# Closure Property of Probabilistic Turing Machines and Alternating Turing Machines with Subalgorithmic Spaces

by
Géza Horváth
Katsushi Inoue
Akira Ito
and
Yue Wang

**Department of Computer Science and Systems Engineering**
**Faculty of Engineering**
**Yamaguchi University**
**Ube, 755-8611 Japan**

**Mailing Address:**
**Katsushi Inoue**
**Department of Computer Science and Systems Engineering**
**Faculty of Engineering**
**Yamaguchi University**
**Ube, 755-8611 Japan**
**Email: inoue@csse.yamaguchi-u.ac.jp**

## 1. Introduction

Freivalds [4] showed a surprising result of the language $\{a^n b^n | n \geq 1\}$ being recognized by a two-way Monte Carlo finite automaton (i.e., a two-way probabilistic finite automaton with error probability less than 1/2). This result influenced many subsequent papers [3,8,14]. As far as we know, it is unknown whether the classes of languages recognized by $o(\log n)$ space bounded two-way Monte Carlo Turing machines [5] and two-way probabilistic Turing machines [7] are closed under concatenation, Kleene closure, and length-preserving homomorphism. By using an adaptation of the proof of the above result, and a separation result by Frievald and Karpinski [5], Section 3 of this paper shows that (1) the class of languages recognized by $o(\log n)$ space-bounded two-way Monte-Carlo Turing machines is not closed under these operations, and (2) the class of languages recognized by $o(\log \log n)$ space-bounded two-way unbounded error probabilistic Turing machines is not closed under these operations.

Many investigations of alternating Turing machines (aTm's) with subalgorithmic spaces have been made [1,2,6,10,12,13]. Chang, Ibarra and Ravikumar [2] showed that the language $\{0^n 1 0^n | n \geq 1\}$ can be accepted by a weakly log log $n$ space-bounded one-way aTm. Ito, Inoue, and Takanami [10] showed that there exists a language accepted by a strongly log log $n$ space-bounded two-way aTm, but not accepted by any weakly $o(\log n)$ space-bounded one-way aTm. Iwama [12] showed that the languages accepted by weakly $o(\log \log n)$ space-bounded two-way aTm's are regular. Furthermore, Braunmühl, Genger and Rettinger [1], Geffert [6], and Liśkiewicz and Reischuk [13] showed that the alternation hierarchy for aTm's with space bounds between log log $n$ and log $n$ is infinite.

Section 4 of this paper answers an open question [10] of whether the class of languages accepted by S($n$) space-bounded two-way aTm's is closed under concatenation, Kleene closure, and length-preserving homomorphism for $\log \log n \leq$ S($n$)$\leq o(\log n)$, and shows that the class mentioned above is not closed under these operations.

## 2. Preliminaries

For each word $w$, $|w|$ denotes the length of $w$, and for each set T, |T| denotes the number of elements of T. See [9] for undefined terms.

A *two-way probabilistic Turing machine* we consider here has a read-only input tape delimited by the left endmarker ₵ and the right endmaker \$, and a semi-infinite read-write work tape, initially blank. Of course, the input head of the machine can move left or right. See [7] for the definitions of this machine. As in Freivalds and Karpinski [5], we distinguish between two types of two-way probabilistic Turing machines: *Two-way Monte Carlo Turing machines* and *two-way unbounded error probabilistic Turing machines.*

We say that a two-way Monte Carlo Turing machine M recognizes language L in space S($n$) if there is a positive constant $\varepsilon$ such that:

(1) for any $x \in$L, the probability of the event "M accepts $x$
in space not exceeding S($|x|$)" exceeds $\frac{1}{2} + \varepsilon$, and
(2) for any $x \notin$L, the probability of the event "M rejects $x$
in space not exceeding S($|x|$)" exceeds $\frac{1}{2} + \varepsilon$.

We say that a two-way unbounded error probabilistic Turing machine M recognizes language L in space S($n$) if:
(1) for any $x \in$L, the probability of the event "M accepts $x$
in space not exceeding S($|x|$)" exceeds $\frac{1}{2}$, and
(2) for any $x \notin$L, the probability of the event "M rejects $x$
in space not exceeding S($|x|$)" exceeds $\frac{1}{2}$.

Let MSPACE(S($n$)) (resp., PSPACE(S($n$))) denote the class of languages recognized by two-way Monte Carlo Turing machines (resp., two-way unbounder error probabilistic Turing machines) in space S($n$).

A *two-way alternating Turing machine* (2aTm) we consider here has a read-only input tape delimited by the left endmaker ¢ and the right endmaker \$, an input head which can move left or right on the input tape, and a semi-infinite read-wrie work tape, initially blank. See [1,2,6,10,12,13] for the definition of 2aTm's.

