Indonesian Journal of Electrical Engineering and Computer Science
Vol. 8, No. 1, October 2017, pp. 36 ~ 42
DOI: 10.11591/ijeecs.v8.i1.pp36-42

Multiset Controlled Grammars: A Normal Form and Closure Properties

Salbiah Ashaari[1], Sherzod Turaev*[2], M. Izzuddin M. Tamrin[3], Abdurahim Okhunov[4], Tamara Zhukabayeva[5]
[1,2,3]Kulliyyah of Information and Communication Technology, International Islamic University Malaysia 53100 Kuala Lumpur, Malaysia
[4]Kulliyyah of Engineering, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia
[5]Faculty of Information Technology, L.N. Gumilyov Eurasian National University, 010008 Astana, Kazakhstan
*Corresponding author, e-mail: salbiah.ash@gmail.com[1], sherzod@iium.edu.my[2], izzuddin@iium.edu.my[3], abdurahimokhun@iium.edu.my[4], tamara_kokenovna@mail.ru[5]

Abstract

Multisets are very powerful and yet simple control mechanisms in regulated rewriting systems. In this paper, we review back the main results on the generative power of multiset controlled grammars introduced in recent research. It was proven that multiset controlled grammars are at least as powerful as additive valence grammars and at most as powerful as matrix grammars. In this paper, we mainly investigate the closure properties of multiset controlled grammars. We show that the family of languages generated by multiset controlled grammars is closed under operations union, concatenation, kleene-star, homomorphism and mirror image.

Keywords*: multiset, regulated grammar, multiset controlled grammar, generative capacity, closure property*

1. Introduction

A regulated grammar is depicted as a grammar with an additional (control) mechanism that able to restrict the use of the productions (a.k.a. *rules*) during derivation process in order to avoid certain derivations as well as to obtain a subset of the language generated in usual way. The primary motivation for introducing regulated grammars came from the fact that a plenty of languages of interest are seen to be non-context free such as the languages with reduplication, multiple agreement or crossed agreement properties. Therefore, the main aim of regulated grammars is to achieve a higher computational power and yet at the same time does not increase the complexity of the model [1, 2].

It is believed that the first regulated grammar, which is a matrix grammar, was introduced by Abraham in 1965 with the idea such in a derivation step, a sequence of productions are applied together [3]. Since then, a plenty of regulated grammars have been introduced and investigated in several papers such as [1-13], where each has a different control mechanism, and provides useful structures to handle a variety of issues in formal languages, programming languages, DNA computing, security, bioinformatics and many other areas.

In this paper, we continue our research on multiset controlled grammars (see [4]); we investigate the closure properties of the family of languages generated by multiset controlled grammars. The study of closure properties is one of a crucial investigation in formal language theory since it provides a meaningful merit in both theory and practice of grammars. This paper is structured as follows. First, we give some basic notations and knowledge concerning to the theory of formal languages in which include grammars with regulated rewriting and set-theoretic operations on languages that will be used throughout the study. Then, we recall the definitions of multiset controlled grammars defined in [4] together with results on their generative power. Then, we demonstrate that for multiset controlled grammars one can construct equivalent normal forms, which will be used in study of the closure properties. In the last section, we put in a nutshell all materials studied previously with some possible future research topics on those grammars.

Received June 6, 2017; Revised August 16, 2017; Accepted September 1, 2017

2. Preliminaries

In this section, we present some basic notations, terminologies and concepts concerning to the formal languages theories, multiset and regulated rewriting grammars that will be used in the following sections. For details, the reader is referred to [1, 2, 14-16]. Throughout the paper, we use the following basic notations. Symbols $\in$ and $\notin$ represent the set membership and negation of set membership of an element to a set. Symbol $\subseteq$ signifies the set inclusion and $\subset$ marks the strict inclusion. Then, for a two sets A and B, $A \subsetneq B$ if $A \subseteq B$ and $A \neq B$. Further, notation $|A|$ is used to portray the cardinality of a set A in which is the number of elements in the set A as well as notation 2^A is used to depict the power set of a set A. Symbol $\emptyset$ denotes the empty set. The sets of integer, natural, real and rational numbers are denoted by $\mathbb{Z}, \mathbb{N}, \mathbb{R}$ and $\mathbb{Q}$, respectively.

