Theoretical Computer Science 104 (1992) 207–233
Elsevier

Learning semilinear sets from examples and via queries*

Yuji Takada

International Institute for Advanced Study of Social Information Science (IIAS-SIS), Fujitsu Limited, 140, Miyamoto, Numazu, Shizuoka 410-03, Japan

Communicated by R. Siromoney
Received November 1990
Revised March 1991

Abstract

Takada, Y., Learning semilinear sets from examples and via queries, Theoretical Computer Science 104 (1992) 207–233.

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 problems of learning semilinear sets from examples and via queries. We shall show that (1) the family of semilinear sets is not learnable only from positive examples, while the family of linear sets is learnable only from positive examples, although the problem of learning linear sets from positive examples seems to be computationally intractable; (2) if for any unknown semilinear set S_u and any conjectured semilinear set S', queries whether or not $S_u \subseteq S'$ and queries whether or not $S' \subseteq S_u$ can be made, there exists a learning procedure which identifies any semilinear set and halts, although the procedure is time-consuming; (3) however, under the same condition, for each fixed dimension, there exist meaningful subfamilies of semilinear sets learnable in polynomial time of the minimum size of representations and, in particular, for any variable dimension, if for any unknown linear set L_u and any conjectured *semilinear* set S', queries whether or not $L_u \subseteq S'$ can be made, the family of linear sets is learnable in polynomial time of the minimum size of representations and the dimension.

1. Introduction

One of the 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. Matrix grammars [14], commutative grammars [5], and

* This is a part of the work in the major R&D of the Fifth Generation Computer Project, conducted under a program set up by MITI. A preliminary version of this paper appeared in the Proceedings of the First International Workshop on Algorithmic Learning Theory (Tokyo, Japan, October 1990).

0304-3975/92/$05.00

Petri nets [13] are the most successful models, for which substantial theories and analysis techniques have been developed enough to apply them to many practical concurrent system 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 the 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 subsemigroups of the set of all lattice points with nonnegative coordinates, and a finite union of linear sets is said to be *semilinear*. For example, for any equal matrix language [15] and simple matrix language [11], an image set on Parikh mapping is semilinear. Also, a reachability set of any weakly persistent Petri net [10, 18] 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 representation of an unknown semilinear set from given examples in the sense of [3, 9], and via queries in the sense of [4]. There are two major models of learning in recent computational learning theory, "PAC-learning model" and "exact-learning model" (cf. [4]). In PAC-learning model, the target concept is said to be probably approximately correctly identified if a representation of either the target concept or a concept slightly different from the target one is found. On the other hand, in the exact-learning model, the target concept is said to be exactly identified only if its representation is found. In PAC-learning model, Abe [1] has studied the polynomial PAC-learnability of semilinear sets. He has shown that the class of semilinear sets of dimensions 1 and 2 is polynomial PAC-learnable but the problem of learning semilinear sets of higher dimensions is hard. Generally, the learning problem in exact-learning model may be harder than the problem in PAC-learning model, especially on the time efficiency. In the exact-learning model, to reduce the time complexity, various types of queries could be used. In this paper, we investigate the problem of learning semilinear sets in the exact-learning model and show the learnability and nonlearnability from positive examples and via various types of queries.

We shall show the following:

- The family of semilinear sets is not learnable only from positive examples, while the family of linear sets is learnable only from positive examples, although the problem of learning linear sets from positive examples seems to be computationally intractable.
- If for any unknown semilinear set S_u and any conjectured semilinear set S', queries whether or not $S_u \subseteq S'$ and queries whether or not $S' \subseteq S_u$ can be made, there exists a learning procedure which identifies any semilinear set and halts, although the procedure is time-consuming.

- However, under the same condition, for each fixed dimension, there exist meaningful subfamilies of semilinear sets learnable in polynomial time of the minimum size of representations. In particular, for any variable dimension, if for any unknown linear set L_u and any conjectured *semilinear* set S', queries whether or not $L_u \subseteq S'$ can be made, the family of linear sets is learnable in polynomial time of the minimum size of representations and the dimension.

These results 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 define a special finite set and according to it, define a special representation for each semilinear set. These special sets and representations 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 only from positive examples but this property is not preserved under finite unions; therefore, the family of semilinear sets is not learnable only from positive examples. In Section 5, we consider the problem of learning linear sets from positive examples. It is shown that the problem of finding the minimum size representation of a linear set consistent with the given positive examples is computationally intractable. This gives us a strong partial evidence that the problem of learning linear sets from positive examples seems to be computationally intractable. In Section 6, we assume that there exists an ideal teacher who answers queries whether or not $S_u \subseteq S'$ and queries whether or not $S' \subseteq S_u$ for any unknown semilinear set S_u and any conjectured semilinear set S'. We present a learning procedure for semilinear sets via such queries. Although our procedure is time-consuming for the whole family of semilinear sets, it is efficient for the problem of learning meaningful subfamilies of semilinear sets, called t-periods semilinear sets, and the family of linear sets. For each fixed dimension, the families of t-periods semilinear sets is learnable in polynomial time of the minimum size of the representations via those queries. In particular, for any variable dimension the family of linear sets is learnable in polynomial time of the minimum size of representations and the dimension via only queries whether or not $L_u \subseteq S'$ for any unknown linear set L_u and any conjectured *semilinear set* S'. Furthermore, we show exponential lower bounds on the number of queries for various types of queries. This should be contrasted with our result on the efficient learning method.

Finally, in Section 7, we apply our results to the problem of learning some parallel computation models and a syntactic pattern recognition. It is shown that for some matrix languages, commutative languages, and Petri nets, there are subfamilies learnable only from positive examples and they are efficiently learnable via queries on inclusions. Also, our results provide some solutions for the problem of learning pictures coding in an appropriate scheme.

2. Preliminaries

Let $\mathbb{N}$ denote the set of all nonnegative integers. For each integer $k \geqslant 1$, let $\mathbb{N}^k = \mathbb{N} \times \cdots \times \mathbb{N}$ (k times) and for each $n \in \mathbb{N}$, $n^k = (n, \ldots, n)$, that is, the value of each coordinate is n. We regard $\mathbb{N}^k$ as a subset of the vector space of all k-tuples of rational numbers over the rational numbers. Thus, for elements $\boldsymbol{u} = (u_1, \ldots, u_k)$ and $\boldsymbol{v} = (v_1, \ldots, v_k)$ in $\mathbb{N}^k$, and n in $\mathbb{N}$, $\boldsymbol{u} + \boldsymbol{v} = (u_1 + v_1, \ldots, u_k + v_k)$, $\boldsymbol{u} - \boldsymbol{v} = (u_1 - v_1, \ldots, u_k - v_k)$, and $n\boldsymbol{u} = (nu_1, \ldots, nu_k)$.

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

A subset L of $\mathbb{N}^k$ is said to be *linear* if and only if there exist an element $\boldsymbol{c}$ and a finite subset P of $\mathbb{N}^k$ such that

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

$\boldsymbol{c}$ is called the *constant* and each $\boldsymbol{p}_i$ is called a *period* of $L(\boldsymbol{c}; P)$. A subset S of $\mathbb{N}^k$ is said to be *semilinear* if and only if S is a finite union of linear subsets of $\mathbb{N}^k$. Note that the empty set is semilinear, being the union of zero linear sets. We may also call a linear subset and a semilinear subset of $\mathbb{N}^k$ a linear set and a semilinear set of the dimension k, respectively.

For any linear set L, if $L = L(\boldsymbol{c}; P)$ then we call $L(\boldsymbol{c}; P)$ a *representation of* L. Let $S = L_1 \cup \cdots \cup L_n$ be a semilinear set such that for each linear set L_i $(1 \leqslant i \leqslant n)$, $L(\boldsymbol{c}_i; P_i)$ is a representation of L_i. Then, we denote a *representation of* S by $L(\boldsymbol{c}_1; P_1) \cup \cdots \cup L(\boldsymbol{c}_n; P_n)$. We note that any linear set and, therefore, any semilinear set might have more than one representation in terms of constants and periods. Therefore, we should distinguish between semilinear sets themselves and their representations, although we may regard a representation as a semilinear set itself if the context is clear.

To define sizes of representations, we assume that each nonnegative integer is represented as a string over the alphabet $\{0\}$ by the usual unary coding. Then, for each nonnegative integer n the *size* of n is defined by $n+1$. For each representation $L(\boldsymbol{c}; P)$ of a linear subset L of $\mathbb{N}^k$, we define its *size* by $\sum_{i=1}^{k} c_i + \sum_{p \in P} \sum_{i=1}^{k} p_i$. Also for each representation $L(\boldsymbol{c}_1; P_1) \cup \cdots \cup L(\boldsymbol{c}_n; P_n)$ of a semilinear subset S of $\mathbb{N}^k$, we define its *size* by $\sum_{i=1}^{n}$ (size of $L(\boldsymbol{c}_i; P_i)$).

