FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 33, No 1 (2018), 41–61
https://doi.org/10.22190/FUMI1801041I

CROSS-MOMENTS COMPUTATION FOR STOCHASTIC CONTEXT-FREE GRAMMARS [*]

Velimir M. Ilić, Miroslav D. Ćirić and Miomir S. Stanković

Abstract. In this paper we consider the problem of efficient computation of cross-moments of a vector random variable represented by a stochastic context-free grammar. Two types of cross-moments are discussed. The sample space for the first one is the set of all derivations of the context-free grammar, and the sample space for the second one is the set of all derivations which generate a string belonging to the language of the grammar. In the past, this problem was widely studied, but mainly for the cross-moments of scalar variables and up to the second order. This paper presents new algorithms for computing the cross-moments of an arbitrary order, while the previously developed ones are derived as special cases.

Keywords: Cross-moments, stochastic context-free grammar, language of the grammar.

1. Introduction

The cross-moments of random variables modeled with stochastic context-free grammars (SCFG) are important quantities in the SCFG modeling [10] and statistics [1]. They are defined as expected value of the product of integer powers of the entries of random vector variable, which can represent string or derivation length, the number of rule occurrences in a derivation or uncertainty associated with the occurring rule. The expectation can be taken either with respect to the sample space of all SCFG derivations or with respect to the sample space of all derivations which generate a string belonging to the language of the grammar. Throughout this paper, the name *cross-moments* is usually used in the former case, while in the latter case the term *conditional cross-moments* is used.

The computation of cross-moments may become demanding if the sample space is large. In the past, this problem was widely studied, but mainly for the cross-

Received January 23, 2018; accepted March 31, 2018.
2010 *Mathematics Subject Classification.* Primary 68Q45 ; Secondary 68W05

[*]The authors were supported in part by Research supported by the Ministry of Science and Technological Development, Republic of Serbia, Grants No. 174013, ON 174026 and III44006

moments of scalar variables (called simply *moments*) and up to the second order. The first order moments computation, such as expected length of derivations and expected string length, is given in [26]. The computation of SCFG entropy is considered in [17]. The procedure for computing the moments of string and derivation length is given in [10], where the explicit formulas for the moments up to the second order are derived. First order conditional cross-moments are considered in [11], where the algorithm for conditional SCFG entropy is derived. A more general algorithm for computing the conditional cross-moments of a vector variable of the second order is derived in [16].

In this paper we give the recursive formulas for computing the cross-moments and the conditional cross-moments of an arbitrary order, for a vector variable which factorizes according to a certain rule which is satisfied in the case of string or derivation length, the number of rule occurrences in derivation or uncertainty associated with the occurring rule. The formulas are derived by differentiation of the recursive equations for the moment-generating function [22], which are obtained from the algorithms for computing the partition function of a SCFG [18] for the cross-moments and with the inside algorithm [15], [8] for the conditional cross-moments.

The paper is organized as follows. Section 2 introduces multi-index notation, which is used throughout the paper, and reviews some preliminary notions about generalized Leibniz's formula, basic algebraic structures, and context free grammars. In Section 3 we give the formal definition of SCFG cross-moments and derive the recursive equations for cross-moments computation. The conditional cross moments are considered in Section 4.

2. Preliminaries

This section provides some basic definitions and theorems which are used in the paper. We review the multiindex formulation of the *Generalized Leibniz's* formula [21], and basic notions from the theory of weighted context free grammars, according to [18] and [19].

2.1. Multiindexes, Multinomial theorem and Generalized Leibniz's formula

Multi-indexes. A multi-index is defined as a tuple of nonnegative integers $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$. We define its dimension as $\dim(\boldsymbol{\alpha}) = d$ and its length as the sum $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_d$. The multi-index factorial is $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_d!$. The zero multi-index is $\mathbf{0} = (0, \dots, 0)$.

If $\boldsymbol{\beta} = (\beta_1, \dots, \beta_d) \in \mathbb{N}_0^d$, we write $\boldsymbol{\beta} < \boldsymbol{\alpha}$ if $\beta_i < \alpha_i$ for $i = 1, \dots, d$. We write $\boldsymbol{\beta} \leqslant \boldsymbol{\alpha}$ provided $\beta_i \leqslant \alpha_i$ for $i = 1, \dots, d$. The sum and difference of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined to be $\boldsymbol{\alpha} \pm \boldsymbol{\beta} = (\alpha_1 \pm \beta_1, \dots, \alpha_d \pm \beta_d)$.

If $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N$ are multi-indexes and $\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_N = \boldsymbol{\alpha}$, we define the multinomial coefficients to be

$$\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_N} = \frac{\boldsymbol{\alpha}!}{\boldsymbol{\beta}_1!\cdots\boldsymbol{\beta}_N!}.$$

For vectors $\boldsymbol{z}_1,\dots,\boldsymbol{z}_N) \in \mathbb{R}^d$ and a multi-index $\boldsymbol{\beta} = (\beta_1,\dots,\beta_d) \in \mathbb{N}_0^d$, the multi-index power is defined to be

$$\boldsymbol{z}^{\boldsymbol{\beta}} = z_1^{\beta_1}\cdots z_d^{\beta_d}.$$

Multinomial theorem and Generalized Leibniz's formula. With these settings, the multinomial theorem [20] can be expressed as

$$\Big(\sum_{i=1}^{N} \boldsymbol{z}_i\Big)^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_N} \prod_{i=1}^{N} \boldsymbol{z}_i^{\boldsymbol{\beta}_i},$$

for a vector $\boldsymbol{z} = (z_1,\dots,z_d) \in \mathbb{R}^d$ and $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_d) \in \mathbb{N}_0^d$.

Let $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_d)$ and let $C_{\boldsymbol{\alpha}}$ denote the set of all functions $u : \mathbb{R}^d \to \mathbb{R}$ having $\boldsymbol{\alpha}$-th partial derivative at zero. For a function $u : \mathbb{R}^d \to \mathbb{R}$, we define the partial derivative at zero of an order $\boldsymbol{\alpha}$ as

$$\mathcal{D}^{(\boldsymbol{\alpha})}\big\{u\big\} = \left.\frac{\partial^{|\boldsymbol{\alpha}|} u(t_1,\dots,t_d)}{\partial^{\alpha_1} t_1 \dots \partial^{\alpha_d} t_d}\right|_{t=0}.$$

Note that $\mathcal{D}^{(\mathbf{0})}\{u\} = u(\boldsymbol{t})$. According to the *generalized Leibniz's formula* [21], the following equality holds:

$$\mathcal{D}^{(\boldsymbol{\alpha})}\big\{FG\big\} = \sum_{\mathbf{0}\leqslant\boldsymbol{\beta}\leqslant\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} \mathcal{D}^{(\boldsymbol{\beta})}\big\{F\big\} \cdot \mathcal{D}^{(\boldsymbol{\alpha}-\boldsymbol{\beta})}\big\{G\big\}, \tag{2.1}$$

for all $F, G \in C_{\boldsymbol{\alpha}}$. The derivative of the product of more than two functions can be found according to [25]

$$\mathcal{D}^{(\boldsymbol{\alpha})}\Big\{\prod_{i=1}^{N} F_i\Big\} = \sum_{\boldsymbol{\beta}_1+\cdot+\boldsymbol{\beta}_m=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_N} \prod_{i=1}^{N} \mathcal{D}^{(\boldsymbol{\beta}_i)}\big\{F_i\big\}, \tag{2.2}$$

for all $F_i \in C_{\boldsymbol{\alpha}};\ i = 1,\dots,N$.

Tuples of elements indexed with multi-indexes. The set of all multi-indexes lower than or equal to $\boldsymbol{\nu}$ is denoted with $\mathcal{A}_{\boldsymbol{\nu}}$,

$$\mathcal{A}_{\boldsymbol{\nu}} = \big\{\boldsymbol{\alpha} \in \mathbb{N}_0^{\dim(\boldsymbol{\nu})} \mid \boldsymbol{\alpha} \leqslant \boldsymbol{\nu}\big\},$$

and $|\mathcal{A}_{\boldsymbol{\nu}}|$ denotes its cardinality.

