**Bachelorarbeit**

# Weighted Multiple Context-Free Grammars over Strong Bimonoids

**Zhiang Liu**

14. November 2016

Technische Universität Dresden
Fakultät Informatik

Erstgutachter: Prof. Dr.-Ing. habil. Heiko Vogler

Zweitgutachter: Dr. rer. nat. Daniel Borchmann

Betreuer: Dipl.-Inf. Tobias Denkinger

# Aufgabenstellung für die Bachelorarbeit

**„Normal forms for weighted multiple context-free grammars“**

Technische Universität Dresden
Fakultät Informatik

| | |
|---|---|
| Student: | Liu, Zhiang |
| Geburtsdatum: | 25. April 1991 |
| Matrikelnummer: | 3846132 |
| Studiengang: | Bachelor Informatik (2009) |
| Studienleistung: | Bachelorarbeit (12 LP, ca. 360 Arbeitsstunden) |
| Beginn am: | 22. August 2016 |
| Einzureichen am: | 14. November 2016 |
| Verantw. Hochschullehrer: | Prof. Dr.-Ing. habil. Heiko Vogler |
| 1. Gutachter: | Prof. Dr.-Ing. habil. Heiko Vogler |
| 2. Gutachter: | Dr. rer. nat Daniel Borchmann |
| Betreuer: | Dipl.-Inf. Tobias Denkinger |

**Gewichtete Grammatiken und Automaten** Grammatiken und Automaten können mit einer Gewichtungsfunktion versehen werden, die jeder Produktion bzw. Transition einen Wert aus einer Gewichtsalgebra zuordnen. Diese Gewichtungsfunktion erzeugt eine Dekoration der Ableitungsbäume bzw. Läufe durch deren Auswertung (in der Gewichtsalgebra) jedem Ableitungsbaum bzw. Lauf ein Gewicht zugeordnet wird. Aus den Gewichten aller Ableitungsbäume bzw. Läufe eines bestimmten Wortes wird dann ein Gewicht für dieses Wort berechnet. Als Gewichtsalgebren werden u.A. Halbringe [DKV09; SS78; Goo99], Verbände [DV12], oder Valuierungsmonoide [DM11; DV14] verwendet.

**Multiple context-free grammars** Natürliche Sprachen weisen Merkmale auf, die von kontextfreien Grammatiken nicht darstellbar sind, z.B. kann ein Teilsatz eine Lücke haben, in die vom Kontext abhängiger Inhalt eingefügt wird. Um allerdings die hohe Parsingkomplexität kontextsensitiver Grammatiken (nämlich PSPACE-complete) zu vermeiden, betrachtet man Formalismen, die diese Lücken zwar darstellen können, aber dennoch polynomiell parsbar sind. Man fasst solche Formalismen unter dem Begriff *mildly context-sensitive formalisms* zusammen. Dazu gehören z.B. head grammars, tree adjoining grammars, combinatory categorial grammars, linear indexed grammars, linear context-free rewriting systems, und minimalist grammars. *Multiple context-free grammars* (kurz: MCFG) wurden von Pollard [Pol84] im Kontext natürlicher Sprachen eingeführt [siehe auch Sek+91]. Es hat sich herausgestellt, dass alle oben genannten (und noch einige weitere) Formalismen bzw. deren Frontsprachen eine kleinere oder die gleiche Sprachklasse wie MCFG erzeugen [Sek+91; VWJ86; WJ88; Vij87; Mic01a; Mic01b]. MCFG haben daher besondere Bedeutung für die Verarbeitung natürlicher Sprache [Eva11].

**Normalformen** Normalformen sind syntaktische Einschränkungen eines Formalismus, die aber keine Einschränkung der erzeugten Sprachklasse zur Folge haben; sie werden u.A. genutzt um die Effizienz von Algorithmen, die mit dem entsprechenden Formalismus arbeiten, zu steigern. MCFG lassen verschiedene Normalformen zu, z.B. non-deleting normal form [Sek+91, Lemma 2.2: (f3)], $\varepsilon$-free normal form [Sek+91, Lemma 2.2: (N3) und (N4)], terminal separated normal form [Sek+91, Lemma 2.2: (N1), (N2), und (N5)], monotone normal form (in der Definition von Kracht [Kra03, Definition 5.4.3] sowie Kuhlmann [Kuh07, Abschnitt 6.2.3] und Kuhlmann [Kuh13, Property 2]), und lexicalized normal form [Kuh07, Definition 618]. Siehe Kuhlmann [Kuh13, Abschnitt 5.1] für einen Überblick.

**Aufgaben** Der Student soll MCFG gewichtet mit starken Bimonoiden formal definieren. Für die starken Bimonoide ist eine Liste konkreter für die Verarbeitung natürlicher Sprache relevanter Beispiele anzugeben. Es sollen gewichtete Versionen der oben genannten Normalformen definiert und deren Universalität unter Angabe der nötigen Einschränkungen an die Gewichtsalgebra durch einen konstruktiven Beweis gezeigt werden. Die Gewichtsalgebra soll dabei jeweils so wenig wie möglich eingeschränkt werden. Desweiteren soll für jede Normalformkonstruktion ein Algorithmus angegeben werden und dessen partielle Korrektheit gezeigt werden. Die Beweise der Terminierung dieser Algorithmen sowie die Betrachtung weiterer (d.h. oben nicht genannter) Normalformen ist wünschenswert, aber in Rahmen der Arbeit optional.

**Form** Die Arbeit muss den üblichen Standards wie folgt genügen. Die Arbeit muss in sich abgeschlossen sein und alle nötigen Definitionen und Referenzen enthalten. Die Urheberschaft von Inhalten – auch die eigene – muss klar erkennbar sein. Fremde Inhalte, z.B. Algorithmen, Konstruktionen, Definitionen, Ideen, etc., müssen durch genaue Verweise auf die entsprechende Literatur kenntlich gemacht werden. Lange wörtliche Zitate sollen vermieden werden. Gegebenenfalls muss erläutert werden, inwieweit und zu welchem Zweck fremde Inhalte modifiziert wurden. Die Struktur der Arbeit muss klar erkenntlich sein, und der Leser soll gut durch die Arbeit geführt werden. Die Darstellung aller Begriffe und Verfahren soll mathematisch formal fundiert sein. Für jeden wichtigen Begriff sollen Erläuterungen und Beispiele angegeben werden, ebenso für die Abläufe der beschriebenen Verfahren. Wo es angemessen ist, sollen Illustrationen die Darstellung vervollständigen. Bei Diagrammen, die Phänomene von Experimenten beschreiben, muss deutlich erläutert werden, welche Werte auf den einzelnen Achsen aufgetragen sind, und beschrieben werden, welche Abhängigkeit unter den Werten der verschiedenen Achsen dargestellt ist. Schließlich sollen alle Lemmata und Sätze möglichst lückenlos bewiesen werden. Die Beweise sollen leicht nachvollziehbar dokumentiert sein.

**Ablauf** Zusätzlich zu den Regelungen der Prüfungsordnung gelten folgende Absprachen zum Ablauf der Arbeit: Student und Betreuer treffen sich regelmäßig. Zu den Treffen informiert der Student den Betreuer über den aktuellen Stand der Arbeit. Der Betreuer beantwortet eventuelle Fragen des Studenten und gibt Rückmeldung zum Fortschritt und zur Qualität des aktuellen Standes der Arbeit. Im Laufe der Bearbeitungszeit, idealerweise etwa zu deren Hälfte, hält der Student einen Statusvortrag zum Fortschritt seiner Arbeit.

Dresden, 2016-08-04

| | |
|---|---|
| Unterschrift von Heiko Vogler | Unterschrift von Liu, Zhiang |

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig und nur unter Zuhilfenahme der angegebenen Literatur verfasst habe.

Dresden 14.11.2016

Unterschrift

# Contents

# 1 Introduction

## 1.1 Intuition of the thesis

The *multiple context-free grammars* (shorted: MCFG) are introduced by Seki et.al. [6, Definition 2.2] for the processing of natural languages as a replacer of context-free grammars (shorted: CFG), which are less powerful. MCFGs extend CFGs by equipping a function for each production, which can get information from non-terminal symbols on the right side, and each non-terminal symbol yields fixed-size tuples of strings, where the size is determined by the function. Each component of the tuple generated by a non-terminal symbol can be used only once in the function within a production.
Syntactic restrictions of MCFG are needed for the developing of algorithms, and some of which are normal forms of MCFG that can be effectively constructed.
Weighted grammars are used widely in many cases that requires an evaluation of the words they derive, like search engines that determine which part in a given keyword is most interesting for users. A strong bimonoid $(\mathcal{A}, +, \cdot, 0, 1)$ does not require commutativity of the operation $\cdot$, which leads to a wider potential application, but notice that some syntactic restrictions of MCFG may also restrict the weight algebra.

## 1.2 Outline of the thesis

In chapter 2 we will fix some important notations that will be used throughout this thesis. Then in chapter 3 we will use the composition representation (introduced by Denkinger [1, Section 2.1]) to recall MCFG, and its syntactic restrictions of the unweighted case, which are non-deleting form [6, Lemma 2.2 f3], $\epsilon$-free form [6, Lemma 2.2 N3+N4], terminal separated form [6, Lemma 2.2 N5], monotone form [4, Definition 5.43, Page 442], strongly monotone form [5, Section 6.2.3], and lexicalized form [5, Definition 618]. In chapter 4 we will list some of the most used strong bimonoids, and recall weighted MCFG, and fix the constructions of syntactic restrictions, which are extended from non-weighted case, where we also fix the restriction they impose on to the weight algebra . In the last chapter we will briefly review the ideas of Proofs.

# 2 Preliminaries

We use $\mathbb{N}$ to denote the set of all natural numbers including 0. The set $\mathbb{N} \setminus \{0\}$ is denoted by $\mathbb{N}_+$. The set $\{k : k \in \mathbb{N}_+, k \leq x\}$ for some $x \in \mathbb{N}_+$ is denoted by $[x]$. The *empty word* is written as $\epsilon$. An *alphabet* is a finite set. We write $A \subseteq_{fin} B$, if $A$ is a finite subset of $B$. We use $|w|$ to refer to the length of the string $w$ over some alphabet, and $w\langle i \rangle$ with $i \in [|w|]$ refers to the $i$-th symbol of $w$. We fix a set of *variables* $X_{(s_1 \ldots s_k, s)} = \{x_i^j : i \in [k], j \in [s_i]\}$ for each $k \in \mathbb{N}, s_1, \ldots, s_k, s \in \mathbb{N}_+$.

**Definition 2.1** (permutations)**.** Let $k \in \mathbb{N}_+$. A *permutation* over $[k]$ is a bijection $\pi : [k] \to [k]$. We say $\pi$ is the identity on $[k]$, if $\pi(i) = i$ for each $i \in [k]$.