We summarize the closure properties on Boolean operations and the properties on the inclusion relation of semilinear sets. The reader may find the formal proofs in [8], for example.

Proposition 2.1 (Ginsburg and Spanier [8]). *The family of semilinear subsets of* $\mathbb{N}^k$ *is effectively closed under union, intersection, and complement.*

Corollary 2.2 (Ginsburg and Spanier [8]). *It is effectively solvable to determine for arbitrary semilinear subsets S_1 and S_2 of $\mathbb{N}^k$, whether* (1) $S_1 \subseteq S_2$, (2) $S_1 = S_2$.

3. Characteristic sets and canonical representations

A finite subset E of a semilinear set S is said to be *descriptive* for S if and only if there is a representation $L(\boldsymbol{c}_1; P_1) \cup \cdots \cup L(\boldsymbol{c}_n; P_n)$ of S such that E includes the set $\bigcup_{i=1}^{n} (\{\boldsymbol{c}_i\} \cup \{\boldsymbol{c}_i + \boldsymbol{p} \mid \boldsymbol{p} \in P_i\})$.

Definition 3.1. Let S be a semilinear set. A *characteristic set* of S is a finite subset $C(S)$ of S such that

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

Lemma 3.2. *Let S_1 and S_2 be semilinear subsets of $\mathbb{N}^k$. The set of minimal elements of $S_1 - S_2$ is finite and effectively found.*

Proof. From Proposition 2.1, $S' = S_1 - S_2$ is a semilinear set and its representation $L(\boldsymbol{c}_1; P_1) \cup \cdots \cup L(\boldsymbol{c}_n; P_n)$ is effectively found from representations of S_1 and S_2. Let C be the set of minimal elements of $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n\}$. It is easy to verify that C is the set of minimal elements of S'. $\square$

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

Proof. Since L is a linear set, there exists the unique minimum element $\boldsymbol{c}$ of L, which is the constant of any representation of L. Let $P_0 = \emptyset$, $E_0 = \{\boldsymbol{c}\}$, and $i = 1$. Repeat the following procedure. Let D be a set of minimal elements of $L - L(\boldsymbol{c}; P_{i-1})$. By Lemma 3.2, D is finite and can be effectively found. Then, let $P_i = P_{i-1} \cup \{\boldsymbol{d} - \boldsymbol{c} \mid \boldsymbol{d} \in D\}$ and $E_i = E_{i-1} \cup D$. If $L = L(\boldsymbol{c}; P_i)$ then let $C(L) = E_i$ and halt. Otherwise, continue the step $i+1$. We note that the equivalence problem for semilinear sets is effectively solvable.

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

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

Definition 3.4. A representation $L(c_1; P_1) \cup \cdots \cup L(c_n; P_n)$ of a semilinear set S is said to be *canonical* if and only if the set $\bigcup_{i=1}^{n}(\{c_i\} \cup \{c_i + p \mid p \in P_i\})$ is the characteristic set of S.

Lemma 3.5. *For any linear set L, a representation $L(c; P)$ of L is canonical if and only if each period is not a linear sum of the other periods.*

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

From Lemmas 3.3 and 3.5, for any linear set L, a canonical representation $L(c; P)$ is unique and is effectively found from any representation of L. Also, this implies that a canonical representation is the minimum size representation. However, there exists a semilinear set such that a characteristic set is not unique; therefore, a canonical representation is not unique and is not thc minimum-size representation. For example, consider two semilinear subsets S_1 and S_2 of $\mathbb{N}^2$, whose representations are $L((0,0); \emptyset) \cup L((1,0); \{(1,0),(0,1)\})$ and $L((0,0); \{(1,0)\}) \cup L((1,1); \{(1,0),(0,1)\})$, respectively. It is easy to verify that $S_1 = S_2$ and sets $C(S_1) = \{(0,0),(1,0),(2,0),(1,1)\}$ and $C(S_2) = \{(0,0),(1,0),(1,1),(2,1),(1,2)\}$ are characteristic sets. Therefore, these representations are canonical.

We also note that given the characteristic set $C(L)$ of a linear set L, the canonical representation of L is effectively found. That is, the constant c is the unique minimum element of $C(L)$ and then the set of periods is $\{p_i \mid q_i - c,\ q_i \in (C(L) - \{c\})\}$.

4. Learnabilities from positive examples

In this section, we consider learnabilities of families of semilinear sets from positive examples. We show that although the family of linear sets is learnable only from positive examples, this learnability is not preserved under finite unions; therefore, the family of semilinear sets is not learnable.

On learning of formal languages, Angluin [3] presented a necessary and sufficient condition for languages to be learnable from positive examples. Note that Angluin's results require only the recursiveness of languages and the recursive enumerability of the family of languages. Hence, all of them are applicable to the problem of learning a recursive enumerable family of 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 subset of $\mathbb{N}^k$. We may assume that each nonempty recursive set has a finite representation such as recursive membership functions. Let "+" and "−" be special symbols. A *positive example* of R is a pair $(+, q)$ such that $q \in R$ and a *negative example* of R is a pair $(-, q)$

such that $q \in \mathbb{N}^k - R$. 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 $\mathbb{N}^k$ appears in σ at least once. 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 once.

A learning procedure is defined to be an effective procedure M whose input is a (positive) presentation of a recursive subset R of $\mathbb{N}^k$ and output is a finite or infinite sequence $W_1, W_2, W_3, \ldots$ of finite representations 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 $\mathbb{N}^k$ and M be a learning procedure. 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 representation 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 $\mathscr{R}$ of nonempty recursive subsets of $\mathbb{N}^k$ is *learnable from* (*positive*) *examples* if and only if there exists a learning procedure which identifies R from (positive) examples for every $R \in \mathscr{R}$.

Condition 1. A recursively enumerable family $\mathscr{R}$ of nonempty recursive subsets of $\mathbb{N}^k$ *satisfies* Condition 1 if and only if there exists an effective procedure which on any input $R \in \mathscr{R}$ enumerates a set T such that

(1) T is finite,
(2) $T \subseteq R$, and
(3) for all $R' \in \mathscr{R}$, 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 $\mathbb{N}^k$ to be learnable from positive examples.

Lemma 4.1 (Angluin [3]). *A recursively enumerable family of nonempty recursive subsets of* $\mathbb{N}^k$ *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 $\mathscr{R}$ of nonempty recursive subsets of $\mathbb{N}^k$ *satisfies* Condition 2 provided that, for every $R \in \mathscr{R}$, there exists a finite set $T \subseteq R$ such that for every $R' \in \mathscr{R}$, if $T \subseteq R'$ then R' is not a proper subset of R.

Lemma 4.2 (Angluin [3]). *If* $\mathscr{R}$ *is a recursively enumerable family of nonempty recursive subsets of* $\mathbb{N}^k$ *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 according to Angluin's results.

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

Proof. Let $L(c; P)$ be the canonical representation of L. Suppose that L' is a linear subset of $\mathbb{N}^k$ such that $C(L) \subseteq L'$ and $L(c'; \{p'_1, \ldots, p'_r\})$ is the canonical representation of L'. Since $C(L) \subseteq L'$, for each q_j of $C(L)$, $q_j = c' + n_{j1} p'_1 + \cdots + n_{jr} p'_r$. Therefore, for each period p_i of $L(c; P)$, $p_i = q_i - c = (n_{i_1} - n_{c_1}) p'_1 + \cdots + (n_{i_r} - n_{c_r}) p'_r$. Hence, for each $q \in L$, there exist $m_1, \ldots, m_r \in \mathbb{N}$ such that $q = c' + m_1 p'_1 + \cdots + m_r p'_r$. □

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

Proof. Let $L(c_1; P_1)$, $L(c_2; P_2)$, $L(c_3; P_3), \ldots$ be an effective enumeration of the canonical representations of all linear subsets of $\mathbb{N}^k$. It is obvious that there exists an effective procedure which on any input $i \geqslant 1$ enumerates a characteristic set C_i of a linear set $L(c_i; P_i)$. By the definition of characteristic sets of linear sets, C_i is finite and $C_i \subseteq L(c_i; P_i)$. Moreover, by Lemma 4.3, for all $j \geqslant 1$, if $C_i \subseteq L(c_j; P_j)$ then $L(c_j; P_j)$ is not a proper subset of $L(c_i; P_i)$. Therefore, the family satisfies Condition 1 and by Lemma 4.1 the proof is completed. □

Thus, the family of linear sets is learnable from positive examples. On the other hand, this learnability is not preserved under finite unions. To show this, we define the following special subfamilies of semilinear sets.