For $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_d)$ and $\boldsymbol{\beta} = (\beta_1,\dots,\beta_d)$, we define the lexicographic order relation $\prec$, so that $\boldsymbol{\alpha} \prec \boldsymbol{\beta}$ if

$$\alpha_1 = \beta_1,\dots,\alpha_n = \beta_n \text{ and } \alpha_{n+1} < \beta_{n+1}.$$

Let $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d) \in \mathbb{N}_0^d$ be a multi-index and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{|\mathcal{A}_\nu|}$ be multi-indexes from $\mathcal{A}_{\boldsymbol{\nu}}$ such that $\mathbf{0} = \boldsymbol{\alpha}_1 \prec \boldsymbol{\alpha}_2 \prec \cdots \prec \boldsymbol{\alpha}_{|\mathcal{A}_\nu|} = \boldsymbol{\nu}$. Let $\boldsymbol{z}$ be a function which associates a real number $z^{(\boldsymbol{\alpha}_i)}$ to each $\boldsymbol{\alpha}_i$ from $\mathcal{A}_{\boldsymbol{\nu}}$, i.e. $\boldsymbol{z}$ is a vector from $\mathbb{R}^{|\mathcal{A}_\nu|}$ indexed by multindexes. We use the following notation for the vector $\boldsymbol{z}$:

$$\boldsymbol{z} = \left(\ z^{(\boldsymbol{\alpha})}\ \right)_{\boldsymbol{\alpha} \in \mathcal{A}_\nu} = (z^{(\boldsymbol{\alpha}_1)}, \ldots, z^{(\boldsymbol{\alpha}_{|\mathcal{A}_\nu|})}).$$

2.2. Semirings

A *monoid* is a triple $(\mathbb{K}, \oplus, 0)$, where $\oplus$ is an associative binary operation on the set $\mathbb{K}$ and 0 is the identity element for $\oplus$, i.e. $a \otimes 0 = 0 \oplus a = a$, for all $a \in \mathbb{K}$. A monoid is commutative if the operation $\oplus$ is commutative.

Example 2.1. Let Σ be a non-empty set. The *free monoid* $\boldsymbol{\Sigma}^* = (\Sigma, \cdot, \epsilon)$ over Σ is a monoid, where the carrier set $\Sigma^* = \{\, a_1 \ldots a_n \mid n \in \mathbb{N}_0,\, a_i \in \Sigma\, (1 \leqslant i \leqslant n)\}$ is the set of all *strings* over Σ and ϵ is the (unique) empty string of length zero. The operation $\cdot$ denotes the composition (concatenation) of strings defined by $\boldsymbol{u}_1 \cdot \boldsymbol{u}_2 = \boldsymbol{u}_1\boldsymbol{u}_2$ for all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \Sigma^*$.

A *semiring* is a tuple $(\mathbb{K}, \oplus, \otimes, 0, 1)$ such that

1. $(\mathbb{K}, \oplus, 0)$ is a commutative monoid with 0 as the identity element for $\oplus$,
2. $(\mathbb{K}, \otimes, 1)$ is a monoid with 1 as the identity element for $\otimes$,
3. $\otimes$ distributes over $\oplus$, i.e. $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ and $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$, for all a, b, c in $\mathbb{K}$,
4. 0 is an annihilator for $\otimes$, i.e. $a \otimes 0 = 0 \otimes a = 0$, for every a in $\mathbb{K}$.

A semiring is commutative if the operation $\otimes$ is commutative. The operations $\oplus$ and $\otimes$ are called the addition and the multiplication in $\mathbb{K}$. For a topology τ we define the topological semiring as a pair $\big(\mathbb{K}, \tau\big)$.

Example 2.2. If $C_{\boldsymbol{\alpha}}$ denotes the set of all functions $u : \mathbb{R}^d \to \mathbb{R}$ having the $\boldsymbol{\alpha}$-th partial derivative at zero, $\mathbf{1}$ is identity function and $\mathbf{0}$ is zero function, then we can define a semiring of $\boldsymbol{\alpha}$-continuous functions as $(C_{\boldsymbol{\alpha}}, \cdot, +, \mathbf{0}, \mathbf{1})$.

2.3. Weighted and stochastic context-free grammars

By a *weighted context-free grammar (WCFG)* over a commutative semiring $\big(\mathbb{K}, +, \cdot, 1, 0\big)$ we mean a tuple $G = \big(\Sigma, \mathcal{N}, S, \mathcal{R}, w\big)$, where

- $\Sigma = \big\{w_1, \ldots, w_{|\Sigma|}\big\}$ is a finite set of *terminals*,
- $\mathcal{N} = \big\{A_1, \ldots, A_{|\mathcal{N}|}\big\}$ is a finite set of *nonterminals* disjoint with Σ,
- $S \in \mathcal{N}$ is called the *start symbol* (throughout the paper it is usually assumed that $S = A_1$),

- $\mathcal{R} \subseteq \mathcal{N} \times (\Sigma \cup \mathcal{N})^*$ is a finite set of rules. A rule $(A, \alpha) \in \mathcal{R}$ is commonly written as $A \to \alpha$, where the nonterminal A is called the *premise*. The set of all rules $A_i \to B_{i,j}$, $B_{i,j} \in (\mathcal{N} \cup \Sigma)^*$ will be denoted by $\mathcal{R}_i$.

- $w : \mathcal{R} \to \mathbb{K}$ is the function called *weight*.

The *left-most rewriting relation* $\Rightarrow$ associated with G is defined as the set of triples $\big(\alpha, \pi, \beta\big) \in (\Sigma \cup \mathcal{N})^* \times \mathcal{R} \times (\Sigma \cup \mathcal{N})^*$, for which there is a terminal string $\boldsymbol{u} \in \Sigma^*$ and a nonterminal string $\delta \in (\Sigma \cup \mathcal{N})^*$, along with a nonterminal $A \in \mathcal{N}$ and a string $\gamma \in (\Sigma \cup \mathcal{N})^*$ such that $\alpha = \boldsymbol{u}A\delta$, $\beta = \boldsymbol{u}\gamma\delta$, and $\pi = A \to \gamma$ is a rule from $\mathcal{R}$. The left-most relation triple $\big(\alpha, \pi, \beta\big)$ will be denoted by $\alpha \overset{\pi}{\Rightarrow} \beta$. The *left-most derivation* (hereinafter the *derivation*) in this grammar is a string $\pi_1, \ldots, \pi_n \in \mathcal{R}^*$ for which there are grammar symbols $\alpha, \beta \in \Sigma \cup \mathcal{N}$ such that we can derive β from α by applying the rewriting rules $\pi_1, \ldots, \pi_n$: $\alpha \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_n}{\Rightarrow} \beta$. The weight function is extended to derivations such that $w\ (\pi_1 \cdots \pi_N) = w(\pi_1) \cdots w(\pi_N)$, for all $\pi_1 \cdots \pi_N \in R^*$. A nonterminal A is *productive* if there exists a derivation $\pi_1 \cdots \pi_k$ such that $A \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_k}{\Rightarrow} \boldsymbol{u}, \boldsymbol{u} \in \Sigma^*$. A nonterminal A is *accessible* from a nonterminal B if there exist derivations $\pi_1 \cdots \pi_k$ such that $B \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_k}{\Rightarrow} \eta A \xi$ where $\eta, \xi \in (\Sigma \cup \mathcal{N})^*$ (if A is accessible from S, then it is simply accessible). A nonterminal A is *useful* if it is accessible and productive (otherwise, it is *useless*). We say that a WCFG has a cycle if there is a derivation $\pi_1, \ldots, \pi_n$, such that for a nonterminal A it holds that $A \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_n}{\Rightarrow} A$. Otherwise, the WCFG is cycle-free.

A weighted context-free grammar $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ over the probability semiring $\big(\mathbb{R}_+, +, \cdot, 0, 1\big)$ is called a *stochastic context-free grammar* (SCFG) if the weight p maps all rules to the real unit interval $[0, 1]$. A *SCFG* is *reduced* if $p(A \to \gamma) > 0$ for all $A \to \gamma \in \mathcal{R}$ and each nonterminal A, and all nonterminals are useful. In this paper we consider only the reduced *SCFGs*. In addition, we assume that the *SCFG* is *proper*, which means that the weight function p gives us a probability distribution over the rules that we can apply, i.e. $\sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) = 1$ for all $1 \leqslant i \leqslant |\mathcal{N}|$.