An alphabet is a finite and nonempty set of symbols or letters, which is denoted by Σ and a string (sometimes referred as word) over the alphabet Σ is a finite sequence of symbols (concatenation of symbols) from Σ. The string without symbols is called null or empty string and denoted by λ. The set of all strings (including λ) over the alphabet, Σ is denoted by Σ^* and $\Sigma^+ = \Sigma^* - \{\lambda\}$. A string w is a substring of a string v if and only if there exist u_1, u_2 such that $v = u_1 w u_2$ where $u_1, u_2, w, v \in \Sigma^*$. String w is a proper substring of v if $w \neq \lambda$ and $w \neq v$. The length of string w, denoted by $|w|$, is the number of symbols in the string. Obviously, $|\lambda| = 0$ and $|wv| = |w| + |v|,\ w, v \in \Sigma^*$. A language L is a subset of Σ^*. A language L is λ -free if $\lambda \notin L$. For a set A, a multiset is a mapping $\mu: A \to \mathbb{N}$. The set of all multisets over A is denoted by $A^{\oplus}$. Then, the set A is a called the basic set of $A^{\oplus}$. For a multiset $\mu \in A^{\oplus}$ and element $a \in A$, $\mu(a)$ represents the number of occurrences of a in μ.

A Chomsky grammar is a quadruple $G = (N, T, S, P)$ where N is an alphabet of nonterminals, T is an alphabet of terminals and $N \cap T = \emptyset$, $S \in N$ is the start symbol and P is a finite set of productions such that $P \subseteq (N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$. We simply use notation $A \to w$ for a production $(A, w) \in P$. A direct derivation relation over $(N \cup T)^*$, denoted by $\Rightarrow$, is defined as $u \Rightarrow v$ provided if and only if there is a rule $A \to w \in P$ such that $u = x_1 A x_2$ and $v = x_1 w x_2$ for some $x_1, x_2 \in (N \cup T)^*$. Since $\Rightarrow$ is a relation, then its nth, $n \geq 0$, power is denoted by $\Rightarrow^n$, its transitive closure by $\Rightarrow^+$, and its reflexive and transitive closure by $\Rightarrow^*$. A string $w \in (N \cup T)^*$ is a sentential form if $S \Rightarrow^* w$. If $w \in T^*$, then w is called a sentence or a terminal string and $S \Rightarrow^* w$ is said to be a successful derivation. We also use the notations $\overset{m}{\Rightarrow}$ or $\xrightarrow{r_0 r_1 \dots r_n}$ to denote the derivation that uses the sequence of rules $m = r_0 r_1 \dots r_n,\ r_i \in P, 1 \leq i \leq n$. The language generated by G, denoted by $L(G)$, is defined as $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$. Two grammars G_1 and G_2 are called to be equivalent if and only if they generate the same language, i.e., $L(G_1) = L(G_2)$. There are five main types of grammars depending on their productions forms $u \to v$:

a) a regular if $v \in T^* \cup NT^*$ and $u \in N$,
b) a linear if $v \in T^* \cup T^* N T^*$ and $u \in N$,
c) a context-free if $v \in (N \cup T)^*$ and $u \in N$,
d) a context-sensitive if $u \in (N \cup T)^* N^+ (N \cup T)^*$ and $v \in (N \cup T)^+$ where $|u| \leq |v|$ and
e) a recursively enumerable or unrestricted if $u \in (N \cup T)^+$ and $v \in (N \cup T)^*$ where u contains at least one nonterminal symbol.

The families of languages generated by arbitrary, context sensitive, context free, regular, linear and finite grammars are denoted by $\mathbf{RE}, \mathbf{CS}, \mathbf{CF}, \mathbf{REG}, \mathbf{LIN}$, and $\mathbf{FIN}$, respectively. For these language families, Chomsky hierarchy holds:

$$\mathbf{FIN} \subset \mathbf{REG} \subset \mathbf{LIN} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{RE}.$$

Before moving to the operations on languages, we recall some definition of regulated grammars mentioned in this study. A matrix grammar is a quadruple $G = (N, T, S, M)$ where N, T and S are defined as for context-free grammar and M is a set of matrices, that are finite sequences of context-free rules from $N \times (N \cup T)^*$. Its language is defined by $L(G) = \{w \in T^* \mid S \overset{\pi}{\Rightarrow} w \text{ and } \pi \in M^*\}$.