**Definition 2.2** (sorted sets)**.** Let $S$ be a countable set. An $S$*-sorted* set is a tuple $(A, \mathit{sort})$, where $A$ is a set and *sort* is a function from $A$ to $S$. For each $s \in S$, we abbreviate the set $\{a \in A : \mathit{sort}(a) = s\}$ by $A_s$.

**Definition 2.3** (trees)**.** Let $\Sigma$ be an $(S^* \times S)$-sorted set. The set of *trees over* $\Sigma$, denoted by $T_\Sigma$, is the smallest $S$-sorted set $T$, such that for every $k \in \mathbb{N}, s, s_1, \ldots, s_k \in S, t_1 \in T_{s_1}, \ldots, t_k \in T_{s_k}$, and $\sigma \in \Sigma_{(s_1 \ldots s_k, s)}$, we have that $\sigma(t_1, \ldots, t_k) \in T_s$.

**Definition 2.4** (characterization of trees)**.** Let $\Sigma$ be an alphabet, $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$. The *height* and *position* of trees, denoted by *height* and *pos* respectively, are

$$\mathit{height}(\xi) = 1 + \max\{\mathit{height}(\xi_i) : i \in [k]\}$$
$$\mathit{pos}(\xi) = \{\epsilon\} \cup \{iv : i \in [k], v \in \mathit{pos}(\xi_i)\}$$

We denote the label at $w$ as $\xi(w)$.

**Definition 2.5** (subtrees)**.** Let $\Sigma$ be an alphabet, $t = \sigma(t_1, ..., t_k) \in T_\Sigma$. For each $w \in pos(t)$, the *subtree* with root at the position $w$ is $t|_w$, such that for each $u \in pos(t|_w)$, there is $t|_w(u) = t(wu)$.

# 3 Multiple context-free grammars

## 3.1 MCFG

In order to describe properties of *weighted multiple context-free grammars*, we recall definitions from Denkinger [1, Section 2.1]:

**Definition 3.1** (composition representations)**.** Let $\Sigma$ be an alphabet, and $u_1, ..., u_s \in (\Sigma \cup X_{(s_1...s_k,s)})^*$, then the string $[u_1, ..., u_s]_{(s_1...s_k,s)}$ is a *composition representation*, if the variable $x_i^j \in X_{(s_1...s_k,s)}$ occurs in $u_1...u_s$ at most once, for each $i \in [k], j \in [s_i]$. We fix the set of all *composition representations* over $\Sigma$ as $R_\Sigma$. We can conceive $R_\Sigma$ as an $\mathbb{N}^* \times \mathbb{N}$-sorted set, where for each $r = [u_1, ..., u_s]_{(s_1...s_k,s)} \in R_\Sigma$, we set $sort(r) = (s_1...s_k, s)$.

**Definition 3.2** (composition functions)**.** For $r = [u_1, ..., u_s]_{(s_1...s_k,s)} \in R_\Sigma$, the *composition function with respect to* $r$ is:

$$f_r\colon (\Sigma^*)^{s_1} \times ... \times (\Sigma^*)^{s_k} \to (\Sigma^*)^s$$
$$f_r((w_1^1, ..., w_1^{s_1}), ..., (w_k^1, ..., w_k^{s_k})) = (u_1', ..., u_s')$$

for all $w_1^1, ..., w_1^{s_1}, ..., w_k^1, ..., w_k^{s_k} \in \Sigma^*$, where for each $\kappa \in [s]$, $u_\kappa'$ is obtained from $u_\kappa$ by replacing every occurrence of $x_i^j \in X_{(s_1...s_k,s)}$ by $w_i^j$ for all $i \in [k], j \in [s_i]$.

In the following we will not distinguish between $r$ and $f_r$. We also write $[u_1, ..., u_s]$ instead of $[u_1, ..., u_s]_{(s_1...s_k,s)}$ when referring to *composition functions*, and write $X_f$ instead of $X_{(s_1...s_k,s)}$ for $f = [u_1, ..., u_s]_{(s_1...s_k,s)}$.

**Example 3.1** (composition functions)**.** Let $\Sigma = \{a, b\}, X = \{x_1^1, x_1^2, x_2^1, x_2^2\}$ and $r = [ax_1^1b, bx_2^2b]_{(22,2)}$, then for the tuple $T := ((ab, ba), (bb, aa))$ we have $f_r(T) = (aabb, baab)$.

We recall the definition of multiple context-free grammars from Seki, Matsumura, Fujii and Kasami [6, Definition 2.2]:

**Definition 3.3** (multiple context-free grammars). A *multiple context-free grammar* (short: MCFG) is a tuple $G = (N, \Sigma, P, S)$ where

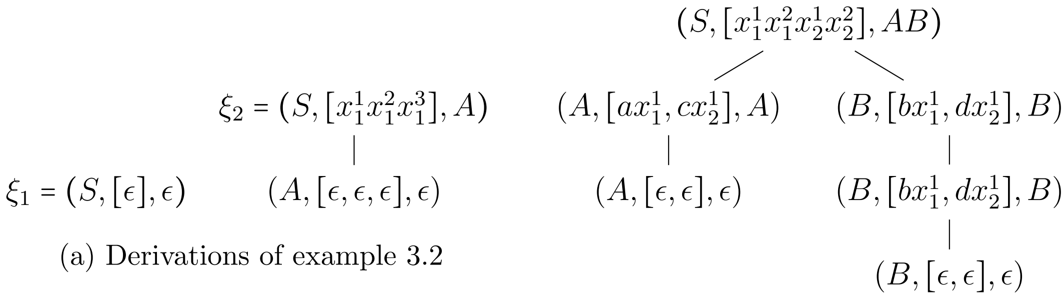
- $N$ is a finite $\mathbb{N}_+$-sorted set (non-terminal symbols),
- $\Sigma$ is an alphabet (terminal symbols),
- $P \subseteq_{fin} \bigcup_{s_1,\ldots,s_k,s \in \mathbb{N}_+} N_s \times (R_\Sigma)_{(s_1 \ldots s_k, s)} \times (N_{s_1} \cdot \ldots \cdot N_{s_k})$.
- $S \in N_1$.

Let an MCFG $G = (N, \Sigma, P, S)$. We conceive $P$ as an $(N^* \times N)$-sorted set with $sort(p) = (A_1 \ldots A_k, A)$ for each $p = (A, f, A_1 \ldots A_k) \in P$. We define a function $fan: P \to \mathbb{N}_+$, where for each production $p = (A, f, A_1 \ldots A_k) \in P, fan(p) = sort(A)$, called *fan-out* of $p$ and $p$ with $k = 0$ is called a *terminating* production. We write $f_p$ to indicate the composition function of $p$. We also call $k$ the rank of $p$. The *fan-out* of $G$ is $max\{fan(p) : p \in P\}$, and an MCFG with fan-out at most $m$ is called an $m$-MCFG.

**Definition 3.4** (derivations of MCFG). Let $G = (N, \Sigma, P, S)$ be an MCFG. Then the set of *derivations of* $G$, denoted by $D_G$, is $(T_P)_S$.
We define the function $yield_G : T_P \to (\Sigma^*)^*$. For each $\xi = p(\xi_1, \ldots, \xi_k) \in T_P$ with $p = (A, f, A_1 \ldots A_k) \in P$, We have:

$$yield_G(\xi) = f(yield_G(\xi_1), \ldots, yield_G(\xi_k))$$

A string $w \in \Sigma$ is called *derivable in* $G$, if there exists a *derivation* $\xi$, such that $yield_G(\xi) = w$. We abbreviate $yield_G$ to *yield*, and denote the set of all strings that is derivable in $G$ by $L(G)$, $D_G(A) := (T_P)_A$ for each $A \in N$, and $D_G(w) := \{\xi : \xi \in D_G, yield(\xi) = w\}$ as *derivations of* $w$ *by* $G$. For each $A, B \in N$ we say $B$ is reachable from $A$, if there is a $\xi_B \in D_G(B)$ and $\xi_A \in D_G(A)$, such that $\xi_B$ is a subtree of $\xi_A$. For each $A \in N$ and $w \in yield(D_G(A))$, we fix $w\langle i \rangle$ with $i \in [sort(A)]$ to refer to the $i$-th component of $w$.




(a) Derivations of example 3.2

(b) Derivations of example 3.3

Figure 1: Derivation Examples

**Example 3.2** (MCFG)**.** For the formal language $L = \{a^n b^n c^n : n \in \mathbb{N}\}$, we have $G = (N, \Sigma, P, S)$, with $P = \{p_1, p_2, p_3, p_4\}, N = \{S, A\}, \Sigma = \{a, b, c\}$ and:

$$p_1 = (S, [\epsilon], \epsilon), \qquad p_2 = (S, [x_1^1 x_1^2 x_1^3], A),$$
$$p_3 = (A, [ax_1^1, bx_1^2, cx_1^3], A), \qquad p_4 = (A, [\epsilon, \epsilon, \epsilon], \epsilon),$$

that $L(G) = L$. The two derivations of $w_2 = \epsilon$ are shown in Figure 1a.

**Example 3.3** (MCFG)**.** For the formal language $L = \{a^n b^m c^n d^m : m, n \in \mathbb{N}\}$, there exists an MCFG $G = (N, \Sigma, P, S)$, such that $L(G) = L$. Consider $P = \{p_1, p_2, p_3, p_4, p_5\}, N = \{S, A, B\}, \Sigma = \{a, b, c, d\}$ and:

$$p_1 = (S, [x_1^1 x_2^1 x_1^2 x_2^2], AB), \qquad p_4 = (B, [bx_1^1, dx_1^2], B),$$
$$p_2 = (A, [ax_1^1, cx_1^2], A), \qquad p_5 = (B, [\epsilon, \epsilon], \epsilon),$$
$$p_3 = (A, [\epsilon, \epsilon], \epsilon),$$

The only derivation of $w_1 = abbcdd$ is shown in Figure 1b.

## 3.2 Normal forms

We recall some syntactical restrictions of MCFGs from the literature.

**Definition 3.5** (Syntactical restrictions to composition functions)**.** Let $\Sigma$ be an alphabet, and $r = [u_1, ..., u_s]_{(s_1 ... s_k, s)} \in R_\Sigma$. We call $r$

- *non-deleting* if all elements of $X_{(s_1 ... s_k, s)}$ occur in $u_1 ... u_s$;

- $\epsilon$-*free* if $\epsilon \notin \{u_1, ..., u_s\}$;
- *terminal separated* if $r$ is in a terminating rule, we have $s = 1$ with $u_1 \in (\Sigma \cup \{\epsilon\})$, and if $r$ is in a non-terminating rule, we have $u_\kappa \in X^*_{(s_1...s_k,s)}$ for each $\kappa \in [s]$;
- *monotone* [4, Definition 5.43, Page 442] if for each fixed $i \in [k]$, all $x_i^j$ with $j \in [s_i]$ occur in the increasing order of $j$ from left to right in $u_1...u_k$;
- *strongly monotone* [5, Section 6.2.3] if $r$ is monotone, and all $x_i^1, i \in [k]$ occur in the increasing order of $i$ from left to right in $u_1...u_k$;
- *lexicalized* [5, Definition 618] if $u_1...u_s$ contains exactly one terminal symbol.