For a stochastic context-free grammar $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ we define the subgrammar $G_i = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p_i\big)$ with the start symbol A_i, where $\mathcal{N}_i$ is the set which consists of A_i and nonterminals accessible from A_i and $\mathcal{R}_i \subseteq \mathcal{R}$ is the set of rules in which only nonterminals from $\mathcal{N}_i$ appear as premises and p_i is restriction of p to $\mathcal{R}_i$, such that $p_i(\pi) = p(\pi)$ for each $\pi \in \mathcal{R}_i$. Note that if G is reduced, then G_i also has this property.

Let $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ be a stochastic context-free grammar and Ω the set of all derivations in G. The grammar G is *consistent* if

$$\sum_{\boldsymbol{\pi} \in \Omega} p(\boldsymbol{\pi}) = 1.$$

Booth and Thompson [2] gave the consistency condition by the following theorem.

Theorem 2.1. *A reduced stochastic context-free grammar G is consistent if $\rho(M) < 1$, where $\rho(M)$ is the absolute value of the largest eigenvalue of the expectation matrix $M = [M_{i,n}], 1 \leqslant i, n \leqslant |\mathcal{N}|$ defined by*

$$M_{i,n} = \sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) r_n(i,j), \tag{2.3}$$

where $r_n(i,j)$ denotes the number of times the nonterminal A_n appears on the right-hand side of the rule $\pi = A_i \to B_{i,j}$.

Note that if G is reduced, then G_i also has this property. In addition, the expectation matrices $\mathrm{M}^{(i)}$ of all subgrammars G_i are the principal submatrices of M, and according to [9] (Corollary 8.1.20), $\rho(\mathrm{M}^{(i)}) \leqslant \rho(\mathrm{M})$. Thus, the conditions from Theorem 2.1 are satisfied and G_i are also consistent, i.e.

$$\sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) = 1, \tag{2.4}$$

for $1 \leqslant i \leqslant |\mathcal{N}|$, where Ω_i is the set of all derivations starting at $A_i \in \mathcal{N}$.

3. Cross-moments computation of SCFG

In this section we first define cross-moments and the moment-generating function of SCFG. After that, we provide formulas for the computation of cross moments up to an arbitrary order. In the end, we retrieve, as special cases, the formulas for the first and the second order moments, previously derived in [2] and [10].

3.1. Cross-moments and moment-generating function of SCFG

Let $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ be reduced and consistent SCFG, and $\boldsymbol{X}_i = \big[X_{i,1}, \dots, X_{i,D}\big]^T : \Omega_i \to \mathbb{R}^D$ be random variables distributed according to the p_i, which are restrictions of p to $\mathcal{R}_i$ ($p_i(\boldsymbol{\pi}) = p(\boldsymbol{\pi})$), for $1 \leqslant i \leqslant |\mathcal{N}|$. The *i-th cross-moment of an order* $\boldsymbol{\nu} = (\nu_1, \dots, \nu_D)$ of $\boldsymbol{X}_i$ is defined with

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i} = \sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) \cdot X_{i,1}(\boldsymbol{\pi})^{\nu_1} \cdots X_{i,D}(\boldsymbol{\pi})^{\nu_d} = \sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi})\ \boldsymbol{X}_i(\boldsymbol{\pi})^{\boldsymbol{\nu}}. \tag{3.1}$$

In this paper we consider the random vectors $\boldsymbol{X}_i : \Omega_i \to \mathbb{R}^D$, which can be represented as the sum of random vectors $\boldsymbol{Y} : \mathcal{R} \to \mathbb{R}^D$:

$$\boldsymbol{X}_i(\pi_1 \cdots \pi_N) = \boldsymbol{Y}\big(\pi_1\big) + \cdots + \boldsymbol{Y}\big(\pi_N\big), \tag{3.2}$$

for all $\pi_1 \cdots \pi_N \in \Omega_i$. This assumption may seem too restrictive, but it holds in some important cases: (1) If $\boldsymbol{X}(\boldsymbol{\pi})$ represents derivation length starting from a nonterminal A_i, then $\boldsymbol{Y}(\pi_i) = 1$; (2) if $\boldsymbol{X}(\boldsymbol{\pi})$ is the length of a string derived from a nonterminal A_i, then $\boldsymbol{Y}(\pi_i)$ equals the number of terminals on the right-hand side of π_i; (3) if $\boldsymbol{X}(\boldsymbol{\pi})$ represents the self-information of derivation $\boldsymbol{\pi}$ [11], then $\boldsymbol{Y}(\pi_i) = -\log p(\pi_i)$.

Following the Proposition 6 from [4], it can be shown that the cross-moments are bounded if the factorization (4.5) holds and, for all $\boldsymbol{t} = (t_1, \dots, t_D)$; $|t_i| < 1$, we have

$$\begin{aligned} &\sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\ \big(\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})\big)^{\boldsymbol{\nu}} < \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\ \boldsymbol{X}_i(\boldsymbol{\pi})^{\boldsymbol{\nu}} < C < \infty \quad \Rightarrow \\ &\qquad \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) \big(\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})\big)^k = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})}. \end{aligned} \tag{3.3}$$

Accordingly, we can define the *i-th moment-generating function (MGF)*, as the function $M_{p,\boldsymbol{X}_i} : \mathbb{R}^D \to \mathbb{R}$, where

$$M_{p,\boldsymbol{X}_i}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})}, \tag{3.4}$$

for all $\boldsymbol{t} \in \mathbb{R}^D$, and the cross-moments can be retrieved from the *MGF* by differentiating:

$$\mu^{(\boldsymbol{\nu})}_{p,\boldsymbol{X}_i} = \frac{\partial^{|\boldsymbol{\nu}|} M_{p,\boldsymbol{X}_i}(\boldsymbol{t})}{\partial^{\nu_1} t_1 \dots \partial^{\nu_D} t_d}\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \mathcal{D}_{\boldsymbol{\nu}}\Big\{M_{p,\boldsymbol{X}_i}\Big\} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\ \boldsymbol{X}_i(\boldsymbol{\pi})^{\boldsymbol{\nu}}. \tag{3.5}$$

Note that

$$\mu^{(0)}_{p,\boldsymbol{X}_i} = \mathcal{D}_{\boldsymbol{0}}\big\{M_{p,\boldsymbol{X}_i}\big\} = \Big(\sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}(\boldsymbol{\pi})}\Big)\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) = 1. \tag{3.6}$$

The direct cross-moments computation by enumerating all derivations is inefficient, since it requires the $\mathcal{O}(|\Omega|)$ operations, and it even becomes infeasible when Ω is an infinite set. On the other hand, if we can derive the expressions for efficient computation of the moment-generating function (4.2), the moment can be retrieved by differentiation.

3.2. Cross-moments computation of SCFG

For the grammar $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$, we define the *moment-generating grammar* $\widetilde{G} = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\big)$ endowed with a topology induced by supremum norm and with the weight function taking values from the semiring of $\boldsymbol{\alpha}$ continuous functions, $w : \mathcal{R} \to C_{\boldsymbol{\alpha}}$, defined with

$$w(\pi) = p(\pi) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi)}, \tag{3.7}$$

for all $\pi \in \mathcal{R}$. A derivation $\boldsymbol{\pi} = \pi_1 \cdots \pi_N$ in G with the weight $p(\boldsymbol{\pi}) = p(\pi_1) \cdots p(\pi_N)$ is also a derivation in $\widetilde{G}$, for which the weight is given with

$$w(\boldsymbol{\pi}) = w(\pi_1) \cdots w(\pi_N) = p(\pi_1) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_1)} \cdots p(\pi_N) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_N)} = p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}(\boldsymbol{\pi})}. \tag{3.8}$$

The i-th *MGF* can now be expressed as the weights sum of derivation in Ω_i as

$$M_{p,\boldsymbol{X}_i}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})} = \sum_{\boldsymbol{\pi} \in \Omega_i} w(\boldsymbol{\pi}). \tag{3.9}$$

Thus, the i-th *MGF* represents the i-th *partition function* of grammar $\widetilde{G}$ and the problem of *MGF* computation is reduced to the problem of the *partition function* computation. By factoring out the first rewriting of each derivation in the sum, using the distributive law, the partition function can be expressed with the system [18]:

$$M_{p,\boldsymbol{X}_i} = \sum_{j=1}^{|\mathcal{R}_i|} w(A_i \to B_{i,j}) \cdot \prod_{k=1}^{|\mathcal{N}|} M_{p,\boldsymbol{X}_k}^{r_k(i,j)}, \tag{3.10}$$

where $1 \leqslant i \leqslant |\mathcal{N}|$ and $r_k(i,j)$ denotes the number of times the nonterminal A_k appears on the right-hand side of the rule $A_i \to B_{i,j}$.

