ICOT Technical Report: TR-489

TR-489

On Learning Semilinear Sets: An Approach to Learning Parallel Computation Models

by
Y. Takada (Fujitsu)

July, 1989

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

On Learning Semilinear Sets: An Approach to Learning Parallel Computation Models [1]

Yuji Takada

International Institute for Advanced Study of Social Information Science (IIAS-SIS)
FUJITSU LIMITED
140, Miyamoto, Numazu, Shizuoka 410-03, Japan
E-mail : yuji%iias.fujitsu.co.jp@uunet.uu.net

[1]This is a part of the work in the major R&D of the Fifth Generation Computer Project, conducted under program set up by MITI.

Abstract

Semilinear sets play an important role in parallel computation models such as matrix grammars, commutative grammars, and Petri nets. In this paper, we consider the problem of learning semilinear sets, that is, the problem of finding a description of a semilinear set. We shall show that

- the family of semilinear sets is not learnable from positive examples, while the family of linear sets is learnable from positive examples, although the problem of learning linear sets from given positive examples seems to be computationally intractable,
- given any teacher who answers queries whether $Q \subseteq Q'$ and queries whether $Q' \subseteq Q$ for any unknown semilinear set Q and any guessed semilinear set Q', there exists a learning algorithm which identifies any semilinear sets and halts, although the algorithm is time-consuming,
- given any teacher who answers queries whether $Q \subseteq Q'$ for any unknown linear set Q and any *semilinear set* Q', there exists a learning algorithm which identifies any linear sets *in polynomial time* and halts.

These results may provide partial solutions to the problem of learning parallel computation models.

1 Introduction

One of major subjects in recent computer science is to formalize and analyze parallel computation of concurrent systems and, for this purpose, several formal models have been proposed. Petri nets [10], commutative grammars [3], and matrix grammars [11] seem to be most successful models, for which substantial theories and analysis techniques have been developed enough to apply them to many practical concurrent systems organization. Although decision problems have been well investigated for these models, there have been few studies up to now from the learning point of view, which may be one of advanced and important subjects in computer science. In this paper, we shed light on the problem of learning these parallel computation models.

A concept which plays an important role in these models is a semilinear set: a subset of lattice points is said to be *linear* if and only if it is a coset of finitely generated sub-semigroups of the set of all lattice points with nonnegative coordinates, and a finite union of linear sets is said to be *semilinear*. For examples, for any equal matrix language [12] and simple matrix language [7], an image set on Parikh mapping is semilinear. Also, a reachability set of any weakly persistent Petri net [13] is semilinear. The semilinearity provides effective solutions for some decision problems on these models.

In this paper, we shall consider semilinear sets from the learning point of view. We consider the problem of learning semilinear sets, that is, the problem of finding a description of an unknown semilinear set. We shall show that

- the family of semilinear sets is not learnable from positive examples, while the family of linear sets is learnable from positive examples, although the problem of learning linear sets from given positive examples seems to be computationally intractable,

- given any teacher who answers queries whether $Q \subseteq Q'$ and queries whether $Q' \subseteq Q$ for any unknown semilinear set Q and any guessed semilinear set Q', there exists a learning algorithm which identifies any semilinear sets and halts, although the algorithm is time-consuming,

- given any teacher who answers queries whether $Q \subseteq Q'$ for any unknown linear set Q and any *semilinear set* Q', there exists a learning algorithm which identifies any linear sets *in polynomial time* and halts.

These results may provide partial solutions to the problem of learning parallel computation models.

In Section 2, the family of linear sets and the family of semilinear sets are formally defined. In Section 3, we note some basic properties of semilinear sets. The family of semilinear sets is closed under Boolean operations and the equivalence problem is effectively solvable. Furthermore, it is shown that the membership problem is NP-complete. These properties shall play important roles in the problem of learning semilinear sets. In Section 4, we show learnabilities from positive examples for the family of linear sets and the family of semilinear sets. It is proved that the family of linear sets is learnable from positive examples, while the family of semilinear sets is not learnable from positive examples. In Section 5, we present a simple learning method for linear sets from positive examples. It seems that the problem of learning linear sets from given examples is computationally intractable. In Section 6, we assume that there exists a teacher who answers queries whether $Q \subseteq Q'$ and queries whether $Q' \subseteq Q$ for any unknown semilinear set Q and any guessed semilinear set Q'. We present a learning algorithm for semilinear sets with such queries. Although the algorithm is time-consuming for the problem of learning semilinear sets, it is efficient for the problem of learning linear sets. Furthermore, we show exponential lower bounds on the number of queries for various types of queries.

Finally, in Section 7, we apply our results to the problem of learning some parallel computation models and a simple picture recognition model. It is shown that for some matrix languages, commutative languages, and Petri nets, there are subfamilies learnable from positive examples and they are efficiently learnable with queries on inclusions. Also, with an appropriate coding, the results suggest that for each single concept of polygons some recognition device is learnable from positive examples while it is not learnable from positive examples for mixed concepts, and that recognition devices are learnable with queries on inclusions. This matches with our intuition on picture recognitions.

2 Preliminaries

Let $\mathbf{N}$ denote the nonnegative integers. For each integer $k \geq 1$, let $\mathbf{N}^k = \mathbf{N} \times \cdots \times \mathbf{N}$ (k times) and for each $n \in \mathbf{N}$, $n^k = (n, \ldots, n)$. We regard $\mathbf{N}^k$ as a subset of the vector space of all k-tuples of rational numbers over the rational numbers. Thus for elements $\mathbf{u} = (u_1, \ldots, u_k)$ and $\mathbf{v} = (v_1, \ldots, v_k)$ in $\mathbf{N}^k$ and n in $\mathbf{N}$, $\mathbf{u} + \mathbf{v} = (u_1 + v_1, \ldots, u_k + v_k)$, $\mathbf{u} - \mathbf{v} = (u_1 - v_1, \ldots, u_k - v_k)$, and $n\mathbf{u} = (nu_1, \ldots, nu_k)$. We may also speak of the linear dependence and the linear independence of elements of $\mathbf{N}^k$.

Let $\leq$ be the relation on $\mathbf{N}^k$ defined by $\mathbf{u} \leq \mathbf{v}$ for elements $\mathbf{u} = (u_1, \ldots, u_k)$ and $\mathbf{v} = (v_1, \ldots, v_k)$ if and only if $u_i \leq v_i$ for each i. In particular, we shall write $\mathbf{u} < \mathbf{v}$ if $\mathbf{u} \leq \mathbf{v}$ and $\mathbf{u} \neq \mathbf{v}$. The relation $\leq$ is a partial order on $\mathbf{N}^k$. Thus we may speak of minimal elements in a subset of $\mathbf{N}^k$. The condition for two elements $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ in $\mathbf{N}^k$ to be incomparable is the existence of i and j such that $u_i < v_i$ and $u_j > v_j$.

We note a following result in [4] without a formal proof:

Proposition 2.1 *For any subset R of $\mathbf{N}^k$, the set D of minimal elements of R is finite and can be effectively found.*

Given an element $\mathbf{c}$ and a subset P of $\mathbf{N}^k$, let $Q(\mathbf{c}, P)$ denote the set

$$Q(\mathbf{c}, P) = \{\mathbf{q} \mid \mathbf{q} = \mathbf{c} + n_1\mathbf{p}_1 + \cdots + n_r\mathbf{p}_r,\ n_i \in \mathbf{N},\ \mathbf{p}_i \in P\}.$$

$\mathbf{c}$ is called the *constant* and each $\mathbf{p}_i$ is called a *period* of $Q(\mathbf{c}, P)$.

A subset Q of $\mathbf{N}^k$ is said to be *linear* if and only if there exist an element $\mathbf{c}$ and a finite subset P of $\mathbf{N}^k$ such that $Q = Q(\mathbf{c}, P)$. Especially, $Q = Q(\mathbf{c}, P)$ is said to be *simple* if and only if the elements of P are linearly independent. A subset Q of $\mathbf{N}^k$ is said to be *semilinear* if and only if Q is a finite union of linear subsets of $\mathbf{N}^k$. Q is said to be *semi-simple* if and only if Q is a finite *disjoint* union of simple sets.

For any linear set Q, if $Q = Q(\mathbf{c}, P)$ then we call $Q(\mathbf{c}, P)$ a *description of* Q. Let $Q = Q_1 \cup \cdots \cup Q_n$ be a semilinear set such that for each linear set Q_i $(1 \leq i \leq n)$, $Q(\mathbf{c}_i, P_i)$ is a description of Q_i. Then, we denote a description of Q by $Q(\mathbf{c}_1, P_1) \cup \cdots \cup Q(\mathbf{c}_n, P_n)$. We note that any linear set, and therefore, any semilinear set might have more than one descriptions in terms of constants and periods. Therefore, we should distinguish between semilinear sets themselves and descriptions of them.

Definition For any positive integer n, a subset Q of $\mathbf{N}^k$ is said to be *n-linear* if and only if Q is a union of exactly n number of linear subsets of $\mathbf{N}^k$ and there is no $i < n$ such that Q is a union of i number of linear subsets of $\mathbf{N}^k$.

Clearly, any linear set is 1-linear and any semilinear set is n-linear for some finite n.

We shall also consider computational complexities of learning. We use the definitions of deterministic and nondeterministic polynomial time computability and reducibility, of classes P and NP, and of NP-hardness and NP-completeness as described in [5].

3 Properties of Semilinear Sets

In this section, we note some basic properties of semilinear sets. These properties shall play important roles in the problem of learning semilinear sets.

Boolean operations and equivalence At first, we summarize the closure properties on Boolean operations and the properties on the inclusion relation of semilinear sets. The reader may find formal proofs of them in [6], for example.

Proposition 3.1 *The family of semilinear subsets of* $\mathbf{N}^k$ *is closed under union, intersection, and complement.*

Corollary 3.2 *It is effectively solvable to determine for arbitrary semilinear sets* Q_1 *and* Q_2*, whether* (1) $Q_1 \subseteq Q_2$, (2) $Q_1 = Q_2$.

Characteristic sets and canonical descriptions A finite subset E of a semilinear set Q is said to be *descriptive* for Q if and only if there is a description $Q(\mathbf{c}_1, P_1) \cup \cdots \cup Q(\mathbf{c}_n, P_n)$ of Q such that E includes the set $\bigcup_{i=1}^{n}(\{\mathbf{c}_i\} \cup \{\mathbf{c}_i + \mathbf{p} \mid \mathbf{p} \in P_i\})$.

