RAIRO. INFORMATIQUE THÉORIQUE ET APPLICATIONS

VILIAM GEFFERT

A hierarchy that does not collapse : alternations in low level space

RAIRO. Informatique théorique et applications, tome 28, n° 5 (1994), p. 465-512

<http://www.numdam.org/item?id=ITA_1994__28_5_465_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 28, n° 5, 1994, p. 465 à 512)

A HIERARCHY THAT DOES NOT COLLAPSE: ALTERNATIONS IN LOW LEVEL SPACE (*)

by Viliam GEFFERT (1)

Abstract. – *The alternation hierarchy of $s(n)$ space bounded machines does not collapse for $s(n)$ below* $\log(n)$. *That is, for each $s(n)$ between* $\log\log(n)$ *and* $\log(n)$ *and each* $k \geqq 2$, Σ_{k-1}-SPACE $(s(n))$ *and* Π_{k-1}-SPACE $(s(n))$ *are proper subsets of* Σ_k-SPACE $(s(n))$ *and also of* Π_k-SPACE $(s(n))$. *Moreover,* Σ_k-SPACE $(s(n))$ *is not closed under complement and intersection, similarly,* Π_k-SPACE $(s(n))$ *is not closed under complement and union.*

Résumé. – *La hiérarchie de machines bornées en espace par $s(n)$ ne s'écroule pas en dessous de* $\log(n)$. *Plus précisément, pour tout $s(n)$ compris entre* $\log\log(n)$ *et* $\log(n)$ *et pour tout* $k \geqq 2$, Σ_{k-1}-SPACE $(s(n))$ *et* Π_{k-1}-SPACE $(s(n))$ *sont des sous-ensembles propres de* Σ_k-SPACE $(s(n))$ *et* Π_k-SPACE $(s(n))$. *De plus,* Σ_k-SPACE $(s(n))$ *n'est pas fermé par complément ni par intersection, et de façon similaire,* Π_k-SPACE $(s(n))$ *n'est pas fermé par complément ni par union.*

1. INTRODUCTION

In the structural complexity theory, many hierarchies have been studied and various relations between them have been established. However, direct proofs showing collapsing or noncollapsing hierarchies are very rare.

For example, the strong exponential time hierarchy is finite, as has been shown in [12], and so is the hierarchy of interactive proof systems [1]. Infinite hierarchies are even more rare. Most of the known results concern classes relativized by oracles ([11, 2, 25]), giving both finite and infinite hierarchies.

During the last few years, very important results have been achieved for the alternation hierarchy of space-bounded computations. First, some space bounded hierarchies were shown to be finite ([15, 24, 19]). These results were then superseded by the result of Immerman and Szelepcsényi showing that the

(*) Received February, 11, 1993, revised December 13, 1993, accepted December 22, 1993.
(1) Department of Computer Science, P. J. Šafárik University, Jesenná 5, 04154 Košice, Slovakia.

nondeterministic space is closed under complement ([13, 22]). This implies that the alternation hierarchy of $s(n)$ space-bounded machines collapses to NSPACE$(s(n))$ = Σ_1-SPACE$(s(n))$, *i.e.*,

$$\Sigma_k\text{-SPACE}(s(n)) = \Pi_k\text{-SPACE}(s(n)) = \Sigma_1\text{-SPACE}(s(n)),$$

for each $k \geq 1$ and each $s(n) \geq \log(n)$. Taking this fact into consideration, the question of whether there is an infinite hierarchy for sublogarithmic space bounds naturally arises.

The first sign indicating that the alternation hierarchy behaves radically different for space below $\log(n)$ was the proof [6] that Σ_1-SPACE$(s(n)) \subsetneqq$ Π_2-SPACE$(s(n))$, for each $s(n)$ between $\log\log(n)$ and $\log(n)$. This result was then slightly improved in [23] by showing that $s(n)$ can be bounded from below by any unbounded fully space constructible function $l(n)$. There exist sublogarithmic, unbounded, and fully space constructible functions, but they are necessarily nonmonotone and hence the corresponding space complexity classes do not contain DSPACE$(\log\log(n))$ ([7, 20, 8]).

The next step was the separation of the first three levels of this hierarchy [9], *i.e.*, Σ_1-SPACE$(s(n)) \subsetneqq \Sigma_2$-SPACE$(s(n)) \subsetneqq \Sigma_3$-SPACE$(s(n))$, symmetrically, Π_1-SPACE$(s(n)) \subsetneqq \Pi_2$-SPACE$(s(n)) \subsetneqq \Pi_3$-SPACE$(s(n))$, for space bounds between $\log\log(n)$ and $\log(n)$. Then the third and fourth levels were separated [16], *i.e.*, Σ_3-SPACE$(s(n)) \subsetneqq \Sigma_4$-SPACE$(s(n))$ and Π_3-SPACE$(s(n)) \subsetneqq \Pi_4$-SPACE$(s(n))$. Finally, it has been shown that the hierarchy does not collapse below the level five [3]; Σ_4-SPACE$(s(n)) \subsetneqq$ Σ_5-SPACE$(s(n))$. Figure 1 summarizes the known results, arrows indicate the proper inclusions.

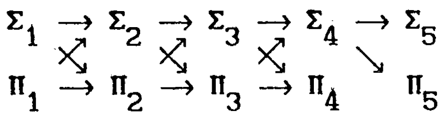

Figure 1

We shall show that the alternation hierarchy of space bounded machines is infinite, namely, that for each $s(n) \geq \log\log(n)$ with $\sup_{n\to\infty} s(n)/\log(n) =$ 0, and each $k \geq 2$, we have

$$\Sigma_{k-1}\text{-SPACE}(s(n)) \subsetneqq \Sigma_k\text{-SPACE}(s(n)),$$
$$\Pi_{k-1}\text{-SPACE}(s(n)) \subsetneqq \Pi_k\text{-SPACE}(s(n)),$$

$$\Sigma_{k-1}\text{-SPACE}(s(n)) \subsetneq \Pi_k\text{-SPACE}(s(n)),$$
$$\Pi_{k-1}\text{-SPACE}(s(n)) \subsetneq \Sigma_k\text{-SPACE}(s(n)).$$

Moreover, Σ_k-SPACE $(s(n))$ and Π_k-SPACE $(s(n))$ are incomparable, *i.e.*,

$$\Sigma_k\text{-SPACE}(s(n)) - \Pi_k\text{-SPACE}(s(n)) \neq \emptyset,$$
$$\Pi_k\text{-SPACE}(s(n)) - \Sigma_k\text{-SPACE}(s(n)) \neq \emptyset.$$

Finally, we show that Σ_k-SPACE $(s(n))$ is not closed under complement and intersection, and that Π_k-SPACE $(s(n))$ is not closed under complement and union. Since machines using less than $\log\log(n)$ space can recognize the regular languages only ([21, 14]), this settles the alternation space hierarchy problem;

– the hierarchy collapses to Σ_1-SPACE $(s(n))$ for the superlogarithmic case,

– the hierarchy is infinite for space bounds between $\log\log(n)$ and $\log(n)$ [this paper],

– the hierarchy collapses to the deterministic constant space for space bounds below $\log\log(n)$.

The open problems of this hierarchy are the exact relations among Σ_0-SPACE $(s(n))$ = DSPACE $(s(n))$, Σ_1-SPACE $(s(n))$ = NSPACE $(s(n))$, and Π_1-SPACE $(s(n))$.

The paper is organized as follows: we begin in Section 2 by giving some basic definitions and lemmas that will be used later.

Section 3 discusses the so-called $n \to n + n!$ method which was used first in [21] to show that the deterministic machines using less than $\log(n)$ space cannot distinguish between inputs 1^n and $1^{n+n!}$. This method has been extended to the nondeterministic case [8]. We shall now generalize this method simultaneously in two directions: first, it can be applied not only to the tally inputs, but also to some binary inputs having a periodic structure. Second, it can be used, in a modified form, for the Σ_k/Π_k-alternating machines as well.

The key observation of the Section 4 is the fact that the computation trees of the alternating machines can be viewed as if they were the trees describing an evaluation order of operators in the ordinary boolean formulas, and hence put into the conjunctive/disjunctive normal forms.

Section 5 brings another new proof technique – the notion of Σ_k/Π_k-SPACE $(s(n))$ resistant strings and languages. Roughly speaking, a pair of strings w_1, w_2 is Σ_k/Π_k-SPACE$(s(n))$ resistant if no machine can use any Σ_k/Π_k-alternating $s(n)$ space bounded machine as its oracle to distinguish between the substrings w_1 and w_2 on the input tape. We shall show that having given languages with Σ_k/Π_k-SPACE$(s(n))$ resistant words, we can design languages that are Σ_{k+1}/Π_{k+1}-SPACE$(s(n))$ resistant.

Section 6 gives an induction base for this anti-oracle mechanism by exhibiting some Σ_2/Π_2-SPACE$(s(n))$ resistant languages, for each $s(n)$ between $\log\log(n)$ and $\log(n)$. Then the infinite space hierarchy is established and some closure properties under boolean operations are shown.

Update

The existence of the infinite hierarchy have been proved independently by two other groups of authors, namely, by M. Liśkiewicz and R. Reischuk [17], and also by B. von Braunmühl, R. Gengler, and R. Rettinger [4], so now there exist three independent solutions. The proof in [17] is also based on the $n \to n + n!$ method, using a different argument, but the witness languages are very similar. The proof in [4] is completely different, its argument holds for the weakly space bounded machines as well, but requires witness languages having a much higher information content.

2. PRELIMINARIES

We shall consider the standard Turing machine having a finite control, a two-way read-only input tape with the input enclosed in two endmarkers, and a separate semi-infinite two-way read-write worktape, initially empty.

The reader is assumed to be familiar with the notion of alternating Turing machine, which is at the same time a generalization of nondeterminism and parallelism. *See* [5] for a more exact definition and properties of alternating machines. We shall now introduce this notion in a slightly different way.

Definition 1: A memory state of a Turing machine is an ordered triple $q = \langle r, u, j \rangle$, where r is a state of the machine's finite control, u is a string of worktape symbols written on the worktape (not including the left endmarker or blank symbols), and j is a position of the worktape head.

A configuration is an ordered pair $p = \langle q, i \rangle$, where q is a memory state and i is a position of the input tape head.

The size of a memory state $q = \langle r, u, j \rangle$ is the length of the worktape space used, *i.e.*, $|u|$. We shall denote it by $/q/$. The size of a configuration $p = \langle q, i \rangle$ is, by definition, $/p/ = /q/$. The size of the initial configuration $p_I = \langle q_I, 0 \rangle = \langle \langle r_I, \varepsilon, 0 \rangle, 0 \rangle$ is zero.

We may assume, without loss of generality, that the machine is not allowed to write the blank symbol on the worktape or reduce the size of its memory state. Therefore, if a configuration p_2 can be reached from p_1 by some computation path, then $/p_2/ \geq /p_1/$.

We also assume that the machine making a constant number of alternations has its set of finite control states divided into pairwise disjoint sets Σ_k, Π_{k-1}, Σ_{k-2}, Π_{k-3}, ... (for Σ_k-alternating machines, $k \in \mathbb{N}$) or Π_k, Σ_{k-1}, Π_{k-2}, Σ_{k-3}, ... (for Π_k-alternating machines) such that if the machine can get, by a single computation step, from a finite control state $r \in \Sigma_l$ to r', then $r' \in \Sigma_l$ or $r' \in \Pi_{l-1}$. Similarly, for $r \in \Pi_l$ we have $r' \in \Pi_l$ or $r' \in \Sigma_{l-1}$.

The finite control states in $\Sigma = \bigcup_l \Sigma_l$ are called existential, those in $\Pi = \bigcup_l \Pi_l$ are universal. Each memory state or configuration inherits the type of the finite control state included.

An alternation is a computation step changing the finite control state $r \in \Sigma_l$ to $r' \in \Pi_{l-1}$, or $r \in \Pi_l$ to $r' \in \Sigma_{l-1}$. Clearly, a computation path beginning in any Σ_k/Π_k-configuration can make at most $k - 1$ alternations, for each $k \geq 1$.

Definition 2: *a)* A configuration p is *Σ_l-accepting*, if it is of type Σ_l and there exists an alternation-free computation path from p to p' such that

(i) either p' is a halting configuration that accepts the input,

(ii) or the machine enters a Π_{l-1}-accepting configuration in the next computation step from p'.

b) A configuration p is *Π_l-accepting*, if it is of type Π_l and each alternation-free path from p

(i) either halts and accepts the input,

(ii) or enters a Σ_{l-1}-accepting configuration.

The rejection is a little more complicated, since infinite cycles must also be considered:

c) p is Σ_l*-rejecting*, if it is a Σ_l-configuration and all alternation-free paths from p are

(i) either halted in configurations that reject the input,

(ii) entering Π_{l-1}-rejecting configurations,

(iii) or executing infinite cycles.

d) p is Π_l*-rejecting*, if it is a Π_l-configuration having an alternation-free path from p that

(i) either halts and rejects the input,

(ii) enters a Σ_{l-1}-rejecting configuration,

(iii) or executes an infinite cycle.

By definition, a Σ_k/Π_k-machine accepts the input if the initial configuration is determined to be Σ_k/Π_k-accepting, respectively.

DEFINITION 3: Let A be an alternating Turing machine and w be its input. We define $\text{Space}_A(w)$ as the size of the maximal configuration that is reachable by A from the initial configuration $p_I = \langle q_I, 0 \rangle$ on the input w (enclosed in the endmarkers "»" and "«"). The machine A is $s(n)$ *space bounded*, if for each input w

$$\text{Space}_A(w) \leq s(|w|). \tag{1}$$

The classes of languages recognizable by alternating $O(s(n))$ space bounded machines making at most $k-1$ alternations, with the initial finite control state existential or universal, will be denoted by Σ_k-SPACE$(s(n))$ or Π_k-SPACE$(s(n))$, respectively.

It is not too difficult to show that, for each machine A, there exists a constant c such that the number of different memory states not using more than S space on the worktape can be bounded by

$$\begin{array}{ll} \text{number of memory states} & \\ \text{of size at most } S & \leq c^S, \\ & c \geq 6, \end{array} \tag{2}$$

for each $S \geq 1$. The condition $c \geq 6$ is technical, it will be used later. It is easy to bound c by any fixed constant from below. (This condition is used to bound some polynomials of c^S by a fixed power of c^S, e.g., we shall need $((c^S)^2+1)+(c^S+1)+(c^S)^2 < (c^S)^3$, for each $S \geq 1$.)

Before passing further, we shall put the machine A into the following normal form:

LEMMA 1: *For each* $s(n)$ *space bounded* Σ_k/Π_k*-alternating Turing machine* A, *there exists an equivalent* Σ_k/Π_k*-SPACE*$(s(n))$ *machine* A' *such that for each input* w, *each* $i = 0, \ldots, |w| + 1$, *and each* $h = 0, \ldots, \text{Space}_A(w)$, *there exists a configuration* p *having used exactly* h *space on the worktape with the input head position equal to* i *that is reachable from the initial configuration on* w.

Proof: We can replace the original machine A by a new machine A' that simulates A but that, each time A is going to extend the worktape space (by rewriting the leftmost blank on the worktape by a nonblank symbol), A' performs the following actions:

a) If A is in an existential configuration, then A', branching existentially, decides whether

*a*1) to carry on the simulation of A,

*a*2) or to move the input head to the left endmarker. Each time the input head is moved one position to the left, A' branches existentially again and

*a*2.1) either moves more to the left (go to a2),

*a*2.2) or extends the worktape space, and then halts and rejects the input.

*a*3) The third computation branch does the same as the second (*a*2), but the input head is moved to the right endmarker.

b) The same actions are taken if A is in universal configuration, but all branches are universal, and the space extension in b2.2 (*cf.* *a*2.2) is terminated by accepting the input.

It is easy to see that for each $i = 0, \ldots, |w| + 1$ and each $h = 0, \ldots, \text{Space}_A(w)$ there exists a configuration $p = \langle q, i \rangle$ of size $/p/ = h$ that is reachable from the initial p_I. The machine A' has more computation paths than does the original machine A, but "new" computation paths have been added so that they cannot affect the accept/reject status of the computation tree, and hence both A and A' recognize the same language. Note that neither the number of alternations nor the space used have been changed. □

DEFINITION 4: Let $S \geqq 0$ and let p be a configuration with the input head positioned on a substring w of input $\alpha\, w\, \beta$, or going to enter w in the next computation step. p *is* S*-bounded on* w, if no computation path beginning in p uses more that S worktape space before it leaves w by crossing its left/right margin for the first time. (But the space used can exceed S once the left/right margin of w has been crossed.)

Clearly, if a configuration p' is reachable from p by a path never leaving w and p is S-bounded on w, then p' is also S-bounded on w.

DEFINITION 5: Let $S \geqq 0$. Strings w_1 *and* w_2 *are S-equivalent* for a machine A, if A has a computation path from the configuration $\langle q_A, i_A \rangle$ entering w_1 to $\langle q_B, i_B \rangle$ leaving w_1 on the input $\alpha w_1 \beta$, for $i_A, i_B \in \{ |\alpha|, |\alpha|+|w_1|+1 \}$, if and only if A has a path from $\langle q_A, i'_A \rangle$ entering w_2 to $\langle q_B, i'_B \rangle$ leaving w_2 on $\alpha w_2 \beta$, for $i'_A, i'_B \in \{ |\alpha|, |\alpha| + |w_2| + 1 \}$, respectively. (The margins of w_1 and w_2 are crossed only in the first and last computation steps). This holds for any q_A, q_B such that $/q_A/ \leq /q_B/ \leq S$, and each α, β.

