On Grammars Controlled by Parikh Vectors

Ralf Stiebe

Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg,
Postfach 4120, 39106 Magdeburg, Germany
stiebe@iws.cs.uni-magdeburg.de

Abstract. We suggest a concept of grammars with controlled derivations where the Parikh vectors of all intermediate sentential forms have to be from a given restricting set. For several classes of restricting sets, we investigate set-theoretic and closure properties of the corresponding language families.

1 Introduction

Grammars with restricted numbers of nonterminal symbols in the sentential forms in the course of the derivation process have been investigated for a long time. Most prominent are the grammars of finite index, introduced by Brainerd [2], where every word of the generated language can be generated using only sentential forms with a bounded number of nonterminal symbols. These grammars (and regulated grammars of finite index, like matrix grammars, as well) have been studied in numerous publications. Ginsburg and Spanier [4] discussed the slightly different concept of a derivation-bounded grammar where only those derivations are permitted that use sentential forms with a bounded number of nonterminals. While being of finite index is a combinatorial property of the grammar and context-free grammars of finite index can by definition only generate context-free languages, the latter concept provides a kind of control for the derivation process and could potentially lead to the generation of languages not in the original language class. However, it has been shown in [4] that derivation-bounded context-free grammars can only generate context-free languages of finite index.

More recently, Stiebe and Turaev [9] introduced capacity-bounded grammars where a capacity function associates to each nonterminal symbol a bound. A derivation is valid if in every sentential form the number of appearances of each symbol is at most its capacity. It could be shown that context-free capacity-bounded grammars generate non-context-free grammars and are strictly weaker than matrix grammars of finite index.

To overcome the limitations of capacity-bounded grammars, in particular the restriction to sentential forms with a bounded number of nonterminal symbols, we will discuss in this paper some more general conditions for the nonterminals in the sentential forms. Probably the most straightforward extension is to allow infinite capacities for some nonterminal symbols. More generally, we will

H. Bordihn, M. Kutrib, and B. Truthe (Eds.): Dassow Festschrift 2012, LNCS 7300, pp. 246–264, 2012.

demand that, for every sentential form in a derivation process, the Parikh vector (restricted to the nonterminal symbols) has to be in a given restricting set. Grammars with such conditions will be called *Parikh vector controlled grammars* in what follows. Depending on the properties of the restricting sets, several language classes can be defined. We will study the relations of these language classes among each other and to known families of languages as well. When encountering previously unknown language classes, we will also investigate closure properties.

Beside this introduction, the paper contains two sections. The necessary definitions and notations are given in Section 2, in its end introducing the notion of Parikh vector controlled grammars. Section 3 contains the results.

2 Definitions

Throughout the paper, we assume that the reader is familiar with basic concepts of formal language theory; for details we refer to [7]. An introduction to regulated rewriting can be found in [3].

The sets of integers and non-negative integers are denoted by $\mathbb{Z}$ and $\mathbb{N}$, respectively. The cardinality of a set S is denoted by $|S|$, and the power set of a set S by $\mathcal{P}(S)$. We use the symbols $\subseteq$ for inclusion and $\subset$ for proper inclusion. In the vector space $\mathbb{Z}^k$, the zero vector is denoted by $\mathbf{0}$ and the i-th unit vector, $1 \leq i \leq k$, by $\boldsymbol{e}_i$ (a reference to the dimension k will usually not be necessary, as it is clear from the context). A subset $M \subseteq \mathbb{N}^k$ is called *linear* if it can be written as $M = \{\boldsymbol{c} + \sum_{i=1}^{n} a_i \boldsymbol{p}_i : a_i \in \mathbb{N}, 1 \leq i \leq n\}$, for appropriate $\boldsymbol{c}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n \in \mathbb{N}^k$. A set is *semilinear* if it is the union of a finite number of linear sets.

A *system of linear inequalities* in n variables is a finite set of inequalities

$$\sum_{j=1}^{n} a_{i,j} x_j \leq b_i, \ (1 \leq i \leq m) \text{ with } a_{i,j}, b_i \in \mathbb{Z}, \text{ for } 1 \leq i \leq m, 1 \leq j \leq n.$$

A non-negative and integral *solution* of above the system of linear inequalities is a vector $(x_1, x_2, \ldots, x_n) \in \mathbb{N}^n$ that satisfies all inequalities. The set of all non-negative and integral solutions of a system of linear inequalities will, for the sake of brevity, simply be referred to as the *solution set* of the given system. In this paper, a system of linear inequalities as above will be called

- *positive* if $a_{i,j} \geq 0$, for all $1 \leq i \leq m, 1 \leq j \leq n$;
- *strictly positive* if furthermore $\sum_{i=1}^{m} a_{i,j} > 0$, for all $1 \leq j \leq n$.

The solution sets of systems of linear inequalities have some useful properties utilized in this paper. The simple proofs are left to the reader.

1. The solution set of a system of linear equations is semilinear.
2. If S_1 and S_2 are solution sets of systems of linear inequalities with disjoint sets of variables then $S_1 \times S_2$ is the solution set of the system containing all inequalities of both systems. Moreover, if both S_1 and S_2 are (strictly) positive, the resulting system is (strictly) positive, too.

3. The solution set of a strictly positive system of linear inequalities is finite.
4. After an appropriate renaming of the variables, the solution set of a positive system of linear inequalities can be written as $S_1 \times \mathbb{N}^k$ where S_1 is the solution set of a strictly positive system of linear inequalities.
5. If $\boldsymbol{x} \in \mathbb{N}^n$ is a solution of a positive system of linear inequalities then every $\boldsymbol{y} \in \mathbb{N}^n$ with $\boldsymbol{y} \leq \boldsymbol{x}$ is a solution of the same system.

The set of finite strings over an alphabet X is denoted by X^*, the *length* of a string $w \in X^*$ by $|w|$, the number of occurrences of a symbol a in w by $|w|_a$ and the number of occurrences of symbols from $Y \subseteq X$ in w by $|w|_Y$. The *empty* string is denoted by λ. Given an ordered alphabet $X = \{a_1, a_2, \ldots, a_n\}$, the *Parikh mapping* is the homomorphism $\Psi : X^* \to \mathbb{N}^n$ sending a_i, $1 \leq i \leq n$, to the i-th unit vector. For a string $w \in X^*$, $\Psi(w)$ is referred to as the *Parikh vector* of w; for a language $L \subseteq X^*$, the *Parikh set* of L is $\Psi(L) = \{\Psi(w) : w \in L\}$. For a subset Y of X with $Y = \{a_{i_1}, a_{i_2}, \ldots, a_{i_m}\}$, $i_1 < i_2 < \ldots < i_m$, let $\Psi_Y : X^* \to \mathbb{N}^m$ be the homomorphism sending a_{i_j} to the i-th unit vector (of $\mathbb{N}^m$) and $x \in X \setminus Y$ to the zero vector (of $\mathbb{N}^m$). In what follows, for any alphabet, an order will be tacitly assumed so that Parikh mappings are used without explicitly mentioning the order.

Besides the AFL operations (union, concatenation, homomorphisms, inverse homomorphisms, intersection with regular sets, Kleene closure) we will consider *nested iterated substitutions* which were extensively investigated by Greibach [5,6]. A substitution is a homomorphism $\tau : X^* \to \mathcal{P}(Y^*)$ where X and Y are alphabets. We extend τ to $(X \cup Y)^*$, where $X \cap Y = \emptyset$, by defining $\tau(a) = \{a\}$ for all $a \in Y$. For $n \geq 0$, τ^n is the substitution defined by $\tau^0(a) = \{a\}$ and $\tau^{n+1}(a) = \tau(\tau^n(a))$, for $a \in X \cup Y$. The *iterated substitution* defined by τ is the substitution τ^∞ defined by $\tau^\infty(a) = \bigcup_{n=0}^{\infty} \tau^n(a)$, for $a \in X \cup Y$. Moreover, τ^∞ is called a *nested iterated substitution* if $a \in \tau(a)$, for all $a \in X \cup Y$. A family of languages $\mathcal{L}$ is closed under nested iterated substitutions if $L \in \mathcal{L}$ and $\tau(a) \in \mathcal{L}$ for every $a \in X$ imply $\tau^\infty(L) \in \mathcal{L}$. It has been shown in [6] that the family of semilinear languages is closed under nested iterated substitutions.