The cross-moments, $\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i} = \mathcal{D}_{\boldsymbol{\alpha}}\Big\{M_{p,\boldsymbol{X}_i}\Big\}$, can be obtained by applying the generalized Leibniz's formula (2.1) to (3.10), which leads us to the following system:

$$\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i} = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta} \leqslant \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} \mathcal{D}_{\boldsymbol{\alpha}-\boldsymbol{\beta}}\big\{w\big(A_i \to B_{i,j}\big)\big\} \cdot \mathcal{D}_{\boldsymbol{\beta}}\Big\{\prod_{k=1}^{|\mathcal{N}|} M_{p,\boldsymbol{X}_k}^{r_k(i,j)}\Big\}, \tag{3.11}$$

where

$$\mathcal{D}_{\boldsymbol{\alpha}-\boldsymbol{\beta}}\big\{w\big(A_i \to B_{i,j}\big)\big\} = p\big(A_i \to B_{i,j}\big) \cdot \boldsymbol{Y}\big(A_i \to B_{i,j}\big)^{\boldsymbol{\alpha}-\boldsymbol{\beta}}, \tag{3.12}$$

since $w(\pi) = p(\pi) e^{\boldsymbol{t}^T \cdot \boldsymbol{Y}}$, for $\pi \in \mathcal{R}$. According to the generalized Leibniz's rule (2.2), we have

$$\mathcal{D}_{\boldsymbol{\beta}}\Big\{\prod_{k=1}^{|\mathcal{N}|} M_{p,\boldsymbol{X}_k}^{r_k(i,j)}\Big\} = \sum_{\boldsymbol{\gamma}_1 + \cdots + \boldsymbol{\gamma}_{|\mathcal{N}|} = \boldsymbol{\beta}} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \mathcal{D}_{\boldsymbol{\gamma}_k}\big\{M_{p,\boldsymbol{X}_k}^{r_k(i,j)}\big\} \tag{3.13}$$

and

$$\begin{aligned} \mathcal{D}_{\boldsymbol{\gamma}_k}\big\{M_{p,\boldsymbol{X}_k}^{r_k(i,j)}\big\} &= \mathcal{D}_{\boldsymbol{\gamma}_k}\Big\{\prod_{l=1}^{r_k(i,j)} M_{p,\boldsymbol{X}_k}\Big\} \\ &= \sum_{\boldsymbol{\delta}_1 + \cdots + \boldsymbol{\delta}_{r_k(i,j)} = \boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_k(i,j)}} \prod_{l=1}^{r_k(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_k}. \end{aligned} \tag{3.14}$$

By substituting (3.14) and (3.13) in (3.11), we obtain:

$$\mu_{p,\boldsymbol{X}_i}^{(\alpha)} = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta} \leqslant \boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{3.15}$$

where

$$Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} p(A_i \to B_{i,j}) \cdot \boldsymbol{Y}(A_i \to B_{i,j})^{\boldsymbol{\alpha}-\boldsymbol{\beta}} \cdot \sum_{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\beta}} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(i,j)}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_k(i,j)}} \prod_{l=1}^{r_k(i,j)} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}. \tag{3.16}$$

To solve the system (3.15), we split it into two parts: one dependent and the other not dependent on $\mu_{p,\boldsymbol{X}_i}^{(\alpha)}$:

$$\mu_{p,\boldsymbol{X}_i}^{(\alpha)} = \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\alpha}) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta} < \boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta}), \tag{3.17}$$

where

$$Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\alpha}) = p(A_i \to B_{i,j}) \cdot W_{i,j}(\boldsymbol{\alpha}) \tag{3.18}$$

and

$$W_{i,j}(\boldsymbol{\alpha}) = \sum_{\gamma_1+\cdots+\gamma_{|\mathcal{N}|}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(i,j)}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_k(i,j)}} \prod_{l=1}^{r_k(i,j)} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}. \tag{3.19}$$

Further, if we set

$$H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(i,j)}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_k(i,j)}} \prod_{l=1}^{r_k(i,j)} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}, \tag{3.20}$$

the expression for $W_{\boldsymbol{\alpha}}(B_{i,j})$ can be rewritten as:

$$W_{i,j}(\boldsymbol{\alpha}) = \sum_{n=1}^{|\mathcal{N}|} H_{i,j}^{(n)}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) + \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}), \tag{3.21}$$

where $H_{i,j}^{(n)}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ stands for $H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ with $\boldsymbol{\gamma}_n = \boldsymbol{\alpha}$ and all other $\boldsymbol{\gamma}$-s equal zero, which, according to (3.20), is

$$H_{i,j}^{(n)}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu_{p,\boldsymbol{X}_n}^{(\delta_l)} \cdot \prod_{\substack{k=1 \\ k\neq n}}^{|\mathcal{N}|} \prod_{l=1}^{r_k(i,j)} \mu_{p,\boldsymbol{X}_k}^{(0)}. \tag{3.22}$$

Finally, after using of $\mu^{(0)}_{p,\boldsymbol{X}_k} = 1$, we obtain

$$H^{(n)}_{i,j}\big(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}\big) = \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n}, \tag{3.23}$$

which can be rewritten using the same procedure as

(3.24)

$$\begin{aligned} H^{(n)}_{i,j}\big(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}\big) &= \sum_{s=1}^{r_n(i,j)} \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_m} \\ &= r_n(i,j) \cdot \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n}. \end{aligned}$$

By substitution of (3.24) in (3.21), it follows that:

$$\begin{aligned} W_{i,j}\big(\boldsymbol{\alpha}\big) &= \sum_{n=1}^{|\mathcal{N}|} r_n(i,j) \cdot \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \\ &\sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n} + \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}\big(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}\big). \end{aligned} \tag{3.25}$$

Further, by substitution of (3.25) and (3.18) in (3.17), the moment can be expressed with:

$$\begin{aligned} \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i} &= \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(i,j) \cdot \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \\ &\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n} + \\ &\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}\big(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}\big) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta}<\boldsymbol{\alpha}} Q_{i,j}\big(\boldsymbol{\alpha}, \boldsymbol{\beta}\big), \end{aligned} \tag{3.26}$$

where $H_{i,j}\big(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}\big)$ and $Q_{i,j}\big(\boldsymbol{\alpha}, \boldsymbol{\beta}\big)$ are given with (3.20) and (3.16). Finally, if we introduce

(3.27)
$$c_i^{(\boldsymbol{\alpha})} = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} +$$
$$\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}\big(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}\big) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta}<\boldsymbol{\alpha}} Q_{i,j}\big(\boldsymbol{\alpha},\boldsymbol{\beta}\big),$$

the equation (3.26) can be more compactly written as:

$$\mu_{p,\boldsymbol{X}_i}^{(\boldsymbol{\alpha})} = \sum_{n=1}^{|\mathcal{N}|} \mathrm{M}_{i,n} \cdot \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\alpha})} + c_i^{(\boldsymbol{\alpha})}, \tag{3.28}$$

or in a matrix form:

$$\mu_p^{(\boldsymbol{\alpha})} = \mathrm{M} \cdot \mu_p^{(\boldsymbol{\alpha})} + \boldsymbol{c}^{(\boldsymbol{\alpha})}, \tag{3.29}$$

where $\mu_p^{(\boldsymbol{\alpha})} = \big[\mu_{p,\boldsymbol{X}_1}^{(\alpha)}, \ldots, \mu_{p,\boldsymbol{X}_{|\mathcal{N}|}}^{(\alpha)}\big]^T$ is the *cross-moment vector* $\boldsymbol{c}^{(\boldsymbol{\alpha})} = \big[c_1^{(\boldsymbol{\alpha})}, \ldots, c_{|\mathcal{N}|}^{(\boldsymbol{\alpha})}\big]^T$ and M is the momentum matrix defined in Theorem 2.1. Since we assume that the condition $\rho(\mathrm{M}) < 1$ given in Theorem 2.1 is satisfied, $\mathrm{I} - \mathrm{M}$ is invertible, and the matrix equation has a unique solution given with

$$\mu_p^{(\boldsymbol{\alpha})} = \big(\mathrm{I} - \mathrm{M}\big)^{-1} \boldsymbol{c}^{(\boldsymbol{\alpha})}. \tag{3.30}$$

