (|W_{\Sigma k}^n| - |W_{\Pi k}^n|) & \text{if } i > |U| W_{\Pi k}^n .
\end{cases}
\]

For \( k = 2 \) the implications above follow from the 1-Alternation Lemma.
To establish the proposition for $k > 2$ we consider the first time when the machine $M$ makes an alternation and inductively use the corresponding properties for the strings $W_{\Sigma k-1}^n$ and $W_{\Pi k-1}^n$. The argument concentrates only on the block in the middle of a $W_{\Sigma k}^n$ string, which is a $W_{\Pi k-1}^n$ word, and analogously for $W_{\Pi k}^n$ strings with a $W_{\Sigma k-1}^n$ word in the middle. The main technical difficulty for the following argument is the possibility that in an accepting computation the machine may just make its first alternation in the middle block, and therefore may notice the difference between the $W_{\Sigma k}^n$ and $W_{\Pi k}^n$ strings. But the Configuration- and Position-Shift-Lemmas imply that there also exist accepting computations with the first alternation outside this critical region.

The details are as follows. Assume that the configuration $(\alpha, i)$ fulfills properties 1. and 2. Let $n \geq N_{M,S}$, and define

- $X := U W_{\Pi k}^n V = U' W_{\Sigma k-1}^n V'$,
- $Y := U W_{\Sigma k}^n V = U' W_{\Pi k-1}^n V'$, where $U' := U \left[ W_{\Sigma k-1}^n 0 \right]^{m_{k,n}}$ and $V' := 0 \left[ W_{\Pi k-1}^n 0 \right]^{m_{k,n}} V$,
- $\Delta := |W_{\Pi k-1}^n 0| \cdot n!$,
- $\tilde{j} := \begin{cases} j & \text{if } j \leq |U'|, \\ j + (|W_{\Pi k-1}^n| - |W_{\Sigma k-1}^n|) & \text{if } j > |X| - |V'|. \end{cases}$

Note that $\tilde{i}$ is defined with respect to the partition of the inputs $X, Y$ with the prefix $U$ and the suffix $V$, where $\tilde{j}$ is taken with respect to the prefix $U'$ and suffix $V'$. Since $|W_{\Pi k-1}^n| - |W_{\Sigma k-1}^n| = |W_{\Pi k}^n| - |W_{\Pi k}^n|$

(i) $\hat{i} = \tilde{i}$ whenever both values are defined.

First we prove the following

**CLAIM 1.** For any memory state $|\alpha_1| \leq |\alpha_2| \leq S(n)$ and all $j_1, j_2$ with $j_1, j_2 \in [0 \ldots |U'|] \cup [|U'| W_{\Sigma k-1}^n] + 1 \ldots |X| + 1]$ holds:

$(\alpha_1, j_1) \models_{M,X} (\alpha_2, j_2) \iff (\alpha_1, \hat{j}_1) \models_{M,Y} (\alpha_2, \tilde{j}_2).$

**Proof.** For suitable $Z_1, Z_2 \in \{0, 1\}^*$ the words considered can be written as $W_{\Sigma k}^n = Z_1^{1^n} Z_2$ and $W_{\Pi k}^n = Z_1^{1^n+n!} Z_2$ if $k$ is odd, and $W_{\Sigma k}^n = Z_1^{1^{n+n!}} Z_2$ and $W_{\Pi k}^n = Z_1^{1^n} Z_2$ for even $k$.

The claim then follows from the Pumping Lemma (Lemma 2.5). \( \square \)

**A.** First we consider existential configurations $(\alpha, i)$. Assume that $\text{acc}_{M}^{\Sigma}(\alpha, i, X)$ is true. Hence there exists an existential computation path from $(\alpha, i)$ to a final or universal configuration $(\beta, j)$:

(ii) $(\alpha, i) \models_{M,X} (\beta, j)$
with the property

\( \text{acc}^{k-1}_M(\beta, j, X) \).

We may assume that

\( j \leq |U'| \quad \text{or} \quad j > |U' W^\varnothing_{\Sigma^{k-1}}| \),

because if \(|U'| < j \leq |U' W^\varnothing_{\Sigma^{k-1}}|\) then for \( Z_1 := U, Z_2 := V, W := W^\varnothing_{\Sigma^{k-1}} 0, \) and \( s := 2m_{k,n} + 1 - n - (n + n!) \) the Configuration-Shift-Lemma implies

\( (\alpha, i) \models_{M, X} (\beta, j - \Delta) \).

Moreover, for \( r := t := m_{k,n} \) and for \( s := 1 \), from the Position-Shift-Lemma we can deduce

\( \text{acc}^{k-1}_M(\beta, j - \Delta, X) \).

Therefore, if \(|U'| < j \leq |U' W^\varnothing_{\Sigma^{k-1}}|\) the configuration \((\beta, j')\) with \( j' := j - \Delta \) instead of \((\beta, j)\) satisfies properties (ii)-(iv).

Since \( \hat{i} = \tilde{i} \) according to (i), Claim 1 applied to (ii) yields

\( (\alpha, \hat{i}) = (\alpha, \tilde{i}) \models_{M, Y} (\beta, \tilde{j}) \).

A terminating configuration \((\beta, j)\) must be accepting because of (ii) and (iii), hence \((\beta, \tilde{j})\) is accepting and \( \text{acc}^{k}_M(\alpha, \hat{i}, Y) \) is true.

