## References


1. H. Alt, *Lower bounds on space complexity for context-free recognition,* Acta Inform. 12, 1979, 33-61.
2. H. Alt, V. Geffert, and K. Mehlhorn, *A lower bound for the nondeterministic space complexity of context-free recognition,* Inform. Process. Lett. 42, 1992, 25-27.
3. H. Alt, and K. Mehlhorn, *Lower bounds for the space complexity of context free recognition,* Proc. 3rd ICALP, 1976, 339-354.
4. B. von Braunmühl, *Alternation for two-way machines with sublogarithmic space*, Proc. 10. STACS, Würzburg, 1993, 5-15.
5. B. von Braunmühl, R. Gengler, and R. Rettinger *The alternation hierarchy for machines with sublogarithmic space is infinite,* Research Report, Universität Bonn, January, 1993.
6. J. Chang, O. Ibarra, B. Ravikumar, and L. Berman, *Some observations concerning alternating Turing machines using small space,* Inform. Proc. Letters 25, 1987, 1-9.
7. R. Freivalds, *Fast Probabilistic Algorithms,* Proc. 8. MFCS, 1979, 57-69.
8. V. Geffert, *Nondeterministic computations in sublogarithmic space and space constructability,* SIAM J. Comput. 20, 1991, 484-498.
9. V. Geffert, *Sublogarithmic $\Sigma_2$ -space is not closed under complement and other separation results,* Technical Report, University of Safarik, 1992.
10. V. Geffert, *Tally version of the Savitch and Immerman-Szelepcsényi theorems for sublogarithmic space,* SIAM J. Comput. 22, 1993, 102-113.
11. V. Geffert, *A hierarchy that does not collapse: alternations in low level space,* manuscript.
12. S. Ginsburg, *The mathematical theory of context-free languages,* McGraw-Hill, 1972.
13. N. Immerman, *Nondeterministic space is closed under complementation,* SIAM J. Comput. 17, 1988, 935-938.
14. M. Liśkiewicz, and R. Reischuk, *Separating the lower levels of the sublogarithmic space hierarchy,* Technical Report, Technische Hochschule Darmstadt, Institut für Theoretische Informatik, 1992, see also Proc. 10. STACS, Würzburg, 1993, 16-27.
15. M. Liśkiewicz, and R. Reischuk, *The sublogarithmic space hierarchy is infinite,* Technical Report, Technische Hochschule Darmstadt, Institut für Theoretische Informatik, January 1993.
16. B. Litow, *On efficient deterministic simulation of Turing machine computations below logspace,* Math. Systems Theory 18, 1985, 11- 18.
17. P. Michel, *A survey of space complexity,* Theoret. Comput. Sci. 101, 1992, 99-132.
18. D. Ranjan, R. Chang, and J. Hartmanis, *Space bounded computations: review and new separation results,* Theoret. Comput. Sci. 80, 1991, 289-302.
19. M. Sipser, *Halting space-bounded computations,* Theoret. Comput. Sci. 10, 1980, 335-338.
20. R. Stearns, *A regularity test for pushdown-machines,* Information and Control, 11, 1967, 323-340.
21. R. Stearns, J. Hartmanis, and P. Lewis, *Hierarchies of memory limited computations,* Proc. 1965 IEEE Conf. Record on Switching Circuit Theory and Logical Design, 1965, 179-190.
22. R. Szelépcsenyi, *The method of forced enumeration for nondeterministic automata,* Acta Informatica 26, 1988, 279-284.
23. A. Szepietowski, *Turing machines with sublogarithmic space,* unpublished manuscript.
24. K. Wagner, Editorial note: *The alternation hierarchy for sublogarithmic space: an exciting race to STACS'93,* Proc. 10. STACS, Würzburg, 1993, 2-4.
25. K. Wagner, and G. Wechsung, *Computational complexity,* Reidel, Dordrech, 1986.

Hence, the $r$-dimensional grid $G := \{\gamma+(k_1\delta_1,\ldots,k_r\delta_r) \mid k_1,\ldots,k_r \in \mathbb{N}\}$ with $\gamma = \alpha+(\hat{n}+\ell\hat{n}!)\beta$ is a subset of $V(\hat{L})$, which implies $G \subseteq V(\tilde{L})$. From this and the property $V(\tilde{L}) \cap C = \emptyset$ 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 $G \cap C \neq \emptyset$. ∎

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 $ASpace(o(\log))$. Moreover, from Theorem 7 it follows that the class of languages recognized by space-bounded ATMs with a constant number of alternations is closed under complement. Hence it follws that the language $L$ does not belong to $\bigcup_{k\in\mathbb{N}} \Sigma_k Space(o(\log))$. This completes the proof of Theorem 6.

# 6 Conclusions

The obvious question remaining is how $\Sigma_1 Space(S)$ and $\Pi_1 Space(S)$ 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 $\Sigma_1 Space(S)$ 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 $L_{\Sigma 2}$ and $L_{\Pi 2}$ are unary – the most stringent form of a bounded language – and still separate $\Sigma_2 Space(S)$ from $\Pi_2 Space(S)$. Thus a separation of the first level would require a syntatically 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 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 sublogartihmic 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 co-$\Sigma_k Space(S)$ and $\Pi_k Space(S)$ for sublogarithmic space bounds $S$ and the class of arbitrary languages.

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

If $V(\tilde{L})$ is not extended then one can show similarly as in [1] that there there exists a nonregular language in $\{a_1\}^* \ldots \{a_{r-1}\}^*$ fulfilling the assumptions of the proposition. Hence, by the inductive hypothesis we obtain a contradiction. Therefore, we can assume that $V(\tilde{L})$ 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 $\beta_1, \ldots, \beta_r \in \mathbb{N}_+$, be vectors such that

$$\forall\, k \in \mathbb{N} \quad \alpha + k\beta \in V(\tilde{L}) \ .$$

Moreover, let $\tilde{M}$ be an ATM which recognizes $\tilde{L}$ in space $S$. Define the function $S'$ by

$$S'(n) \ := \ S\Big(\sum_{i=1}^{r} \alpha_i \ + \ n \ \sum_{i=1}^{r} \beta_i\Big) \ .$$

Since $S \in o(\log)$ also $S' \in o(\log)$. Let $\hat{n} := \mathcal{N}_{\tilde{M},S'}$. Then we define

$$\hat{R} \ := \ \{a_1^{\alpha_1+(\hat{n}+\ell_1\hat{n}!)\beta_1} \ldots \ a_r^{\alpha_r+(\hat{n}+\ell_r\hat{n}!)\beta_r} \mid \ell_1, \ldots, \ell_r \in \mathbb{N}\} \quad \text{and} \quad \hat{L} \ := \ \hat{R} \cap \tilde{L} \ .$$

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 $\ell_i \geq 0$

$$\hat{s} \ := \ Space_{\tilde{M}}(a_1^{\alpha_1+(\hat{n}+\ell_1\hat{n}!)\beta_1} \ldots \ a_r^{\alpha_r+(\hat{n}+\ell_r\hat{n}!)\beta_r}) \ = \ S'(a_1^{\alpha_1+\hat{n}\beta_1} \ldots \ a_r^{\alpha_r+\hat{n}\beta_r}) \ . \qquad \text{(i)}$$

Let $\hat{M}$ be an ATM which performs the following algorithm:

**Step 1.** Check deterministically if the input $X$ has the form $a_1^{\alpha_1+(\hat{n}+\ell_1\hat{n}!)\beta_1} \ldots \ a_r^{\alpha_r+(\hat{n}+\ell_r\hat{n}!)\beta_r}$ for some integers $\ell_1, \ldots, \ell_r$.
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 $\tilde{M}$.

It is obvious that $\hat{M}$ accepts an input $X = a_1^{\alpha_1+(\hat{n}+\ell_1\hat{n}!)\beta_1} \ldots a_r^{\alpha_r+(\hat{n}+\ell_r\hat{n}!)\beta_r}$ iff $\tilde{M}$ accepts $X$. Hence, we have $L(\hat{M}) = \hat{L}$. It is easy to see that step 1 can be performed within space $O(\log \hat{n}!)$, which is a constant. Moreover from (i) it follows that step 2 also requires only constant space $\hat{s}$. Hence $\hat{M}$ recognizes $\hat{L}$ within constant space. ∎

**Proof of Claim 2:** A set of the form

$$\{\gamma + (k_1\delta_1, \ldots, k_r\delta_r) \mid k_1, \ldots, k_r \in \mathbb{N}\}$$

with $\gamma \in \mathbb{N}^r$ and $\delta_1, \ldots, \delta_r \in \mathbb{N}$ is called a *grid*. We show that if $\hat{L}$ is regular then there exists an $r$-dimensional grid in $V(\tilde{L})$.

Assume that $\hat{L}$ is regular. Then, using the pumping lemma for regular languages one can show that there exist integers $\ell \geq 0$ and $\delta_1, \ldots, \delta_r > 0$ such that for all $k_1, \ldots, k_r \geq 0$

$$\alpha + (\hat{n} + \ell\hat{n}!)\beta + (k_1\delta_1, \ldots, k_r\delta_r) \ \in \ V(\hat{L}) \ .$$

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 non-regular. ∎

Using a similar proof one can show that the language

$$L_{=} \;:=\; \{1^n 0 1^n : n \in \mathbb{N}\}$$

is not in $ASpace(o(\log))$, 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 9** 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 6** Let $L \subseteq \{a_1\}^* \ldots \{a_r\}^*$ be semilinear and let $L, \overline{L} \in ASpace(S)$ 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 $\tilde{L} \in ASpace(S)$ that is also nonregular and for which there exists an $r$-dimensional cone $C$ such that $V(\tilde{L}) \cap C = \emptyset$. To this end, methods developed by Alt and Mehlhorn in [1], [3] 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 $R$ 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 $\tilde{L} := L_1$ if $L_1$ is nonregular and $\tilde{L} := L_2$ otherwise. The language $\tilde{L}$ is semilinear since the class of semilinear sets is closed under Boolean operations ([12]). Moreover, $\tilde{L} \in ASpace(S)$, because $L, \overline{L} \in ASpace(S)$ and $V(\tilde{L}) \cap C = \emptyset$ for the $r$-dimensional cone $C$.

**Definition 10** 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

$$\forall\, k \in \mathbb{N} \quad \alpha + k\beta \in K \;.$$

1. $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ are not closed under complementation.
2. $\Sigma_k Space(S)$ is not closed under intersection,
3. $\Pi_k Space(S)$ is not closed under union.
4. $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ are not closed under concatenation.

(1) follows immediately from Lemma 12, Theorem 8 and the following equations: $L_{\Sigma k} = L_k \cap \overline{L}_{\Pi k}$, and $L_{\Pi k} = L_k \cap \overline{L}_{\Sigma k}$, where $L_k$ is the regular language introduced in Definition 6.