A *finite automaton* is a tuple $\mathcal{A} = (Z, X, z_0, F, \delta)$ where Z is a finite set of states, X is a finite input alphabet, $z_0 \in Z$ is the initial state, $F \subseteq Z$ is the set of accepting states, and $\delta \subseteq Z \times X \times Z$ is the transition relation. The successor relation $\vdash$ over $Z \times X^*$ is defined as $(z, v) \vdash (z', v')$ iff $v = av'$ and $(z, a, z') \in \delta$. The reflexive and transitive closure of $\vdash$ is denoted by $\vdash^*$. The language accepted by $\mathcal{A}$ is $L(\mathcal{A}) = \{w \in X^* : (z_0, w) \vdash^* (z_f, \lambda), \text{ for some } z_f \in F\}$.

A *grammar* is a quadruple $G = (V, \Sigma, S, R)$ where V and Σ are two finite disjoint alphabets of *nonterminal* and *terminal* symbols, respectively, $S \in V$ is the *start symbol* and $R \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ is a finite set of *rules*. G is called a GS grammar[1] if $R \subseteq V^+ \times (V \cup \Sigma)^*$ and a context-free grammar if $R \subseteq V \times (V \cup \Sigma)^*$. A string $x \in (V \cup \Sigma)^*$ *directly derives* a string $y \in (V \cup \Sigma)^*$ in G, written as $x \Rightarrow_G y$, if and only if there is a rule $\alpha \to \beta \in R$ such that $x = x_1 \alpha x_2$ and $y = x_1 \beta x_2$ for some $x_1, x_2 \in (V \cup \Sigma)^*$. The reflexive

[1] This kind of grammar was introduced by Ginsburg and Spanier and for this reason named GS grammar here.

and transitive closure of the relation $\Rightarrow_G$ is denoted by $\Rightarrow_G^*$. A derivation using the sequence of rules $\pi = r_1 r_2 \cdots r_k$, $r_i \in R$, $1 \leq i \leq k$, is denoted by $\overset{\pi}{\Rightarrow}_G$ or $\overset{r_1 r_2 \cdots r_k}{\Longrightarrow}_G$. The *language* generated by G, denoted by $L(G)$, is defined by $L(G) = \{w \in \Sigma^* : S \Rightarrow_G^* w\}$. If G is clear from the context, the subscript G will be omitted in the notation. The family of languages generated by context-free grammars is denoted **CF**.

We next give some prerequisites concerning grammars with controlled derivations. Unless stated otherwise, extensive explanations, proofs and reference to the original literature can be found in [3]. A *matrix grammar* is a quadruple $G = (V, \Sigma, S, M)$ where V, Σ, S are defined as for a context-free grammar, M is a finite set of *matrices* which are finite strings (or finite sequences) over a set R of context-free rules. The *language* generated by a matrix grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \overset{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $r_1 r_2 \cdots r_n$ is a concatenation of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, $k \geq 1$. The family of languages generated by matrix grammars is denoted by **MAT**.

A *grammar with regular control* is a quintuple $G = (V, \Sigma, S, R, L)$ where $G' = (V, \Sigma, S, R)$ is a context-free grammar and $L \subseteq R^*$ is a regular language. The language of G is defined by $L(G) = \{w \in \Sigma^* : S \overset{\pi}{\Rightarrow} w, \text{ for some } \pi \in L\}$. It is known that the family of languages generated by grammars with regular control is **MAT**.

A *valence grammar over* $\mathbb{Z}^k$ is a quintuple $G = (V, \Sigma, S, R, \mathbb{Z}^k)$ where V, Σ, S are defined as in a context-free grammar, and R is a finite set of valence rules $(A \to \beta, \boldsymbol{r})$, where $A \to \beta$ is a rule and $\boldsymbol{r} \in \mathbb{Z}^k$. The direct derivation relation $\Rightarrow$ over $(V \cup \Sigma)^* \times \mathbb{Z}^k$ is defined by:

$$(\gamma, \boldsymbol{z}) \Rightarrow (\gamma', \boldsymbol{z'}) \text{ iff}$$
$$\gamma = \gamma_1 A \gamma_2, \gamma' = \gamma_1 \beta \gamma_2 \text{ and } \boldsymbol{z'} = \boldsymbol{z} + \boldsymbol{r} \text{ for some } (A \to \beta, \boldsymbol{r}) \in P.$$

The language generated by G is $L(G) = \{w \in T^* : (S, \mathbf{0}) \Rightarrow^* (w, \mathbf{0})\}$.

A *positive valence grammar over* $\mathbb{Z}^k$ [8] is defined like a valence grammar with the additional condition $\boldsymbol{z} \geq \mathbf{0}$ in the definition of the derivation relation $(\gamma, \boldsymbol{z}) \Rightarrow (\gamma', \boldsymbol{z'})$.[2] It has be shown in [8] that the family of languages generated by positive valence grammars is **MAT**.

A *programmed grammar with appearance checking* is defined as a sextuple $G = (V, \Sigma, S, R, \sigma, \phi)$ where (V, Σ, S, R) is a context-free grammar, and σ and ϕ are mappings from R into $\mathcal{P}(R)$. For a rule $r \in R$, $\sigma(r)$ and $\phi(r)$ are called the *success field* and the *failure field* of r, respectively. The derivation relation over $(V \times \Sigma)^* \times R$ is defined as follows. If $r : A \to \alpha$ is a rule in R then $(\beta, r) \Rightarrow (\beta', r')$ iff either $\beta = \beta_1 A \beta_2$, $\beta' = \beta_1 \alpha \beta_2$ and $r' \in \sigma(r)$ or $|\beta|_A = 0$ and $r' \in \phi(r)$. The language generated by G is $L(G) = \{w \in T^* : (S, r) \Rightarrow^* (w, r'), r, r' \in R\}$. It is known that programmed grammars with appearance checking generate the family of recursively enumerable languages [3].

[2] Actually, $\boldsymbol{z} \geq \mathbf{0}$ and $\boldsymbol{z'} \geq \mathbf{0}$ were required in [8]. The definition given here is equivalent to the previous one, since the zero vector has to be reached in the final step. It will be technically useful later.

A context-free grammar G is of *index* $k \in \mathbb{N}$ if every word $w \in L(G)$ has a derivation with at most k nonterminal symbols in every sentential form. G is of *finite index* if such a k exists. The family of languages generated by context-free grammars of finite index is denoted by $\mathbf{CF}_{fi}$. For grammars with regulated rewriting, the concept of finite index is defined analogously. It is known that matrix grammars of finite index and programmed grammars of finite index generate the same family of languages $\mathbf{MAT}_{fi}$.

A *capacity-bounded grammar* [9] is a quintuple $G = (V, \Sigma, S, R, \kappa)$ where $G' = (V, \Sigma, S, R)$ is a grammar and $\kappa : V \to \mathbb{N}$ is a capacity function assigning to each nonterminal a bound. The direct derivation relation $\Rightarrow$ over $(V \cup \Sigma)^* \times \mathbb{Z}^k$ is defined by $\alpha \Rightarrow_G \beta$ iff $\alpha \Rightarrow_{G'} \beta$ and $|\alpha|_A \leq \kappa(A)$ and $|\beta|_A \leq \kappa(A)$, for all $A \in V$. It has been shown that capacity-bounded GS grammars are equivalent to matrix grammars of finite index while capacity-bounded context-free grammars generate a proper subset of $\mathbf{MAT}_{fi}$.

Finally, we give the definition of the generative device to be investigated.

Definition 1. *A* Parikh vector controlled grammar *is defined as a quintuple* $G = (V, \Sigma, S, R, C)$ *where* $G' = (V, \Sigma, S, R)$ *is a grammar and* $C \subseteq \mathbb{N}^n$, $n = |V|$ *is a set of admitted nonterminal Parikh vectors, referred to as the* restricting set *of* G. *The derivation relation* $\Rightarrow_G$ *is defined as* $\alpha \Rightarrow_G \beta$ *iff* $\alpha \Rightarrow_{G'} \beta$ *and* $\Psi_V(\alpha) \in C$. *The language of* G *is defined as* $L(G) = \{w \in \Sigma^* : S \Rightarrow_G^* w\}$.

Note that by this definition only the Parikh vectors of the *nonterminal* sentential forms have to be within the restricting set.