Definition Let Q be a semilinear set. A *characteristic set* of Q is a finite subset $C(Q)$ of Q such that

1. $C(Q)$ is descriptive for Q, and
2. for any proper subset E of $C(Q)$, E is not descriptive for Q.

Proposition 3.3 *For any linear set* Q*, the characteristic set* $C(Q)$ *of* Q *is unique and can be effectively found from* Q.

Proof. Since Q is a linear set, there exists the unique minimal element $\mathbf{c}$ of Q. Let $P_0 = \emptyset$, $E_0 = \{\mathbf{c}\}$, and $i = 1$. Repeat the following procedure: Let D be a set of minimal elements of $Q - Q(\mathbf{c}, P_{i-1})$. D is finite and can be effectively found. Then, let $P_i = P_{i-1} \cup \{\mathbf{d} - \mathbf{c} \mid \mathbf{d} \in D\}$ and $E_i = E_{i-1} \cup D$. If $Q = Q(\mathbf{c}, P_i)$ then let $C(Q) = E_i$ and halt. Otherwise, continue the step $i+1$. We note that the equivalence problem of semilinear sets is effectively solvable.

Since Q is a linear set, this procedure halts and outputs a finite set $C(Q)$. The construction of this procedure ensures that $C(Q)$ is descriptive for Q.

Clearly, any description of Q has as a constant the unique minimal element $\mathbf{c}$ of Q. On each step i ($i \geq 1$), for any element $\mathbf{d} \in D$, $\mathbf{d} - \mathbf{c}$ must be a period of any description of Q. Otherwise, there are some finite subset $R = \{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$ of E_i such that $\mathbf{r}_i \leq \mathbf{d}$ for any $\mathbf{r}_i \in R$ and $\mathbf{d} = n_1(\mathbf{r}_1 - \mathbf{c}) + \cdots + n_m(\mathbf{r}_m - \mathbf{c})$ for positive integers $n_1, \ldots, n_m$. However, the construction of the procedure ensures that $R \subseteq Q(\mathbf{c}, P_{i-1})$, so $\mathbf{d} \notin D$, contradiction. Therefore, for any element $\mathbf{q} \in C(Q) - \{\mathbf{c}\}$, $\mathbf{q} - \mathbf{c}$ must be a period of any description of Q. Hence, any descriptive subset E for Q must contain all elements of $C(Q)$. This completes the proof. □

Definition A description $Q(\mathbf{c}_1, P_1) \cup \cdots \cup Q(\mathbf{c}_n, P_n)$ of a semilinear set Q is said to be *canonical* if and only if the set $\bigcup_{i=1}^{n}(\{\mathbf{c}_i\} \cup \{\mathbf{c}_i + \mathbf{p} \mid \mathbf{p} \in P_i\})$ is the characteristic set of Q.

In particular, for any linear set, we have the following proposition:

Proposition 3.4 *For any linear set Q, a description $Q(\mathbf{c}, P)$ of Q is canonical if and only if each period is not linear sum of the other periods.*

Proof. Let $Q(\mathbf{c}, P)$ be a description of a linear set Q and let $C(Q)$ be the set $\{\mathbf{c}\} \cup \{\mathbf{c} + \mathbf{p} \mid \mathbf{p} \in P\}$. Then, since the constant $\mathbf{c}$ is the unique minimum element of Q and P is a finite subset of $\mathbf{N}^k$, the set $C(Q)$ is the characteristic set of Q if and only if each period is not linear sum of the other periods. □

From Propositions 3.3 and 3.4, for any linear set Q, a canonical description $Q(\mathbf{c}, P)$ is unique and is effectively found from any description of Q. However, there exists a semilinear set such that a characteristic set is not unique, therefore, a canonical description is not unique. For example, consider two semilinear subsets Q_1 and Q_2 of $\mathbf{N}^2$, whose descriptions are $Q((0,0), \emptyset) \cup Q((1,0), \{(1,0),(0,1)\})$ and $Q((0,0), \{(1,0)\}) \cup Q((1,1), \{(1,0),(0,1)\})$, respectively. It is easy to verify that $Q_1 = Q_2$ and sets $C(Q_1) = \{(0,0),(1,0),(2,0),(1,1)\}$ and $C(Q_2) = \{(0,0),(1,0),(1,1),(2,1),(1,2)\}$ are characteristic sets. Therefore, these descriptions are canonical.

We also note that given the characteristic set $C(Q)$ of a linear set Q, the canonical description of Q is effectively found. That is, the constant $\mathbf{c}$ is the unique minimum element of $C(Q)$ and then the set of periods is $\{\mathbf{p}_i \mid \mathbf{q}_i - \mathbf{c},\ \mathbf{q}_i \in C(Q) - \{\mathbf{c}\}\}$.

Time complexity of the membership problem Finally, we show the time complexity of the membership problem for semilinear sets. As we will show later, this plays an important role in the problem of learning them.

The problem is effectively solvable. However, we show that it is computationally intractable.

Theorem 3.5 *For any fixed positive integer k, given a canonical description $Q(\mathbf{c}, P)$ of a linear subset of $\mathbf{N}^k$ and an element $\mathbf{q}$ in $\mathbf{N}^k$, the problem of deciding whether $\mathbf{q} \in Q(\mathbf{c}, P)$ is NP-complete.*

Proof. We denote the membership problem for linear sets in the following way:

LINEAR SET MEMBERSHIP (LM)

INSTANCE: A canonical description $Q(\mathbf{c}, P)$ of a linear subset of $\mathbf{N}^k$ and an element $\mathbf{q}$ of $\mathbf{N}^k$.

QUESTION: Is $\mathbf{q}$ an element in $Q(\mathbf{c}, P)$?

Note that k is a *fixed* positive integer.

Consider the following procedure: Given $Q(\mathbf{c}, P)$ and $\mathbf{q}$,

step 0 let $\mathbf{q}_0 = \mathbf{q} - \mathbf{c}$, $i = 1$,

step i choose a period $\mathbf{p}$ in P, nondeterministically, and let $\mathbf{q}_i = \mathbf{q}_{i-1} - \mathbf{p}$,
if $\mathbf{q}_i = 0^k$, then output *TRUE* and halt,
else go to *step i+1*.

Clearly, there exists a nondeterministic Turing machine which executes the procedure in polynomial time of the size of inputs, and it outputs *TRUE* and halts if and only if $\mathbf{q} \in Q(\mathbf{c}, P)$.

To see that the problem **LM** is NP-hard, consider the following problem:

EXACT COVER (XC)

INSTANCE: A set X and a collection C of subsets of X.

QUESTION: Does C contain an exact cover for X, i.e., a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?

This problem is known as an NP-complete problem (see [8]). We exhibit a polynomial time reduction to **LM** of **XC**.

Let $X = \{x_1, \ldots, x_n\}$ be a set and $C = \{c_1, \ldots, c_m\}$ be a collection of subsets of X. Without loss of generality, we assume that X and C are ordered sets. Given X and C, we constructs a canonical description $Q(0, P)$ of a linear subset of $\mathbf{N}$ and show that C contains an exact cover for X if and only if $\mathbf{q} \in Q(0, P)$, where

$$\mathbf{q} = \sum_{j=1}^{m} 2^{n+m(n-1)+j} + \sum_{i=1}^{n} 2^{i+m(i-1)}.$$

For each $x_i \in X$ and each $c_j \in C$, define

$$g(x_i, c_j) = \begin{cases} 1 & \text{if} \quad x_i \in c_j \\ 0 & \text{if} \quad x_i \notin c_j \end{cases}$$

For each $c_j \in C$, define

$$\mathbf{p}_j = 2^{n+m(n-1)+j} + \sum_{i=1}^{n} g(x_i, c_j) 2^{i+m(i-1)}.$$

Also, we define

$$\mathbf{p}_{m+1} = 2^{n+m(n-1)+1}$$

Then, define

$$P = \{\mathbf{p}_j \,|\, 1 \leq j \leq m+1\}$$

Clearly, the indicated construction of P from X and C is carried out in polynomial time of the numbers of elements of X and C.

We first show that a description $Q(0, P)$ is canonical. Obviously, $\mathbf{p}_{m+1}$ is not a linear sum of the other periods. Assume that a period $\mathbf{p}_j$ is a linear sum of the other periods. Let x_i be an element in c_j. If there exist $i'_1, \ldots, i'_t$ such that $2^{i+m(i-1)} = n'_1 2^{i'_1+m(i'_1-1)} + \cdots + n'_t 2^{i'_t+m(i'_t-1)}$, every $n'_1, \ldots, n'_t$ are positive integers, and every $i'_1, \ldots, i'_t$ are less than i, then $n'_1 + \cdots + n'_t \geq 2^{m+1}$, so $2^{n+m(n-1)+j} < (n'_1 + \cdots + n'_t)2^{n+m(n-1)}$, contradiction. Thus, for any $c_j \in C$, $\mathbf{p}_j$ must not be a linear sum of the other periods. Therefore, from Proposition 3.4, $Q(0, P)$ is canonical.

Suppose that C' is an exact cover for X. Then we define the coefficients of periods as follows: For each $j = 1, \ldots, m$, if $c_j \in C'$ then the coefficient n_j of $\mathbf{p}_j$ is 1, while if $c_j \notin C'$ then $n_j = 0$. Also, we define the coefficient n_{m+1} of the period $\mathbf{p}_{m+1}$ by

$$n_{m+1} = \sum_{c_{j'} \in C - C'} 2^{j'-1}$$

The construction of $Q(0, P)$ and $\mathbf{q}$ ensures that $\mathbf{q} \in Q(0, P)$.

Conversely, suppose that $\mathbf{q} \in Q(0, P)$. As we have shown, for any i $(1 \leq i \leq n)$, $2^{i+m(i-1)} \neq \sum_{l=1}^{i-1} n_l 2^{l+m(l-1)}$ for any $n_1, \ldots, n_i$ less than 2^{m+1}. Therefore, for any i, there exists exactly one period $\mathbf{p}_j$ such that $\mathbf{p}_j$ is constructed from c_j which has x_i and the coefficient of $\mathbf{p}_j$ is 1. Let C' be a set which has c_j such that the coefficient of $\mathbf{p}_j$ is 1. It is easy to verify that C' is an exact cover for X.