**Definition 3.6** (Syntactical restrictions to MCFG)**.** Let $G = (N, \Sigma, P, S)$ be an MCFG, then we call $G$

- $\epsilon$-*free* if the composition function of all productions are $\epsilon$-*free*, or $(S, [\epsilon], \epsilon)$ is the only production whose composition function is not $\epsilon$-free, and if $(S, [\epsilon], \epsilon) \in P$, then $S$ does not occur on the right side of any productions;
- *non-deleting, terminal separated, monotone, strongly monotone, or lexicalized* if the composition functions of all productions are *non-deleting, terminal separated, monotone, strongly monotone, or lexicalized*, respectively.

**Definition 3.7** (tuple restriction)**.** Let $G = (N, \Sigma, P, S)$ be an MCFG, $yield\colon T_P \to (\Sigma^*)^*$, and $\Psi = \{s_1, ..., s_n\} \subseteq \mathbb{N}_+$ with $s_1, ..., s_n \in \mathbb{N}_+$ in increasing order. The *tuple restriction* with respect to $\Psi$ is $[\![\Psi]\!]\colon (\Sigma^*)^* \to (\Sigma^*)^*$, where for a given $w = (w_1, ..., w_k) \in (\Sigma^*)^*$ with $w_1, ..., w_k \in \Sigma^*$ and $k \geq s_n$, we have $[\![\Psi]\!](w) = (w_{s_1}, ..., w_{s_n})$.
We fix $yield_\Psi = yield; [\![\Psi]\!]$, and $n[M] = |\{i\colon i \in M, n > i\}|$ for each $n \in \mathbb{N}_+, M \subset \mathbb{N}_+$.

**Definition 3.8** (restricted composition function)**.** Let $\Sigma$ be some alphabet, $f = [u_1, ..., u_s]_{(s_1...s_k,s)}$ be a composition function, $\Psi \subseteq [s], \Psi_1 \subseteq [s_1], ..., \Psi_k \subseteq [s_k]$. The *restricted composition function* with respect to $\Psi, \Psi_1, ..., \Psi_k$ and $f$ is

$$f_{\Psi,\Psi_1,...,\Psi_k} \colon (\Sigma^*)^{|\Psi_1|} \times ... \times (\Sigma^*)^{|\Psi_k|} \to (\Sigma^*)^{|\Psi|},$$

The function $f_{\Psi,\Psi_1,...,\Psi_k}$ is obtained from $f$: we replace each $x_i^j \in X_{(s_1...s_k,s)}$ with $j \notin \Psi_i$ by $\epsilon$, and delete each component $u_l$ with $l \notin \Psi$, and replace each remaining $x_i^j \in X_{(s_1...s_k,s)}$ by $x_i^{j-j[[s_i]\setminus\Psi_i]}$.

**Lemma 3.1** (restricted composition function)**.** Let $\Sigma$ be some alphabet, $f = [u_1, ..., u_s]_{(s_1...s_k,s)}$ be a composition function, and $\Psi \subseteq [s], \Psi_1 \subseteq [s_1], ..., \Psi_k \subseteq [s_k]$. Let $v_1 \in (\Sigma^*)^{s_1}, ..., v_k \in (\Sigma^*)^{s_k}$, such that for each $x_i^j \in X_{(s_1...s_k,s)}$, if $x_i^j$ occurs in $[\![\Psi]\!](u_1, ..., u_s)$, and $j \notin \Psi_i$, then $v_i\langle j\rangle = \epsilon$. We have

$$f_{\Psi,\Psi_1,...,\Psi_k}([\![\Psi_1]\!](v_1), ..., [\![\Psi_k]\!](v_k)) = [\![\Psi]\!](f(v_1, ..., v_k)).$$

*Proof.* Let $(w_1, ..., w_l) = [\![\Psi]\!](f(v_1, ..., v_k))$, and $\Psi = \{p_1, ..., p_l\}$ with $p_1 < p_2 < ... < p_l$. Then we have $w_i = [u_{p_i}](v_1, ..., v_k)$. Let $u'_{p_i}$ be constructed from $u_{p_i}$ by Definition 3.8 for each $i \in [l]$. There is $[u_{p_i}](v_1, ..., v_k) = [u'_{p_i}]([\![\Psi_1]\!](v_1), ..., [\![\Psi_k]\!](v_k))$, since each component deleted by $\Psi_1, ..., \Psi_k$ but used by some variable in $u_{p_i}$ is $\epsilon$, and this this variable is replaced by $\epsilon$ in $u'_{p_i}$, and each component preserved by $\Psi_1, ..., \Psi_k$ is used in the same position of $u'_{p_i}$ and $u_{p_i}$ according to the variable renaming in Definition 3.8. Thus

$$\begin{aligned}
[\![\Psi]\!](f(v_1, ..., v_k)) &= [\![\Psi]\!]([u_1, ..., u_s](v_1, ..., v_k)) \\
&= [u_{p_1}, ..., u_{p_l}](v_1, ..., v_k) \\
&= [u'_{p_1}, ..., u'_{p_l}]([\![\Psi_1]\!](v_1), ..., [\![\Psi_k]\!](v_k)) && \text{(by the text above)} \\
&= f_{\Psi,\Psi_1,...,\Psi_k}([\![\Psi_1]\!](v_1), ..., [\![\Psi_k]\!](v_k)) && \text{(by Definition 3.8)}
\end{aligned}$$

□

**Theorem 3.1** (Non-deleting)**.** [6, Lemma 2.2 f3] For every $m$-MCFG $G$ there is a non-deleting $m$-MCFG $G'$, such that $L(G) = L(G')$.

*Construction.* Let $G = (N, \Sigma, P, S)$. We recall the construction by Seki, Matsumura, Fujii, and Kasami [6, Lemma 2.2 f3]. We construct $G' = (N', \Sigma, P', S[\{1\}])$, where

- $N' = \{A[\Psi]: A \in N, \Psi \subseteq [sort(A)]\}$, where $sort(A[\Psi]) = |\Psi|$ for each $A \in N$ and $\Psi \subseteq [sort(A)]$.
- $P' = \{(A[\Psi], f_{\Psi,\Psi_{f,1,\Psi},\dots,\Psi_{f,k,\Psi}}, A_1[\Psi_{f,1,\Psi}]\dots A_k[\Psi_{f,k,\Psi}]): (A, f, A_1\dots A_k) \in P, \Psi \subseteq [sort(A)]\}$, where for each $s_1, \dots, s_k, s \in \mathbb{N}, f \subseteq (R_\Sigma)_{(s_1\dots s_k, s)}$, and $\Psi \subseteq [s]$, we fix $\Psi_{f,i,\Psi} = \{j \in [s_i]: \exists l \in \Psi. x_i^j \text{ occurs in } u_l\}$, for each $i \in [k]$[1].

*Proof.* The function $f_{\Psi,\Psi_{f,1,\Psi},\dots,\Psi_{f,k,\Psi}}$ of each constructed production is non-deleting by the construction of $\Psi_{f,1,\Psi}, \dots, \Psi_{f,k,\Psi}$ for each $p = (A, f, A_1\dots A_k) \in P$ and $\Psi \subseteq [sort(A)]$. We fix the function $g: P' \to P$ that assigns the original production to its constructed one, and $\hat{g}: T_{P'} \to T_P$ that applies $g$ position-wise. We can conceive that $\hat{g}$ is a bijection between $(T'_P)_{A[\Psi]}$ and $(T_P)_A$ with a given $\Psi$ for each $A \in N$, and $g$ is a bijection between $P'$ with a fixed $\Psi$ and $P$, proved by Denkinger [2, Lemma 5]. Let $p = (A, [u_1, \dots, u_s], A_1\dots A_k) \in P, d = p(d_1, \dots, d_k) \in D_G(A), \Psi = \{i: i \in [s], u_i \neq \epsilon\}$, $d' = p'(d'_1, \dots, d'_k) = \hat{g}^{-1}(d)$, and $\Psi_{f,1,\Psi}, \dots, \Psi_{f,k,\Psi}$ are fixed according to the construction. Then we have the following structural induction over $[\![\Psi]\!](yield(d)) = yield(d')$:

$$
\begin{aligned}
&[\![\Psi]\!](yield(d)) \\
=&[\![\Psi]\!](yield(p(d_1, \dots, d_k))) \\
=&f_{\Psi,\Psi_{f,1,\Psi},\dots,\Psi_{f,k,\Psi}}([\![\Psi_{f,1,\Psi}]\!](yield(d_1)), \dots, [\![\Psi_{f,k,\Psi}]\!](yield(d_k))) && \text{(Lemma 3.1)} \\
=&f_{\Psi,\Psi_{f,1,\Psi},\dots,\Psi_{f,k,\Psi}}(yield(d'_1), \dots, yield(d'_k)) && \text{(induction hypothesis)} \\
=&yield(d')
\end{aligned}
$$

Thus for each derivation $d \in D_G$, and for each $d' \in D_{G'}, d' = \hat{g}^{-1}(d)$, we have

$$yield(d) = [\![\{1\}]\!](yield(d)) = yield(d')$$

Hence $G'$ is non-deleting, and $L(G) = L(G')$. □

**Example 3.4** (non-deleting form). For some MCFG $G = (N, \Sigma, P, S)$ with $\Sigma = \{a, b, c\}$,

$$p_1 = (S, [x_1^1 x_1^2 x_1^3], A), \qquad p_2 = (A, [ax_1^1, bx_1^2, cx_1^3, x_1^4], A),$$

[1] Note that each $\Psi_i$ with $i \in [k]$ is defined as the set of positions of unused variables of $A_i$ in the original paper, but of used variables here.

$$p_3 = (A, [\epsilon, \epsilon, \epsilon, b], \epsilon)$$

after the procedure from Theorem 3.1, and removal of all productions containing non-terminal symbols that are not reachable from $S[\{1\}]$, we have:

$p'_1 = (S[\{1\}], [x_1^1 x_1^2 x_1^3], A[\{1,2,3\}])$, $p'_2 = (A[\{1,2,3\}], [ax_1^1, bx_1^2, cx_1^3], A[\{1,2,3\}])$, $p'_3 = (A[\{1,2,3\}], [\epsilon, \epsilon, \epsilon], \epsilon)$

**Theorem 3.2** ($\epsilon$-free)**.** [6, Lemma 2.2 N3+N4] For every $m$-MCFG $G$ there is an $\epsilon$-free $m$-MCFG $G'$, such that $L(G) = L(G')$.