We can view the computation of a 2aTm M as a tree whose nodes are labeled by configurations. A *configuration* of M is of the form $(i, (q, \gamma, k))$, where $i$ is the input tape head position, and *component*$(q, \gamma, k)$ represents the state of the finite control, the non-blank contents of the work tape, and the work tape head position. If $q$ is the state associated with configuration $c$, then $c$ is said to be a *universal* (resp., *existential, accepting*) configuration if $q$ is universal (resp., existential, accepting) state. The *initial configuration* of M is $I_M$=(0,$(q_0, \lambda, 1)$), where $q_0$ is the initial state of M and $\lambda$ is the null string. A *computation tree* of M on input $w$ is a tree such that the root is labeled by $I_M$ and the children of any nonleaf node labeled by a universal (resp., existential) configuration include all (resp., one) of the immediate successors (of M on $w$) of that configuration. A computation tree is *accepting* if it is finite and all the leaves are labeled by accepting configurations. M *accepts* an input $w$ if there is an accepting computation tree of M on $w$.

Let $l$ be a non-negative integer and $c = (i, (q, \gamma, k))$ be a configuration of M. $c$ is $l$ *space-bounded* if $|\gamma| \leq l$.

A computation tree of M (on some input) is $l$ *space-bounded* if each node of the tree is labeled by a $l$ space-bunded configuration of M.

Let S($n$): $N \rightarrow N \cup \{0\}$ be a function, where $N$ denotes the set of all the positive integers. M is *weakly S(n) space-bounded* if for every input $w$ of length $n$, $n \geq 1$, that is accepted by M, there exists an S($n$) space-bounded accepting computation tree of M on $w$. M is *strongly S(n) space-bounded* if for every input $w$ of length $n$ (accepted by M or not), $n \geq 1$, any computation tree of M on $w$ is S($n$) space-bounded.

Let weak-ASPACE(S($n$)) (resp., strong-ASPACE(S($n$))) denote the class of languages accepted by weakly(resp., strongly) S($n$) space-bounded 2aTm's.

## 3. Closure Property of Probabilistic Turing Machines

This section shows that MSPACE($o(\log n)$) and PSPACE($o(\log\log n)$) are not closed under concatenation, Kleene closure, and length preserving homomorphism. The following two lemmas (which were given by Freivalds and Karpinski [5]) are used to get our desired result.

**Lemma 3.1.** Let A,B$\subseteq \Sigma^*$ with A$\cap$B$= \emptyset$ (empty set). Suppose that there are an infinite set I of positive integers, and functions G($n$), H($n$) such that G($n$) is a fixed polynomial in $n$, and for each $n \in$I, there is a set W($n$) of words in $\Sigma^*$ such that:
(1) $|w| \leq$G($n$) for each word $w \in$W($n$),
(2) there is a constant $c > 1$ such that $|$W($n$)$| \geq c^n$ for each $n \in$I, and
(3) for every $n \in$I and every $w, w' \in$W($n$) with $w \neq w'$, there are words $u, v \in \Sigma^*$ such that:
(a) $|uwv| \leq$H($n$), $|uw'v| \leq$H($n$), and
(b)

$$\text{either } \begin{cases} uwv \in \mathrm{A} \\ uw'v \in \mathrm{B} \end{cases} \text{ or } \begin{cases} uwv \in \mathrm{B} \\ uw'v \in \mathrm{A}. \end{cases}$$

Then, if a two-way Monte Carlo Turing machine with space bound S($n$) separates A and B, then S(H($n$)) cannot be $o(\log n)$.

**Lemma 3.2.** Let A,B$\subseteq \Sigma^*$ with A$\cap$B$= \emptyset$. Suppose that there is an infinite set I of positive integers and a function H($n$) such that for each $n \in$I, there is an ordered set of pairs of words W($n$)=$\{(u_1, v_1), (u_2, v_2), ..., (u_n, v_n)\}$ such that for every string $\gamma(1)\gamma(2)...\gamma(n) \in \{0,1\}^n$, there is a word $w$ such that

$$\begin{cases} u_i w v_i \in \mathrm{A}, \text{ if } \gamma(i) = 1, \\ u_i w v_i \in \mathrm{B}, \text{ if } \gamma(i) = 0, \end{cases}$$

and $|u_i w v_i| \leq$H($n$) for all $i \in \{1, 2, ..., n\}$. Then, if a two-way unbounded error probabilistic Turing machine with space bound S($n$) separates A and B, then S(H($n$)) cannot be $o(\log\log n)$.

The following lemma is a key one.