Thus, even if $k = 1$, the problem **LM** is NP-hard. This completes the proof. □

Note that if a description $Q(\mathbf{c}_1, P_1) \cup \ldots \cup Q(\mathbf{c}_n, P_n)$ of a semilinear set is canonical, then each description $Q(\mathbf{c}_i, P_i)$ of a linear set is also canonical. Therefore, we have the following straightforward corollary of Theorem 3.5:

Corollary 3.6 *For any fixed positive integer k, given a canonical description of a semilinear subset Q of $\mathbf{N}^k$ and an element $\mathbf{q}$ in $\mathbf{N}^k$, the problem of deciding whether $\mathbf{q} \in Q$ is NP-complete.*

Remark 1 Given a description $Q(\mathbf{c}, P)$ of a simple subset of $\mathbf{N}^k$ and an element $\mathbf{q}$ in $\mathbf{N}^k$, the problem of deciding whether $\mathbf{q} \in Q(\mathbf{c}, P)$ is solvable in polynomial time by the famous elimination method. Therefore, for semi-simple sets, the problem is also solvable in polynomial time.

4 Learnabilities from Positive Examples

In this section, we consider learnabilities of families of semilinear sets from positive examples.

On learning of formal languages, Angluin [1] presented a necessary and sufficient condition for languages to be learnable from positive examples. Note that the Angluin's results require only the recursiveness of languages. Hence, all of them are applicable to the problem of learning recursive sets, straightforwardly. In the sequel, we apply them to the problem of learning semilinear sets.

Let k be a fixed positive integer and R be a nonempty recursive subsets of $\mathbf{N}^k$. We may assume that each nonempty recursive sets has a finite description such as recursive membership functions. Let "+", "−" be special symbols. A *positive example* of R is a pair $(+, \mathbf{p})$ such that $\mathbf{p} \in R$ and a *negative example* of R is a pair $(-, \mathbf{q})$ such that $\mathbf{q} \in \mathbf{N}^k - R$. A *presentation of* R is an infinite sequence $\sigma = s_1, s_2, s_3, \ldots,$ of positive and negative examples such that any element of $\mathbf{N}^k$ appears in σ at least one time. A *positive presentation of* R is an infinite sequence $\sigma = s_1, s_2, s_3, \ldots,$ of positive examples such that any element of R appears in σ at least one time.

A learner is defined to be an effective procedure whose input is a (positive) presentation of a recursive subset R of $\mathbf{N}^k$ and output is a finite or infinite sequence $W_1, W_2, W_3, \ldots$ of finite descriptions of recursive subsets. Each element W_i in an output sequence of M is called a *conjecture of* M.

Let σ be a (positive) presentation of a recursive subset R of $\mathbf{N}^k$ and M be a learner. M is said to *identify* R *from (positive) examples* if and only if for every (positive) presentation σ of R there exists a positive integer n such that W_n is a description of R, and M outputs W_n and halts, or outputs $W_n, W_{n+1}, W_{n+2}, \ldots,$ such that $W_n = W_{n+1} = W_{n+2} \cdots$, forever. In particular, we call the latter identification criterion an *identification in the limit*.

A recursively enumerable family $\mathcal{R}$ of nonempty recursive subsets of $\mathbf{N}^k$ is *learnable from (positive) examples* if and only if there exists a learner which identifies R from (positive) examples for every $R \in \mathcal{R}$.

Condition 1 A recursively enumerable family $\mathcal{R}$ of nonempty recursive subsets of $\mathbf{N}^k$ *satisfies Condition 1* if and only if there exists an effective procedure which on any input $R \in \mathcal{R}$

enumerates a set T such that

1. T is finite,
2. $T \subseteq R$, and
3. for all $R' \in \mathcal{R}$, if $T \subseteq R'$ then R' is not a proper subset of R.

The next lemma shows that Condition 1 is a necessary and sufficient condition for a recursively enumerable family of nonempty recursive subsets of $\mathbf{N}^k$ to be learnable from positive examples.

Lemma 4.1 (Angluin) *A recursively enumerable family of nonempty recursive subsets of $\mathbf{N}^k$ is learnable from positive examples if and only if it satisfies Condition 1.*

The following condition is simply Condition 1 with the requirement of effective enumerability of T dropped.

Condition 2 We say a recursively enumerable family $\mathcal{R}$ of nonempty recursive subsets of $\mathbf{N}^k$ *satisfies Condition 2* provided that, for every $R \in \mathcal{R}$, there exists a finite set $T \subseteq R$ such that for every $R' \in \mathcal{R}$, if $T \subset R'$ then R' is not a proper subset of R.

Lemma 4.2 (Angluin) *If $\mathcal{R}$ is a recursively enumerable family of nonempty recursive subsets of $\mathbf{N}^k$ that is learnable from positive examples, then it satisfies Condition 2.*

This lemma may be used to show that a family of semilinear sets is not learnable from positive examples.

In the rest of this section, we shall show learnabilities of families of semilinear sets based on Angluin's results.

Lemma 4.3 *Let Q be a linear subset of $\mathbf{N}^k$ and $C(Q)$ be the characteristic set of Q. Then, for any linear subset Q' of $\mathbf{N}^k$, if $C(Q) \subseteq Q'$ then $Q \subseteq Q'$.*

Proof. Let $Q(\mathbf{c}, P)$ be the canonical description of Q. Suppose that Q' is a linear subset of $\mathbf{N}^k$ such that $C(Q) \subseteq Q'$ and $Q(\mathbf{c}', \{\mathbf{p}'_1, \ldots, \mathbf{p}'_r\})$ is the canonical description of Q'. Since $C(Q) \subseteq Q'$, for each $\mathbf{q}_j$ of $C(Q)$, $\mathbf{q}_j = \mathbf{c}' + n_{j_1}\mathbf{p}'_1 + \cdots + n_{j_r}\mathbf{p}'_r$. Therefore, for each period $\mathbf{p}_i$ of $Q(\mathbf{c}, P)$, $\mathbf{p}_i = \mathbf{q}_i - \mathbf{c} = (n_{i_1} - n_{c_1})\mathbf{p}'_1 + \cdots + (n_{i_r} - n_{c_r})\mathbf{p}'_r$. Hence, for each $\mathbf{q} \in Q$, there exist $m_1, \ldots, m_r \in \mathbf{N}$ such that $\mathbf{q} = \mathbf{c}' + m_1\mathbf{p}'_1 + \cdots + m_r\mathbf{p}'_r$. $\square$

Theorem 4.4 *For any positive integer k, the family of linear subsets of $\mathbf{N}^k$ is learnable from positive examples.*

Proof. Let $Q(\mathbf{c}_1, P_1), Q(\mathbf{c}_2, P_2), Q(\mathbf{c}_3, P_3), \ldots,$ be an effective enumeration of the canonical descriptions of all linear subsets of $\mathbf{N}^k$. It is obvious that there exists an effective procedure which on any input $i \geq 1$ enumerates a characteristic set C_i of a linear set $Q(\mathbf{c}_i, P_i)$. By definition of characteristic sets of linear sets, C_i is finite and $C_i \subseteq Q(\mathbf{c}_i, P_i)$. Moreover, by Lemma 4.3, for all $j \geq 1$, if $C_i \subseteq Q(\mathbf{c}_j, P_j)$ then $Q(\mathbf{c}_j, P_j)$ is not a proper subset of $Q(\mathbf{c}_i, P_i)$. Therefore, the family satisfies Condition 1 and by Lemma 4.1 the proof is completed. □

Corollary 4.5 *For any positive integer k, the family of simple subsets of $\mathbf{N}^k$ is learnable from positive examples.*

Thus, the family of linear sets is learnable from positive examples. On the other hand, for $k \geq 2$ and $n \geq 2$, the family of n-linear subsets of $\mathbf{N}^k$ is not learnable from positive examples, as shown in the followings:

Lemma 4.6 *For any positive integer $k \geq 2$, the family of 2-linear subsets of $\mathbf{N}^k$ is not learnable from positive examples.*

Proof. At first, we show the case $k = 2$. Consider the 2-linear set $Q = Q_1 \cup Q_2$, where $Q_1 = Q((0,0), \emptyset)$ and $Q_2 = Q((1,1), \{(1,0),(0,1)\})$. Q is a 2-linear subset of $\mathbf{N}^2$ (see [4], for example).

Let $T = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$ be any nonempty finite subset of Q. Consider the 2-linear set $Q^T = Q_1^T \cup Q_2^T$ (cf. Figure 1), where

$$\begin{aligned} Q_1^T &= Q((1,1), \{\mathbf{q}_i - (1,1) \mid \mathbf{q}_i = (1,s) \in T\}) \\ Q_2^T &= Q((0,0), \{\mathbf{q}_j \mid \mathbf{q}_j = (q_1, q_2) \in T,\ q_1 \neq 1\}). \end{aligned}$$

Canonical descriptions of Q_1^T and Q_2^T are effectively found from the above descriptions. Clearly, $T \subseteq Q^T$ and it is easy to verify that $Q^T \subseteq Q$. For each $\mathbf{q}_i \in T$ let $\mathbf{q}_i = (q_{i_1}, q_{i_2})$. Let q_{M_1} be the maximum integer of $q_{1_1}, \ldots, q_{m_1}$. Then, $\mathbf{q}_M = (q_{M_1} + 1, 1)$ is in Q but not in Q^T, so Q^T is a proper subset of Q. Thus Condition 2 fails. The cases $k > 2$ are proved by the similar arguments. □

The following theorem is proved by the trivial extension of the proof of Lemma 4.6.

Theorem 4.7 *For any $k \geq 2$ and any $n \geq 2$, the family of n-linear subsets of $\mathbf{N}^k$ is not learnable from positive examples.*

Proof. Let n be an integer greater than 2. Consider the n-linear set $Q = Q_1 \cup \cdots \cup Q_n$ of $\mathbf{N}^2$, where for i $(1 \leq i \leq n-1)$, $Q_i = Q((i-1,0), \emptyset)$ and $Q_n = Q((n-1,1), \{(1,0),(0,1)\})$. It is easy to verify that Q is an n-linear subset of $\mathbf{N}^2$.