Definition 4.5. Let n be a positive integer. An *n-linear* set S is a semilinear set which is a union of exactly n linear sets but cannot be expressed by a union of i linear sets for any $i < n$.

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

Proof. At first, we show the case $k = 2$. Consider the 2-linear set $S = L_1 \cup L_2$, where $L_1 = L((0,0); \emptyset)$ and $L_2 = L((1,1); \{(1,0),(0,1)\})$. L is a 2-linear subset of $\mathbb{N}^2$ (see [6], for example).

Let $T = \{q_1, \ldots, q_m\}$ be any nonempty finite subset of S. Consider the 2-linear set $S^T = L_1^T \cup L_2^T$ (cf. Fig. 1), where

$$L_1^T = L((1,1); \{q_i - (1,1) \mid q_i = (1,s) \in T\}),$$

$$L_2^T = L((0,0); \{q_j \mid q_j = (q_1, q_2) \in T,\ q_1 \neq 1\}).$$

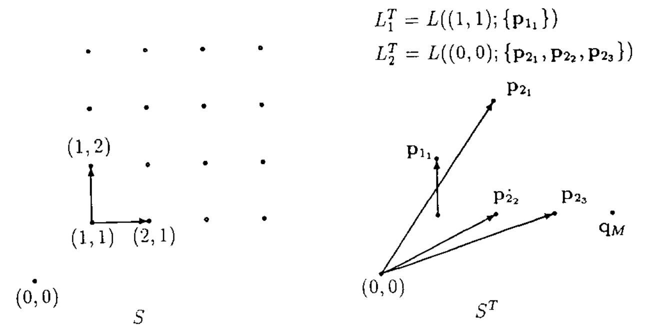


Fig. 1. The construction of S^T.

Canonical representations of L_1^T and L_2^T are effectively found from the above representations. Clearly, $T \subseteq S^T$ and it is easy to verify that $S^T \subseteq S$. For each $\boldsymbol{q}_i \in T$ let $\boldsymbol{q}_i = (q_{i_1}, q_{i_2})$. Let q_{M_1} be the maximum integer of $q_{l_1}, \ldots, q_{m_1}$. Then, $\boldsymbol{q}_M = q_{M_1} + 1, 1)$ is in S but not in S^T, so S^T is a proper subset of S. Thus, Condition 2 fails. The cases $k > 2$ are proved by similar arguments. $\square$

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

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

Proof. Let n be an integer greater than 2. Consider the n-linear set $S = L_1 \cup \cdots \cup L_n$ of $\mathbb{N}^2$, where for i $(1 \leqslant i \leqslant n-1)$, $L_i = L((i-1, 0); \emptyset)$ and $L_n = L((n-1, 1); \{(1, 0), (0, 1)\})$. It is easy to verify that S is an n-linear subset of $\mathbb{N}^2$.

Let $T = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m\}$ be any nonempty finite subset of S. Consider the n-linear set $S^T = L_1^T \cup \cdots \cup L_n^T$, where

$$L_i^T = L((i-1, 0); \emptyset) \quad \text{for } 1 \leqslant i \leqslant n-2,$$

$$L_{n-1}^T = L((n-1, 1); \{\boldsymbol{q}_j - (n-1, 1) \mid \boldsymbol{q}_j = (n-1, s) \in T\}),$$

$$L_n^T = L((n-2, 0); \{\boldsymbol{q}_j - (n-2, 0) \mid \boldsymbol{q}_j = (q_1, q_2) \in T,\ q_1 \neq n-1\}).$$

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

This theorem implies that the learnability from positive examples for linear sets is not preserved under finite unions.

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

This corollary also follows from Gold's result [9] because the family of all (nonempty) semilinear subsets of $\mathbb{N}^k$ is so-called *superfinite*.

5. Intractability of learning from examples

One of the problems which plays an important role in learning from examples is the problem of finding a minimal-size representation consistent with the given examples. If this problem were solvable efficiently, one could construct an efficient learning procedure.

Abe [1] has shown that the problem of finding a minimal representation of a semilinear set consistent with the given positive and negative examples is computationally intractable. In this section, we consider the problem of finding minimum size representations of linear sets consistent with the given positive examples and show that this problem is also computationally intractable unless P = NP. This gives us a strong partial evidence that the problem of finding optimal representations is hard even for linear sets and that although the family of linear sets is learnable only from positive examples (Theorem 4.4), the problem of learning linear sets from positive examples seems to be computationally intractable.

We consider the following problem.

Find minimum-size representations (FMR)

Instance: A finite subset E of $\mathbb{N}^k$.

Question: Find the minimum-size representation of a linear set which contains all elements of E.

If there were a polynomial-time algorithm to solve this problem FMR, then we could construct a learning procedure which makes a conjecture in polynomial time for each time and identifies any linear set in the limit only from positive examples in the following way. For each time t, the learning procedure makes a conjecture W_t by executing the algorithm which solves FMR on input $\{s_1, s_2, \ldots, s_t\}$. It is easy to verify that this learning procedure identifies L_u in the limit because for any positive presentation of an unknown linear set L_u, there exists a positive integer i such that the set $\{s_1, s_2, \ldots, s_i\}$ includes the characteristic set of L_u and then the conjecture is the canonical representation of L_u. However, if $P \neq NP$ then there is no polynomial-time algorithm which solves FMR.

Theorem 5.1. *If* $P \neq NP$, *then there is no polynomial-time algorithm which solves the problem* FMR.

Proof. Suppose that there exists an algorithm AF that runs in polynomial time and is such that for any subset E of $\mathbb{N}^k$, AF on input E outputs the minimum size representation $L(\boldsymbol{c}; P)$ of a linear subset of $\mathbb{N}^k$ which contains all elements of E. We shall use AF to construct a polynomial-time algorithm to decide whether $\boldsymbol{q} \in L(\boldsymbol{c}; P)$

for an arbitrary element $q \in \mathbb{N}^k$ and the canonical representation $L(c; P)$. Since the membership problem of linear sets is NP-complete shown in [17] and independently described in [1], this will imply P = NP, proving the theorem.

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

6. Learning semilinear sets via queries

As we have shown in the previous section, the problem of learning from examples seems to be computationally intractable even for linear sets. One of the common learning methods for improving the computational efficiency is using various types of queries. In this section, we consider the problem of learning semilinear sets via queries.

In previous sections, we had no assumption on the source presenting examples. Here, we assume that there exists an ideal teacher who can answer questions of a learning procedure and the learning procedure gets information from the teacher. We consider the following types of queries of learning procedures. Let $\mathscr{S}$ denote a (sub)family of semilinear sets which a learning procedure should learn and $\mathscr{H}$ denote a set of representations which a learning procedure outputs as conjectures.

(1) *Membership queries.* We denote this type of queries by MEM($\mathscr{S}$). For any unknown semilinear set $S_u \in \mathscr{S}$, a learning procedure can ask whether or not $q \in S_u$ for any $q \in \mathbb{N}^k$ and a teacher answers *yes* if $q \in S_u$ and *no* if $q \notin S_u$.

(2) *Equivalence queries.* We denote this type of queries by EQ($\mathscr{S}, \mathscr{H}$). For any unknown semilinear set $S_u \in \mathscr{S}$, a learning procedure can ask whether or not $S_u = H$ for any representation $H \in \mathscr{H}$ and a teacher answers *yes* if $S_u = H$ and *no* otherwise. If the answer is *no*, the teacher also gives the learning procedure an element $q \in (S_u - H) \cup (H - S_u)$.

(3) *Subset queries.* We denote this type of queries by SUB($\mathscr{S}, \mathscr{H}$). For any unknown semilinear set $S_u \in \mathscr{S}$, a learning procedure can ask whether or not $H \subseteq S_u$ for any representation $H \in \mathscr{H}$ and a teacher answers *yes* if $H \subseteq S_u$ and *no* otherwise. If the answer is *no*, the teacher also gives the learning procedure an element $q \in H - S_u$.

(4) *Superset queries.* We denote this type of queries by SUPER($\mathscr{S}, \mathscr{H}$). For any unknown semilinear set $S_u \in \mathscr{S}$, a learning procedure can ask whether or not $S_u \subseteq H$ and a teacher answers *yes* if $S_u \subseteq H$ and *no* otherwise. If the answer is *no*, the teacher also gives the learning procedure an element $q \in Q - H$.

For queries other than membership queries, 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. We denote the restricted versions with the subscript r such as $\mathrm{EQ}_r(\mathscr{S},\mathscr{H})$, $\mathrm{SUB}_r(\mathscr{S},\mathscr{H})$, $\mathrm{SUPER}_r(\mathscr{S},\mathscr{H})$.