**Lemma 3.3.** Let
$\mathrm{L}_1 = \{a^{m_1}1a^{m_2}1...1a^{m_k} \mid k \geq 2 \;\&\; \forall i(1 \leq i \leq k)\, [m_i \geq 1] \;\&\; m_1 = m_k\}$,
$\mathrm{L}_2 = \{1a^m \mid m \geq 1\}^*$,
$\mathrm{L}_3 = \{a^{m_1}1a^{m_2}1...1a^{m_k} \mid k \geq 2 \;\&\; \forall i(1 \leq i \leq k)\, [m_i \geq 1]\}$, and
$\mathrm{L}_4 = \{a^{m_1}b_1a^{m_2}b_2...b_{k-1}a^{m_k} \mid k \geq 2 \;\&\; \forall i(1 \leq i \leq k)\, [m_i \geq 1] \;\&$
$\exists j(1 \leq j \leq k-1)\, [b_j = 2 \;\&\; \forall r(1 \leq r \leq k-1, r \neq j)\, [b_r = 1] \;\&\; m_1 = m_{j+1}]\}$.
Then,
(1) $\mathrm{L}_1 \in$ MSPACE(0), and thus $\in$ PSPACE(0),
(2) $\mathrm{L}_1 \cup \mathrm{L}_2 \in$ MSPACE(0), and thus $\in$ PSPACE(0),
(3) $\mathrm{L}_3 \in$ MSPACE(0), and thus $\in$ PSPACE(0),
(4) $\mathrm{L}_4 \in$ MSPACE(0), and thus $\in$ PSPACE(0),
(5) $\mathrm{L}_1\mathrm{L}_2 \notin$ MSPACE($o(\log n)$), and
(6) $\mathrm{L}_1\mathrm{L}_2 \notin$ PSPACE($o(\log\log n)$).

**Proofs** of (1),(2), and (4): By an adaptation of the proof of the fact [4] that $\{a^n b^n \mid n \geq 1\} \in$ MSPACE(0).

**Proof** of (3): Obvious.
**Proof** of (5): We first note that $L_1L_2 = \{a^{m_1}1a^{m_2}1...1a^{m_k} \mid k \geq 1 \ \& \ \forall i(1 \leq i \leq k)[m_i \geq 1] \ \& \ \exists j(2 \leq j \leq k)[m_1 = m_j]\}$.

For any integer $n \geq 1$, let $V(n) = \{1a^{m_1}1a^{m_2}1...1a^{m_n} \in \{1,a\}^+ \mid \forall i(1 \leq i \leq n)[1 \leq m_i \leq n]\}$. For each $w = 1a^{m_1}1a^{m_2}1...a^{m_n} \in V(n)$ let $contents(w) = \{a^j \mid j = m_i \text{ for some } i(1 \leq i \leq n)\}$. Divide V(n) into contents-equivalence classes by making $w$ and $w'$ *contents-equivalent* if $contents(w) = contents(w')$. There are

$$contents(n) = \binom{n}{1} + \binom{n}{2} + ... + \binom{n}{n} = 2^n - 1$$

contents-equivalence classes of words in $V(n)$. We denote by $W(n)$ the set of all the representatives arbitrarily chosen from these $contents(n)$ contents-equivalence classes. For each word $w \in W(n)$, $|w| \leq G(n) \triangleq (n+1)n$, which is a fixed polynomial in n. Let I be the set of positive integers greater than or equal to 2. Thus, for any $n \in I$, $|W(n)| = contents(n) = 2^n - 1 \geq c^n$ for some constant $c > 1$. It is easily seen that for every $n \in I$ and every $w, w' \in W(n)$ with $w \neq w'$, there are words $u = a^k$ $(1 \leq k \leq n)$, $v = \epsilon$ such that

(a) $|uwv| \leq H(n) \triangleq G(n) + n$, $|uw'v| \leq H(n)$, and
(b) either $\{uwv \in L \ \& \ uw'v \in \overline{L}\}$ or $\{uw'v \in L \ \& \ uwv \in \overline{L}\}$,

where for any language T, $\overline{T}$ denotes the complement of T.

Thus, by Lemma 3.1, if a two-way Monte Carlo Turing machine with space bound $S(n)$ recognizes $L_1L_2$, then $S(H(n))$ can not be $o(\log n)$, and thus $S(n)$ can not be $o(\log n)$. This completes the proof of '$L_1L_2 \notin MSPACE(o(\log n))$'.
**Proof** of (6): Let I be the set of positive integers, and let H: I→I be the function such that $H(n) = 2n + \frac{n(n+1)}{2}$.
For each $n \in I$, let $\overline{W}(n) = \{(u_1,v_1),(u_2,v_2),...,(u_n,v_n)\}$ be the ordered set of pairs of words such that for each $i$, $1 \leq i \leq n$, $u_i = a^i$ and $v_i = \epsilon$.
Furthermore, for the string $\alpha(1)\alpha(2)...\alpha(n) \in \{0,1\}^n$, let $0 < k_1 < k_2 < ... < k_l$ be all the values of $i$ such that $\alpha(i) = 1$, and $w_{\alpha(1)\alpha(2)...\alpha(n)} = 1a^{k_1}1a^{k_2}...1a^{k_l}$ be the string corresponding to $\alpha(1)\alpha(2)...\alpha(n)$.
It is easy to see that for each $n \in I$ and for each string $\alpha(1)\alpha(2)...\alpha(n) \in \{0,1\}^n$,

$$\begin{cases} u_i w_{\alpha(1)\alpha(2)...\alpha(n)} v_i \in L_1L_2 \text{ if } \alpha(i) = 1, \\ u_i w_{\alpha(1)\alpha(2)...\alpha(n)} v_i \in \overline{L_1L_2} \text{ if } \alpha(i) = 0, \end{cases}$$

and $|u_i w_{\alpha(1)\alpha(2)...\alpha(n)} v_i| \leq H(n)$ for all $i \in \{1,2,...,n\}$.