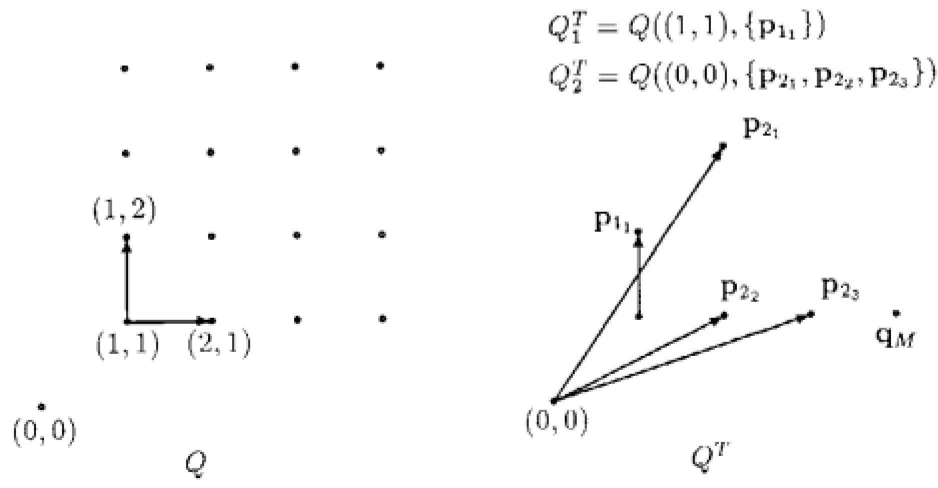


Figure 1: Construction of Q^T

Let $T = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$ be any nonempty finite subset of Q. Consider the n-linear set $Q^T = Q_1^T \cup \cdots \cup Q_n^T$, where

$$\begin{aligned} Q_i^T &= Q((i-1,0),\emptyset) \quad \text{for } 1 \le i \le n-2 \\ Q_{n-1}^T &= Q((n-1,1),\{\mathbf{q}_j - (n-1,1) \mid \mathbf{q}_j = (n-1,s) \in T\}) \\ Q_n^T &= Q((n-2,0),\{\mathbf{q}_j - (n-2,0) \mid \mathbf{q}_j = (q_1,q_2) \in T,\ q_1 \ne n-1\}) \end{aligned}$$

Then, a canonical description of each Q_i^T is effectively found from the above corresponding description. From the proof of Lemma 4.6, it is easy to verify that $T \subseteq Q^T$ and Q^T is a proper subset of Q. Thus Condition 2 fails. □

Corollary 4.8 *For any integer $k \ge 2$, the family of semilinear subsets of $\mathbf{N}^k$ is not learnable from positive examples.*

The questions whether the family of semilinear subsets of $\mathbf{N}$ is learnable from positive examples and whether the family of semi-simple subsets of $\mathbf{N}^k$ is learnable from positive examples are open.

5 A Simple Learning Method for Linear Sets

In this section, we present a learner which identifies any linear set in the limit from positive examples.

Let k be a fixed positive integer and Q be an unknown linear subset of $\mathbf{N}^k$. As described in Section 3, if the characteristic set of Q is found, then the canonical description of Q is

```
Procedure ID1
Input:  A positive presentation s_1, s_2, s_3, ..., of a linear set Q.
Output: A sequence Q_1, Q_2, Q_3, ..., of descriptions.

E_0 := ∅;
P_0 := ∅;
c := 0^k;

For each i ≥ 1 do
    Begin
        Read (+, q_i);
        E_i := E_{i-1} ∪ {q_i};
        If      Q(c, P_{i-1}) is consistent with E_i
        then    P_i := P_{i-1}, output Q(c, P_i), and go to i + 1 step;

        If      found the unique minimum element q of E_i
        then    c := q;
        else    c := 0^k;

        While Q(c, P_i) is not consistent with E_i do
            Begin
                find a minimal element q of E_i − Q(c, P_i);
                P_i := P_i ∪ {q − c};
            End;
        Output a description Q(c, P_i);
        go to i + 1 step;
    End;
```

Figure 2: The learner *ID1*

effectively found. Therefore, the learner *ID1*, illustrated in Figure 2, tries to find the characteristic set from the given examples. *ID1* never changes a conjecture while it is consistent with the given examples, that is, it contains the given examples. When a conjecture is not consistent with the examples, *ID1* constructs a new conjecture.

Lemma 5.1 *Let Q be a linear subset of $\mathbf{N}^k$. Given a finite subset $R(Q)$ of Q which includes the characteristic set of Q, the learner ID1 constructs the canonical description of Q.*

Proof. Since $R(Q)$ includes the characteristic set of Q, *ID1* finds the unique minimum element of it, which is precisely the constant $\mathbf{c}$ of the canonical description of Q. Also, Proposition 3.3 and the construction of *ID1* ensure that *ID1* finds each period $\mathbf{p}_i$ of the canonical description of Q. □

Since for any positive presentation $\sigma = s_1, s_2, s_3, \ldots$, there exists a positive integer i such that the set of given examples in $s_1, s_2, \ldots, s_i$ includes the characteristic set of Q, by Lemma 5.1, we have the following theorem:

Theorem 5.2 *The learner ID1 identifies any linear subset of* $\mathbf{N}^k$ *in the limit from positive examples.*

Remark 2 A learner is said to *make an overgeneral conjecture* provided that in the process it outputs a description of a proper superset of the linear set which should be identified. It is easy to verify that *ID1* never makes overgeneral conjectures. However, *ID1* does not always output descriptions of subsets of the correct linear set. If *ID1* cannot find a constant, then 0^k is assumed to be a constant. Then, the conjecture constructed by *ID1* contains elements not in the correct linear set.

Unfortunately, *ID1* uses membership of examples, which is an NP-complete problem as we have shown, so *ID1* is time-consuming. If there is a polynomial-time algorithm to solve the problem of finding the canonical description of a linear set consistent with the given examples, then we could have a learner which makes a conjecture in polynomial time for each time and identifies any linear set in the limit. However, we give some partial evidence for the difficulty of the case.

Theorem 5.3 *If* $P \neq NP$, *then there is no polynomial-time algorithm to solve the following problem: given a finite subset* E *of* $\mathbf{N}^k$, *find the canonical description* $Q(\mathbf{c}, P)$ *of a linear subset of* $\mathbf{N}^k$ *which contains all elements of* E.

Proof. Suppose that there exists an algorithm AF that runs in polynomial time and is such that for any subset E of $\mathbf{N}^k$, AF on input E outputs the canonical description $Q(\mathbf{c}, P)$ of a linear subset of $\mathbf{N}^k$ which contains all elements of E. We shall use AF to construct a polynomial-time algorithm to decide whether $\mathbf{q} \in Q(\mathbf{c}, P)$ for an arbitrary element $\mathbf{q} \in \mathbf{N}^k$ and the canonical description $Q(\mathbf{c}, P)$. Since this latter problem is NP-complete shown in Theorem 3.5, this will imply $P = NP$, proving the theorem.

Let $\mathbf{q}$ be an element in $\mathbf{N}^k$ and $Q(\mathbf{c}, P)$ be the canonical description of a linear subset of $\mathbf{N}^k$. We may construct the characteristic set C of $Q(\mathbf{c}, P)$ in polynomial time. Run AF on input $C \cup \{\mathbf{q}\}$ and denote the output by $Q(\mathbf{c}', P')$. Since the canonical description is unique for any linear set, if $\mathbf{c}' = \mathbf{c}$ and $P = P'$ then $\mathbf{q} \in Q(\mathbf{c}, P)$, otherwise, $\mathbf{q} \notin Q(\mathbf{c}, P)$. We may test whether $\mathbf{c} = \mathbf{c}'$ and $P = P'$ in polynomial time, we complete the proof. □

Remark 3 All processes of *ID1* other than membership are done in polynomial time of the size of inputs.

Let Q be an unknown linear subset of $\mathbf{N}^k$. For each time $i \geq 1$, let $(+, \mathbf{q}_1), \ldots, (+, \mathbf{q}_i)$ be a finite subsequence of a positive presentation and $E = \{\mathbf{q}_1, \ldots, \mathbf{q}_i\}$. Also, let m be the maximum integer appearing in the elements of E. Then, since E has at most i elements, a unique minimum element of E, if there exists, is found in polynomial time of i, k, and m. On the other hand, **While** loop is executed at most $i - 1$ times. In each loop, for each $\mathbf{q} \in E - Q(\mathbf{c}, P_i)$, a period is computable in polynomial time of i and k. Therefore, a description of a linear set Q is constructed in polynomial time of i, k, and m. Hence, all processes other than membership are done in polynomial time of i, k and m.

Consider the family of simple subsets of $\mathbf{N}^k$. This family is also learnable from positive examples by Corollary 4.5. As we have noted above, the membership problem for simple sets is solvable in polynomial time, and from Remark 3, all processes of *ID1* other than membership are done in polynomial time of the size of inputs. Therefore, in this case, *ID1* might construct a description in polynomial time for each time i. If the correct constant is given, *ID1* constructs a description of a simple set, so *ID1* decides membership of each examples correctly. However, if the correct constant is not given, then the constructed description might not represent a simple set and might decide membership incorrectly. Therefore, if a new minimum element is found, a learner should reconstruct a conjecture, so *ID1* must check the minimum element before checking consistency. Then, it is easy to verify that *ID1* identifies any simple sets in the limit from positive examples. Hence, we have the following:

Theorem 5.4 *For the family of simple subsets of* $\mathbf{N}^k$, *there exists a learner which, for each time* i $(i \geq 1)$, *constructs a canonical description in polynomial time of* i, k *and* m, *where* m *is the maximum integer appearing in the given examples.*

6 Learning Semilinear Sets with Queries

In this section, we consider the problem of learning semilinear sets with queries. In previous sections, we had no assumption on presentations of examples. In this time, we assume that there exists a teacher who can answer questions of a learner and the learner get informations from the teacher.

We consider the following types of learners' queries: Let Q be an unknown semilinear subset of $\mathbf{N}^k$ and Q_i be a conjecture of a learner.

- *Membership.* A learner asks whether $\mathbf{q} \in Q$ for any $\mathbf{q} \in \mathbf{N}^k$ and a teacher answers *yes* if $\mathbf{q} \in Q$ and *no* if $\mathbf{q} \notin Q$.