*Construction.* Let $G = (N, \Sigma, P, S)$. We use the construction by Seki et. al. [6, Lemma 2.2], and $G' = (N', \Sigma, P', S')$, where

- $N' = \{A[\Psi] : A \in N, \Psi \subseteq [sort(A)]\} \cup \{S'\}$,
- $P' = \{(S', [\epsilon], \epsilon) : \epsilon \in L(G)\} \cup \{(S', [x_1^1], S[\{1\}])\} \cup \{(A[\Psi_{f,\Psi_1,...,\Psi_k}], f_{\Psi_{f,\Psi_1,...,\Psi_k},\Psi_1,...,\Psi_k}, A_1[\Psi_1]...A_k[\Psi_k]) : (A, f, A_1...A_k) \in P, \Psi_1 \subseteq [sort(A_1)], ..., \Psi_k \subseteq [sort(A_k)]\}$, where for each $f = [u_1, ..., u_s]_{(s_1...s_k,s)}, \Psi_1 \subseteq [sort(A_1)], ..., \Psi_k \subseteq [sort(A_k)]$, we fix $\Psi_{f,\Psi_1,...,\Psi_k} \subseteq [sort(A)]$, such that for each $l \in \Psi_{f,\Psi_1,...,\Psi_k}$, $u_l$ contains at least one terminal symbol, or at least one variable from the set $\{x_i^j : x_i^j \in X_{(s_1...s_k,s)}, j \in \Psi_i\}$.

*Proof.* It is obvious that each production other than $(S', [\epsilon], \epsilon)$ is $\epsilon$-free, hence the constructed $G'$ is $\epsilon$-free. Let $P'' = P' \setminus (\{(S', [\epsilon], \epsilon)\} \cup \{(S', [x_1^1], S[\{1\}])\})$. We fix the function $g : P'' \to P$ that assigns the original production to its constructed one, and the function $\hat{g} : T_{P''} \to T_P$ applying $g$ position-wise. We can conceive that $\hat{g}$ is a bijection between $(T_{P''})_{A[\Psi_{f,\Psi_1,...,\Psi_k}]}$ and $(T_P)_A$ according to Denkinger [2, Lemma 5]. For each $p = (A, [u_1, ..., u_s], A_1...A_k) \in P, d = p(d_1, ..., d_k) \in D_G(A), d'' = p'(d''_1, ..., d''_k) = \hat{g}^{-1}(d), \Psi_1 \subseteq [sort(A_1)], ..., \Psi_k \subseteq [sort(A_k)]$, we have by structural induction over $d$:

$$\begin{aligned} & [\![\Psi_{f,\Psi_1,...,\Psi_k}]\!](yield(d)) \\ =& [\![\Psi_{f,\Psi_1,...,\Psi_k}]\!](yield(p(d_1, ..., d_k))) \\ =& f_{\Psi_{f,\Psi_1,...,\Psi_k},\Psi_1,...,\Psi_k}([\![\Psi_1]\!](yield(d_1)), ..., [\![\Psi_k]\!](yield(d_k))) && \text{(Lemma 3.1)} \end{aligned}$$

$$=f_{\Psi_{f,\Psi_1,\dots,\Psi_k},\Psi_1,\dots,\Psi_k}(yield(d''_1),\dots,yield(d''_k)) \qquad \text{(induction hypothesis)}$$
$$=yield(d'')$$

For $w \in L(G)$ with $w = \epsilon$, we have $w \in L(G')$ because of the production $(S', [\epsilon], \epsilon)$. For each $w \in L(G)$ with $w \neq \epsilon$, we use the fact that

$$yield(D_G) = [\![\{1\}]\!](yield(D_G)) = yield(D_{G''}) = yield(D_{G'}).$$

Hence $L(G) = L(G')$. □

**Example 3.5** ($\epsilon$-free form)**.** We use the Example 3.2 to show a construction of $\epsilon$-free form. For the productions:

$$p_1 = (S, [\epsilon], \epsilon), \qquad p_2 = (S, [x_1^1 x_1^2 x_1^3], A),$$
$$p_3 = (A, [a x_1^1, b x_1^2, c x_1^3], A), \qquad p_4 = (A, [\epsilon, \epsilon, \epsilon], \epsilon),$$

after the procedure from Theorem 3.2, and removal of all productions containing non-terminal symbols that are not reachable from $S'$, we have:

$$p_\epsilon = (S', [\epsilon], \epsilon), \qquad p_s = (S', [x_1^1], S[\{1\}]),$$
$$p'_2 = (S[\{1\}], [x_1^1 x_1^2 x_1^3], A[\{1,2,3\}]), \quad p'_3 = (A[\{1,2,3\}], [a x_1^1, b x_1^2, c x_1^3], A[\{1,2,3\}]),$$
$$p''_3 = (A[\{1,2,3\}], [a, b, c], A[\varnothing]), \qquad p'_4 = (A[\varnothing], [], \epsilon).$$

**Theorem 3.3** (Terminal separated)**.** [6, Lemma 2.2 N5] For every $m$-MCFG $G$ there is a terminal separated $m$-MCFG $G'$, such that $L(G) = L(G')$.

*Construction.* Let $G = (N, \Sigma, P, S)$. We construct $G' = (N', \Sigma, P', S)$, where

- $N' = \{A : A \in N\} \cup \{\boxed{a} : a \in \Sigma \cup \{\epsilon\}\}$,
- $P' = \{(A, f_{a_1,\dots,a_n}, A_1 \dots A_k \boxed{a_1} \dots \boxed{a_n}) : (A, f, A_1 \dots A_k) \in P, a_1, \dots, a_n$ are terminal symbols in $f\} \cup \{(\boxed{a}, [a], \epsilon) : a \in \Sigma\}$, where $f_{a_1,\dots,a_n}$ is obtained from $f$ by replacing $a_i$ by $x_{k+i}^1$ for each $i \in [n]$.

*Proof.* We fix a function $g : P \to P'$ that assigns to each $p \in P$ the production constructed from it, and a function $\hat{g} : T_P \to T_{P'}$ applying $g$ position-wise, where $\hat{g}(p(t_1, \dots, t_k)) = g(p)(\hat{g}(t_1), \dots, \hat{g}(t_k), (\boxed{a_1}, [a_1], \epsilon), \dots, (\boxed{a_n}, [a_n], \epsilon))$ for each tree

$p(t_1, ..., t_k) \in T_P$, where $a_1, ..., a_n$ are the terminal symbols in the composition function of $p$.

Then $g^{-1}$ is a function, since $g(p)(\hat{g}(t_1), ..., \hat{g}(t_k), (\boxed{a_1}, [a_1], \epsilon), ..., (\boxed{a_n}, [n], \epsilon))$ is unique for each $p(t_1, ..., t_k)$. Let $\xi = p(\xi_1, ..., \xi_k) \in T_P$, and

$$\xi' = \hat{g}(\xi) = g(p)(\hat{g}(\xi_1), ..., \hat{g}(\xi_k), (\boxed{a_1}, [a_1], \epsilon), ..., (\boxed{a_n}, [n], \epsilon)),$$

then we have the following structural induction for $\mathit{yield}(\xi) = \mathit{yield}(\hat{g}(\xi))$:

$$\begin{aligned} \mathit{yield}(\xi) &= f_p(\mathit{yield}(\xi_1), ..., \mathit{yield}(\xi_k)) \\ &= f_p(\mathit{yield}(\hat{g}(\xi_1)), ..., \mathit{yield}(\hat{g}(\xi_k))) && \text{(induction hypothesis)} \\ &= f_{g(p)}(\mathit{yield}(\hat{g}(\xi_1)), ..., \mathit{yield}(\hat{g}(\xi_k)), a_1, ..., a_n) && (*) \\ &= \mathit{yield}(\hat{g}(\xi)) \end{aligned}$$

The $(*)$ holds since $a_i$ in $f_p$ is replaced by $x^1_{k+i}$ in $f_{p'}$, which refers to $\boxed{a_i}$, and for the unique $t \in D_{G'}(\boxed{a_i})$ there is $\mathit{yield}(t) = a_i$, for each $i \in [n]$. Since the function $\hat{g}$ is bijective, and there is $D_{G'} = \hat{g}(D_G)$, we have $\mathit{yield}(D_G) = \mathit{yield}(\hat{g}(D_G)) = \mathit{yield}(D_{G'})$, hence $L(G) = L(G')$. □

**Example 3.6** (Terminal separated form)**.** We use the Example 3.2 to show the construction. For the productions:

$$p_1 = (S, [\epsilon], \epsilon), \qquad p_2 = (S, [x^1_1 x^2_1 x^3_1], A),$$
$$p_3 = (A, [a x^1_1, b x^2_1, c x^3_1], A), \qquad p_4 = (A, [\epsilon, \epsilon, \epsilon], \epsilon),$$

we have the terminal separated form:

$$p'_1 = (S, [x^1_1], \boxed{\epsilon}), \qquad p'_2 = (S, [x^1_1 x^2_1 x^3_1], A),$$
$$p'_3 = (A, [x^1_2 x^1_1, x^1_3 x^2_1, x^1_4 x^3_1], A \boxed{a} \boxed{b} \boxed{c}), \qquad p'_4 = (A, [x^1_1, x^1_2, x^1_3], \boxed{\epsilon} \boxed{\epsilon} \boxed{\epsilon}),$$
$$p_\epsilon = (\boxed{\epsilon}, [\epsilon], \epsilon), \qquad p_a = (\boxed{a}, [a], \epsilon)$$
$$p_b = (\boxed{b}, [b], \epsilon), \qquad p_c = (\boxed{c}, [c], \epsilon)$$

**Theorem 3.4** (Strongly monotone)**.** For every $m$-MCFG $G$ there is a strongly monotone $m$-MCFG $G'$, such that $L(G) = L(G')$.

*Construction.* Let $G = (N, \Sigma, P, S)$. We fix $G' = (N', \Sigma, P', S[\pi_0])$, where $\pi_0$ is the identity on $\{1\}$.

- $N' = \{A[\pi]: A \in N\}$, where $\pi$ is a permutation over $[sort(A)]$, and
- $P' = \{(A[\pi], ([u'_{\pi(1)}, ..., u'_{\pi(s)}]_{(s_1...s_k,s)}), A_{\gamma_{f,\pi}(1)}[\pi_{f,\gamma_{f,\pi}(1)}]...A_{\gamma_{f,\pi}(k)}[\pi_{f,\gamma_{f,\pi}(k)}]) : (A, [u_1, ..., u_s]_{(s_1...s_k,s)}, A_1...A_k) \in P, \pi$ is permutation over $[k]\}$, where for each $f = [u_1, ..., u_s]_{(s_1...s_k,s)}$ and the permutation $\pi$ over $[k]$,
  - $\pi_{f,i}$ is a permutation over $[s_i]$ for each $i \in [k]$, such that $x_i^{\pi^{-1}_{f,i}(1)}, ..., x_i^{\pi^{-1}_{f,i}(k)}$ occur in that order in $u_{\pi(1)}...u_{\pi(s)}$, and
  - the monotone form $[u''_{\pi(1)}, ..., u''_{\pi(s)}]$ is obtained from $[u_{\pi(1)}, ..., u_{\pi(s)}]$ by replacing each $x_i^j \in X_{(s_1...s_k,s)}$ by $x_i^{\pi_{f,i}(j)}$, and
  - $\gamma_{f,\pi}$ is a permutation over $[k]$, such that $x^1_{\gamma^{-1}_{f,\pi}(1)}, ..., x^1_{\gamma^{-1}_{f,\pi}(k)}$ occur in that order in $u''_{\pi(1)}...u''_{\pi(s)}$, and
  - the strongly monotone form $[u'_{\pi(1)}, ..., u'_{\pi(s)}]$ is obtained from $[u''_{\pi(1)}, ..., u''_{\pi(s)}]$ by replacing each $x_i^j \in X_{(s_1...s_k,s)}$ by $x^j_{\gamma_{f,\pi}(i)}$.