By (i) $A_{\Sigma k}, B_{\Sigma k} \in \Sigma_k Space(\text{llog})$ and $A_{\Pi k}, B_{\Pi k} \in \Pi_k Space(\text{llog})$. On the other hand, from Proposition 4 $A_{\Sigma k} \cap B_{\Sigma k} \notin \Sigma_{k+1} Space(o(\log))$ and $A_{\Pi k} \cup B_{\Pi k} \notin \Pi_{k+1} Space(o(\log))$. This proves (2) and (3).

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

$$L_k^1 := L_k \cup \{\varepsilon\}$$

where $\varepsilon$ denotes the empty string and

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

Obviously, both languages belong to $\Pi_k Space(\text{llog})$, but from Theorem 8 follows

$$L_k^1 \, L_k^2 = L_{\Sigma k} \notin \Pi_k Space(o(\log)) .$$

∎

# 5 Lower Space Bounds for Context-Free Languages

**Proposition 5** $L_{\neq} = \{1^n 0 1^m : n \neq m\} \notin ASpace(o(\log))$.

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

$$Space_A(1^{\hat{n}} 0 1^{\hat{n}}) = Space_A(1^{\hat{n}+k\hat{n}!} 0 1^{\hat{n}+\ell\hat{n}!}). \tag{i}$$

Let

$$\hat{s} = Space_A(1^{\hat{n}} 0 1^{\hat{n}}). \tag{ii}$$

For this fixed $\hat{n}$ we define the following language $\hat{L} = \{1^{\hat{n}+k\hat{n}!} 0 1^{\hat{n}+\ell\hat{n}!} : k, \ell \in \mathbb{N} \text{ and } k \neq \ell\}$, 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 $1^{\hat{n}+k\hat{n}!} 0 1^{\hat{n}+\ell\hat{n}!}$ 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 $A$.

**Proof.** It is well known that for any function $S$ the classes $\Sigma_k Space(S)$ are closed under union, and symmetrically the $\Pi_k Space(S)$ are closed under intersection (see e.g. [25]). Hence by (i), $A_{\Sigma k} \cap B_{\Sigma k} \in \Pi_{k+1} Space(\text{llog})$ and $A_{\Pi k} \cup B_{\Pi k} \in \Sigma_{k+1} Space(\text{llog})$. To prove that $A_{\Sigma k} \cap B_{\Sigma k} \notin \Sigma_{k+1} Space(o(\log))$ and $A_{\Pi k} \cup B_{\Pi k} \notin \Pi_{k+1} Space(o(\log))$ first we modify Proposition 2 in the following way:

**Proposition 2′** Let $k \geq 2$ and $M$ be an ATM of space complexity $S$ with $S \in o(\log)$. Then there exists a bound $S'' \in o(\log)$ such that for all $n \geq \mathcal{N}_{M,S''}$ and words $W_1, W_2 \in \{W^n_{\Sigma k}, W^n_{\Pi k}\}$

$$Space_M(W_1\, 0\, W_2) \leq S''(n) .$$

**Proof.** Let $S''(n) := S(2p_k(n) + 1)$, where $p_k$ is the polynomial specified in the proof of Proposition 2. It is easy to check that the proof of Proposition 2 generalizes to this situation. ∎

Let us assume, to the contrary, that $A_{\Sigma k} \cap B_{\Sigma k} \in \Sigma_{k+1} Space(S)$, for some $S \in o(\log n)$. Let $M$ be an $S$ space-bounded $\Sigma_{k+1}$ TM for $A_{\Sigma k} \cap B_{\Sigma k}$. Choose $n \in \mathcal{F}$ sufficiently large. By Lemma 13 $W^n_{\Sigma k} \in L_{\Sigma k}$ 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

$$(\alpha_0, 0) \models_{M,X} (\beta, j) , \tag{ii}$$

and

$$\mathsf{acc}^k_M(\beta, j, X) . \tag{iii}$$

(The trivial case that $M$ accepts $X$ without alternation could be handled similarly.) Now let $Y_1 := W^n_{\Sigma k}\, 0\, W^n_{\Pi k}$ and $Y_2 := W^n_{\Pi k}\, 0\, W^n_{\Sigma k}$. By Proposition 2′ there exists $S'' \in o(\log n)$ such that

$$Space_M(X),\ Space_M(Y_1),\ Space_M(Y_2) \quad \leq \quad S''(n) .$$

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

$$(\alpha_0, 0) \models_{M,Y_1} (\beta, j) \quad \text{and} \quad \mathsf{acc}^k_M(\beta, j, Y_1)$$

if $j \leq |W^n_{\Sigma k}\, 0|$ and otherwise

$$(\alpha_0, 0) \models_{M,Y_2} (\beta, \hat{j}) \quad \text{and} \quad \mathsf{acc}^k_M(\beta, \hat{j}, Y_2) ,$$

where $\hat{j} = j + |Y_2| - |X|$. Hence $M$ also accepts input $Y_1$ or $Y_2$. This yields a contradiction since, by Lemma 13, $Y_1, Y_2 \notin A_{\Sigma k} \cap B_{\Sigma k}$.

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

∎

This result can be applied to **prove Theorem 3:**

For all $k \geq 2$ and any $S \in \texttt{SUBLOG}$ holds:

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

$$\begin{aligned} k &:= A(m) \\ n &:= f(m) . \end{aligned}$$

By (i) and (ii) $n \geq \mathcal{N}_{M,S'}$. 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 2. 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

$$\begin{aligned} Space_M(X) &\leq S(m) \leq \max\{S(m') \mid f(m') = n\} = S'(n) \qquad \text{and} \\ Alter_M(X) &\leq B(m) - 1 \leq \exp S(m) \leq \exp S'(n) . \end{aligned}$$

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 2 one can show that

$$Space_M(W_{\Sigma k}^n\ 0^{t(m)}) ,\ Space_M(W_{\Pi k}^n\ 0^{t(m)}) = Space_M(X) \leq S'(n) .$$

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

$$Alter_M(W_{\Sigma k}^n\ 0^{t(m)}) ,\ Alter_M(W_{\Pi k}^n\ 0^{t(m)}) = Alter_M(X) \leq B(m) - 1 = k - 1 .$$

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 13 holds $W_{\Pi k}^n \in L_{\Pi k}$, hence $M$ has to accept $W_{\Pi k}^n\ 0^{t(m)}$ But this means that $\mathtt{acc}_M^k(\alpha_0, 0, W_{\Pi k}^n\ 0^{t(m)})$ is true, where $(\alpha_0, 0)$ is the initial configuration of $M$. From Proposition 1 we conclude that $\mathtt{acc}_M^k(\alpha_0, 0, W_{\Sigma k}^n\ 0^{t(m)})$ holds, too. Therefore $M$ accepts $W_{\Sigma k}^n\ 0^{t(m)}$, which by Lemma 13 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 $W_{\Pi k}^n\ 0^{t(m)}$. ∎

# 4 Closure Properties

In this section we discuss closure properties of $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ classes for sublogarithmic bounds $S$. First for any integer $k \geq 2$ 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

$$A_{\Sigma k}, B_{\Sigma k} \in \Sigma_k Space(\text{llog}) \quad \text{and} \quad A_{\Pi k}, B_{\Pi k} \in \Pi_k Space(\text{llog}) . \tag{i}$$

**Proposition 4** For all $k \geq 2$ holds:

$$\begin{aligned} A_{\Sigma k} \cap B_{\Sigma k} &\in \Pi_{k+1} Space(\text{llog}) \setminus \Sigma_{k+1} Space(o(\log)) , \\ A_{\Pi k} \cup B_{\Pi k} &\in \Sigma_{k+1} Space(\text{llog}) \setminus \Pi_{k+1} Space(o(\log)) . \end{aligned}$$

For $m \geq m_0$ we can bound $h$ by

$$h(m) \;\leq\; \exp\left(\frac{\log m}{2}\right) \;=\; m^{1/2} \;,$$
$$h(m) \;\geq\; \frac{\exp \operatorname{llog} m}{3 \log m \;/\; \operatorname{llog} m} \;=\; \frac{\operatorname{llog} m}{3} \;\geq\; 3 \;.$$

and hence $f(m) \in \mathcal{F}$. Moreover, from Lemma 14 it follows

$$f(m) \;\geq\; h(m)^{1/4} \;\geq\; \left(\frac{1}{3} \operatorname{llog} m\right)^{1/4} \;. \tag{i}$$

Define the function $S' : \mathbb{N} \to \mathbb{N}$ as follows

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

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

**Lemma 16** $S' \in o(\log)$.

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

$$S \in o\left(\frac{\log}{A}\right) \qquad \text{and} \qquad \log A \;\leq\; \operatorname{llog} m \;\leq\; S \;.$$

This implies

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

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

$$t(m) \;:=\; m - p_{A(m)}(f(m))$$

where $p_d(n)$ has already been defined in the proof of Proposition 2, and note that

$$p_{A(m)}(f(m)) \;\leq\; (3\; A(m)\; f(m))^{A(m)} \;\leq\; m \;.$$

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

$$m \;\geq\; \max\{m_0,\; \exp\exp 3(\mathcal{N}_{M,S'})^4\} \qquad \text{and} \qquad A(m) = B(m) \tag{ii}$$

Obviously, for all $k \geq 3$ holds $F(\ell_k) = p_k$, and furthermore, $F(n) < p_k$ for any $n < \ell_k$. Therefore $\ell_k \in \mathcal{F}$. Since it is well known that $p_{k+1} \leq 2p_k$ we can conclude

$$p_{k+1} \leq 2\, p_i^{r_{k,i}+1} \ ,$$

which implies the upper bound for $\ell_{k+1}$:

$$\ell_{k+1} \ = \ \prod_{i=1}^{k} p_i^{r_{k+1,i}} \ \leq \ \ell_k \cdot 2^{k-1} \cdot p_k \cdot \prod_{i=1}^{k-1} p_i \ \leq \ (\ell_k)^4 \ .$$

∎

**Definition 8** Let $A : \mathbb{N} \to \mathbb{N}$ be a function with $A(n) \geq 2$ for all $n$ and define

$$\begin{aligned} L_\Sigma(A) &:= \{X \mid X = W0^r \text{ for some } r \in \mathbb{N} \text{ and } W \in L_{\Sigma k} \text{ for some } k \leq A(|X|)\} \ , \\ L_\Pi(A) &:= \{X \mid X = W0^r \text{ for some } r \in \mathbb{N} \text{ and } W \in L_{\Pi k} \text{ for some } k \leq A(|X|)\} \ . \end{aligned}$$

The separating results for $A$-alternation-bounded space classes (Theorem 2) follow from the propositions below.