A learning procedure is said to *identify* an unknown semilinear set S *via types of queries* $t_1,\ldots,t_n$ if and only if it identifies S making any query whose type is one of $t_1,\ldots,t_n$ and halts. A family of semilinear sets is said to be *learnable* via types of queries $t_1,\ldots,t_n$ if and only if there exists a learning procedure which identifies any semilinear set of the family via $t_1,\ldots,t_n$.

6.1. Learning via restricted subset queries and restricted superset queries

Let $\mathscr{S}$ denote the family of semilinear subsets of $\mathbb{N}^k$, and $\mathscr{H}_s$ denote the set of all representations of semilinear subsets of $\mathbb{N}^k$. We first show that there exists a learning procedure for $\mathscr{S}$ via $\mathrm{SUB}_r(\mathscr{S},\mathscr{H}_s)$ and $\mathrm{SUPER}_r(\mathscr{S},\mathscr{H}_s)$.

Let S be a semilinear set. For each i $(0\leqslant i)$, we define an ith *slice* D_i of S recursively:

(1) $D_0=\emptyset$,

(2) $D_i=\{\boldsymbol{q}\mid \boldsymbol{q}$ is a minimal element of $S-\bigcup_{j=0}^{i-1}D_j\}$.

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

Definition 6.1. A *representative set* of a semilinear set S is a finite subset $R(S)=\bigcup_{i=0}^{t}D_i$ of S constituted from t slices such that

(1) $R(S)$ is descriptive for S, and

(2) for any nonnegative integer j such that $j<t$, $\bigcup_{i=0}^{j}D_i$ is not descriptive for S.

Note that for any semilinear set its representative set is a superset of its characteristic set.

Lemma 6.2. *For any semilinear set S, the representative set $R(S)$ of S 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 $S-E_{i-1}$ and let $E_i=E_{i-1}\cup D$. From Lemma 3.2, D is finite and can be effectively found. For each $\boldsymbol{q}\in D$,

(1) for each $L\{\boldsymbol{c}_j;P_j)$ in W, if $\boldsymbol{c}<\boldsymbol{q}$ and $L(\boldsymbol{c};P_j\cup\{\boldsymbol{q}-\boldsymbol{c}\})\subseteq S$, then add $L(\boldsymbol{c}_j;P_j\cup\{\boldsymbol{q}-\boldsymbol{c}\})$ to W,

(2) add $L(\boldsymbol{q};\emptyset)$ to W.

Let W be a set of representations of linear sets obtained with the above modifications. If $\bigcup_{L(\boldsymbol{c}_j;P_j)\in W}L(\boldsymbol{c}_j;P_j)$ is a representation of S, then let $R(S)=E_i$ 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 L of S for which E_i is descriptive, W has a representation of L. Therefore, there exists some t such that E_t is descriptive for S, so the procedure outputs $E=E_t$ and halts. Then, obviously, for any s such that $s<t$, E_s is not descriptive for S. $\square$

We note that a representation constructed by the procedure in the proof of Lemma 6.2 may have representations $L(\boldsymbol{c}_1;P_1)$ and $L(\boldsymbol{c}_2;P_2)$ of linear sets such that $P_1 \subseteq P_2$. Then, $L(\boldsymbol{c}_1;P_1)$ is redundant. We can effectively eliminate such redundant representations of linear sets.

Lemma 6.3. *Let S be a semilinear subset of $\mathbb{N}^k$. Let n be the number of representations of linear sets, p be the maximum number of periods of the linear sets, and q be the maximum integer appearing in the periods in the minimum size representation of S. Then the cardinality of the representative set $R(S)$ is bounded by nq^p.*

Proof. We first note that the proof of Lemma 6.2 ensures that the constructed representation by the procedure in the proof has all representations of linear sets appearing in the minimum size representation of S.

Let $p_{\min}$ be the minimum nonzero integer appearing in the periods of the minimum size representation. Also, let $\boldsymbol{p}_{\max}$ be the period having q in some coordinate and $\boldsymbol{p}_{\min}$ be the period having $p_{\min}$ in some coordinate.

For some positive integer i, if $ip_{\min}>q$ for any k then the ith slice must contain $\boldsymbol{p}_{\max}$. Since $p_{\min}\geqslant 1$, $i\leqslant q$, that is, $R(S)$ constituted from at most q slices. Then for each $L(\boldsymbol{c};\{\boldsymbol{p}_1,\dots,\boldsymbol{p}_r\})$ in the minimum size representation of S, $R(S)$ may have any $\boldsymbol{q}=\boldsymbol{c}+n_1\boldsymbol{p}_1+\cdots+n_r\boldsymbol{p}_r$, where $n_i\leqslant q$ for each i $(1\leqslant i\leqslant r)$; therefore, $R(S)$ has at most q^r elements of $L(\boldsymbol{c};\{\boldsymbol{p}_1,\dots,\boldsymbol{p}_r\})$. Since the maximum number of periods of the linear sets is p, $R(S)$ has at most nq^p elements of S. $\square$

The learning procedure for semilinear sets, described in the following, identifies a semilinear set based on the procedure described in the proof of Lemma 6.2.

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

Let S' be any proper semilinear subset of S. On input S', the procedure FP, illustrated in Fig. 2, finds a set of the minimal elements of S_u-S' via $\mathrm{SUPER}_r(\mathscr{S},\mathscr{H}_s)$.

The procedure FP begins queries whether $S_u\subseteq(E\cup U\cup L(\boldsymbol{q}+\boldsymbol{e}[1];P_e))$ with $U=\emptyset$ and $\boldsymbol{q}=0^k$ (cf. Fig. 3a). For each i $(1\leqslant i\leqslant k)$, FP continues queries until ith value of $\boldsymbol{q}$ is greater than the minimum ith value in the minimal elements which have not been found yet (cf. Fig. 3b, where $i=1$). Then, FP adds $L(\boldsymbol{q}+\boldsymbol{e}[i];P_e)$ to U and continues queries for $i+1$ coordinate (cf. Fig. 3c, where $i=1$). This U guarantees that any minimal element whose value of the ith coordinate is greater than the one of $\boldsymbol{q}$ is contained in U. In this way, FP finds the value of each ith coordinate.

Procedure *FP*
Input: A representation of a semilinear subset S' of $\mathbb{N}^k$.
Output: A finite subset D of $\mathbb{N}^k$.
Query: $\mathrm{SUPER}_r(\mathcal{S},\mathcal{H}_s)$.

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

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

Output D and halt;

Fig. 2. The procedure *FP*.

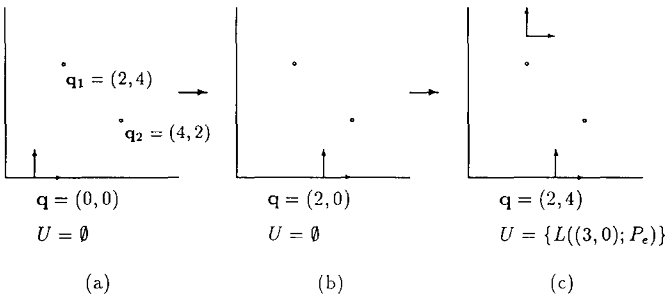


Fig. 3. How *FP* finds minimal elements.

Lemma 6.4. *Let S_u be a semilinear subset of $\mathbb{N}^k$ and S' a proper semilinear subset of S_u. On input S', the procedure FP makes at most $nk(m+1)$ restricted superset queries and outputs a finite set D of minimal elements of $S_u - S'$, where m is the maximum integer appearing in the elements of D and n is the cardinality of D.*

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

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

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

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

(1) $\boldsymbol{q}>\boldsymbol{r}$,

(2) $\boldsymbol{q}\leqslant\boldsymbol{r}$, and

(3) $\boldsymbol{q}$ and $\boldsymbol{r}$ are incomparable.

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

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

For each i $(1\leqslant i\leqslant k)$, 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 procedure ID, illustrated in Fig. 4, runs the procedure FP repeatedly, finds slices, and constructs a representation in the same way described in the proof of Lemma 6.2.

Proposition 6.5. *The procedure ID identifies any semilinear set via* $SUB_r(\mathscr{S},\mathscr{H}_s)$ *and* $SUPER_r(\mathscr{S},\mathscr{H}_s)$.