For a universal \((\beta, j)\) we apply the induction hypothesis. Because of (iv) the requirements 1. and 2. of the proposition are fulfilled for \( k - 1 \) and \( i := \tilde{j} \). Property (i) implies for this choice of \( i \) that \( \hat{i} = j \). Therefore, in (iii) replacing \( j \) by \( \hat{i} \) one can conclude

\[ \text{acc}^{k-1}_M(\beta, \hat{i}, X) \implies \text{acc}^{k-1}_M(\beta, i, X) = \text{acc}^{k-1}_M(\beta, \tilde{j}, Y). \]

Hence, we can conclude that \( \text{acc}^{k}_M(\alpha, \hat{i}, Y) \) holds. This proves the proposition for existential configurations.

B.) Now let us consider universal configurations \((\alpha, i)\), for which \( \text{acc}^{k}_M(\alpha, \hat{i}, Y) \) holds. We have to show that \( \text{acc}^{k}_M(\alpha, i, X) \) is true.

Claim 2. For input \( X \) any universal computation path starting in \((\alpha, i)\) is finite.

Proof. Assume, to the contrary, that there exists an infinite computation path which is universal and starts in \((\alpha, i)\). This means that there exists a universal configuration \((\beta, j)\) such that

\( (\alpha, i) \models_{M, X} (\beta, j) \models_{M, X} (\beta, j) \).

We can assume that

\( j \leq |U'| \quad \text{or} \quad j > |U' W^\varnothing_{\Sigma^{k-1}}| \)

because if \(|U'| < j \leq |U' W^\varnothing_{\Sigma^{k-1}}|\) the Configuration-Shift-Lemma implies

\( (\alpha, i) \models_{M, X} (\beta, j - \Delta) \models_{M, X} (\beta, j - \Delta) \).

Hence, (v) and (vi) are fulfilled for \( j' := j - \Delta \). Form (i) and Claim 1 follows

\( (\alpha, \hat{i}) = (\alpha, \tilde{i}) \models_{M, Y} (\beta, \tilde{j}) \models_{M, Y} (\beta, \tilde{j}) \).
This means that for input $Y$ there exists an infinite computation path, which is universal and starts in $(\alpha, \hat{i})$. We get a contradiction to $\text{acc}^k_M(\alpha, i, Y)$. 

Now we want to show that for any final or existential configuration $(\beta, j)$ that can be reached from $(\alpha, i)$ on a universal computation path holds

$$\text{acc}^{k-1}_M(\beta, j, X).$$

According to Claim 2 this proves $\text{acc}^k_M(\alpha, i, X)$. Let $(\alpha, i) \models_{M, X} (\beta, j)$. Two cases will be distinguished.

**Case 1.** $j \leq |U'|$ or $j > |U' W_{n, \Sigma, k-1}|$.

From Claim 1 it follows that

$$(\alpha, \hat{i}) = (\alpha, \tilde{i}) \models_{M, Y} (\beta, \tilde{j}).$$

The assumption $\text{acc}^k_M(\alpha, \hat{i}, Y)$ implies

$$(\alpha, \hat{i}) \models_{M, Y} (\beta, \tilde{j}) \quad (\text{vii})$$

For a final configuration $(\beta, j)$ one can conclude from property (vii) that $\beta$ must be accepting, hence $\text{acc}^{k-1}_M(\beta, j, X)$ holds.

For an existential $(\beta, j)$ the same implication holds using the induction hypothesis.

**Case 2.** $|U'| < j \leq |U' W_{n, \Sigma, k-1}|$.

The Configuration-Shift-Lemma implies

$$(\alpha, i) \models_{M, X} (\beta, j - \Delta).$$

In the proof of Case 1 it was shown for the configuration $(\beta, j - \Delta)$ that

$$\text{acc}^{k-1}_M(\beta, j - \Delta, X)$$

holds. Using the Position-Shift-Lemma we obtain $\text{acc}^{k-1}_M(\beta, j, X)$. This completes the proof of Proposition 3.8.

Next, we will show that the second requirement of the proposition above is always fulfilled.

**Proposition 3.9.** Let $k \geq 2$ and $M$ be an ATM of space complexity $S$ with $S \in o(\log n)$. Then there exists a bound $S' \in o(\log n)$ such that for all $n \geq N_{M, S'}$

$$\text{Space}_M(W^n_{\Pi, k}) \leq S'(n) \quad \text{and} \quad \text{Space}_M(W^n_{\Sigma, k}) \leq S'(n).$$

**Proof.** The idea of the proof is as follows. If in $W^n_{\Pi, k}$ and $W^n_{\Sigma, k}$ all substrings generated in the recursive construction which are multiples of $n!$, are cancelled, then the remaining word has a length $p_k(n)$, which is polynomial in $n$. Using the Small-Space-Bound-Lemma, which shows that a sublogarithmic space-bounded machine $M$ does not notice a difference when an arbitrary block of the input is added $n!$ times, it follows that $M$ must obey a space bound $S(p_k(n))$ on $W^n_{\Pi, k}$ and $W^n_{\Sigma, k}$. If $S$ grows sublogarithmically in $n$ so does $S(p_k(n))$.

Below the technical details of this proof are outlined. Let

$$V^1_2(n) := 1^n.$$

For $d \geq 3$ define

$$V^1_d(n) := \left[V^1_{d-1}(n) \cdot 0\right]^{2dn+1},$$
and for $i = 2, \ldots, d - 1$

$$V^i_d(n) := \left[V^{i-1}_{d-1}(n) 0 \right]^{2m_{d,n}+1}.$$ 

Define also a sequence of polynomials $p_d(n)$ as follows:

$$p_2(n) := n \quad \text{and for } d \geq 3 \quad p_d(n) := (2dn + 1) \cdot (p_{d-1}(n) + 1).$$

Obviously, for any $d \geq 2$ and for all $n$

$$p_d(n) = |V^1_d(n)|.$$ 

Let $M$ be an ATM of space complexity $S$ with $S \in o(\log)$. Define $S'(n) := S(p_k(n))$.