- *Equivalence*. A learner asks whether $Q = Q_i$ for any conjecture Q_i and a teacher answers *yes* if $Q = Q_i$ and *no* if $Q \neq Q_i$. If the answer is *no*, the teacher also gives the learner an element $\mathbf{q} \in (Q - Q_i) \cup (Q_i - Q)$.
- *Subset*. A learner asks whether $Q_i \subseteq Q$ for any conjecture Q_i and a teacher answers *yes* if $Q_i \subseteq Q$ and *no* otherwise. If the answer is *no*, the teacher also gives the learner an element $\mathbf{q} \in Q_i - Q$.
- *Superset*. A learner asks whether $Q \subseteq Q_i$ for any conjecture Q_i and a teacher answers *yes* if $Q \subseteq Q_i$ and *no* otherwise. If the answer is *no*, the teacher also gives the learner an element $\mathbf{q} \in Q - Q_i$.

For the queries other than membership, the returned element is called a *counterexample*. We shall also consider *restricted* versions of equivalence, subset, and superset queries, for which the answers are just *yes* and *no*, with no counterexample provided.

A learner with restricted subset and restricted superset queries We first show that if *restricted subset* and *restricted superset* queries are available, there exists a learner which identifies any semilinear subset of $\mathbf{N}^k$ and halts.

Let Q be a semilinear set. For each i $(0 \leq i)$, define D_i recursively:

1. $D_0 = \emptyset$,
2. $D_i = \{\mathbf{q} \,|\, \mathbf{q}$ is a minimal element of $Q - \bigcup_{j=0}^{i-1} D_j\}$.

Then, each D_i is finite and for each distinct i and j, D_i and D_j are disjoint.

Definition A *representative set* of a semilinear set Q is a finite subset $R(Q) = \bigcup_{i=0}^{t} D_i$ of Q such that

1. $R(Q)$ is descriptive for Q, and
2. for any nonnegative integer s such that $s < t$, $\bigcup_{j=0}^{s} D_j$ is not descriptive for Q.

Proposition 6.1 *For any semilinear set Q, the representative set $R(Q)$ of Q is unique and can be effectively found.*

Proof. Let $W = \emptyset$, $E_0 = \emptyset$, and $i = 1$. Repeat the following procedure: Let D be a set of minimal elements of $Q - E_{i-1}$ and let $E_i = E_{i-1} \cup D$. From Proposition 2.1, D is finite and can be effectively found. For each $\mathbf{q} \in D$,

1. for each $Q(\mathbf{c}_j, P_j)$ in W, if $\mathbf{c} < \mathbf{q}$ and $Q(\mathbf{c}, P_j \cup \{\mathbf{q}-\mathbf{c}\}) \subseteq Q$, then add $Q(\mathbf{c}_j, P_j \cup \{\mathbf{q}-\mathbf{c}\})$ to W,
2. add $Q(\mathbf{q}, \emptyset)$ to W.

Let W be a set of descriptions of linear sets obtained with the above modifications. If $\bigcup_{Q(\mathbf{c}_j,P_j)\in W} Q(\mathbf{c}_j, P_j)$ is a description of Q, then let $R(Q) = E_i$ and halt. Otherwise, continue the step $i+1$.

We note that the inclusion problem is effectively solvable for semilinear sets.

On each step i, the construction of the procedure ensures that for any linear subset Q_L of Q for which E_i is descriptive, W has a description of Q_L. Therefore, there exists some t such that Q_t is a descriptive for Q, so the procedure outputs $E = E_t$ and halts. Then, obviously, for any s such that $s < t$, E_s is not descriptive for Q. □

We note that a description constructed by the procedure in the proof of Proposition 6.1 might have descriptions $Q(\mathbf{c}_1, P_1)$ and $Q(\mathbf{c}_2, P_2)$ of linear sets such that $P_1 \subseteq P_2$. Then, $Q(\mathbf{c}_1, P_1)$ is redundant. We can effectively eliminate such redundant descriptions of linear sets.

The learner for semilinear sets, described in the following, identifies a semilinear set based on the procedure described in the proof of Proposition 6.1.

Let Q be an unknown semilinear subset of $\mathbf{N}^k$. We denote by $\mathbf{e}[i]$ an element of $\mathbf{N}^k$ which has 1 as the value of ith coordinate and 0 as the values of the other coordinates, and denote by P_e the set $\{\mathbf{e}[i] \mid 1 \le i \le k\}$.

Let Q' be any proper semilinear subset of Q. On input Q', the algorithm *FP*, illustrated in Figure 3, finds a set of the minimal elements of $Q - Q'$.

The algorithm *FP* begins queries whether $Q \subseteq (E \cup U \cup Q(\mathbf{q}+\mathbf{e}[1], P_e))$ with $U = \emptyset$ and $\mathbf{q} = 0^k$ (cf. Figure 4 (a)). For each i ($1 \le i \le k$), *FP* continues queries until ith value of $\mathbf{q}$ is equal to the minimum ith value in the minimal elements which have not been found yet (cf. Figure 4 (b), where $i = 1$). Then, *FP* adds $Q(\mathbf{q}+\mathbf{e}[i], P_e)$ to U and continues queries for $i+1$ coordinate (cf. Figure 4 (c), where $i = 1$). This U guarantees that any minimal element whose value of ith coordinate is greater than the one of $\mathbf{q}$ is contained in U. In this way, *FP* finds the minimal elements.

Lemma 6.2 *Let Q be a semilinear subset of $\mathbf{N}^k$ and Q' be a proper semilinear subset of Q. On input Q', the algorithm FP makes at most $nk(m+1)$ queries and outputs a finite set D of minimal elements of $Q - Q'$, where m is the maximum integer appearing in the elements of D and n is the cardinality of D.*

Algorithm *FP*
Input: A description of a semilinear subset Q' of $\mathbf{N}^k$.
Output: A finite subset D of $\mathbf{N}^k$.
Query: Restricted superset queries.

$D := \emptyset$;
$E := Q'$;

Do
    Begin
        $i := 1$;
        $\mathbf{q} := 0^k$;
        $U := \emptyset$;
        While $i \leq k$ **do**
            Begin
                Ask the teacher whether $Q \subseteq (E \cup U \cup Q(\mathbf{q} + \mathbf{e}[i], P_e))$;
                If the answer is *no*
                then $U := U \cup Q(\mathbf{q} + \mathbf{e}[i], P_e)$;
                $i := i + 1$;
                else $\mathbf{q} := \mathbf{q} + \mathbf{e}[i]$;
            End;
        $D := D \cup \{\mathbf{q}\}$;
        $E := E \cup Q(\mathbf{q}, P_e)$;
    End;
Until the teacher answers *yes* to the query $Q \subseteq E$;

Output D and halt;

Figure 3: The algorithm *FP*

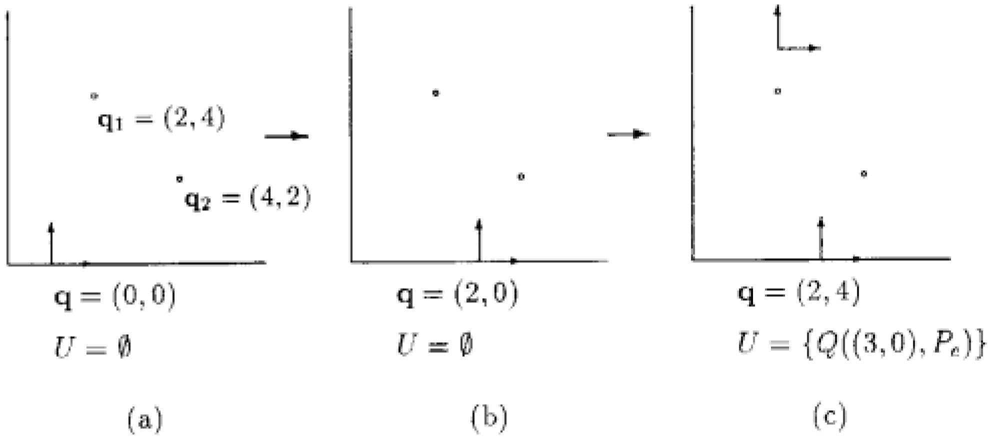


Figure 4: How *FP* finds minimal elements

Proof. Let $\prec$ be the relation on $\mathbf{N}^k$ defined as follows: Let $\mathbf{u} = (u_1, \ldots, u_k)$ and $\mathbf{v} = (v_1, \ldots, v_k)$ be elements of $\mathbf{N}^k$. Suppose that i is the minimum index such that $u_j = v_j$ for any j less than i. Then, $\mathbf{u} \prec \mathbf{v}$ if and only if $u_i < v_i$. The relation $\prec$ is a lexicographical order on $\mathbf{N}^k$.

Assume that $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_m\}$ is a totally ordered set with respect to $\prec$. We shall show that Algorithm *FP* finds all elements of D from $\mathbf{d}_1$ to $\mathbf{d}_m$.

Suppose that $\mathbf{d} = (d_1, \ldots, d_k)$ be a minimum element of D which is not found yet. At first, we show that in the **While** loop, for each i $(1 \le i \le k)$, the teacher answers *no* to the query $Q \subseteq (E \cup U \cup Q(\mathbf{d}_i, P_e))$ if and only if $\mathbf{d}_i = (d_1, \ldots, d_{i-1}, d_i + 1, 0, \ldots, 0)$.

Let $\mathbf{q} = (q_1, \ldots, q_i, 0, \ldots, 0)$ be any element of $\mathbf{N}^k$ such that $q_j \le d_j$ for each j $(1 \le j \le i)$. Furthermore, let $\mathbf{r} = (r_1, \ldots, r_k)$ be any element of $Q - E$. There are following three cases of $\mathbf{r}$ which we should consider:

1. $\mathbf{q} > \mathbf{r}$,
2. $\mathbf{q} \le \mathbf{r}$, and
3. $\mathbf{q}$ and $\mathbf{r}$ are incomparable.