LEMMA 2: *Let $\alpha w_1 \beta$ and $\alpha w_2 \beta$ be some inputs for a machine A such that w_1, w_2 are S-equivalent for some $S \geq 0$. Then A can get from a configuration p to p' on the input $\alpha w_1 \beta$ if and only if A can get from p to p' on $\alpha w_2 \beta$, for any p, p' satisfying $/p/ \leq /p'/ \leq S$, with the input head positioned on α or β. (Since w_1, w_2 may be of different lengths, the input head positions of p and p' are relative here, to the left margins of α or β.)*

Proof: The argument is a straightforward induction on the number of times the input head crosses the margins of w_1 and w_2 on inputs $\alpha w_1 \beta$ and $\alpha w_2 \beta$, respectively, using the fact that no configuration can use more than S space along the path from p to p'. Paths from p to p' may be different inside w_1, w_2, but they are equal outside w_1, w_2. (See *fig.* 2.) □

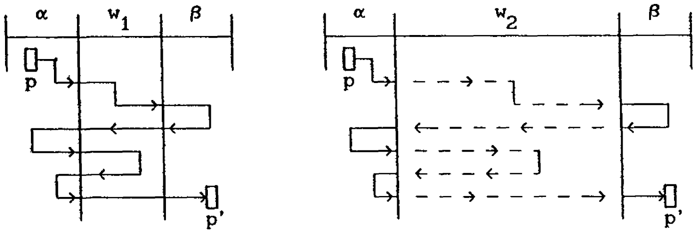

Figure 2

LEMMA 3: *If w_1, w_2 are S-equivalent, then $\alpha_0 w_1 \alpha_1 w_1 \alpha_2 \ldots \alpha_{n-1} w_1 \alpha_n$ and $\alpha_0 w_2 \alpha_1 w_2 \alpha_2 \ldots \alpha_{n-1} w_2 \alpha_n$ are S-equivalent, for any $\alpha_0, \alpha_1, \ldots, \alpha_n$.*

Proof: As a special case of Lemma 2, for p entering $\alpha w_1 \beta / \alpha w_2 \beta$ and p' leaving $\alpha w_1 \beta / \alpha w_2 \beta$, we get that if w_1, w_2 are S-equivalent, then so are $\alpha w_1 \beta$ and $\alpha w_2 \beta$. The rest of the argument is a straightforward induction on n. □

Individual computation paths not using more than S space cannot distinguish S-equivalent w_1, w_2 for inputs $\alpha\, w_1\, \beta$ and $\alpha\, w_2\, \beta$. But beware; even within S space, an alternating machine may reject $\alpha\, w_1\, \beta$ but accept $\alpha\, w_2\, \beta$. This can be achieved by a "cooperation" of several computation paths. Consider the situation shown by Figure 3. Symbols "&" and "∨" represent universal and existential decisions, respectively. The sets of configurations reachable on the margins of w_1 and w_2 are the same. But if p_1, p_2 are Π_l-rejecting and p_3, p_4 Π_l-accepting configurations, then p is Π_{l+2}-rejecting on $\alpha\, w_1\, \beta$ but Π_{l+2}-accepting on $\alpha\, w_2\, \beta$.

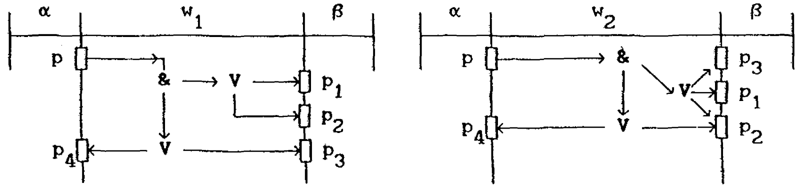

Figure 3

Therefore, the S-equivalence is "weak", it does not guarantee the equal acceptance. However, it does guarantee the equal amount of worktape space used.

LEMMA 4: *Let w_1, w_2 be S-equivalent for some $S \geq 0$. Then*

a) a configuration p is S-bounded on $\alpha\, w_1\, \beta$ if and only if it is S-bounded on $\alpha\, w_2\, \beta$, for any α, β, and each p with the input head positioned on α or β. (The input head positions are relative to the left margins of α or β.)

b) p is S-bounded on w_1 if and only if it is S-bounded on w_2, for each p that is going to enter w_1, w_2 by crossing their left/right margins in the next computation step.

Proof: *a)* We shall show that if the configuration p is S-bounded on $\alpha\, w_1\, \beta$ then it is S-bounded on $\alpha\, w_2\, \beta$. The converse is also true, by a very similar argument.

Suppose that p is S-bounded on $\alpha\, w_1\, \beta$, but not S-bounded on $\alpha\, w_2\, \beta$. Then the machine must enter a configuration using more than S space on w_2, since the segments of computation paths taking place on α or β are exactly the same for $\alpha\, w_1\, \beta$ and $\alpha\, w_2\, \beta$, unless the space used exceeds S. Therefore, there exists a configuration p' on w_2, reachable from p by a path never leaving $\alpha\, w_2\, \beta$, such that the machine is going to extend the worktape space from S to $S + 1$ in the next step.

Before doing so, by Lemma 1, our machine in the normal form decides whether to carry on or to move the input head to the left/right endmarker. That is, we have computation paths that move the input head outside w_2 and then extend the worktape space, *i.e.*, we have a configuration p'' of size S, with the input head positioned on α (or β), reachable from p on $\alpha\, w_2\, \beta$, that is going to use space $S+1$ in the next step. By Lemma 2, p'' is also reachable from p on $\alpha\, w_1\, \beta$. But this is a contradiction, since p is S-bounded on $\alpha\, w_1\, \beta$.

b) The argument for (*b*) is a special case of (*a*), with $\alpha = \beta = \varepsilon$, for paths that enter w_1, w_2 by the first computation steps. Here we analyze configurations reachable from p that are leaving w_1, w_2 by crossing their margins. □

DEFINITION 6: Let p be a configuration with the input head positioned on a substring w of input $\alpha\, w\, \beta$, or going to enter w in the next step. We define $Ex_{p,\,w}$, *the exit set of w for p*, as the set of all configurations reachable from p on w that are leaving w by crossing its margins.

LEMMA 5: *a*) *If a configuration p' is reachable from p by a path never leaving w, then* $\mathrm{Ex}_{p',\,w} \subseteq \mathrm{Ex}_{p,\,w}$.

b) *If a configuration p is S-bounded on w, then $/p''/ \leq S$ for each* $p'' \in \mathrm{Ex}_{p,\,w}$.

c) *If a configuration p is S-bounded on S-equivalent strings w_1 and w_2, then* $\mathrm{Ex}_{p,\,w_1} = \mathrm{Ex}_{p,\,w_2}$, *for each p going to enter w_1, w_2 in the next step. (By Lemma 4b, it is sufficient to suppose that p is S-bounded on w_1 or w_2.)*

The following technical lemma shows an important property of sublogarithmic functions. This lemma will be used later.

LEMMA 6: *For each function $s(n)$ satisfying* $\lim_{n\to\infty} s(n)/\log(n) = 0$, *each* $c \geq 6$, *and each* $H \geq 1$, *there exists* $\check{n} \geq 2$ *such that*

$$\left(c^{s\,(n^H)}\right)^6 < \sqrt{n} \leq \lceil \sqrt{n}\, \rceil < \frac{n}{2} < n-1 < n, \quad \textit{for each } n \geq \check{n}.$$

Proof: If $\lim_{n\to\infty} s(n)/\log(n) = 0$, then for each $\varepsilon > 0$ there exists $\check{n} \geq 2$ such that $s(n)/\log(n) < \varepsilon$, for each $n \geq \check{n}$. Among others, $n^H \geq n \geq \check{n}$, if $H \geq 1$ and $n \geq \check{n}$. Hence, for each $H \geq 1$ and each $\varepsilon > 0$, we have $\check{n} \geq 2$ such that $s(n^H)/\log(n^H) < \varepsilon$, for each $n \geq \check{n}$. But

$\varepsilon = 1/2.H.6. \log(c) > 0$, for $H \geq 1$ and $c \geq 6$. Thus, for each $c \geq 6$ and each $H \geq 1$, we have $\check{n} \geq 2$ such that

$$\frac{s(n^H)}{\log(n^H)} < \frac{1}{2.\mathrm{H}.6.\ \log(c)}, \quad \text{and hence also}$$

$$\left(c^{s(n^H)}\right)^6 < \sqrt{n},$$

for each $n \geq \check{n}$. Since $\sqrt{n} \leq \lceil \sqrt{n} \rceil < \frac{n}{2} < n - 1 < n$, for each $n \geq 7$, this completes the proof of the lemma. □

The condition $\lim_{n \to \infty} s(n)/\log(n) = 0$ is equal to $\sup_{n \to \infty} s(n)/\log(n) = 0$, for each $s(n) : \mathbb{N} \to \mathbb{N}$.

3. THE $N \to N + N!$ METHOD

It was shown in [8] that a nondeterministic machine using less than $\log(n)$ space cannot distinguish between inputs 1^n and $1^{n+in!}$, for each $i \geq 0$. In this section, we shall extend this result from the tally inputs to binary inputs with a periodic structure. Namely, Lemma 3, Lemma 4, and Theorems 1/2 in [8] are actually special cases of Lemma 7, Lemma 8, and Theorem 1 of this section, respectively. Simultaneously, the "$n \to n + n!$" method will be generalized to the alternating machines with a constant number of alterations.

LEMMA 7: *Let $S \geq 1$ and $d \geq 1$. Then, for each input of the form w^m such that $|w| = d$ and $m > (c^S)^6 = M^6$, we have that if there exists a computation path from a configuration $p_1 = \langle q_1, i \rangle$ to $p_2 = \langle q_2, i \rangle$, $/p_1/ \leq /p_2/ \leq S$, such that the input head never visits the right (left) margin of w^m, then the shortest computation path from p_1 to p_2 never moves the input head farther than $M^2.d = (c^S)^2.d$ positions to the right (left, respectively) of i.*

That is, each S space bounded computation path beginning and ending at the same input position has a "short-cut" not wider than M^2 blocks of w.

Proof: The argument is very similar to the proof of Lemma 3 in [8] but, instead of all input tape positions on a tally input, we shall rather consider memory states at block boundaries between adjacent w' s.

Suppose that the furthest configuration along the computation path from p_1 to p_2 is $p_F = \langle q_F, h \rangle$, with $h - i > M^2.d$. Let q_j be the last memory state along the path from p_1 to p_F such that the input head was at the left margin of the j-th block w to the left of the position i, for $j = 1, \ldots, M^2 + 1$, and let t_j be the first memory state along the path from p_F to p_2 with the input head back at the left margin of the same w-block. (See *fig.* 4, where the q_j's and t_j's have been represented by rectangles.)

Since, by (2), there are at most M^2 different pairs of memory states not using more than S space, there must be at least one pair of memory states in the sequence $(q_1, t_1), (q_2, t_2), \ldots, (q_{M^2+1}, t_{M^2+1})$ which is repeated. Thus we have $j' < j''$ such that $(q_{j'}, t_{j'}) = (q_{j''}, t_{j''}) = (q, t)$.

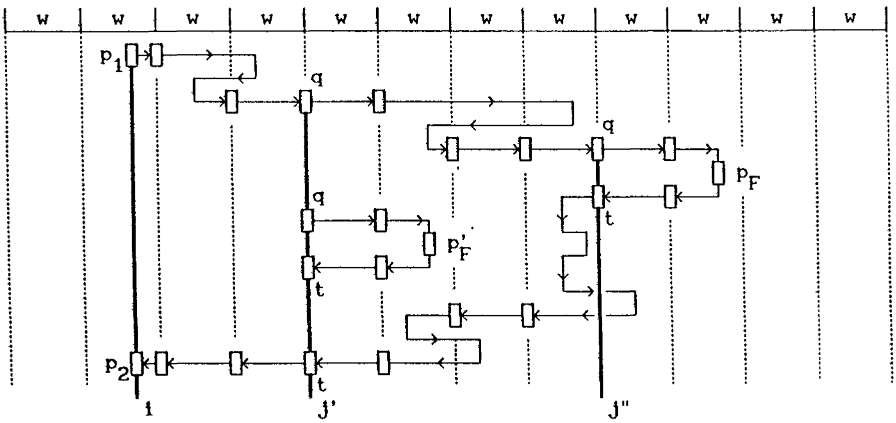

Figure 4

But then we can remove the computation paths from $q_{j'}$ to $q_{j''}$ and from $t_{j''}$ to $t_{j'}$, and we have again a valid computation path from p_1 to p_2. The path from $q_{j''}$ to $t_{j''}$ via p_F is shifted more to the left, by an integer multiple of $d = |w|$. This is possible even for w over the binary tape alphabet, because w^m has the periodic structure and the input head scans identical symbols on the tape positions that are equal modulo $d = |w|$. This process can be repeated until we obtain the shortest computation path from p_1 to p_2.

The argument holds not only for nondeterministic machines, but also for the alternating machines. However, it is possible that the configuration $p_F = \langle q_F, h \rangle$ is Σ_l/Π_l-accepting while $p'_F = \langle q_F, h - d.(j'' - j') \rangle$ is Σ_l/Π_l-rejecting, or vice versa. For example, there may exist another computation path from p_F that is not reachable from p'_F. $\square$

The next lemma shows that each computation path on the periodic input w^m is independent from block positions, *i.e.*, it can be "moved" freely along the input tape, by integer multiples of the block length $|w|$, provided that it does not consume more than S space and begins/ends at least $M^2 + 1 = (c^S)^2 + 1$ w-blocks away from either margin.

LEMMA 8: *Let $S \geq 1$, $d \geq 1$, and let w^m be an input for the machine A such that $|w| = d$ and $m > (c^S)^6 = M^6$. Then, if there exists a computation path from a configuration $\langle q_1, i \rangle$ to $\langle q_2, i + h \rangle$, $/q_1/ \leq /q_2/ \leq S$, such that the input head never visits either of the margins, there exists a path from $\langle q_1, j \rangle$ to $\langle q_2, j + h \rangle$, for each j satisfying*

$$(M^2 + 1).d \leq j \leq (m - (M^2 + 1)).d + 1,$$
$$(M^2 + 1).d \leq j + h \leq (m - (M^2 + 1)).d + 1,$$
$$j \bmod d = i \bmod d.$$

Proof: The argument is obvious; since, by Lemma 7 (see *fig.* 5), the shortest path from $\langle q_1, i \rangle$ to $\langle q_2, i + h \rangle$ never moves the input head more than M^2 w-blocks to the left of i, nor M^2 w-blocks to the right of $i + h$. Such computation paths can be moved along the input tape by integer multiples of $d = |w|$. □

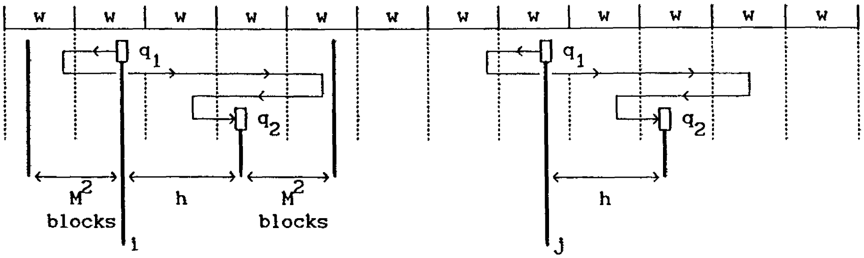

Figure 5

In the same spirit, we can generalize Theorems 1 and 2 in [8] from tally inputs to the periodic binary inputs, *i.e.*, traversals on the strings w^m and $w^{m+im!}$ begin and end in the same memory states, for each sufficiently large m and each $i \geq 0$.

THEOREM 1: *Let $S \geq 1$, $d \geq 1$, $i \geq 0$, and let w^m, $w^{m+im!}$ be inputs for the machine A such that $|w| = d$ and $m > (c^S)^6 = M^6$. Then, for any memory states q_1, q_2 satisfying $/q_1/ \leq /q_2/ \leq S$, the machine A has a computation path from the configuration $\langle q_1, 0 \rangle$ to $\langle q_2, m.d + 1 \rangle$ on the input w^m if and only if A has a path from $\langle q_1, 0 \rangle$ to $\langle q_2, (m + im!).d + 1 \rangle$*

on the input $w^{m+im!}$. *(The margins of* w^m *and* $w^{m+im!}$ *are crossed only in the first and last computation steps.) A similar statement can be formulated for traversals from right to left.*

Proof: $(m \to m + im!)$ By (2), the number of different memory states using at most S space is bounded by $c^S = M \leq M^6 < m$, and hence our machine A, traversing the input w^m from left to right, must enter some memory state twice crossing the boundaries between adjacent w-blocks. That is, A executes a loop that traverses h w-blocks, *i.e.*, of length $h.d$, for some $h \leq M < m$. This loop can be iterated $F = i \,.\, \prod_{j=1, j \neq h}^{m} j$ more times, which gives a valid path traversing the input $w^{m+im!}$, since $h.d.F = i.m!\,.\,d$.

$(m + im! \to m)$ The converse is not so simple since A is far from repeating regularly any loop it gets in. Still, using the Lemma 8, one can show that, for each computation path traversing the input $w^{m+im!}$, with $m > M^6$, A has a path that begins and ends in the same configurations at the margins of $w^{m+im!}$ and that iterates regularly a "short" loop, of length $h \leq M < m$ w-blocks, such that the portions of the input tape traversed before and after this iteration are also "short", of lengths at most M^4 w-blocks. But then this loop is iterated at least $F = i \,.\, \prod_{j=1, j \neq h}^{m} j$ times on the input $w^{m+im!}$, if $m > M^6 = (c^S)^6$ and $c \geq 6$. Cutting the first F iterations of this loop out of the computation path, we shall get a valid computation traversing the input w^m.

For a more detailed proof, the reader is referred to Theorems 1 and 2 in [8]. The only difference is that here we do not consider all input tape positions on tally inputs, but rather positions at block boundaries on the periodic binary inputs. The fact that our machine is not nondeterministic but alternating does not play an important role in the above considerations, we simply ignore the acceptance status of the whole computation tree and concentrate on reachability along a single computation path only. □

As a direct consequence of Lemma 7 and Theorem 1, we obtain:

LEMMA 9: *Let* $S \geq 1$, $d \geq 1$, *and* $i \geq 0$. *Then the words* w^m *and* $w^{m+im!}$ *are* S*-equivalent, for each* $m > (c^S)^6 = M^6$ *and* $|w| = d$.

