

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 $\{0^n 10^n | n \geq 1\}$ 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 \mathcal{C} and the right endmaker \mathcal{S} , 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 ε 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 $\text{MSPACE}(S(n))$ (resp., $\text{PSPACE}(S(n))$) denote the class of languages recognized by two-way Monte Carlo Turing machines (resp., two-way unbounded 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 endmarker \mathcal{C} and the right endmarker \mathcal{S} , an input head which can move left or right on the input tape, and a semi-infinite read-write 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, γ, 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 λ 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-bounded* 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 $\text{weak-ASPACE}(S(n))$ (resp., $\text{strong-ASPACE}(S(n))$) 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 $\text{MSPACE}(o(\log n))$ and $\text{PSPACE}(o(\log \log n))$ 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 Σ^* 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) $|uvw| \leq H(n)$, $|uw'v| \leq H(n)$, and
 - (b)

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

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 $W(n) = \{(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)\}$ such that for every string $\gamma(1)\gamma(2)\dots\gamma(n) \in \{0, 1\}^n$, there is a word w such that

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

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

$$L_1 = \{a^{m_1} 1 a^{m_2} 1 \dots 1 a^{m_k} \mid k \geq 2 \ \& \ \forall i (1 \leq i \leq k) [m_i \geq 1] \ \& \ m_1 = m_k\},$$

$$L_2 = \{1 a^m \mid m \geq 1\}^*,$$

$$L_3 = \{a^{m_1} 1 a^{m_2} 1 \dots 1 a^{m_k} \mid k \geq 2 \ \& \ \forall i (1 \leq i \leq k) [m_i \geq 1]\}, \text{ and}$$

$$L_4 = \{a^{m_1} b_1 a^{m_2} b_2 \dots b_{k-1} a^{m_k} \mid k \geq 2 \ \& \ \forall i (1 \leq i \leq k) [m_i \geq 1] \ \&$$

$$\exists j (1 \leq j \leq k-1) [b_j = 2 \ \& \ \forall r (1 \leq r \leq k-1, r \neq j) [b_r = 1] \ \& \ m_1 = m_{j+1}]\}.$$

Then,

- (1) $L_1 \in \text{MSPACE}(0)$, and thus $\in \text{PSPACE}(0)$,
- (2) $L_1 \cup L_2 \in \text{MSPACE}(0)$, and thus $\in \text{PSPACE}(0)$,
- (3) $L_3 \in \text{MSPACE}(0)$, and thus $\in \text{PSPACE}(0)$,
- (4) $L_4 \in \text{MSPACE}(0)$, and thus $\in \text{PSPACE}(0)$,
- (5) $L_1 L_2 \notin \text{MSPACE}(o(\log n))$, and
- (6) $L_1 L_2 \notin \text{PSPACE}(o(\log \log n))$.

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

Proof of (3): Obvious.

Proof of (5): We first note that $L_1L_2 = \{a^{m_1}1a^{m_2}1\dots1a^{m_k} \mid k \geq 1 \ \& \ \forall i(1 \leq i \leq k)[m_i \geq 1] \ \& \ \exists j(2 \leq j \leq k)[m_j = m_i]\}$.

For any integer $n \geq 1$, let $V(n) = \{1a^{m_1}1a^{m_2}1\dots1a^{m_n} \in \{1, a\}^+ \mid \forall i(1 \leq i \leq n)[1 \leq m_i \leq n]\}$. For each $w = 1a^{m_1}1a^{m_2}1\dots a^{m_n} \in V(n)$ 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

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

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 $\{uwv \in L \ \& \ uw'v \in \bar{L}\}$ or $\{uw'v \in L \ \& \ uwv \in \bar{L}\}$,

where for any language T , \bar{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 ' $L_1L_2 \notin \text{MSPACE}(o(\log n))$ '.

Proof of (6): Let I be the set of positive integers, and let $H: I \rightarrow I$ be the function such that $H(n) = 2n + \frac{n(n+1)}{2}$.

For each $n \in I$, let $\bar{W}(n) = \{(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)\}$ 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 $\alpha(1)\alpha(2)\dots\alpha(n) \in \{0, 1\}^n$, let $0 < k_1 < k_2 < \dots < k_l$ be all the values of i such that $\alpha(i) = 1$, and $w_{\alpha(1)\alpha(2)\dots\alpha(n)} = 1a^{k_1}1a^{k_2}\dots1a^{k_l}$ be the string corresponding to $\alpha(1)\alpha(2)\dots\alpha(n)$.