An additive valence grammar is a 5-tuples $G = (N, T, S, P, v)$ where N, T, S, P are defined as for a context-free grammar and v is a mapping from P into set $\mathbb{Z}$ (set $\mathbb{Q}$). The language generated by the grammar consists of all string $w \in T^*$ such that there is a derivation S

$\xRightarrow{r_1 r_2 \dots r_n} w$ where $\sum_{k=1}^{n} v(r_k) = 0$. The families of languages generated by matrix and additive valence grammars (with erasing rues) are denoted by $\mathbf{MAT}, a\mathbf{VAL}$, ($\mathbf{MAT}^{\lambda}, a\mathbf{VAL}^{\lambda}$), respectively. Now, we recall some set-theoretic operations that will be used to investigate the closure properties of a grammar. Let L_1 and L_2 be two languages. Then, the union (∪), intersection (∩), difference (−) and concatenation (·) of L_1 and L_2 are defined as:

$$L_1 \cup L_2 = \{w : w \in L_1 \ or \ w \in L_2\}$$
$$L_1 \cap L_2 = \{w : w \in L_1 \ and \ w \in L_2\}$$
$$L_1 - L_2 = \{w : w \in L_1 \ and \ w \notin L_2\}$$
$$L_1 \cdot L_2 = \{w_1 w_2 : \ w_1 \in L_1 \ and \ w_2 \in L_2\}$$

The complement of $L \subseteq \Sigma^*$ with respect to Σ^* is defined as $\bar{L} = \Sigma^* - L$. For a language L, the iterations of L is defined as:

$L^0 = \{\lambda\}$,
$L^1 = L$,
$L^2 = LL$,
...
$L^* = \bigcup_{i \geq 0} L^i$ *(iteration closure: Kleene star).*
$L^+ = \bigcup_{i \geq 1} L^i$ (positive iteration closure: Kleene plus).

Given two alphabets Σ_1, Σ_2, a mapping $h: {\Sigma_1}^* \to {\Sigma_2}^*$ is called a morphism or synonymously a homomorphism if and only if:
(i) for every $w \in {\Sigma_1}^*$, there exists $v \in {\Sigma_2}^*$ such that $v = h(w)$ and v is distinct,
(ii) for every $w, v \in {\Sigma_1}^* : h(wv) = h(w)h(v)$.

A morphism is λ-free if for every $w \in {\Sigma_1}^*$, $h(w) \neq \lambda$. Then, a morphism is known as an isomorphism when for every $w, v \in {\Sigma_1}^*$, if $h(w) = h(v)$ then $w = v$.

For a word $w = w_1 w_2 \dots w_n$, $w_i \in \Sigma, 1 \leq i \leq n$, the mirror image of w (a.k.a. reversal) is obtaining by writing the word w in the reverse order such $w_n \dots w_1 w_2$ and it is denoted by w^R. Therefore, for a language $L \subseteq \Sigma^*$, we defined its mirror image as $L^R = \{w : w^R \in L\}$.

3. Multiset Controlled Grammars: Definitions and Generative Power

Informally, a multiset controlled grammar is a context-free grammar $G = (N, T, S, P)$ equipped with an arithmetic expression over multisets on terminals. For each production $r: A \to w \in P$, the multiset $\omega[r] \in T^{\oplus}$, called "counter", is defined representing the terminal occurrences in w. For example, if $T = \{a, b\}$ and $r: A \to aaAb \in P$ is a production, then $\omega[r] = (2, 1)$. A derivation in the grammar is successful if only if the value of the expression of multisets resulted from the derivation in a true relation with a certain threshold α [4]. The formal definition of multiset controlled grammars is portrayed in the following definition.