Proof: First, by Theorem 1, a configuration p_2 leaving w^m to the right is reachable from p_1 entering w^m from the left, $/p_1/ \leq /p_2/ \leq S$, if and only if the corresponding traversal is possible on the input $w^{m+im!}$, for each $i \geq 0$ and each $m > M^6$. The same holds, by symmetry, for traversals from right to left.

Second, for each computation path from p_1 to p_2 not using more than S space, beginning and ending at the left margin of $w^{m+im!}$, and never crossing its right margin, there exists, by Lemma 7, a path from p_1 to p_2 that never moves the input head farther than M^2 w-blocks to the right from the left margin. Since $M^2 \leq M^6 < m$, we have enough room to run this computation on both $w^{m+im!}$ and w^m. The same holds for computations that begin and end at the right margins of w^m and $w^{m+im!}$. □

The strings w^m and $w^{m+im!}$, for $m > (c^S)^6$, have some important properties. By Lemma 3, $\alpha\, w^m\, \beta$ and $\alpha\, w^{m+im!}\, \beta$ are also S-equivalent, for any α and β. Moreover, by Lemma 4, no machine tries to use more than S space on $\alpha\, w^{m+im!}\, \beta$ unless it tries to do so on $\alpha\, w^m\, \beta$. The next theorem shows that configurations having their input head positions exactly $m!$ w-blocks apart and sufficiently far from either margin must have an equal acceptance status on the input $w^{m+im!}$.

THEOREM 2: *Let $S \geq 1$, $d \geq 1$, $i \geq 1$, and let $w^{m+im!}$ be an input for the machine A such that $|w| = d$ and $m > (c^S)^6 = M^6$. The alternating machine A has an accepting computation tree with the root in a configuration $p_l = \langle q_l, j \rangle$ if and only if A has an accepting tree with the root in $p'_l = \langle q_l, j + m!.\, d \rangle$, for any Σ_l/Π_l-configurations p_l, p'_l that are S-bounded on $w^{m+im!}$, and each j satisfying*

$$(m+l.(m+m!)).\,d \leq j \leq j+m!.\,d \leq (m+i.m!-(m+l.(m+m!))).d+1.$$

(*I.e.*, p_l, p'_l of alternating level Σ_l/Π_l are at least $m+l.(m+m!)$ w-blocks away from either margin. This is possible, for example, if $i \geqq 4l + 3$.)

Proof: The argument uses induction on the alternating level l. Because no computation path beginning in p_l or p'_l uses more than S space before reaching the left/right margin of $w^{m+im!}$, we can use Theorem 1 and Lemmas 7, 8, and 9.

First, suppose that the configuration $p_l = \langle q_l, j \rangle$ is Π_l-rejecting. We shall show that then so is $p'_l = \langle q_l, j + m!.d \rangle$. If p_l is Π_l-rejecting, then at least one computation path beginning in p_l must reject the input. We have the following cases to consider:

1) The rejecting computation path alternates, *i.e.*, it enters a Σ_{l-1}-rejecting configuration $p_{l-1} = \langle q_{l-1}, h \rangle$. There are now the following subcases:

1*a*) The rejecting path alternates not moving the input head farther than $m + m!$ w-blocks away from the position j, and hence $|h - j| \leq (m + m!).d$. Since both $p_l = \langle q_l, j \rangle$ and $p'_l = \langle q_l, j + m!.d \rangle$ are at least $m + l.(m + m!)$ w-blocks away from either margin of $w^{m+im!}$, configurations $p_{l-1} = \langle q_{l-1}, h \rangle$ and $p'_{l-1} = \langle q_{l-1}, h + m!.d \rangle$ are at least $m + (l - 1).(m + m!)$ w-blocks away from the margins. (See *fig.* 6.)

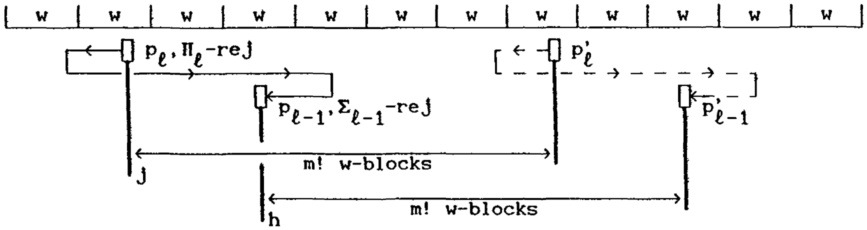

Figure 6

Further, by Lemma 8, if p_{l-1} is reachable from p_l then p'_{l-1} is reachable from p'_l, because positions j, $j + m!.d$, h, and $h + m!.d$ are all at least $m > M^2 + 1$ w-blocks away from either margin, for each $l \geq 1$. Now, using the induction hypothesis for $l' = l - 1$, we have that if the configuration p_{l-1} is Σ_{l-1}-rejecting, then p'_{l-1} is also Σ_{l-1}-rejecting. But then p'_l must be Π_l-rejecting, because it has a computation path that enters a Σ_{l-1}-rejecting configuration.

1*b*) The rejecting computation path moves the input head farther than $m + m!$ w-blocks away from j.

(i) Suppose that the rejecting path gets too far to the left. Let $p_B = \langle q_B, j - m.d \rangle$ be the first configuration with the input head positioned m w-blocks to the left of j, and let $p_A = \langle q_A, j \rangle$ be the last configuration along the path from p_l to p_B with the input head position equal to j. (See *fig.* 7.)

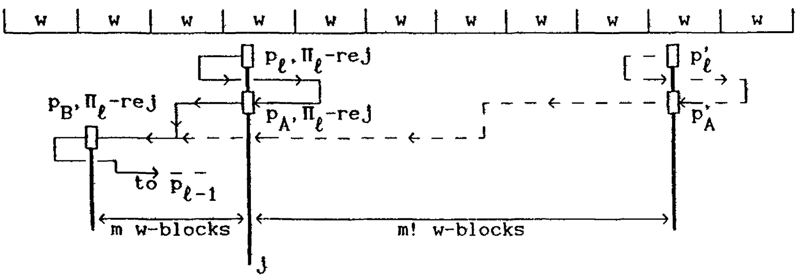

Figure 7

Clearly, p_A and p_B are Π_l-rejecting. By Lemma 8, $p'_A = \langle q_A, j + m!\,.\,d \rangle$ is reachable from p'_l, since both j and $j + m!\,.\,d$ are at least $m > M^2 + 1$ w-blocks away from either margin. Moreover, p_B is reachable from p'_A because, by Theorem 1 and Lemma 9, the machine has a path traversing the tape segment w^m if and only if it has a corresponding path traversing $w^{m+m!}$. But then p'_l is also Π_l-rejecting, since some paths from p_l and p'_l enter the same Σ_{l-1}-rejecting configuration p_{l-1}.

(ii) The same holds if the rejecting path from p_l gets too far to the right; now some paths from p_l and p'_l share a common configuration p_B lying $m + m!$ w-blocks to the right of j.

2) Suppose that p_l is Π_l-rejecting because some computation path enters an infinite cycle, making no alternation at all. By a reasoning very similar to Case 1, we can show that (*a*) either the entire cycle is executed between the positions $j - (m + m!)\,.\,d$ and $j + (m + m!)\,.\,d$ and then p'_l has a parallel path with the same infinite cycle at the distance $m!$ w-blocks apart; (*b*) or at least a part of the infinite cycle lies farther then $m + m!$ w-blocks away from j. But then p_l and p'_l share the common cycle. In both cases, p'_l is Π_l-rejecting.

3) Finally, some alternation-free path beginning in p_l may halt in a configuration that rejects the input. Again, either this path does not move the head "too far" and then we have a parallel path for p'_l, or else some paths from p_l and p'_l share the same halting and rejecting configuration.

Thus, we have shown that if $p_l = \langle q_l, j \rangle$ is Π_l-rejecting then so is $p'_l = \langle q_l, j + m!\,.\,d \rangle$. It is not too hard to see that if p'_l is Π_l-rejecting then, by symmetry, p_l must also be Π_l-rejecting. Therefore, p_l is Π_l-accepting if and only if p'_l is Π_l-accepting.

By a very similar reasoning, we can show that p_l is Σ_l-accepting if and only if p'_l is Σ_l-accepting. The main difference is that, instead of rejecting computation paths beginning in Π_l-rejecting configurations, we analyze *accepting* paths beginning in Σ_l-accepting configurations. Further, Case 2 need not be considered, since no accepting path beginning in the Σ_l-accepting p_l or p'_l can be an alternation-free infinite cycle.

To complete the proof, we have to show that the induction hypothesis holds for $l = 1$, *i.e.*, for Σ_1/Π_1-configurations. However, the structure of the proof for $l = 1$ is exactly the same as for $l > 1$, with Case 1 eliminated (no more alternations ahead). Note that Case 1*a* was the only place where the induction hypothesis was required. □

4. LOGIC BEHIND ALTERNATION

In this section we introduce the notion of a characteristic boolean function $f_{p,w}$ for a configuration p positioned on a substring w of input $\alpha\, w\, \beta$, which allows us to investigate the machine's behavior inside w and outside w separately. This notion is based on the fact that the computation trees of alternating machines can be viewed as if they were the trees representing ordinary boolean formulas composed of AND and OR operators only. Then we shall present some properties of such functions.

DEFINITION 7: Let p be a configuration with the input head positioned on a substring w of input $\alpha\, w\, \beta$, or going to enter w in the next computation step. *A characteristic function* $f_{p,w}$ is a boolean function that is obtained as follows: Take the computation tree the branches of which represent all possible computations beginning in the configuration p. (All input head positions are relative to the left margin of w.)

(i) Then each branch of the tree is pruned as soon as it reaches a configuration t that is leaving w by crossing its left/right margin. The leaf node now corresponding to t is then assigned a boolean variable x_t.

(ii) Each branch that represents an infinite cycle never leaving w is pruned as soon as it enters the same configuration for the second time. The resulting leaf node is then assigned a boolean constant 0 (FALSE).

(iii) Each leaf node that represents a halting configuration reachable from the root p by a path never leaving w is assigned a boolean constant 0 or 1 (FALSE or TRUE), depending on whether it rejects or accepts the input, respectively.

(iv) Each internal node representing an existential/universal configuration is assigned a boolean operator "$\vee$"/"&" (OR/AND), respectively. An internal node having exactly one son is ignored, *i.e.*, it is assigned a unary operator of identity.

(v) If p is S-bounded on w, for some $S \geq 1$, then the resulting tree is finite and represents the evaluation order of operators for the boolean function $f_{p,w}(x_{t_1}, \ldots, x_{t_h})$, with the formal parameter list $x_{t_1}, \ldots, x_{t_h}$ corresponding to $\mathrm{Ex}_{p,w} = \{ t_1, \ldots, t_h \}$, the set of all configurations reachable from p on w that are leaving w by crossing its margins ("exits" of w for p).

Figure 8 presents an example of the tree-to-function transformation that is described above:

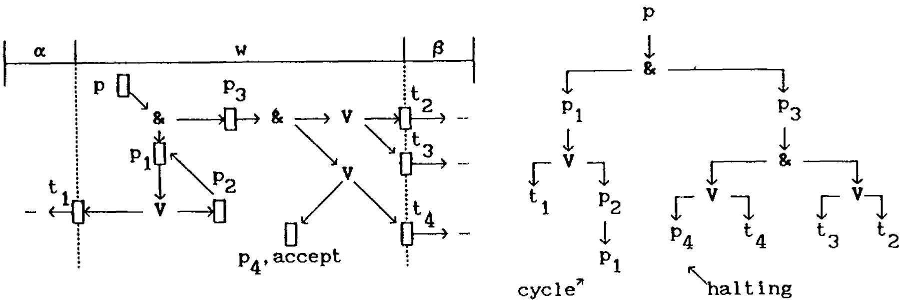

Figure 8

For the configurations p, we obtain

$$f_{p,w}(x_{t_1}, x_{t_2}, x_{t_3}, x_{t_4}) = (x_{t_1} \, V \, 0) \,\&\, ((1 \, V \, x_{t_4}) \,\&\, (x_{t_3} \, V \, x_{t_2}))$$
$$= x_{t_1} \,\&\, (x_{t_2} \, V \, x_{t_3}),$$

which reflects the fact that p is an accepting configuration (*i.e.*, p has an accepting tree on $\alpha w \beta$) if (i) t_1 is accepting and (ii) at least one of t_2, t_3 is accepting. The acceptation status of p on w depends on the set of exit configurations, *i.e.*, on $\mathrm{Ex}_{p,w} = \{t_1, t_2, t_3, t_4\}$. Note that the result is actually independent from $t_4 \in \mathrm{Ex}_{p,w}$ because the accept/reject status of t_4 is overridden by another computation path.

If $f_{p,w}(x_{t_1}, \ldots, x_{t_h})$ is independent from each configuration $t_j \in \mathrm{Ex}_{p,w}$ then it is a constant function returning always the same boolean value. Similarly, if no computation path beginning in p leaves w, then $f_{p,w}$ is a constant function with the empty parameter list, *i.e.*, with $h = 0$. Such functions will be denoted by $f_{p,w}(\,)$.

It is easy to see that boolean functions composed by OR/AND operators only (no NOT's) can be put into the conjunctive/disjunctive normal forms so that no clause contains a negated variable, *i.e.*, for the conjunctive normal form we have either $f(x_1, \ldots, x_h) = $ constant 0/1, or

$$f(x_1, \ldots, x_h) = K_1 \,\&\, K_2 \,\&\, \ldots \,\&\, K_f,$$

such that each of the clauses $K_1, \ldots, K_f$ is of the form

$$K_j = (x_{e_1} \vee x_{e_2} \vee \ldots \vee x_{e_g}),$$

where $A_j = \{x_{e_1}, \ldots, x_{e_g}\} \subseteq \{x_1, \ldots, x_h\}$. Similarly, for the disjunctive normal form, we get either a constant or

$$f(x_1, \ldots, x_h) = K_1 \vee K_2 \vee \ldots \vee K_f,$$

where

$$K_j = (x_{e_1} \,\&\, x_{e_2} \,\&\, \dots \,\&\, x_{e_g}),$$

for each j. These normal forms are obtained by the use of the distributive rules and some other simple transformations (like, for example, $1 \,\&\, \alpha \Rightarrow \alpha$, $0 \,\&\, \alpha \Rightarrow 0$, ...).

DEFINITION 8: Let $C' = (C'_1, \dots, C'_h)$, $C'' = (C''_1, \dots, C''_h)$, $h \geq 0$, be boolean vectors. We write $C' \leq C''$, if $C'_j \leq C''_j$ for each $j \in \{1, \dots, h\}$, $C' \neg \leq C''$, if $C'_j > C''_j$ for some $j \in \{1, \dots, h\}$. (As is usual, $0 < 1$.) A boolean function $f(x_1, \dots, x_h)$ *is monotone*, if $f(C') \leq f(C'')$ for each $C' \leq C''$. We write $f' \leq f''$ for two boolean functions $f'(x_1, \dots, x_h)$ and $f''(x_1, \dots, x_h)$, if $f'(C) \leq f''(C)$ for each C.

It is easy to see that each characteristic boolean function $f_{p,w}$ is monotone, since the operators AND, OR are monotone and the monotone compositions of monotone functions must also be monotone. We shall now present some properties of monotone functions that will be used later.

LEMMA 10: *Let $f'(x_1, \dots, x_h)$ and $f''(x_1, \dots, x_h)$ be monotone functions.*

a) If $f' \leq f''$ and $f'(C') > f''(C'')$, for some C', C'', then $C' \neg \leq C''$.

b) If for each C', C'' we have that $f'(C') > f''(C'')$ implies $C' \neg \leq C''$, then $f' \leq f''$.

The next two lemmas show that the conjunctive/disjunctive normal forms of the monotone functions f' and f'' are closely related, if $f' \leq f''$.

LEMMA 11: *Let $f'(x_1, \dots, x_h)$ and $f''(x_1, \dots, x_h)$ be monotone functions, $f' \leq f''$. If*

$$f''(x_1, \dots, x_h) = K''_1 \,\&\, \dots \,\&\, K''_{\mathfrak{f}''}$$
$$f'(x_1, \dots, x_h) = K'_1 \,\&\, \dots \,\&\, K'_{\mathfrak{f}'}$$

are the conjunctive normal forms for f'', f', then for each clause of f'' there exists a clause of f' composed of a subset of its variables only, i.e., for each $j'' \in \{1, \dots, \mathfrak{f}''\}$ there exists $j' \in \{1, \dots, \mathfrak{f}'\}$ such that

$$K''_{j''} = (x_{e''_1} \vee \dots \vee x_{e''_{\mathfrak{g}''}}),$$
$$K'_{j'} = (x_{e'_1} \vee \dots \vee x_{e'_{\mathfrak{g}'}}),$$

with

$$A'_{j'} = \{x_{e'_1}, \dots, x_{e'_{\mathfrak{g}'}}\} \subseteq A''_{j''} = \{x_{e''_1}, \dots, x_{e''_{\mathfrak{g}''}}\}.$$

The proof is a straightforward contradiction. Supposing that f'' has a clause $K''_{j''}$ such that each clause of f' contains a variable outside $A''_{j''}$, we can easily find $\check{C}$ satisfying $1 = f'(\check{C}) > f''(\check{C}) = 0$. By a very similar argument, we can show a corresponding property for the disjunctive normal forms.

LEMMA 12: *Let $f'(x_1, \ldots, x_h)$ and $f''(x_1, \ldots, x_h)$ be monotone functions, $f' \leq f''$. If*

$$f''(x_1, \ldots, x_h) = K''_1 \vee \ldots \vee K''_{\mathfrak{f}''}$$
$$f'(x_1, \ldots, x_h) = K'_1 \vee \ldots \vee K'_{\mathfrak{f}'}$$

are the disjunctive normal forms for f'', f', then for each clause of f' there exists a clause of f'' composed of a subset of its variables only, i.e., *for each $j' \in \{1, \ldots, \mathfrak{f}'\}$ there exists $j'' \in \{1, \ldots, \mathfrak{f}''\}$ such that*

$$K''_{j''} = (x_{e''_1} \,\&\ldots\&\, x_{e''_{\mathfrak{g}''}}),$$
$$K'_{j'} = (x_{e'_1} \,\&\ldots\&\, x_{e'_{\mathfrak{g}'}}),$$

with

$$A''_{j''} = \{x_{e''_1}, \ldots, x_{e''_{\mathfrak{g}''}}\} \subseteq A'_{j'} = \{x_{e'_1}, \ldots, x_{e'_{\mathfrak{g}'}}\}.$$

The next two theorems state that even partial decompositions into conjunctions/disjunctions are closely related for $f' \leq f''$.