Thus by Lemma 3.2, if a two-way unbounded error probabilistic Turing machine with space bound $S(n)$ recognizes $L_1L_2$, then $S(H(n))$ can not be $o(\log\log n)$ and thus $S(n)$ can not be $o(\log\log n)$. This completes the proof of '$L_1L_2 \notin PSPACE(o(\log\log n))$'. ∎
By using Lemma 3.3., we can get the following theorem.
**Theorem 3.1.** MSPACE($o(\log n)$) and PSPACE($o(\log\log n)$) are not closed under concatenation, Kleene closure, and length-preserving homomorphism.

**Proof.** Let $L_i$, $i \in \{1, 2, 3, 4\}$, be the languages described in Lemma 3.3.
Concatenation: Nonclosure under concatenation follows from Lemma 3.3 (1), (5) and (6), and from the obvious fact that $L_2 \in$ MSPACE($o(\log n)$)$\cap$ PSPACE($o(\log\log n)$).
Kleene closure: It follows that $(L_1 \cup L_2)^* \cap L_3 = L_1 L_2 \notin$ MSPACE($o(\log n)$)$\cup$ PSPACE($o(\log\log n)$) (from Lemma 3.3 (5) and (6)). From this, Lemma 3.3 (2) and (3), and from the obvious fact that MSPACE($o(\log n)$) and PSPACE($o(\log\log n)$) are closed under intersection with regular languages, nonclosure under Kleene closure follows.
Length-preserving homomorphism: Nonclosure under length-preserving homomorphism follows from Lemma 3.3 (4), (5) and (6), and from the fact that $g(L_4) = L_1 L_2$, where $g : \{1, 2, a\} \to \{1, a\}$ is a length-preserving homomorphism such that $g(1) = g(2) = 1$ and $g(a) = a$. ∎

## 4. Closure Property of ASPACE($o$(log $n$))

This section shows that weak-ASPACE(S($n$)) and strong-ASPACE(S($n$)) are not closed under concatenation, Kleene closure, and length-preserving homomorphism for any log log $n \leq$S($n$)=$o(\log n)$. This result answers an open question in [10].

We first introduce a new idea of "rejecting computation tree" which was introduced in [11].

Given a 2aTm M, we write $c \vdash_{M,x} c'$ if configuration $c'$ is derived from configuration $c$ in one step of M on an input tape $x$. Let $C_M$ be the set of all the configurations of M. For each $c \in C_M$, let $Succ_{M,x}(c) = \{c' \in C_M | \, c \vdash_{M,x} c'\}$. If $Succ_{M,x}(c) = \emptyset$, then $c$ is said to be a halting configuration of M on $x$.

Let $l$ be a non-negative integer. An *$l$ space-bounded rejecting computation tree* of M on input $x$ is a (possibly infinite) nonempty labeled tree with the following properties:

(1) Each internal node $v$ (non leaf node) of the tree is labeled with an $l$ space-bounded configuration of M, $label(v)$.
(2) The root node is labeled with $I_M$.
(3) If $v$ is an internal node, and $label(v)$ is universal, then $v$ has exactly one child $u$ such that $label(u) \in Succ_{M,x}(label(v))$.
(4) If $v$ is an internal node, $label(v)$ is existential and $Succ_{M,x}(label(v)) = \{c_1, c_2, ..., c_k\}$, then $v$ has exactly $k$ children $v_1, v_2, ..., v_k$ such that $label(v_i) = c_i (1 \leq i \leq k)$.
(5) Each leaf node is a halting configuration which is not accepting, or a configuration which is not $l$ space-bounded.

A *reduced graph of $l$ space-bounded rejecting computation tree* (abbreviated by RG($l$)) of M on input $x$ is a finite, labeled directed multi-graph G=(V',E,$label|$

V') obtained from an $l$ space bounded rejecting computation tree T=(V,E,$label$) of M on $x$ by identifying nodes $v$ and $v'$ such that $label(v) = label(v')$ where V'⊆V and the labeling function $label$|V': V'→$C_M$ is injective.

For any directed graph G, let V(G) and E(G) denote the sets of nodes and edges of G, respectively.

Let $\delta_G^- = \{v' \in V(G) \mid (v', v) \in E(G)\}$ for each node $v$ ∈V(G). Obviously, for a reduced graph of $l$ space-bounded rejecting computation tree, there exists at most one node $v$ such that $\delta_G^-(v)$=0 labeled with $I_M$.