Definition 1 [4] A multiset controlled grammar is a 6-tuples $G = (N, T, S, P, \oplus, F)$ where N, T and S are defined as for a context-free grammar, P is a finite subset of $N \times (N \cup T)^* \times T^{\oplus}$ and $F: T^{\oplus} \to \mathbb{Z}$ is a linear or nonlinear function. A triple $(A, w, \omega) \in P$ is written as $A \to w[\omega]$. If $F(a_1, a_2, \dots, a_n), a_i \in T, 1 \leq i \leq n$ is a linear, then it is in the form of $F(a_1, a_2, \dots, a_n) = \sum_{i=1}^{n} c_i \mu(a_i) + c_0$ where $c_i \in \mathbb{Z}, 0 \leq i \leq n$. Then, as a nonlinear function F, we can consider logarithms, polynomials, rational, exponential, power and so on. Thus, the language generated by multiset controlled grammar is defined by $L(G, \alpha, *) = \{w \mid w[\omega] \in L(G),\ F(\omega) * \alpha\}$ where the relation $* \in \{=, <, >, \leq, \geq\}$, $\alpha \in W, W \subseteq \mathbb{Z}$ is a cut point set and $L(G) = \{w[\omega] \in T^* \times T^{\oplus} \mid S \stackrel{\pi}{\Rightarrow} w_n[\omega_n]$ where $\pi = r_1 r_2 \cdots r_n\}$.

Definition 2 A multiset controlled grammar $G = (N, T, S, P, \oplus, F)$ is called
a) regular if each production in the grammar has the form such $A \to wB[\omega]$ or $A \to w[\omega]$ where $w \in T^*$ and $A, B \in N$.
b) linear if each production in the grammar has the form such $A \to w_1 B w_2[\omega]$ or $A \to w[\omega]$ where $w, w_1, w_2 \in T^*$and $A, B \in N$.
c) context-free if each production in the grammar has the form such $A \to x[\omega]$ where $x \in (N \cup T)^*$ and $A \in N$.

The families of languages generated by multiset controlled regular, linear and context-free grammars with and without erasing rules are denoted by $m\mathbf{REG}, m\mathbf{LIN}, m\mathbf{CF}$ ($m\mathbf{REG}^{\lambda}, m\mathbf{LIN}^{\lambda}, m\mathbf{CF}^{\lambda}$) respectively. We also use bracket notation $m\mathbf{X}^{[\lambda]}, \mathbf{X} \in \{\mathbf{REG}, \mathbf{LIN}, \mathbf{CF}\}$ to show that a statement holds both cases of with and without erasing rules.The following theorem shows the computational powers possessed by multiset controlled grammars. Combining the results above, we form the following relations as in Figure 1.

Theorem 1 [4]

- $\mathbf{REG} \subset m\mathbf{REG}$.
- $\mathbf{CF} \subset m\mathbf{CF}$.
- $\mathbf{LIN} \subset m\mathbf{LIN}$.
- $m\mathbf{REG} - \mathbf{CF} \neq \emptyset$.
- $m\mathbf{CF} - a\mathbf{VAL} \neq \emptyset$.
- $m\mathbf{CF}^{[\lambda]} \subseteq \mathbf{MAT}^{[\lambda]}$.

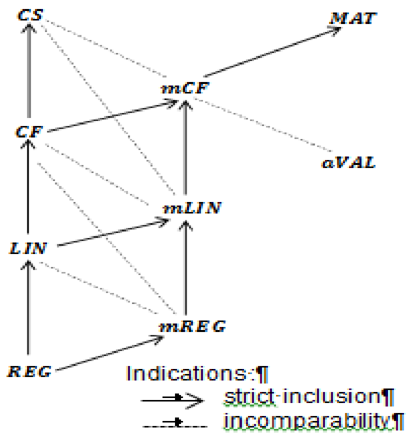

Figure 1. The hierarchy of families of language generated by multiset controlled grammar.

4. A Multiset Chomsky Normal Form

A normal form is introduced with the intent to transform a grammar into a standard form by imposing it with restrictions. In formal language theory, a variety of normal forms were first investigated and developed to solve the rudimentary problems involving context-free languages such for making it easy to analyze and to construct proofs regarding particular properties of the grammars, i.e., for testing emptiness and deciding membership matters with more easily. One of the most useful, well-constructed and popular normal forms to deal with context-free grammar is Chomsky normal form (CNF) due to its simple structure (binary tree). The use of CNF allows easily to determine whether a string is generated by the context-free grammar or not using polynomial time algorithms (for instance, CYK algorithm).