THEOREM 3: *Let $f'(x_1, \ldots, x_h)$ and $f''(x_1, \ldots, x_h)$ be monotone, $f' \leq f''$. If $1 = f'(C') > f''(C'') = 0$ for some C', C'', and f'' can be partitioned into $f''(x_1, \ldots, x_h) = f_A(x_{e_1}, \ldots, x_{e_\mathfrak{g}}) \,\&\, f_B(x_1, \ldots, x_h)$, for some monotone f_A, f_B, with $A = \{x_{e_1}, \ldots, x_{e_\mathfrak{g}}\} \subseteq B = \{x_1, \ldots, x_h\}$, such that $f_A(C''_{e_1}, \ldots, C''_{e_\mathfrak{g}}) = 0$, then C' must differ from C'' in a formal parameter of f_A,* i.e., *there exists $x_e \in A$ such that $1 = C'_e > C''_e = 0$.*

Proof: Since $f_A(C''_{e_1}, \ldots, C''_{e_\mathfrak{g}}) = 0$ and $f_A(C'_{e_1}, \ldots, C'_{e_\mathfrak{g}}) \geq f''(C') \geq f'(C') = 1$, we have that f_A is not a constant function, and hence its transformation into the conjunctive normal form does not degenerate into a single constant.

Thus $f_A(C''_{e_1}, \ldots, C''_{e_\mathfrak{g}}) = 0$ implies that f_A has a clause not satisfied for C'', *i. e.*, we have $K''_A = (x_{a''_1} \vee \ldots \vee x_{a''_\mathfrak{d}})$ with $\{x_{a''_1}, \ldots, x_{a''_\mathfrak{d}}\} \subseteq A$ and $C''_{a''_1} = C''_{a''_2} = \ldots = C''_{a''_\mathfrak{d}} = 0$. But we can find a conjunctive normal form for $f'' = f_A \& f_B$ containing all clauses for f_A, and hence, by Lemma 11, f' has a clause K'_A composed of a subset of $\{x_{a''_1}, \ldots, x_{a''_\mathfrak{d}}\}$. Since $f'(C') = 1$,

K'_A is satisfied for C', hence, there exists $x_e \in \{x_{a''_1}, \ldots, x_{a''_0}\} \subseteq A$ such that $C'_e = 1$ and $C''_e = 0$. □

A similar theorem holds for decompositions of f' into disjunctions. The corresponding proof mirrors Theorem 3, using the disjunctive normal forms and Lemma 12, instead of Lemma 11.

THEOREM 4: *Let $f'(x_1, \ldots, x_h)$ and $f''(x_1, \ldots, x_h)$ be monotone, $f' \leq f''$. If $1 = f'(C') > f''(C'') = 0$ for some C', C'', and f' can be partitioned into $f'(x_1, \ldots, x_h) = f_A(x_{e_1}, \ldots, x_{e_g}) \vee f_B(x_1, \ldots, x_h)$, for some monotone f_A, f_B, with $A = \{x_{e_1}, \ldots, x_{e_g}\} \subseteq B = \{x_1, \ldots, x_h\}$, such that $f_A(C'_{e_1}, \ldots, C'_{e_g}) = 1$, then C' must differ from C'' in a formal parameter of f_A,* i. e., *there exists $x_e \in A$ such that $1 = C'_e > C''_e = 0$.*

5. ALTERNATION RESISTANCE

We are now ready to state and prove main theorems. First, we shall introduce the notion of Σ_k/Π_k, S-resistant words, which compensates us for the defects of S-equivalence mentioned in Section 2.

DEFINITION 9: An ordered pair of words (w', w'') is Σ_k, S-resistant (Π_k, S-resistant) for a machine A, if

a) w' and w'' are S-equivalent,

b) for each configuration p,

*b*1) going to enter w' and w'' by crossing their left/right margins in the next computation step,

*b*2) that is S-bounded on w' and on w'',

*b*3) of alternating level Σ_k or less (Π_k or less, respectively)

we have $f_{p,w'} \leq f_{p,w''}$.

By (*a*), individual computation paths not using more than S space cannot distinguish w' from w'' on inputs $\alpha w' \beta$ and $\alpha w'' \beta$. By Lemma 4*b*, it is then sufficient to suppose that p is S-bounded on one of them only. Further, by Lemma 5*c*, we have $\mathrm{Ex}_{p,w'} = \mathrm{Ex}_{p,w''}$, *i. e.*, the functions $f_{p,w'}$ and $f_{p,w''}$ have the same formal parameter list and the accept/reject statuses of p on the inputs $\alpha w' \beta$ and $\alpha w'' \beta$ depend on the accept/reject statuses of the same configurations leaving w' and w'' for the first time. The condition $f_{p,w'} \leq f_{p,w''}$ indicates that these statuses are closely related, unless the machine A uses more space or a higher level of alternation. The next theorem clarify the above idea.

THEOREM 5: *Let* (w', w'') *be a* Π_k, S*-resistant pair. Then, for each* α *and* β,

a) $\alpha\, w'\, \beta$ *and* $\alpha\, w''\, \beta$ *are* S*-equivalent,*

b) for each configuration p,

b1) with the input head positioned outside w' *and* w'', i. e., *on* α *or on* β *(the input head positions are relative to the left margins of* α *or* β*),*

b2) that is S*-bounded on* $\alpha\, w'\, \beta$ *and on* $\alpha\, w''\, \beta$,

b3) of alternating level Π_k *or less*

we have $f_{p,\, \alpha\, w'\, \beta} \leq f_{p,\, \alpha\, w''\, \beta}$.

The same holds for the $\Sigma_{k'}$ S*-resistant* (w', w'') *and each* p *of alternating level* Σ_k *or less.*

As a special case, for p entering $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ in the next computation step, we get immediately that $(\alpha\, w'\, \beta,\ \alpha\, w''\, \beta)$ is again a Π_k/Σ, S-resistant pair, respectively.

Proof: (a) follows easily from Lemma 3. Now, let p be any configuration satisfying ($b1$), ($b2$), and ($b3$). Because p is S-bounded on $\alpha\, w'\, \beta$ and on $\alpha\, w''\, \beta$, and w', w'' are S-equivalent, $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ have the same set of exit configurations for p, *i. e.*, $\mathrm{Ex}_{p,\, \alpha\, w'\, \beta} = \mathrm{Ex}_{p,\, \alpha\, w''\, \beta} = \{p_1, \ldots, p_h\}$ for some $p_1, \ldots, p_h$ leaving $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$. Thus, $f_{p,\, \alpha\, w'\, \beta}$ and $f_{p,\, \alpha\, w''\, \beta}$ have the same formal parameter list $x_{p_1}, \ldots, x_{p_h}$.

We have to show that $f_{p,\, \alpha\, w'\, \beta} \leq f_{p,\, \alpha\, w''\, \beta}$. By Lemma 10$b$, it is sufficient to show, for each $C' = (C'_{p_1}, \ldots, C'_{p_h})$ and $C'' = (C''_{p_1}, \ldots, C''_{p_h})$, that if $1 = f_{p,\, \alpha\, w'\, \beta}(C') > f_{p,\, \alpha\, w''\, \beta}(C'') = 0$, then $C' \neg \leq C''$, *i. e.*, there exists $p_e \in \{p_1, \ldots, p_h\}$ with $1 = C'_{p_e} > C''_{p_e} = 0$. In other words, we shall interrupt all computation paths beginning in p on $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ as soon as they are leaving $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ ($\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ may be substrings of some longer inputs) and assign some accept/reject statuses $C' = (C'_{p_1}, \ldots, C'_{p_h})$ and $C'' = (C''_{p_1}, \ldots, C''_{p_h})$ to the configurations $p_1, \ldots, p_h$ leaving $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$, respectively. We show that if this assignment causes that p is accepting on $\alpha\, w'\, \beta$ but rejecting on $\alpha\, w''\, \beta$, then, for some p_e leaving $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$, we had to assign the accept status on $\alpha\, w'\, \beta$, but reject status on $\alpha\, w''\, \beta$.

Before proving this, we shall show two slightly weaker claims for configurations reachable from p on $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$.

CLAIM 1: Let r be a configuration

– with the input head positioned outside w' and w'',

– reachable from p by a path never leaving $\alpha w' \beta$ or $\alpha w'' \beta$, (hence, S-bounded on $\alpha w' \beta$ and $\alpha w'' \beta$, and of alternating level Π_k or less),

– Π_l-accepting on $\alpha w' \beta$, but Π_l-rejecting on $\alpha w'' \beta$, for some $l \leq k$, then

(i) either $1 = C'_{p_e} > C''_{p_e} = 0$, for some $p_e \in \mathrm{Ex}_{p,\alpha w' \beta} = \mathrm{Ex}_{p,\alpha w'' \beta}$,

(ii) or there exists a configuration r'

– with the input head positioned outside w' and w'',

– reachable from r by a path never leaving $\alpha w' \beta$ or $\alpha w'' \beta$,

– $\Sigma_{l'}/\Pi_{l'}$, -accepting on $\alpha w' \beta$, but $\Sigma_{l'}/\Pi_{l'}$, -rejecting on $\alpha w'' \beta$, for some $l' < l \leq k$ (*i. e.*, of alternating level at most Σ_{l-1}).

Proof of Claim 1: Because r is Π_l-rejecting for $\alpha w'' \beta$, there must be a rejecting computation path beginning in r on $\alpha w'' \beta$. We now have the following cases:

0) The rejecting path reaches the margin making no alternation and leaves $\alpha w'' \beta$ in a configuration p_e that is Π_l-rejecting. (See *fig.* 9).

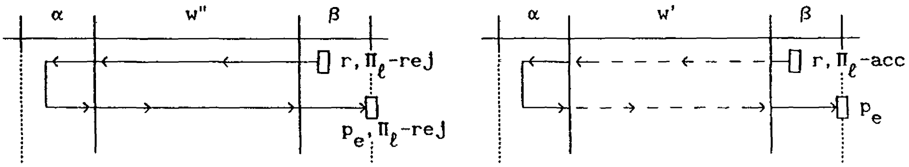

Figure 9

Because w', w'' are S-equivalent and r is S-bounded on $\alpha w' \beta$ and $\alpha w'' \beta$, p_e is also reachable from r at the corresponding margin of $\alpha w' \beta$. Since r is Π_l-accepting on $\alpha w' \beta$, all alternation-free paths from r must be accepting. Therefore, p_e is Π_l-accepting on $\alpha w' \beta$. Thus we have $p_e \in \mathrm{Ex}_{p,\alpha w' \beta} = \mathrm{Ex}_{p,\alpha w'' \beta}$ with $C'_{p_e} = 1$ and $C''_{p_e} = 0$.

1*a*) The rejecting path from the Π_l-rejecting r alternates outside w'' on $\alpha w'' \beta$, *i. e.*, it enters a Σ_{l-1}-rejecting configuration r' with the input head positioned on α or β. (See *fig.* 10)

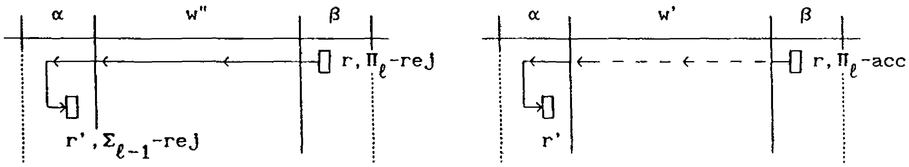

Figure 10

But then the Σ_{l-1}-configuration r' is also reachable from the Π_l-accepting r on $\alpha\, w'\, \beta$. Since all computation paths from r on $\alpha\, w'\, \beta$ must be accepting, we have that r' is Σ_{l-1}-accepting on $\alpha\, w'\, \beta$, but Σ_{l-1}-rejecting on $\alpha\, w''\, \beta$.

1*b*) The rejecting path from r alternates inside w'' on $\alpha\, w''\, \beta$, *i. e.*, it enters a Σ_{l-1}-rejecting r'' with the head on w''.

Let r' be the last configuration crossing the border of w'' along the path from r to r''. r' is a Π_l-configuration, moreover, it is rejecting, since we have a path from r' to r''. (See *fig.* 11).

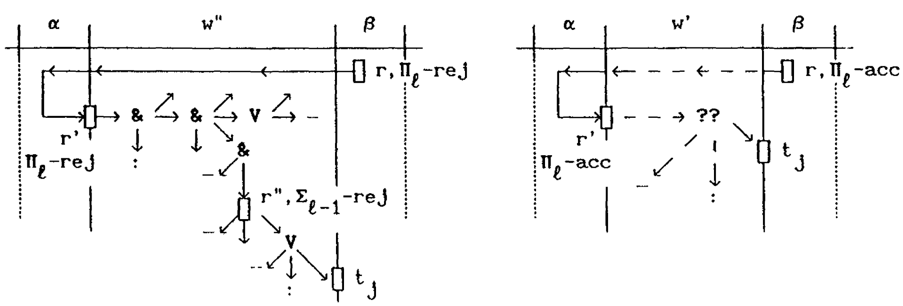

Figure 11

All branches are universal along the path from r' to r'' and hence $f_{r',\,w''}$ can be expressed in the form

$$f_{r',\,w''}(x_{r_1},\, \ldots,\, x_{r_{\mathfrak{f}}}) = f_{r'',\,w''}(x_{t_1},\, \ldots,\, x_{t_{\mathfrak{g}}})\ \&\ \bar{f}(x_{r_1},\, \ldots,\, x_{r_{\mathfrak{f}}}),$$

$\uparrow$ branch to r'' $\qquad\qquad$ $\uparrow$ all other branches

where $\mathrm{Ex}_{r'',\,w''} = \{t_1,\, \ldots,\, t_{\mathfrak{g}}\} \subseteq \mathrm{Ex}_{r',\,w''} = \{r_1,\, \ldots,\, r_{\mathfrak{f}}\}$ denote the sets of exit configurations of w'' for r'' and r', respectively. Let $\breve{C}'' = (\breve{C}''_{r_1},\, \ldots,\, \breve{C}''_{r_{\mathfrak{f}}})$ denote the accept/reject statuses for the exit configurations $r_1,\, \ldots,\, r_{\mathfrak{f}}$ on the string $\alpha\, w''\, \beta$. Because both r' and r'' are rejecting on $\alpha\, w''\, \beta$, we have

$$\breve{C}''_{r'} = f_{r',\,w''}(\breve{C}''_{r_1},\, \ldots,\, \breve{C}''_{r_{\mathfrak{f}}}) = f_{r'',\,w''}(\ldots)\ \&\ \bar{f}(\ldots) = 0,$$

with

$$\breve{C}''_{r''} = f_{r'',\,w''}(\breve{C}''_{r_t},\, \ldots,\, \breve{C}''_{t_{\mathfrak{g}}}) = 0.$$

On the other hand, r' is also reachable from the Π_l-accepting r on $\alpha\, w'\, \beta$. Because r' is reachable from p via r, it is S-bounded on $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$ (hence, S-bounded on w' and w'') and of alternating level Π_k or less. (It is of level Π_l, with $l \leq k$.) Further, (w', w'') is a Π_k, S-resistant pair and hence

$$f_{r', w'} \leq f_{r', w''}.$$

$f_{r', w'}$ has the same formal parameter list, corresponding to $\mathrm{Ex}_{r', w'} = \mathrm{Ex}_{r', w''} = \{r_1, \ldots, r_f\}$. Since all alternation-free paths from r on $\alpha\, w'\, \beta$ must be successful, r' must be Π_l-accepting on $\alpha\, w'\, \beta$ and hence the accept/reject statuses $\check{C}' = (\check{C}'_{r_1}, \ldots, \check{C}'_{r_f})$ of exists on $\alpha\, w'\, \beta$ must satisfy

$$\check{C}'_{r'} = f_{r', w'}(\check{C}'_{r_1}, \ldots, \check{C}'_{r_f}) = 1.$$

But then, by Theorem 3, $\check{C}'$ must differ from $\check{C}''$ in a formal parameter of $f_{r'', w''}$, *i. e.*, $1 = \check{C}'_{t_j} > \check{C}''_{t_j} = 0$, for some $t_j \in \mathrm{Ex}_{r'', w''} \subseteq \mathrm{Ex}_{r', w''} = \mathrm{Ex}_{r', w'} = \{r_1, \ldots, r_f\}$. In other words, there exists a configuration t_j reachable from r' on both $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$, having just left w' and w'' by crossing their margins, that is accepting on $\alpha\, w'\, \beta$, but rejecting on $\alpha\, w''\, \beta$. Moreover, t_j is reachable from the Σ_{l-1}-rejecting r'' and therefore it is of alternating level Σ_{l-1} or less.

All cases above were confirming the hypothesis of the Claim 1. We shall now show that all cases that remain to consider lead to contradictions and hence cannot happen.

2*a*) Suppose that the Π_l-rejecting r on $\alpha\, w''\, \beta$ has an infinite cycle, making no alternation at all, and that at least a part of this cycle lies outside w''. (See *fig.* 12.)