**Lemma 15** For any $S \in \texttt{SUBLOG}$ and all functions $A \geq 2$ computable in space $S$ holds:

$$\begin{aligned} L_\Sigma(A) &\in \Sigma_A Space(S) \ , \\ L_\Pi(A) &\in \Pi_A Space(S) \ . \end{aligned}$$

**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** For any $S \in \texttt{SUBLOG}$ and for all functions $A$ and $B$ with $1 < A \leq_* B$ and $B \cdot S \in o(\log)$ holds:

$$\begin{aligned} L_\Sigma(A) &\notin \Pi_B Space(S) \ , \\ L_\Pi(A) &\notin \Sigma_B Space(S) \ . \end{aligned}$$

**Proof.** Let $S \in \texttt{SUBLOG}$ 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 $A(m) < \frac{\log m}{\operatorname{llog} m}$ for all $m \geq m_0$. Define functions $h$ and $f$ as follows

$$\begin{aligned} h(m) &:= \frac{\exp\left(\frac{\log m}{A(m)}\right)}{3\, A(m)} \ , \\ f(m) &:= \max\{\ell \mid \ell \in \mathcal{F} \cup \{0\},\ \ell \leq h(m)\} \ . \end{aligned}$$

Therefore, by (i), we obtain that

$$Space_M(V_k^{k-1}(n)) \ \leq \ S'(n) \ . \qquad \text{(ii)}$$

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

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

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

Now we are ready to prove Theorem 8. Let us assume that $M$ is a $\Sigma_k$ TM accepting $L_{\Pi k}$ in sublogarithmic space $S$. By Proposition 2 there exists a function $S' \in o(\log)$ such that for any $n \geq \mathcal{N}_{M,S'}$

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

Let $n$ with $n \in \mathcal{F}$ be an integer larger than $\mathcal{N}_{M,S'}$ (such an $n$ exists since $\mathcal{F}$ is infinite). By Lemma 13 $W^n_{\Pi k} \in L_{\Pi k}$, hence $M$ has to accept $W^n_{\Pi k}$, which means that $\mathsf{acc}^k_M(\alpha_0, 0, W^n_{\Pi k})$ is true, where $(\alpha_0, 0)$ is the initial configuration of $M$. From Proposition 1 we conclude that $\mathsf{acc}^k_M(\alpha_0, 0, W^n_{\Sigma k})$ holds, too, and hence $M$ accepts $W^n_{\Sigma k}$, which by Lemma 13 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 $W^n_{\Pi k}$. ∎

## 3.2 Unbounded Number of Alternations

In the previous section we have proved the lower space bounds for recognizing $L_{\Sigma k}$ and $L_{\Pi k}$ on ATMs with a constant number of alternation. These results hold for all languages $L_{\Sigma k}$ and $L_{\Pi k}$ defined on the base of a subset of natural numbers $\mathcal{F}$ with the properties as in Definition 6. In this section we fix the set $\mathcal{F}$ to the example given at the beginning of this section:

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

The following property of $\mathcal{F}$ will be useful in a proof of lower space bounds on ATMs working with unbounded number of alternations.

**Lemma 14** *For any $x \geq 3$ there exists an $\ell \in \mathcal{F}$ such that*

$$x^{1/4} \leq \ell \leq x \ .$$

**Proof.** For $x < 12$ the claim holds choosing $\ell = 3$. To prove it for $x \geq 12$ we will construct a sequence $\ell_3, \ell_4, \ell_5, \ldots$ of elements in $\mathcal{F}$ such that $\ell_3 = 12$ and for all $k \geq 3$

$$\ell_{k+1} \leq (\ell_k)^4 \ .$$

Let $p_i$ denote the $i$-th prime number and define for $k \geq 3$

$$\ell_k \ := \ \prod_{i=1}^{k-1} p_i^{r_{k,i}} \qquad \text{with} \quad r_{k,i} \ := \ \lfloor \log_{p_i} p_k \rfloor \ .$$

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

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

**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 multiplies 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

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

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:

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

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 $n \geq \mathcal{N}_{M,S'}$.

Since $M$ is $S$ space-bounded

$$Space_M(V^1_k(n)) \ \leq \ S(p_k(n)) \ = \ S'(n) \ . \tag{i}$$

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

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

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

$$\begin{aligned} V^i_k(n) \ &= \ W^{a+n}\ Z_1\ W^{a+n}\ Z_2 \ldots Z_{r-1}\ W^{a+n}\ Z_r \ , \\ V^{i+1}_k(n) \ &= \ W^{a+n+bn!}\ Z_1\ W^{a+n+bn!}\ Z_2 \ldots Z_{r-1}\ W^{a+n+bn!}\ Z_r \ . \end{aligned}$$

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

$$Space_M(V^i_k(n)) \ \leq \ S'(n) \qquad \Longrightarrow \qquad Space_M(V^{i+1}_k(n)) \ \leq \ S'(n).$$

We can assume that

$$j \le |U'| \quad \text{or} \quad j > |U'\,W^n_{\Sigma k-1}| \tag{vi}$$

because if $|U'| < j \le |U'\,W^n_{\Sigma k-1}|$ 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 $\mathtt{acc}^k_M(\alpha, \hat{i}, Y)$. ∎

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

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

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

**Case 1.** $j \le |U'|$ 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 $\mathtt{acc}^k_M(\alpha, \hat{i}, Y)$ implies

$$\mathtt{acc}^{k-1}_M(\beta, \tilde{j}, Y) \tag{vii}$$

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

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

**Case 2.** $|U'| < j \le |U'\,W^n_{\Sigma k-1}|$.

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

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

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

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

**A.)** First we consider existential configurations $(\alpha, i)$. Assume that

$$\mathtt{acc}^k_M(\alpha, i, X)$$

is true. Hence there exists an existential computation path from $(\alpha, i)$ to a final or universal configuration $(\beta, j)$:

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

with the property

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

We may assume that

$$j \leq |U'| \quad \text{or} \quad j > |U' \, W^n_{\Sigma k-1}| , \tag{iv}$$

because if $|U'| < j \leq |U' \, W^n_{\Sigma k-1}|$ then for $Z_1 := U$, $Z_2 := V$, $W := W^n_{\Sigma k-1}\, 0$, 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

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

Therefore, if $|U'| < j \leq |U' \, W^n_{\Sigma k-1}|$ 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 $\mathtt{acc}^k_M(\alpha, \hat{i}, Y)$ 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

$$\mathtt{acc}^{k-1}_M(\beta, \hat{i}, X) \quad \Longrightarrow \quad \mathtt{acc}^{k-1}_M(\beta, i, Y) \;=\; \mathtt{acc}^{k-1}_M(\beta, \tilde{j}, Y) .$$

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

**B.)** Now let us consider universal configurations $(\alpha, i)$, for which $\mathtt{acc}^k_M(\alpha, \hat{i}, Y)$ holds. We have to show that $\mathtt{acc}^k_M(\alpha, i, X)$ 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

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

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

$$\hat{i} \;=\; \begin{cases} i & \text{if } i \le |U|, \\ i + (|W^n_{\Sigma k}| - |W^n_{\Pi k}|) & \text{if } i > |U\ W^n_{\Pi k}|. \end{cases}$$

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 $W^n_{\Sigma k-1}$ and $W^n_{\Pi k-1}$. The argument concentrates only on the block in the middle of a $W^n_{\Sigma k}$ string, which is a $W^n_{\Pi k-1}$ word, and analogously for $W^n_{\Pi k}$ strings with a $W^n_{\Sigma k-1}$ 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 $W^n_{\Sigma k}$ and $W^n_{\Pi k}$ strings. But the Configuration- and Position-Shift-Lemmata 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 $n \ge \mathcal{N}_{M,S}$, and define

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

Note that $\hat{i}$ 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

$$|W^n_{\Pi k-1}| - |W^n_{\Sigma k-1}| \;=\; |W^n_{\Sigma k}| - |W^n_{\Pi k}|$$

$$\hat{i} \;=\; \tilde{i} \qquad \text{whenever both values are defined.} \tag{i}$$

First we prove the following

**Claim 1** For any memory state $|\alpha_1| \le |\alpha_2| \le S(n)$ and all $j_1, j_2$ with

$$j_1, j_2 \in\ [0\ \ldots\ |U'|]\ \cup\ [|U'\ W^n_{\Sigma k-1}| + 1\ \ldots\ |X| + 1]$$

holds:

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

**Proof.** For suitable $Z_1, Z_2 \in \{0,1\}^*$ the words considered can be written as

$$\begin{aligned} W^n_{\Sigma k} &= Z_1\ 1^n\ Z_2 \quad \text{and} \quad W^n_{\Pi k} = Z_1\ 1^{n+n!}\ Z_2 \qquad \text{if } k \text{ is odd, and} \\ W^n_{\Sigma k} &= Z_1\ 1^{n+n!}\ Z_2 \quad \text{and} \quad W^n_{\Pi k} = Z_1\ 1^n\ Z_2 \qquad \text{for even } k. \end{aligned}$$

The claim then follows from the Pumping Lemma (Lemma 3). ∎

**Theorem 8** For any $k \geq 2$ holds

$$\begin{aligned} L_{\Sigma k} &\notin \Pi_k Space(o(\log)) , \\ L_{\Pi k} &\notin \Sigma_k Space(o(\log)) . \end{aligned}$$

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

**Definition 7** For $n \in \mathcal{F}$ define words

$$W^n_{\Sigma 2} := 1^{n+n!} \quad \text{and} \quad W^n_{\Pi 2} := 1^n ,$$

and for $k \geq 3$

$$\begin{aligned} W^n_{\Sigma k} &:= \left[W^n_{\Sigma k-1}\, 0\right]^{m_{k,n}} W^n_{\Pi k-1}\, 0 \left[W^n_{\Sigma k-1}\, 0\right]^{m_{k,n}} , \\ W^n_{\Pi k} &:= \left[W^n_{\Sigma k-1}\, 0\right]^{m_{k,n}} W^n_{\Sigma k-1}\, 0 \left[W^n_{\Sigma k-1}\, 0\right]^{m_{k,n}} , \end{aligned}$$

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

From the definition follows easily

**Lemma 13** For $k \geq 2$ and every $n \in \mathcal{F}$

$$\begin{aligned} W^n_{\Sigma k} \in L_{\Sigma k} \quad &\text{and} \quad W^n_{\Sigma k} \notin L_{\Pi k} , \\ W^n_{\Pi k} \in L_{\Pi k} \quad &\text{and} \quad W^n_{\Pi k} \notin L_{\Sigma k} . \end{aligned}$$

Let $k \geq 2$ and $S \in \texttt{SUBLOG}$ be a space bound. We will prove Theorem 8 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 $W^n_{\Sigma k}$, too. Similarly, if a $\Pi_k$ TM $M$ accepts $L_{\Sigma k}$ in space $S$ then for large $n \in \mathcal{F}$ it accepts $W^n_{\Pi k}$ and hence makes a mistake. Recall that $\mathcal{N}_{M,S}$ denotes the constant defined for $M$ and $S$ in Section 2.