Obviously, $S' \in o(\log)$. Let $n$ be an integer with $n \geq N_{M,S'}$.

Since $M$ is $S$ space-bounded

(i) \hspace{1em} $\text{Space}_M(V^1_k(n)) \leq S(p_k(n)) = S'(n).$

It is easy to check that for any $n$ and for any $i \in [1..k-2]$ there are words $Z_1, Z_2, \ldots, Z_r$ over the alphabet $\{0\}$, where

$$r := \prod_{i=k-i+2}^{k} 2m_{t,n} + 1,$$

(for $i = 1$ take $r := 1$), such that for $W := V^1_{k-i}(n) 0$, $a := 2n(k - i) + n + 1$ and $b := 2(k - i + 1)$:

$$V^i_k(n) = W^{a+n} Z_1 W^{a+n} Z_2 \ldots Z_{r-1} W^{a+n} Z_r,$$

$$V^{i+1}_k(n) = W^{a+n+bn1} Z_1 W^{a+n+bn1} Z_2 \ldots Z_{r-1} W^{a+n+bn1} Z_r.$$ 

By the Small-Space-Bound-Lemma the following implications hold for $i = 1, \ldots, k - 2$

$$\text{Space}_M(V^i_k(n)) \leq S'(n) \quad \implies \quad \text{Space}_M(V^{i+1}_k(n)) \leq S'(n).$$

Therefore, by (i), we obtain that

(ii) \hspace{1em} $\text{Space}_M(V^{k-1}_k(n)) \leq S'(n).$

Now let $\hat{W}^n_{\Sigma k}$ denote a word $W^r_{\Sigma k}$ where all substrings $1^{n+1}$ are reduced to $1^n$. Similarly, $W^n_{\Pi k}$ is obtained from $W^n_{\Pi k}$. Obviously, by the Small-Space-Bound-Lemma, $\text{Space}_M(\hat{W}^n_{\Sigma k}) \leq S'(n)$ implies $\text{Space}_M(W^n_{\Sigma k}) \leq S'(n)$ and $\text{Space}_M(\hat{W}^n_{\Pi k}) \leq S'(n)$ implies $\text{Space}_M(W^n_{\Pi k}) \leq S'(n)$. The proposition holds since

$$\hat{W}^n_{\Sigma k} = \hat{W}^n_{\Pi k} = V^{k-1}_k(n)$$

and by (ii) the space used by $M$ on input $V^{k-1}_k(n)$ is bounded by $S'(n)$.

Now we are ready to prove Theorem 3.5. Let us assume that $M$ is a $\Sigma_k$TM accepting $L_{\Pi k}$ in sublogarithmic space $S$. By Proposition 3.9 there exists a function $S' \in o(\log)$ such that for any $n \geq N_{M,S'}$

$$\text{Space}_M(W^n_{\Pi k}) \leq S'(n) \quad \text{and} \quad \text{Space}_M(W^n_{\Sigma k}) \leq S'(n).$$

Let $n$ with $n \in F$ be an integer larger than $N_{M,S'}$ (such an $n$ exists since $F$ is infinite). By Lemma 3.7 $W^n_{\Pi k} \in L_{\Pi k}$, hence $M$ has to accept $W^n_{\Pi k}$, which means that $\text{acc}^k_M(\alpha_0, 0, W^n_{\Pi k})$ is true, where $(\alpha_0, 0)$ is the initial configuration of $M$. From Proposition 3.8 we conclude that $\text{acc}^k_{\Sigma k}(\alpha_0, 0, W^n_{\Sigma k})$ holds, too, and hence $M$ accepts $W^n_{\Sigma k}$, which by Lemma 3.7 does not belong to $L_{\Pi k}$ a contradiction.

In the same way one shows that if $M$ is a $\Pi_k$TM that accepts $L_{\Sigma k}$ in space $S$, then $M$ accepts $W^n_{\Pi k}$. 

\[ \square \]
3.3. Unbounded Number of Alternations. Let us now consider ATMs with a nonconstant bounding function \( A \) for the number of alternations. The separating results for \( A \)-alternation-bounded space classes (Theorem 1.5) follow from the propositions below.

Definition 3.10. Let \( A : \mathbb{N} \to \mathbb{N} \) be a function with \( A(n) \geq 2 \) for all \( n \) and define
\[
L_\Sigma(A) := \{ X \mid X = W0^r \text{ for some } r \in \mathbb{N} \text{ and } W \in L_{\Sigma k} \text{ for some } k \leq A(|X|) \}
\]
\[
L_\Pi(A) := \{ X \mid X = W0^r \text{ for some } r \in \mathbb{N} \text{ and } W \in L_{\Pi k} \text{ for some } k \leq A(|X|) \}.
\]

Lemma 3.11. For any \( S \in \text{SUBLOG} \) and all functions \( A \geq 2 \) computable in space \( S \) holds:
\[
L_\Sigma(A) \in \Sigma_A \text{Space}(S),
\]
\[
L_\Pi(A) \in \Pi_A \text{Space}(S).
\]

Proof. On input \( X = W0^r \) the machine first computes \( a := A(|X|) \) and initializes a counter with that value. It remains to check whether \( W \in L_{\Sigma k} \) for some \( k \leq a \). This can be done similarly as in the case for fixed \( k \), decrementing the counter each time an alternation has been performed.

For functions \( A, B : \mathbb{N} \to \mathbb{N} \) let \( A \leq^* B \) denote that \( A(m) \leq B(m) \) for all \( m \in \mathbb{N} \) with equality for infinitely many \( m \).