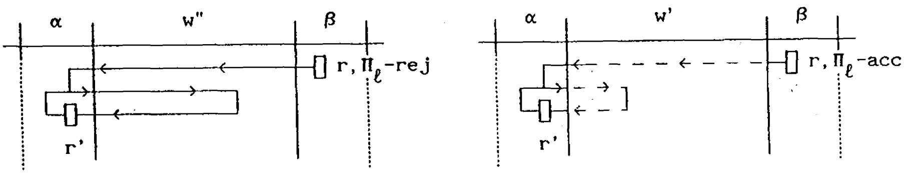

Figure 12

Thus, we can find a configuration r' positioned outside w'' such that (*a*) r' is reachable from r on $\alpha\, w''\, \beta$, (*b*) r' is reachable from r' on $\alpha\, w''\, \beta$. But then r' is also reachable from r on $\alpha\, w'\, \beta$, similarly, r' is reachable from r' on $\alpha\, w'\, \beta$. This gives an alternation-free cycle reachable from the Π_l-acceptating r on $\alpha\, w'\, \beta$, which is a contradiction.

2*b*) Suppose that the entire cycle is executed within w''. (See *fig.* 13.)

Let r' be the last configuration along the path from r to the cycle crossing the border of w''. By a reasoning very similar to Case 1*b*, r' is Π_l-rejecting

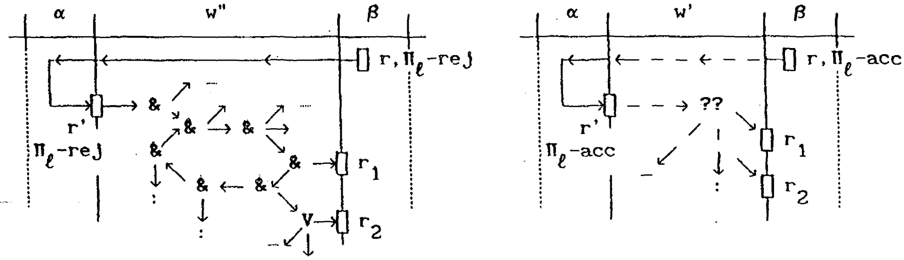

Figure 13

on $\alpha\, w''\, \beta$ but Π_l-accepting on $\alpha\, w'\, \beta$. Because we have an alternation-free path from the universal r' into the cycle on w'', $f_{r', w''}$ can be expressed in the form

$$f_{r', w''}(x_{r_1}, \ldots, x_{r_f}) = 0 \ \& \ \bar{f}(x_{r_1}, \ldots, x_{r_f}),$$

branch for cycle ↑ ↑ other branches

i. e., $f_{r', w''}$ is a constant function returning always zero and overriding the accept/reject statuses $\breve{C}'' = (\breve{C}''_{r_1}, \ldots, \breve{C}''_{r_f})$ of exit configurations. Because r' is S-bounded on w' and w'', of alternating level Π_l, $l \leq k$, and (w', w'') is a Π_k, S-resistant pair, we have $f_{r', w'} \leq f_{r', w''}$ and therefore $f_{r', w'}$ is also a constant function returning always zero.

On the other hand, r' is Π_l-accepting on $\alpha\, w'\, \beta$ and hence $\breve{C}'_{r'} = f_{r', w'}(\breve{C}'_{r_1}, \ldots, \breve{C}'_{r_f}) = 1$ for some $\breve{C}' = (\breve{C}'_{r_1}, \ldots, \breve{C}'_{r_f})$, which is a contradiction.

3) Finally, suppose that the Π_l-rejecting r has an alternation-free path that halts in a rejecting configuration on $\alpha\, w''\, \beta$. There are two subcases again, corresponding to Case 2*a* and 2*b*: Either the machine halts outside w'' on $\alpha\, w''\, \beta$, and then the same halting and rejecting configuration is also reachable from the Π_l-accepting r on $\alpha\, w'\, \beta$, or it halts inside w''. But then we can find a configuration r' crossing the border of w'/w'' such that $f_{r', w''}(\ldots) = 0 \ \& \ \bar{f}(\ldots)$ due to a halting path that rejects, $f_{r', w'} \leq f_{r', w''}$, but $f_{r', w'}(\breve{C}') = 1$ for some $\breve{C}'$. In either case, this is a contradiction.

This completes the proof of the Claim 1. A very similar claim can be formulated for existential configurations.

CLAIM 2: Let r be a configuration

– with the input head positioned outside w' and w'',

– reachable from p by a path never leaving $\alpha\, w'\, \beta$ or $\alpha\, w''\, \beta$, (hence, S-bounded on $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$, and of alternating level Π_k or less),

– Σ_l-accepting on $\alpha\, w'\, \beta$, but Σ_l-rejecting on $\alpha\, w''\, \beta$, for some $l < k$, then

(i) either $1 = C'_{p_e} > C''_{p_e} = 0$, for some $p_e \in \mathrm{Ex}_{p,\, \alpha\, w'\, \beta} = \mathrm{Ex}_{p,\, \alpha\, w''\, \beta}$,

(ii) or there exists a configuration r'

– with the input head positioned outside w' and w'',

– reachable from r by a path never leaving $\alpha\, w'\, \beta$ or $\alpha\, w''\, \beta$,

– $\Sigma_{l'}/\Pi_{l'}$-accepting on $\alpha\, w'\, \beta$, but $\Sigma_{l'}/\Pi_{l'}$-rejecting on $\alpha\, w''\, \beta$, for some $l' < l < k$ (*i. e.*, of alternating level at most Π_{l-1}).

Proof of Claim 2: The argument mirrors the proof of Claim 1 but, instead of the rejecting paths beginning in the Π_l-rejecting r on $\alpha\, w''\, \beta$, we analyze *accepting* paths beginning in the Σ_l-accepting r on $\alpha\, w'\, \beta$. The only exceptions are Cases $2a$ and $2b$ that correspond to nothing in Claim 2, because no accepting path can be an infinite cycle. To illustrate what Alice can see through the looking glass, we shall review Case $1b$ (alternation inside).

Suppose that the existential configuration r is Σ_l-rejecting on $\alpha\, w''\, \beta$, but Σ_l-accepting on $\alpha\, w'\, \beta$, because it has a successful computation path that enters a Π_{l-1}-accepting r'' positioned inside w'.

Then r', the last configuration crossing the border of w' along the path from r to r'' is Σ_l-accepting on $\alpha\, w'\, \beta$. All branches are existential along the path from r' to r'', and hence

$$f_{r',\, w'}\,(x_{r_1},\, \ldots,\, x_{r_\mathfrak{f}}) = \underset{\uparrow\ \text{branch to } r''}{f_{r'',\, w'}\,(x_{t_1},\, \ldots,\, x_{t_\mathfrak{g}})} \vee \underset{\uparrow\ \text{other branches}}{\bar{f}\,(x_{r_1},\, \ldots,\, x_{r_\mathfrak{f}})},$$

with $\mathrm{Ex}_{r'',\, w'} = \{t_1, \ldots, t_\mathfrak{g}\} \subseteq \mathrm{Ex}_{r',\, w'} = \{r_1, \ldots, r_\mathfrak{f}\}$. For the accept/reject statuses of exit configurations on $\alpha\, w'\, \beta$ we then get

$$\check{C}'_{r'} = f_{r',\, w'}\,(\check{C}'_{r_1},\, \ldots,\, \check{C}'_{r_\mathfrak{f}}) = f_{r'',\, w'}\,(\ldots) \vee \bar{f}\,(\ldots) = 1,$$

with

$$\check{C}'_{r''} = f_{r'',\, w'}\,(\check{C}'_{t_1}, \ldots, \check{C}'_{t_\mathfrak{g}}) = 1.$$

On the other hand, r' is reachable from the Σ_l-rejecting r on $\alpha\, w''\, \beta$. No path beginning in the Σ_l-rejecting r can be successful and therefore r' is Σ_l-rejecting on $\alpha\, w''\, \beta$. For the exits on $\alpha\, w''\, \beta$ this gives

$$\check{C}''_{r'} = f_{r',\, w''}\,(\check{C}''_{r_1},\, \ldots,\, \check{C}''_{r_\mathfrak{f}}) = 0.$$

Since $\mathrm{Ex}_{r',\, w'} = \mathrm{Ex}_{r',\, w''}$ and $f_{r',\, w'} \leq f_{r',\, w''}$ (by the same argument as in Case $1b$ of Claim 1), using Theorem 4 instead of Theorem 3, we get

$1 = \check{C}'_{t_j} > \check{C}''_{t_j} = 0$, for some $t_j \in \mathrm{Ex}_{r'', w'}$, *i. e.*, there is a configuration that is accepting on $\alpha\, w'\, \beta$ but rejecting on $\alpha\, w''\, \beta$, positioned outside w', w'', and of alternating level at most Π_{l-1}, because it is reachable from the Π_{l-1}-configuration r''. This proves the Claim 2.

Proof of Theorem 5, continued: Recall that if the configuration p satisfies $(b1)$, $(b2)$, and $(b3)$, then $\mathrm{Ex}_{p, \alpha\, w'\, \beta} = \mathrm{Ex}_{p, \alpha\, w''\, \beta} = \{p_1, \ldots, p_h\}$. It remains to show that if $1 = f_{p, \alpha\, w'\, \beta}(C') > f_{p, \alpha\, w''\, \beta}(C'') = 0$, for some $C' = (C'_{p_1}, \ldots, C'_{p_h})$ and $C'' = (C''_{p_1}, \ldots, C''_{p_h})$ representing the accept/reject statuses of exit configurations $p_1, \ldots, p_h$ on the margins of $\alpha\, w'\, \beta$ and $\alpha\, w''\, \beta$, respectively, then $1 = C'_{p_e} > C''_{p_e} = 0$, for some $p_e \in \{p_1, \ldots, p_h\}$.

Suppose that p is Π_k-accepting on $\alpha\, w'\, \beta$ but Π_k-rejecting on $\alpha\, w''\, \beta$, for some C' and C''. Then, by Claim 1, for $r = p$, we get

(i) either $1 = C'_{p_e} > C''_{p_e} = 0$ for some p_e and we are done,

(ii) or there must exist $r^{(1)}$ with the input head positioned outside w' and w'', reachable from p by a path never leaving $\alpha\, w'\, \beta$ or $\alpha\, w''\, \beta$, of alternating level $l' < k$, that is $\Sigma_{l'}/\Pi_{l'}$-accepting on $\alpha\, w'\, \beta$ but $\Sigma_{l'}/\Pi_{l'}$-rejecting on $\alpha\, w''\, \beta$.

If, for example, $r^{(1)}$ is an existential configuration, then we can use Claim 2 and get

(i) either $1 = C'_{p_e} > C''_{p_e} = 0$ for some p_e and we are done,

(ii) or there must exist $r^{(2)}$ with the input head positioned outside w' and w'', reachable from $r^{(1)}$ by a path never leaving $\alpha\, w'\, \beta$ or $\alpha\, w''\, \beta$ (hence, reachable from p), of alternating level $l'' < l' < k$, $\Sigma_{l''}/\Pi_{l''}$-accepting on $\alpha\, w'\, \beta$ but $\Sigma_{l''}/\Pi_{l''}$-rejecting on $\alpha\, w''\, \beta$. (If $r^{(1)}$ is universal, we use Claim 1 again.) . . . .

This process cannot be repeated more than k times and hence, sooner or later, we must get $1 = C'_{p_e} > C''_{p_e} = 0$.

This completes the proof of the theorem. The argument for the Σ_k, S-resistant (w', w'') is the same, but the starting alternation level is existential. □

Before passing further, we shall review the problems that we are going to tackle on the way from the resistant words to resistant languages. Suppose that, for some language L', we have $w'_+ \in L'$ and $w'_- \notin L'$ such that a $\Sigma_k/\Pi_k - \mathrm{SPACE}(s(n))$ machine A' cannot distinguish w'_+ from w'_-. But w'_+ and w'_- are quite long and the space of size $s(|w'_+|)$ or $s(|w'_-|)$ might be sufficient for A' to distinguish them. Therefore, we provide also a third example w'_0 (we do not care whether $w'_0 \in L'$), that restrains A' from using too much space, *i. e.*, A' cannot use more space on the inputs w'_+ or w'_- than on w'_0. (Still, A' can use substantially more space on other inputs of equal lengths.)

In addition, for each $G \geq 0$, we claim that no Σ_{k+G}/Π_{k+G}-SPACE$(s(n))$ machine A can use any Σ_k/Π_k-SPACE$(s(n))$ machine A' as its subprogram (roughly speaking, as its oracle) to distinguish w'_+ from w'_-. (Now they can be some substrings of longer inputs.) We shall call such languages Σ_k/Π_k-SPACE$(s(n))$ resistant. Having given a Σ_k-SPACE$(s(n))$ resistant language L', we shall design a Π_{k+1}-SPACE$(s(n))$ resistant language L with counterexamples w_+, w_-, and w_0, that are composed of w'_+, w'_-, and w'_0. But two problems arise here: First, A can use more alternations than k, second, the worktape space limit has been increased from $s(|w'_+|)$ or $s(|w'_-|)$ to $s(|\alpha w'_+ \beta|)$ or $s(|\alpha w'_- \beta|)$, respectively.

Thus, to design counterexamples w'_+, w'_-, and w'_0, we need some *a priori* information about the environment in which these counterexamples will be used, among others, about w_+, w_-, and w_0. This "*a priori* information" allows us to fool any Σ_{k+G}/Π_{k+G}-SPACE$(s(n))$ machine, for arbitrarily large $G \geq 0$.

Languages separating Σ_k-SPACE$(s(n))$ from Π_k-SPACE$(s(n))$, for $k \geq 2$, have a simple block structure. The structure of the blocks can be described by a sequence of regular languages R_2, R_3, R_4, ... defined as follows:

Definition 10: Let $\{a, b\}$ denote a two-letter alphabet. Then

$$R_2 = a^+,$$
$$R_k = b\,(R_{k-1}\,b)^+, \quad \text{for each } k \geq 3.$$

It is easy to show, by induction on k, that $w \in R_k$ begins with $b^{k-2}\,a\ldots$, ends by $\ldots ab^{k-2}$, and does not contain more than $2k-5$ consecutive b's. This implies that it can be partitioned unambiguously into $w = bu_1\,bu_2\,b\ldots bu_{\mathfrak{f}}b$, for some $u_1, \ldots, u_{\mathfrak{f}} \in R_{k-1}$. That is, if $w = bu'_1\,bu'_2\,b\ldots bu'_{\mathfrak{g}}\,b$, for some $u'_1, \ldots, u'_{\mathfrak{g}} \in R_{k-1}$, then $\mathfrak{g} = \mathfrak{f}$ and $u_1 = u'_1, \ldots, u_{\mathfrak{f}} = u'_{\mathfrak{f}}$. This partition is determined by the positions of substrings $ab^{k-3}\,bb^{k-3}\,a = ab^{2k-5}\,a$ in w.

The next definition will be used to generate the counterexamples "w_0" that restrains Σ_{k+G}/Π_{k+G}-SPACE$(s(n))$ machines from using too much space, provided that we are given w'_0, restraining Σ_k/Π_k-SPACE$(s(n))$ machines, and $G \geq 0$, the rank of environment.

DEFINITION 11: Let $G \geq 0$ and $w \in \{a, b\}^*$, $|w| \geq 2$. We define $E(G, w)$, *the environment of rank G for w*, by

$$E(0, w) = w,$$
$$E(G+1, w) = E(G, b(wb)^{|w|-1}), \quad \text{for each } G \geq 0.$$

For example, $E(1, aaa) = E(0, b(aaab)^{3-1}) = baaabaaab$. $E(G, w)$ is also a string composed of $|w|-1$ consecutive $E(G-1, w)$ blocks, enclosed in b's. It is easy to see, by induction on G, for each $G \geq 0$ and each w, that

$$|E(G, w)| = |w|^{2^G}. \tag{3}$$

Note also that if $w \in R_k$, then $E(G, w) \in R_{k+G}$. We are now ready to present a formal definition of the Σ_k/Π_k-SPACE$(s(n))$ resistant language.

DEFINITION 12: A language L is Σ_k-SPACE$(s(n))$ resistant (Π_k-SPACE$(s(n))$ resistant), if , for each $s(n)$ space bounded alternating machine A, each $G \geq 0$, and each $\breve{n} \geq 0$,

a) there exist $w_+ \in R_k \cap L$, $w_- \in R_k - L$, and $w_0 \in R_k$, such that

b) $|w_0| \geq \breve{n}$,

c) w_+, w_-, and w_0 are $\text{Space}_A(E(G, w_0))$-equivalent,

d) (w_+, w_-) is a Σ_k, $\text{Space}_A(E(G, w_0))$-resistant pair.
(Π_k, $\text{Space}_A(E(G, w_0))$-resistant pair, respectively.)

Thus, we must fool each $s(n)$ space bounded machine A making an arbitrary number of alternations, however, (d) concerns configurations of alternating level at most Σ_k/Π_k only. Such configurations may be viewed as "oracle entry points" giving answers to some partial questions as the computation demands. We claim that such entry points cannot be used to distinguish $w_+ \in L$ from $w_- \notin L$.

Second, the worktape space limit for such entry points is as much as $\text{Space}_A(E(G, w_0))$, *i. e.*, the worktape space used by A on the input $E(G, w_0)$. (*See* also Def. 3 and Def. 11.) Note that A may potentially use $s(|w_0|^{2^G})$ space on the input $E(G, w_0)$, by (3). Thus, for arbitrarily large G, we should find w_+, w_-, and w_0 so that w_+ and w_- cannot be distinguished if they are inserted into inputs of length $|w_0|^{2^G}$. However, the condition (c) ensures that A does not try to use too much space on inputs $\alpha w_+ \beta$ or $\alpha w_- \beta$ unless it tries to do so on $\alpha w_0 \beta$, by Lemma 4*a*. The condition (b) orders a lower bound on the length of w_0 and, indirectly, on the lengths of w_+ and w_-.

THEOREM 6: *Let L' be a Σ_{k-1}-SPACE $(s(n))$ resistant language, for some $k \geq 3$, with $\sup_{n \to \infty} s(n)/\log(n) = 0$. Then the language*

$$L = \{w \in R_k;\ w = bw_1\, bw_2\, b \ldots bw_{\mathfrak{f}}\, b,\ \exists j \in \{1, \ldots, \mathfrak{f}\}:\ w_j \in L',\ w_1, \ldots, w_{\mathfrak{f}} \in R_{k-1}\}$$

is Π_k-SPACE $(s(n))$ resistant.