The main objective of this paper is to study the generative power of Parikh vector controlled grammars with respect to properties of the restricting sets. To avoid complicated notations, we just enumerate the types of restricting sets and the respective language families. Let $G = (V, \Sigma, S, R, C)$ be a Parikh vector controlled grammar with $|V| = n$. G is of

- *type 1* if $C = [0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$, $k_1, k_2, \ldots, k_n \in \mathbb{N}$;
- *type 2* if C is the solution set of a strictly positive system of linear inequalities;
- *type 3* if C is finite;
- *type 4* if $C = C_1 \times \mathbb{N}^{n-j}$ where $j \in \{0, 1, \ldots n\}$ and $C_1 = [0, k_1] \times [0, k_2] \times \cdots \times [0, k_j]$, $k_1, k_2, \ldots, k_j \in \mathbb{N}$;
- *type 5* if C is the solution set of a positive system of linear inequalities; (equivalently, if $C = C_1 \times \mathbb{N}^{n-j}$ where $j \in \{0, 1, \ldots n\}$ and $C_1 \subseteq \mathbb{N}^j$ is the solution set of a strictly positive system of linear inequalities;)
- *type 6* if $C = C_1 \times \mathbb{N}^{n-j}$ where $j \in \{0, 1, \ldots n\}$ and C_1 is a finite subset of $\mathbb{N}^j$;
- *type 7* if C is the solution set of a system of linear inequalities;
- *type 8* if C is semilinear.

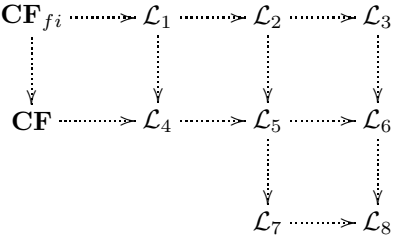
The restricting sets are usually given by defining conditions. Instead of explicitly giving a nonterminal Parikh vector $\Psi_V(\beta)$ we will often refer to its components $|\beta|_A$. In particular, a system of inequalities for a nonterminal alphabet V will sometimes be written as $\sum_{A \in V} a_{i,A} |\beta|_A \leq b_i$ $(1 \leq i \leq m)$.

Note that Parikh vector controlled grammars of type 1 are simply the capacity-bounded context-free grammars and that the types 4,5,6 are extensions of the types 1,2,3, respectively, by adjoining nonterminals that are no subject to any restrictions. In what follows, let $\mathcal{L}_i$, $1 \leq i \leq 8$, denote the family of languages generated by Parikh vector controlled grammars of type i.

3 Results

We will mainly investigate the generative power of Parikh vector controlled grammars of the different types. In cases where the language families do not coincide with previously known families we will also study closure properties with respect to the AFL operations.

Lemma 1. *The following inclusions hold for the language families* $\mathbf{CF}_{fi}$, $\mathbf{CF}$ *and* $\mathcal{L}_1, \ldots, \mathcal{L}_8$ *(a dotted arrow represents a not necessarily proper inclusion; disconnected families need not to be incomparable).*



Proof. All inclusions follow easily from the definitions and some elementary properties. More specifically,

- $\mathbf{CF}_{fi} \subseteq \mathbf{CF}, \mathcal{L}_1 \subseteq \mathcal{L}_4, \mathcal{L}_2 \subseteq \mathcal{L}_5, \mathcal{L}_3 \subseteq \mathcal{L}_6$, $\mathbf{CF} \subseteq \mathcal{L}_4$ and $\mathcal{L}_5 \subseteq \mathcal{L}_7$ hold directly by definition;
- $\mathbf{CF}_{fi} \subseteq \mathcal{L}_1$holds as a grammar (V, Σ, R, S) of finite index k generates the same language as the Parikh vector controlled grammar (V, Σ, R, S, C) where $C = [0, k]^{|V|}$, see [9];
- $\mathcal{L}_1 \subseteq \mathcal{L}_2$ and $\mathcal{L}_4 \subseteq \mathcal{L}_5$ are valid because a set $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$ is the solution set of the system of linear inequalities $x_i \leq k_i$ $(i = 1, 2, \ldots, k)$;
- $\mathcal{L}_2 \subseteq \mathcal{L}_3$ and $\mathcal{L}_5 \subseteq \mathcal{L}_6$ are valid because the solution set of a strictly positive system of linear inequalities is finite;
- $\mathcal{L}_6 \subseteq \mathcal{L}_8$ and $\mathcal{L}_7 \subseteq \mathcal{L}_8$ are true as the restricting sets of grammars of type 6 and 7 are semilinear.

□

It is of course known that the proper inclusion $\mathbf{CF}_{fi} \subset \mathbf{CF}$ holds. Moreover, in [9] the proper inclusions $\mathbf{CF}_{fi} \subset \mathcal{L}_1 \subset \mathbf{MAT}_{fi}$ have been shown. We will now investigate the properness of the remaining inclusions and try to relate the $\mathcal{L}_i$ to known families of languages.

3.1 The Families $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$

We will first study the grammars with a finite restricting set. While $\mathcal{L}_1$ is known to be a proper subfamily of $\mathbf{MAT}_{fi}$, it turns out that $\mathcal{L}_2$ and $\mathcal{L}_3$ coincide with $\mathbf{MAT}_{fi}$. To this end, we will prove the inclusions $\mathcal{L}_3 \subseteq \mathbf{MAT}_{fi}$ and $\mathbf{MAT}_{fi} \subseteq \mathcal{L}_2$.

Lemma 2. $\mathcal{L}_3 \subseteq \mathbf{MAT}_{fi}$.

Proof. Let $G = (V, \Sigma, S, R, C)$ be a Parikh vector controlled grammar with $V = \{A_1, A_2, \ldots, A_n\}$, $S = A_1$ and C a finite subset of $\mathbb{N}^n$. The idea is to construct a grammar with regular control, where the control language is given by a finite automaton whose state keeps track of the nonterminal Parikh vector of the derived sentential form. More specifically, let $\mathcal{A} = (C \cup \{\mathbf{0}\}, R, \boldsymbol{e_1}, \{\mathbf{0}\}, \delta)$ be the deterministic finite automaton with the transition function δ defined as follows. If $r : A_i \to \alpha$ is a rule in R, $\boldsymbol{x} = (x_1, \ldots, x_n)$ is in C and $\boldsymbol{y}$ is defined by $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{e}_i + \Psi_V(\alpha)$ then

$$\delta(\boldsymbol{x}, r) = \begin{cases} \boldsymbol{y}, & \text{if } x_i > 0 \text{ and } \boldsymbol{y} \in C \cup \{\mathbf{0}\} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

It is easy to prove by induction that a sequence $\rho = r_1 r_2 \cdots r_m$ reaches a state $\boldsymbol{z} \in C \cup \{\mathbf{0}\}$ iff ρ is a possible derivation sequence in G and leading to a sentential form with nonterminal Parikh vector $\boldsymbol{z}$. Hence, $L(\mathcal{A})$ is the set of all correct terminal derivation sequences in G and the grammar with regular control $G' = (V, \Sigma, S, R, L(\mathcal{A}))$ generates the same language as G. □

Lemma 3. $\mathbf{MAT}_{fi} \subseteq \mathcal{L}_2$.

Proof. In [9] it has been shown that matrix grammars of finite index are equivalent to capacity-bounded GS grammars. We will therefore show how to simulate a capacity-bounded GS grammar by a Parikh vector controlled grammar with a restricting set defined by a strictly positive system of linear inequalities. Let $G = (V, \Sigma, S, R, \kappa)$ be a capacity-bounded GS grammar. As proved in [9, Lemma 3], we can assume that any word from $L(G)$ can be derived replacing in each derivation step a maximal nonterminal block. (A maximal nonterminal block is a substring over V which cannot be extended to a longer substring over V. As G is capacity-bounded there is only a finite number of maximal nonterminal blocks.) Then we construct the equivalent Parikh vector controlled grammar $G' = (V', \Sigma, [S], R', C)$ where we devise the nonterminal alphabet V' and the set of rules R' like in [9] as

$$V' = \{[\alpha] : \alpha \in V^+ \text{ is a maximal nonterminal block in } G\},$$
$$R' = \{[\alpha] \to w_0[\beta_1]w_1[\beta_2]\cdots w_{k-1}[\beta_k]w_k : \alpha \to w_0\beta_1 w_1\beta_2\cdots w_{k-1}[\beta_k]w_k \in R,$$
$$\text{where } \beta_1, \ldots, \beta_k \in V^+, w_0, w_k \in \Sigma^*, w_1, \ldots, w_{k-1} \in \Sigma^+\}.$$

With $V' = \{[\alpha_1], [\alpha_2] \ldots, [\alpha_n]\}$, the restricting set C is defined as the solution set of the system of linear inequalities

$$\sum_{i=1}^{n} |\alpha_i|_A \cdot x_i \leq \kappa(A), \qquad A \in V.$$

For a word $\beta \in (V \cup \Sigma)^*$ with all maximal blocks in V', let $[\beta] \in (V' \cup \Sigma)^*$ be the word obtained by replacing every maximal block α in β by $[\alpha]$. It is now easily checked by induction on the derivation steps that a sentential form $\beta \in (V \cup \Sigma)^*$ can be derived in G iff all its maximal blocks are in V' and $[\beta]$ can be derived in G'. □

Corollary 1. $\mathcal{L}_1 \subset \mathcal{L}_2 = \mathcal{L}_3 = \mathbf{MAT}_{fi}$.

