

Commutative Lambek Grammars

Tikhon Pshenitsyn

*Department of Mathematical Logic and Theory of Algorithms,
Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow,
119991, Russian Federation.

*Department of Mathematical Logic, Steklov Mathematical Institute of
Russian Academy of Sciences, 8 Gubkina St., Moscow, 119991, Russian
Federation.

Corresponding author(s). E-mail(s): ptihon@yandex.ru;

Abstract

Lambek categorial grammars is a class of formal grammars based on the Lambek calculus. Pentus proved in 1993 that they generate exactly the class of context-free languages without the empty word. In this paper, we study categorial grammars based on the Lambek calculus with the permutation rule LP. Of particular interest is the product-free fragment of LP called the Lambek-van Benthem calculus LBC. Buszkowski in his 1984 paper conjectured that grammars based on the Lambek-van Benthem calculus (LBC-grammars for short) generate exactly permutation closures of context-free languages. In this paper, we disprove this conjecture by presenting a language generated by an LBC-grammar that is not a permutation closure of any context-free language. Firstly, we introduce an ad-hoc modification of vector addition systems called linearly-restricted branching vector addition systems with states and additional memory (LRBVASSAMs for short) and prove that the latter are equivalent to LBC-grammars. Then we construct an LRBVASSAM that generates a non-semilinear set and thus disprove Buszkowski's conjecture.

Since Buszkowski's conjecture is false, not so much is known about the languages generated by LBC-grammars or by LP-grammars. The equivalence of LRBVASSAMs and LBC-grammars allows us to establish a number of their properties. We show that LP-grammars generate the same class of languages as LBC-grammars; that is, removing product from LP does not decrease expressive power of corresponding categorial grammars. We also prove that this class of languages is closed under union, intersection, concatenation, and Kleene plus.

Keywords: Lambek calculus, categorial grammar, formal language, vector addition system, branching vector addition system with states, semilinear set

1 Introduction

Lambek categorial grammars is a class of formal grammars based on the Lambek calculus L, which is a logic designed to model syntax of natural languages [1]. It is a substructural logic of the intuitionistic logic, namely, it is obtained from the latter by dropping the structural rules such as weakening, contraction, and permutation. In the Lambek calculus, formulas are built from variables (called *primitive formulas*) using left division \backslash , right division $/$, and product \bullet ; divisions are directed versions of linear logic implication. A Lambek categorial grammar consists of an assignment of a finite number of formulas to each symbol of the alphabet and of a distinguished formula S . Then, a word $a_1 \dots a_n$ belongs to the language generated by this grammar if and only if each symbol a_i can be replaced by a corresponding formula T_i in such a way that the sequent $T_1, \dots, T_n \Rightarrow S$ is derivable in the Lambek calculus. Therefore the grammar derivation mechanism relies on the notion of provability in L.

The famous result proved in [2, 3] states that Lambek categorial grammars generate exactly context-free languages without the empty word. This establishes a non-trivial equivalence of the type-logical approach and of the rule-based one. The statement has two directions. The first one proved by Gaifman in [2] says that any context-free grammar that does not generate the empty word can be converted into an equivalent Lambek categorial grammar. Gaifman's proof uses the Greibach normal form for context-free grammars, and it is quite straightforward. Conversely, Pentus proved in [3] that Lambek categorial grammars generate only context-free languages. Namely, Pentus showed how to transform a given Lambek categorial grammar into an equivalent context-free grammar. Although Pentus' construction is based on an intuitively simple idea, the proof of its correctness is non-trivial, it involves several delicate techniques including the free-group interpretation and the binary reduction lemma.