Proposition 3.12. For any \( S \in \text{SUBLOG} \) and for all functions \( A \) and \( B \) with \( 1 < A \leq^* B \) and \( B \cdot S \in o(\log) \) holds:
\[
L_\Sigma(A) \notin \Pi_B \text{Space}(S),
\]
\[
L_\Pi(A) \notin \Sigma_B \text{Space}(S).
\]

Proof. Let \( S \in \text{SUBLOG} \) and let \( A, B \) be functions with \( 1 < A \leq^* B \) and \( B \cdot S \in o(\log) \). These assumptions imply that there exists a constant \( m_0 \geq \exp \exp 9 \) such that \( A(m) < \frac{\log m}{\log \log m} \) for all \( m \geq m_0 \). Define functions \( h \) and \( f \) as follows
\[
h(m) := \frac{\exp \left( \frac{\log m}{A(m)} \right)}{3 A(m)},
\]
\[
f(m) := \max \{ \ell \mid \ell \in \mathcal{F} \cup \{0\}, \ell \leq h(m) \}.
\]

For \( m \geq m_0 \) we can bound \( h \) by
\[
h(m) \leq \exp \left( \frac{\log m}{2} \right) = m^{1/2},
\]
\[
h(m) \geq \frac{\exp \log m}{3 \log m / \log \log m} = \frac{\log m}{3} \geq 3.
\]

and hence \( f(m) \in \mathcal{F} \). Moreover, from lemma 3.1 follows
\[
(i) \quad f(m) \geq h(m)^{1/3} \geq \left( \frac{1}{3} \log m \right)^{1/3}.
\]
Define the function $S' : \mathbb{N} \to \mathbb{N}$ as follows

$$S'(n) := \max \left( \{0\} \cup \{S(m) \mid f(m) = n\} \right).$$

Because $f$ grows unboundedly $S'(n)$ will always be a finite number.

**Lemma 3.13.** $S' \in o(\log)$.

**Proof.** First we show that $S \in o(\log f)$. By assumption,

$$S \in o \left( \frac{\log A}{A} \right) \quad \text{and} \quad \log A \leq \log m \leq S.$$

This implies

$$S \in o \left( \frac{\log A}{A} - S \right) = o \left( \frac{\log A}{A} - \log A \right) = o(\log h) = o(\log f).$$

Thus, if $n$ goes to $\infty$

$$\frac{S(n)}{\log f(n)} \to 0$$

and

$$\frac{S'(n)}{\log n} = \max_{m \mid f(m) = n} \frac{S(m)}{\log n} = \max_{m \mid f(m) = n} \frac{S(m)}{\log f(m)}.$$

If $n$ goes to $\infty$ also $m$ has to do this, and hence all quotients converge to 0. But this means that $S' \in o(\log)$.

Consider the function $t$ defined by

$$t(m) := m - p_A(f(m))$$

where $p_A(n)$ has already been defined in the proof of Proposition 3.9, and note that

$$p_A(f(m)) \leq (3 A(m) f(m))^A(m) \leq m.$$

Thus, $t(m) \geq 0$.

Now let $M$ be an ATM that works in space $S(|X|)$ and makes at most $B(|X|) - 1$ alternations. Let $m$ be an integer with

(ii) $m \geq \max\{m_0, \exp\exp 3(N_{M,S'})^4\}$ and $A(m) = B(m)$

Such an $m$ exists since $A \leq B$. Then define

$$k := A(m) \quad \text{and} \quad n := f(m).$$

By (i) and (ii) $n \geq N_{M,S'}$. Moreover, $n \in \mathcal{F}$ and $M$ makes no more then $k - 1$ alternations on any input of length $m$. Let

$$X := V_k^1(n) 0^{t(m)}$$

with the word $V_k^1(n)$ defined as in the proof of Proposition 3.9. Since the length of $V_k^1(n)$ is $p(k,n)$ the string $X$ is of length $m$. From the definition of $S'$ follows that

$$\text{Space}_M(X) \leq S(m) \leq \max\{S(m') \mid f(m') = n\} = S'(n)\text{and}$$

$$\text{Alter}_M(X) \leq B(m) - 1 \leq \exp S(m) \leq \exp S'(n).$$
Hence, for the machine \( M \) and the function \( S' \) the assumptions of the Small-Space-Bound-Lemma and the Small-Alternation-Bound-Lemma are fulfilled. Using the Small-Space-Bound-Lemma for the input \( X \) in the similar way as in the proof of Proposition 3.9 one can show that

\[
Space_M(W^{n}_{\Sigma_k} 0^{t(m)}) ,\ Space_M(W^{n}_{\Pi_k} 0^{t(m)}) = Space_M(X) \leq S'(n) .
\]

Similarly, by the Small-Alternation-Bound-Lemma one obtains that

\[
Alter_M(W^{n}_{\Sigma_k} 0^{t(m)}) ,\ Alter_M(W^{n}_{\Pi_k} 0^{t(m)}) = Alter_M(X) \leq B(m) - 1 = k - 1 .
\]

Now we can finish the proof. Let us assume that \( M \) is a \( \Sigma_{B} \) TM accepting \( L_{\Pi}(A) \) in space \( S \). By Lemma 3.7 holds \( W^{n}_{\Pi_k} \in L_{\Pi_k} \), hence \( M \) has to accept \( W^{n}_{\Pi_k} 0^{t(m)} \). But this means that \( acc^k_M(\alpha_0, 0, W^{n}_{\Pi_k} 0^{t(m)}) \) is true, where \( (\alpha_0, 0) \) is the initial configuration of \( M \). From Proposition 3.8 we conclude that \( acc^k_M(\alpha_0, 0, W^{n}_{\Sigma_k} 0^{t(m)}) \) holds, too. Therefore \( M \) accepts \( W^{n}_{\Sigma_k} 0^{t(m)} \), which by Lemma 3.7 does not belong to \( L_{\Pi}(A) \) – a contradiction.