Proof. Let $R(S_u)=\bigcup_{i=0}^{t}D_i$ be the representative set of an unknown semilinear set S_u. By running the procedure FP repeatedly, from Lemma 6.4, ID finds each slice D_i and in the sequel, finds $R(S_u)$. Then, by Lemma 6.2, ID constructs a representation of S_u, so the teacher must answer *yes* for the query whether or not $S_u\subseteq\bigcup_{L(\boldsymbol{c}_i;P_i)\in W}L(\boldsymbol{c}_i;P_i)$. This completes the proof. □

We note that a constructed representation may have redundant representations of linear sets. However, such representations can be removed in the obvious way via $\mathrm{SUPER}_r(\mathscr{S},\mathscr{H}_s)$.

```
Procedure ID
Output:   A representation of an unknown semilinear subset S_u of N^k.
Query:    SUB_r(S,H_s) and SUPER_r(S,H_s).

W := ∅;
R(S_u) := ∅;

While the teacher replies no to a query whether S_u ⊆ ∪_{L(c_i;P_i)∈W} L(c_i; P_i) do
    Begin
        Run Procedure FP on input R(S_u) and get an output D;
        R(S_u) := R(S_u) ∪ D;
        While D is not empty do
            Begin
                let d be an element in D;
                For each L(c_i; P_i) of W such that c_i < d do
                    Begin
                        Ask the teacher whether L(c_i; P_i ∪ {d − c_i}) ⊆ S_u;
                        If    the answer is yes
                        then  W := W ∪ {L(c_i; P_i ∪ {d − c_i})};
                    End;
                W := W ∪ L(d; ∅);
            End;
    End;

Output W and halt;
```

Fig. 4. The procedure ID.

Thus, the learning procedure ID identifies any semilinear set via restricted subset and restricted superset queries and halts. However, ID is time-consuming. Let m be the minimum size of representations of S_u. ID may make $x2^{x-1}$ conjectures in the worst case, where x is the cardinality of the representative set $R(S_u)$ of S_u, which is at most m^{m+1}. Therefore, the total running time of ID is bounded by an exponential of m.

Although ID is time-consuming in the general case, there exists a meaningful subfamily for which ID is efficient.

Definition 6.6. Let t be a positive integer. A *t-periods semilinear set* $S^{(t)}$ is a semilinear set which has a representation $L(c_1; P_1) \cup \cdots \cup L(c_n; P_n)$ such that the cardinality of P_i is at most t for any i $(1 \leqslant i \leqslant n)$.

As Abe [2] has shown, the Parikh-image[1] of any language accepted by a commutative deterministic finite automaton is a k-periods semilinear set, where k is the cardinality of the alphabet.

Let $\mathscr{S}_t$ be the family of t-periods semilinear subsets of $\mathbb{N}^k$, and $\mathscr{H}_t$ be the set of all representations of t-periods semilinear subsets of $\mathbb{N}^k$.

Theorem 6.7. *Let k be a fixed dimension. The procedure ID identifies any t-periods semilinear set $S^{(t)}$ and halts via $SUB_r(\mathscr{S}_t, \mathscr{H}_i)$ and $SUPER_r(\mathscr{S}_t, \mathscr{H}_i)$ where if $k \leqslant t$, $i = t$,*

[1] For the definition, see Section 7.

otherwise $i=k$. The total running time of ID is bounded by a polynomial of m, where m is the minimum size of representation of $S^{(t)}$.

Proof. Let $R(S^{(t)})$ be the representative set of $S^{(t)}$. Then the number of representations of linear sets constructible from $R(S^{(t)})$ is

$$\sum_{i=1}^{t} \frac{(x+i-1)!}{i!(x-1)!} \leqslant t x^{t+1},$$

where x is the cardinality of $R(S^{(t)})$, which is $x=m^{t+1}$ from Lemma 6.3 because the number of representations of linear sets in the minimum size representation of $S^{(t)}$ and the maximum integer appearing in the periods are bounded by m. Therefore, the number of restricted subset queries is bounded by a polynomial of m. The number of restricted superset queries used by FP is bounded by a polynomial of the cardinality of $R(S^{(t)})$, the maximum integer appearing in $R(S^{(t)})$, and the dimension k by Lemma 6.4. Also, the number of restricted superset queries used by ID for the final check is bounded by a polynomial of m. Therefore, the total number of queries is bounded by a polynomial of m.

Since representations the procedure FP outputs for queries have k-periods semilinear sets, if $k>t$ then some representations ID outputs are not in $\mathscr{H}_t$ but in $\mathscr{H}_k$. □

We note that the size of constructed representation is bounded by a polynomial of m.

Finally, we consider the case of learning linear sets. Let $\mathscr{L}$ be the family of linear subsets of $\mathbb{N}^k$ and $\mathscr{H}_\ell$ be the set of all representations of linear subsets of $\mathbb{N}^k$. Also let $\mathscr{H}_e$ be the set of representations $\{L(\boldsymbol{c}_1; P_e) \cup \cdots \cup L(\boldsymbol{c}_n; P_e) \mid \boldsymbol{c}_i \in \mathbb{N}^k\}$.

Theorem 6.8. *There exists a learning procedure which identifies any linear subset L_u of $\mathbb{N}^k$ in polynomial time of the dimension k and the minimum size m of representations of L_u via $SUPER_r(\mathscr{L}, \mathscr{H}_\ell \cup \mathscr{H}_e)$.*

Proof. We modify ID in the following way. At first, the modified ID executes FP on input the empty set. Then FP outputs a constant $\boldsymbol{c}$ of L_u. Given a representation of a linear subset L' of L_u instead of a finite set of elements of L_u, FP outputs a finite subset D of minimal elements in $L_u - L'$. It is easy to verify that $\boldsymbol{d}-\boldsymbol{c}$ must be a period of L_u for each $\boldsymbol{d} \in D$ and, in the sequel, the modified ID finds a representation of L_u. Then, the set of all elements found by FP is the characteristic set of L_u, so the found representation is canonical.

Since the number of restricted superset queries used by FP is bounded by a polynomial of the maximum integer appearing in the characteristic set $C(L_u)$ of L_u and the cardinality of $C(L_u)$, which are bounded by a polynomial of m, the number of queries used by the modified ID is bounded by a polynomial of m. □

We note that since FP makes queries for k-periods semilinear sets, the learning procedure for linear sets makes queries on semilinear sets. However, the time complexity of this learning procedure is bounded by a polynomial even if we consider the dimension k as a parameter.

6.2. Lower bounds for queries

In [4], Angluin has presented lower-bound techniques for queries. According to her techniques, we show exponential lower bounds of the number of queries needed for learning semilinear sets via queries in the case where the dimension k is considered as a parameter. These should be contrasted with the results in the previous section and Abe's results [1]. In particular, we show that any procedure that identifies any linear subset of $\mathbb{N}^k$ and halts via membership, equivalence, and subset queries must make at least 2^k-1 queries in the worst case. This should be contrasted with Theorem 6.8.

We denote by $\mathscr{S}$ the family of semilinear subsets of $\mathbb{N}^k$ and by $\mathscr{L}$ the family of linear subsets of $\mathbb{N}^k$. We also denote by $\mathscr{H}_s$ the set of all representations of semilinear subsets of $\mathbb{N}^k$ and by $\mathscr{H}_\ell$ the set of all representations of linear subsets of $\mathbb{N}^k$. Let C be a finite set of all elements in $\mathbb{N}^k$ such that the value of each coordinate is 0 or 1. Let $\mathscr{S}_1=\{\{\boldsymbol{c}\}\mid \boldsymbol{c}\in C\}$ be a subfamily of linear subsets of $\mathbb{N}^k$. Also, let $\mathscr{S}_2=\{C-\{\boldsymbol{c}\}\mid \boldsymbol{c}\in C\}$ be a subfamily of semilinear subsets of $\mathbb{N}^k$. Then, clearly, $|\mathscr{S}_1|=|\mathscr{S}_2|=|C|=2^k$ and each linear set in $\mathscr{S}_1$ is disjoint, where $|S|$ denotes the cardinality of the set S. The family $\mathscr{S}_1$ consists of linear sets and the family of $\mathscr{S}_2$ consists of semilinear sets.

Proposition 6.9. *Any procedure that identifies any semilinear subset of $\mathbb{N}^k$ and halts via $MEM(\mathscr{S})$, $EQ_r(\mathscr{S},\mathscr{H}_s)$, and $SUB(\mathscr{S},\mathscr{H}_s)$ must make at least 2^k-1 queries in the worst case for a given dimension k.*

Proof. Consider the following teacher. For a restricted equivalence query with the conjecture L_*, the answer is *no*, and the (at most one) L_i such that $L_*=L_i$ is removed from $\mathscr{S}_1$. For a membership query with the element $\boldsymbol{q}$, the answer is *no*, and if $\boldsymbol{q}\in C$ then the (at most one) L_i such that $\boldsymbol{q}\in L_i$ is removed from $\mathscr{S}_1$. For a subset query with the conjecture L_*, if $L_*=\emptyset$ then the answer is *yes*. Otherwise, the answer is *no* and any element $\boldsymbol{q}$ in $\mathbb{N}^k$ is selected as the counterexample. If the counterexample $\boldsymbol{q}$ is in C then the (at most one) element L_i such that $\boldsymbol{q}\in L_i$ is removed from $\mathscr{S}_1$.