*Proof.* The permutations used in the construction build strongly monotone form for each production. Hence $G'$ is strongly monotone. Let $G' = (N', \Sigma, P', S[\pi_0])$ be the grammar constructed from $G$ with $\pi_0$ the identity on $\{1\}$. Let the function $g: P' \to P$ assign to the constructed production the original one it is constructed from, and let function $\hat{g}: T_{P'} \to T_P$, where for each tree $t' = p'(t'_1, ..., t'_k) \in T_{P'}$, and $\gamma_{f,\pi}$ for the construction of $t'$, we have $\hat{g}(t') = g(p')(\hat{g}(t'_{\gamma^{-1}_{f,\pi}(1)}), ..., \hat{g}(t'_{\gamma^{-1}_{f,\pi}(k)})) \in T_P$ with $yield(t') = yield(t)$. Then $\hat{g}$ is a bijection between $(T_{P'})_{A[\pi]}$ and $(T_P)_A$ for each $A \in N$ and $\pi \in sort(A)$, and $g$ is a bijection between $P'$ in $(T_{P'})_{A[\pi]}$ and $P$ in $(T_P)_A$. For each tree $t = p(t_1, ..., t_k) \in (T_P)_A$, and $\hat{g}^{-1}(t) = g^{-1}(p)(\hat{g}^{-1}(t_{\gamma_{f,\pi}(1)}), ..., \hat{g}^{-1}(t_{\gamma_{f,\pi}(k)})) = \hat{g}^{-1}(t) \in (T_{P'})_{A[\pi]}$, we fix the function $yield_\pi: T_P \to (\Sigma^*)^*$ with

$$yield_\pi(t) = (yield(t)\langle\pi(1)\rangle, ..., yield(t)\langle\pi(|yield(t)|)\rangle)$$

For each $i \in [|yield(t)|]$, and $\pi_1, ..., \pi_k$ for the construction for $t_1, ..., t_k$, respectively, we can conceive that

$$f_{g^{-1}(p)}(yield_{\pi_{\gamma_{f,\pi}(1)}}(t_{\gamma_{f,\pi}(1)}), ..., yield_{\pi_{\gamma_{f,\pi}(k)}}(t_{\gamma_{f,\pi}(k)}))\langle i\rangle = yield(t)\langle\pi^{-1}(i)\rangle,$$

since the $l$-th component of $f_{g^{-1}(p)}(yield_{\pi_{\gamma_{f,\pi(1)}}}(t_{\gamma_{f,\pi(1)}}), ..., yield_{\pi_{\gamma_{f,\pi(k)}}}(t_{\gamma_{f,\pi(k)}}))$ is obtained from $(\pi^{-1}(l))$-th component of $yield(t)$ for each $l \in [|yield(t)|]$, and each variable $x_i^j \in X_{(s_1 ... s_k, s)}$ in $p$ has the same position as the variable $x_{\gamma_{f,\pi}(i)}^{\pi_{f,i}(j)}$ in $g^{-1}(p)$, by the construction, where $x_{\gamma_{f,\pi}(i)}^{\pi_{f,i}(j)}$ gets the $(\pi_{f,i}(j))$-th component of $yield_{\pi_{\gamma_{f,\pi}(i)}}(t_{\gamma_{f,\pi}(i)})$, which is the $j$-th component of $yield(t_i)$ got by $x_i^j$. We have the following structural induction over $t$:

$$
\begin{aligned}
& yield_\pi(t) \\
&= (yield(t)\langle\pi(1)\rangle, ..., yield(t)\langle\pi(|yield(t)|)\rangle) \\
&= f_{g^{-1}(p)}(yield_{\pi_{\gamma_{f,\pi}(1)}}(t_{\gamma_{f,\pi}(1)}), ..., yield_{\pi_{\gamma_{f,\pi}(k)}}(t_{\gamma_{f,\pi}(k)})) && \text{(by text above)} \\
&= f_{g^{-1}(p)}(yield(\hat{g}^{-1}(t_{\gamma(1)})), ..., yield(\hat{g}^{-1}(t_{\gamma(k)}))) && \text{(induction hypothesis)} \\
&= yield(\hat{g}^{-1}(t))
\end{aligned}
$$

Hence $yield(D_G) = yield_{\pi_0}(D_G) = yield(D_{G'}), L(G) = L(G')$. □

**Example 3.7** (strongly monotone form)**.** For an MCFG with productions:

$$
\begin{aligned}
p_1 &= (S, [x_1^2 x_2^1 x_2^2 x_1^1], AB), & p_4 &= (B, [bx_1^1, dx_1^2], B), \\
p_2 &= (A, [ax_1^1, cx_1^2], A), & p_5 &= (B, [\epsilon, \epsilon], \epsilon), \\
p_3 &= (A, [\epsilon, \epsilon], \epsilon),
\end{aligned}
$$

we obtained the strongly monotone form using the construction in Theorem 3.4, and deletes all non-terminal symbols that are not reachable from $S[\pi_0]$:

$$
\begin{aligned}
p_1' &= (S[\pi_0], [x_1^1 x_2^1 x_2^2 x_1^2], A[\pi_{f_{p_1},1}]B[\pi_{f_{p_1},2}]), & p_4' &= (B[\pi_{f_{p_1},2}], [bx_1^1, dx_1^2], B[\pi_{f_{p_1},2}]), \\
p_2' &= (A[\pi_{f_{p_1},1}], [cx_1^1, ax_1^2], A[\pi_{f_{p_2},1}]), & p_5' &= (B[\pi_{f_{p_1},2}], [\epsilon, \epsilon], \epsilon), \\
p_2'' &= (A[\pi_{f_{p_2},1}], [cx_1^1, ax_1^2], A[\pi_{f_{p_2},1}]), & p_3' &= (A[\pi_{f_{p_1},1}], [\epsilon, \epsilon], \epsilon), \\
p_3'' &= (A[\pi_{f_{p_2},1}], [\epsilon, \epsilon], \epsilon),
\end{aligned}
$$

where $\pi_0, \pi_{f_{p_1},2}$ are the identities on $\{1\}, \{1, 2\}$, respectively, and $\pi_{f_{p_1},1} = \pi_{f_{p_2},1}$ are both the permutation $(1, 2)$ over $\{1, 2\}$.

**Definition 3.9** (terminal depth of trees)**.** Let $G = (N, \Sigma, P, S)$ be an MCFG with $\epsilon \notin L(G)$. The *terminal depth of trees* is $dep\colon T_P \to \mathbb{N}$. For each $t \in T_P$, we fix

$$dep(t) = \begin{cases} \infty, \text{ if for each } w \in pos(t), f_{t(w)} \text{ contains no terminal symbols} \\ \min\{|w|\colon w \in pos(t), f_{\xi(w)} \text{ contains a terminal symbol}\}, \text{ otherwise} \end{cases}$$

For each $A \in N$, if there exists an $m \in \mathbb{N}$, such that $dep(t) \leq m$ for each $t \in D_G(A)$, then the depth of $A$, denoted by $Dep(A)$, is $m$, otherwise we set $Dep(A) = \infty$. We call $\max\{Dep(A)\colon A \in N\}$ the depth of $G$. If the depth of some MCFG $G$ is $m$, then we say $G$ is $m$-restricted.

**Example 3.8** (non $m$-restricted MCFG)**.** Let an MCFG $G = (N, \Sigma, P, S)$ with $P = \{p_1, p_2, p_3, p_4\}, N = \{S, A\}, \Sigma = \{a, b, c\}$ and

$$p_1 = (S, [x_1^1 x_1^2 x_1^3], A), \qquad p_2 = (A, [x_1^2, x_1^3, x_1^1], A),$$
$$p_3 = (A, [ax_1^1, bx_1^2, cx_1^3], A), \qquad p_4 = (A, [a, b, c], \epsilon).$$

For each $m \in \mathbb{N}$, there exists a tree $dep(p_2(\underbrace{p_2(...(p_2}_{m}(p_4))))) = m + 1$, then $Dep(A) = \infty$. Hence $G$ is not restricted by any $m \in \mathbb{N}$.

**Lemma 3.2** (non-deleting $\epsilon$-free form)**.** Let $m \in \mathbb{N}_+$. For every $m$-MCFG $G$ there is a non-deleting $\epsilon$-free $m$-MCFG $G'$, such that $L(G) = L(G')$.

*Proof.* We can obtain $G'$ by applying the construction of $\epsilon$-free and non-deleting form in that order, in Theorem 3.2 and 3.1, respectively, since the construction of non-deleting form does not introduce new $\epsilon$'s. □

**Lemma 3.3** (restricted MCFG)**.** Let $m \in \mathbb{N}_+$. For each $m$-restricted non-deleting $\epsilon$-free MCFG $G$, there is an $(m-1)$-restricted non-deleting $\epsilon$-free MCFG $G'$, such that $L(G) = L(G')$.
*Construction.* Let $G = (N, \Sigma, P, S)$ be a non-deleting $\epsilon$-free MCFG of depth $m$ for some $m \in \mathbb{N}_+$. We construct $G' = (N, \Sigma, P', S)$ with

- $P' = \{p\colon p \in P, dep(t) \neq 1 \text{ for each } t \in T_P \text{ with } t(\epsilon) = p\} \cup \{p = (A, f \circ f_l, A_1 ... A_{l-1} A_{l+1} ... A_k B_1 ... B_x)\colon p = (A, f, A_1 ... A_{l-1} A_l A_{l+1} ... A_k) \in P, dep(t) = 1$ for each

$t \in T_P$ with $t(\epsilon) = (A, f, A_1...A_k), p_l = (A_l, f_l, B_1...B_x) \in P$ for smallest $l \in [k]$, such that $f_l$ contains terminal symbols$\}$, where $f \circ f_l$ is obtained from $f$ by replacing each $x_i^j \in X_f$ with $i = l$ by $f_l\langle j\rangle$, where each $x_{i'}^{j'} \in X_{f_l}$ in $f_l\langle j\rangle$ is replaced by $x_{i'+k-1}^{j'}$, and replacing each $x_i^j \in X_f$ with $i \neq l$ by $x_{i-i[\{l\}]}^j$.