**Proposition 1** 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 $i > |U\, W^n_{\Pi k}|$ and
2. $Space_M(\alpha, i,\ U\, W^n_{\Pi k}\, V) \leq S(n)$ and $Space_M(\alpha, \hat{i},\ U\, W^n_{\Sigma k}\, V) \leq S(n)$

holds:

$$\begin{aligned} \mathsf{acc}^k_M(\alpha, i, U\, W^n_{\Pi k}\, V) &\Longrightarrow \mathsf{acc}^k_M(\alpha, \hat{i}, U\, W^n_{\Sigma k}\, V) && \text{if } (\alpha, i) \text{ is existential,} \\ \mathsf{acc}^k_M(\alpha, \hat{i}, U\, W^n_{\Sigma k}\, V) &\Longrightarrow \mathsf{acc}^k_M(\alpha, i, U\, W^n_{\Pi k}\, V) && \text{if } (\alpha, i) \text{ is universal.} \end{aligned}$$

**Lemma 11** For this specific $\mathcal{F}$

$$L_{\mathcal{F}} \in \Pi_2 Space(\text{llog}) \quad \text{and} \quad \overline{L}_{\mathcal{F}} \in \Sigma_2 Space(\text{llog}) \ .$$

**Proof.** We describe llog space-bounded $\Pi_2$TMs $M_\Pi$ and $\Sigma_2$TMs $M_\Sigma$ that recognize the language $L_{\mathcal{F}}$ , resp. the complement of $L_{\mathcal{F}}$ .

The machine $M_\Pi$ works as follows. It checks first whether the input is of the form $1^n$ , for some integer $n > 2$ . Then, to verify the condition $\forall\, \ell \in [3\ \dots\ n-1] \quad F(\ell) < F(n)$ , $M_\Pi$

- deterministically computs $F(n)$ and writes down the binary representation of $F(n)$ on the tape;
- universally guesses an integer $\ell \in [3 \dots n-1]$ ; to this end the machine moves in universal mode the input head from the left to the right and it stops the head $\ell$ positions from the right end of the string $1^n$ ;
- existentially guesses an integer $k \in [1 \dots F(n)-1]$ and then moving the input head to the right, checks deterministically whether $k$ divides $\ell$ . It accepts, if $k$ does not divide $\ell$ .

The machine $M_\Sigma$ also checks at te beginning whether the input is of the form $1^n$ , for some integer $n$ . $M_\Sigma$ accepts if this condition does not hold. Otherwise the machine writes down on the work tape $F(n)$ in binary and tests whether

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

Similarly as for machines $M_\Pi$ 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 $L_{\mathcal{F}}$ and that $M_\Sigma$ recognizes $\overline{L}_{\mathcal{F}}$ .

The proof that these machines have low space complexity is based on the fact that for some constant $c$ , $F(n) < c \log n$ for each positive integer $n$ . Hence, the binary representation of $F(n)$ has length $O(\log\log n)$ . ∎

Thus languages $L_{\mathcal{F}}$ as described in Definition 6 exist. For the base case of the following inductive separation we also need the property that $L_{\mathcal{F}} \notin \Sigma_2 Space(o(\log))$ and symmetrically that $\overline{L}_{\mathcal{F}} \notin \Pi_2 Space(o(\log))$ . This has been shown for the example above explicitely in [14]. Below we will give a general argument showing that this property simply follows from the condition $n \in \mathcal{F}$ and $n + n! \notin \mathcal{F}$ .

## 3.1 ATMs with a Constant Number of Alternations

**Lemma 12** For any $k \geq 2$ holds

$$\begin{aligned} L_{\Sigma k} &\in \Sigma_k Space(\text{llog}) \ , \\ L_{\Pi k} &\in \Pi_k Space(\text{llog}) \ . \end{aligned}$$

The proof of these properties is straightforward using the fact that $L_{\mathcal{F}} \in \Pi_2 Space(\text{llog})$ and $\overline{L}_{\mathcal{F}} \in \Sigma_2 Space(\text{llog})$ . The separation now follows from the following

**Theorem 7** *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 co-$\Pi_k$ for $Z$-bounded languages (Theorem 4) now follows easily.

# 3 Hierarchies

For a subset of the natural numbers $\mathcal{A}$ let $L_{\mathcal{A}}$ be the language over the single letter alphabet $\{1\}$ defined by $1^n \in L_{\mathcal{A}}$ iff $n \in \mathcal{A}$.

**Definition 6** Define $L_2 := \{1\}^+$, and for $k \geq 3$ $L_k := (L_{k-1}\ \{0\})^+$.
Let $\mathcal{F}$ be an infinite subset of the natural numbers with the properties:

- $n \in \mathcal{F} \quad \Longrightarrow \quad n + n! \notin \mathcal{F}$ and
- $L_{\mathcal{F}} \in \Pi_2 Space(\text{llog})$ and $\overline{L}_{\mathcal{F}} \in \Sigma_2 Space(\text{llog})$.

Then we define

$$\begin{aligned} L_{\Pi 2} &:= L_{\mathcal{F}} \text{ and } L_{\Sigma 2} := \{1\}^+ \cap \overline{L}_{\mathcal{F}} , \\ L_{\Sigma k} &:= \{w_1 0 w_2 0 \ldots 0 w_p 0 \mid p \in \mathbb{N},\ w_i \in L_{k-1} \text{ and } \exists\, i \in [1...p] \quad w_i \in L_{\Pi k-1}\} , \\ L_{\Pi k} &:= \{w_1 0 w_2 0 \ldots 0 w_p 0 \mid p \in \mathbb{N},\ w_i \in L_{k-1} \text{ and } \forall\, i \in [1...p] \quad w_i \in L_{\Sigma k-1}\} . \end{aligned}$$

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 $L_{\Pi k} = L_k \cap \overline{L}_{\Sigma k}$). An example for a set $\mathcal{F}$ is the following:

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

where $F(n)$ denotes the smallest positive integer that does not divide $n$. It is easy to show that $\mathcal{F}$ is infinite. The property, $n \in \mathcal{F}$ implies $n + n! \notin \mathcal{F}$, follows from

**Lemma 10** For any integer $n > 2$ holds $n + n! \notin \mathcal{F}$.

**Proof.** This 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 $F(n + n!) \leq F(n)$. ∎

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

On the other hand, a sequence $\mathcal{C}$ 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 $\mathcal{M}_b^2 + 1$ apart and within $\mathcal{C}$ 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 $2(Z(|X|)+1) \cdot \mathcal{M}_{b_t} + 1$ many crossings, or

a4) within the last $2\mathcal{M}_{b_t}^3 + 1$ 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

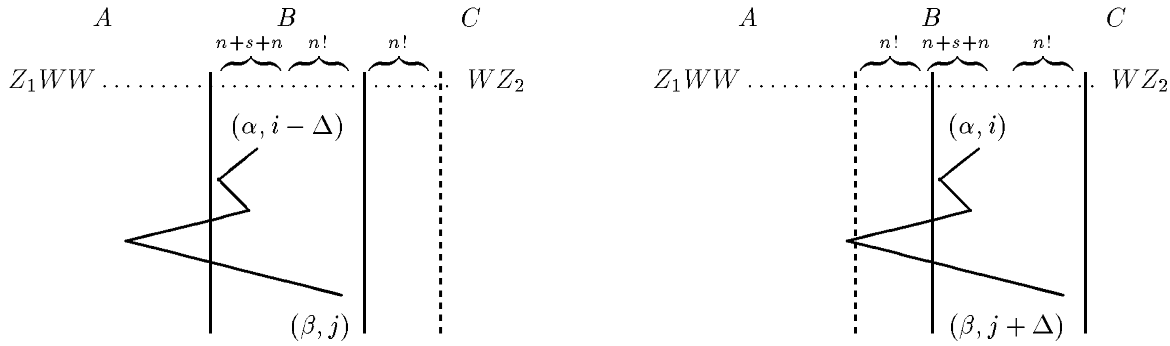
b2) $M$ has just finished a long turn that contains only universal configurations.

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 furtherst 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 $Alter_{M'}(X) \leq Alter_M(X)$ . To see that all computations of $M'$ are finite, first notice that if $M$ does not make progress inifinitely often $M'$ will stop the simulation eventually. Assume that $M'$ does not stop on some path. If $Alter_M(X) < \infty$ this cannot be due to alternations nor to crossings of $M$ since their 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 1 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 $\mathcal{M}_b^2$ 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.




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

Fig. 6

Otherwise the input head is located in $B$, i.e. $|A| < j \leq |AB|$ (see Fig. 6a). By Lemma 7 (2.), one can deduce that $(\alpha, i) \models_{M,X} (\beta, j+\Delta)$, which implies $\mathtt{acc}_M^{k-2}(\beta, j+\Delta, X)$. 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 $\mathtt{acc}_M^{k-2}(\beta, j, X)$, which completes the proof. ∎

## 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.

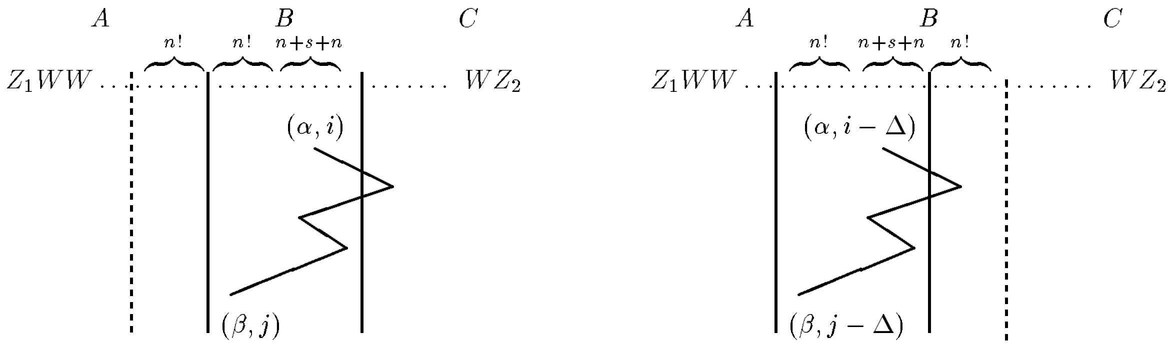
**Lemma 9** *For every $S$-space-bounded ATM $M$ there exists an ATM $M'$, which is also $S$-space-bounded, such that for all $Z$-bounded strings $X$ holds:*

- *$M'$ accepts $X$ iff $M$ accepts $X$,*
- *$Alter_{M'}(X) \leq Alter_M(X)$,*
- *if $Alter_M(X) < \infty$ 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, $\mathcal{M}_b$ denotes the number of memory states of $M$ as defined in Section 2.1.