3.2 The Families $\mathcal{L}_4$, $\mathcal{L}_5$ and $\mathcal{L}_6$

The Parikh vector controlled grammars of types 4,5,6 can be seen as Parikh vector controlled grammars of types 1,2,3, respectively, extended by sets of non-restricted nonterminal symbols. In other words, the nonterminal set V of a Parikh vector controlled grammar of type 4,5 or 6 can be decomposed as $V = V_1 \cup V_2$ where V_1 is subject to the restrictions as in Parikh vector controlled grammars of types 1,2,3, respectively, and V_2 is not at all restricted.

Essentially, we will show that $\mathcal{L}_4$ and $\mathcal{L}_5$ are obtained from $\mathcal{L}_1$ and $\mathcal{L}_2$ by nested iterated substitutions, while $\mathcal{L}_6$ is equal to the family of matrix languages **MAT**. In particular, $\mathcal{L}_4$ and $\mathcal{L}_5$ are proper subfamilies of **MAT**.

We start with the following "replacement lemma" for languages from $\mathcal{L}_4$, which is virtually the same result as for capacity-bounded grammars given in [9].

Lemma 4. *For any infinite language $L \in \mathcal{L}_4$, there are a constant n and a finite set $\mathcal{M}$ of infinite languages from $\mathcal{L}_4$ such that, for every word $z \in L$ with $|z| \geq n$, there are a decomposition $z = uvw$, $|v| \leq n$, and a language $L' \in \mathcal{M}$ such that $uv'w \in L$, for all $v' \in L'$.*

Proof. As the claim of the lemma, the proof is virtually the same as that for capacity-bounded grammars in [9]. Consider some Parikh vector controlled grammar $G = (V, \Sigma, S, R, C)$ of type 4 such that $L = L(G)$. For $A \in V$, let $G_A = (V, \Sigma, R, A, C)$ and $L_A = L(G_A)$; clearly, $L_A = \{w \in \Sigma^* : A \Rightarrow_G^* w\}$. The following assertions hold for any derivation in G involving A:

- If $\alpha A \beta \Rightarrow_G^* uvw$, where $\alpha, \beta \in (V \cup \Sigma)^*$, $u, v, w \in \Sigma^*$ and v is the yield of A, then $v \in L_A$. (Given a derivation $\alpha A \beta \Rightarrow_G^* uvw$, construct a derivation of v from A by keeping the derivation steps arising from A. The Parikh vectors of the sentential forms in the second derivation are less or equal to those of the corresponding sentential forms in the first derivation, hence the derivation $A \Rightarrow_G^* v$ is valid.)

– On the other hand, for all $u, v, w \in \Sigma^*$ such that $v \in L_A$, the relation $uAw \Rightarrow_G^* uvw$ holds. (Given a derivation $A \Rightarrow_G^* v$, do the same derivation steps starting from uAw. The nonterminal Parikh vectors of the sentential forms in the second derivation are equal to those of the corresponding sentential forms in the first derivation, hence the derivation $uAw \Rightarrow_G^* uvw$ is valid.)

The nonterminal set V can be decomposed as $V = V_{inf} \cup V_{fin}$, where

$$V_{inf} = \{A \in V : L_A \text{ is infinite}\},$$
$$V_{fin} = \{A \in V : L_A \text{ is finite}\}.$$

We choose $\mathcal{M} = \{L_A : A \in V_{inf}\}$ and $n = r \cdot \max\{|w| : w \in \bigcup_{A \in V_{fin}} L_A\}$, where r is the longest length of a right side in a rule of R. For a derivation of $z \in L$ with $|z| > n$, consider the last sentential form α with a symbol from V_{inf}. Let this symbol be A. All other nonterminals in α are from V_{fin}, and none of them generates a subword containing A in the further derivation process. We get thus another derivation of z in G by postponing the rewriting of A until all other nonterminals have vanished by applying on them the derivation sequence of the original derivation. This new derivation has the form

$$S \Rightarrow^* \alpha \Rightarrow^* uAw \Rightarrow^* uvw = z.$$

The length of v can be estimated by $|v| \leq n$, as A is in the first step replaced by a word over $(\Sigma \cup V_{fin})$ of length at most r. By the remarks in the beginning of the proof, any word $uv'w$ with $v' \in L_A$ can be derived in G. □

The replacement lemma can be used to show that certain languages are not in $\mathcal{L}_4$. This implies some limitations for $\mathcal{L}_4$, similar to those of $\mathcal{L}_1$ shown in [9].

Corollary 2. *$\mathcal{L}_4$ and $\mathbf{MAT}_{fi}$ are incomparable, while $\mathcal{L}_4$ is a proper subset of $\mathcal{L}_5$.*

Proof. Using the same arguments as in [9], it can be shown that the language $L = \{a^n b^n c^n : n \geq 1\}$ does not satisfy the consequence of the replacement lemma, hence it is not in $\mathcal{L}_4$. On the other hand, L is a language from $\mathbf{MAT}_{fi} = \mathcal{L}_2$ and thus in $\mathcal{L}_5$. Together with the inclusions $\mathbf{CF} \subseteq \mathcal{L}_4 \subseteq \mathcal{L}_5$, this proves the claims. □

Next we prove a useful result concerning derivations in Parikh vector controlled grammars of type 4 or 5.

Lemma 5. *Let $G = (V, \Sigma, S, R, C)$ be a Parikh vector controlled grammar of type 4 or 5, and let $V = V_1 \cup V_2$ be a partition of V such that the appearance of symbols from V_2 is unrestricted. Then every word in $L(G)$ can be derived such that whenever the current sentential form contains a symbol from V_1, a symbol from V_1 will be replaced.*

Proof. Consider a derivation $S \Rightarrow^* \gamma \Rightarrow^* w$ of a word $w \in L(G)$ where the first derivation step replacing a symbol from V_2 although a symbol from V_1 is present, is after generating γ. We will construct a derivation of w with the same number of derivation steps such that a symbol from V_1 is replaced in γ. The claim of the lemma then follows by induction.

We can decompose γ and w as

$$\gamma = \alpha_0 A_1 \alpha_1 A_2 \cdots \alpha_{m-1} A_m \alpha_m,$$
$$w = u_0 v_1 u_1 v_2 \cdots u_{m-1} v_m u_m,$$

where $A_1, A_2, \ldots, A_m$ are the symbols from V_2 in γ, $u_0, u_1, \ldots, u_m$ are the subwords of w derived from $\alpha_0, \alpha_1, \ldots, \alpha_m$, $v_1, v_2, \ldots, v_m$ are the subwords of w derived from $A_1, A_2, \ldots, A_m$.

Consider the derivation $S \Rightarrow^* \gamma \Rightarrow^* \gamma' \Rightarrow^* w$ with

$$\gamma' = u_0 A_1 u_1 A_2 \cdots u_{m-1} A_m u_m,$$

where the derivation steps replacing the A_i and their derivatives are first omitted, thus yielding γ', and then executed in the same sequence, yielding w. This derivation is valid as $u_0, u_1, \ldots, u_m$ and $A_1, A_2, \ldots, A_m$ do not contain symbols from V_1. □

Lemma 6. *Every language $L \in \mathcal{L}_5$ over the alphabet Σ can be represented as $L = L' \cap \Sigma^*$ where L' is the nested iterated substitution of languages from $\mathcal{L}_2$.*

Proof. Let $G = (V, \Sigma, S, R, C_1 \times \mathbb{N}^{n-l})$ be a Parikh vector controlled grammar where $V = \{A_1, A_2, \ldots, A_n\}$ and $C_1 \subseteq \mathbb{N}^l$ is the solution set of a strictly positive system of linear inequalities. The nonterminal set V is partitioned as $V = V_1 \cup V_2$, where $V_1 = \{A_1, \ldots, A_l\}$ is the set of restricted symbols and $V_2 = \{A_{l+1}, \ldots, A_n\}$ is the set of non-restricted symbols. Without loss of generality, we assume that S does not appear on the right-hand side of any rule and belongs to V_2. For every $A \in V_2$, let L_A be the language generated by the Parikh vector controlled grammar $G_A = (V_1 \cup \{A'\}, V_2 \cup \Sigma, A', R_A, C_1 \times [0,1])$ where

$$R_A = \{A' \to \alpha : A \to \alpha \in R\} \cup \{B \to \alpha : B \to \alpha \in R, B \in V_1\}.$$

G_A is of type 2 as $C_1 \times [0,1] \subseteq \mathbb{N}^{l+1}$ is the solution set of the system of inequalities for C_1 (in the variables $x_1, \ldots, x_n$) with the additional inequality $x_{l+1} \leq 1$. Obviously, L_A is the set of all sentential forms that can be derived in G from A by replacing, except for the first step, only symbols from V_1. We claim that $L(G) = \Sigma^* \cap \tau^\infty(S)$ where $\tau(A) = L_A \cup \{A\}$ for $A \in V_2$ and $\tau(a) = \{a\}$ for $a \in \Sigma$.