In the same way, one can show that if \( M \) is a \( \Pi_{B} \) TM that accepts \( L_{\Sigma}(A) \) in space \( S \) then \( M \) accepts \( W^{n}_{\Pi_k} 0^{t(m)} \).

\[
4. \textbf{Closure Properties.} \text{ In this section we discuss closure properties of classes } \Sigma_k \text{Space}(S) \text{ and } \Pi_k \text{Space}(S) \text{ for sublogarithmic bounds } S. \text{ First for any integer } k \geq 2 \text{ we define the languages}
\]

\[
A_{\Sigma_k} := L_k \{0\} L_{\Sigma_k} , \quad B_{\Sigma_k} := L_{\Sigma_k} \{0\} L_k ,
\]

and symmetrically

\[
A_{\Pi_k} := L_k \{0\} L_{\Pi_k} , \quad B_{\Pi_k} := L_{\Pi_k} \{0\} L_k .
\]

It is easy to see that

(i) \( A_{\Sigma_k}, B_{\Sigma_k} \in \Sigma_k \text{Space}(\log) \) and \( A_{\Pi_k}, B_{\Pi_k} \in \Pi_k \text{Space}(\log) \).

\[
\text{Proposition 4.1. For all } k \geq 2 \text{ holds:}
\]

\[
A_{\Sigma_k} \cap B_{\Sigma_k} \in \Pi_{k+1} \text{Space}(\log) \setminus \Sigma_{k+1} \text{Space}(o(\log)) ,
\]

\[
A_{\Pi_k} \cup B_{\Pi_k} \in \Sigma_{k+1} \text{Space}(\log) \setminus \Pi_{k+1} \text{Space}(o(\log)) .
\]

\[
\text{Proof. It is well known that for any function } S \text{ the classes } \Sigma_k \text{Space}(S) \text{ are closed under union, and symmetrically the } \Pi_k \text{Space}(S) \text{ are closed under intersection (see e.g. [25]). Hence by (i), } A_{\Sigma_k} \cap B_{\Sigma_k} \in \Pi_{k+1} \text{Space}(\log) \text{ and } A_{\Pi_k} \cup B_{\Pi_k} \in \Sigma_{k+1} \text{Space}(\log). \text{ To prove that } A_{\Sigma_k} \cap B_{\Sigma_k} \notin \Sigma_{k+1} \text{Space}(\log) \text{ and } A_{\Pi_k} \cup B_{\Pi_k} \notin \Pi_{k+1} \text{Space}(\log) \text{ first we modify Proposition 3.9 in the following way:}
\]

\[
\text{Proposition 4.2. Let } k \geq 2 \text{ and } M \text{ be an ATM of space complexity } S \text{ with } S \in o(\log). \text{ Then there exists a bound } S'' \in o(\log) \text{ such that for all } n \geq N_M,S'' \text{ and words } W_1, W_2 \in \{W^n_{\Sigma_k} , W^n_{\Pi_k}\}
\]

\[
Space_M(W_1 \ 0 \ W_2) \leq S''(n) .
\]

\[
\text{Proof. Let } S''(n) := S(2p_k(n) + 1), \text{ where } p_k \text{ is the polynomial specified in the proof of Proposition 4.2. It is easy to check that the proof of Proposition 4.2 generalizes to this situation. } \]
Let us assume, to the contrary, that \( A_{\Sigma k} \cap B_{\Sigma k} \in \Sigma_{k+1}\text{space}(S) \), for some \( S \in \omega(\log n) \). Let \( M \) be an \( S \) space-bounded \( \Sigma_{k+1} \text{TM} \) for \( A_{\Sigma k} \cap B_{\Sigma k} \). Choose \( n \in \mathcal{F} \) sufficiently large. By Lemma 3.7 \( W^{n}_{\Sigma k} \in L_{\Sigma k} \) hence \( M \) has to accept

\[
X = W^{n}_{\Sigma k} 0 W^{n}_{\Sigma k}
\]

which means that there exists an existential computation path starting in initial configuration \((\alpha_0,0)\) and ending in a universal configuration \((\beta,j)\), with

\[
(ii) \quad (\alpha_0,0) \models_{M,X} (\beta,j),
\]

and

\[
(iii) \quad \text{acc}^{k}_{M}(\beta,j,X).
\]

(The trivial case that \( M \) accepts \( X \) without alternation could be handled similarly.)

Now let \( Y_1 := W^{n}_{\Sigma k} 0 W^{n}_{\Sigma k} \) and \( Y_2 := W^{n}_{\Sigma k} 0 W^{n}_{\Sigma k} \). By Proposition 4.2 there exists \( n'' \in \omega(\log n) \) such that

\[
\text{Space}_M(X), \text{Space}_M(Y_1), \text{Space}_M(Y_2) \leq n''(n).
\]

Therefore, applying Claim 1 (from the Proof of Proposition 3.8) and Proposition 3.8 to (ii) and (iii), resp., we obtain

\[
(\alpha_0,0) \models_{M,Y_1} (\beta,j) \quad \text{and} \quad \text{acc}^{k}_{M}(\beta,j,Y_1)
\]

if \( j \leq |W^{n}_{\Sigma k} 0| \) and otherwise

\[
(\alpha_0,0) \models_{M,Y_2} (\beta,j) \quad \text{and} \quad \text{acc}^{k}_{M}(\beta,j,Y_2),
\]

where \( \hat{j} = j + |Y_2| - |X| \). Hence \( M \) also accepts input \( Y_1 \) or \( Y_2 \). This yields a contradiction since, by Lemma 3.7, \( Y_1, Y_2 \notin A_{\Sigma k} \cap B_{\Sigma k} \).