Let us call by a *crossing* 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 $\mathcal{C} = C_u, C_{u+1}, \ldots, C_v$ of consecutive configurations of a computation path on $X$ is a *long turn* if $\mathcal{C}$ 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 $\mathcal{C}$

- either the input head visits position $i + \mathcal{M}_b^2$, but never moves to the left of $i$,
- or it visits position $i - \mathcal{M}_b^2$, but never moves to the right of $i$,




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

Fig. 5

In this case using property (ii) and Lemma 7 (2.) – for $Z_1' := Z_1\ W^{r-2(n+n!)}, Z_2' := W^{t-2n} Z_2$ , 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

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

and hence $\mathtt{acc}_M^{k-1}(\alpha, i - \Delta, X)$ . 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

$$(\alpha, i - \Delta) \models_{M,X} (\beta, j) \quad \text{implies} \quad \mathtt{acc}_M^{k-2}(\beta, j, X) \ .$$

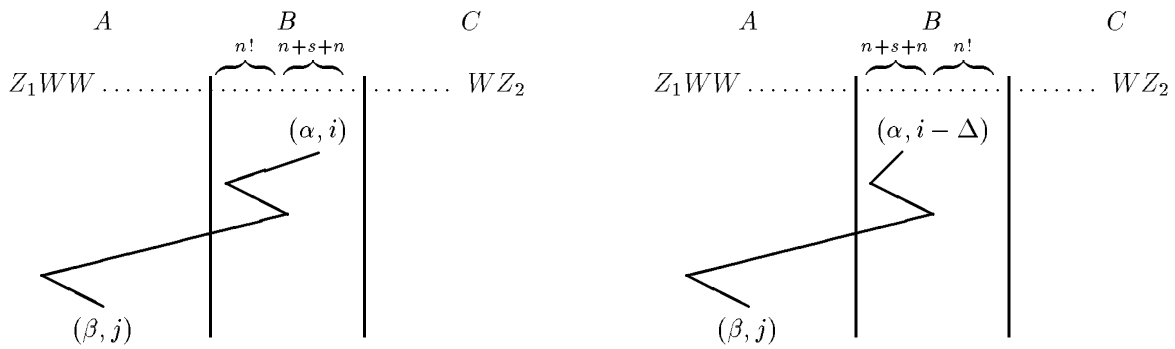
Remember that because of Claim 1 only finite paths have to be considered. Let $(\beta, j)$ be such a configuration. Divide the input $X$ into three regions $A, B, C$ as above. Depending on which region is visited by the input head in configuration $(\beta, j)$ , two cases are considered. If the input head is in region $A$ or $C$ (as in Fig. 4b) then from Lemma 7 (3.) we obtain that $(\alpha, i) \models_{M,X} (\beta, j)$ . $\mathtt{acc}_M^{k-1}(\alpha, i, X)$ thus implies $\mathtt{acc}_M^{k-2}(\beta, j, X)$ .

(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:

$$\begin{aligned} A &:= Z_1 W^{r-(n+n!)} , \\ B &:= W^{n!} W^n W^s W^n , \\ C &:= W^{t-n} Z_2 . \end{aligned}$$

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 $j > |AB|$.




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

Fig. 4

From property (ii) and Lemma 7 (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 $\mathtt{acc}_M^{k-1}(\alpha, i - \Delta, X)$.

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

**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

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

If $|Z_1 W^n| < j \leq |Z_1 W^{r+s+t-(n+n!)}|$ then Lemma 7 (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 $\mathtt{acc}_M^{k-1}(\alpha, i, X)$.

On the other hand, if $j \leq |Z_1 W^n|$ or $j > |Z_1 W^{r+s+t-(n+n!)}|$ then by Lemma 7 (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 $\alpha$ and $\beta$ because by the second assumption $|\alpha| \leq |\beta| \leq S(n)$. 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 $\mathtt{acc}_M^1(\alpha, i, X)$ there exists an accepting $(\beta, j)$ with

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

Using the Configuration-Shift-Lemma one can conclude that

$$\begin{array}{llll} (\alpha, i-\Delta) & \models_{M,X} & (\beta, j-\Delta) & \text{if } |Z_1 W^{n+n!}| < j \leq |Z_1 W^{r+s+t-n}|, \text{ and} \\ (\alpha, i-\Delta) & \models_{M,X} & (\beta, j) & \text{otherwise.} \end{array}$$

Since $\beta$ is accepting $\mathtt{acc}_M^1(\alpha, i-\Delta, X)$ 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 $\mathtt{acc}_M^1(\alpha, i-\Delta, X)$ holds. Let $(\beta, j)$ with $(\alpha, i-\Delta) \models_{M,X} (\beta, j)$ be a final configuration. By Lemma 7

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

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

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

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

Now let $k > 2$ and consider an existential configurations $(\alpha, i)$. 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

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

and

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

1. $(\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, i) \models_{M,X} (\beta, j-\Delta)$,
2. $(\alpha, j) \models_{M,X} (\beta, \ell) \iff (\alpha, j-\Delta) \models_{M,X} (\beta, \ell-\Delta)$,
3. $(\alpha, j) \models_{M,X} (\beta, i) \iff (\alpha, j-\Delta) \models_{M,X} (\beta, i)$.

**Proof.** First note that the conditions on $j$ and $\ell$ guarantee that all positions $j,\ \ell,\ j-\Delta,\ \ell-\Delta$ 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 $\hat{\imath} := i$ if $i \le |Z_1|$, otherwise $\hat{\imath} := i - \Delta$. Using the Pumping Lemma twice – first for the input pair $X, X'$ and then for $X', X''$ – we obtain:

$$(\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, \hat{\imath}) \models_{M,X'} (\beta, j-\Delta) \iff (\alpha, i) \models_{M,X''} (\beta, j-\Delta)$$
$$(\alpha, j) \models_{M,X} (\beta, \ell) \iff (\alpha, j-\Delta) \models_{M,X'} (\beta, \ell-\Delta) \iff (\alpha, j-\Delta) \models_{M,X''} (\beta, \ell-\Delta)$$
$$(\alpha, j) \models_{M,X} (\beta, i) \iff (\alpha, j-\Delta) \models_{M,X'} (\beta, \hat{\imath}) \iff (\alpha, j-\Delta) \models_{M,X''} (\beta, i)$$

The claim of the lemma follows because $X'' = X$. ∎

In the inductive argument for the proof of Theorem 1 (Proposition 1 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 8 (Position Shift)** Let $k \ge 2$, $r, s, t$ be integers with $r, t \ge m_{k,n}$ and $s \ge 1$, and let $Z_1, Z_2, W \in \{0,1\}^*$ be arbitrary strings. Then for $X = Z_1\, W^r\, W^s\, W^t\, Z_2$ and for any configuration $(\alpha, i)$ fulfilling the requirements

1. $|Z_1\, W^r| < i \le |Z_1\, W^r\, W^s|$ and
2. $Space_M(\alpha, i, X) \le S(n)$ and $Space_M(\alpha, i-\Delta, X) \le S(n)$

holds:

$$\mathsf{acc}_M^{k-1}(\alpha, i, X) \iff \mathsf{acc}_M^{k-1}(\alpha, i-\Delta, X) .$$

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

$$\mathsf{acc}_M^{k-1}(\alpha, i, X) \implies \mathsf{acc}_M^{k-1}(\alpha, i-\Delta, X) .$$

A similar argument yields the opposite implication. Let

$$\mathsf{acc}_M^{k-1}(\alpha, i, X) \qquad \text{(i)}$$

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.

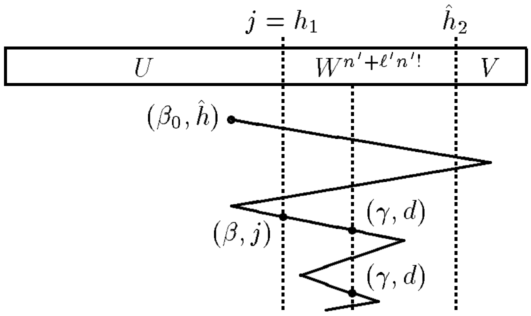



Fig. 3

Let $(\beta, j)$, for $j = h_1$ or $j = \hat{h}_2$, be the last configuration of $\mathcal{C}_1$. Without loos of generality, assume that $j = h_1$. Since $\mathcal{C}_2$ is infinite there exsists $h_1 < d < \hat{h}_2$ and a memory state $\gamma$ such that $(\gamma, d)$ occurs on $\mathcal{C}_2$ 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 1 implies that there exists a computation path $\mathcal{D}$ such that $\mathcal{D}$ starts and ends in $(\gamma, d)$, and that the input head is never moved farther than $\mathcal{M}^2_{S(n)} \cdot |W|$ positions to the left nor to the right of $d$. Let $\mathcal{C}^1_2$ denote the part of $\mathcal{C}_2$ between $(\beta, j)$ and the first $(\gamma, d)$ on $\mathcal{C}_2$. Using Lemma 1 and 2 one can easily construct from $\mathcal{C}^1_2$ a computation path $\mathcal{D}^1$ such that

- $\mathcal{D}^1$ starts in $(\beta, j)$,
- $\mathcal{D}^1$ ends in $(\gamma, d')$, for some $d'$ such that

$$d' < j + \mathcal{M}^2_{S(n)}(\mathcal{M}_{S(n)} + 1) \cdot |W| \quad \text{and} \quad d' \geq \min(d, j + \mathcal{M}^3_{S(n)} \cdot |W|)$$

- the input head is never moved to the left of $j$ nor to the right of

$$j + \mathcal{M}^2_{S(n)}(\mathcal{M}_{S(n)} + 2) \cdot |W| \;\leq\; j + n' \cdot |W|\,.$$

Finally, let $\mathcal{C}'_1$ denote a computation path for input $X$ starting in $(\beta_0, \hat{h})$ and ending in $(\beta, j)$. By Lemma $3'$ such a path exists. $M$ starting in $(\beta_0, \hat{h})$ and making the same sequence of moves as in $\mathcal{C}'_1\mathcal{D}^1\mathcal{D}\mathcal{D}\mathcal{D}...$ makes an infinite universal loop on $X$.