Provided that the we have computed the inverse $\big(\mathrm{I} - \mathrm{M}\big)^{-1}$, which does not depend on $\boldsymbol{\alpha}$, the cross-moment is completely determined by the term $\boldsymbol{c}^{(\boldsymbol{\alpha})}$, which depends on all cross-moments of the order lower than $\boldsymbol{\alpha}$ and can be computed using (3.27). In the following sections, we derive $\boldsymbol{c}^{(\boldsymbol{\alpha})}$ for scalar random variables up to the second order, and retrieve the previous results for the first and second order moments [2], [10] as a special case of the equation (3.30).

3.3. First order moments computation of SCFG

In the case of the first order moments $\boldsymbol{\alpha} = (1)$ and the expression (3.9) reduces to the expectation of $\boldsymbol{X}_i$,

$$\mu_{p,\boldsymbol{X}_i}^{(1)} = \sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) \boldsymbol{X}_i(\boldsymbol{\pi}). \tag{3.31}$$

The moment vector, $\mu_p^{(\boldsymbol{\alpha})} = \big[\mu_{p,\boldsymbol{X}_1}^{(1)}, \ldots, \mu_{p,\boldsymbol{X}_{|\mathcal{N}|}}^{(1)}\big]$, is computed as in the equation (3.30),

$$\mu_p^{(1)} = \big(\mathrm{I} - \mathrm{M}\big)^{-1} \boldsymbol{c}^{(1)}, \tag{3.32}$$

where $c^{(1)} = \big[c_1^{(1)}, \ldots, c_{|\mathcal{N}|}^{(1)}\big]^T$. The first and second sum in the expression (3.27) for $c_i^{(\boldsymbol{\alpha})}$ reduce to zero and $c_i^{(1)} = \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(1,0)$, or, after the use of the expression (3.16) for $Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta})$,

$$c_i^{(1)} = \sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \cdot \boldsymbol{Y}\big(A_i \to B_{i,j}\big). \tag{3.33}$$

Let $\pi_1 \cdots \pi_N$ be a derivation starting at the start symbol A_1 and ending with a string $\boldsymbol{u} \in \Sigma^*$. If we set $\boldsymbol{Y}\big(A_i \to B_{i,j}\big) = 1$, according to (4.5), we have $\boldsymbol{X}_1(\pi_1 \cdots \pi_N) = \sum_{n=1}^{N} \boldsymbol{Y}\big(\pi_n\big) = N$, i.e. $\boldsymbol{X}_1$ is the length of the derivation. According to the expression (3.31), the moment $\mu_{p,\boldsymbol{X}_1}^{(1)}$ is the expected derivation length which agrees with [2] and [10].

Similarly, if we set $\boldsymbol{Y}\big(A_i \to B_{i,j}\big) = \sum_{n=1}^{|\Sigma|} t_n(i,j)$, where $t_n(i,j)$ denotes the number of terminals in the string $B_{i,j}$, the variable $\boldsymbol{X}_1(\pi_1 \cdots \pi_N)$ reduces to the length of the word derived from $\pi_1 \cdots \pi_N$. In this case, the moment $\mu_{p,\boldsymbol{X}_1}^{(1)}$ reduces to the expected string length and the formula (3.33) reduces to the result from [2].

3.4. Second order moments computation of SCFG

The formula for the second order moments is somewhat more complicated. In the case when $\boldsymbol{\alpha} = (2)$, $c_i^{(\boldsymbol{\alpha})}$ is reduced to:

(3.34)
$$\begin{aligned} c_i^{(2)} = &\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<2}} \binom{2}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} + \\ &\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\gamma_1+\cdots+\gamma_{|R|}=2 \\ \gamma_1,\ldots,\gamma_{|R|}<2}} H_{i,j}\big(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|R|}\big) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}\big(2,0\big) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}\big(2,1\big). \end{aligned}$$

The first sum in the previous expression can be transformed to:

$$\begin{aligned} &\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(i,j)}=2 \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}<2}} \binom{2}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(i,j)}} \prod_{l=1}^{r_n(i,j)} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} = \\ &\qquad \sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \sum_{n=1}^{|\mathcal{N}|} r_n(i,j)\big(r_n(i,j)-1\big) \cdot \big(\mu_{p,\boldsymbol{X}_n}^{(1)}\big)^2. \end{aligned} \tag{3.35}$$

To compute the second sum we introduce $H_{i,j}^{(a,b)}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|})$, which is $H_{i,j}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ with $\boldsymbol{\gamma}_a=\boldsymbol{\gamma}_b=1$ and with all other $\boldsymbol{\gamma}$-s equals to zero. We have:

$$H_{i,j}^{(a,b)}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = 2 \sum_{\delta_1+\dots+\delta_{r_a(i,j)}=\boldsymbol{\gamma}_a} \binom{\boldsymbol{\gamma}_a}{\boldsymbol{\delta}_1,\dots,\boldsymbol{\delta}_{r_a(i,j)}} \prod_{l=1}^{r_a(i,j)} \mu_{p,\boldsymbol{X}_k}^{(\boldsymbol{\delta}_l)}$$
$$\sum_{\delta_1+\dots+\delta_{r_b(i,j)}=\boldsymbol{\gamma}_b} \binom{\boldsymbol{\gamma}_b}{\boldsymbol{\delta}_1,\dots,\boldsymbol{\delta}_{r_b(i,j)}} \prod_{l=1}^{r_b(i,j)} \mu_{p,\boldsymbol{X}_a}^{(\boldsymbol{\delta}_l)} \cdot \prod_{\substack{k=1\\k\neq a,b}}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\dots+\boldsymbol{\delta}_{r_k(i,j)}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\dots,\boldsymbol{\delta}_{r_k(i,j)}} \prod_{l=1}^{r_k(i,j)} \mu_{p,\boldsymbol{X}_b}^{(\boldsymbol{\delta}_l)}, \tag{3.36}$$

and

$$H_{i,j}^{(a,b)}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = 2\cdot \sum_{c=1}^{r_a(i,j)} \mu_{p,\boldsymbol{X}_k}^{(1)} \cdot \sum_{d=1}^{r_b(i,j)} \mu_{p,\boldsymbol{X}_k}^{(1)} = 2\cdot r_a(i,j)\cdot r_b(i,j)\cdot \mu_{p,\boldsymbol{X}_a}^{(1)}\mu_{p,\boldsymbol{X}_b}^{(1)}. \tag{3.37}$$

By substitution of the second sum in (3.34),

$$\begin{aligned}
&\sum_{j=1}^{|R_i|} p(A_i\to B_{i,j}) \sum_{\substack{\gamma_1+\dots+\gamma_{|\mathcal{N}|}\\=2\\ \gamma_1,\dots,\gamma_{|\mathcal{N}|}<2}} H_{i,j}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|}) \\
&\quad= \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|}\sum_{b=a+1}^{|\mathcal{N}|} H_{i,j}^{(a,b)}(\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_{|\mathcal{N}|}) \\
&\quad= 2\cdot \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|}\sum_{b=a+1}^{|\mathcal{N}|} r_a(i,j)\cdot r_b(i,j)\cdot \mu_{p,\boldsymbol{X}_a}^{(1)}\mu_{p,\boldsymbol{X}_b}^{(1)} \\
&= \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|}\sum_{b=1}^{|\mathcal{N}|} r_a(i,j)\cdot r_b(i,j)\cdot \mu_{p,\boldsymbol{X}_a}^{(1)}\mu_{p,\boldsymbol{X}_b}^{(1)} - \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(i,j)^2 \big(\mu_{p,\boldsymbol{X}_n}^{(1)}\big)^2.
\end{aligned} \tag{3.38}$$