To prove the inclusion $L(G) \supseteq \Sigma^* \cap \tau^\infty(S)$ we show by induction that every word from $\tau^\infty(S)$ is derivable in G. The induction basis is correct, as $\tau^0(S) = \{S\}$ and S is derivable. Now assume that every word in $\tau^n(S)$ is derivable. By definition, a word $w \in \tau^{n+1}(S)$ can be written as $w = w_1 w_2 \cdots w_m$ where $w_i \in \tau(X_i)$, $X_i \in \Sigma \cup V_2$, and $w' = X_1 X_2 \cdots X_m \in \tau^n(S)$. Now w can be derived in G by first

generating $w' \in \tau^n(S)$ and then deriving sequentially from each X_i the subword w_i. The subderivations $w_1 \cdots w_{i-1} X_i X_{i+1} \cdots X_m \Rightarrow^* w_1 \cdots w_{i-1} w_i X_{i+1} \cdots X_m$ are valid as $w_1, \ldots, w_{i-1}, X_{i+1}, \ldots, X_m$ contain no symbols from V_2.

To prove $L(G) \subseteq \Sigma^* \cap \tau^\infty(S)$ we can restrict to derivations where a symbol from V_1 is replaced when present. We will show by induction that every sentential form over $(V_2 \cup \Sigma)$ obtained in such a derivation is from $\tau^\infty(S)$. The claim is true for S. Now consider some sentential form $\alpha \in (V_2 \cup \Sigma)^*$ with $\alpha \in \tau^\infty(S)$. It is decomposed as $\alpha = \alpha_1 A \alpha_2$ where $A \in V_2$ is the next symbol to be replaced. The next sentential form α' over $(V_2 \cup \Sigma)$ is reached when all symbols from V_1 that originate from the A replaced in the first step are rewritten. Hence, it has the shape $\alpha' = \alpha_1 \beta \alpha_2$ where $\beta \in L_A \subseteq \tau(A)$. By $\alpha_1 \in \tau(\alpha_1)$, $\alpha_2 \in \tau(\alpha_2)$ and the induction hypothesis $\alpha \in \tau^\infty(S)$ we conclude $\alpha' \in \tau^\infty(S)$. □

If the grammar G in the proof is of type 4 then all grammars constructed in the further course are of type 1. This implies:

Corollary 3. *Every language $L \in \mathcal{L}_4$ over the alphabet Σ can be written as $L = L' \cap \Sigma^*$ where L' is the nested iterated substitution of languages from $\mathcal{L}_1$.*

Since all languages in $\mathcal{L}_2 = \mathbf{MAT}_{fi}$ are semilinear and by the closure of the semilinear languages under nested iterated substitution, we can furthermore conclude:

Corollary 4. *Any language in $\mathcal{L}_5$ is semilinear.*

Let us now study the closure properties of $\mathcal{L}_4$ and $\mathcal{L}_5$. As regards $\mathcal{L}_5$, the well-known constructions to show the closure of the context-free languages under the AFL operations can be adapted.

Theorem 1. *The family $\mathcal{L}_5$ is a full AFL.*

Proof. We need to show closure under union, concatenation, Kleene closure, homomorphisms, inverse homomorphisms and intersection with regular languages. Let $G_1 = (V_1, \Sigma, S_1, R_1, C_1)$ and $G_2 = (V_2, \Sigma, S_2, R_2, C_2)$ be Parikh vector controlled grammars of type 5. Without loss of generality, suppose that $V_1 \cap V_2 = \emptyset$. For the mentioned operations, we give now the respective constructions.

Union. Let $G' = (V', \Sigma, S', R', C')$ where $V' = V_1 \cup V_2 \cup \{S'\}$, $R' = \{S' \to S_1, S' \to S_2\} \cup R_1 \cup R_2$ and $C' = C_1 \times C_2 \times \mathbb{N}$.
The first derivation step produces either S_1 or S_2. In the first case, only rules from R_1 are used in the rest of the derivation. Since $\mathbf{0} \in C_2$, the derivation is valid iff every encountered Parikh vector is from $C_1 \times \{\mathbf{0}\} \times \{0\}$, i.e., iff from the second step on it is valid in G_1. Analogously, if S_2 is produced, the derivation is valid iff from the second step on it is valid in G_2. Hence, $L(G') = L(G_1) \cup L(G_2)$.

Concatenation. Set $G' = (V', \Sigma, S', R', C')$ where $V' = V_1 \cup V_2 \cup \{S'\}$, $R' = \{S' \to S_1 S_2\} \cup R_1 \cup R_2$ and $C' = C_1 \times C_2 \times \mathbb{N}$. The first derivation step produces $S_1 S_2$. Since S_1 and S_2 derive only sentential forms over $\Sigma \cup V_1$ and $\Sigma \cup V_2$, respectively, and since $\mathbf{0} \in C_1$, we can restrict to derivations where symbols

from V_1 are replaced as long as they are present. Such a derivation is of the form $S' \Rightarrow S_1S_2 \Rightarrow^*_{G_1} w_1S_2 \Rightarrow^*_{R_2} w_1w_2$ with $w_1w_2 \in \Sigma^*$. The subderivation $S_1S_2 \Rightarrow^*_{G_1} w_1S_2$ is valid iff every Parikh vector is in $C_1 \times \{\boldsymbol{e}_1\} \times \{0\}$, i.e., iff $w_1 \in L(G_1)$ and $\boldsymbol{e}_1 \in C_2$. The subderivation $wS_2 \Rightarrow^*_{R_2} w_1w_2$ is valid iff every Parikh vector is in $\mathbf{0} \times C_2 \times \{0\}$, i.e., iff $w_2 \in L(G_2)$. Since $w_2 \in L(G_2)$ implies $\boldsymbol{e}_1 \in C_2$, the complete derivation is valid iff $w_1 \in L(G_1)$ and $w_2 \in L(G_2)$.

Kleene Closure. Let $G' = (V', \Sigma, S', R', C')$ where $V' = V_1 \cup \{S'\}$, $R' = \{S' \to S_1S', S' \to \lambda\} \cup R_1$ and $C' = C_1 \times \mathbb{N}$. We can restrict to derivations where a symbol from V_1 is replaced if present. Such a derivation has the form

$$S' \Rightarrow S_1S' \Rightarrow^*_{G_1} w_1S' \Rightarrow w_1S_1S' \Rightarrow^*_{G_1} w_1w_2S' \Rightarrow^* w_1w_2 \cdots w_nS' \Rightarrow w_1w_2 \cdots w_n$$

with $w_1, w_2, \ldots, w_n \in \Sigma^*$. A subderivation

$$w_1 \cdots w_{i-1}S' \Rightarrow w_1 \cdots w_{i-1}S_1S' \Rightarrow^*_{G_1} w_1 \cdots w_{i-1}w_iS'$$

is valid iff every encountered sentential form is in $C_1 \times \{1\}$, i.e., iff $w_i \in L(G_1)$.

Homomorphisms. Let $h : \Sigma^* \to \Delta^*$ be a homomorphism. We extend h to a mapping from $(\Sigma \cup V_1)^*$ to $(\Delta \cup V_1)^*$ by setting $h(A) = A$, for all $A \in V_1$. Now set $G' = (V_1, \Delta, S_1, R', C)$ where $R' = \{A \to h(\alpha) : A \to \alpha \in R_1\}$. A sentential form β can be derived in the context-free grammar associated with G_1 iff $h(\beta)$ can be derived in the context-free grammar associated with G'. Moreover, the nonterminal Parikh vectors of β and $h(\beta)$ are equal. Hence, β is derivable in G_1 iff $h(\beta)$ is so in G' and thus $L(G') = h(L(G_1))$.