It is easy to see that for each $n \in I$ and for each string $\alpha(1)\alpha(2)\dots\alpha(n) \in \{0, 1\}^n$,

$$\begin{cases} u_i w_{\alpha(1)\alpha(2)\dots\alpha(n)} v_i \in L_1L_2 & \text{if } \alpha(i) = 1, \\ u_i w_{\alpha(1)\alpha(2)\dots\alpha(n)} v_i \in \bar{L}_1\bar{L}_2 & \text{if } \alpha(i) = 0, \end{cases}$$

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

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 ' $L_1L_2 \notin \text{PSPACE}(o(\log \log n))$ '. ■

By using Lemma 3.3., we can get the following theorem.

Theorem 3.1. $\text{MSPACE}(o(\log n))$ and $\text{PSPACE}(o(\log \log n))$ 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 $L_2 \in \text{MSPACE}(o(\log n)) \cap \text{PSPACE}(o(\log \log n))$.

Kleene closure: It follows that $(L_1 \cup L_2)^* \cap L_3 = L_1 L_2 \notin \text{MSPACE}(o(\log n)) \cup \text{PSPACE}(o(\log \log n))$ (from Lemma 3.3 (5) and (6)). From this, Lemma 3.3 (2) and (3), and from the obvious fact that $\text{MSPACE}(o(\log n))$ and $\text{PSPACE}(o(\log \log n))$ 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 $g : \{1, 2, a\} \rightarrow \{1, a\}$ is a length-preserving homomorphism such that $g(1) = g(2) = 1$ and $g(a) = a$. ■

4. Closure Property of $\text{ASPACE}(o(\log n))$

This section shows that weak- $\text{ASPACE}(S(n))$ and strong- $\text{ASPACE}(S(n))$ 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 $c \stackrel{|}{\underset{M,x}{\vdash}} c'$ 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 $\text{Succ}_{M,x}(c) = \{c' \in C_M \mid c \stackrel{|}{\underset{M,x}{\vdash}} c'\}$. If $\text{Succ}_{M,x}(c) = \emptyset$, 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 , $\text{label}(v)$.
- (2) The root node is labeled with I_M .
- (3) If v is an internal node, and $\text{label}(v)$ is universal, then v has exactly one child u such that $\text{label}(u) \in \text{Succ}_{M,x}(\text{label}(v))$.
- (4) If v is an internal node, $\text{label}(v)$ is existential and $\text{Succ}_{M,x}(\text{label}(v)) = \{c_1, c_2, \dots, c_k\}$, then v has exactly k children v_1, v_2, \dots, v_k such that $\text{label}(v_i) = c_i$ ($1 \leq i \leq k$).
- (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 $\text{RG}(l)$) of M on input x is a finite, labeled directed multi-graph $G = (V, E, \text{label})$

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 $V' \subseteq V$ and the labeling function $label|_{V'}: V' \rightarrow C_M$ 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 $v \in V(G)$. 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 $v \in V(G)$. 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

$$L_5 = \{B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_kccw'_1cw'_2c\dots cw'_r \in \{0, 1, \#, c\}^+ \mid$$

$$n \geq 2 \ \& \ k \geq 1 \ \& \ r \geq 1 \ \& \ \forall i(1 \leq i \leq k)[w_i \in \{0, 1\}^+] \ \&$$

$$\forall j(1 \leq j \leq r-1)[w'_j \in \{0, 1\}^+] \ \& \ w'_r \in \{0, 1\}^{\lceil \log n \rceil} \ \&$$

$$\forall l(1 \leq l \leq k)[w_l \neq w'_l]\}, \text{ where for each } m(1 \leq m \leq n), B(m) \text{ denotes}$$

the binary representation (with no leading zeros) of the integer m ,

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

$$L_7 = \{B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_kccw'_1cw'_2c\dots cw'_r \in \{0, 1, \#, c\}^+ \mid$$

$$n \geq 2 \ \& \ k \geq 1 \ \& \ r \geq 1 \ \& \ \forall i(1 \leq i \leq k)\forall j(1 \leq j \leq r) [w_i, w'_j \in \{0, 1\}^+], \text{ and}$$

$$L_8 = \{B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_kcc_1w'_1c_2w'_2\dots c_rw'_r \in \{0, 1, \#, c, d\}^+ \mid$$

$$n \geq 2 \ \& \ k \geq 1 \ \& \ r \geq 1 \ \& \ \exists i(1 \leq i \leq r) [c_i = d \ \& \ w'_i \in \{0, 1\}^{\lceil \log n \rceil} \ \&$$

$$\forall j(1 \leq j \leq k) [w_j \neq w'_j] \ \& \ \forall l(1 \leq l \leq r, l \neq i)[c_l = c \ \& \ w'_l \in \{0, 1\}^+]\}.$$