Now, (3.34) reduces to

$$c_i^{(2)} = CR_i + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,0) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,1), \tag{3.39}$$

where

$$CR_i = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|}\sum_{b=1}^{|\mathcal{N}|} r_a(i,j)\cdot r_b(i,j)\cdot \mu_{p,\boldsymbol{X}_a}^{(1)}\mu_{p,\boldsymbol{X}_b}^{(1)} - \sum_{j=1}^{|\mathcal{R}_i|} p(A_i\to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(i,j)\big(\mu_{p,\boldsymbol{X}_n}^{(1)}\big)^2 \tag{3.40}$$

and

$$Q_{i,j}(2,0) = p\big(A_i\to B_{i,j}\big)\cdot Y\big(A_i\to B_{i,j}\big)^2, \tag{3.41}$$

$$\begin{aligned}
Q_{i,j}(2,1) &= 2\cdot p\big(A_i\to B_{i,j}\big)\cdot Y\big(A_i\to B_{i,j}\big) \sum_{n=1}^{|\mathcal{N}|}\sum_{a=1}^{r_n(i,j)} \mu_{p,\boldsymbol{X}_n}^{(1)} \\
&= 2\cdot p\big(A_i\to B_{i,j}\big)\cdot Y\big(A_i\to B_{i,j}\big) \sum_{n=1}^{|\mathcal{N}|} r_n(i,j)\mu_{p,\boldsymbol{X}_n}^{(1)}.
\end{aligned} \tag{3.42}$$

If we set $Y\big(A_i \to B_{i,j}\big) = 1$ for all $A_i \to B_{i,j} \in \mathcal{R}$, $\boldsymbol{X}_1$ becomes derivation length. The formula for computing the second order moments of derivation length is given in [10] and it can be derived from the equation (3.39), since

$$\begin{aligned}\textstyle\sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,0) &= \textstyle\sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \cdot Y\big(A_i \to B_{i,j}\big)^2 = \\ &= \textstyle\sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) = 1,\end{aligned} \tag{3.43}$$

$$\begin{aligned}\textstyle\sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,1) &= 2 \cdot \textstyle\sum_{n=1}^{|\mathcal{N}|} \Big(\sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \cdot r_n(i,j)\Big)\mu^{(1)}_{p,\boldsymbol{X}_n} \\ &= 2 \cdot \textstyle\sum_{n=1}^{|\mathcal{N}|} e_{i,n}\mu^{(1)}_{p,\boldsymbol{X}_n} = 2 \cdot \mu^{(1)}_{p,\boldsymbol{X}_n} - 2,\end{aligned} \tag{3.44}$$

where the last equation follows from (3.30), and

$$c_i^{(2)} = CR_i + 2 \cdot \mu^{(1)}_{p,\boldsymbol{X}_n} - 1. \tag{3.45}$$

Finally, by substituting (3.45) in (3.30), we obtain

$$\mathrm{m}^{(\boldsymbol{\alpha})}\{\boldsymbol{X}\} = \big(\mathrm{I} - \mathrm{M}\big)^{-1} \cdot \Big(\mathrm{CR}_i + 2 \cdot \mathrm{m}_1 - \mathbf{1}\Big), \tag{3.46}$$

where $\mathrm{CR}_i = \big[CR_1, \ldots, CR_{|\mathcal{N}|}\big]$ and $\mathbf{1} = \big[1, \ldots, 1\big]$, in agreement with [10].

4. Conditional cross-moments computation of SCFG

In this section we consider the computation of conditional SCFG cross moments. We derive two equivalent versions of the algorithm for the computation, based on inside algorithm. The first one is obtained in the same manner as in Section 3 via conditional moment-generating function. In the second version we use dynamic programming over the binomial semiring, which relates the algorithm to the previously developed special cases by Hwa [11], [5], for the first order moments, and by Li and Eisner [16] for higher order cross-moments.

4.1. Conditional cross-moments and conditional moment-generating function of SCFG

Let $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ be reduced SCFG and let $\Omega_i(\boldsymbol{u})$ be a set of derivations starting at $A_i \in \mathcal{N}$ and finishing at $\boldsymbol{u}$. Let $\boldsymbol{X}_i = \big[X_{i,1}, \ldots, X_{i,D}\big]^T : \Omega_i(\boldsymbol{u}) \to \mathbb{R}^D$, where $\boldsymbol{u} \in \Sigma^*$, be random variables distributed according to the p_i, which are restrictions of p to $\mathcal{R}_i$ ($p_i(\boldsymbol{\pi}) = p(\boldsymbol{\pi})$), for $1 \leqslant i \leqslant |\mathcal{N}|$. The *i-th conditional cross-moment of an order* $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_D)$ of $\boldsymbol{X}_i$ conditioned on $\boldsymbol{u}$ is defined with

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) \cdot X_{i,1}(\boldsymbol{\pi})^{\nu_1} \cdots X_{i,D}(\boldsymbol{\pi})^{\nu_d} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\ \boldsymbol{X}_i(\boldsymbol{\pi})^{\boldsymbol{\nu}}. \tag{4.1}$$

The *i-th conditional moment-generating function (MGF)*, conditioned on $\boldsymbol{u}$, $M_{p,\boldsymbol{X}_i|\boldsymbol{u}} : \mathbb{R}^D \to \mathbb{R}$, is defined as:

$$M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})e^{\boldsymbol{t}^T\boldsymbol{X}_i(\boldsymbol{\pi})}, \tag{4.2}$$

for all $\boldsymbol{t} \in \mathbb{R}^D$, and the cross-moments can be retrieved from the *MGF* by differentiating:

$$(4.3) \qquad \mu^{(\boldsymbol{\nu})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \frac{\partial^{|\boldsymbol{\nu}|} M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t})}{\partial^{\nu_1} t_1 \dots \partial^{\nu_D} t_d}\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \mathcal{D}_{\boldsymbol{\nu}}\Big\{M_{p,\boldsymbol{X}_i|\boldsymbol{u}}\Big\} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\ \boldsymbol{X}_i(\boldsymbol{\pi})^{\boldsymbol{\nu}}.$$

and

$$(4.4) \qquad \mu^{(0)}_{p,\boldsymbol{X}_i|=} \mathcal{D}_{\boldsymbol{0}}\{M_{p,\boldsymbol{X}_i}\} = \Big(\sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T\boldsymbol{X}(\boldsymbol{\pi})}\Big)\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi}) = 1.$$

Let $\boldsymbol{X}_i : \Omega_i(\boldsymbol{u}) \to \mathbb{R}^D$ be represented as the sum of random vectors $\boldsymbol{Y} : \mathcal{R} \to \mathbb{R}^D$:

$$(4.5) \qquad \boldsymbol{X}_i(\pi_1 \cdots \pi_N) = \boldsymbol{Y}\big(\pi_1\big) + \cdots + \boldsymbol{Y}\big(\pi_N\big),$$

for all $\pi_1 \cdots \pi_N \in \Omega_i$.

Then, for G we can construct the conditional MGF grammar $\widetilde{G} = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\big)$ with a weight function which takes values from the semiring of $\boldsymbol{\nu}$ continuous functions $w : \mathcal{R} \to C_{\boldsymbol{\nu}}$, defined as:

$$(4.6) \qquad w(\pi) = p(\pi) e^{\boldsymbol{t}^T\boldsymbol{Y}(\pi)},$$

for all $\pi \in \mathcal{R}$.

A derivation $\boldsymbol{\pi} = \pi_1 \cdots \pi_N$ in G with the weight function $p(\boldsymbol{\pi}) = p(\pi_1) \cdots p(\pi_N)$ is a derivation in $\widetilde{G}$, with the weight function

$$(4.7) \quad w(\boldsymbol{\pi}) = w(\pi_1) \cdots w(\pi_N) = p(\pi_1) e^{\boldsymbol{t}^T\boldsymbol{Y}(\pi_1)} \cdots p(\pi_N) e^{\boldsymbol{t}^T\boldsymbol{Y}(\pi_N)} = p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T\boldsymbol{X}_i(\boldsymbol{\pi})}.$$

The i-th conditional MGF of the vector random variable $\boldsymbol{X}_i$ can now be computed as a sum of derivations in $\widetilde{G}$:

$$(4.8) \qquad M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi}\in\Omega(\boldsymbol{u})} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T\boldsymbol{X}_i(\boldsymbol{\pi})} = \sum_{\boldsymbol{\pi}\in\Omega(\boldsymbol{u})} w(\boldsymbol{\pi}).$$

In this manner, the computation of conditional *MGF* can be performed using the inside algorithm [8] over the semiring of $\boldsymbol{\nu}$-continuous functions at zero, which is done in the following section.