Inverse Homomorphisms. It suffices to show closure under inverse alphabetic homomorphisms (see, e.g., [1]). Let $h : \Delta^* \to \Sigma^*$ be an alphabetic homomorphism, i.e., a homomorphism sending each $a \in \Delta$ to a word from $\Sigma \cup \{\lambda\}$. Without loss of generality, assume that $\Sigma \cap \Delta = \emptyset$. Then G' is constructed as $G' = (V_1 \cup \Sigma \cup \{\Lambda\}, \Delta, S_1, R', C \times \mathbb{N}^{|\Sigma|+1})$ with $\Lambda \notin V_1 \cup \Sigma \cup \Delta$ and the set of rules

$$\begin{aligned} R' = R_1 \cup \{a' \to \Lambda a, a' \to a\Lambda : a' \in \Sigma, a \in \Delta, h(a) = a'\} \cup \\ \{\Lambda \to a\Lambda : a \in \Delta, h(a) = \lambda\} \cup \{\Lambda \to \lambda\}. \end{aligned}$$

By Lemma 5 we can restrict to derivations where a symbol from V_1 is replaced if present. This way we apply in the first phase the rules from R_1 generating a word $w \in \Sigma^*$. Then the remaining rules can be used to generate an arbitrary word in $h^{-1}(w)$. Since the same restrictions as in G_1 apply to V_1, exactly the words from $L(G_1)$ can be generated in the first phase. In the second phase, a word $w \in L(G_1)$ can be transformed to any of its preimages under h by replacing every symbol in w by one of its preimages under h and inserting symbols from Δ whose images under h is λ. Hence, $L(G') = h^{-1}(L(G_1))$.

Intersection with Regular Sets. Let $\mathcal{A} = (Z, \Sigma, z_0, Q, \delta)$ be a finite automaton. Without loss of generality, assume that Q contains only the single state q.

Construct $G' = (V', \Sigma, S', R', C')$ such that $V' = Z \times (V_1 \cup \Sigma) \times Z$, $S' = (z_0, S, q)$,

$$R' = \{(z, A, z') \to (z, x_1, z_1)(z_1, x_2, z_2) \cdots (z_{r-1} x_r z') : A \to x_1 x_2 \cdots x_r \in R_1\} \cup \{(z, A, z) \to \lambda : A \to \lambda \in R\} \cup \{(z, a, z') \to a : (z, a, z') \in \delta\},$$

and C' is defined such that in the defining system of inequalities for C every term of the form $k \cdot |\alpha|_A$, $k \in \mathbb{N}, A \in V_1$, is replaced by $k \cdot \sum_{z,z' \in Z} |\alpha'|_{(z,A,z')}$. Again, we can restrict to derivations where a symbol from $Z \times V_1 \times Z$ is replaced, if possible. So, in a first derivation phase we generate a word $\beta = (z_0, a_1, z_1)(z_1, a_2, z_2) \cdots (z_{n-1}, a_n, q)$, $a_1, a_2, \ldots, a_n \in \Sigma$. It can be shown by induction that a sentential form can be generated in the first phase iff it has the shape $\alpha' = (z_0, x_1, z_1')(z_1', x_2, z_2') \cdots (z_{m-1}', x_m, q)$, $x_1, x_2, \ldots, x_m \in V_1 \cup \Sigma$ and $\alpha = x_1 x_2 \cdots x_m$ can be derived in G_1. In particular, α' satisfies the constraints of G' iff α satisfies the constraints of G_1 because $|\alpha|_A = \sum_{z,z' \in Z} |\alpha'|_{(z,A,z')}$, for all $A \in V_1$. In a second phase, the intermediate word β can be transformed to $w = a_1 a_2 \cdots a_n$ iff it describes a successful run of $\mathcal{A}$ on w. Hence, w can be generated by G' iff it is in $L(G_1)$ and $L(\mathcal{A})$. □

All constructions but the last result in a Parikh vector controlled grammar of type 4 if both G_1 and G_2 are of type 4. We can conclude:

Corollary 5. *The family $\mathcal{L}_4$ is closed under union, concatenation, Kleene closure, homomorphisms and inverse alphabetic homomorphisms.*

Regarding the remaining two AFL operations, we can prove nonclosure of $\mathcal{L}_4$ by help of the "replacement lemma".

Corollary 6. *$\mathcal{L}_4$ is not closed under intersection with regular sets and inverse homomorphisms.*

Proof. Using Lemma 4, it can be shown that the languages

$$L_1 = \{a^{3n} x^3 y b^{3n} a^{3n} x^2 y b^{3n} : n \geq 1\},$$
$$L_2 = \{a^{3n} c b^{3n} a^{3n} d b^{3n} : n \geq 1\}$$

are not in $\mathcal{L}_4$. However, as discussed in [9], there are a language $L \in \mathcal{L}_1 \subseteq \mathcal{L}_4$, a regular set M and a homomorphism g such that $L_1 = L \cap M$ and $L_2 = g^{-1}(L)$. □

By a slight modification of the proof of Lemma 6, we can also show that $\mathcal{L}_4$ and $\mathcal{L}_5$ are closed under nested iterated substitutions.

Theorem 2. *$\mathcal{L}_4$ and $\mathcal{L}_5$ are closed under nested iterated substitutions.*

A full AFL which is closed under nested iterated substitutions has been termed a *superAFL* by Greibach [6]. We can therefore give the following characterization of $\mathcal{L}_5$.

Corollary 7. *$\mathcal{L}_5$ is the least superAFL containing* $\mathbf{MAT}_{fi}$.

Finally, we are going to prove that grammars of type 6 generate exactly the family of matrix languages. This is achieved by giving simulations showing equivalence to grammars with regular control.

Lemma 7. $\mathcal{L}_6 \subseteq \mathbf{MAT}$.

Proof. The construction is similar to that in the proof of Lemma 2.
Let $G = (V, \Sigma, S, R, C_1 \times \mathbb{N}^{n-l})$ be a Parikh vector controlled grammar where $V = \{A_1, A_2, \ldots, A_n\}$ and $C_1 \subseteq \mathbb{N}^l$ is finite. Without loss of generality we can assume that S does not appear on the right-hand of any rule, $S = A_1$ and $C_1 \subseteq [0,1] \times \mathbb{N}^{l-1}$ (the last assumption implies $l > 0$). We set $V_1 = \{A_1, \ldots, A_l\}$, $V_2 = \{A_{l+1}, \ldots, A_n\}$. The automaton for the regular control language keeps track of the nonterminals from V_1, hence its state set is basically C_1, a subset of $\mathbb{N}^l$. Let $\mathcal{A} = (C_1 \cup \{\mathbf{0}\}, R, \boldsymbol{e_1}, \{\mathbf{0}\}, \delta)$ be the deterministic finite automaton with the transition function δ defined as follows. If $r : A_i \to \alpha$ is a rule in R with $1 \leq i \leq l$, $\boldsymbol{x} = (x_1, \ldots, x_l)$ is in C_1 and $\boldsymbol{y}$ is defined by $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{e}_i + \Psi_{V_1}(\alpha)$ then

$$\delta(\boldsymbol{x}, r) = \begin{cases} \boldsymbol{y}, & \text{if } x_i > 0 \text{ and } \boldsymbol{y} \in C_1 \cup \{\mathbf{0}\} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

If $r : A_i \to \alpha$ is a rule in R with $l < i \leq n$, $\boldsymbol{x} = (x_1, \ldots, x_l)$ is in C_1 and $\boldsymbol{y}$ is defined by $\boldsymbol{y} = \boldsymbol{x} + \Psi_{V_1}(\alpha)$ then

$$\delta(\boldsymbol{x}, r) = \begin{cases} \boldsymbol{y}, & \text{if } \boldsymbol{y} \in C_1 \cup \{\mathbf{0}\} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

It is easy to prove by induction that a sequence $\rho = r_1 r_2 \cdots r_m$ reaches a state $\boldsymbol{z} \in C$ iff ρ is a possible derivation sequence in G and leading to a sentential form α with $\Psi_{V_1}(\alpha) = \boldsymbol{z}$. Hence, $L(\mathcal{A})$ is the set of all correct terminal derivation sequences in G and the grammar with regular control $G' = (V, \Sigma, S, R, L(\mathcal{A}))$ generates the same language as G. □

Lemma 8. $\mathbf{MAT} \subseteq \mathcal{L}_6$.