*Proof.* $G'$ is non-deleting and $\epsilon$-free, since the construction does not delete the information used by any variable or introduce new $\epsilon$'s.
Let $(A, f, A_1...A_{l-1}A_lA_{l+1}...A_k) \in P$ with $dep(t) = 1$ for each $t \in T_P$ with $t(\epsilon) = (A, f, A_1...A_k)$, and $(A_l, f_l, B_1...B_x) \in P$ such that $f_l$ contains terminal symbols. Then for $w_i \in (\Sigma^*)^{sort(A_i)}, u_j \in (\Sigma^*)^{sort(B_j)}$, for each $i \in [k]$ and $j \in [x]$, there is

$$\begin{aligned} &f(w_1, ..., w_{l-1}, f_l(u_1, ..., u_x), w_{l+1}, ..., w_k) \\ =&(f \circ f_l)(w_1, ..., w_{l-1}, w_{l+1}, ..., w_k, u_1, ..., u_x) \end{aligned} \qquad (*)$$

The $(*)$ holds, since each variable $x_i^j \in X_f$ replaced by a component of $f_l$ has the same value as the component that replaces it, and each other variable $x_i^j \in X_f$ has the same value as the variable at the same position in $f \circ f_l$. We show $L(G) = L(G')$ by proving $L(G) \subseteq L(G')$ and $L(G') \subseteq L(G)$.

- We show that for each $t' \in T_{P'}$, there exits a $t \in T_P$, such that $yield(t') = yield(t)$ by structural induction over $t'$. Let $t' = p'(t'_1, ..., t'_k) \in T_{P'}$, where $p' = (A, f', A_1...A_k)$.
  - If $p' \in P$, then by induction hypothesis there exist $t_1, ..., t_k \in T_P$, such that $yield(t_i) = yield(t'_i)$ for each $i \in [k]$. Hence $yield(p'(t'_1, ..., t'_k)) = yield(p'(t_1, ..., t_k))$.
  - if $p' \notin P$, then there exists an $n \in [k]$, such that there is a production $q = (B, f_B, A_n...A_k) \in P$, and $p = (A, f, A_1...A_{l-1}A_lBA_{l+1}...A_{n-1}) \in P$, with $f' = f \circ f_B$ by construction. For each $t' = p'(t'_1, ..., t'_k) \in T_{P'}$, we have by induction hypothesis, that there exist $t_1, ..., t_{l-1}, t_l, t_{l+1}, ..., t_{n-1}, t_n$, $t_{n+1}, ..., t_k \in T_P$, such that $yield(t_i) = yield(t'_i)$ for each $i \in [k]$. By $(*)$ there exists $t = p(t_1, ..., t_{l-1}, t_l, q(t_n, ..., t_k), t_{l+1}, ..., t_{n-1}) \in T_P$, such that $yield(t) = yield(t')$. It follows that $L(G') \subseteq L(G)$.
- We show that for each $t \in T_P$, there exists a $t' \in T_{P'}$, such that $yield(t') = yield(t)$ by structural induction over $t$. Let $t = t = p(t_1, ..., t_k) \in T_P$, where $p = (A, f, A_1...A_{l-1}A_lA_{l+1}...A_k) \in P$.

- If $p \in P'$, then by induction hypothesis there exist $t'_1, ..., t'_k \in T_{P'}$, such that $yield(t_i) = yield(t'_i)$ for each $i \in [k]$. Hence $yield(p(t'_1, ..., t'_k)) = yield(p(t_1, ..., t_k))$.
- if $p \notin P'$, then there is a production $p_l = (A_l, f_l, B_1...B_x) \in P$, such that $f_l$ contains a terminal symbol, and $p' = (A, f \circ f_l, A_1...A_{l-1}A_{l+1}...A_kB_1...B_x) \in P'$. For each tree $t = p(t_1, ..., t_{l-1}, p_l(\xi_1, ..., \xi_x), t_{l+1}, ..., t_k) \in T_P$, we have by induction hypothesis, that there exist $t'_1, ..., t'_{l-1}, t'_{l+1}, ..., t'_k, \xi'_1, ..., \xi'_x \in T_{P'}$, such that $yield(t'_i) = yield(t_i)$ for each $i \in [k] \setminus \{l\}$, and $yield(\xi'_i) = yield(\xi_i)$ for each $i \in [x]$. By construction there exists $t' = p'(t'_1, ..., t'_{l-1}, t'_{l+1}, ..., t'_k, \xi'_1, ..., \xi'_x) \in T_{P'}$, and by $(*)$ we have $yield(t') = yield(t)$. It follows that $L(G) \subseteq L(G')$.

Since for each tree $t \in T_P$ with depth $m$, and each $w \in pos(t)$ with $dep(t|_w) = 1$, terminal symbols are constructed into the production $t(w)$, then for the tree $t'$ constructed from $t$, there is $dep(t') = (m-1)$. Hence $G'$ is $(m-1)$-restricted, and $L(G') = L(G)$. □

**Example 3.9** (construction of $(m-1)$-restricted MCFG)**.** Let an MCFG $G = (N, \Sigma, P, S)$, where $N = \{S, A, B, C, D\}$, $\Sigma = \{a, b, c, d\}$, $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, and

$$p_1 = (S, [x_1^1 x_2^1 x_1^2 x_2^2], AB), \qquad p_2 = (A, [x_1^1, x_1^2], C),$$
$$p_3 = (B, [x_1^1, x_1^2], D), \qquad p_4 = (C, [ax_1^1, bx_1^2], C),$$
$$p_5 = (D, [cx_1^1, dx_1^2], D), \qquad p_6 = (C, [a, b], \epsilon),$$
$$p_7 = (D, [c, d], \epsilon)$$

The depth of $G$ is 2. We apply the construction in Lemma 3.3, and get $G' = \{N, \Sigma, P', S\}$ with $P' = \{p_1, p'_2, p'_3, p_4, p_5, p_6, p_7, p''_2, p''_3\}$, and

$$p_1 = (S, [x_1^1 x_2^1 x_1^2 x_2^2], AB), \qquad p'_2 = (A, [ax_1^1, bx_1^2], C),$$
$$p'_3 = (B, [cx_1^1, dx_1^2], D), \qquad p_4 = (C, [ax_1^1, bx_1^2], C),$$
$$p_5 = (D, [cx_1^1, dx_1^2], D), \qquad p_6 = (C, [a, b], \epsilon),$$
$$p_7 = (D, [c, d], \epsilon), \qquad p''_2 = (A, [a, b], \epsilon),$$
$$p''_3 = (B, [c, d], \epsilon)$$

The depth of $G'$ is 1.

**Lemma 3.4** (0-restricted MCFG)**.** Let $m \in \mathbb{N}$. For each $m$-restricted MCFG $G$, there is an 0-restricted MCFG $G'$, such that $L(G) = L(G')$.

*Proof.* Since the constructions of non-deleting and $\epsilon$-free form both preserve the structure and each terminal symbol, then for each $m$-restricted MCFG $G$, there is an $m$-restricted non-deleting $\epsilon$-free MCFG $G''$ with $L(G) = L(G'')$.
Thus we apply the construction in Lemma 3.3 on $G''$ for $m$ times, and get the 0-restricted MCFG $G'$ with $L(G) = L(G')$. □

**Theorem 3.5** (lexicalized)**.** Let $m \in \mathbb{N}$. For each $m$-restricted MCFG $G$ with $\epsilon \notin L(G)$, there is a lexicalized MCFG $G'$, such that $L(G) = L(G')$.

*Construction.* Let $G = (N, \Sigma, P, S)$. We use the Lemma 3.4, and construct $G$ to 0-restricted $G'' = (N'', \Sigma, P'', S'')$, hence $L(G) = L(G'')$, and each production has at least one terminal symbol. We use the construction of terminal separated form in Theorem 3.3 to obtain $G'$, but for each production $(A, f, A_1 ... A_k) \in P''$ with terminal symbols $a_1, ..., a_n$ in $f$, we execute the construction over $\boxed{a_i}$ for each $i \in [n]$ with $i > 1$ only.

*Proof.* Since each production in $P''$ contains at least one terminal symbol in the composition function, then $G'$ is lexicalized. With the similar proof as in Theorem 3.3 we have $L(G) = L(G')$. □

**Example 3.10** (lexicalized form)**.** Let an MCFG $G = (N, \Sigma, P, S)$ with $N = \{S, A\}, \Sigma = \{a, b, c\}$, and

$$p_1 = (S, [x_1^1 x_1^2 x_1^3], A), \qquad p_2 = (A, [a x_1^1, b x_1^2, c x_1^3], A),$$
$$p_3 = (A, [a, b, c], \epsilon).$$

We apply the construction, and delete all non-terminal symbols that are not reachable from $S$. We have $G' = (N \cup \{\boxed{b}, \boxed{c}\}, \Sigma, P', S)$ with

$$p'_1 = (S, [a x_1^1 x_2^1 x_1^2 x_3^1 x_1^3], A \boxed{b} \boxed{c}), \qquad p'_2 = (A, [a x_1^1, x_2^1 x_1^2, x_3^1 x_1^3], A \boxed{b} \boxed{c}),$$
$$p'_3 = (A, [a, x_1^1, x_2^1], \boxed{b} \boxed{c}), \qquad p''_1 = (S, [a x_1^1 x_2^1], \boxed{b} \boxed{c}),$$
$$p_b = (\boxed{b}, [b], \epsilon), \qquad p_c = (\boxed{c}, [c], \epsilon).$$

# 4 Weighted multiple context-free grammers over strong bimonoids

## 4.1 Strong bimonoids

A *monoid* is an algebra $(\mathcal{A}, \cdot, 1)$ where $\cdot$ is associative and 1 is neutral element over the operation $\cdot$. An algebra $(\mathcal{A}, +, \cdot, 0, 1)$ is called a *strong bimonoid*, if:

- $(\mathcal{A}, +, 0)$ is a commutative monoid (i.e. the operation + is commutative),
- $(\mathcal{A}, \cdot, 1)$ is a monoid,
- for each $a \in \mathcal{A} : 0 \cdot a = 0 = a \cdot 0$.

A *strong bimonoid* $(\mathcal{A}, +, \cdot, 0, 1)$ is commutative, if $(\mathcal{A}, \cdot, 1)$ is commutative. A complete bimonoid is a commutative bimonoid equipped with an infinitary sum operation $\sum$. In this paper we write $\mathcal{A}$ instead of $(\mathcal{A}, +, \cdot, 0, 1)$.