At any point, for each $L_i\in\mathscr{S}_1$, L_i is compatible with the answers to the queries so far. A procedure which identifies a semilinear set and halts must reduce the cardinality of $\mathscr{S}_1$ by at most one. Since each query removes at most one element from the set $\mathscr{S}_1$, to identify any $L_i\in\mathscr{S}_1$, at least 2^k-1 queries are required in the worst case. This completes the proof. □

This proposition should be contrasted with the fact of Theorem 6.7 that our learning procedure runs in polynomial time for a fixed dimension k.

Since the empty set is not a linear set, with a minor modification of the proof of Proposition 6.9, we have the following proposition.

Proposition 6.10. *Any procedure that identifies any linear subset of* $\mathbb{N}^k$ *and halts via* $MEM(\mathscr{L})$, $EQ(\mathscr{L},\mathscr{H}_\ell)$, *and* $SUB(\mathscr{L},\mathscr{H}_\ell)$ *must make at least* 2^k-1 *queries in the worst case for a given dimension k.*

Proof. The proof of Proposition 6.9 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 L_i be a conjecture. Since $\emptyset$ is not a linear set, $L_i \neq \emptyset$. The counterexample is any element $\boldsymbol{q}$ in L_*. The (at most one) element L_* is removed from $\mathscr{S}_1$. $\square$

This proposition should be contrasted with the fact of Theorem 6.8 that the time complexity of our learning procedure is bounded by a polynomial even if we consider k as a parameter.

For the family of semilinear sets, we show the dual result of Proposition 6.9:

Proposition 6.11. *Any procedure that identifies any semilinear subset of* $\mathbb{N}^k$ *and halts via* $MEM(\mathscr{S})$, $EQ_r(\mathscr{S},\mathscr{H}_s)$, *and* $SUPER(\mathscr{S},\mathscr{H}_s)$ *must make at least* 2^k-1 *queries in the worst case for a given dimension k.*

Proof. Consider the following teacher. For a restricted equivalence query with the conjecture S_*, the answer is *no*, and the (at most one) S_i such that $S_* = S_i$ is removed from $\mathscr{S}_2$. For a membership query with the element $\boldsymbol{q}$, the answer is *no*, and if $\boldsymbol{q} \in C$ then the (at most one) S_i such that $\boldsymbol{q} \in S_i$ is removed from $\mathscr{S}_1$. For a superset query with the conjecture S_*, if $C \subseteq S_*$ then the answer is *yes*. Otherwise, the answer is *no* and any element $\boldsymbol{q}$ in $\mathbb{N}^k$ is selected as the counterexample. If the counterexample $\boldsymbol{q}$ is in C then the (at most one) element S_i such that $\boldsymbol{q} \notin S_i$ is removed from $\mathscr{S}_2$. $\square$

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 [15], simple matrix languages [11], and weakly persistent Petri nets [18] are semilinear sets. In this section, we consider the problem of learning these models based on our methods described above.

7.1. Learning strictly bounded equal matrix languages and picture languages

Let Σ be an *alphabet*, i.e., a finite set of symbols and Σ^* the set of all strings over Σ containing the null string λ. A *language* M over Σ is a subset of Σ^*. 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 M over an alphabet Σ is said to be *strictly bounded* if and only if $M \subseteq a_1^* \ldots a_k^*$ where $\Sigma = \{a_1, \ldots, a_k\}$. In general, a language M over Σ is said to be *bounded* if and only if there exist words $w_1, \ldots, w_k \in \Sigma^*$ such that $M \subseteq w_1^* \ldots w_k^*$.

Let k be a positive integer. An *equal matrix grammar of order k*, abbreviated EMG_k, is a $(k+3)$-tuple $G=(N_1,\ldots,N_k,\Sigma,\Pi,S)$. $N_1,\ldots,N_k$ are finite nonempty pairwise disjoint sets of *nonterminals*. S is not in $N_1\cup\cdots\cup N_k\cup\Sigma$ and is called the *start symbol*. Π is a finite nonempty set of the following three types of *matrix rules*:

(1) *initial matrix rules* of the form $[S\to u_1A_1\ldots u_kA_k]$,

(2) *nonterminal matrix rules* of the form $[A_1\to u_1B_1,\ldots,A_k\to u_kB_k]$, and

(3) *terminal matrix rules* of the form $[A_1\to u_1,\ldots,A_k\to u_k]$,

where for each i $(1\leqslant i\leqslant k)$, A_i,B_i are in N_i and $u_i\in\Sigma^*$. An *equal matrix grammar*, abbreviated *EMG*, is an equal matrix grammar of any *finite* order k.

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

Let $G=(N_1,\ldots,N_k,\Sigma,\Pi,S)$ be an EMG_k. We denote $N_1\cup\cdots\cup N_k\cup\Sigma\cup\{S\}$ by V. We define the relation $\Rightarrow$ between strings in V^*. For any $x,y\in V^*$, $x\Rightarrow y$ if and only if either

(1) x is the initial symbol S and the initial matrix rule $[S\to y]$ is in Π, or

(2) there exist strings $u_1,\ldots,u_k$ over Σ such that $x=u_1A_1\ldots u_kA_k$, $y=u_1z_1\ldots u_kz_k$, 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 $x\overset{*}{\Rightarrow}y$ if either $x=y$ or there exist $x_0,\ldots,x_n\in V^*$ such that $x=x_0$, $y=x_n$, and $x_i\Rightarrow x_{i+1}$ for each i. The sequence $x_0,\ldots,x_n$ is called a *derivation* (from x_0 to x_n) and is denoted by

$$x_0\Rightarrow\cdots\Rightarrow x_n.$$

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

$$M(G)=\{w\in\Sigma^*\mid S\overset{*}{\Rightarrow}w\}.$$

A language M is said to be an *equal matrix language of order k*, abbreviated EML_k, if and only if there exists an EMG_k G such that $M=M(G)$ holds.

The family of *EML*s contains some context-sensitive languages. For example, the context-sensitive language $\{a^nb^nc^n\mid n\geqslant 1\}$ is an EML_3. Also, there exists a context-free language which is not an EML_k for any k [11]. For example, consider the language $M=\bigcup_{i\geqslant 0}\{a^nb^n\mid n\geqslant 1\}^i$. M is a context-free language but it is not an EML_k for any k.

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

The Parikh mapping defined as follows connects EML_ks with semilinear subsets of $\mathbb{N}^k$.

Definition 7.1. 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 $\mathbb{N}^k$ 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 $\psi(w_1\ldots w_n)=\sum_{i=1}^n\psi(w_i)$ for each $w_i\in\Sigma^*$. For any EML_k M, we call a subset $\psi(M)=\{\psi(w)\mid w\in M\}$ of $\mathbb{N}^k$ *the Parikh-image of an* EML_k M.

The following theorem is due to Siromoney [15].

Theorem 7.2 (Siromoney [15]). *Let $\Sigma=\{a_1,\dots,a_k\}$ be an alphabet. For any strictly bounded language M over Σ, M is generated by an EMG_k G if and only if the Parikh-image of M is a semilinear subset S of $\mathbb{N}^k$. Moreover, the EMG_k G is effectively found from a representation of S and vice versa.*

For any semilinear subset S of $\mathbb{N}^k$, an EMG_k G which generates an $SB\text{-}EML_k$ is effectively constructed from a representation of S in the following manner. It is enough to show the case that S is a linear set. Let $L(\boldsymbol{c};\{\boldsymbol{p}_1,\dots,\boldsymbol{p}_r\})$ be a representation of the linear subset L of $\mathbb{N}^k$. Also, let $\boldsymbol{c}=(c_1,\dots,c_k)$ and $\boldsymbol{p}_i=(p_i^1,\dots,p_i^k)$. Then $G=(N_1,\dots,N_k,\Sigma,\Pi,S)$, where $\Sigma=\{a_1,\dots,a_k\}$, each $N_i=\{A_i\}$, and Π consists of the following matrix rules:

$$[S\to a_1^{c_1}A_1\dots a_k^{c_k}A_k],$$

$$[A_1\to\lambda,\ \dots,\ A_k\to\lambda],$$

$$[A_1\to a_1^{p_i^1}A_1,\ \dots,\ A_k\to a_k^{p_i^k}A_k] \quad \text{for each } i\ (1\leqslant i\leqslant r).$$

Obviously, the time complexity of the construction of an EMG_k from a representation of a semilinear set is a polynomial time.

From Theorem 7.2, we may regard the learning problem for *SB-EML*s as the learning problem for semilinear sets.