Let $\delta_G^+(v) = \{v' \in V(G) \mid (v, v') \in E(G)\}$ for each node $v$ ∈V(G). An RG($l$) G is *regular* if $|\delta_G^+(v)|$=1 for each node $v$ labeled with a universal configuration.

The following fact is used to get our desired result.

**Fact 4.1.** Let M be a 2aTm, $x$ be a word, and $l$ be a non-negative integer. The following statements are equivalent:

(1) There doesn't exist an $l$ space-bounded accepting computation tree of M on $x$.

(2) There exists a regular RG($l$) of M on $x$.

**Lemma 4.1.** Let

$L_5 = \{B(1)\#B(2)\#...\#B(n)cw_1cw_2c...cw_kccw'_1cw'_2c...cw'_r \in \{0,1,\#,c\}^+ \mid$
$n \geq 2 \,\&\, k \geq 1 \,\&\, r \geq 1 \,\&\, \forall i(1 \leq i \leq k)[w_i \in \{0,1\}^+] \,\&$
$\forall j(1 \leq j \leq r-1)[w'_j \in \{0,1\}^+] \,\&\, w'_r \in \{0,1\}^{\lceil \log n \rceil} \,\&$
$\forall l(1 \leq l \leq k)[w_l \neq w'_r]\}$, where for each $m(1 \leq m \leq n)$, B($m$) denotes the binary representation (with no leading zeros) of the integer $m$,

$L_6 = \{cw | w \in \{0,1\}^+\}^*$,

$L_7 = \{B(1)\#B(2)\#...\#B(n)cw_1cw_2c...cw_kccw'_1cw'_2c...cw'_r \in \{0,1,\#,c\}^+ \mid$
$n \geq 2 \,\&\, k \geq 1 \,\&\, r \geq 1 \,\&\, \forall i(1 \leq i \leq k)\forall j(1 \leq j \leq r)\, [w_i, w'_j \in \{0,1\}^+]\}$,and

$L_8 = \{B(1)\#B(2)\#...\#B(n)cw_1cw_2c...cw_kcc_1w'_1c_2w'_2...c_rw'_r \in \{0,1,\#,c,d\}^+ \mid$
$n \geq 2 \,\&\, k \geq 1 \,\&\, r \geq 1 \,\&\, \exists i(1 \leq i \leq r)\, [c_i = d \,\&\, w'_i \in \{0,1\}^{\lceil \log n \rceil} \,\&$
$\forall j(1 \leq j \leq k)\, [w_j \neq w'_i] \,\&\, \forall l(1 \leq l \leq r,\, l \neq i)[c_l = c \,\&\, w'_l \in \{0,1\}^+]]\}$.

Then

(1) $L_5$ ∈ strong-ASPACE(log log $n$),
(2) $L_5 \cup L_6$ ∈ strong-ASPACE(log log $n$),
(3) $L_7$ ∈ strong-ASPACE(log log $n$),
(4) $L_8$ ∈ strong-ASPACE(log log $n$), and
(5) $L_5L_6$ ∉ weak-ASPACE($o(\log n)$).

**Proofs** of (1)-(4): By standard techniques as in [10]. We leave the proofs to the reader as an easy exercise.

**Proof** of (5): Suppose to the contrary there is a weakly L($n$) space-bounded 2aTm M which accepts $L_5L_6$.

For each $n \geq 2$, let

W($n$) ={B(1)#B(2)#...#B($n$)$cw_1cw_2c...cw_{p(n)}ccw_1cw_2c...cw_{p(n)}$|$\forall i(1 \leq i \leq p(n))$ $[w_i \in \{0,1\}^{\lceil \log n \rceil}]$}, where $p(n) = 2^{\lceil \log n \rceil}$.

As easily seen, each $x$ in W($n$) is not in $L_5L_6$. Thus, from Fact 4.1, for each $x \in$ W($n$), there exists a fixed regular reduced graph of L($r(n)$) space-bounded rejecting computation tree of M on $x$, where $r(n)$ is the length of each word in W($n$) and $r(n) = O(n \log n)$. We denote this graph by G($x$).

For each $x$ =B(1)#B(2)#...#B($n$)$cw_1cw_2c...cw_{p(n)}ccw_1cw_2c...cw_{p(n)}$ in W($n$), we call the left part of $x$ (i.e., B(1)#B(2)#...#B($n$)$cw_1cw_2c...cw_{p(n)}$) the *left segment* of $x$, and the right part of $x$ (i.e., $ccw_1cw_2c...cw_{p(n)}$) the *right segment* of $x$.

For each $x \in$ W($n$), we partition V(G($x$)), the set of nodes of G($x$), as follows:

$$\mathrm{V}(\mathrm{G}(x))=\mathrm{V}_{left}(\mathrm{G}(x))\cup\mathrm{V}_{right}(\mathrm{G}(x)),$$

where $\mathrm{V}_{left}(\mathrm{G}(x))$ (resp., $\mathrm{V}_{right}(\mathrm{G}(x))$) denotes the set of nodes of G($x$) which are labeled by configurations representing that the input head of M is on the left segment of $x$ or on the left endmaker ¢ (resp., on the right segment of $x$ or on the right endmaker \$).