**Example 4.1** (Strong bimonoids)**.** There are some strong bimonoids used in the processing of natural languages.

- Complete commutative semiring, e.g.

$$\begin{aligned} &\text{Boolean semiring } \mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1), \\ &\text{probability semiring Pr} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1), \\ &\text{Viterbi semiring } ([0,1], \max, \cdot, 0, 1). \end{aligned}$$

- Complete lattice,
- The tropical bimonoid, $(\mathbb{R}_{\geq 0} \cup \{\infty\}, +, \min, 0, \infty)$.
- The algebra $([0,1], \oplus, \cdot, 0, 1)$ with either $a \oplus b = a + b - a \cdot b$ or $a \oplus b = \min\{a + b, 1\}$

## 4.2 Weighted MCFG over strong bimonoids

We use the definition of *weighted MCFG* as presented by Denkinger [3, Definition 3] to define a *weighted MCFG over strong bimonoids.*

**Definition 4.1** ($\mathcal{A}$-weighted MCFG)**.** An $\mathcal{A}$-*weighted MCFG* is a tuple $(N, \Sigma, P, S, \mu)$ such that $(N, \Sigma, P, S)$ is an MCFG, $\mu : P \to \mathcal{A}\backslash\{0\}$, and $\mathcal{A}$ is a complete strong bimonoid.

**Definition 4.2** (weighted language of an $\mathcal{A}$-weighted MCFG)**.** Let $G = (N, \Sigma, P, S, \mu)$ be an $\mathcal{A}$-weighted MCFG over a complete strong bimonoid $\mathcal{A}$, $D_G := D_{(N,\Sigma,P,S)}$. For each $\xi = p(\xi_1, ..., \xi_k) \in D_G$. The *weight function of* $G$, denoted by $wt_G$, is inductively defined below:

$$wt_G: D_G \to \mathcal{A},\ \ wt_G(\xi) = wt_G(\xi_1) \cdot ... \cdot wt_G(\xi_k) \cdot \mu(p)$$

We abbreviate $wt_G$ to $wt$. Then we have *weighted language of* $G$, denoted by $[\![G]\!]$:

$$[\![G]\!]: \Sigma^* \to \mathcal{A},\ [\![G]\!](w) = \sum_{\xi \in D_G(w)} wt_G(\xi)$$

where $\sum$ is the infinitary sum operation of the complete strong bimonoid $\mathcal{A}$.

**Example 4.2** (simple example of a weighted MCFG)**.** Let $G = (N, \Sigma, P, S, \mu)$ be an weighted MCFG over $(\mathbb{R}, +, \cdot, 0, 1)$, where $N = \{S, A\}, \Sigma = \{a, b, c\}, P = (p_1, p_2, p_3, p_4)$ with:

$$\begin{aligned} &p_1 = (S, [\epsilon], \epsilon), && \mu(p_1) = 1, && p_2 = (S, [x_1^1 x_1^2 x_1^3], A), && \mu(p_2) = 3, \\ &p_3 = (A, [ax_1^1, bx_1^2, cx_1^3], A), && \mu(p_3) = 5, && p_4 = (A, [\epsilon, \epsilon, \epsilon], \epsilon), && \mu(p_4) = 2, \end{aligned}$$

then for $w = \epsilon$, $[\![G]\!](w) = \sum_{\xi \in D_G(w)} wt(\xi) = wt(\xi_1) + wt(\xi_2) = 1 + 3 \cdot 2 = 7$.

**Example 4.3** (simple example of a weighted MCFG)**.** We use the former example but using the weighting bimonoid $Lang_{\mathbb{N}} = (\mathcal{P}(\mathbb{N}^*), \cup, \cdot, \varnothing, \{\epsilon\})$, where $\mu$:

$$\begin{aligned} &p_1 = (S, [\epsilon], \epsilon), && \mu(p_1) = \{1\}, && p_2 = (S, [x_1^1 x_1^2 x_1^3], A), && \mu(p_2) = \{2\}, \\ &p_3 = (A, [ax_1^1, bx_1^2, cx_1^3], A), && \mu(p_3) = \{3\}, && p_4 = (A, [\epsilon, \epsilon, \epsilon], \epsilon), && \mu(p_4) = \{0\}, \end{aligned}$$

then for $w = \epsilon$, $[\![G]\!](w) = \sum_{\xi \in D_G(w)} wt(\xi) = wt(\xi_1) \cup wt(\xi_2) = \{1\} \cup \{20\} = \{1, 20\}$, where $\xi_1$ and $\xi_2$ are the same as in the example 3.2.

## 4.3 Normal forms of weighted MCFG over strong bimonoids

An $\mathcal{A}$-weighted MCFG $G = (N, \Sigma, P, S, \mu)$ is *non-deleting, $\epsilon$-free, terminal separated, monotone, strongly monotone, or lexicalized*, if $(N, \Sigma, P, S)$ is non-deleting, $\epsilon$-free, terminal separated, monotone or strongly monotone, respectively. From now on we use $\mathcal{A} = (\mathcal{A}, +, \cdot, 0, 1)$ to denote an arbitrary complete strong bimonoid.

**Corollary 4.1** (non-deleting normal form for $\mathcal{A}$-weighted MCFG)**.** For each $\mathcal{A}$-weighted MCFG $G$ there exists a non-deleting $\mathcal{A}$-weighted MCFG $G'$, such that $[\![G]\!] = [\![G']\!]$.

*Proof.* Let $G = (N, \Sigma, P, S, \mu)$. We recall Theorem 3.1. There exists a bijection $g: P' \to P$ with a given $\Psi$. We set $\mu' = g; \mu$ with the same proof by Denkinger [2, Lemma 2.7]. Hence $D_G = D_{G'}, [\![G]\!] = [\![G']\!]$. □

**Corollary 4.2** ($\epsilon$-free normal form for $\mathcal{A}$-weighted MCFG)**.** For each $\mathcal{A}$-weighted MCFG $G$ there exists an $\epsilon$-free $\mathcal{A}$-weighted MCFG $G'$, such that and $[\![G]\!] = [\![G']\!]$.

*Proof.* Let $G = (N, \Sigma, P, S, \mu)$. We fix $G' = (N', \Sigma, P', S', \mu')$ with

- $N', P', S'$ are exactly defined as in Theorem 3.2,
- for each production $p' = (A[\Psi_{f,\Psi_1,\dots,\Psi_k}], f_{\Psi,\Psi_1,\dots,\Psi_k}, A_1[\Psi_1]\dots A_k[\Psi_k]) \in P'$, we fix $\mu'(p') = \mu((A, f, A_1 \dots A_k))$,
- we fix $\mu'((S', [x_1^1], S[\{1\}])) = 1$, and $\mu'((S', [\epsilon], \epsilon)) = [\![G]\!](\epsilon)$.

Let $P'' = P' \setminus (\{(S', [\epsilon], \epsilon)\} \cup \{(S', [x_1^1], S[\{1\}])\})$. We recall $g$ and $\hat{g}$ from Theorem 3.2. Let $\xi'' = p''(\xi_1'', \dots, \xi_k'') \in T_{P''}, \xi = \hat{g}(\xi''), \xi_1 = \hat{g}(\xi_1''), \dots, \xi_k = \hat{g}(\xi_k'')$, and $p = g(p'')$ for some $\Psi_1, \dots, \Psi_k$. Since $g$ preserves the structure of trees in $T_{P''}$, and $g$ is a bijection, we have $wt_G(\xi) = wt_{G'}(\xi'')$.
Let $q = (S', [x_1^1], S[\{1\}])$, then for each $w \in L(G)$,
if $w = \epsilon$, then

$$[\![G']\!](w) = \mu(S', [\epsilon], \epsilon) = [\![G]\!](\epsilon)$$

Otherwise,

$$[\![G']\!](w) = \sum_{\xi \in D_{G'}(w)} wt_{G'}(\xi)$$

$$
\begin{aligned}
&= \sum_{q(\xi'') \in D_{G'}(w)} \mu'(q) \cdot wt_{G'}(\xi'') \\
&= \sum_{q(\xi'') \in D_{G'}(w)} 1 \cdot wt_G(\hat{g}(\xi'')) \\
&= \sum_{\xi \in D_G(w)} 1 \cdot wt_G(\xi) \\
&= [\![G]\!](w)
\end{aligned}
$$

Hence $D_G = D_{G'}, [\![G]\!] = [\![G']\!]$. □

**Corollary 4.3** (terminal separated normal form for $\mathcal{A}$-weighted MCFG)**.** For each $\mathcal{A}$-weighted MCFG $G$ there exists a terminal separated $\mathcal{A}$-weighted MCFG $G'$, such that $D_G = D_{G'}$, and $[\![G]\!] = [\![G']\!]$.

*Proof.* We recall Theorem 3.3. Let $\xi = p(\xi_1, ..., \xi_k) \in T_P$, $\xi' = p'(\xi'_1, ..., \xi'_k, \xi_{k+1}, ..., \xi_{k+n}) \in T'_{P'}$. We fix the weight of each production with a fresh non-terminal symbol on the left side to 1, and fix $\mu'(p') = \mu(p)$ for each $p' \in P'$ constructed from $p \in P$. Then we have

$$
\begin{aligned}
wt_G(\xi) &= wt_G(\xi_1) \cdot ... \cdot wt_G(\xi_k) \cdot \underbrace{1 \cdot ... \cdot 1}_{n} \cdot \mu(p) \\
&\overset{IH}{=} wt_{G'}(\xi'_1) \cdot ... \cdot wt_{G'}(\xi'_k) \cdot \underbrace{1 \cdot ... \cdot 1}_{n} \cdot \mu(p) \\
&= wt_{G'}(\xi'_1) \cdot ... \cdot wt_{G'}(\xi'_k) \cdot wt_{G'}(\xi_{k+1}) \cdot ... \cdot wt_{G'}(\xi_{k+n}) \cdot \mu'(p') \\
&= wt_{G'}(\xi'),
\end{aligned}
$$

Hence $[\![G]\!] = [\![G']\!]$. □

**Corollary 4.4** (monotone normal form for $\mathcal{A}$-weighted MCFG)**.** For each $\mathcal{A}$-weighted MCFG $G$ there exists a monotone $\mathcal{A}$-weighted MCFG $G'$, such that $[\![G]\!] = [\![G']\!]$.

*Proof.* We recall the bijection $g$ and the construction in Theorem 3.4. Let $G = (N, \Sigma, P, S, \mu)$, then we construct $G' = (N', \Sigma, P', S\langle\rangle, g^{-1}; \mu)$, where $(N', \Sigma, P', S\langle\rangle)$ is obtained by the construction from $G$. Since the construction without the permutation of non-terminal symbols, which is used for the strongly monotone form, preserves the structure of each tree, we have $[\![G]\!] = [\![G']\!]$. □

**Corollary 4.5** (strongly monotone normal form for $\mathcal{A}$-weighted MCFG)**.** For each $\mathcal{A}$-weighted MCFG $G$, where $\mathcal{A}$ is commutative, there exists a strongly monotone $\mathcal{A}$-weighted MCFG $G'$, such that $[\![G]\!] = [\![G']\!]$.