4.2. Conditional cross-moments computation of SCFG

Let $\widetilde{G} = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\big)$ be a weighted context-free grammar over a commutative semiring $\big(\mathbb{K}, +, \cdot, 1, 0\big)$, and $\Omega_i(\boldsymbol{u})$ be a set of all derivations which derive $\boldsymbol{u} \in \Sigma^*$ starting from a nonterminal A_i. The i-th *inside weight* of the weighted grammar $\widetilde{G}$ is the function $\sigma_i : \Sigma^* \to \mathbb{K}$, defined as the sum of weights of all derivations starting from $\Omega_i(\boldsymbol{u})$:

$$\sigma_i(\boldsymbol{u}) = \sum_{\boldsymbol{\pi}\in\Omega_i(\boldsymbol{u})} w(\boldsymbol{\pi}), \tag{4.9}$$

for $1 \leqslant i \leqslant |\mathcal{R}|$ and $\boldsymbol{u} \in \Sigma^*$. Let $A_i \to B_{i,j} \in \mathcal{R}$ and

$$B_{i,j} = \boldsymbol{v}_1 A_{i_1} \boldsymbol{v}_2 A_{i_2} \cdots \boldsymbol{v}_k A_{i_k} v_{k+1}, \tag{4.10}$$

where $\boldsymbol{v}_i \in \Sigma^*$ and $A_{i_n} \in \mathcal{N}$. For the cycle-free reduced grammars the inside weight can be computed using the *inside algorithm* [8] and [24] which, after recursive application of

$$\sigma_i(\boldsymbol{u}) = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma^* \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{v}_k\boldsymbol{u}_k\boldsymbol{v}_{k+1}}} w(A_i \to B_{i,j}) \cdot \prod_{j=1}^{k} \sigma_{i_j}(\boldsymbol{u}_j), \tag{4.11}$$

ends with the equation in which only rules $A_i \to \boldsymbol{u}, \boldsymbol{u} \in \Sigma^*$ appear on the right-hand side:

$$\sigma_i(\boldsymbol{u}) = w(A_i \to \boldsymbol{u}). \tag{4.12}$$

If $\widetilde{G}$ is the moment-generating grammar for G as defined in Section 4.1, the value of the the i-th inside weight corresponds to the i-th conditional moment-generating function,

$$\sigma_i(\boldsymbol{u}) = M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) \quad \text{and} \quad \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \mathcal{D}_{\boldsymbol{\alpha}}\{\sigma_i(\boldsymbol{u})\}, \tag{4.13}$$

for all $1 \leqslant i \leqslant |\mathcal{N}|$. Thus, an efficient computation procedure of the higher order cross-moments can be obtained by applying the generalized Leibniz's formula (2.1) to (4.11), which leads us to the recursive equations:

(4.14)
$$\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} \sum_{\beta\leqslant\alpha} \binom{\alpha}{\beta} p(A_i \to B_{i,j}) \cdot \boldsymbol{Y}(A_i \to B_{i,j})^{\boldsymbol{\alpha}-\boldsymbol{\beta}}$$
$$\times \sum_{\boldsymbol{\gamma}_1+\cdots\gamma_k=\beta} \prod_{j=1}^{k} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_k} \mu^{(\gamma_j)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u_j}}$$

with the base case

$$\mu^{(\gamma)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p(A_i \to \boldsymbol{u}) \cdot \boldsymbol{Y}(A_i \to \boldsymbol{u})^{\gamma}. \tag{4.15}$$

Note that the recursive algorithm (4.14)-(4.15) can always be implemented in the iterative manner using some of the procedures considered in [8].

The equations (4.14)-(4.15) can also be expressed in semiring dynamic programming form, as shown in the section below.

4.3. Conditional SCFG cross-moments computation using binomial semiring dynamic programming

Let us introduce the mapping $\mathcal{B}^{(\boldsymbol{\nu})} : C_{\boldsymbol{\nu}} \to \mathbb{R}^{|A_{\boldsymbol{\nu}}|}$, which associates an ordered tuple to any $f \in C_{\boldsymbol{\nu}}$:

$$\mathcal{B}^{(\boldsymbol{\nu})}(f) = \left(\mathcal{D}_{\boldsymbol{\alpha}}(f)\right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}. \tag{4.16}$$

In accordance to Leibniz's formulae we obtain

$$\mathcal{B}^{(\boldsymbol{\nu})}(f+g) = \mathcal{B}^{(\boldsymbol{\nu})}(f) \oplus \mathcal{B}^{(\boldsymbol{\nu})}(g), \tag{4.17}$$

$$\mathcal{B}^{(\boldsymbol{\nu})}(f \cdot g) = \mathcal{B}^{(\boldsymbol{\nu})}(f) \otimes \mathcal{B}^{(\boldsymbol{\nu})}(g), \tag{4.18}$$

where the $\oplus$ and $\otimes$ are defined with

$$u \oplus v = \left(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})}\right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}, \tag{4.19}$$

$$u \otimes v \quad = \left(\textstyle\sum_{\boldsymbol{\beta} \leqslant \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} u^{(\boldsymbol{\beta})} \cdot v^{(\boldsymbol{\alpha}-\boldsymbol{\beta})} \right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}, \tag{4.20}$$

for all $u, v \in \mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|}$. Therefore, the mapping $\mathcal{B}^{(\boldsymbol{\nu})}$ maps the semiring of $\boldsymbol{\nu}$ continuous functions in the binomial semiring of an order $\boldsymbol{\nu}$ [23], which is defined as the tuple $(\mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, where the identities for $\oplus$ and $\otimes$ are respectively given with

$$\begin{aligned} \mathbf{0} \quad &= (\underbrace{0, 0, \ldots, 0}_{|\mathcal{A}_{\boldsymbol{\nu}}| \text{ times}}) \\ \mathbf{1} \quad &= (1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_{\boldsymbol{\nu}}|-1 \text{ times}}). \end{aligned}$$

By using of $\mathcal{B}^{(\boldsymbol{\nu})}$, all the cross moments can be represented as an order tuple

$$\mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_i\big(\boldsymbol{u}\big)\big\} = \left(\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} \right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}, \tag{4.21}$$

where $\sigma_i\big(\boldsymbol{u}\big)$ stands for the inside weight. In this way, the equations for the cross moments computation can be represented in the binomial semiring dynamic form

$$\mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_i(\boldsymbol{u})\big\} = \bigoplus_{j=1}^{|\mathcal{R}_i|} \bigoplus_{\substack{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k \in \Sigma^* \\ \boldsymbol{u} = \boldsymbol{v}_1 \boldsymbol{u}_1 \boldsymbol{v}_2 \cdots \boldsymbol{v}_k \boldsymbol{u}_k \boldsymbol{v}_{k+1}}} w(A_i \to B_{i,j}) \otimes \bigotimes_{j=1}^{k} \mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_{i_j}(\boldsymbol{u}_j)\big\}, \tag{4.22}$$

with the base case

$$\mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_i\big(\boldsymbol{u}\big)\big\} = \mathcal{B}^{(\boldsymbol{\nu})}\big\{w\big(A_i \to \boldsymbol{u}\big)\big\}. \tag{4.23}$$

The algorithm given by equations (4.14)-(4.15) or, equivalently, (4.22)-(4.23) can be considered as a generalization of the algorithms by Li and Eisner [16] for the cross-moments of order $\boldsymbol{\alpha} = (1,1)$ and by Hwa [11] for the cross-moments of order $\boldsymbol{\alpha} = (1)$. Li and Eisner introduced the second order entropy semiring [13], which is the binomial semiring of the order $(1,1)$, and ran the inside algorithm on it. The algorithm for the moments of order $\boldsymbol{\alpha} = (1)$ is provided by Hwa [11], where conditional entropy is considered. As noted in [5], Hwa's algorithm can be obtained by running the inside algorithm over the first order entropy semiring [7], [13], [12], [14] which is the binomial semiring of the order (1). Hwa's algorithm is considered in the following subsection.