This completes the proof of the first implication of the lemma. Let us now assume that $\mathtt{acc}^2_M(\alpha, \hat{\imath}, Y)$ holds for a universal configuration $(\alpha, i)$. If $\mathtt{acc}^2_M(\alpha, i, X)$ 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 $\mathcal{C}$ of $M$ on $X$ started in $(\beta_0, h)$ is rejecting or along $\mathcal{C}$ $M$ makes at least one alternation. Using the similar methods as above one can show that $\mathtt{acc}^2_M(\alpha, \hat{\imath}, Y)$ does not hold, too – contradiction. ∎

## 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 7 (Configuration Shift)** Let $X = Z_1\ W^{n+n!}\ W^s\ W^n\ Z_2$ 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 $i$ with $i \leq |Z_1|$ or $i > |Z_1\ W^{n+n!}\ W^s\ W^n|$ and any $j, \ell \in [\,|Z_1\ W^{n+n!}| + 1\ \ldots\ |Z_1\ W^{n+n!}\ W^s|\,]$ holds:

Let $\mathcal{C}$ be an infinite universal computation path for input $X'$ that starts in $(\beta_0, \hat{h})$. From $\mathcal{C}$ 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 $hh_2 := h_2 + |W^{\ell' n'!}|$. 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 $\mathcal{C}$ (see the figure below).*

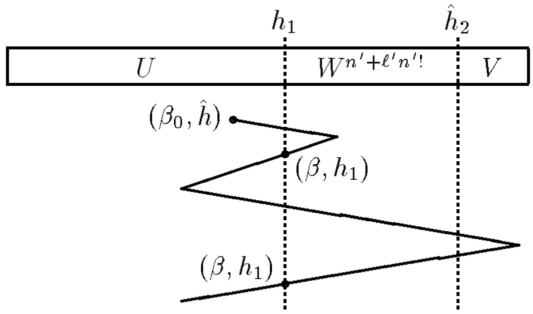



Fig. 1

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 $\mathcal{C}$ at least twice. From Lemma $3'$ one can conclude that

$$
\begin{array}{lll}
(\beta_0, \hat{h}) & \models_{M,X} & (\beta, h_1) \quad \text{and} \\
(\beta, h_1) & \models_{M,X} & (\beta, h_1)\ .
\end{array}
$$

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 $\mathcal{C}$ is similar to this one.

**Case 2:** *There is an initial part $\mathcal{C}_1$ of $\mathcal{C}$ and an infinite rest $\mathcal{C}_2$ of $\mathcal{C}$ such that in $\mathcal{C}_2$ $M$ scans only the input to the left of $h_1$ or to the right of $\hat{h}_2$ (see the figure below).*

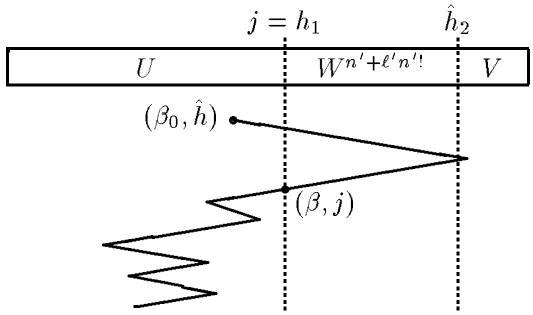



Fig. 2

Let $(\beta, j)$, for $j = h_1$ or $j = \hat{h}_2$, be the last configuration of $\mathcal{C}_1$. From the Lemma $3'$ we have that $(\beta, j)$ is reachable from $(\beta_0, \hat{h})$ on $X$, too. Let $\mathcal{C}'_1$ denote a computation path from $(\beta_0, \hat{h})$ to $(\beta, j)$ for input $X$. Then $\mathcal{C}'_1 \mathcal{C}_2$ is an infinite computation path for $X$.

**Case 3:** *There is an initial part $\mathcal{C}_1$ of $\mathcal{C}$ and an infinite rest $\mathcal{C}_2$ of $\mathcal{C}$ such that in $\mathcal{C}_2$ $M$ scans only the string $W^{n'+\ell' n'!}$ (see the figure below).*

## 2.3 Fooling ATMs by Pumping the Input

**Lemma 6 (1-Alternation)** For any configuration $(\alpha, i)$ with

- $i \leq |Z_1|$ or $i > |Z_1\, W^n|$ and
- $Space_M(\alpha, i, X) \leq S(n)$ and $Space_M(\alpha, \hat{\imath}, Y) \leq S(n)$ holds:

$$\begin{array}{lcll} \mathtt{acc}^2_M(\alpha, i, X) & \Longrightarrow & \mathtt{acc}^2_M(\alpha, \hat{\imath}, Y) & \text{if } (\alpha, i) \text{ is existential, and} \\ \mathtt{acc}^2_M(\alpha, \hat{\imath}, Y) & \Longrightarrow & \mathtt{acc}^2_M(\alpha, i, X) & \text{for universal } (\alpha, i). \end{array}$$

**Proof.** Assume that $(\alpha, i)$ fulfils both conditions above. First, let this configuration be existential and let $\mathtt{acc}^2_M(\alpha, i, X)$ 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) $(\alpha, i) \models_{M,X} (\beta_0, h)$, and

(B) each computation path $\mathcal{C}$ on input $X$ that starts in $(\beta_0, h)$ is finite. In addition, along each such $\mathcal{C}$ $M$ does not alternate, and the final configuration of $\mathcal{C}$ 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 $a := (n'+1)(n'+2)\dots n$ 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 $\mathcal{N}_{M,S} \leq n \leq (n'+1)^2$ from Lemma $3'$ (for $n := n'$ and $m := n$) and by (A) it follows that

$$(\alpha, \hat{\imath}) \models_{M,X'} (\beta_0, \hat{h})$$

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

$$\mathtt{acc}^1_M(\beta_0, \hat{h}, X')$$

holds. Below we prove that this predicate is true.

Assume, to the contrary, that $\mathtt{acc}^1_M(\beta_0, \hat{h}, X')$ 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 $3'$, 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).

position in $[|Z_1W^{n_1}|+1\ldots|Z_1W^{n_1+n_2+\ell' n_2!}|]$ then, applying Lemma $3'$ 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 $(\alpha, \hat{i})$ reaches $(\beta, \hat{j})$ 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 $\mathcal{C}$ between $(\alpha, \hat{i})$ and $(\beta, \hat{j})$ of $k-1$ alternations such that $M$ alternate only on the prefix $Z_1W^{n_1}$ and suffix $W^{n_3}Z_2$. Assume to the contary that

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

where $\ell' := \ell n(n-1)\ldots(\lfloor\sqrt{n}\rfloor+1)$. 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 $W^{n_3} = W^{n_3'}W^{m'}W^{n_3''}$ for some integers $n_3'$ and $n_3''$. Below it is shown that $\mathcal{C}$ can be cut and paste such that in the obtained computation path $M$ does not alternate with the input head over $W^{m+1}$. This yields a contradiction since $m$ was the largest integer of this property.

Let us define the following head position bounds

$$\begin{array}{llll} L_1 := |Z_1W^{n_1}| & R_1 := L_1 + |W^m| + 1 \\ L_2 := |Z_1W^{n_1+m+n_3'}| & R_2 := L_2 + |W^{\lfloor\sqrt{n}\rfloor}| + 1 \ . \end{array}$$

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

$$R_2 + |W^{\lfloor\sqrt{n}\rfloor}| \leq |Z_1W^{n+\ell n!}| \tag{ii}$$

Let $\mathcal{C}'$ be a subsequence of computations of $\mathcal{C}$ which starts and ends with the head position in $\{L_1, R_2\}$. We claim that $\mathcal{C}'$ can be modified to the computation path of the same number of alternations, which starts and ends in the same configurations as $\mathcal{C}'$ and such that $M$ does not alternate with the head positions in $[L_1, R_1 + |W|]$. Only the case when $\mathcal{C}'$ 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 $\mathcal{C}'$ and $(\beta_2, R_2)$ the last one. Moreover let $(\beta_1, R_1)$ be the first configuration in $\mathcal{C}'$ 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 one can show that

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

Moreover by Lemma 2 we have:

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

Therefore for $\delta := c_1c_2|W|$ holds

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

By (ii) $M$ making the same moves as in $\mathcal{C}'$ between $(\beta_1, R_1)$ and $(\alpha_2, L_2)$, reaches $(\alpha_2, L_2 + \delta)$ when started in $(\beta_1, R_1 + \delta)$. Therefore there is a computation path which starts in $(\alpha_1, L_1)$ ends in $(\beta_2, R_2)$ of the same number of alternations as $\mathcal{C}'$ 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. ∎

Let us consider first that $\mathcal{C}'$ is a tail of $\mathcal{C}$. By Lemma 2 there exists a constant $c$, with $1 \leq c \leq \mathcal{M}_b$ 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 $\mathcal{C}'$ when started in $(\alpha_R, R)$ then we obtain a computation for $M$ with the same number of alternations as in $\mathcal{C}'$ but with the head never moving to the right of $i' - c \cdot |W|$.

Assume now that $\mathcal{C}'$ is not a tail of $\mathcal{C}$. Then the last configuration of $\mathcal{C}'$ has a form $(\alpha'_L, L)$, for some memory state $\alpha'_L$. Let $(\alpha'_R, R)$ be the last configuration in $\mathcal{C}'$ with the head position $R$. By Lemma 2 there exist constats $c_1, c_2$, with $1 \leq c_1, c_2 \leq \mathcal{M}_b$ such that $M$ starting in $(\alpha_L, L)$ reaches $(\alpha_R, R - c_1 c_2 \cdot |W|)$ and starting in $(\alpha'_R, R - c_1 c_2 \cdot |W|)$ reaches $(\alpha'_L, L)$. It is obvious that $M$ starting in $(\alpha_R, R - c_1 c_2 \cdot |W|)$ and making the same sequence of moves as between $(\alpha_R, R)$ and $(\alpha'_R, R)$ in $\mathcal{C}'$, reaches $(\alpha'_R, R - c_1 c_2 \cdot |W|)$. Hence we obtain a computation path of the same number of alternations as in $\mathcal{C}'$ that starts and ends also in $(\alpha_L, L)$ and $(\alpha'_L, L)$, resp. but with the head never moving to the right of $i' - c_1 c_2 \cdot |W|$. ∎

Note that by Claim 1 and the assumption that $Alter_M(\alpha, i, X) \leq \exp S(n)$ it follows that if $M$ with $Y$ on the input tape starts in $(\alpha, \hat{\imath})$ and makes $k-1$ alternations with the hed 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 $(\alpha, \hat{\imath})$ makes $k - 1 = \exp S(n) + 1$ alternations such that the head is never moved farther than $\delta_k$ positions from $\hat{\imath}$. By the assumption that $n \geq N_{M,S}$ we conclude:

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

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

**Claim 2** Let $k - 1 \leq \exp S(n)$. Then for any memory state $\beta$ and for $j \in \{|Z_1|, |Z_1 W^n| + 1\}$ holds: $M$ started in $(\alpha, i)$ with $X$ on the input tape reaches $(\beta, j)$ making $k-1$ alternations iff $M$ started in $(\alpha, \hat{\imath})$ with the input $Y$ reaches $(\beta, \hat{\jmath})$ making also $k - 1$ alternations.