Since $\mathbf{d}$ is a minimum element of $Q - E$, there is no element $\mathbf{r}$ of $Q - E$ such that $\mathbf{q} > \mathbf{r}$. If $\mathbf{q} \le \mathbf{r}$, then $\mathbf{r} \in Q(\mathbf{q}, P_e)$. If $\mathbf{q}$ and $\mathbf{r}$ are incomparable, then there is some t $(1 \le t < i)$ such that $r_t < q_t$ and $q_s \le r_s$ for any s $(s < t)$ (We note that any minimal element $\mathbf{d}'$ such that $\mathbf{d}' \prec \mathbf{d}$ is already found). Then, from the construction of *FP*, U should already include a linear set $Q((c_1, \ldots, c_{t-1}, 0, \ldots, 0), P_e)$ such that $c_j \le r_j$ for each $j < t$, so $\mathbf{r} \in U$. Therefore, for any $\mathbf{q} = (q_1, \ldots, q_i, 0, \ldots, 0)$ such that $q_j \leq d_j$ for each j $(1 \leq j \leq i)$, the teacher must answer *yes* for a query $Q \subseteq (E \cup U \cup Q(\mathbf{q}, P_e))$.

From the assumption, Q' does not contain $\mathbf{d}$ and, from the construction of *FP*, any linear set added to U by *FP* only contains elements which is incomparable to $\mathbf{d}$. Therefore, for each i, the teacher must answer *no* to the query $Q \subseteq (E \cup U \cup Q(\mathbf{d}_i, P_e))$ if and only if $\mathbf{d}_i = (d_1, \ldots, d_{i-1}, d_i+1, 0, \ldots, 0)$, so the algorithm *FP* finds $\mathbf{d}$. Since $Q' \cup \{Q(\mathbf{d}, P_e) \mid \mathbf{d} \in D\}$ contains all elements of Q, when all elements of D are found, the teacher must answer *yes*, so *FP* outputs D and halts.

For each i $(1 \leq i \leq k)$, *FP* makes queries at most $m+1$ times, so for each element of D, *FP* makes queries at most $k(m+1)$ times. In the sequel, *FP* makes at most $nk(m+1)$ queries. This completes the proof. □

The learning algorithm *IDWQ*, illustrated in Figure 5, runs Algorithm *FP* repeatedly, finds minimal elements which have not been found yet, and constructs a description in the same way described in the proof of Proposition 6.1.

Theorem 6.3 *Given any teacher who answers restricted subset and restricted superset queries for any semilinear subset of* $\mathbf{N}^k$*, then the algorithm IDWQ outputs a description of an unknown semilinear set* Q *and halts.*

Proof. Let $R(Q) = \bigcup_{i=0}^{t} D_i$ be the representative set of an unknown semilinear set Q. By running Algorithm *FP* repeatedly, with Lemma 6.2, *IDWQ* finds each D_i and in the sequel, finds $R(Q)$. Then, by Proposition 6.1, *IDWQ* constructs a description of Q, so the teacher must answer *yes* for the query whether $Q \subseteq \bigcup_{Q(\mathbf{c}_i, P_i) \in W} Q(\mathbf{c}_i, P_i)$. This completes the proof. □

We note that a constructed description may have redundant descriptions of linear sets. However, such descriptions can be removed in the obvious way with restricted subset queries.

Thus, the learner *IDWQ* identifies any semilinear subset of $\mathbf{N}^k$ and halts with restricted subset and restricted superset queries. However, *IDWQ* is time-consuming. Let n be the cardinality of the representative set $R(Q)$ of Q. Then, *IDWQ* makes $n2^{n-1}$ number of conjectures in the worst case. Therefore, the total running time of *IDWQ* is bounded by an exponential of k, m and n, where m is the maximum integer appearing in $R(Q)$.

In the case of learning linear sets, subset queries may not be needed. Let Q be an unknown linear set. At first, on input the empty set, *FP* outputs a constant $\mathbf{c}$ of Q. Given a description of a linear subset Q' of Q instead of a finite set of elements of Q, *FP* outputs a finite subset D of minimal elements in $Q - Q'$. It is easy to verify that $\mathbf{d} - \mathbf{c}$ must be a

Algorithm *IDWQ*
Output: A description of an unknown semilinear subset Q of $\mathbf{N}^k$.
Query: Restricted subset and restricted superset queries.

$W := \emptyset$;
$R(Q) := \emptyset$;

While the teacher replies *no* to a query whether $Q \subseteq \bigcup_{Q(\mathbf{c}_i, P_i) \in W} Q(\mathbf{c}_i, P_i)$ **do**

Begin
Run Algorithm *FP* on input $R(Q)$ and get an output D;
$R(Q) := R(Q) \cup D$;
While D is not empty **do**
Begin
let $\mathbf{d}$ be an element in D;
For each $Q(\mathbf{c}_i, P_i)$ of W such that $\mathbf{c}_i < \mathbf{d}$ **do**
Begin
Ask the teacher whether $Q(\mathbf{c}_i, P_i \cup \{\mathbf{d} - \mathbf{c}_i\}) \subseteq Q$;
If the answer is *yes*
then $W := W \cup \{Q(\mathbf{c}_i, P_i \cup \{\mathbf{d} - \mathbf{c}_i\})\}$;
End;
$W := W \cup Q(\mathbf{d}, \emptyset)$;
End;
End;

Output W and halt;

Figure 5: The algorithm *IDWQ*

period of Q for each $\mathbf{d} \in D$ and, in the sequel, *IDWQ* finds a description of Q. Then, the set of all elements found by *FP* is the characteristic set of Q, so the found description is canonical. Therefore, we have the following:

Corollary 6.4 *Let Q be an unknown linear subset of $\mathbf{N}^k$. Given any teacher who answers restricted superset queries $Q \subseteq Q'$ for any* semilinear *subset Q' of $\mathbf{N}^k$, then the algorithm IDWQ outputs a canonical description of an unknown linear set Q and halts. The total running time of IDWQ is bounded by a polynomial in k, m, and n, where m is the maximum integer appearing in the characteristic set $C(Q)$ of Q and n is the cardinality of $C(Q)$.*

Lower bounds for queries In [2], Angluin has presented lower bound techniques for queries. Based on her techniques, we show exponential lower bounds on the number of queries needed for learning semilinear sets using equivalence, membership, and subset queries.

Let C be a finite set of all elements in $\mathbf{N}^k$ such that a value of each coordinate is 0 or 1. Let $\mathcal{S} = \{Q(\mathbf{c}, \emptyset) \mid \mathbf{c} \in C\}$ be a subfamily of linear subsets of $\mathbf{N}^k$. Then, clearly, $|C| = |\mathcal{S}| = 2^k$ and each linear set in $\mathcal{S}$ is disjoint.

Theorem 6.5 *Any algorithm that identifies any semilinear subset of $\mathbf{N}^k$ and halts with membership, restricted equivalence, and subset queries must make at least $2^k - 1$ queries in the worst case.*

Proof. Consider the following teacher: For a restricted equivalence query with the conjecture Q_i, the answer is *no*, and the (at most one) Q_j such that $Q_i = Q_j$ is removed from $\mathcal{S}$. For a membership query with the element $\mathbf{q}$, the answer is *no*, and the (at most one) Q_j such that $\mathbf{q} \in Q_j$ is removed from $\mathcal{S}$. For a subset query with the conjecture Q_i, if $Q_i = \emptyset$ then the answer is *yes*. Otherwise, the answer is *no* and any element $\mathbf{q}$ in $\mathbf{N}^k$ is selected as the counterexample. The (at most one) element Q_j such that $\mathbf{q} \in Q_j$ is removed from $\mathcal{S}$.

At any point, for each $Q_i \in \mathcal{S}$, Q_i is compatible with the answers to the queries so far. An algorithm which identifies a semilinear sets and halts must reduce the cardinality of $\mathcal{S}$ at most one. Each query removes at most one element from the set $\mathcal{S}$, so $2^k - 1$ queries are required in the worst case. $\square$

Since the empty set is not a linear set, with a minor modification of the proof of Theorem 6.5, we have the following:

Theorem 6.6 *Any algorithm that identifies any linear subset of $\mathbf{N}^k$ and halts with membership, equivalence, and subset queries must make at least $2^k - 1$ queries in the worst case.*

Proof. The proof of Theorem 6.5 may be modified as follows: The answers to queries are the same, except that a counterexample must be provided when an equivalence query is answered *no*. Let Q_i be a conjecture. Since $\emptyset$ is not a linear set, $Q_i \neq \emptyset$. The counterexample is any element $\mathbf{q}$ in Q_i. The (at most one) element Q_i is removed from $\mathcal{S}$. □

7 Applications to Parallel Computation Models

Semilinear sets are closely related to some parallel computation models via Parikh mappings. For examples, image sets on Parikh mappings of equal matrix languages [12], simple matrix languages [7], and weakly persistent Petri nets [13] are semilinear sets. In this section, we consider the problem of learning these models based on our methods described in the above.

Strictly bounded equal matrix languages Let Σ be an *alphabet*, i.e., a finite set of symbols and Σ^* be the set of all strings over Σ containing the null string λ. For a string w, $w^0 = \lambda$ and $w^i = w^{i-1}w$ for each integer $i \geq 1$, and $w^* = \{w^i \mid i \geq 0\}$.

A *language* L over Σ is a subset of Σ^*.

Definition A language L over an alphabet Σ is said to be *strictly bounded* if and only if $L \subseteq a_1{}^* \cdots a_k{}^*$ where $\Sigma = \{a_1, \ldots, a_k\}$.

In general, a language L over Σ is said to be *bounded* if and only if there exist words $w_1, \ldots, w_k \in \Sigma^*$ such that $L \subseteq w_1^* \cdots w_k^*$.

Definition An *equal matrix grammar* (abbreviated *EMG*) *of order* k is a 4-tuple $G = (N, \Sigma, \Pi, S)$, where

1. S is the *initial symbol*.
2. N is a finite nonempty set consisting of k-tuples $(A_1, A_2, \ldots, A_k)$, called a *nonterminal*, such that for any pair $(A_1, A_2, \ldots, A_k)$ and $(B_1, B_2, \ldots, B_k)$ of N, $\{A_1, A_2, \ldots, A_k\} \cap \{B_1, B_2, \ldots, B_k\} = \emptyset$.
3. Π is a finite nonempty set consisting of the following types of *matrix rules*:
   (a) $[S \to w_1A_1w_2A_2 \cdots w_kA_k]$,
   (b) $[A_1 \to w_1B_1,\ A_2 \to w_2B_2,\ \ldots A_k, \to w_kB_k]$,
   (c) $[A_1 \to w_1,\ A_2 \to w_2,\ \ldots,\ A_k \to w_k]$,

   where $w_1, w_2, \ldots, w_k \in \Sigma^*$, S is the initial symbol, and $(A_1, A_2, \ldots, A_k)$, $(B_1, B_2, \ldots, B_k)$ are nonterminals.