Then

- (1) $L_5 \in \text{strong-ASPACE}(\log \log n)$,
- (2) $L_5 \cup L_6 \in \text{strong-ASPACE}(\log \log n)$,
- (3) $L_7 \in \text{strong-ASPACE}(\log \log n)$,
- (4) $L_8 \in \text{strong-ASPACE}(\log \log n)$, and
- (5) $L_5L_6 \notin \text{weak-ASPACE}(o(\log n))$.

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

$$W(n) = \{B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_{p(n)}ccw_1cw_2c\dots cw_{p(n)} \mid \forall i(1 \leq i \leq p(n))$$

$$[w_i \in \{0, 1\}^{\lceil \log n \rceil}]\}, \text{ where } p(n) = 2^{\lceil \log n \rceil}.$$

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 $x = B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_{p(n)}ccw_1cw_2c\dots cw_{p(n)}$ in $W(n)$,

we call the left part of x (i.e., $B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_{p(n)}$) the *left segment* of x , and the right part of x (i.e., $ccw_1cw_2c\dots cw_{p(n)}$) 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:

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

where $V_{left}(G(x))$ (resp., $V_{right}(G(x))$) 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 \mathcal{Q} (resp., on the right segment of x or on the right endmaker \mathcal{S}).

We then extract the set set of nodes in $V_{left}(G(x))$ (resp., $V_{right}(G(x))$) 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

$$\begin{aligned} V_{left}^{\leftarrow}(G(x)) &= \{v \in V_{left}(G(x)) \mid (v', v) \in E(G(x)) \& v' \in V_{right}(G(x))\}, \\ V_{right}^{\rightarrow}(G(x)) &= \{v \in V_{right}(G(x)) \mid (v', v) \in E(G(x)) \& v' \in V_{left}(G(x))\}, \end{aligned}$$

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

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

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

where

$$\begin{aligned} E_{left}(G(x)) &= \{(v, v') \in E(G(x)) \mid v \in V_{left}(G(x)) \& v' \in V_{left}(G(x))\}, \\ E_{right}(G(x)) &= \{(v, v') \in E(G(x)) \mid v \in V_{right}(G(x)) \& v' \in V_{right}(G(x))\}, \\ E_{\leftarrow}(G(x)) &= \{(v, v') \in E(G(x)) \mid v \in V_{right}(G(x)) \& v' \in V_{left}^{\leftarrow}(G(x))\}, \\ E_{\rightarrow}(G(x)) &= \{(v, v') \in E(G(x)) \mid v \in V_{left}(G(x)) \& v' \in V_{right}^{\rightarrow}(G(x))\}. \end{aligned}$$

We let

$$Cross\text{-}Pair(G(x)) = \langle label(V_{left}^{\leftarrow}(G(x))), label(V_{right}^{\rightarrow}(G(x))) \rangle.$$

For each word $x = B(1)\#B(2)\#\dots\#B(n)cw_1cw_2c\dots cw_{p(n)}ccw_1cw_2c\dots cw_{p(n)} \in W(n)$, let $contents(x) = \{w \in \{0, 1\}^{\lceil \log n \rceil} \mid w = w_i \text{ for some } 1 \leq i \leq p(n)\}$. 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

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

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,

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

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

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

[Proof. Suppose to the contrary that $Cross\text{-}Pair(G(x)) = Cross\text{-}Pair(G(y))$.

From $G(x)$ and $G(y)$, we construct the following graph $G(x) \oplus G(y)$:

$$\begin{aligned} V(G(x) \oplus G(y)) &= V_{left}(G(x)) \cup V_{right}(G(y)), \\ E(G(x) \oplus G(y)) &= E_{left}(G(x)) \cup E_{right}(G(y)) \cup E_{\leftarrow} \cup E_{\rightarrow}, \end{aligned}$$

where

$E_{\leftarrow} = \{(u, v') \in V_{right}G(y) \times V_{left}^{\leftarrow}(G(x)) \mid (u, u') \in E_{\leftarrow}(G(y)) \& (v, v') \in E_{\leftarrow}(G(x)) \& label(u') = label(v')\}$, and
 $E_{\rightarrow} = \{(v, u') \in V_{left}G(x) \times V_{right}^{\rightarrow}(G(y)) \mid (v, v') \in E_{\rightarrow}(G(x)) \& (u, u') \in E_{\rightarrow}(G(y)) \& label(u') = label(v')\}$.