Similarly, one can show that if a \( \Pi_{k+1} \text{TM} \) accepts \( A_{\Sigma k} \cup B_{\Sigma k} \) within space \( S \in \omega(\log n) \), then it has to reject \( X \), but it also rejects input \( Y_1 \) or \( Y_2 \), which both belong to \( A_{\Sigma k} \cup B_{\Sigma k} \) – a contradiction!

This result can be applied to prove Theorem 1.7:

For all \( k \geq 2 \) and any \( S \in \text{SUBLOG} \) holds:

1. \( \Sigma_k \text{space}(S) \) and \( \Pi_k \text{space}(S) \) are not closed under complementation.
2. \( \Sigma_k \text{space}(S) \) is not closed under intersection,
3. \( \Pi_k \text{space}(S) \) is not closed under union.
4. \( \Sigma_k \text{space}(S) \) and \( \Pi_k \text{space}(S) \) are not closed under concatenation.

(1) follows immediately from Lemma 3.4, Theorem 3.5 and the following equations: \( L_{\Sigma k} = L_k \cap \overline{T}_{\Pi k} \) and \( L_{\Pi k} = L_k \cap \overline{T}_{\Sigma k} \), where \( L_k \) is the regular language introduced in Definition 3.2.

By (i) \( A_{\Sigma k}, B_{\Sigma k} \in \Sigma_k \text{space}(\log) \) and \( A_{\Pi k}, B_{\Pi k} \in \Pi_k \text{space}(\log) \). On the other hand, from Proposition 4.1 \( A_{\Sigma k} \cap B_{\Sigma k} \notin \Sigma_{k+1} \text{space}(\omega(\log)) \) and \( A_{\Pi k} \cup B_{\Pi k} \notin \Pi_{k+1} \text{space}(\omega(\log)) \). This proves (2) and (3).

Property (4) for \( \Sigma_k \) classes follows from the fact that for any \( k \geq 2 \) \( L_{\Sigma k} \{0\} = A_{\Sigma k} \cap B_{\Sigma k} \) does not belong to \( \Sigma_k \text{space}(\omega(\log)) \), but \( L_{\Sigma k} \in \Sigma_k \text{space}(\log) \). To see that \( \Pi_k \text{space}(S) \) is not closed under concatenation define the languages

\[
L^1_k := L_k \cup \{\varepsilon\}
\]
where ε denotes the empty string and

\[ L_k^2 := \{ w_1 w_2 0 \ldots w_p 0 \mid p \in \mathbb{N}, w_i \in L_{k-1} \text{ and } w_1 \in L_{\Pi k-1} \} . \]

Obviously, both languages belong to Π_k Space(\log), but from Theorem 3.5 follows

\[ L_k^1 L_k^2 = L_{\Sigma k} \notin \Pi_k \text{Space}(o(\log)) . \]

5. Lower Space Bounds for Context-Free Languages.

**Proposition 5.1.** \( L_\neq = \{ 1^n 01^m : n \neq m \} \notin A\text{Space}(o(\log)). \)

*Proof.* Let us assume, to the contrary, that \( L_\neq \) is recognized by an ATM A in space \( S \), for some \( S \in o(\log) \). Let \( S'(n) := S(2n + 1) \). Obviously, \( S' \in o(\log) \). Let \( \hat{n} := N_{M,S'} \). Then by the Small-Space-Bound-Lemma for all \( k, \ell \geq 0 \)

(i) \[ \text{Space}_A(1^{\hat{n}} 01^r) = \text{Space}_A(1^{\hat{n}+k\hat{n}} 01^{\hat{n}+\ell\hat{n}}). \]

Let

(ii) \[ \hat{s} = \text{Space}_A(1^{\hat{n}} 01^r). \]

For this fixed \( \hat{n} \) we define the language \( \hat{L} = \{ 1^{\hat{n}+k\hat{n}} 01^{\hat{n}+\ell\hat{n}} : k, \ell \in \mathbb{N} \text{ and } k \neq \ell \} \), and construct an automaton \( \hat{A} \) that recognizes \( \hat{L} \). \( \hat{A} \) performs the following algorithm:

**Step 1.** Check deterministically if the input \( X \) has the form \( 1^{\hat{n}+k\hat{n}} 01^{\hat{n}+\ell\hat{n}} \) for some integers \( k \) and \( \ell \); reject and stop if this condition does not hold.

**Step 2.** Move the head to the first symbol of the input and start to simulate the machine \( A \).

It is obvious that \( \hat{A} \) accepts an input \( X = 1^{\hat{n}+k\hat{n}} 01^{\hat{n}+\ell\hat{n}} \) if and only if \( A \) accepts \( X \). Hence we have \( L(\hat{A}) = \hat{L} \). It is easy to see that step 1 can be performed within space \( O(\log \hat{n}) \), which is a constant. Moreover from (i) and (ii) it follows that step 2 also requires only constant space \( \hat{s} \). Hence \( \hat{A} \) recognizes \( \hat{L} \) within constant space. We get a contradiction, since \( \hat{L} \) is nonregular. \( \square \)

Using a similar proof one can show that the language

\[ L_\epsilon := \{ 1^n 01^n : n \in \mathbb{N} \} \]

is not in \( A\text{Space}(o(\log)) \), too.

The rest of this section is devoted to the lower space bounds for a large subset of nonregular context-free languages.

The block structure of a bounded language \( L \) can equivalently be represented using a finite alphabet \( \{ a_1, \ldots, a_r \} \). Then \( L \) is a subset of \( \{ a_1 \}^* \ldots \{ a_r \}^* \).