A context-free grammar is said to be in CNF if and only if all its productions are in form of $A \to XY$ and $A \to x$ where A, X, Y are variables and x is exactly a terminal. Here, we prove that our multiset controlled context-free grammars can also be transformed in to equivalent CNFs.

Theorem 1 For any multiset controlled context-free grammar G_m, there exists an equivalent multiset controlled context-free grammar $G_m{}'$ in multiset Chomsky normal form (mCNF).

Proof: Let G_m be a multiset controlled context-free grammar. Then, any such grammar can be converted into an equivalent grammar $G_m{}'$ where all its productions are in form of $A \overset{\mathbf{0}}{\to} XY$ or/and $A \overset{\omega}{\to} x$ with $\omega > 0$, where $\mathbf{0}$ is the zero vector, A, X, Y are variables and x is a terminal. It is done in three phases.

Phase 1. We construct a grammar G_1 that is equivalent to grammar G_m and does not have any production in the form of $A \to X$ where $A, X \in N$. Suppose that we have productions $A \longrightarrow X$ in G_m that lead to a series of form of derivation such

$$A \overset{\omega_1}{\to} X_1 \overset{\omega_2}{\to} X_2 \overset{\omega_3}{\to} \cdots \overset{\omega_n}{\to} X_n \overset{\omega_{n+1}}{\longrightarrow} X \overset{\omega_{n+2}}{\longrightarrow} p \text{ with } p \notin N.$$

Accordingly, we substitute all such "sequence" productions $A \overset{\omega_1}{\rightarrow} X_1, X_1 \overset{\omega_2}{\rightarrow} X_2, \dots, X_n \xrightarrow{\omega_{n+1}} X$ in G_m by a single production $A \overset{\omega}{\rightarrow} p$, where $\omega = \omega_1 + \omega_2 + \cdots + \omega_{n+1}$. Thus, the grammar G_1 is equivalent to the grammar G_m.

Phase 2. We construct a grammar G_2 that is equivalent to grammar G_1 with condition such all productions in G_2 are not in the form of

$$A \overset{\omega}{\rightarrow} x_1 x_2 \cdots x_n, \omega > 0, n > 2$$

where x_is are terminals. For every rule of the form above, we introduce a new rule $A \overset{\mathbf{0}}{\rightarrow} Y_1 Y_2 \cdots Y_n$ where all where x_i terminals are replaced with new variables Y_is, and rules of the form $Y_i \overset{\mathbf{1}_i}{\rightarrow} x_i$ for each x_i where $\mathbf{1}_i$ is the vector containing a signle one which is at position i. Therefore, we get G_2 with all its productions are only in the forms $A \overset{\omega}{\rightarrow} x, x \in T$ or/and $A \overset{\mathbf{0}}{\rightarrow} X_1 \cdots X_n$, $n \geq 2$, $X_1, X_2, \dots, X_n \in N$. Here, it is obvious that grammar G_2 is equivalent to grammar G_1.

Phase 3. We construct a grammar G'_m that is equivalent to grammar G_2 where all its productions are only in the form of $A \overset{\omega}{\rightarrow} x$ or $A \overset{\mathbf{0}}{\rightarrow} XY$ with $\omega > 0$, $A, X, Y \in N$ and $x \in T$. Consider a production in form of $A \overset{0}{\rightarrow} X_1 \cdots X_n$ with $n > 2$ in G_2. Then, we substitute this production with the productions

$$\begin{aligned} &A \overset{0}{\rightarrow} X_1\, Y_1 \\ &Y_1 \overset{0}{\rightarrow} X_2\, Y_2 \\ &\quad\vdots \\ &Y_{n-2} \overset{0}{\rightarrow} Y_{n-1}\, Y_n \end{aligned}$$

where $Y's$ are new nonterminals. Thus, the obtained grammar G'_m is equivalent to grammar G_m, which is multiset Chomsky normal form.