Proof. Let $G = (V, \Sigma, S, R, L)$ be a grammar with regular control and let $\mathcal{A} = (Z, R, z_0, F, \delta)$ be a finite automaton accepting L. The proof strategy is to construct a Parikh vector controlled grammar of type 6 that simulates the steps of G while simultaneously keeping track of the state of the automaton. Formally, we construct $G' = (V', \Sigma, S', R', C)$ where the set of nonterminals is

$$V' = V \cup \{S'\} \cup Z \cup V_R \cup V_\delta \text{ with } V_R = \{A_r, B_r : r \in R\}, V_\delta = \{X_t, Y_t : t \in \delta\},$$

R' contains the following rules:

- $S' \to z_0 S$;
- for each rule $r : A \to \alpha$ in R, the rules $A \to A_r$, $A_r \to B_r$, $B_r \to \alpha$;

- for each transition $t = (z, r, z')$ in δ, the rules $z \to X_t$, $X_t \to Y_t$, $Y_t \to z'$;
- for each $z_f \in F$, the rule $z_f \to \lambda$;

and C is defined by the following constraints on each nonterminal sentential form in a derivation process:

1. Exactly one symbol from $\{S'\} \cup Z \cup V_\delta$ is present.
2. At most one symbol from V_R is present.
3. If a symbol from Z is present, then no symbol A_r, $r \in R$, is allowed.
4. If a symbol of the form X_t with $t \in \delta$, $t = (z, r, z')$ is present, then the only admissible symbol from V_R is A_r.
5. If a symbol of the form Y_t with $t \in \delta$ is present, then at least one symbol from V_R is present.

First note that G' is indeed of type 6, since the total number of symbols from $\{S'\} \cup Z \cup V_\delta \cup V_R$ is bounded by 2, while the symbols from V are unrestricted.

In the first step of a derivation in G', the rule $S' \to z_0 S$ is applied; the last derivation step of each successful derivation is of the form $z_f w \Rightarrow w$ with $z_f \in F, w \in \Sigma^*$. Now consider a sentential form $z\beta$ with $z \in Z$ and $\beta \in (V \cup \Sigma)^*$ where β contains at least one nonterminal symbol. Because of restriction 3, the symbol z has to be rewritten in the first step using some rule $z \to X_t$; let $t = (z, r, z')$ and $r : A \to \alpha$. In the next step, the rule $X_t \to Y_t$ cannot be applied by restriction 5, so restriction 4 requires the rule $A \to A_r$ to be used. This implies that β can be decomposed as $\beta_1 A \beta_2$ and the sentential form reached after the second step is $X_t \beta_1 A_r \beta_2$. By restriction 2 no other symbol from V can be rewritten in the next derivation steps. Restriction 4 forbids the application of $A_r \to B_r$, so the next applied rule has to be $X_t \to Y_t$ yielding $Y_t \beta_1 A_r \beta_2$. In the next step Y_t cannot be replaced by z' because of restriction 3; hence A_r must be rewritten to reach the sentential form $Y_t \beta_1 B_r \beta_2$. Now the only admissible rule is $Y_t \to z'$ due to restriction 5, giving $z' \beta_1 B_r \beta_2$. Finally, restriction 4 requires the application of $B_r \to \alpha$ which derives $z' \beta_1 \alpha \beta_2$. Hence, every sentential form reachable from $z\beta$ in six steps has the form $z'\beta'$ where β' can be directly derived in G from β using rule r and z can be transferred by r to z' in $\mathcal{A}$. On the other hand, every such sentential form $z'\beta'$ can be derived from $z\beta$ using the above derivation sequence thus completing the proof. □

Corollary 8. $\mathcal{L}_4 \subset \mathcal{L}_5 \subset \mathcal{L}_6 = \mathbf{MAT}$.

3.3 The Families $\mathcal{L}_7$ and $\mathcal{L}_8$

Finally, we discuss grammars whose restricting sets are solution sets of arbitrary systems of linear inequalities (type 7) or semilinear sets (type 8). While the first variant turns out to be equivalent to matrix grammars, the second can generate all recursively enumerable languages.

Lemma 9. $\mathbf{MAT} \subseteq \mathcal{L}_7$.

Proof. The same construction as in the proof of Lemma 8 can be used. We need just to verify that the restrictions on the Parikh sets can be established by a system of linear inequalities. Indeed, the five restrictions can be reformulated as follows:

1. $|\alpha|_{S'} + \sum_{z \in Z} |\alpha|_z + \sum_{X \in V_\delta} |\alpha|_X = 1$.
2. $\sum_{X \in V_R} |\alpha|_X \leq 1$.
3. $\sum_{z \in Z} |\alpha|_z + \sum_{r \in R} |\alpha|_{A_r} \leq 1$.
4. $|\alpha|_{X_t} + \sum_{Y \in V_R \setminus \{A_r\}} |\alpha|_Y \leq 1$, for every $t \in \delta$ where $t = (z, r, z')$.
5. $|\alpha|_{Y_t} - \sum_{Y \in V_R} |\alpha|_Y \leq 0$, for every $t \in \delta$ where $t = (z, r, z')$.

Only the last kind of inequalities is not obvious. It follows since the count of Y_t is limited by one (in condition 2). □

The reverse inclusion $\mathcal{L}_7 \subseteq \mathbf{MAT}$ can be quite easily shown by the construction of a positive valence grammar where the compliance with each of the inequalities is accomplished by a dimension of the valence vector. However, the construction below (Lemma 11) will lead to a positive valence grammar with a slightly different acceptance condition than the usual one. We will therefore first show a technical result regarding positive valence grammars. Let $G = (V, \Sigma, S, R, \mathbb{Z}^k)$ be a positive valence grammar and $\boldsymbol{t} \in \mathbb{Z}^k$ a vector. Then we define $L(G, \boldsymbol{t})$ as the set of all words w for which a derivation

$$(S, \mathbf{0}) \Rightarrow (\alpha_1, \boldsymbol{z}_1) \Rightarrow \cdots \Rightarrow (\alpha_r, \boldsymbol{z}_r) \Rightarrow (w, \boldsymbol{t}) \text{ with } \boldsymbol{z}_i \geq \mathbf{0}, 1 \leq i \leq r,$$

exists. Note that $\boldsymbol{t}$ needs not to be in $\mathbb{N}^k$.

Lemma 10. *For every positive valence grammar G over $\mathbb{Z}^k$ and every $\boldsymbol{t} \in \mathbb{Z}^k$, there is a positive valence grammar G' (over $\mathbb{Z}^{k+1}$) such that $L(G') = L(G, \boldsymbol{t})$.*

Proof. Let $G = (V, \Sigma, S, R, \mathbb{Z}^k)$ be a positive valence grammar and $\boldsymbol{t} \in \mathbb{Z}^k$. The idea of the construction is to add an extra vector $-\boldsymbol{t}$ in the final derivation step. To this end the simulating grammar G' needs for its nonterminal alphabet a copy of V and one additional dimension in the valence vectors. Hence, the nonterminal alphabet of G' is $V \cup V' \cup \{S_0\}$ where V' is a disjoint copy of V and $S_0 \notin V \cup V'$ is the new start symbol. In what follows, the copy of a nonterminal symbol $A \in V$ in V' will be denoted by A'; moreover, the vectors in $\mathbb{Z}^{k+1}$ will be written in the form $(\boldsymbol{y}, z)$ where $\boldsymbol{y} \in \mathbb{Z}^k$ and $z \in \mathbb{Z}$. The set R' of valence rules in G' is defined as

$$\begin{aligned} R' = {} & \{(S_0 \to S, (\mathbf{0}, 1))\} \cup \{(A \to A', (\mathbf{0}, -1)) : A \in V\} \cup \\ & \{(A' \to \alpha, (\boldsymbol{z}, 1) : (A \to \alpha, \boldsymbol{z}) \in R\} \cup \\ & \{(A' \to \alpha, (\boldsymbol{z} - \boldsymbol{t}, 0)) : (A \to \alpha, \boldsymbol{z}) \in R\}. \end{aligned}$$

We will prove by induction over n that a pair $(\beta, (\boldsymbol{y}, z))$ with $z \geq 0$ is derivable in G' in $2n + 1$ steps, iff either

– $(\beta, \boldsymbol{y})$ is derivable in G in n steps and $z = 1$ or
– $(\beta, \boldsymbol{y} + \boldsymbol{t})$ is derivable in G in n steps and $z = 0$.