Proof: Let A be an $s(n)$ space bounded machine and let c be a constant for A satisfying (2), *i. e.*, the number of reachable memory states for each input w is bounded by $c^{s(|w|)}$. Let $G \geq 0$ and $\breve{n} \geq 0$. Define $G' = G + 1$ and take $\breve{n}'$ so that

$$\breve{n}' \geq \max\{\breve{n}, 2\}, \tag{4}$$

and

$$(c^{s(n^{2^{G+1}})})^6 < n - 1, \quad \text{for each } n \geq \breve{n}'. \tag{5}$$

By Lemma 6, using $H = 2^{G+1}$, such $\breve{n}'$ does exist. Because the language L' is Σ_{k-1}-SPACE $(s(n))$ resistant, we have, for any given A, G', and $\breve{n}'$, that

a') there exist $w'_+ \in R_{k-1} \cap L'$, $w'_- \in R_{k-1} - L'$, and $w'_0 \in R_{k-1}$, such that

b') $|w'_0| \geq \breve{n}'$,

c') w'_+, w'_-, and w'_0 are $\text{Space}_A(E(G', w'_0))$-equivalent,

d') (w'_+, w'_-) is a Σ_{k-1}, $\text{Space}_A(E(G', w'_0))$-resistant pair.

We have to find w_+, w_-, and w_0 with the corresponding properties for the language L. Define

$$w_0 = b(w'_0\, b)^m,$$

where

$$m = |w'_0| - 1. \tag{6}$$

Clearly, $w_0 \in R_k$, since $w'_0 \in R_{k-1}$ and $m \geq 1$, by (a'), (b'), and (4). Further, by (b') and (4), $|w_0| = |w'_0|^2 \geq \breve{n}$. Because

$$E(G, w_0) = E(G, b(w'_0\, b)^{|w'_0|-1}) = E(G+1, w'_0) = E(G', w'_0), \tag{7}$$

by Definition 11, we can modify (c') and (d') as follows:

c'') w'_+, w'_-, and w'_0 are $\text{Space}_A(E(G, w_0))$-equivalent,

d'') (w'_+, w'_-) is a Σ_{k-1}, $\text{Space}_A(E(G, w_0))$-resistant pair.

Now, define an extended version of w_0 by

$$w_E = b(w'_0\, b)^{m+(4k+3)\cdot m!}.$$

Note that the length of w_E depends on the alternating level k. First, we shall prove that w_0 and w_E are $\mathrm{Space}_A(E(G, w_0))$-equivalent: It is easy to show that

$$|E(G, w_0)| = |w_0'|^{2^{G+1}}, \tag{8}$$

using (7), (3), and $G' = G + 1$. Because $|w_0'| \geq \check{n}'$, by (b'), we can use (5) and get

$$({}_c s^{(|w_0'|^{2^{G+1}})})^6 < |w_0'| - 1. \tag{9}$$

But then

$$({}_c\mathrm{Space}_{A(E(G, w_0))})^6 < m, \tag{10}$$

using (1), (8), (9), and (6). This implies, by Lemma 9 and 3, that $w_0 = b(w_0' b)^m$ and $w_E = b(w_0' b)^{m+im!}$ (with $i = 4k + 3$) are $\mathrm{Space}_A(E(G, w_0))$-equivalent, because the number of $(w_0' b)$-blocks in w_0 is large enough, compared to the worktape space limit for the input $E(G, w_0) = E(G', w_0')$. The design of inputs satisfying (9) and (6) plays a dominant role here. Finally, define

$$w_- = b(w_-' b)^{m+(4k+3)\cdot m!},$$
$$w_+ = b(w_-' b)^{m+2k\cdot m!}\, w_+'\, b(w_-' b)^{(2\cdot m!-1)+2k\cdot m!+m!}.$$

Clearly, $w_+, w_- \in R_k$, since $w_+', w_-' \in R_{k-1}$, by (a'). The strings w_+ and w_- consist of the same number of blocks as w_E, since replacing all w_0'-blocks by the w_-'-blocks transforms w_E into w_-. The string w_+ differs from w_- in the block on the position $m + 2k \cdot m! + 1$ only, where it has w_+' instead of w_-'. (See *fig.* 14 for the structure of w_+ and w_-.)

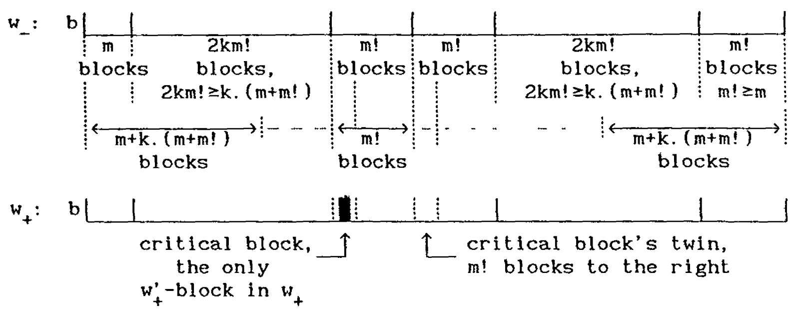

Figure 14

It is obvious that $w_+ \in L$ *and* $w_- \notin L$, because w_+ contains one $w'_+ \in L'$ while w_- is composed of the $w'_- \notin L'$ blocks only. (The partition of w_+ and w_- into the strings in R_{k-1} is anambiguous, hence, for example, we cannot get $w_- = bu_1\, bu_2\, b \dots bu_{\mathfrak{f}}\, b$ for some $u_1, \dots, u_{\mathfrak{f}} \in R_{k-1}$ so that $\exists\, u_j \in L'$. *See* also the remark below Def. 10.)

Because w'_+, w'_-, *and* w'_0 *are* $\text{Space}_A\,(E\,(G,\, w_0))$*-equivalent* by (c″), we have, by Lemma 3, that w_+ and w_- are $\text{Space}_A\,(E\,(G,\, w_0))$-equivalent to w_E. Since w_E is $\text{Space}_A\,(E\,(G,\, w_0))$-equivalent to w_0, by (10) and Lemma 9, we get that w_+, w_-, *and* w_0 *are* $\text{Space}_A\,(E\,(G,\, w_0))$*-equivalent.*

It only remains to prove that $(w_+,\, w_-)$ is a Π_k, $\text{Space}_A\,(E\,(G,\, w_0))$-resistant pair, *i. e.*, that $f_{p,\, w_+} \leq f_{p,\, w_-}$ for each configuration p that is (i) going to enter w_+ and w_- by crossing their boundaries, (ii) $\text{Space}_A\,(E\,(G_1,\, w_0))$-bounded on w_+ and w_-, (iii) of alternating level Π_k or less.

Because w_+ differs from w_- in the single w'_+-block only and w'_+, w'_- are $\text{Space}_A\,(E\,(G,\, w_0))$-equivalent by (c″), we have, for each p satisfying (i), (ii), and (iii), that $\text{Ex}_{p,\, w_+} = \text{Ex}_{p,\, w_-} = \{p_1, \dots, p_h\}$, for some configurations $p_1, \dots, p_h$ leaving w_+ and w_-. By Lemma 10*b*, it is sufficient to show that if $1 = f_{p,\, w_+}(C') > f_{p,\, w_-}(C'') = 0$, for some $C' = (C'_{p_1}, \dots, C'_{p_h})$ and $C'' = (C''_{p_1}, \dots, C''_{p_h})$ representing the accept/reject statuses of exit configurations, then $1 = C'_{p_e} > C''_{p_e} = 0$ for some exit configuration p_e.

Suppose that p is Π_k-accepting on w_+ but Π_k-rejecting on w_-. Because there must exist a rejecting computation path beginning in p on w_-, we have the following cases to consider:

0) The rejecting path leaves w_- making no alternation. Because w'_+, w'_- are $\text{Space}_A\,(E\,(G,\, w_0))$-equivalent and p is $\text{Space}_A\,(E\,(G,\, w_0))$-bounded on w_+ and w_-, we get, by the same reasoning as in Case 0 of Theorem 5, that $1 = C'_{p_e} > C''_{p_e} = 0$ for some exit configuration $p_e \in \text{Ex}_{p,\, w_+} = \text{Ex}_{p,\, w_-}$.

1*a*) The rejecting path from the Π_k-rejecting p alternates outside the critical block on w_-, entering a Σ_{k-1}-rejecting p'. (See *fig.* 15.)

Since p' is also reachable from the Π_k-accepting p on w_+, we have that p' is (i) positioned outside the critical block, (ii) $\text{Space}_A\,(E\,(G,\, w_0))$-bounded on w_+ and w_- (because it is reachable from p), (iii) Σ_{k-1}-accepting on w_+ but Σ_{k-1}-rejecting on w_-.

Because $(w'_+,\, w'_-)$ is a Σ_{k-1}, $\text{Space}_A\,(E\,(G,\, w_0))$-resistant pair, by (d″), we obtain that $f_{p',\, w_+} \leq f_{p',\, w_-}$, using Theorem 5.

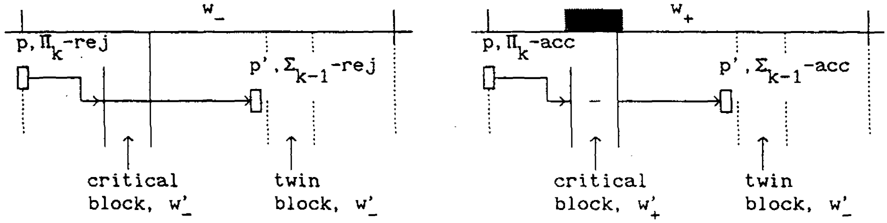

Figure 15

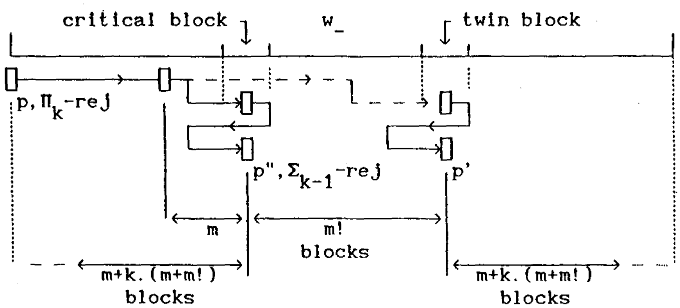

Figure 16

On the other hand, we also have $1 = f_{p', w_+}(C'_{r_1}, \ldots, C'_{r_f}) > f_{p', w_-}(C''_{r_1}, \ldots, C''_{r_f}) = 0$, where $\mathrm{Ex}_{p', w_+} = \mathrm{Ex}_{p', w_-} = \{r_1, \ldots, r_f\} \subseteq \mathrm{Ex}_{p, w_+} = \mathrm{Ex}_{p, w_-} = \{p_1, \ldots, p_h\}$ denote the sets of exits of w_+ and w_- for p' and p, respectively.

By Lemma 10*a*, this is possible only if $1 = C'_{r_e} > C''_{r_e} = 0$, for some $r_e \in \{r_1, \ldots, r_f\} \subseteq \{p_1, \ldots, p_h\}$, *i. e.*, we have a configuration r_e leaving w_+ and w_-, reachable from p via p', that is accepting for w_+ but rejecting for w_-.

1*b*) The rejecting path from p alternates inside the critical block on w_-, where it enters a Σ_{k-1}-rejecting $p'' = \langle q, j \rangle$. (See *fig.* 16.)

Note that both the critical block and its twin, lying $m!$ blocks to the right, are at least $m + k \cdot (m + m!)$ blocks away from either margin of w_-. (See also *fig.* 14.) Among others, this implies that the computation path had to traverse at least m blocks, for p at the left margin, or at least $m + m!$ blocks, for p placed at the right, along the way from p to p''.

Let p' be the configuration having the same memory state as p'', with the input head positioned exactly $m!$ blocks more to the right, *i. e.*, $p' = \langle q, j + (|w'_0| - 1)! \cdot (|w'_-| + 1) \rangle$. Since $m > (c^{\mathrm{Space}_A(E(G, w_0))})^6$, by (10), and p is $\mathrm{Space}_A(E(G, w_0))$-bounded on w_-, we have that p' is also reachable from p, by the use of Theorem 1 and Lemma 8. Moreover, if p'' is Σ_{k-1}-rejecting, then p' must also be Σ_{k-1}-rejecting on w_-, by Theorem 2, because both p'' and p' are $\mathrm{Space}_A(E(G, w_0))$-bounded on w_- (they are reachable from p) and sufficiently far from either margin.

Therefore, for each rejecting path from p that alternates inside the critical block on w_-, there exists another rejecting path that alternates outside the critical block. This reduces Case 1*b* to Case 1*a*.

All remaining cases lead to contradictions and hence cannot happen:

2*a*) If an alternation-free path from p enters an infinite cycle and at least a part of this cycle lies outside the critical block on w_-, then we can find a corresponding infinite cycle that is reachable from the Π_k-accepting p on w_+, by the same argument as in Case 2*a* of Theorem 5, which is a contradiction.

2*b*) If the entire cycle is executed inside the critical block on w_-, then there exists at least one more infinite cycle, reachable from p inside the twin block, by a reasoning very similar to Case 1*b*, using Theorem 1 and Lemma 8. This reduces Case 2*b* to Case 2*a*.

3) The argument for an alternation-free path beginning in the Π_k-rejecting p on w_- that halts and rejects the input is almost the same as for the infinite cycle, giving a contradiction.

This shows $f_{p, w_+} \leq f_{p, w_-}$ for each p of alternating level Π_k and also of Π_{k-1}. For the levels Σ_{k-1} or less, we obtain $f_{p, w_+} \leq f_{p, w_-}$ directly, by Theorem 5 and (d''). This completes the proof of the theorem, since we have just shown that (w_+, w_-) *is a* Π_k, $\mathrm{Space}_A(E(G, w_0))$-*resistant pair.* □

The above theorem has its counterpart describing the relationship between Π_{k-1}- and Σ_k-SPACE$(s(n))$ resistant languages.

THEOREM 7: *Let* L' *be a* Π_{k-1}-SPACE$(s(n))$ *resistant language, for some* $k \geq 3$, *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$. *Then the language*

$$L = \{w \in R_k;\ w = bw_1\, bw_2\, b \ldots bw_{\mathfrak{f}}\, b,\\ \forall j \in \{1, \ldots, \mathfrak{f}\} : w_j \in L',\ w_1, \ldots, w_{\mathfrak{f}} \in R_{k-1}\}$$

is Σ_k-SPACE$(s(n))$ *resistant.*

Proof: The argument is very similar to the proof of Theorem 6, so we point out the main differences only. First, w_+ and w_- are defined by

$$w_+ = b\,(w'_+\, b)^{m+(4k+3)\cdot m!},$$
$$w_- = b\,(w'_+\, b)^{m+2k\cdot m!}\, w'_-\, b\,(w'_+\, b)^{(2\cdot m!-1)+2k\cdot m!+m!},$$

so here $w_+ \in L$ is homogeneous while $w_- \notin L$ contains a single block $w'_- \notin L'$.

Second, we prove that (w_+, w_-) is a Σ_k, $\text{Space}_A(E(G, w_0))$-resistant pair and therefore we consider a configuration p crossing the boundaries of w_+ and w_- that is existential, *i. e.*, Σ_k-accepting on w_+ but Σ_k-rejecting on w_-. Our analysis begins with a successful path starting in the Σ_k-accepting p on the homogeneous w_+, using the fact that the machine cannot distinguish the critical block from its twin, and that all paths starting in the Σ_k-rejecting p on w_- must be rejecting. (In Theorem 6, we considered a rejecting path starting in the Π_k-rejecting p on the homogeneous w_-. Compare, for example, Case 1*b* for Claim 1 and Claim 2 in Theorem 5.) □

It is easy to show that no Σ_k/Π_k-SPACE$(s(n))$ machine is able to recognize a Σ_k/Π_k-SPACE$(s(n))$ resistant language.

THEOREM 8: *a) If L is a* Π_k-SPACE$(s(n))$ *resistant language then* $L \notin \Pi_k$-SPACE$(s(n))$.

b) If L is a Σ_k-SPACE$(s(n))$ *resistant language then* $L \notin \Sigma_k$-SPACE$(s(n))$.

Proof: Let L be a Π_k-SPACE$(s(n))$ resistant language. Then, for each Π_k-SPACE$(s(n))$ machine A, $G = 0$, and $\breve{n} = 2$,

a) there exist $w_+ \in R_k \cap L$, $w_- \in R_k - L$, and $w_0 \in R_k$, such that

b) $|w_0| \geq 2$,

c) w_+, w_-, and w_0 are $\text{Space}_A(w_0)$-equivalent (since $E(0, w_0) = w_0$, by Definition 11),

d) (w_+, w_-) is a Π_k, $\text{Space}_A(w_0)$-resistant pair.

By (*d*) and Theorem 5, for $\alpha = \gg$ and $\beta = \ll$ (where "$\gg$" and "$\ll$" denote the left and right endmarker, respectively), we obtain that $f_{p, \gg w_+ \ll} \leq f_{p, \gg w_- \ll}$ for each configuration p that is (i) positioned outside w_+ and w_-, *i. e.*, on the left or right endmarker, (ii) $\text{Space}_A(w_0)$-bounded on $\gg w_+ \ll$ and on $\gg w_- \ll$, (iii) of alternating level Π_k or less.

The initial configuration p_I of our Π_k-SPACE$(s(n))$ machine satisfies (i) and (iii) automatically. It is not very hard to show that it also satisfies (ii). By Definition 3 and 4, p_I is $\text{Space}_A(w_0)$-bounded on $\gg w_0 \ll$, since $\text{Space}_A(w_0)$ is defined as the maximal amount of space used by any configuration that is reachable from the initial p_I on the string $\gg w_0 \ll$. By (*c*), w_+, w_-, and w_0 are $\text{Space}_A(w_0)$-equivalent and hence, by Lemma 4*a*, p_I is $\text{Space}_A(w_0)$-bounded on $\gg w_+ \ll$ and on $\gg w_- \ll$.