We then extract the set set of nodes in $\mathrm{V}_{left}(\mathrm{G}(x))$ (resp., $\mathrm{V}_{right}(\mathrm{G}(x))$) that are labeled by configurations which M enters just after the input head crosses from the right segment of $x$ to the left segment of $x$ (resp., from the left segment of $x$ to the right segment of $x$). That is, we have

$\mathrm{V}^{\leftarrow}_{left}(\mathrm{G}(x)) = \{v \in \mathrm{V}_{left}(\mathrm{G}(x)) | (v', v) \in \mathrm{E}(\mathrm{G}(x)) \& v' \in \mathrm{V}_{right}(\mathrm{G}(x))\}$,

$\mathrm{V}^{\rightarrow}_{right}(\mathrm{G}(x)) = \{v \in \mathrm{V}_{right}(\mathrm{G}(x)) | (v', v) \in \mathrm{E}(\mathrm{G}(x)) \& v' \in \mathrm{V}_{left}(\mathrm{G}(x))\}$,

where E(G($x$)) denotes the set of edges of G($x$).

Furthermore, we partition E(G($x$)) as follows:

$\mathrm{E}(\mathrm{G}(x))=\mathrm{E}_{left}(\mathrm{G}(x))\cup\mathrm{E}_{right}(\mathrm{G}(x))\cup\mathrm{E}_{\leftarrow}(\mathrm{G}(x))\cup \mathrm{E}_{\rightarrow}(\mathrm{G}(x))$,

where

$\mathrm{E}_{left}(\mathrm{G}(x)) = \{(v, v') \in \mathrm{E}(\mathrm{G}(x)) | v \in \mathrm{V}_{left}(\mathrm{G}(x)) \ \& \ v' \in \mathrm{V}_{left}(\mathrm{G}(x))\}$,

$\mathrm{E}_{right}(\mathrm{G}(x)) = \{(v, v') \in \mathrm{E}(\mathrm{G}(x)) | v \in \mathrm{V}_{right}(\mathrm{G}(x)) \ \& \ v' \in \mathrm{V}_{right}(\mathrm{G}(x))\}$,

$\mathrm{E}_{\leftarrow}(\mathrm{G}(x)) = \{(v, v') \in \mathrm{E}(\mathrm{G}(x)) | v \in \mathrm{V}_{right}(\mathrm{G}(x)) \ \& \ v' \in \mathrm{V}^{\leftarrow}_{left}(\mathrm{G}(x))\}$,

$\mathrm{E}_{\rightarrow}(\mathrm{G}(x)) = \{(v, v') \in \mathrm{E}(\mathrm{G}(x)) | v \in \mathrm{V}_{left}(\mathrm{G}(x)) \ \& \ v' \in \mathrm{V}^{\rightarrow}_{right}(\mathrm{G}(x))\}$.

We let

$\textit{Cross-Pair}(\mathrm{G}(x)) = < label(\mathrm{V}^{\leftarrow}_{left}(\mathrm{G}(x))), label(\mathrm{V}^{\rightarrow}_{right}(\mathrm{G}(x)))>$.

For each word $x =$B(1)#B(2)#...#B($n$)$cw_1cw_2c...cw_{p(n)}ccw_1cw_2c...cw_{p(n)} \in$ W($n$), let contents($x$) $= \{w \in \{0,1\}^{\lceil \log n \rceil} | w = w_i$ for some $1 \leq i \leq p(n)\}$. For any two words $x, y \in$ W($n$), divide W($n$) into contents-equivalence classes by making $x$ and $y$ *contents-equivalent* if contents($x$)=contents($y$).

There are

$$\mathrm{contents}(n) = \binom{p(n)}{1} + \binom{p(n)}{2} + ... + \binom{p(n)}{p(n)} = 2^{p(n)} - 1$$

contents-equivalence classes.

We denote by CONTENTS($n$) the set of all the representatives arbitrarily chosen from these contents($n$) contents-equivalence classes. Of course,

$$|\mathrm{CONTENTS}(n)|=\mathrm{contents}(n)=2^{p(n)} - 1.$$

**Proposition 4.1** *For two different elements $x, y \in CONTENT(n)$,*

$$\textit{Cross-Pair}(\mathrm{G}(x)) \neq \textit{Cross-Pair}(\mathrm{G}(y)).$$

[**Proof.** Suppose to the contrary that $\textit{Cross-Pair}(\mathrm{G}(x)) = \textit{Cross-Pair}(\mathrm{G}(y))$. From G($x$) and G($y$), we construct the following graph G($x$)⨁G($y$):

$\mathrm{V}(\mathrm{G}(x)\bigoplus\mathrm{G}(y)) = \mathrm{V}_{left}(\mathrm{G}(x))\cup\mathrm{V}_{right}(\mathrm{G}(y))$,