Intuitively, $G(x) \oplus G(y)$ is the graph obtained by connecting the part of $G(x)$ which corresponds to the left segment of x with the part of $G(y)$ which corresponds to the right segment of y (see Fig.1). From our assumption that $Cross-Pair(G(x)) = Cross-Pair(G(y))$, it is easy to see that the following fact holds.

Fact 4.2. (1) For any $v \in V_{left}(G(x))$, $label(\delta_{G(x) \oplus G(y)}^+(v)) = label(\delta_{G(x)}^+(v))$, and
(2) for any $v \in V_{right}(G(y))$, $label(\delta_{G(x) \oplus G(y)}^+(v)) = label(\delta_{G(y)}^+(v))$.

We assume without loss of generality that

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

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 $V(G(x) \oplus G(y))$).

We consider the following depth-first search on $V(G(x) \oplus G(y))$ starting at v_0 :

- ① $v := v_0$;
- ② for each $v_i \in V(G(x) \oplus G(y))$ such that $\delta^+(v) = \{v_1, v_2, \dots, v_k\}$:
 - 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 $n \geq 2$,

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

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 ,

$$contents(n) > C(n).$$

Therefore, such a large n , there must exist two different x, y in $CONTENTS(n)$ such that $Cross-Pair(G(x)) = Cross-Pair(G(y))$. 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-SPACE($S(n)$) and strong-SPACE($S(n)$) 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 $L_5 \in \text{strong-SPACE}(\log \log n)$.

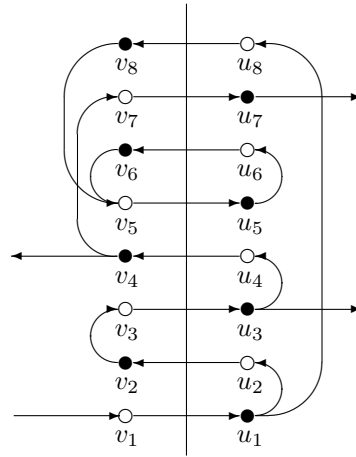
Kleene closure: It follows that $(L_5 \cup L_6)^* \cap L_7 = L_5 L_6 \notin \text{weak-SPACE}(o(\log n))$ (from Lemma 4.1 (5)). From this, Lemma 4.1 (2) and (3), and from the obvious fact that $\text{strong-SPACE}(L(n))$ and $\text{weak-SPACE}(L(n))$ are closed under intersection for any function $L(n)$, 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 $h : \{0, 1, \#, c, d\} \rightarrow \{0, 1, \#, c\}$ is a length-preserving homomorphism such that $h(0)=0$, $h(1)=1$, $h(\#)=\#$ and $h(c) = h(d) = c$. ■

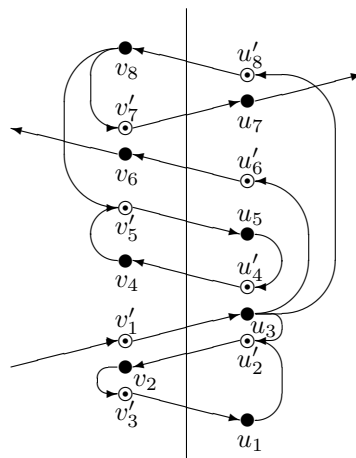
5. Conclusion

We conclude this paper by giving the following open problem:

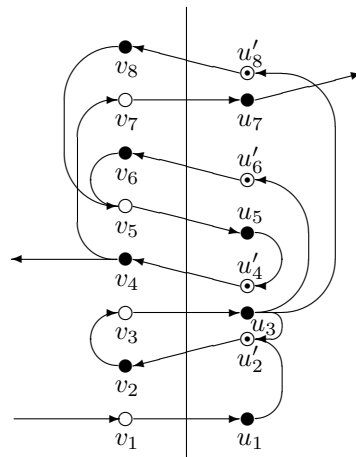
- Is closed $\text{PSPACE}(L(n))$ under concatenation, Kleene closure and length-preserving homomorphism for $\log \log n \leq L(n) = o(\log n)$?



(1) $G(x)$



(2) $G(y)$



(3) $G(x) \oplus G(y)$

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

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.