Thus, $f_{p_I, \gg w_+ \ll} \leq f_{p_I, \gg w_- \ll}$. We may assume, without loss of generality, that our machine has been programmed correctly and never tries to move its input head to the left/right from the left/right endmarker, respectively. This implies that $f_{p_I, \gg w_+ \ll}$ and $f_{p_I, \gg w_- \ll}$ are constant functions with the empty formal parameter lists, *i. e.*, we have $f_{p_I, \gg w_+ \ll}(\,) \leq f_{p_I, \gg w_- \ll}(\,)$. But for $w_+ \in L$ and $w_- \notin L$ we need $f_{p_I, \gg w_+ \ll}(\,) = 1$ and $f_{p_I, \gg w_- \ll}(\,) = 0$. Hence, the machine A does not recognize L.

The same argument holds also for Σ_k-SPACE $(s(n))$. □

6. THE HIERARCHY

In this section, we shall give an induction base for the mechanism described in Section 5 by showing some Σ_2/Π_2-SPACE $(s(n))$ resistant languages, which allows us to present languages separating Σ_k-SPACE $(s(n))$ from Π_k-SPACE $(s(n))$, for each $s(n)$ below $\log(n)$ and $k \geq 2$. This yields the infinite hierarchy. Finally, we shall show that Σ_k-SPACE $(s(n))$ is not closed under complement and intersection, similarly, Π_k-SPACE $(s(n))$ is not closed under complement and union. Before doing this, we need to present some Σ_1/Π_1 and Σ_2/Π_2 resistant pairs of strings over a single letter alphabet.

THEOREM 9: *For each $s(n)$ space bounded alternating machine A, each $G \geq 0$, and each $\breve{n} \geq 0$, there exists $\breve{n}' \geq \breve{n}$ such that, for each $n \geq \breve{n}'$,*

a) $a^{\lceil\sqrt{n}\rceil}$ and $a^{\lceil\sqrt{n}\rceil + n!}$ are $\text{Space}_A(E(G, a^n))$*-equivalent,*
a^n and $a^{n+n!}$ are $\text{Space}_A(E(G, a^n))$*-equivalent,*

b) $(a^{\lceil\sqrt{n}\rceil}, a^{\lceil\sqrt{n}\rceil+n!})$ and $(a^{\lceil\sqrt{n}\rceil+n!}, a^{\lceil\sqrt{n}\rceil})$ are Σ_1, $\text{Space}_A(E(G, a^n))$*-resistant as well as Π_1,* $\text{Space}_A(E(G, a^n))$*-resistant pairs,*

c) $(a^{n+n!}, a^n)$ is a Π_2, $\text{Space}_A(E(G, a^n))$*-resistant pair,*

d) $(a^n, a^{n+n!})$ is a Σ_2, $\text{Space}_A(E(G, a^n))$*-resistant pair.*

Proof: (*a*) Define $\mathcal{R} = \lceil\sqrt{n}\rceil$. Using Lemma 6 for $H = 2^G$, find $\breve{n}' \geq \breve{n}$ so that

$$(c^{s(n^{2^G})})^6 < \mathcal{R} < \frac{n}{2} < n, \tag{11}$$

for each $n \geq \breve{n}'$, where c is a machine dependent constant satisfying (2). But then, for each $n \geq \breve{n}'$,

$$(c^{\text{Space}_A(E(G, a^n))})^6 \leq (c^{s(n^{2^G})})^6 < \mathcal{R} < \frac{n}{2} < n, \tag{12}$$

using (1), (3), $|a^n| = n$, and (11). By Lemma 9, this implies that a^n, $a^{n+n!}$ are $\mathrm{Space}_A(E(G, a^n))$-equivalent, and that $a^{\mathcal{R}}$, $a^{\mathcal{R}+i\cdot\mathcal{R}!}$ are $\mathrm{Space}_A(E(G, a^n))$-equivalent, for each $i \geq 0$. Because $n!$ is an integer multiple of $\mathcal{R}!$, we have that $a^{\mathcal{R}}$, $a^{\mathcal{R}+n!}$ are $\mathrm{Space}_A(E(G, a^n))$-equivalent.

(*b*) Let $w' = a^{\mathcal{R}}$ and $w'' = a^{\mathcal{R}+n!}$. We shall show that (w', w'') is a Π_1, $\mathrm{Space}_A(E(G, a^n))$-resistant pair. (All other cases are almost identical, interchanging w' with w'' and/or analyzing existential paths instead of universal.)

Because w', w'' are $\mathrm{Space}_A(E(G, a^n))$-equivalent by (*a*), we have $\mathrm{Ex}_{p,w'} = \mathrm{Ex}_{p,w''}$, for each configuration p that is (i) going to enter w' and w'', (ii) $\mathrm{Space}_A(E(G, a^n))$-bounded on w' and w'', (iii) of alternating level Π_1. It is not too hard to prove that $f_{p,w'} \leq f_{p,w''}$. Again, it is sufficient to show that if p is Π_1-accepting on w', but Π_1-rejecting on w'', then some configuration $p_e \in \mathrm{Ex}_{p,w'} = \mathrm{Ex}_{p,w''}$ must be Π_1-accepting on w', but Π_1-rejecting on w''.

*b*0) If p is Π_1-rejecting on w'' because of a rejecting path that leaves w'' in a Π_1-rejecting configuration $p_e \in \mathrm{Ex}_{p,w''}$, then we are done.

*b*1) The rejecting path started in the Π_1-configuration has no alternations.

*b*2) Suppose that some path from p enters an infinite cycle on $w'' = a^{\mathcal{R}+n!}$, where $\mathcal{R} = \lceil\sqrt{n}\rceil$. Using $M^6 = (c^{\mathrm{Space}_A(E(G,a^n))})^6 < \mathcal{R} < \mathcal{R} + n!$, by (12), we shall find another cycle that never moves the input head farther than $M^3 = (c^{\mathrm{Space}_A(E(G,a^n))})^3$ positions away from p placed at the left/right margin of w''. Since $M^3 \leq M^6 < \mathcal{R}$, we have enough room to enter this cycle from the Π_1-accepting p on $w' = a^{\mathcal{R}}$, which is a contradiction.

The proof is based on the observation that each cycle beginning and ending in the same configuration p_C can be, by Lemma 7, replaced by a cycle from p_C to p_C never moving the head farther than M^2 positions away from p_C. Second, we may then assume that p_C (reachable from p) is at most $j \leq (M^2+1)+(M+1)+M^2$ positions away from p, for, if the computation path from p to p_C gets too far, then we can find two configurations $p_1 = \langle q, j_1\rangle$ and $p_2 = \langle q, j_2\rangle$, having the same memory state q, such that both j_1 and j_2 are at least M^2+1 positions away from p. Using Lemma 8, we can then cut the path from p_1 to p_2 out and shift the cycle from p_C to p_C closer to p. This process can be repeated until we obtain a cycle never moving the head farther than $(M^2+1)+(M+1)+M^2 \leq M^3 \leq M^6 < \mathcal{R}$ positions away from p. (For a more detailed proof, the reader is referred to [9], Theorem 2.)

*b*3) The argument for a path beginning in p on $w'' = a^{\mathcal{R}+n!}$ that halts and rejects the input is very similar to Case *b*2. Again, we can find another path that halts never moving the input head farther than $M^3 = (c^{\text{Space}_A(E(G, a^n))})^3 \le M^6 < \mathcal{R}$ positions away from p, so we have enough room to run this rejecting path from the Π_1-accepting p on $w' = a^{\mathcal{R}}$, which is a contradiction.

This completes the proof of (*b*).

(*c*) We shall show that $(a^{n+n!}, a^n)$ is a Π_2, $\text{Space}_A(E(G, a^n))$-resistant pair. Cases *c*0, *c*2, and *c*3, *i. e.*, moving out, cycle, and halting parallel Cases *b*0, *b*2, and *b*3, respectively. Therefore, they are omitted. We shall now concentrate on Case *c*1, *i. e.*, on alternation.

*c*1*a*) Suppose that p is Π_2-accepting on $a^{n+n!}$. Further, suppose that p is Π_2-rejecting on a^n because some path enters a Σ_1-rejecting p', positioned at least $\mathcal{R} = \lceil\sqrt{n}\,\rceil$ positions away from the left margin of a^n.

Then $a^{n+n!}$ and a^n can be expressed in the form $a^{n+n!} = \alpha\, a^{\mathcal{R}+n!}\,\beta$, $a^n = \alpha\, a^{\mathcal{R}}\,\beta$, where $\alpha = \varepsilon$ and $\beta = a^{n-\mathcal{R}}$. (See *fig.* 17.)

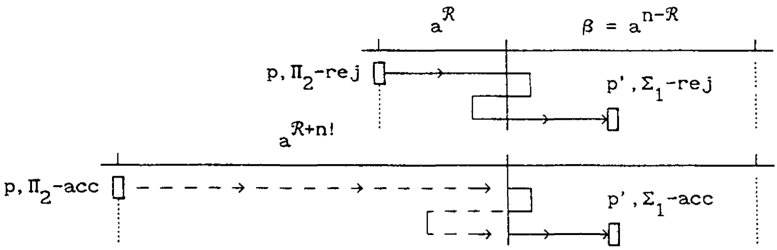

Figure 17

But then p' is (i) positioned on β, *i. e.*, outside $a^{\mathcal{R}+n!}$ and $a^{\mathcal{R}}$ on $\alpha\, a^{\mathcal{R}+n!}\,\beta$ and $\alpha\, a^{\mathcal{R}}\,\beta$, respectively, (ii) $\text{Space}_A(E(G, a^n))$-bounded on $\alpha\, a^{\mathcal{R}+n!}\beta$ and $\alpha\, a^{\mathcal{R}}\,\beta$ (because it is reachable from p), (iii) Σ_1-accepting on $\alpha\, a^{\mathcal{R}+n!}\,\beta$ but Σ_1-rejecting on $\alpha\, a^{\mathcal{R}}\,\beta$. (The head positions are relative to the left margin of β.)

Because $(a^{\mathcal{R}+n!}, a^{\mathcal{R}})$ is a Σ_1, $\text{Space}_A(E(G, a^n))$-resistant pair, by (*b*), this is possible only if there exists p_e leaving $\alpha\, a^{\mathcal{R}+n!}\,\beta$ and $\alpha\, a^{\mathcal{R}}\,\beta$ that is Σ_1-accepting for $\alpha\, a^{\mathcal{R}+n!}\,\beta$ but Σ_1-rejecting for $\alpha\, a^{\mathcal{R}}\,\beta$, by the use of Theorem 5 and Lemma 10a. (*Cf.* also Case 1*a* in Theorem 6.)

*c*1*b*) If the rejecting path from the Π_2-rejecting p on a^n alternates closer than $\mathcal{R} = \lceil\sqrt{n}\rceil$ positions to the left margin, then it alternates farther than $\mathcal{R}$ positions away from the right margin, since $\mathcal{R} < n/2$, by (12). Then the same argument can be used for $a^{n+n!}$ and a^n partitioned into $a^{n+n!} = \alpha\, a^{\mathcal{R}+n!}\, \beta$, $a^n = \alpha\, a^{\mathcal{R}}\, \beta$, with $\alpha = a^{n-\mathcal{R}}$, $\beta = \varepsilon$, and p' positioned on α.

This completes the proof of (*c*). The converse does not hold, *i. e.*, $(a^n, a^{n+n!})$ is not necessarily a Π_2, $\text{Space}_A(E(G, a^n))$-resistant pair, since a rejecting path from the Π_2-rejecting p on $a^{n+n!}$ may alternate in p' positioned in the middle of $a^{n+n!}$ so the segment of length $n!$ is neither to the left, nor the right of p'. However, the converse does hold for Σ_2-resistance:

(*d*) The proof that $(a^n, a^{n+n!})$ is a Σ_2, $\text{Space}_A(E(G, a^n))$-resistant pair is very similar to (*c*). Here we suppose that p is Σ_2-accepting on a^n but Σ_2-rejecting on $a^{n+n!}$. Hence, the analysis begins from the accepting computation path started in p on a^n, *i. e.*, on the shorter string again. □

We are now ready to present the languages that separate Σ_k-SPACE $(s(n))$ from Π_k-SPACE $(s(n))$.

DEFINITION 13: Let

$$f(n) = \text{ the first number that does not divide } n.$$

Then

$$S_2 = \{a^n;\ f(n) \le \max\{f(1), \ldots, f(n-1)\},\ n \ge 1\},$$

$$P_2 = \{a^n;\ f(n) > \max\{f(1), \ldots, f(n-1)\},\ n \ge 1\},$$

$$S_k = \{w \in R_k;\ w = bw_1\, bw_2\, b \ldots bw_{\mathfrak{f}}\, b,\ \exists\, j \in \{1, \ldots, \mathfrak{f}\} : w_j \in P_{k-1},\ w_1, \ldots, w_{\mathfrak{f}} \in R_{k-1}\},$$

$$P_k = \{w \in R_k;\ w = bw_1\, bw_2\, b \ldots bw_{\mathfrak{f}}\, b,\ \forall\, j \in \{1, \ldots, \mathfrak{f}\} : w_j \in S_{k-1},\ w_1, \ldots, w_{\mathfrak{f}} \in R_{k-1}\},$$

for each $k \ge 3$.

LEMMA 13: *$f(n)$ is unbounded,* i. e., *for each $h \ge 0$ there exists $n \ge 0$ such that $f(n) > h$, and $f(n) = f(n + n!)$, for each $n \ge 2$.*

Proof: Since $h!$ is divisible by each $j \le h$, we have $f(h!) > h$. Clearly, $f(2) = f(2 + 2!)$. For each $n \ge 3$, $n - 1$ does not divide n. Thus, the first "nondivisor" of n is at most $n - 1$, *i. e.*, $f(n) \in \{1, \ldots, n-1\}$. Therefore, it is sufficient to show that $j \in \{1, \ldots, n-1\}$ divides n if and only if it divides $n + n!$

(i) If j divides n, then it divides also $n + n!$, since $n + n!$ is an integer multiple of n.

(ii) Suppose that j divides $n + n!$, *i. e.*, $n + n! = j \cdot l_1$, for some integer $l_1 \geq 1$. But $j \leq n - 1$ must also divide $n!$, *i. e.*, $n! = j \cdot l_2$, for some $l_2 < l_1$. This gives $n = (n + n!) - n! = j \cdot (l_1 - l_2)$, *i. e.*, j divides n. □

S_2 and P_2 are simplified versions of the languages that were used to separate Σ_2-SPACE$(s(n))$ from Π_2-SPACE$(s(n))$ in [9]. We shall now prove a stronger statement, namely, their space resistance.

THEOREM 10: *For each $k \geq 2$ and each $s(n)$ with $\sup_{n \to \infty} s(n)/\log(n) = 0$, P_k is Σ_k-SPACE$(s(n))$ resistant and S_k is Π_k-SPACE$(s(n))$ resistant.*

Proof: First, we shall show that P_2 is Σ_2-SPACE$(s(n))$ resistant. Let A be an $s(n)$ space bounded alternating machine, $G \geq 0$, and $\breve{n} \geq 0$. By Theorem 9 and Lemma 13, we can find $\breve{n}' \geq \max\{\breve{n}, 2\}$ so that, for each $n \geq \breve{n}'$,

a) a^n and $a^{n+n!}$ are $\text{Space}_A(E(G, a^n))$-equivalent,

b) $(a^n, a^{n+n!})$ is a Σ_2, $\text{Space}_A(E(G, a^n))$-resistant pair,

c) $(a^{n+n!}, a^n)$ is a Π_2, $\text{Space}_A(E(G, a^n))$-resistant pair,

d) $f(n) = f(n + n!)$.

We need to find $n \geq \breve{n}'$ so that $a^n \in P_2$ but $a^{n+n!} \notin P_2$. By Lemma 13, we can find *minimal* N satisfying $f(n) > \max\{f(1), \ldots, f(\breve{n}')\}$, *i. e.*,

$$f(N) > \max\{f(1), \ldots, f(\breve{n}')\}, \quad \text{but}$$
$$f(n) \leq \max\{f(1), \ldots, f(\breve{n}')\}, \quad \text{for each } n < N.$$

This gives $f(N) > f(n)$, for each $n < N$, and therefore $f(N) > \max\{f(1), \ldots, f(N-1)\}$, *i. e.*, $a^N \in P_2$. Note that we have also $N > \breve{n}' \geq \breve{n}$, since $f(j) \leq \max\{f(1), \ldots, f(j), \ldots, f(\breve{n}')\}$, for each $j \leq \breve{n}'$. On the other hand, $f(N + N!) = f(N)$, and hence $f(N + N!) \leq \max\{f(1), \ldots, f(N), \ldots, f(N + N! - 1)\}$, *i. e.*, $a^{N+N!} \notin P_2$. Now, it is easy to see that $w_+ = a^N$, $w_- = a^{N+N!}$, and $w_0 = a^N$ satisfy

(i) $w_+ \in R_2 \cap P_2$, $w_- \in R_2 - P_2$, and $w_0 \in R_2$,

(ii) $|w_0| \geq \breve{n}$,

(iii) w_+, w_-, and w_0 are $\text{Space}_A(E(G, w_0))$-equivalent, by (*a*),

(iv) (w_+, w_-) is a Σ_2, $\text{Space}_A(E(G, w_0))$-resistant pair, by (*b*),

i. e., the language P_2 is Σ_2-SPACE$(s(n))$ resistant.

We also get that the language S_2 is Π_2-SPACE$(s(n))$ resistant, since $a^{N+N!} \in S_2$, $a^N \notin S_2$, and $(a^{N+N!}, a^N)$ is a Π_2, $\text{Space}_A(E(G, a^N))$-resistant pair, by (*c*), using the same argument for $\bar{w}_+ = a^{N+N!}$, $\bar{w}_- = a^N$, and $\bar{w}_0 = a^N$.