According to Theorem 7.2, we consider meaningful subfamilies of *SB-EML*s.

Definition 7.3. For each positive integer n, an *SB-EML* M is said to be *n-SB-EML* if and only if $\psi(M)$ is an n-linear set.

Thus, a *1-SB-EML* is an *SB-EML* whose Parikh-image is a linear set and an *n-SB-EML* is an *SB-EML* whose Parikh-image is an n-linear set.

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

$$\psi^{-1}(L(c_1;P_1)),\ \psi^{-1}(L(c_2;P_2)),\ \psi^{-1}(L(c_3;P_3)),\ \dots$$

is an indexed family of 1-fold $SB\text{-}EML_k$s. Then, from Theorems 4.4 and 7.2, the family of $1\text{-}SB\text{-}EML_k$s is learnable from positive examples. On the other hand, from Theorems 4.7 and 7.2, for the family of $1\text{-}SB\text{-}EML_k$s, the learnability from positive examples is not preserved under finite unions. Therefore, for each positive integer n such that $n>1$, the family of $n\text{-}SB\text{-}EML_k$s is not learnable from positive examples.

From Proposition 6.5, there exists a learning procedure which identifies any *n-SB-EML* and halts via restricted subset and restricted superset queries. From Theorem 6.7, for each fixed positive integer k, there exists an efficient learning procedure via restricted subset and restricted superset queries for the family of

$SB\text{-}EML_k$ whose Parikh-images are t-periods semilinear sets for some fixed t. Furthermore, from Theorem 6.8, there exists a procedure which identifies any 1-$SB\text{-}EML$ in polynomial time and halts via restricted superset queries on $SB\text{-}EML$s.

Siromoney et al. [16] have presented a syntactic method for pattern recognition using EMGs. In their method, patterns are encoded as strings of chain codes which consist of 8 symbols (cf. Fig. 5). A pattern is encoded starting from its northwest corner. When there is more than one direction to be followed in encoding a pattern, the order of priority is $e_{\mathrm{r}}, e_{\mathrm{ne}}, e_{\mathrm{u}}, e_{\mathrm{nw}}, e_{\ell}, e_{\mathrm{sw}}, e_{\mathrm{d}}$, and e_{se}, that is, anticlockwise, starting from right. Thus, the English letter "T" will be encoded as "$e_{\mathrm{r}}e_{\mathrm{r}}e_{\ell}e_{\mathrm{d}}e_{\mathrm{d}}$ (cf. Fig. 6) and the representation of "T" of any size is encoded as "$e_{\mathrm{r}}^n e_{\mathrm{r}}^n e_{\ell}^n e_{\mathrm{d}}^{2n}$", where n is an integer greater than 0. The EMG_5 generating "T" is given by $G=(N_1, \ldots, N_5, \Sigma, \Pi, S)$, where $N_1=\{A\}$, $N_2=\{B\}$, $N_3=\{C\}$, $N_4=\{D\}$, $N_5=\{E\}$, $\Sigma=\{e_{\mathrm{r}}, e_{\ell}, e_{\mathrm{d}}\}$, and Π consists of the following matrix rules:

$$[S \rightarrow ABCDE],$$

$$[A \rightarrow e_{\mathrm{r}}A,\ B \rightarrow e_{\mathrm{r}}B,\ C \rightarrow e_{\ell}C,\ D \rightarrow e_{\mathrm{d}}D,\ E \rightarrow e_{\mathrm{d}}E],$$

$$[A \rightarrow e_{\mathrm{r}},\ B \rightarrow e_{\mathrm{r}},\ C \rightarrow e_{\ell},\ D \rightarrow e_{\mathrm{d}},\ E \rightarrow e_{\mathrm{d}}].$$

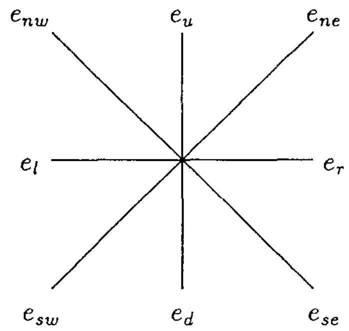


Fig. 5. Chain codes.

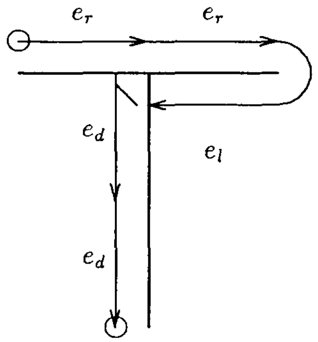


Fig. 6. An encoded letter "T".

This grammar generates the language $M_{\mathrm{T}}=\{e_{\mathrm{r}}^{n}e_{\mathrm{r}}^{n}e_{\ell}^{n}e_{\mathrm{d}}^{2n} \mid n\geqslant 1\}$ representing "T" of all sizes. If we regard the representation of "T" as $e_{\mathrm{r}_1}^{n}e_{\mathrm{r}_2}^{n}e_{\ell_3}^{n}e_{\mathrm{d}_4}^{n}e_{\mathrm{d}_5}^{n}$ other than $e_{\mathrm{r}}^{n}e_{\mathrm{r}}^{n}e_{\ell}^{n}e_{\mathrm{d}}^{2n}$ then we have 1-*SB-EML*$_5$ instead of *EML*$_5$. Thus, the language representing "T" is learnable only from positive examples by our result. Also, our results suggest that this representation is learnable via restricted superset queries on *SB-EML*$_5$s in polynomial time. In the same way of encoding, the six English letters "H", "K", "U", "V", "X", and "Y" are learnable only from positive examples although three letters "D", "F", and "P" are not learnable. Also, these letters are learnable via restricted subset and restricted superset queries on *SB-EML*s.

A similar encoding method can be applied to some simple pictures. Consider the problem of describing polygons, illustrated in Fig. 7, in similar representations. They can be encoded as illustrated in Fig. 8. Then, these representation 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 $M_{\mathrm{S}}=\{u_1^n u_2^n u_3^n u_4^n \mid n\geqslant 1\}$, so it is an *SB-EML*.

On pictures described in strings over the symbols, which denote unit lines from the Cartesian plane considered as a square grid, Maurer et al. [12] have studied the various properties.

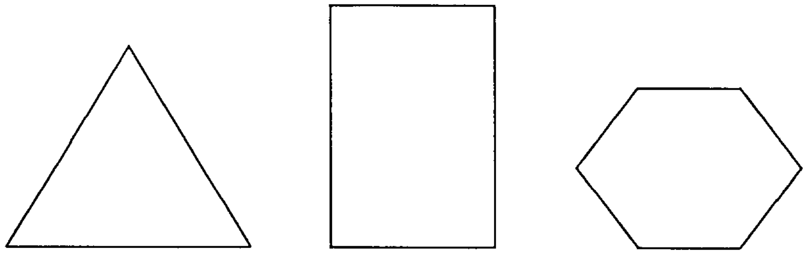
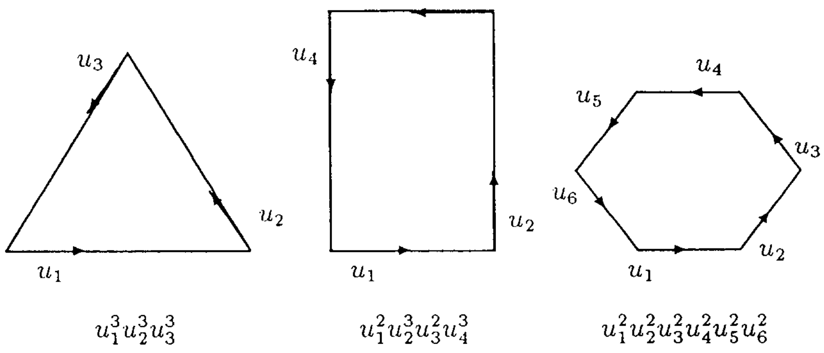


Fig. 7. Polygons.



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

Fig. 8. Polygons described in string languages.

For the problem of learning this kind of encoded pictures, our results suggest that each concept of polygons described in *SB-EMLs* is learnable from positive examples, while mixed concepts of them are not so. For example, consider the concept "square" is the language $L_S=\{u_1^n u_2^n u_3^n u_4^n \mid n\geqslant 1\}$. The Parikh-image of L_S is a linear set $\psi_{(u_1,u_2,u_3,u_4)}(L_S)=\{(1,1,1,1)+n(1,1,1,1)\mid n\in\mathbb{N}\}$. Therefore, L_S is a 1-*SB-EML* and learnable from positive examples. On the other hand, "rectangle in which vertical lines are two or three times longer than horizontal lines" is the language $L_{2,3}=\{u_1^n u_2^{2n} u_3^n u_4^{2n}\mid n\geqslant 1\}\cup\{u_1^n u_2^{3n} u_3^n u_4^{3n}\mid n\geqslant 1\}$. The Parikh-image of $L_{2,3}$ is a semilinear set $\psi_{(u_1,u_2,u_3,u_4)}(L_{2,3})=\{(1,2,1,2)+n(1,2,1,2)\mid n\in\mathbb{N}\}\cup\{(1,3,1,3)+n(1,3,1,3)\mid n\in\mathbb{N}\}$, so it is not learnable from positive examples. This matches with our intuition.