*Proof.* Let $G = (N, \Sigma, P, S, \mu)$.We recall the function $\hat{g}$ and $g$ in Theorem 3.4, and construct $G' = (N', \Sigma, P', S[(1)], g^{-1}; \mu)$, where $N', P'$ are constructed in the same way in Theorem 3.4. For each $A \in N$, $\pi \in sort(A)$, and tree $t' = p'(t'_{\gamma(1)}, ..., t'_{\gamma(k)}) \in (T_{P'})_{A[\pi]}$ with $\gamma$ obtained from $\pi$ by the construction, there is a unique $t = \hat{g}(t') = g(p')(t_1, ..., t_k) \in (T_P)_A$. We have by structural induction over $wt_G(t')$

$$\begin{aligned}
wt_{G'}(t') =& wt_{G'}(t'_{\gamma(1)}), ..., wt_{G'}(t'_{\gamma(k)}) \cdot (g^{-1}; \mu)(p') \\
=& wt_G(t_{\gamma(1)}), ..., wt_G(t_{\gamma(k)}) \cdot (g^{-1}; \mu)(p') && \text{(induction hypothesis)} \\
=& wt_G(t_1), ..., wt_G(t_k) \cdot (g^{-1}; \mu)(p') && \text{(commutative of } \cdot\text{)} \\
=& wt_G(t_1), ..., wt_G(t_k) \cdot \mu(p) \\
=& wt_G(t)
\end{aligned}$$

Hence $[\![G]\!] = [\![G']\!]$. □

Let $m \in \mathbb{N}$, $\mathcal{A}$ be a strong bimonoid. For each $\mathcal{A}$-weighted MCFG $G = (N, \Sigma, P, S, \mu)$, we say that $G$ is $m$-restricted, if $(N, \Sigma, P, S)$ is $m$-restricted. We say an MCFG $G = (N, \Sigma, P, S)$ is *unambiguous*, if for each $w \in (\Sigma^*)^*$, there exists at most one $t \in T_P$, such that $w = yield(t)$. An $\mathcal{A}$-weighted MCFG $G = (N, \Sigma, P, S, \mu)$ is unambiguous, if $(N, \Sigma, P, S)$ is unambiguous.

**Lemma 4.1** (unambiguous MCFG)**.** Let $m \in \mathbb{N}_+$. For each $m$-restricted unambiguous MCFG $G$, the $(m-1)$-restricted MCFG $G_{m-1}$ constructed from $G$ according to Lemma 3.3 is unambiguous.

*Proof.* Let $G = (N, \Sigma, P, S), G_{m-1} = (N, \Sigma, P_{m-1}, S)$. We recall Lemma 3.3. For each tree $t \in T_P$, there is a unique tree $t' \in T_{P_{m-1}}$ constructed from $t$, such that $yield(t) = yield(t')$.
We assume that $G_{m-1}$ is ambiguous. Then for some $w \in (\Sigma^*)^*$, there exist different trees $t'_1, ..., t'_n \in T_{P_{m-1}}$, such that $yield(t'_i) = w$ for each $i \in [n]$. Thus there exist $t_1, ..., t_n \in T_P$ that construct $t'_1, ..., t'_n$, respectively, such that $yield(t_i) = w$ for each $i \in [n]$, but then $G$ is ambiguous. ↯ □

**Corollary 4.6** (lexicalized normal form for $\mathcal{A}$-weighted MCFG)**.** Let $m \in \mathbb{N}$. For each $m$-restricted unambiguous $\mathcal{A}$-weighted MCFG $G$, where $\mathcal{A}$ is commutative and distributive, there exists a lexicalized $\mathcal{A}$-weighted MCFG $G'$, such that $[\![G]\!] = [\![G']\!]$.

*Proof.* Let $m \in \mathbb{N}, G = (N, \Sigma, P, S, \mu)$ be an $m$-restricted unambiguous $\mathcal{A}$-weighted MCFG. We recall the construction in Lemma 3.3. We fix the relation $g \subseteq P \times P'$ that assigns to each $p \in P$ all $p' \in P'$ it constructs, and fix the function $h: P \times P' \to P$ that assigns to each $(p, p') \in P \times P'$, where $p$ constructs $p'$, the $p_l$, according to the construction. We fix the function $\hat{g}: T_P \to T_{P'}$ that assigns to each tree in $T_P$ the tree it constructs.
We construct construct the $(m-1)$-restricted $\mathcal{A}$-weighted MCFG $G_{m-1} = (N_{m-1}, \Sigma, P_{m-1}, S, \mu_{m-1})$ from $G$ with

- $N_{m-1}, P_{m-1}$ are the same as $N', P'$ respectively in Lemma 3.3,
- for each $p' = (A, f \circ f_l, A_1 ... A_{l-1} A_{l+1} ... A_k B_1 ... B_x) \in P_{m-1}$,
  - if $p' \in P$, then $\mu_{m-1}(p') = \mu(p')$,
  - otherwise, we set $\mu_{m-1}(p') = \sum_{p \in g^{-1}(p')} \mu(h(p, p')) \cdot \mu(p)$.

We have by structural induction over $t'$

$$\begin{aligned}
wt_{G_{m-1}}(t') =& wt_{G_{m-1}}(t'_1) \cdot ... \cdot wt_{G_{m-1}}(t'_{l-1}) \cdot wt_{G_{m-1}}(t'_{l+1}) \cdot ... \cdot wt_{G_{m-1}}(t'_k) \cdot \\
& wt_{G_{m-1}}(\xi'_1) \cdot ... \cdot wt_{G_{m-1}}(\xi'_x) \cdot \mu_{m-1}(p') \\
=& wt_{G_{m-1}}(t'_1) \cdot ... \cdot wt_{G_{m-1}}(t'_{l-1}) \cdot wt_{G_{m-1}}(t'_{l+1}) \cdot ... \cdot wt_{G_{m-1}}(t'_k) \cdot \\
& wt_{G_{m-1}}(\xi'_1) \cdot ... \cdot wt_{G_{m-1}}(\xi'_x) \cdot \Big( \sum_{p \in g^{-1}(p')} \mu(h(p, p')) \cdot \mu(p) \Big) \\
& \text{(by construction)} \\
=& \sum_{p \in g^{-1}(p')} wt_{G_{m-1}}(t'_1) \cdot ... \cdot wt_{G_{m-1}}(t'_{l-1}) \cdot wt_{G_{m-1}}(t'_{l+1}) \cdot ... \cdot wt_{G_{m-1}}(t'_k) \cdot \\
& wt_{G_{m-1}}(\xi'_1) \cdot ... \cdot wt_{G_{m-1}}(\xi'_x) \cdot \mu(p) \cdot \mu(h(p, p')) \cdot \mu(p) \\
& \text{(distributive)} \\
=& \sum_{p \in g^{-1}(p')} wt_{G_{m-1}}(t'_1) \cdot ... \cdot wt_{G_{m-1}}(t'_{l-1}) \cdot (wt_{G_{m-1}}(\xi'_1) \cdot ... \cdot wt_{G_{m-1}}(\xi'_x) \cdot \\
& \mu(h(p, p'))) \cdot wt_{G_{m-1}}(t'_{l+1}) \cdot ... \cdot wt_{G_{m-1}}(t'_k) \cdot \mu(p) \quad \text{(commutative)} \\
=& \sum_{p \in g^{-1}(p')} \Big( \sum_{t_1 \in \hat{g}(t'_1)} wt_G(t_1) \Big) \cdot ... \cdot \Big( \sum_{t_{l-1} \in \hat{g}(t'_{l-1})} wt_G(t_{l-1}) \Big) \cdot
\end{aligned}$$

$$((\sum_{\xi_1 \in \hat{g}(\xi_1')} wt_G(\xi_1)) \cdot \ldots \cdot (\sum_{\xi_x \in \hat{g}(\xi_x')} wt_G(\xi_x) \cdot \mu(h(p,p')))) \cdot$$

$$(\sum_{t_{l+1} \in \hat{g}(t_{l+1}')} wt_G(t_{l+1})) \cdot \ldots \cdot (\sum_{t_k \in \hat{g}(t_k')} wt_G(t_k)) \cdot \mu(p)$$

(induction hypothesis)

$$= \sum_{t \in \hat{g}(t')} wt_G(t) \qquad \text{(dirstributive, definition of } wt\text{)}$$

For each string $w \in \Sigma^*$, since $G_{m-1}$ and $G$ are unambiguous, there are unique $\xi \in T_P$ and $\xi' \in T_{P'}$, such that $yield(\xi) = yield(\xi') = w$. Then we have

$$\begin{aligned}
[\![G_{m-1}]\!](w) =& wt_{G_{m-1}}(t') \\
=& \sum_{t \in \hat{g}(t')} wt_G(t) \\
=& wt_G(t) && (t \text{ is unique}) \\
=& [\![G]\!](w).
\end{aligned}$$

Hence $[\![G_{m-1}]\!] = [\![G]\!]$.

We repeat the former construction, until a 0-restricted $\mathcal{A}$-weighted MCFG $G_0 = (N_0, \Sigma, P_0, S, \mu_0)$ is generated, thus $[\![G_0]\!] = [\![G]\!]$. We construct the lexicalized $\mathcal{A}$-weighted MCFG $G' = (N', \Sigma, P', S)$ from $G_0$ with

- $N' = N_0 \cup \{\boxed{a} : a \in \Sigma\}$,
- $P'$ is constructed the same as in Theorem 3.5,
- $\mu'$ is constructed the same as in Corollary 4.3.

$[\![G_0]\!] = [\![G']\!]$. holds with the same proof of Corollary 4.3. Hence $[\![G']\!] = [\![G]\!]$. □

# 5 Conclusion

We have formally recalled *weighted MCFGs over strong bimonoids*, and modified the existing construction of above mentioned forms except of lexicalized form. The construction of lexicalized form is brand new, and works only for a syntactically restricted subset of MCFG.

There are no restrictions to the weight algebra for the construction of normal

forms, if a bijection exists between original trees and constructed ones, and the construction preserves structure of each tree. The constructions of non-deleting, monotone and $\epsilon$-free forms fulfill exactly that condition. For terminal separated form, the construction extends each trees with branches consisting of terminating productions only, which almost preserve the structure of each tree, thus there are no restrictions the weight algebra either. During the construction of strongly monotone forms, there are permutations of non-terminal symbols on the right side of some productions, which leads to the requirement of commutativity of the weight algebra. The construction of lexicalized form for unweighted case requires a restricted MCFG, and changes structure of each tree severely by merging some children trees to the father node. The construction of lexicalized form for a weighted MCFG, however, requires further restriction to MCFG, and commutativity and distributivity of the weight algebra. There could exist other weight functions for lexicalized form with different further restrictions to an MCFG.
Since the above claims are based on each construction, further work could be the searching for some alternative constructions with less restrictions.

# References

[1] Tobias Denkinger. An automata characterisation for multiple contest-free languages. *CoRR*, 2016.

[2] Tobias Denkinger. A Chomsky-Schützenberger representation for weighted multiple context-free languages. *CoRR*, June 2016.

[3] Tobias Denkinger. *A Chomsky-Schützenberger representation for weighted multiple context-free languages*. 2016.

[4] Marcus Kracht. *The mathematics of language*, volume 63. Walter de Gruyter, 2003.

[5] Marco Kuhlmann. *Dependency structures and lexicalized grammars*. PhD thesis, 2007.

[6] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.