Example 1 Let $G_1 = (\{A, B, S\}, \{a, b, c\}, S, P, \bigoplus, F)$ be a multiset context-free grammar where P consists of the following productions:

$r_0 : S \rightarrow AB[(0,0,0)]$,
$r_1 : A \rightarrow aAb[(1,1,0)]$,
$r_2 : B \rightarrow cB[(0,0,1)]$,
$r_3 : A \rightarrow ab[(1,1,0)]$,
$r_4 : B \rightarrow c[(0,0,1)]$,
and $F(a, b, c) = \mu(a) + \mu(b) + (-1)\mu(b) + (-1)\mu(c)$.

Then, to convert the grammar G_1 into Chomsky normal form, we proceed as below:
First, replace $A \rightarrow aAb[(1,1,0)]$ with $A \rightarrow T_a A T_b[(0,0,0)]$, $T_a \rightarrow a[(1,0,0)]$, $T_b \rightarrow b[(0,1,0)]$. Then, $B \rightarrow cB[(0,0,1)]$ by $B \rightarrow T_c B[(0,0,0)]$, $T_c \rightarrow c[(0,0,1)]$. Next, $A \rightarrow ab[(1,1,0)]$ by $A \rightarrow T_a T_b[(0,1,0)]$. Last, replace $A \rightarrow T_a A T_b[(0,0,0)]$ by $A \rightarrow T_a C[(0,0,0)]$ and $C \rightarrow A T_b[(0,0,0)]$.

Hence, we can have a multiset controlled context free grammar in Chomsky normal form with productions such:

$r_0 : S \rightarrow AB[(0,0,0)]$,
$r_1 : A \rightarrow T_a C[(0,0,0)]$,
$r_2 : C \rightarrow A T_b[(0,0,0)]$,
$r_3 : B \rightarrow T_c B[(0,0,0)]$,
$r_4 : A \rightarrow T_a T_b[(0,0,0)]$,
$r_5 : B \rightarrow c[(0,0,1)]$,
$r_6 : T_a \rightarrow a[(1,0,0)]$,
$r_7 : T_b \rightarrow b[(0,1,0)]$,
$r_8 : T_c \rightarrow c[(0,0,1)]$,
and $F(a, b, c) = \mu(a) + \mu(b) + (-1)\mu(b) + (-1)\mu(c)$.

5. Closure Properties

Closure properties are often handy in proving theoretical properties of grammars and languages as well as in constructing new and complex languages from existing languages. Therefore, here by using the standard proof, we investigate the closure properties that can be owned by multiset controlled grammars.

The families of languages generated by multiset controlled regular, linear and context-free grammars with linear counter (F) functions are denoted by $m\mathbf{REG}_l, m\mathbf{LIN}_l, m\mathbf{CF}_l$.

Lemma 1 (union) The families $m\mathbf{REG}_l$, $m\mathbf{LIN}_l$ and $m\mathbf{CF}_l$ are closed under union operation.

Proof: Let L_1 and L_2 be two languages in $\mathbf{X}$ with $\mathbf{X} \in \{m\mathbf{REG}, mLIN, mCF\}$ generated by multiset controlled grammars $G_1 = (N_1, T, S_1, P_1, \oplus_1, F_1)$ and $G_2 = (N_2, T, S_2, P_2, \oplus_2, F_2)$, respectively, where F_1 and F_2 are linear functions. Without loss of generality, we assume that $N_1 \cap N_2 = \emptyset$, and set $N = N_1 \cup N_2 \cup \{S\}$ where S is a new nonterminal symbol. Then, we define the grammars $G = (N, T, S, P, \oplus, F)$ where $P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$ and $F = F_1 + F_2$. Thus, it is not difficult to notice that:

$$L(G, \alpha, *) = L(G_1, \alpha, *) \cup L(G_2, \alpha, *).$$

Lemma 2 (Kleene-star) The family of $m\mathbf{REG}$ and $m\mathbf{CF}$ are closed under Kleene-star operation.