The assertion is true for $n = 0$ since the only pair derivable in one step in G' is $(S, (\mathbf{0}, 1))$. The first step in a derivation in G' is $(S_0, (\mathbf{0}, 0)) \Rightarrow (S, (\mathbf{0}, 1))$. Now suppose that the assertion has been shown for $n = k$. Consider some pair $(\beta, (\boldsymbol{y}, z))$ derived in G' in $2k+1$ steps. By induction hypothesis, $\beta \in (V \cup \Sigma)^*$ and $z \in \{0, 1\}$ hold. The next derivation step has to apply a rule $(A \to A', (\mathbf{0}, -1))$ yielding $(\beta_1 A' \beta_2, (\boldsymbol{y}, z-1))$ where $\beta = \beta_1 A \beta_2$ is a decomposition of β. If $z = 0$, no further derivation step is possible. If $z = 1$, the next step must use a rule of either of the forms $(A' \to \alpha, (\boldsymbol{z}, 1))$ or $(A' \to \alpha, (\boldsymbol{z} - \boldsymbol{t}, 0))$ in order to keep the last component non-negative. In the first case, the resulting pair is $(\beta_1 \alpha \beta_2, (\boldsymbol{y} + \boldsymbol{z}, 1))$. The step is valid iff $\boldsymbol{y} + \boldsymbol{z} \geq \mathbf{0}$, i.e., iff $(\beta_1 \alpha \beta_2, \boldsymbol{y} + \boldsymbol{z})$ is directly derivable from $(\beta, \boldsymbol{y})$ in G. In the second case, the resulting pair is $(\beta_1 \alpha \beta_2, (\boldsymbol{y} + \boldsymbol{z} - \boldsymbol{t}, 0))$. Hence, the induction hypothesis is true for $n = k + 1$.

Since all sentential forms generated in $2n$ steps by G' are nonterminal, the language of G' is found as

$$\begin{aligned} L(G') &= \{w \in \Sigma^* : (w, (\mathbf{0}, 0)) \text{ derivable in } G' \text{ in } 2n+1 \text{ steps }, n \geq 0\} \\ &= \{w \in \Sigma^* : (w, \boldsymbol{t}) \text{ derivable in } G \text{ in } n \text{ steps }, n \geq 0\} = L(G, \boldsymbol{t}), \end{aligned}$$

as claimed. □

Lemma 11. $\mathcal{L}_7 \subseteq \mathbf{MAT}$.

Proof. Let $G = (V, \Sigma, S, R, C)$ be a Parikh vector controlled grammar where $V = \{A_1, A_2, \ldots, A_n\}$, $S = A_1$ and $C \subseteq \mathbb{N}^n$ is the solution set of a system of m linear inequalities

$$\sum_{j=1}^{n} a_{i,j} x_j + b_i \geq 0, \ (1 \leq i \leq m).$$

We construct the positive valence grammar $G' = (V \cup \{S'\}, \Sigma, S', R', \mathbb{Z}^m)$ with the start symbol $S' \notin V$ and the set of valence rules R' constructed as follows.

– The starting rule is $(S' \to A_1, (z_1, \ldots, z_m))$ with $z_i = a_{i,1} + b_i$, $1 \leq i \leq m$.
– For any rule $A_r \to \alpha$ with $\Psi_V(\alpha) = (y_1, \ldots, y_n)$, R' contains the valence rule $(A_r \to \alpha, (z_1, \ldots, z_m))$ with $z_i = \sum_{j=1}^{n} a_{i,j} y_j - a_{i,r}$.

It is easy to verify by induction on the number of derivation steps that G' can generate a pair $(\alpha, (z_1, \ldots, z_m))$ iff G can generate α and $\Psi_V(\alpha) = (x_1, \ldots, x_n)$ satisfies $z_i = \sum_{j=1}^{n} a_{i,j} x_j + b_i$, for $1 \leq i \leq m$. Hence, G produces the same language as G' with the target vector $(b_1, \ldots, b_m)$. □

Lemma 12. $\mathcal{L}_8$ *is the family of recursively enumerable languages.*

Proof. We will simulate a programmed grammar with appearance checking by a Parikh vector controlled grammar with a semilinear restricting set. Let $G = (V, \Sigma, S, R, \sigma, \phi)$ be a programmed grammar with appearance checking. Then we construct the Parikh vector controlled grammar $G' = (V', \Sigma, S', R', C)$ where

$$\begin{aligned} V' &= V \cup \{S'\} \cup \{X_r, Y_r, Z_r, F_r, A_r, B_r : r \in R\}, \\ R' &= \{S' \Rightarrow SX_r : r \in R\} \cup \\ &\quad \{X_r \to Y_r, Y_r \to Z_r : r \in R\} \cup \{Z_r \to X_s : s \in \sigma(r)\} \cup \\ &\quad \{X_r \to F_r : r \in R\} \cup \{F_r \to X_f : f \in \phi(r)\} \cup \\ &\quad \{A \to A_r, A_r \to B_r, B_r \to \alpha : (r : A \to \alpha) \in R\}, \end{aligned}$$

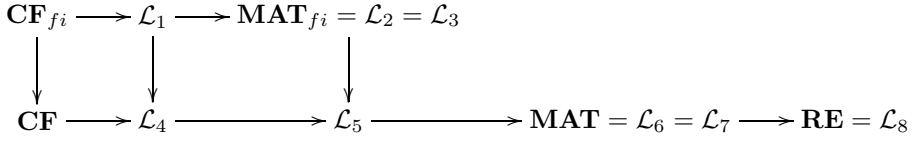
and C is defined by the following constraints on the Parikh vector for a nonterminal sentential form:

1. One symbol from $\{S'\} \cup \{X_r, Y_r, Z_r, F_r : r \in R\}$ is present.
2. At most one symbol from $\{A_r, B_r : r \in R\}$ is present.
3. If X_s is present then no symbol A_r is present, $r, s \in R$.
4. If Y_s is present then no symbol B_r is present, $r, s \in R$.
5. If Z_r is present then one of the symbols A_r, B_r is present.
6. If F_s is present, $s \in R$, then no symbol from $\{A_r, B_r : r \in R\}$ is present.
7. If F_r is present, for $r : A \to \alpha$, then A is not present.

It is easy to see that each of the constraints describes a semilinear set. The set C is the intersection of all these sets and thus semilinear, too. The correctness proof is similar to that in Lemma 7. The last constraint models the appearance checking case. It is the only one that cannot be described by a system of linear inequalities. □

4 Conclusions

We have introduced Parikh vector controlled grammars and investigated several restrictions on the Parikh sets of sentential forms. The results concerning the generative power with respect to the different restrictions can be summarized as follows (arrows indicating strict inclusions, disconnected families being incomparable).



A particularly interesting family is $\mathcal{L}_5$ defined by grammars whose restricting sets are solutions of positive systems of linear inequalities. This language family does not coincide with any of the formerly known classes and is a superAFL of semilinear languages.

It remains to study the power of non-erasing grammars of the respective types. It might be also worthwhile to investigate connections to other variants of regulated rewriting. For instance, a characterization of random context grammars by an appropriate Parikh vector control could be helpful to settle the longstanding question if random context grammars are equivalent to matrix grammars.

References

1. Berstel, J.: Transductions and Context-Free Languages. Teubner-Verlag, Stuttgart (1979)
2. Brainerd, B.: An analog of a theorem about context-free languages. Information and Control 11, 561–567 (1968)
3. Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. Springer (1989)
4. Ginsburg, S., Spanier, E.: Derivation bounded languages. Journal of Computer and System Sciences 2, 228–250 (1968)
5. Greibach, S.: Full AFLs and nested iterated substitution. Information and Control 16(1), 7–35 (1970)
6. Greibach, S.: A generalization of Parikh's semilinear theorem. Discrete Mathematics 2, 347–355 (1972)
7. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)
8. Stiebe, R.: Positive valence grammars. In: Csuhaj-Varjú, E., Kintala, C., Wotschke, D., Vaszil, G. (eds.) Fifth International Workshop Descriptional Complexity of Formal Systems, pp. 186–197. MTA SZTAKI, Budapest (2003)
9. Stiebe, R., Turaev, S.: Capacity bounded grammars. Journal of Automata, Languages and Combinatorics 15, 175–194 (2010)