$\mathrm{E}(\mathrm{G}(x)\bigoplus\mathrm{G}(y)) = \mathrm{E}_{left}(\mathrm{G}(x))\cup\mathrm{E}_{right}(\mathrm{G}(y))\cup\mathrm{E}_{\leftarrow}\cup\mathrm{E}_{\rightarrow}$,

where

$E_{\leftarrow} = \{(u, v') \in V_{right}G(y)) \times V^{\leftarrow}_{left}(G(x)) | (u, u') \in E_{\leftarrow}(G(y)) \& (v, v') \in E_{\leftarrow}(G(x)) \&$ $label(u') = label(v')\}$, and
$E_{\rightarrow} = \{(v, u') \in V_{left}G(x)) \times V^{\rightarrow}_{right}(G(y)) | (v, v') \in E_{\rightarrow}(G(x)) \& (u, u') \in E_{\rightarrow}(G(y)) \&$ $label(u') = label(v')\}$.
Intuitively, $G(x) \bigcirc\!\!\!| \, G(y)$ is the graph obtained by connecting the part of $G(x)$ which correspondes to the left segment of $x$ with the part of $G(y)$ which correspondes to the right segment of $y$ (see Fig.1). From our assumption that $Cross\text{-}Pair(G(x)) = Cross\text{-}Pair(G(y))$, it is easy to see that the following fact holds.
**Fact 4.2.** (1) For any $v \in V_{left}(G(x)), label(\delta^{+}_{G(x)\bigcirc\!\!\!|\,G(y)}(v)) = label(\delta^{+}_{G(x)}(v))$, and
(2) for any $v \in V_{right}(G(y)), label(\delta^{+}_{G(x)\bigcirc\!\!\!|\,G(y)}(v)) = label(\delta^{+}_{G(y)}(v))$.

We assume without loss of generality that

$$\text{contents}(y) - \text{contents}(x) \neq \emptyset \text{ (empty set)}.$$

Now, consider the word $z_1 z_2$ such that
(i) $z_1$ is identical with the left segment of $x$, and
(ii) $z_2$ is identical with the right segment of $y$.

Let $v_0$ be the node of $G(x)$ labeled by $I_M$ (note that $v_0$ is in $V(G(x)\bigcirc\!\!\!|\,G(y))$. We consider the following depth-first search on $V(G(x)\bigcirc\!\!\!|\,G(y))$ starting at $v_0$:
① $v := v_0$;
② for each $v_i \in V(G(x)\bigcirc\!\!\!|\,G(y))$ such that $\delta^{+}(v) = \{v_1, v_2, ..., v_k\}$:
- if $v_i$ has not been searched, then set $v := v_i$ and repeat ②.
- if every $v_i$ in $\delta^{+}(v)$ has searched, then return to $v$.

From Fact 4.2 we can easily see that the sequence of values of variable $v$ above constructs a regular reduced graph of $L(r(n))$ space-bounded rejecting computation tree of M on $z_1 z_2$. This contradicts the fact that $z_1 z_2$ is in $L_5 L_6$. This completes the proof of Proposition 4.1.]

For each $n \geq 2$,

$$C(n) = \{Cross\text{-}Pair(G(x)) | x \in \text{CONTENTS}(n)\}.$$

Then

$$|C(n)| \leq 2^{2 \cdot e[n]},$$

where $e[n] = sL(r(n))t^{L(r(n))}$, $s$ and $t$ are the numbers of states and work tape symbols of M, respectively.

Since $L(n) = o(\log n)$, it follows that for large $n$,

$$\text{contents}(n) > C(n).$$

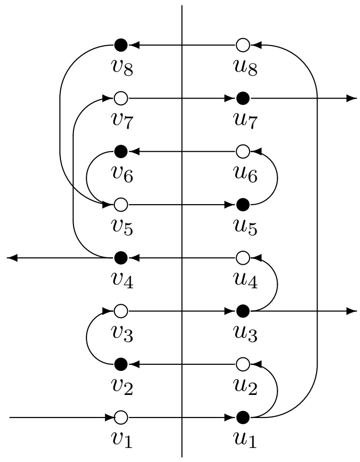
Therefore, such a large $n$, there must exist two different $x, y$ in CONTENTS$(n)$ such that $Cross\text{-}Pair(G(x)) = Cross\text{-}Pair(G(y))$. This contradicts Proposition 4.1, which completes the proof of Lemma 4.1 (5). ∎
By using Lemma 4.1., we can get the following theorem.
**Theorem 4.1.** For each function $\log\log n \leq S(n) = o(\log n)$, weak-ASPACE(S($n$)) and strong-ASPACE(S($n$)) are not closed under concatenation, Kleene closure,