**Proof.** We prove the claim for $i = |Z_1|$ and $j = |Z_1 W^n| + 1$. In the other cases a similar proof can be used.

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

$$k \leq \exp S(n) + 1 \leq 2^{\frac{1}{2}\log n - 2} + 1 \leq \lfloor \sqrt{n} \rfloor$$

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

$$n_1 + n_2 + n_3 = n \quad \text{and} \quad \lfloor \sqrt{n} \rfloor \leq n_2 \leq n \tag{i}$$

such that $M$ alternates only on the prefix $Z_1 W^{n_1}$ and on the suffix $W^{n_3} Z_2$, but not on $W^{n_2}$. Hence by Lemma 3′, for $i', j' \in [0 \ldots |Z_1 W^{n_1}|] \cup [|Z_1 W^{n_1+n_2}| + 1 \ldots |X| + 1]$, $m' := n$, $n' := n_2$ and $\ell' := \ell n(n-1) \ldots (n_2 + 1)$

$$(\alpha', i') \models_{M,X} (\beta', j') \qquad \Longleftrightarrow \qquad (\alpha', \hat{\imath}') \models_{M,Y} (\beta', \hat{\jmath}')$$

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 computation path of $k-1$ alternations which starts in $(\alpha, \hat{\imath})$ and ends in $(\beta, \hat{\jmath})$ for the input $Y$.

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 $(\alpha, \hat{\imath})$ and ends in $(\beta, \hat{\jmath})$ and such that $M$ does not alternate with the head

Otherwise, using a similar pumping argument one can show that $M$ on input $X$ can reach a configuration $(\alpha, \bar{j})$, 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 $|\beta| > Space_M(X)$.

∎

**Lemma 5 (Small Alternation Bound)**

$$Space_M(X) \leq S(n) \quad \text{and} \quad Alter_M(X) \leq \exp S(n) \quad \Longrightarrow \quad Alter_M(Y) = Alter_M(X) .$$

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

$$Space_M(\alpha, i, X) \leq S(n)$$

and

$$Alter_M(\alpha, i, X) \leq \exp S(n) .$$

Assume that $k$ is an arbitrary positive integer and let

$$\delta_k := k \cdot \mathcal{M}_b^2 \cdot (\mathcal{M}_b + 1) \cdot |W| ,$$

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

**Claim 1** Let $M$ starting in $(\alpha, \hat{\imath})$ makes $k-1$ alternations with the input head never moved beyond $W^{n+\ell n!}$. Then there exists a computation for $M$ of $k-1$ alternations such that it starts also in $(\alpha, \hat{\imath})$ but the input head is never moved farther than $\delta_k$ positions to the right of $\hat{\imath}$, if $\hat{\imath} = |Z_1|$ and to the left of $\hat{\imath}$, if $\hat{\imath} = |Z_1 W^{n+\ell n!}| + 1$.

**Proof.** We show this claim for $\hat{\imath} = |Z_1|$. A similar proof can be used in the case $\hat{\imath} = |Z_1 W^{n+\ell n!}| + 1$.

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 $M$ started in $(\alpha, \hat{\imath})$ 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

$$i + \mathcal{M}_b^2 \cdot |W| \leq L < R \leq i' \quad \text{and} \quad R - L \geq \mathcal{M}_b^2 \cdot (\mathcal{M}_b + 1) \cdot |W| ,$$

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

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

Below we show how to cut and paste $\mathcal{C}'$ 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 $i' > i + \delta_k$.

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 $\hat{i}$ denotes the corresponding position within $Y$. Thus

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

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

1. $(\alpha, i) \models_{M,X} (\beta, j) \iff (\alpha, \hat{i}) \models_{M,Y} (\beta, \hat{j})$ ,
2. $(\alpha, i) \models^{\star}_{M,X} (\beta, j) \iff (\alpha, \hat{i}) \models^{\star}_{M,Y} (\beta, \hat{j})$ .

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

**Lemma 3′** Let $n$ and $m$ be integers with $\mathcal{N}_{M,S} \leq m \leq (n+1)^2$ and let $\alpha, \beta$ be memory states with $|\alpha| \leq |\beta| \leq S(m)$. Then for any $i, j \in [0 \ldots |Z_1|\,] \cup [\,|Z_1 W^n| + 1 \ldots |X| + 1\,]$ the properties 1. and 2. above hold.

These claims can be proven using the method developed in [8] and the fact that $\mathcal{M}^6_{S(m)} < n$.

## 2.2 Space and Alternation Bounds

**Lemma 4 (Small Space Bound)**

$$Space_M(X) \leq S(n) \implies Space_M(Y) = Space_M(X) \ .$$

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

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

$$(\alpha_0, 0) \models^{\star}_{M,Y} (\alpha, \hat{j}) \models^{\star}_{M,Y} (\beta, \hat{j}') \ .$$

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

$$(\alpha_0, 0) \models^{\star}_{M,X} (\alpha, j) \models^{\star}_{M,X} (\beta, j') \ .$$

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:

$$\begin{array}{llllll} (\alpha, i) & \models_{i,j} & (\alpha_1, h(p_1)) & \models_{i,j} & (\alpha_1, h(q_1)) & \models_{i,j} \\ & & (\alpha_2, h(p_2)) & \models_{i,j} & (\alpha_2, h(q_2)) & \models_{i,j} \\ & & & \dots & & \\ & & (\alpha_t, h(p_t)) & \models_{i,j} & (\alpha_t, h(q_t)) & \models_{i,j} \quad (\beta, j) \ . \end{array}$$

Since there are $t$ pairs $(p_s, q_s)$ and the difference between any pair is at most $\mathcal{M}_b$, by the pigeon hole principle there exists an integer $c \in [1 \dots \mathcal{M}_b]$ and $t/\mathcal{M}_b = \mathcal{M}_b$ pairs $(p_{s_1}, q_{s_1}), (p_{s_2}, q_{s_2}), \dots$ with identical difference $c$, that means $q_{s_\ell} - p_{s_\ell} = c$ for $\ell = 1, 2, \dots, \mathcal{M}_b$. Define $\delta' := c \cdot |W|$.

Let $d$ be an arbitrary integer in $[1 \dots \mathcal{M}_b]$ and define $\alpha'_\ell := \alpha_{s_\ell}$ and $i_\ell := h(p_{s_\ell})$. Then we obtain:

$$\begin{array}{llllll} (\alpha, i) & \models_{i,j} & (\alpha'_1, i_1) & \models_{i,j} & (\alpha'_1, i_1 + \delta') & \models_{i,j} \\ & & (\alpha'_2, i_2) & \models_{i,j} & (\alpha'_2, i_2 + \delta') & \models_{i,j} \\ & & & \dots & & \\ & & (\alpha'_d, i_d) & \models_{i,j} & (\alpha'_d, i_d + \delta') & \models_{i,j} \quad (\alpha'_{d+1}, i_{d+1}) \ := \ (\beta, j) \end{array}$$

The input $X$ contains a sequence of identical blocks $W$ between the positions $i$ and $j$. For any $\ell \in [1 \dots d]$, $M$ starting in $(\alpha'_\ell, i_\ell + \delta')$ reaches $(\alpha'_{\ell+1}, i_{\ell+1})$ without mowing 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

$$\begin{array}{llll} (\alpha, i) & \models_{i,j} & (\alpha'_1, i_1) & \models_{i,j} \\ & & (\alpha'_2, i_2 - \delta') & \models_{i,j} \\ & & \dots & \\ & & (\alpha'_d, i_d - (d-1)\delta') & \models_{i,j} \quad (\beta, j - d\delta') \ , \end{array}$$

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 exsists a computation path that starts in $(\alpha, i + \delta)$ and ends in $(\beta, j)$. ∎

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

$$(\mathcal{M}_{S(n)}^6 + 1)^2 \quad < \quad n$$

and

$$S(n) \quad < \quad \frac{1}{2} \log n \ - 2 \ .$$

**Remark:** In this section all claims following hold for any integer $n \geq \mathcal{N}_{M,S}$.

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 "$n \to n + n!$ 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

$$X \ = \ Z_1 \, W^n \, Z_2 \qquad \text{and} \qquad Y \ = \ Z_1 \, W^{n+\ell n!} \, Z_2 \ ,$$

## 2.1 Inputs of a Periodic Structure

In this section some properties of TM computations for binary inputs of the form $Z_1 WW \ldots W Z_2$ will be described. Let $M$ be an ATM. Then for any integer $b \geq 0$ we define

$$\mathcal{M}_b \quad := \quad \#\{\alpha \mid \alpha \text{ is a memory state of } M \text{ with } |\alpha| \leq b\} \ .$$

The first two results characterize "short" computations i.e. computations restricted to substrings $WW \ldots W$.

**Lemma 1** 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 $\mathcal{M}_b^2 \cdot |W|$ or more positions to the left of $\min(i, j)$ nor to the right of $\max(i, j)$.

This Lemma is a generalization of a result in [16]. It is easy to check that the same counting argument also yields this claim.

**Lemma 2** Let $|i - j| \geq \mathcal{M}_b^2 \, (\mathcal{M}_b + 1) \cdot |W|$. 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 $c \in [1..\mathcal{M}_b]$ such that for all $d \in [1..\mathcal{M}_b]$ there is a computation path satisfying ( ♠ ) which starts in $(\alpha, i)$ and ends in $(\beta, j - d \cdot \operatorname{sgn}(j - i) \cdot c \cdot |W|)$, where $\operatorname{sgn}(z) := z/|z|$.
- Moreover, there also exists a computation path satisfying ( ♠ ) that starts in $(\alpha, i + d \cdot \operatorname{sgn}(j - i) \cdot c \cdot |W|)$ and ends in $(\beta, j)$.

**Proof.** In the folowing we will only discuss the case $i < j$ when considering the computation from configuration $(\alpha, i)$ to $(\beta, j)$. Let $|i - j| \geq \mathcal{M}_b^2(\mathcal{M}_b + 1) \cdot |W|$.

Define for integers $p \geq 0$ the function $h(p) := i + p \cdot |W|$ and let $t := \mathcal{M}_b^2$. Partition integers in $[1 \ldots \mathcal{M}_b^3]$ into the $t$ intervals $[L_1, R_1], [L_2, R_2] \ldots [L_t, R_t]$ of equal length $\mathcal{M}_b$ with boundaries

$$\begin{aligned} L_s &:= (s-1)(\mathcal{M}_b + 1) + 1 \\ R_s &:= L_s + \mathcal{M}_b \ . \end{aligned}$$

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 configurations there must exist a pair with positions $p_s < q_s \in [L_s, R_s]$ and identical memory states $\alpha_s$.