Proof $(m\mathbf{REG})$: For a given multiset controlled regular language L, let $G = (N, T, P, S, \oplus, F)$ be a multiset controlled regular grammar with $L = L(G)$. Then, it is not difficult to notice that the language L^* is generated by multiset controlled regular grammar

$$G' = \{N \cup \{S'\}, T, P \cup \{S' \to \lambda, S' \to S\} \cup \{A \to wS : A \to w \in P, w \in T^*\}, S', \oplus, F\}$$

where S' is a new nonterminal symbol.

Proof $(m\mathbf{CF})$: Let a language L is generated by multiset controlled context-free grammar $G = (N, T, P, S, \oplus, F)$. Then, it is easy to see that the language L^* is generated by mCF grammar such $G' = \{N \cup \{S'\}, T, P \cup \{S' \to SS' | \lambda\}, S', \oplus, F\}$ where S' is a new nonterminal symbol.

Lemma 3 (homomorphism) The families $m\mathbf{REG}$, $m\mathbf{LIN}$ and $m\mathbf{CF}$ are closed under homomorphism.

Proof: Let $L \in \mathbf{X}$, $\mathbf{X} \in \{m\mathbf{REG}, m\mathbf{LIN}, m\mathbf{CF}\}$ be a language generated by a multiset controlled grammar $G = (N, T, P, S, \oplus, F)$ and let $h: T^* \to {T_1}^*$ be a homomorphism. Then, there is a multiset controlled grammar $G' = (N, T_1, P_1, S_1, \oplus, F')$ such that $L(G') = h(L)$.

1. ***regular:*** for every production in the form of $r: A \to wX[\omega]$ in P, we construct the production $h(r): A \to h(w)X[\omega']$ in P_1 where $w \in T^*$, $X \in N \cup \{\lambda\}$ and $\omega \in T^{\oplus}, \omega' \in T_1^{\oplus}$;
2. ***linear:*** for every production in the form of $r: A \to w_1Xw_2[\omega]$ in P, we construct the production $h(r): A \to h(w_1)Xh(w_2)[\omega']$, where $w_1, w_2 \in T^*$, $X \in N \cup \{\lambda\}$ and $\omega \in T^{\oplus}, \omega' \in T_1^{\oplus}$;
3. ***context-free:*** for every production in the form of $r: A \to w_1X_1w_2X_2 \dots w_kX_kw_{k+1}[\omega]$, $k \geq 0$, we construct the production $h(r): A \to h(w_1)X_1h(w_2)X_2 \dots h(w_k)X_kh(w_{k+1})[\omega]$, $k \geq 0$ where $w_i, \in T^*, 1 \leq i \leq k+1$, $X_i \in N \cup \{\lambda\}, 1 \leq i \leq k$ and $\omega \in T^{\oplus}, \omega' \in T_1^{\oplus}$.

We define ω' in the above productions as $\omega' = 0$ if $|h(w)| = 0$, $\omega' = \omega$ if $|h(w)| = 1$, and $\omega' = \omega/|h(w)|$ if $|h(w)| > 1$. Then F' has the same cooficient for each symbol $a' = h(a) \in T_1$ as $a \in T$. In every successful derivation in G generating the string $w \in T^*$, we replace $r \in P$ with $h(r) \in P_1$ in the corresponding derivation and obtain $h(w) \in {T_1}^*$. Thus $h(L) = L(G')$.

Lemma 4 (mirror image) The families $m\mathbf{REG}$, $m\mathbf{LIN}$ and $m\mathbf{CF}$ are closed under mirror image operation.

Proof: Let L be a language generated by a multiset controlled regular grammar (linear grammar, context-free in Chomsky normal form grammar) $G = (N, T, P, S, \oplus, F)$, i.e., $L = L(G)$. Then, we define a multiset controlled regular grammar (linear grammar, context-free in Chomsky normal form grammar) $G' = (N, T, S, P', \oplus, F)$ such that $L(G') = L(G)^R$ by performing reverse operation on production rules of the grammar G. It is clear that:

1. $L \in m\mathbf{REG}$: for each production rule of the form $A \to wX[\omega]$ in P, we define the production $A \to Xw^R[\omega]$ where $w \in T^*$, $X \in N \cup \{\lambda\}$ and $\omega \in T^{\oplus}$;

2. $L \in m\mathbf{LIN}$: for each production rule of the form $A \to w_1Xw_2[\omega]$ in P, we define the production $A \to w_2^RXw_1^R[\omega]$ where $w_1, w_2 \in T^*$, $X \in N \cup \{\lambda\}$ and $\omega \in T^{\oplus}$;
3. $L \in m\mathbf{CF}$: for every production rule of the forms $A \to XY[\omega]$ and $A \to a[\omega]$ with $X, Y \in N, a \in T$ and $\omega \in T^{\oplus}$, we define the productions $A \to YX[\omega]$ and $A \to a[\omega]$.