and length-preserving homomorphism.
**Proof**. Let $L_i$, $i \in \{5,6,7,8\}$, be the languages described in Lemma 4.1.
Concatenation: Nonclosure under concatenation follows from Lemma 4.1 (1) and (5), and from the obvious fact that $L_5 \in$ strong-ASPACE(log log n).
Kleene closure: It follows that $(L_5 \cup L_6)^* \cap L_7 = L_5 L_6 \notin$weak-ASPACE(o(log n)) (from Lemma 4.1 (5)). From this, Lemma 4.1 (2) and (3), and from the obvious fact that strong-ASPACE(L($n$)) and weak-ASPACE(L($n$)) are closed under intersection for any function L($n$), nonclosure under Kleene closure follows.
Length-preserving homomorphism: Nonclosure under length-preserving homomorphism follows from Lemma 4.1 (4) and (5), and from the fact that $h(L_8) = L_5 L_6$, where $h : \{0, 1, \#, c, d\} \to \{0, 1, \#, c\}$ is a length-preserving homomorphism such that $h(0)$=0, $h(1)$=1, $h(\#)$=# and $h(c) = h(d) = c$. ∎

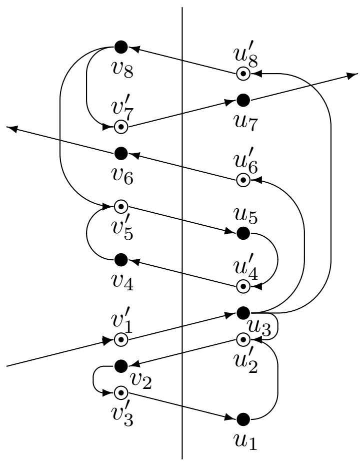
## 5. Conclusion

We conclude this paper by giving the following open problem:

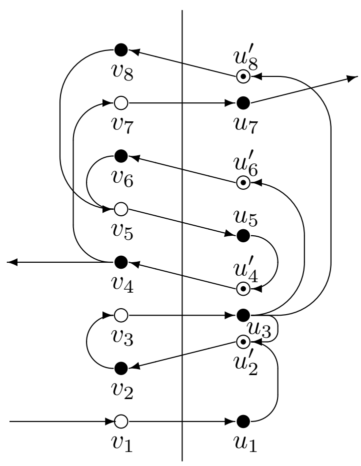
- Is closed PSPACE(L($n$)) under concatenation, Kleene closure and length-preserving homomorphism for log log $n \leq$L($n$) $= o(\log n)$?




(1) G($x$)




(2) G($y$)




(3) G($x$)$\bigcirc\!\!\!\!|\,$G($y$)

**Fig. 1.** Connection of graphs G($x$) and G($y$), where for simplicity, we identify node $v$ with its lavel, $label(v)$.

## References

[1] B. V. Braunmühl, R. Gengler and R. Rettinger, "The alternation hierarchy for subalgorithmic space is infinite", Comput. Complexity 3, pp. 207-230, 1993.

[2] J. H. Chang, O. H. Ibarra and B. Ravikumar, "Some observations concerning alternating Turing machines using small space", Information Processing Letters 25, pp. 1-9, 1987.

[3] C. Dwork and L. Stockmeyer, "A time complexity gap for two-way probabilistic finite-state automata", SIAM J. COMPUT. 19, pp. 1011-1023, 1990.

[4] R. Freivalds, "Probabilistic two-way machines", Proceedings of the International Symposium on Mathematical Foundations of Computer Science, LNCS 118, pp. 33-45, 1981.

[5] R. Freivalds and M. Karpinski, "Lower space bounds for randomized computation", Proceedings of ICALP'94, LNCS 820, pp. 580-592, 1994.

[6] V. Geffert, "A hierarchy that does not collapse: alternations in low level space", Informatique théorique et Applications / Theoretical Informatics and Applications 28, no.5, pp. 462-512, 1994.

[7] J. Gill, "Computational complexity of probabilistic Turing machines", SIAM J. COMPUT. 6, no.4, pp. 675-695, 1977.

[8] A.G. Greenberg and A. Weiss, "A lower bound for probabilistic algorithms for finite state machines", JCSS, Vol. 33, pp. 88-105, 1986.

[9] J. E. Hopcroft and J. D. Ullman, "Formal Languages and Their Relation to Automata", Addison-Wesley, Reading, Mass., 1969.

[10] A. Ito, K. Inoue and I. Takanami, "A note on alternating Turing machines using small space", IEICE Trans. E 70 (10), pp. 990-996, 1987.

[11] A. Ito, T. Okazaki, K. Inoue and Y. Wang, "Nonclosure under complementation of two-dimensional alternating $o(\log \log m)$ space complexity class", IEICE Trans. Vol. I81-D-I, no 6, pp. 593-603, 1998.

[12] K. Iwama, "ASPACE($o(\log \log n)$) is regular", SIAM J. COMPUT. 22, pp. 207-221, 1993.

[13] M. Liśkiewicz and R. Reischuk, "The sublogarithmic alternating space world", SIAM J. COMPUT. 25, pp. 828-861, 1996.

[14] J. Wang, " A note on two-way probabilistic automata", Information Processing Letters 43, pp. 321-326, 1992.