**Definition 5.2.** Let \( V(L) \) denote the set \( \{ (v_1, \ldots, v_r) \in \mathbb{N}^r \mid a_1 v_1^* \ldots a_r v_r \in L \} \). Sets of the form \( \{ \alpha + n_1 \beta_1 + \ldots + n_k \beta_k \mid n_1, \ldots, n_k \in \mathbb{N} \} \) with \( \alpha, \beta_1, \ldots, \beta_k \in \mathbb{N}^r \), are called linear sets. A finite union of linear sets is a semilinear set. A language \( L \) is semilinear if \( L \subseteq \{ a_1 \}^* \ldots \{ a_r \}^* \) and \( V(L) \) is a semilinear set.

**Proposition 5.3.** Let a language \( L \subseteq \{ a_1 \}^* \ldots \{ a_r \}^* \) be semilinear and let \( L, \Sigma \in A\text{Space}(S) \) for some \( S \in o(\log) \). Then \( L \) is regular.

*Proof.* For \( r = 1 \) the proposition is true because every semilinear tally language is regular. Let us assume that \( r > 1 \) and that the proposition holds for \( r - 1 \). Sets of the form

\[ \{ \alpha + q_1 \gamma_1 + \ldots + q_k \gamma_k \mid q_1, \ldots, q_k \in \mathbb{R}^+ \} \]
with \(\gamma_1, \ldots, \gamma_k \in \mathbb{N}^r\) are called cones (see [1]). Assume now, to the contrary, that \(L\) is nonregular. To show that this cannot occur we first construct a semilinear language \(\hat{L} \in \text{ASpace}(S)\) that is also nonregular and for which there exists an \(r\)-dimensional cone \(C\) such that \(V(\hat{L}) \cap C = \emptyset\). To this end, methods developed by Alt and Mehlhorn in [1], [4] will be used.

**Lemma ([1]).** There exists an \(r\)-dimensional cone \(C\) and a regular language \(R \subseteq \{a_1\}^* \ldots \{a_r\}^*\) with 
\[
V(L) \cap C = V(R) \cap C.
\]

Let \(L\) and \(C\) be as in the lemma. Define \(L_1 := L \setminus R\) and \(L_2 := R \setminus L\). Obviously \(L_1\) or \(L_2\) is nonregular since \(L\) is nonregular. We set \(\hat{L} := L_1\) if \(L_1\) is nonregular and \(\hat{L} := L_2\) otherwise. The language \(\hat{L}\) is semilinear since the class of semilinear sets is closed under Boolean operations ([12]). Moreover, \(\hat{L} \in \text{ASpace}(S)\), because \(L, \overline{L} \in \text{ASpace}(S)\) and \(V(\hat{L}) \cap C = \emptyset\) for the \(r\)-dimensional cone \(C\).

**Definition 5.4.** Let us call a set \(K \subseteq \mathbb{N}^r\) extended if there exists \(\alpha \in \mathbb{N}^r\) and \(\beta \in \mathbb{N}^r_+\) such that 
\[
\forall \ k \in \mathbb{N} \quad \alpha + k\beta \in K.
\]

**Remark.** In [1] a different definition of extended set has been used. However it is easy to check that both definitions are equivalent.

If \(V(\hat{L})\) is not extended then one can show similarly as in [1] that there exists a nonregular language in \(\{a_1\}^* \ldots \{a_r\}^*\) fulfilling the assumptions of the proposition. Hence, by the inductive hypothesis we obtain a contradiction. Therefore, we can assume that \(V(\hat{L})\) is extended. Let \(\alpha = (\alpha_1, \ldots, \alpha_r)\) and \(\beta = (\beta_1, \ldots, \beta_r)\) with \(\alpha_1, \ldots, \alpha_r \in \mathbb{N}\) and \(\beta_1, \ldots, \beta_r \in \mathbb{N}^r_+\), be vectors such that 
\[
\forall \ k \in \mathbb{N} \quad \alpha + k\beta \in V(\hat{L}).
\]
Moreover, let \(\hat{M}\) be an ATM which recognizes \(\hat{L}\) in space \(S\). Define the function \(S'\) by 
\[
S'(n) := S\left(\sum_{i=1}^{r} \alpha_i + n \sum_{i=1}^{r} \beta_i\right).
\]
Since \(S \in o(\log )\) also \(S' \in o(\log )\). Let \(\hat{n} := N_{\hat{M}, S'}\). Then we define 
\[
\hat{R} := \{a_1^{\alpha_1+(\hat{n}+\ell_1\hat{\beta}_1)} \ldots a_r^{\alpha_r+(\hat{n}+\ell_r\hat{\beta}_r)} | \ell_1, \ldots, \ell_r \in \mathbb{N}\} \quad \text{and} \quad \hat{L} := \hat{R} \cap \hat{L}.
\]
A contradiction will be obtained from the following claims

**Claim 1.** \(\hat{L}\) can be recognized in constant space.

**Claim 2.** \(\hat{L}\) is nonregular.

**Proof (of Claim 1).** Using for every \(i = 1, \ldots, r\) \(\beta_i\)-times the Small-Space-Bound Lemma we obtain that for any sequence of integers \(\ell_i \geq 0\)
\[
(i) \hat{s} := \text{Space}_{\hat{M}}(a_1^{\alpha_1+(\hat{n}+\ell_1\hat{\beta}_1)} \ldots a_r^{\alpha_r+(\hat{n}+\ell_r\hat{\beta}_r)}) = S'(a_1^{\alpha_1+(\hat{n}\hat{\beta}_1)} \ldots a_r^{\alpha_r+(\hat{n}\hat{\beta}_r)}).
\]
Let \(\hat{M}\) be an ATM which performs the following algorithm:
Step 1. Check deterministically if the input $X$ has the form
\[
a_1^{α_1+(\bar{n}+ε_1n!)}β_1 \ldots a_r^{α_r+(\bar{n}+ε_rn!)}β_r
\]
for some integers $ε_1, \ldots, ε_r$. Reject and stop if this condition does not hold.