4.4. First order conditional moments computation of SCFG

In the case of first order conditional moments $\boldsymbol{\alpha} = (1)$, the i-th conditional cross-moment (4.1) is the expectation of $\boldsymbol{X}_i$,

$$\mu^{(1)}_{p,\boldsymbol{X}_1|\boldsymbol{u}} = \sum_{\boldsymbol{\pi}\in\Omega_i(\boldsymbol{u})} p(\boldsymbol{\pi})\boldsymbol{X}_i(\boldsymbol{\pi}). \tag{4.24}$$

In this case, the recursive equations (4.14)-(4.15) reduce to

$$\mu^{(0)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\dots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} p\big(A_i \to B_{i,j}\big) \cdot \prod_{j=1}^{k} \mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u_j}}, \tag{4.25}$$

(4.26)

$$\mu^{(1)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\dots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} p\big(A_i \to B_{i,j}\big) \cdot \boldsymbol{Y}\big(A_i \to B_{i,j}\big) \cdot \prod_{j=1}^{k} \mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u_j}} + p\big(A_i \to B_{i,j}\big) \cdot \sum_{n=1}^{k} \mu^{(1)}_{p,\boldsymbol{X}_{i_n}|\boldsymbol{u_n}} \prod_{\substack{j=1\\ j\neq n}}^{k} \mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u_j}},$$

with the base case:

$$\mu^{(0)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p\big(A_i \to \boldsymbol{u}\big), \qquad \mu^{(1)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p\big(A_i \to \boldsymbol{u}\big) \cdot Y\big(A_i \to \boldsymbol{u}\big). \tag{4.27}$$

In [11], Hwa considered the conditional entropy of the grammar given in Chomsky form for which $B_{i,j} = v_1 A_{i_1} v_2 A_{i_2} v_3$ and v_1, v_2, v_3 are equal to the empty string. The conditional entropy is obtained as the moment $\mu^{(1)}_{p,\boldsymbol{X}_1|\boldsymbol{u}}$, where $\boldsymbol{X}_1(\boldsymbol{\pi}) = -\log p(\boldsymbol{\pi})$, for all $\boldsymbol{\pi} \in \Omega_1$, and Hwa's algorithm can be retrieved by imposing Chomsky form condition in (4.14)-(4.15), with $\boldsymbol{Y}(\pi_i) = -\log p(\pi_i)$.

5. Conclusion

In this paper we considered the problem of computing the cross-moments and the conditional cross-moments of a vector variable represented by a stochastic context-free grammar. We proposed new algorithms, derived by differentiation of the recursive equations for the moment-generating function [22], which are obtained from the algorithms for computing the partition function of a SCFG [18] for the cross-moments and with the inside algorithm [15], [8] for the conditional cross-moments. In this way, we obtained the algorithms which can be considered as a generalization of the previously developed formulas for moments [10], [26] and conditional cross-moments [11], [16].

The computation of cross-moments may be demanding and often infeasible. The proposed method for its solution via the computation of moment-generating function turned out to be very elegant and powerful. In the future, we hope that this idea can successfully be reused in the theory of formal languages for the computation of cross moments of string and tree automata [3], [6].

REFERENCES

1. Bonnie Berger. The fourth moment method. *SIAM Journal on Computing*, 26(4):1188–1207, 1997.
2. T. L. Booth and R. A. Thompson. Applying probability measures to abstract languages. *IEEE Trans. Comput.*, 22:442–450, May 1973.
3. Jorge Calera-Rubio and Rafael C. Carrasco. Computing the relative entropy between regular tree languages. *Information Processing Letters*, 68(6):283 – 289, 1998.
4. Zhiyi Chi. Statistical properties of probabilistic context-free grammars. *Comput. Linguist.*, 25(1):131–160, March 1999.
5. Shay B. Cohen, Robert J. Simmomns, and Noah A. Smith. Products of weighted logic programs. *Theory and Practice of Logic Programming*, 11:263–296, 2011.
6. Corinna Cortes and Mehryar Mohri. Distribution kernels based on moments of counts. In *In proceedings of the twenty-first international Conference on Machine Learning (ICML 2004)*, 2004.
7. Corinna Cortes, Mehryar Mohri, Ashish Rastogi, and Michael Riley. On the computation of the relative entropy of probabilistic automata. *Int. J. Found. Comput. Sci.*, 19(1):219–242, 2008.
8. Joshua Goodman. Semiring parsing. *Comput. Linguist.*, 25(4):573–605, 1999.
9. Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, New York, NY, USA, 1985.
10. Sandra E. Hutchins. Moments of string and derivation lengths of stochastic context-free grammars. *Information Sciences*, 4(2):179 – 191, 1972.
11. Rebecca Hwa. Sample selection for statistical grammar induction. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, pages 45–52, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

12. Velimir M Ilić, Dejan I Mančev, Branimir T Todorović, and Miomir S Stanković. Gradient computation in linear-chain conditional random fields using the entropy message passing algorithm. *Pattern Recognition Letters*, 33(13):1776–1784, 2012.

13. Velimir M. Ilic, Miomir S. Stankovic, and Branimir T. Todorovic. Entropy message passing. *IEEE Transactions on Information Theory*, 57(1):219–242, 2011.

14. Velimir M. Ilić, Miomir S. Stanković, and Branimir T. Todorović. Entropy semiring forward-backward algorithm for HMM entropy computation. *Transactions on Advanced Research*, 8(2):8–15, 2012.

15. K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 4(1):35 – 56, 1990.

16. Zhifei Li and Jason Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 40–51, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

17. Mark-Jan Nederhof and Giorgio Satta. Computation of distances for regular and context-free probabilistic languages. *Theor. Comput. Sci.*, 395(2-3):235–254, April 2008.

18. Mark-Jan Nederhof and Giorgio Satta. Computing partition functions of pcfgs. *Research on Language and Computation*, 6:139–162, 2008. 10.1007/s11168-008-9052-8.

19. Mark-Jan Nederhof and Giorgio Satta. Probabilistic parsing. In *New Developments in Formal Languages and Applications*, volume 113 of *Studies in Computational Intelligence*, pages 229–258. Springer Berlin / Heidelberg, 2008.

20. M.H. Protter. *Basic Elements of Real Analysis*. S.Axler and others. Springer, 1998.

21. Xavier Saint Raymond. *Elementary Introduction to the Theory of Pseudodierential Operators (Studies in Advanced Mathematics)*. CRC Press, Boca Raton, 1991.

22. Lund R.B. Elementary probability theory with stochastic processes and an introduction to mathematical finance (4th ed.). *The American Statistician*, 58:173–174, May 2004.

23. Velimir M. Ilić; Miomir S. Stanković and Branimir T. Todorović. Computation of cross-moments using message passing over factor graphs. *Adv. Math. Commun.*, 6(3):363–384, 2012.

24. Frdric Tendeau. Computing abstract decorations of parse forests using dynamic programming and algebraic power series. *Theoretical Computer Science*, 199(1-2):145 – 166, 1998.

25. A. B. Thaheema and A. Laradjia. Classroom note: A generalization of leibniz rule for higher derivatives. *International Journal of Mathematical Education in Science and Technology*, 34(6):905–907, 2001.

26. C. S. Wetherell. Probabilistic languages: A review and some open questions. *ACM Comput. Surv.*, 12:361–379, December 1980.

Velimir Ilić
Mathematical Institute of the Serbian Academy of Sciences and Arts
Kneza Mihaila 36, 11001 Beograd, p.p. 367, Serbia
velimir.ilic@gmail.com

Miroslav D. Ćirić
University of Niš, Faculty of Science and Mathematics
Department of Computer Science
Višegradska 33, 18000 Niš, Serbia
mciric@pmf.ni.ac.rs

Miomir S. Stanković
University of Niš, Faculty of Occupational Safety Niš
Čarnojevićeva 10A, 18000 Niš, Serbia
miomir.stankovic@gmail.com