## 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. The input word is stored on the input tape between end-markers $\$$.

**Definition 4** 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 locaton 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 partioned 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 5** Let

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

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

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

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).

$$\mathtt{acc}^k_M(\alpha, i, X)$$

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 $Alter_M(X) \; := \; Alter_M(\alpha_0, 0, X)$.

**Definition 3** 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. $DSpace(S)$ denotes the class of languages accepted by $S$ space-bounded DTMs and $weakDSpace(S)$ 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 $weakDSpace(\text{llog})$ contains languages that do not belong to $co\text{-}weakDSpace(o(\log))$.

In [6] 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 $L_{\neq}$ by ATMs proves the following

**Theorem 5** $weakDSpace(\text{llog}) \setminus ASpace(o(\log)) \neq \emptyset$.

As consequences one obtains

**Corollary 3** *For any* $k \geq 1$ *and each* $S \in \texttt{SUBLOG}$

$$\begin{array}{rcll} \Sigma_k Space(S) & \subset & weak\Sigma_k Space(S) & \text{and} \\ \Pi_k Space(S) & \subset & weak\Pi_k Space(S). & \end{array}$$

**Corollary 4** *For each* $S \in \texttt{SUBLOG}$

$$ASpace(S) \quad \subset \quad weakASpace(S) .$$

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 6** *Let* $L$ *be a nonregular deterministic context-free, a strictly nonregular language, or a nonregular context-free bounded language, then* $L \notin \bigcup_{k \in \mathbb{N}} \Sigma_k Space(o(\log))$. *Furthermore, for ATMs without any bound on the number of alternations it is not possible that* $L$ *and* $\overline{L}$ *both belong to* $ASpace(o(\log))$.

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].

Note that the class of functions that are approximable from below in space $S \in \texttt{SUBLOG}$ is quite large. For example, it contains the bounds $\lfloor \log \rfloor$, $\lfloor \log^{1/2} \rfloor$, $\lfloor \mathrm{llog} \rfloor$ and $\lfloor \log^* \rfloor$. Thus one obtains:

1. $\Sigma_{\mathrm{llog}}\, Space(\mathrm{llog}) \quad \subset \quad \Sigma_{(\mathrm{llog})+1} Space(\mathrm{llog})$.
2. $\Sigma_S Space(S) \quad \subset \quad \Sigma_{S+1} Space(S) \quad$ for $S = \log^{1/2-\epsilon}$, where $0 < \epsilon < 1/2$.
3. $\bigcup_{k \in \mathbb{N}} \Sigma_k Space(S) \quad \subset \quad \Sigma_{\log^*} Space(S) \qquad$ for $S \in \Omega(\mathrm{llog}) \cap o(\log / \log^*)$.

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

**Theorem 3** *For any $S \in \texttt{SUBLOG}$ and all $k > 1$ $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ are not closed under complementation and concatenation. Moreover, $\Sigma_k Space(S)$ is not closed under intersection and $\Pi_k Space(S)$ 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, which says that sublogarithmic $\Sigma_k Space$ and $\Pi_k Space$ 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 $\Pi_k Space(S) = co\text{-}\Sigma_k Space(S)$ 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 sublogarithimic space bounded deterministic TMs [19]: For bounded languages it can be shown that there exist equivalent ATMs that always halt. This implies

**Theorem 4** *Let $S \in \texttt{SUBLOG}$ 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:*

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

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

$$L_{\neq} \ := \ \{1^n 0 1^m \mid n \neq m\}$$

does not belong to $ASpace(o(\log))$. It is interesting to note that this language – but not its complement – can be recognized even by a deterministic machine in *weak space* llog. Furthermore, a probabilistic machine can recognize this language with arbitrary small error even in constant space [7].

holds for all levels $k$ . The base case is the existence of a language that separates $\Pi_2 Space(\text{llog})$ from $\Sigma_2 Space(o(\log))$ . Its complement separates $\Sigma_2 Space(\text{llog})$ from $\Pi_2 Space(o(\log))$ .

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 $n$ space, but not by any $\Pi_k$ TM that 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 explicitely 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** For all $k > 1$ holds

$$\begin{aligned} \Sigma_k Space(\text{llog}) \setminus \Pi_k Space(o(\log)) &\neq \emptyset \qquad \text{and} \\ \Pi_k Space(\text{llog}) \setminus \Sigma_k Space(o(\log)) &\neq \emptyset. \end{aligned}$$

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 $S \in \texttt{SUBLOG}$ . 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** For any $S \in \texttt{SUBLOG}$ and all $k \geq 1$ holds

$$\begin{aligned} \Sigma_k Space(S) &\subset \Sigma_{k+1} Space(S) \ , \\ \Pi_k Space(S) &\subset \Pi_{k+1} Space(S) \ . \end{aligned}$$

Independently the existence of this strict hierarchy has been shown by von Braunmühl with coauthors [5]. 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 2** A function $A : \mathbb{N} \rightarrow \mathbb{N}$ 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 $n \in \mathbb{N}$ .

**Theorem 2** For any pair of functions $S \in \texttt{SUBLOG}$ and $A > 1$ with $A \cdot S \in o(\log)$ , where $A$ is approximable from below in space $S$ , holds:

$$\begin{aligned} \Sigma_A Space(S) \setminus \Pi_A Space(S) &\neq \emptyset \ , \\ \Pi_A Space(S) \setminus \Sigma_A Space(S) &\neq \emptyset \ . \end{aligned}$$

**Corollary 2** For any $S$ and $A$ as in the theorem above holds:

$$\begin{aligned} \Sigma_A Space(S) &\subset \Sigma_{A+1} Space(S) \ , \\ \Pi_A Space(S) &\subset \Pi_{A+1} Space(S) \ . \end{aligned}$$

# 1 Introduction

It is well known that if a deterministic or nondeterministic TM uses less than llog space then the machine can recognize only regular languages, and that there exist non-regular languages in $DSpace(\text{llog})$. Therefore, let $\texttt{SUBLOG} := \Omega(\text{llog}) \cap o(\log)$ denote the set of all nontrivial sublogarithmic space bounds, where llog abbreviates the twice iterated logarithmic function $n \mapsto \log\log n$. 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 llog and log. One of the most exciting problem of this type is whether the closure under complement for NTM

$$NSpace(S) \;=\; co\text{-}NSpace(S)$$

shown by Immerman and Szelépcsenyi [13],[22] remains valid for sublogarithmic space bounds. If this equality were not valid for a function $S \in \texttt{SUBLOG}$ then obviously $DSpace(S) \subset NSpace(S)$ [1].

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

**Definition 1** Let $Z : \mathbb{N} \to \mathbb{N}$ be a function. 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 $NSpace(S) = co\text{-}NSpace(S)$ 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 $\Sigma_k Space(S)$ is not closed under complementation for any $S \in \texttt{SUBLOG}$ and all $k > 1$.

Recall that for $k \geq 1$ the class $\Sigma_k Space(S)$ 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. $\Pi_k Space(S)$ denotes the set of languages accepted by the same kind of machines, except that they start in a universal state. By definition $\Sigma_1 Space(S) = NSpace(S)$. We will also consider ATMs with a non-constant bound $A$ for the number of alternations. In this case, the notation $\Sigma_A Space(S)$ and $\Pi_A Space(S)$ is used.

By standard techniques it follows from Immerman-Szelépcsenyi's theorem that for $S \in \Omega(\log)$, and for all $k \geq 1$

$$\Sigma_1 Space(S) = \Sigma_k Space(S) = \Pi_k Space(S).$$

Note that these techniques do not work for sublogarithmic space bounds. Recently, Chang et al. ([6]) have shown that there is a language in $\Pi_2 Space(\text{llog})$ that does not belong to $NSpace(o(\log))$. Clearly, this proves that for space bounds $S$ between llog and log, the alternating $S$ space hierarchy does not collapse to the first level and that $\Sigma_1 Space(S) \subset \Pi_2 Space(S)$. 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 develope techniques to investigate properties of sublogarithmic computations and then generalize them to an inductive proof that the separation of the $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ classes

[1] In [25, p.419] it is incorrectly cited that $DSpace(S) \subset NSpace(S)$, for $S \in \texttt{SUBLOG}$, thus the problem if $DSpace(S) = NSpace(S)$ is still open for any $S \in \Omega(\text{llog})$ (see Remark 6.1 in [17]).

# 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 llog denote the logarithmic function log iterated twice, and $\Sigma_k Space(S)$ , $\Pi_k Space(S)$ 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

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

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

$$\Sigma_1 Space(S) \subset \Sigma_2 Space(S) \subset \Sigma_3 Space(S) \subset \ldots$$
$$\subset \Sigma_k Space(S) \subset \Sigma_{k+1} Space(S) \subset \ldots$$

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 $S \in \Omega(\text{llog}) \cap o(\log)$ and all $k > 1$ $\Sigma_k Space(S)$ and $\Pi_k Space(S)$ are not closed under complementation and concatenation. Moreover, $\Sigma_k Space(S)$ is not closed under intersection, and $\Pi_k Space(S)$ is not closed under union.

It is an interesting open question whether for sublogarithmic bounds $S$ the property that $\Pi_k Space(S)$ is the complement of $\Sigma_k Space(S)$ is fulfilled. This is a nontrivial problem since there is no obvious way how to detect infinite computation paths. Here, we generalize Sipser's result on halting space-bound computations for sublogarithimic space bounded deterministic TMs [19] to ATMs that recognize bounded languages. For the class of $Z$-bounded languages with $Z \leq \exp S$ we obtain the equality

$$co\text{-}\Sigma_k Space(S) = \Pi_k Space(S) .$$

We also consider the space requirement for the recognition of nonregular context-free languages. Alt, Geffert and Mehlhorn have recently shown a logarithmic lower bound for nondeterministic TMs [2]. We improve this result obtaining the same lower bound for ATMs. Thus this last result shows that even alternations do not increase the power of sublogarithmic machines substantially.

Finally, we investigate the power of weak vs. strong sublogarithmic space bounded machines.

# The Sublogarithmic Space World

Maciej Liśkiewicz*

Rüdiger Reischuk†

TR-93-048

August 1993

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

**AMS(MOS) subject classifications.** 68Q05, 68Q10, 68Q25, 68Q45

---

*Institut für Theoretische Informatik, Technical University Darmstadt,
Alexanderstraße 10, 64283 Darmstadt, Germany
e-mail: liskiewi@iti.informatik.th-darmstadt.de
Research partially supported by the Alexander-von-Humboldt-Stiftung
†Institut für Theoretische Informatik, Technical University Darmstadt,
Alexanderstraße 10, 64283 Darmstadt, Germany
e-mail: reischuk@iti.informatik.th-darmstadt.de
Part of this work was done during a visit at the ICSI in Berkeley