By a straightforward induction on k, using Theorems 6 and 7, we obtain that P_k is Σ_k-SPACE$(s(n))$ resistant and S_k is Π_k-SPACE$(s(n))$ resistant, for each $k \geq 2$. □

The above result implies immediately, by Theorem 8, that $P_k \notin \Sigma_k$-SPACE$(s(n))$ and $S_k \notin \Pi_k$-SPACE$(s(n))$, for no $s(n)$ below $\log(n)$. Changing the initial alternation level, using a method described by Szepietowski in [23], we can easily design $O(\log\log(n))$ space bounded machines for P_k and S_k.

THEOREM 11: *$P_k \in \Pi_k$-SPACE$(\log\log(n))$ and $S_k \in \Sigma_k$-SPACE$(\log\log(n))$, for each $k \geq 2$.*

Proof: First, we shall show that $P_2 \in \Pi_2$-SPACE$(\log\log(n))$. Our machine first deterministically computes $f(n)$, checking if n is divisible by j, for $j = 2, 3, 4\ldots$ until it finds the first nondivisor of n. Then, branching universally, the machine moves along the input tape and, at each position $h < n$, verifies if $f(h) < f(n)$. We do not have to compute the first nondivisor of h exactly, it is sufficient, branching existentially, to find $g \in \{1, \ldots, f(n) - 1\}$ and verify that this g does not divide h.

Note that we store j, $f(n)$, and g on the worktape, but not h. Since $\log(f(n)) \in O(\log\log(n))$, this much space is sufficient. (For proof, *see* e. g. [18].)

The Σ_2-SPACE$(\log\log(n))$ machine for S_2 is very similar. Having computed $f(n)$, find existentially $h < n$ with $f(h) \geq f(n)$ and, branching universally, verify that each $g \in \{1, \ldots, f(n) - 1\}$ divides h.

Now we can show that $P_k \in \Pi_k$-SPACE$(\log\log(n))$, for each $k \geq 2$. The machine first checks if the input $w \in R_k$. If yes, then $w = bw_1\, bw_2\, b \ldots bw_{\mathfrak{f}}\, b$, for some $w_1, \ldots, w_{\mathfrak{f}} \in R_{k-1}$. In addition, this partition is unique and determined by the positions of substrings $ab^{2k-5}a$ in w. Thus, branching universally at each $ab^{2k-5}a$, verify if $w_j \in S_{k-1}$, for each $j \in \{1, \ldots, \mathfrak{f}\}$. This is done as follows. Each $w_j \in R_{k-1}$ can be uniquely partitioned into the strings in R_{k-2}, their boundaries are determined by the positions of substrings $ab^{2(k-1)-5}a$. Thus, branching existentially at each $ab^{2(k-1)-5}a$, find a segment that is in $P_{k-2}\ldots$ Finally, at the lowest level, check if the tape segment $u \in R_2 = a^+$, enclosed in b's, is in P_2 (for k even), or in S_2 (for k odd), using the algorithm described above.

The initial checking for $w \in R_k$ as well as searching for the segment boundaries, level by level, can be done in constant space, using the finite state control. The worktape is needed at the lowest level only, to check if some $u \in P_2/S_2$. The space used is then bounded by $O(\log\log(|u|))$, for some $|u| \leq n$, *i. e.*, by $O(\log\log(n))$.

Similarly, $S_k \in \Sigma_k$-SPACE$(\log\log(n))$, for each $k \geq 2$. The only difference is that the topmost level branching is existential. □

COROLLARY 1: *For each* $k \geq 2$ *and each* $s(n)$ *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$,

$$\Sigma_k\text{-SPACE}(\log\log(n)) - \Pi_k\text{-SPACE}(s(n)) \neq \emptyset,$$

and also,

$$\Pi_k\text{-SPACE}(\log\log(n)) - \Sigma_k\text{-SPACE}(s(n)) \neq \emptyset.$$

Moreover, it is obvious that Σ_i/Π_i-SPACE$(s(n)) \subseteq \Sigma_{i+1}/\Pi_{i+1}$-SPACE$(s(n))$, for each $i \geq 1$. From this we have:

COROLLARY 2: *For each* $k \geq 2$ *and each* $s(n)$ *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$,

$$\Sigma_k\text{-SPACE}(\log\log(n)) - \Sigma_{k-1}\text{-SPACE}(s(n)) \neq \emptyset,$$

$$\Sigma_k\text{-SPACE}(\log\log(n)) - \Pi_{k-1}\text{-SPACE}(s(n)) \neq \emptyset,$$

$$\Pi_k\text{-SPACE}(\log\log(n)) - \Sigma_{k-1}\text{-SPACE}(s(n)) \neq \emptyset,$$

$$\Pi_k\text{-SPACE}(\log\log(n)) - \Pi_{k-1}\text{-SPACE}(s(n)) \neq \emptyset.$$

That is, the alternating space hierarchy does not collapse between $\log\log(n)$ and $\log(n)$:

COROLLARY 3: *For each* $k \geq 2$ *and* $s(n) \geq \log\log(n)$ *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$,

$$\Sigma_{k-1}\text{-SPACE}(s(n)) \subsetneq \Sigma_k\text{-SPACE}(s(n)),$$
$$\Sigma_{k-1}\text{-SPACE}(s(n)) \subsetneq \Pi_k\text{-SPACE}(s(n)),$$

$$\Pi_{k-1}\text{-SPACE}(s(n)) \subsetneq \Sigma_k\text{-SPACE}(s(n)),$$
$$\Pi_{k-1}\text{-SPACE}(s(n)) \subsetneq \Pi_k\text{-SPACE}(s(n)).$$

COROLLARY 4: *For each* $k \geq 2$ *and* $s(n) \geq \log\log(n)$ *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$, Σ_k-SPACE $(s(n))$ *and* Π_k-SPACE $(s(n))$ *are not closed under complement.*

Proof: It is easy to show, by induction on k, that $S_k = R_k - P_k$ and $P_k = R_k - S_k$, for each $k \geq 2$. Should Σ_k-SPACE $(s(n))$ be closed under complement for some $k \geq 2$ and some $s(n)$ above $\log\log(n)$, we have $S_k^c = R_k^c \cup P_k \in \Sigma_k$-SPACE $(s(n))$. Since Σ_k-SPACE $(s(n))$ is closed under intersection with regular sets, $(R_k^c \cup P_k) \cap R_k = P_k \in \Sigma_k$-SPACE $(s(n))$, using $P_k \subseteq R_k$. But this is a contradiction for space bounds below $\log(n)$. The argument for Π_k-SPACE $(s(n))$ is almost the same. □

The tools presented above allow us to draw some further consequences:

THEOREM 12: *For each* $k \geq 2$ *and* $s(n) \geq \log\log(n)$ *with* $\sup_{n\to\infty} s(n)/\log(n) = 0$, Σ_k-SPACE $(s(n))$ *is not closed under intersection and* Π_k-SPACE $(s(n))$ *is not closed under union.*

Proof: Suppose that Π_k-SPACE $(s(n))$ is closed under union, for some $k \geq 2$ and some $s(n) \geq \log\log(n)$. Since $P_k \in \Pi_k$-SPACE $(\log\log(n))$ and R_k is regular, $P_k\$R_k$, $R_k\$P_k \in \Pi_k$-SPACE $(\log\log(n))$, where \$ denotes a new symbol. Using the union hypothesis, we have a Π_k-SPACE $(s(n))$ machine A recognizing $L = \{w_1\$w_2 \in R_k\$R_k;\ w_1 \in P_k$ or $w_2 \in P_k\}$. We can now easily replace A by a new Π_k-SPACE $(s(n))$ machine A' recognizing $L' = P_k \cup L$, *not using* the union hypothesis: First, A' checks whether the symbol \$ is present on the input tape. If yes, use A to determine if the input $w \in L$. If no, then simulate A imitating that the input string is $w\$w$. The only thing we have to remember, within the finite state control, is whether the input head is positioned on the first or on the second copy of w. If A reaches the right endmarker (on the first copy of w), interrupt the simulation, move the head to the left endmarker, and pretend that \$ has been crossed from left to right. Then carry on the second (nonexistent) copy of w. If A moves back to the left endmarker, imitate crossing \$ from right to left. Clearly, A' uses exactly the same amount of space on the inputs w and $w\$w$, for each $w \in \{a, b\}^*$, *i. e.*, $\text{Space}_{A'}(w) = \text{Space}_{A'}(w\$w)$.

Because P_k is a Σ_k-SPACE $(s(n))$ resistant language for each $s(n)$ below $\log(n)$, we have, using A', $G = 0$, and $\breve{n} = 2$, some strings $w_+ \in P_k$, $w_- \notin P_k$, and $w_0 \in R_k$ such that w_+, w_-, and w_0 are $\text{Space}_{A'}(w_0)$-equivalent and (w_+, w_-) is a Σ_k, $\text{Space}_{A'}(w_0)$-resistant pair. Now, consider the inputs $w_0\$w_0$, $w_-\$w_-$, $w_-\$w_+$, and $w_+\$w_-$. They are all

$\mathrm{Space}_{A'}$ (w_0)-equivalent, by Lemma 3. Since the initial configuration p_I is trivially $\mathrm{Space}_{A'}$ (w_0)-bounded on the string $\gg w_0 \ll$, we have also that p_I is $\mathrm{Space}_{A'}$ (w_0)-bounded on $\gg w_0\$w_0 \ll$. This follows from $\mathrm{Space}_{A'}$ $(w_0) = \mathrm{Space}_{A'}$ $(w_0\$w_0)$. By Lemma 4, p_I is then $\mathrm{Space}_{A'}$ (w_0)-bounded on $\gg w_-\$w_- \ll$, $\gg w_-\$w_+ \ll$, and $\gg w_+\$w_- \ll$.

Since $w_- \notin P_k$, the input $w_-\$w_-$ must be rejected by the Π_k-SPACE $(s(n))$ machine A' and therefore there must exist a rejecting computation path beginning in p_I on $\gg w_-\$w_- \ll$. Suppose, for example, that this computation path alternates outside the first w_-.

Then we have a Σ_{k-1}-rejecting configuration p that is (i) placed outside the first w_- on $\gg w_-\$w_- \ll$. But p is also reachable from p_I on $\gg w_+\$w_- \ll$, where it is placed outside w_+, since w_+, w_- are $\mathrm{Space}_{A'}$ (w_0)-equivalent and p_I is $\mathrm{Space}_{A'}$ (w_0)-bounded on $\gg w_-\$w_- \ll$ and on $\gg w_+\$w_- \ll$.

Clearly, (ii) p is $\mathrm{Space}_{A'}$ (w_0)-bounded on $\gg w_-\$w_- \ll$ and on $\gg w_+\$w_- \ll$ (it is reachable from p_I), and (iii) it is of alternating level Σ_{k-1}.

Because (w_+, w_-) is a Σ_k, $\mathrm{Space}_{A'}$ (w_0)-resistant pair, we get, using Theorem 5 for $\alpha = \gg$ and $\beta = \$w_- \ll$, that $f_{p, \gg w_+\$w_- \ll}(\,) \leq f_{p, \gg w_-\$w_- \ll}(\,)$. On the other hand, p is Σ_{k-1}-rejecting on $\gg w_-\$w_- \ll$, by assumption, but Σ_{k-1}-accepting on $\gg w_+\$w_- \ll$, since $w_+ \in P_k$ (hence, $w_+\$w_- \in L'$), p is reachable from p_I on $\gg w_+\$w_- \ll$, and all alternation-free paths from the Π_k-accepting p_I on $\gg w_+\$w_- \ll$ must be successful. This gives $f_{p, \gg w_+\$w_- \ll}(\,) = 1$ and $f_{p, \gg w_-\$w_- \ll}(\,) = 0$, which is a contradiction.

If the rejecting path beginning in p_I on $\gg w_-\$w_- \ll$ alternates inside the first w_-, then it alternates outside the second w_- and we can use almost the same argument for $\gg w_-\$w_+ \ll$. All other cases, *i. e.*, an infinite cycle or halting are also very similar and therefore they are omitted.

The corresponding proof showing that Σ_k-SPACE $(s(n))$ is not closed under intersection uses the language $S_k \cup S_k\$S_k$. $\square$

In general, though $P_k \in \Pi_k$-SPACE (log log (n)), we cannot check the input $\$w_1\$w_2\$\ldots\$w_\mathfrak{f}\$$ for any logical relation other than $w_1 \in P_k \,\&\ldots\&\, w_\mathfrak{f} \in P_k$ not using a different alternation level or at least log (n) space. However, if $\mathfrak{f}$ is a fixed constant, then Σ_{k+2}/Π_{k+2}-SPACE (log log (n)) is sufficient, because any relation can be put into the disjunctive/conjuctive normal form and the complement of P_k is in Σ_k-SPACE (log log (n)). The same holds for $S_k \in \Sigma_k$-SPACE (log log (n)) and the relation $w_1 \in S_k \vee \ldots \vee w_\mathfrak{f} \in S_k$.

Some important problems remain open, namely, the relations among DSPACE $(s(n))$ = Σ_0-SPACE $(s(n))$ = Π_0-SPACE $(s(n))$, NSPACE $(s(n))$ = Σ_1-NSPACE $(s(n))$, and Π_1-SPACE $(s(n))$. The partial answer for the tally sets has been achieved, *i. e.*, Σ_1-SPACE $(s(n)) \cap a^* = \Pi_1$-SPACE $(s(n)) \cap a^*$ for each $s(n)$, independent of whether $s(n)$ is above $\log(n)$ or space constructible [10]. Quite surprisingly, this does not imply that the hierarchy collapses to Σ_1 on the tally sets, since Σ_2-SPACE $(s(n)) \cap a^* \neq \Pi_2$-SPACE $(s(n)) \cap a^*$ ([9] or [this paper]). The problem DSPACE $(s(n))$ versus NSPACE $(s(n))$ is also open for the superlogarithmic case.

ACKNOWLEDGEMENTS

The author thanks Branislav Rovan and Robert Szelepcsényi for several helpful remarks concerning this work.

REFERENCES

1. L. Babai and S. Moran, Arthur-Merlin games: A randomized proof-system, and a hierarchy of complexity classes, *J. Comput. Syst. Sciences*, 1988, *36*, pp. 254-276.
2. T. Baker, J. Gill and R. Solovay, Relativizations of the $P = ?NP$ question, *SIAM J. Comput.*, 1975, *4(4)*, pp. 431-442.
3. B. von Braunmuhl, Alternation for two-way machines with sublogarithmic space, to appear in *Proc. of STACS'93.*
4. B. von Braunmuhl, R. Gengler and R. Rettinger, *The alternation hierarchy for machines with sublogarithmic space is infinite*, Research Report 8589–CS, University of Bonn, 1993.
5. A. K. Chandra, D. Kozen and L. J. Stockmeyer, Alternation, *J. Assoc. Comput. Mach.*, 1981, *28*, pp. 114-133.
6. J. H. Chang, O. H. Ibarra, B. Ravikumar and L. Berman, Some observations concerning alternating Turing machines using small space, *Inform. Process. Lett.*, 1987, *25*, pp. 1-9 (Erratum: *27*, 1988, 53).
7. A. R. Freedman and R. E. Ladner, Space bounds for processing contentless inputs, *J. Comput. Syst. Sciences*, 1975, *11*, pp. 118-128.
8. V. Geffert, Nondeterministic computations in sublogarithmic space and space constructibility, *SIAM J. Comput.*, 1991, *20(3)*, pp. 484-498.
9. V. Geffert, Sublogarithmic Σ_2-SPACE is not closed under complement and other separation results, *RAIRO Theoretical Informatics and Applications*, 1993, *27(4)*, pp. 349-366.
10. V. Geffert, Tally versions of the Savitch and Immerman-Szelepcsényi theorems for sublogarithmic space, *SIAM J. Comput.*, 1993, *22(1)*, pp. 102-113.
11. J. Hartmanis, The collapsing hierarchies, *EATCS Bulletin*, 1987, *33*, pp. 26-39.
12. L. A. Hemachandra, The strong exponential hierarchy collapses, *Proc. of 19th ACM STOC Conference*, 1987, pp. 110-122.

13. N. Immerman, Nondeterministic space is closed under complement, *SIAM J. Comput.*, 1988, *17*, pp. 935-938.
14. K. Iwama, ASPACE (o (log log)) is regular, *SIAM J. Comput.*, 1983, *22(1)*, pp. 136-146.
15. B. Jenner, B. Kirsig and K. Lange, The logarithmic alternation hierarchy collapses: $A\,\Sigma_2^L = A\,\Pi_2^L$, *Information and Computation,* 1989, *80,* pp. 269-288.
16. M. Liśkiewicz and R. Reischuk, Separating the lower levels of the sublogarithmic space hierarchy, to appear in *Proc. of STACS'93.*
17. M. Liśkiewicz and R. Reischuk, *The sublogarithmic space hierarchy is infinite*, Technical Report, Technical University of Darmstadt, 1993.
18. D. Ranjan, R. Chang and J. Hartmanis, Space bounded computations: Review and new separation results, *Theoret. Comp. Sci.*, 1991, *80*, pp. 289-302.
19. U. Schöning and K. Wagner, Collapsing oracle hierarchies, census functions, and logarithmically many queries, *Proc. of STACS'88*, Springer-Verlag, LNCS, 294, 1988, pp. 91-97.
20. J. Seiferas, *A note on notions of tape constructibility*, Technical Report CSD-TR-187, Pennsylvania State University, 1976.
21. R. E. Stearns, J. Hartmanis and P. M. Lewis II, Hierarchies of memory limited computations, *Proc. of 1965 IEEE Conference Record on Switching Circuit Theory and Logical Design*, 1965, pp. 179-190.
22. R. Szelepcsényi, The method of forced enumeration for nondeterministic automata, *Acta Informatica*, 1988, *26*, pp. 279-284.
23. A. Szepietowski, Some remarks on the alternating hierarchy and closure under complement for sublogarithmic space, *Information Processing Letters*, 1989, *33*, pp. 73-78.
24. S. Toda, Σ_2-SPACE (n) is closed under complement, *J. Comput. Syst. Sciences*, 1987, *35*, pp. 145-152.
25. A. Yao, Separating the polynomial time hierarchy by oracles, *Proc. of 26th FOCS*, 1985, pp. 1-10.