Then, it is not difficult to see that $L(G') = L^R$.

Theorem 2 $m\mathbf{REG}$ and $m\mathbf{CF}$ are closed under union, Kleene-star, homomorphism and mirror image operations.

Theorem 3 $m\mathbf{LIN}$ is closed under union, homomorphism and mirror image operations.

6. Conclusion

In a nutshell, we have reviewed back the definition and computational powers of multiset controlled grammars defined in [4] where in addition we have investigated the closure properties of multiset controlled grammars. However, there are still vast questions about other closure properties, decidability problems and etc to be answered.

Acknowledgements

This research has been supported by the grants RIGS16-368-0532 and FRGS13-074-0315 of Ministry of Education, Malaysia through International Islamic University Malaysia.

References

[1] Dassow J, Paun G. Regulated Rewriting in Formal Language Theory. Springer Berlin Heidelberg: New York, 1989.
[2] Meduna A, Zemek P. Regulated Grammars and Automata. Springer Berlin Heidelberg: New York, 2014.
[3] Abraham A. Some Questions of Phrase Structure Grammars. *Computational Linguistics*. 1965; 4: 61-70.
[4] Ashaari S, Turaev S, M Tamrin MI, Okhunov A, Zhukabayeva T. Multiset Controlled Grammars. *Journal of Theoretical and Applied Information Technology*. 2017.
[5] Ashaari S, Turaev S, Okhunov A. Structurally and Arithmetically Controlled Grammars. *International Journal on Perceptive and Cognitive Computing*. 2016; 2(2): 25-35.
[6] Lobo D, Vico FJ, Dassow J. Graph Grammars with String-Regulated Rewriting. *Theoretical Computer Science*. 2011; 412(43): 6101-6111.
[7] Stiebe R. On Grammars Controlled by Parikh Vectors. In: Language Alive, LNCS 7300. Springer Berlin Heidelberg: New York. 2012: 246-264.
[8] Dassow J, Turaev S. Petri Net Controlled Grammars with a Bounded Number of Additional Places. *Journal of Acta Cybernetica*. 2010; 19: 609-634.
[9] Dassow J, Turaev S. k-Petri Net Controlled Grammars. In: Language and Automata Theory and Applications, LATA 2008, LNCS 5196, Springer Berlin Heidelberg: New York. 2008: 209-220.
[10] Dassow J, Turaev S. *Arbitrary Petri Net Controlled Grammars*. Proceedings of 2nd International Workshop on Non-Classical Formal Languages in Linguistics. Tarragona, Spain. 2008: 27-40.
[11] Stiebe R, Turaev S. Capacity Bounded Grammars and Petri Nets. In: J. Dassow, G. Pighizzini, B. Truthe (Eds.), 11th International Workshop on Descriptional Complexity of Formal Systems (DCFS 2009) EPTCS 3. 2009: 193-203.
[12] Castano J.M. Global Index Grammars and Descriptive Power. *Journal of Logic, Language and Information*. 2004; 13(4): 403-419.
[13] Kudlek M, Martin C, Paun G. Toward a Formal Macroset Theory. In: Calude, C.S Paun, G, Rozenberg, G., Salomaa, A. (Eds.), Multiset Processing, WMC 2000, LNCS 2235, Springer Berlin Heidelberg: New York. 2001: 123-133.
[14] Salomaa A. Formal languages. Academic Press: New York. 1973.
[15] Sipser M. Introduction to the Theory of Computation. 3rd edn. Cengage Learning: United States of America. 2013.
[16] Martin-Vide C. Formal Languages for Linguists: Classical and Nonclassical Models. Oxford Handbook of Computational Linguistics. Oxford University Press: Oxford. 2001.