Step 2. Move the head to the first symbol of the input and start to simulate the machine $\hat{M}$.

It is obvious that $\hat{M}$ accepts an input $X = a_1^{α_1+(\bar{n}+ε_1n!)}β_1 \ldots a_r^{α_r+(\bar{n}+ε_rn!)}β_r$ iff $\hat{M}$ accepts $X$. Hence, we have $L(\hat{M}) = \hat{L}$. It is easy to see that step 1 can be performed within space $O(\log \bar{n}!)$, which is a constant. Moreover from (i) it follows that step 2 also requires only constant space $\hat{s}$. Hence $\hat{M}$ recognizes the language $\hat{L}$ within constant space.

Proof (of Claim 2). A set of the form
\[
\{ γ + (k_1δ_1, \ldots, k_rδ_r) | k_1, \ldots, k_r \in \mathbb{N} \}
\]
with $γ \in \mathbb{N}^r$ and $δ_1, \ldots, δ_r \in \mathbb{N}$ is called a grid. We show that if $\hat{L}$ is regular then there exists an $r$-dimensional grid in $V(\hat{L})$.

Assume that $\hat{L}$ is regular. Then, using the pumping lemma for regular languages one can show that there exist integers $ℓ ≥ 0$ and $δ_1, \ldots, δ_r > 0$ such that for all $k_1, \ldots, k_r ≥ 0$
\[
α + (\bar{n} + ℓ\bar{n}!)β + (k_1δ_1, \ldots, k_rδ_r) \in V(\hat{L}).
\]
Therefore, the $r$-dimensional grid $G := \{ γ + (k_1δ_1, \ldots, k_rδ_r) | k_1, \ldots, k_r \in \mathbb{N} \}$ with $γ = α + (\bar{n} + ℓ\bar{n}!)β$ is a subset of $V(\hat{L})$, which implies $G \subseteq V(\hat{L})$. From this and the property $V(\hat{L}) \cap C = \emptyset$ shown above we obtain that $G \cap C = \emptyset$ for the $r$-dimensional cone $C$. This yields a contradiction to the following result.

Lemma ([1]). Let $G \subseteq \mathbb{N}^r$ be an $r$-dimensional grid and let $C \subseteq \mathbb{N}^r$ be an $r$-dimensional cone. Then $G \cap C ≠ \emptyset$.

This completes the proof of the proposition.

Recall that a language $L$ is called strictly nonregular if there are strings $u, v, w, x$ and $y$ such that $L \cap \{ u \}^* \{ v \}^* \{ w \}^* \{ x \}^* \{ y \}$ is context-free and nonregular. It was shown by Stearns ([20]) that every nonregular deterministic context-free language is strictly nonregular. Therefore, from the proposition above we obtain immediately that if $L$ is a nonregular deterministic context-free, a strictly nonregular language, or a nonregular context-free bounded language, then for ATMs without any bound on the number of alternations it is not possible that $L$ and $\overline{L}$ both belong to $\text{ASpace}(o(\log n))$. Moreover, from Theorem 2.13 it follows that the class of languages recognized by space-bounded ATMs with a constant number of alternations is closed under complement. Hence it follows that the language $L$ does not belong to $\bigcup_{k \in \mathbb{N}} \Sigma_k \text{Space}(o(\log n))$. This completes the proof of Theorem 1.13.

6. Conclusions. The obvious question remaining is how classes $\Sigma_k \text{Space}(S)$ and $Π_i \text{Space}(S)$ compare. It is somewhat annoying that the techniques developed in this paper do not give any help for the case $k = 1$. It is not completely unrealistic to believe that both classes may be equal, which would give the novel result that a hierarchy is infinite, although its first level collapses.

If one restricts to bounded languages $\Sigma_k \text{Space}(S)$ is closed under complementation and both classes are identical, which has been shown in [2] and [23]. But for $k = 2$ the situation changes completely. The languages $L_{Σ_2}$ and $L_{Π_2}$ are unary...
the most stringent form of a bounded language—and still separate $\Sigma_2^\text{Space}(S)$ from $\Pi_2^\text{Space}(S)$. Thus a separation of the first level would require a syntactically more complex languages than the second level. For $k > 2$ the languages $L_{\Sigma^k}$ and $L_{\Pi^k}$ used in this paper to establish the separation are no longer bounded. But by Proposition 4.1 the third level can also be separated using simple bounded languages $A_{\Sigma^2} \cap B_{\Sigma^2}$ and $A_{\Pi^2} \cup B_{\Pi^2}$ that both are subsets of $\{1\}^* \{0\} \{1\}^*$.

Nothing seems to be known for level 4 and higher. Thus, the sublogarithmic space hierarchy for bounded languages may be even more complex. We have made some observations leading to the conjecture that for bounded languages this hierarchy might indeed consist of only a finite number of distinct levels.

Finally, it would be nice to characterize the exact relationship between classes $\text{co-}\Sigma_k^\text{Space}(S)$ and $\Pi_k^\text{Space}(S)$ for sublogarithmic space bounds $S$ and the class of arbitrary languages.

REFERENCES

[3] H. Alt and K. Mehlhorn, A language over a one symbol alphabet requiring only $O((\log \log n)$ space, SIGACT Newsletter, 1975, pp. 31–33.
[8] V. Geffert, Nondeterministic computations in sublogarithmic space and space constructabil-