Also, our results suggest that these concepts are learnable via restricted subset and restricted superset queries. In particular, there exists a learning procedure which identifies any single concept via restricted superset queries on mixed concepts.

7.2. Commutative grammars and Petri nets

Commutative grammars are closely related to Petri nets and also to matrix grammars [5].

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 $\Sigma=\{a_1,\dots,a_k\}$, then a commutative word $\omega\in\Sigma^{\otimes}$ will be written in the form $\omega=a_1^{i_1}\dots a_k^{i_k}$, where $i_1,\dots,i_k\in\mathbb{N}$.

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 $\underset{c}{\Rightarrow}$ between elements in $V^{\otimes}$. For any $\alpha_1,\alpha_2\in V^{\otimes}$, $\alpha_1\underset{c}{\Rightarrow}\alpha_2$ if and only if $\alpha_1=\beta\gamma$, $\alpha_2=\beta\delta$, and $\gamma\to\delta$ is a production in Π_c for some $\beta\in V^{\otimes}$. $\overset{*}{\underset{c}{\Rightarrow}}$ denotes the reflexive and transitive closure of $\underset{c}{\Rightarrow}$. The *language generated by* G_c, denoted by $CL(G_c)$, is the set

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

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

Definition 7.4. A *k-bounded CG* is a *CG* $G_c=(N,\Sigma,\Pi_c,S)$ such that each production in Π_c is of the form

(1) $S\to\alpha A_1\dots A_k$, where $A_1,\dots,A_k\in N-\{S\}$ and $\alpha\in\Sigma^{\otimes}$, or

(2) $A_1\dots A_k\to\alpha B_1\dots B_k$, where each A_i and B_i is in $N-\{S\}$ for each i $(1\leqslant i\leqslant k)$ and $\alpha\in\Sigma^{\otimes}$.

A k-bounded CG may be regarded as a model of the interaction of k numbers of sequential machines. Also, Crespi-Reghizzi and Mandrioli [5] have shown that k-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 k-bounded CLs is defined in the same way as for EMLs. Let $\Sigma=\{a_1,\ldots,a_k\}$ be an alphabet. The Parikh mapping ψ_c is the function from $\Sigma^{\otimes}$ into $\mathbb{N}^k$ defined by $\psi_c(\omega)=(i_1,\ldots,i_k)$, where $\omega=a_1^{i_1}\ldots a_k^{i_k}$. Note that ψ_c is a one-to-one mapping.

The following proposition is due to Crespi-Reghizzi and Mandrioli [5].

Proposition 7.5. *For any k-bounded CL C, $\psi_c(C)$ is a semilinear set.*

Also, we have the converse.

Proposition 7.6. *Given a positive integer k, an alphabet Σ, and a representation $L(\boldsymbol{c}_1;P_1)\cup\cdots\cup L(\boldsymbol{c}_n;P_n)\}$ of a semilinear subset S of $\mathbb{N}^k$, a k-bounded CG G_c such that $\psi_c(CL(G_c))=S$ is effectively found.*

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

(1) $N=\{S,A_1,\ldots,A_k\}$,

(2) for $\boldsymbol{c}=(c_1,\ldots,c_k)$, $S\rightarrow a_1^{c_1}\ldots a_k^{c_k}A_1\ldots A_k$ is in Π_c,

(3) for each $\boldsymbol{p}_i=(i_1,\ldots,i_k)\in P$, $A_1\ldots A_k\rightarrow a_1^{i_1}\ldots a_k^{i_k}A_1\ldots A_k$ is in Π_c, and

(4) for each A_i $(1\leqslant i\leqslant k)$, $A_i\rightarrow\lambda$ is in Π_c.

It is easy to verify that $\psi_c(CL(G_c))=S$. □

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

We consider the learning problem of k-bounded CLs. In this case, a learning procedure should find a k-bounded CG which is consistent with the given commutative words. By similar arguments in the case of EMLs, the family of k-bounded CLs whose Parikh-images are linear is learnable from positive examples while families of k-bounded CLs whose Parikh-images are n-linear $(n>1)$ are not learnable from positive examples. Furthermore, there exists a learning procedure for k-bounded CLs via restricted subset and restricted superset queries. In particular, for the family of k-bounded CLs whose Parikh-images are linear, there exists a learning procedure which identifies in polynomial time any k-bounded CL whose Parikh-image is linear via restricted superset queries on *any* k-bounded CLs.

Given a commutative grammar, we can effectively construct a Petri net as described in [5]. We simply illustrate in Fig. 9 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 $\Pi_c=\{S\rightarrow aA_1A_2$, $A_1A_2\rightarrow abbA_1A_2$, $A_1A_2\rightarrow\lambda\}$. Then, $\psi_c(CL(G_c))=L((1,0);\{(1,2)\})$.

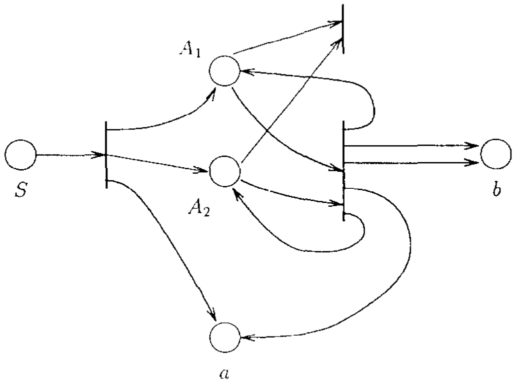


Fig. 9. A Petri net corresponding to a 2-bounded commutative grammar.

8. Concluding remarks

We have shown that the family of semilinear subsets of $\mathbb{N}^k$ 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 via 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 "semantical" property or "behavioral" property. If there is an effective method to construct representations of models from representations 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 representations, 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 representation of its semilinear reachability set. This is one of further research problems.

Acknowledgments

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

References

[1] N. Abe, Polynomial learnability of semilinear sets, in: *Proc. 2nd Workshop on Computational Learning Theory* (1989) 25–40.
[2] N. Abe, Learning commutative deterministic finite state automata in polynomial time, in: *Proc. First Internat. Workshop on Algorithmic Learning Theory* (1990) 223–235.
[3] D. Angluin, Inductive inference of formal languages from positive data, *Inform. and Control* **45** (1980) 117–135.
[4] D. Angluin, Queries and concept learning, *Machine Learning* **2** (1988) 319–342.
[5] S. Crespi-Reghizzi and D. Mandrioli, Petri nets and commutative grammars, Internal Report 74–5, Laboratorio di Calcolatori, Instituto di Electtrotecnica ed Electtronica del Politecnico di Milano, 1974.
[6] S. Eilenberg and M.P. Schützenberger, Rational sets in commutative monoids, *J. Algebra*, **13** (1969) 173–191.
[7] S. Ginsburg, *The Mathematical Theory of Context Free Languages* (McGraw-Hill, New York, 1966).
[8] S. Ginsburg and E.H. Spanier, Bounded Algol-like languages, *Trans. Amer. Math. Soc.* **113** (1964) 333–368.
[9] E. Gold, Language identification in the limit, *Inform. and Control*, **10** (1967) 447–474.
[10] K. Hiraishi and A. Ichikawa, On structural conditions for weak persistency and semilinearity of Petri nets, *Theoret. Comput. Sci.* **93** (1992) 185–199.
[11] O.H. Ibarra, Simple matrix languages, *Inform. and Control*, **17** (1970) 359–394.
[12] H.A. Maurer, G. Rozenberg and E. Welzl, Using string languages to describe picture languages, *Inform. and Control*, **54** (1982) 155–185.
[13] J.L. Peterson, *Petri Net Theory and the Modeling of Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1981).
[14] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[15] R. Siromoney, On equal matrix languages, *Inform. and Control*, **14** (1969) 135–151.
[16] R. Siromoney, A. Huq, M. Chandrasekaran and K.G. Subramanian, Pattern classification with equal matrix grammars, in: *Proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (1990) 440–449.
[17] Y. Takada, On learning semilinear sets: An approach to learning parallel computation models, Research Report 92, IIAS-SIS, Fujitsu Ltd., 1989.
[18] H. Yamasaki, On weak persistency of Petri nets, *Inform. Process. Lett.* **13** (3) (1981) 94–97.