An *equal matrix grammar* is an *EMG* of any *finite* order k.

We denote $\Sigma \cup N \cup \{S\}$ by V.

Let $G = (N, \Sigma, \Pi, S)$ be an *EMG* of order k. We define the relation $\Longrightarrow$ between strings in V^*. For any $x, y \in V^*$, $x \Longrightarrow y$ if and only if

1. x is the initial symbol S and the initial matrix rule $[S \to y]$ is in Π, or
2. there exist strings $u_1, \ldots, u_k$, $v_1, \ldots, v_k$ over Σ such that $x = u_1 A_1 v_1 \cdots u_k A_k v_k$, $y = u_1 z_1 v_1 \cdots u_k z_k v_k$, and the matrix rule $[A_1 \to z_1, \ldots, A_k \to z_k]$ in Π.

For any $x, y \in V^*$, we write $x \stackrel{*}{\Longrightarrow} y$ if either $x = y$ or there exist $x_0, \ldots, x_n \in V^*$ such that $x = x_0$, $y = x_n$, and $x_i \Longrightarrow x_{i+1}$ for each i. The sequence $x_0, \ldots, x_n$ is called a *derivation* (from x_0 to x_n) and is denoted by

$$x_0 \Longrightarrow \cdots \Longrightarrow x_n.$$

The *language generated by* G, denoted $L(G)$, is the set

$$L(G) = \{w \in \Sigma^* \mid S \stackrel{*}{\Longrightarrow} w\}.$$

Definition A language L is said to be an *equal matrix language* (abbreviated *EML*) if and only if there exists an *EMG* G such that $L = L(G)$ holds.

The family of *EMLs* contains some context-sensitive languages. For example, the context-sensitive language $\{a^n b^n c^n \mid n \geq 1\}$ is an *EML*. Also, there exists a context-free language which is not an *EML* (Ibarra [7]). For example, consider the language $L = \bigcup_{i \geq 0} \{a^n b^n \mid n \geq 1\}^i$. L is a context-free language but it is not an *EML*.

We shall consider the learning problem for a *strictly bounded equal matrix language* (abbreviated *SBEML*). Again, the family of *SBEMLs* contains some context-sensitive languages and there exists a context-free language not in the family.

The Parikh mapping defined as follows connects *EMLs* with semilinear subsets of $\mathbf{N}^k$.

Definition Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet. *The Parikh mapping* $\psi_{(a_1,\ldots,a_k)}$ or ψ when $(a_1, \ldots, a_k)$ is understood, is the function from Σ^* into $\mathbf{N}^k$ defined by $\psi(w) = (\#_{a_1}(w), \ldots, \#_{a_k}(w))$, where $\#_{a_i}(w)$ is the number of occurrences of a_i in w.

Thus $\psi(\lambda) = 0^k$ and $\psi(w_1 \cdots w_n) = \sum_{i=1}^{n} \psi(w_i)$ for each $w_i \in \Sigma^*$. We call $\psi(L) = \{\psi(w) \mid w \in L\}$ *the Parikh set of an EML* L.

The following theorem is due to Siromoney [12]:

Theorem 7.1 (Siromoney) *Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet. For any strongly bounded language L over Σ, L is generated by an EMG G of order k if and only if the Parikh set of L is a semilinear subset Q of $\mathbf{N}^k$. Moreover, an EMG G is effectively found from a description of Q and vice versa.*

For any semilinear set Q, an *EMG G* which generates an *SBEML* is effectively constructed from a description of Q in the following manner: It is enough to show the case that Q is a linear set. Let $Q(\mathbf{c}, \{\mathbf{p}_1, \ldots, \mathbf{p}_r\})$ be a description of the linear set Q. Also, let $\mathbf{c} = (c_1, \ldots, c_k)$ and $\mathbf{p}_i = (p_i^1, \ldots, p_i^k)$. Then $G = (N, \Sigma, \Pi, S)$ where $\Sigma = \{a_1, \ldots, a_k\}$, $N = \{(A_1, \ldots, A_k)\}$, and Π consists of the following matrix rules:

$$[S \rightarrow a_1^{c_1} A_1 \cdots a_k^{c_k} A_k]$$
$$[A_1 \rightarrow \lambda, \ldots, A_k \rightarrow \lambda]$$
$$[A_1 \rightarrow a_1^{p_i^1} A_1, \ldots, A_k \rightarrow a_k^{p_i^k} A_k] \qquad \text{for each } i \ (1 \leq i \leq r).$$

From Theorem 7.1, we may regard the learning problem for *SBEMLs* as the learning problem for semilinear sets.

From these, we can consider meaningful subfamilies of *SBEMLs*:

Definition For each positive integer n, an *SBEML L* is said to be n-*SBEML* if and only if $\psi(L)$ is an n-linear set.

Thus, a 1-*SBEML* is an *SBEML* whose Parikh set is a linear set and an n-*SBEML* is an *SBEML* whose Parikh set is an n-linear set.

Consider the problem of learning *SBEMLs*. In this case, a learner should find an *EMG* which is consistent with the given strings. As described above, via a Parikh mapping, an element of $\mathbf{N}^k$ can be constructed from a given string and an *EMG* can also be constructed from a description of semilinear subset of $\mathbf{N}^k$. Therefore, for the Parikh mapping ψ,

$$\psi^{-1}(Q(\mathbf{c}_1, P_1)), \psi^{-1}(Q(\mathbf{c}_2, P_2)), \psi^{-1}(Q(\mathbf{c}_3, P_3)), \ldots,$$

is an indexed family of 1-linear *SBEMLs*. Then, from Theorems 7.1 and 4.4, the family of 1-*SBEMLs* is learnable from positive examples. On the other hand, from Theorem 4.7 and 7.1, for each positive integer n such that $n > 1$, the family of n-*SBEMLs* is not learnable from positive examples.

Also, from Theorem 6.3, given any teacher who answers restricted subset and restricted superset queries for any n-*SBEML*, then there exists an algorithm which identifies any n-*SBEML* and halts. Furthermore, from Corollary 6.4, given any teacher who answers restricted superset queries for any n-*SBEML*, then there exists an algorithm which identifies any 1-*SBEML* in polynomial time and halts.

Small classes of commutative grammars and Petri nets Commutative grammars are closely related to Petri nets and also to matrix grammars [3].

Let Σ be an alphabet. Then, let $\Sigma^{\otimes}$ denote the free commutative monoid generated by Σ with the unit element λ. Each element in $\Sigma^{\otimes}$ is called a *commutative word*. If $\Sigma = \{a_1, \ldots, a_k\}$, then a commutative word $\omega \in \Sigma^{\otimes}$ will be written in the form $\omega = a_1^{i_1} \cdots a_k^{i_k}$ where $i_1, \ldots, i_k \in \mathbf{N}$.

A *commutative grammar* (abbreviated *CG*) is a 4-tuple $G^c = (N, \Sigma, \Pi^c, S)$, where

1. N is a finite nonempty set of *nonterminals*,
2. Π^c is a finite nonempty set of *productions* of the form $\alpha \to \beta$, where $\alpha \in N^{\otimes} - \{\lambda\}$ and $\beta \in (N \cup \Sigma)^{\otimes}$, and
3. S is a special nonterminal called the *start symbol*.

We denote by V the set $N \cup \Sigma$.

Let $G^c = (N, \Sigma, \Pi^c, S)$ be a *CG*. We define the relation $\underset{c}{\Longrightarrow}$ between elements in $V^{\otimes}$. For any $\alpha_1, \alpha_2 \in V^{\otimes}$, $\alpha_1 \underset{c}{\Longrightarrow} \alpha_2$ if and only if $\alpha_1 = \beta\gamma$, $\alpha_2 = \beta\delta$, and $\gamma \to \delta$ is a production in Π^c for some $\beta \in V^{\otimes}$. $\overset{*}{\underset{c}{\Longrightarrow}}$ denotes the reflexive and transitive closure of $\underset{c}{\Longrightarrow}$. The *language generated by* G_c, denoted by $L(G_c)$, is the set

$$L(G_c) = \{\omega \in \Sigma^{\otimes} \mid S \overset{*}{\underset{c}{\Longrightarrow}} \omega\}.$$

A *commutative language* (abbreviated *CL*) is a language generated by a *CG*.

Definition A *t-bounded CG* is a *CG* $G_c = (N, \Sigma, \Pi_c, S)$ such that each production in Π_c is of the form

1. $S \to \alpha A_1 A_2 \cdots A_s$, where $A_1, A_2, \ldots, A_s \in N - \{S\}$, $\alpha \in \Sigma^{\otimes}$, and $s \leq t$, or
2. $\beta \to \alpha\gamma$, where $\alpha \in \Sigma^{\otimes}$, $\beta\gamma \in (N - \{S\})^{\otimes}$, $|\gamma| \leq |\beta| \leq t$.

A t-bounded *CG* may be regarded as a model of the interaction of t numbers of sequential machines. Also, Crespi-Reghizzi and Mandrioli [3] have shown that t-bounded *CG* may represent a synchronization process in a modular *CG*, which may be regarded as a model of modular Petri nets.

The Parikh mapping of t-bounded *CL*s is defined in the same way as for *EML*s. Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet. The Parikh mapping ψ_c is the function from $\Sigma^{\otimes}$ into $\mathbf{N}^k$ defined by $\psi_c(\omega) = (i_1, \ldots, i_k)$ where $\omega = a_1^{i_1} \cdots a_k^{i_k}$. Note that ψ_c is a one-to-one mapping.

The following is due to Crespi-Reghizzi and Mandrioli [3]:

Proposition 7.2 *For any t-bounded CL L_c, $\psi_c(L_c)$ is a semilinear set.*

Also, we have the converse:

Proposition 7.3 *Given a positive integer t, an alphabet Σ, and a description $Q(\mathbf{c}_1, P_1) \cup \cdots \cup Q(\mathbf{c}_n, P_n)\}$ of a semilinear subset Q of $\mathbf{N}^k$, a t-bounded CG G_c such that $\psi_c(L(G_c)) = Q$ is effectively found.*

Proof. It is enough to show the case that Q is a linear set. Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet and $Q(\mathbf{c}, \{\mathbf{p}_1, \ldots, \mathbf{p}_r\})$ be a description of Q. Then define $G_c = (N, \Sigma, \Pi_c, S)$ as follows:

1. $N = \{S, A_1, \ldots, A_t\}$,
2. for $\mathbf{c} = (c_1, \ldots, c_k)$, $S \to a_1^{c_1} \cdots a_k^{c_k} A_1 \cdots A_t$ is in Π_c,
3. for each $\mathbf{p}_i = (i_1, \ldots, i_k) \in P$, $A_1 \cdots A_t \to a_1^{i_1} \cdots a_k^{i_k} A_1 \cdots A_t$ is in Π_c, and
4. for each A_i $(1 \leq i \leq t)$, $A_i \to \lambda$ is in Π_c.

It is easy to verify that $\psi_c(L(G_c)) = Q$. □

It is easy to verify that given an alphabet and a description of a semilinear set, we can construct a 1-bounded *CG*.

We consider the learning problem of t-bounded *CLs*. In this case, a learner should find a t-bounded *CG* which is consistent with the given commutative words. By the similar arguments in the case of *EMLs*, the family of t-bounded *CLs* whose Parikh sets are linear is learnable from positive examples while families of t-bounded *CLs* whose Parikh sets are n-linear $(n > 1)$ are not learnable from positive examples. Furthermore, given any teacher who answers restricted subset and restricted superset queries, then there exists an algorithm which identifies any t-bounded *CL* and halts. In particular, for the family of t-bounded *CLs* whose Parikh sets are linear, given any teacher who answers restricted superset queries for *any* t-bounded *CLs*, then there exists an algorithm which identifies any t-bounded *CL* whose Parikh set is linear in polynomial time and halts.

Given a commutative grammar, we can effectively construct a Petri net as described in [3]. We simply illustrate in Figure 6 a Petri net corresponding to a 2-bounded *CG* $G_c = (N, \Sigma, \Pi_c, S)$, where $N = \{S, A_1, A_2\}$, $\Sigma = \{a, b\}$, and $\Pi_c = \{S \to aA_1A_2,\ A_1A_2 \to abbA_1A_2,\ A_1 \to \lambda,\ A_2 \to \lambda\}$. Then, $\psi_c(L(G_c)) = Q((1,0), \{(1,2)\})$.

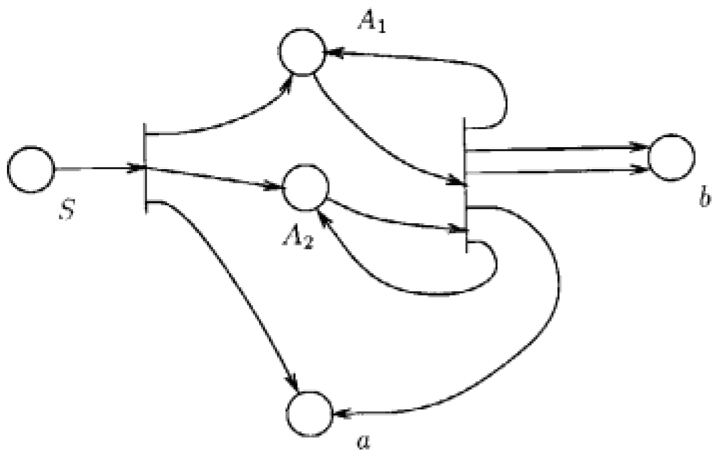


Figure 6: Petri net corresponding to t-bounded commutative grammar

Simple picture recognitions Some simple pictures could be recognized by grammars with an appropriate coding. Consider the problem of describing polygons, illustrated in Figure 7, in string languages. One of the most simple answers for the problem is to describe them in sequences of symbols which represent unit lines, as illustrated in Figure 8. Then, these strings have the same form $u_1^{n_1} \cdots u_m^{n_m}$, where each symbol u_i denotes a unit line. For example, a set of squares is described as the language $L_S = \{u_1^n u_2^n u_3^n u_4^n \mid n \geq 1\}$, so it is an *SBEML*.

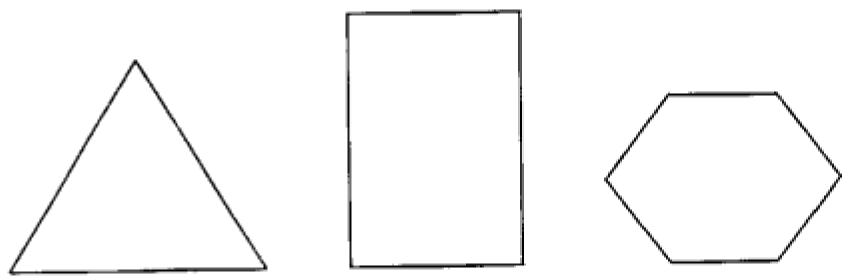
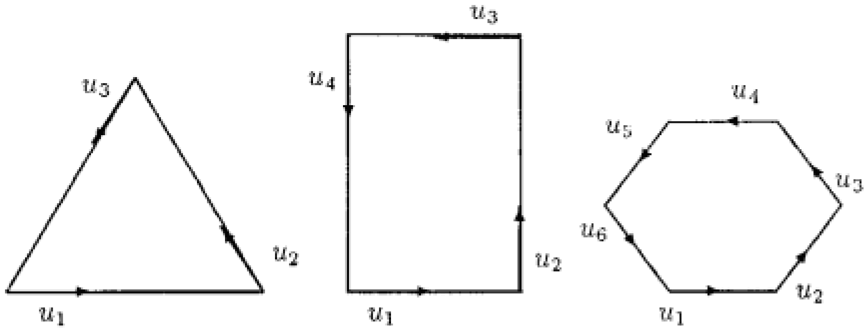


Figure 7: Polygons

On pictures described in strings over the symbols, which denote unit lines from the Cartesian plane considered as a square grid, Maurer et al. have studied various properties in [9].

We have shown that the family of *SBEMLs* is not learnable from positive examples, while the family of 1-*SBEMLs* is learnable from positive examples. These results suggest that

each concept of polygons described in *SBEMLs* is learnable from positive examples,



$u_1^3u_2^3u_3^3$ $\quad$ $u_1^2u_2^3u_3^2u_4^3$ $\quad$ $u_1^2u_2^2u_3^2u_4^2u_5^2u_6^2$

Figure 8: Polygons described in string languages

while mixed concepts of them are not so.

For example, consider the concept "square" is the language $L_S = \{u_1^n u_2^n u_3^n u_4^n \mid n \geq 1\}$. The Parikh set of L_S is a linear set $\psi_{(u_1,u_2,u_3,u_4)}(L_S) = \{(1,1,1,1) + n(1,1,1,1) \mid n \in \mathbf{N}\}$. Therefore, L_S is a 1-*SBEML* and learnable from positive examples. On the other hand, "rectangular in which vertical lines are two or three times longer than horizontal lines" is the language $L_{2,3} = \{u_1^n u_2^{2n} u_3^n u_4^{2n} \mid n \geq 1\} \cup \{u_1^n u_2^{3n} u_3^n u_4^{3n} \mid n \geq 1\}$. The Parikh set of $L_{2,3}$ is a semilinear set $\psi_{(u_1,u_2,u_3,u_4)}(L_{2,3}) = \{(1,2,1,2) + n(1,2,1,2) \mid n \in \mathbf{N}\} \cup \{(1,3,1,3) + n(1,3,1,3) \mid n \in \mathbf{N}\}$, so it is not learnable from positive examples. This matches with our intuition.

Also, our results suggest that these concepts are learnable with a teacher who can answer restricted subset and restricted superset queries. In particular, there exists a learner which identifies any single concept with a teacher who can answer restricted superset queries for any mixed concepts.

These concepts are also described in t-bounded *CLs*. In this case, a model of recognition devices can be described in Petri nets and then it stresses a side of parallel computation.

8 Concluding Remarks

We have shown that the family of semilinear subsets of $\mathbf{N}^k$ is not learnable from positive examples, while the family of linear subsets is learnable from positive examples. Also, we have presented a learning method for semilinear sets with restricted subset and restricted superset queries. In the case of linear sets, this method is efficient.

For parallel computation models such as commutative grammars and Petri nets, the semilinearity is a property on "semantics" of them. If there is an effective method to construct representations of models from descriptions of semilinear sets and vice versa, then our learning methods for semilinear sets provide the learning methods for them (any parallel computation models dealt with here is one of such cases). However, to solve the problem of constructing representations from semantic descriptions, we may need to study from a different point of view. For example, reachability sets of weakly persistent Petri nets are semilinear, but it seems difficult to reconstruct a representation of a given weakly persistent Petri net from a description of its semilinear reachability set. This is one of further research problems.

Acknowledgements

The author is indebted to Dr. Kaname Kobayashi for his useful suggestion and warm encouragement. He is very grateful to Yasubumi Sakakibara, his colleague, and Dr. Takashi Yokomori, University of Electro-Communications, who worked through an earlier draft of the paper and many suggestions. The author would also like to thank Kunihiko Hiraishi, Toshiro Minami, and Kazuhiro Yokoyama for their helpful comments and suggestions.

References

[1] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[3] S. Crespi-Reghizzi and D. Mandrioli. *Petri nets and commulative grammars*. Internal Report 74-5, Laboratorio di Calcolatori, Instituto di Electtrotecnica ed Electtronica del Politecnico di Milano, 1974.

[4] S. Eilenberg and M. P. Schützenberger. Rational sets in commutative monoids. *Journal of Algebra*, 13:173–191, 1969.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[6] S. Ginsburg. *The Mathematical Theory of Context Free Languages*. McGraw-Hill, New York, 1966.

[7] O. H. Ibarra. Simple matrix languages. *Information and Control*, 17:359–394, 1970.

[8] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcker, editors, *Complexity of Computer Computations*, pages 85–103, Plenum Press, New York, 1972.

[9] H. A. Maurer, G. Rozenberg, and E. Welzl. Using string languages to describe picture languages. *Information and Control*, 54:155–185, 1982.

[10] J. L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, 1981.

[11] A. Salomaa. *Formal Languages.* Academic Press, Inc., New York, 1973.

[12] R. Siromoney. On equal matrix languages. *Information and Control*, 14:135–151, 1969.

[13] H. Yamasaki. On weak persistency of petri nets. *Information Processing Letters*, 13(3):94–97, 1981.