After Lambek's seminal work [1], numerous modifications and extensions of L have been introduced for different purposes. One of such modifications is the Lambek calculus with the permutation rule, which we also call the *commutative Lambek calculus*. It is obtained from the Lambek calculus by adding the commutativity postulate for product: $A \bullet B \Leftrightarrow B \bullet A$. The Lambek calculus with the permutation rule LP was initially studied by van Benthem in [4], and it has many applications in linguistics. In van Benthem's paper, the calculus is defined in the sequent form using only the undirected division operation (which is in fact the same operation as intuitionistic linear logic implication $A \multimap B$). In [5], this calculus is called the Lambek-van Benthem calculus LBC. Note that division is denoted by (A, B) in [4], but, in this paper, we shall always denote division as $A \backslash B$ following the notation of the Lambek calculus.

In [6], the following axiomatization for LBC is presented¹:

$$\frac{}{A \Rightarrow A} \quad \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, B, A, \Delta \Rightarrow C} \quad \frac{\Pi, A \Rightarrow B}{\Pi \Rightarrow A \backslash B} \quad \frac{\Gamma, B \Rightarrow C \quad \Pi \Rightarrow A}{\Gamma, \Pi, A \backslash B \Rightarrow C}$$

Here A, B, C are formulas and Π, Γ, Δ are sequences of formulas. It is required that Π is non-empty; this requirement is called *Lambek's restriction*, and it is motivated

¹Here, however, we use a slightly different notation.

by linguistic applications [7, Section 2.5]. The Lambek calculus with the permutation rule, which also includes product, is denoted by LP; it is introduced in Section 2.2. Note that LBC is simply the product-free fragment of LP, and LP conservatively extends LBC.

Given the Lambek calculus with the permutation rule (or its fragment LBC), one can define the class of categorial grammars based on LP in the same way as Lambek grammars are defined on the basis of L. Then one could ask what class of languages such grammars generate. We call grammars based on the commutative Lambek calculus LP-grammars.

In [5, 6], it is proved that categorial grammars based on the Lambek-van Benthem calculus generate all permutation closures of context-free languages. However, the question whether the converse holds or not has remained open. Buszkowski in his 1984 paper conjectures the following:

Conjecture 1 *Grammars based on the Lambek van-Benthem calculus LBC generate exactly permutation closures of context-free languages.*

Moreover, Buszkowski proves in [5] that grammars of order² less than or equal to 1 based on LBC generate exactly permutation closures of context-free languages. This might seem to be a partial solution to the problem justifying the conjecture.

To my best knowledge, the question whether Buszkowski’s conjecture is true has remained open. For instance, in [8, p. 230] the question of existence of a Pentus-like proof for LP-grammars is mentioned as being open. Stepan L. Kuznetsov [9] introduced this problem to me conjecturing that an example of an LP-grammar can be constructed such that it generates a language being not a permutation closure of a context-free language. Hence he claimed that Buszkowski’s conjecture is false.

In this paper, we disprove Buszkowski’s conjecture by showing that there is a grammar based on the Lambek-van Benthem calculus generating a language that is not the permutation closure of any context-free language. This confirms Kuznetsov’s conjecture. We do this by introducing a formalism equivalent to LP-grammars, which is called *linearly-restricted branching vector addition systems with states and additional memory (LRBVASSAM)*³. We prove that LRBVASSAMs generate exactly Parikh images of languages generated by LP-grammars. The definition of LRBVASSAM and the proof are inspired by our study of the relation between double-pushout hypergraph grammars and hypergraph Lambek grammars [10]. In the cited article, we deal with a general formalism extending the Lambek calculus to hypergraph structures and investigate expressivity of the corresponding class of categorial grammars. Nicely, the methods used for studying hypergraph Lambek grammars gave us useful enough insights for solving the problem concerning the expressive power of LP-grammars.

This paper is organized as follows:

- In Section 2, we introduce preliminary definitions, in particular, we define all the calculi of interest and recall their basic properties.

²An LBC-grammar $\langle S, \triangleright \rangle$ is of order less than or equal to 1 if and only if $a \triangleright T$ implies that T is of the form $p_1 \setminus (\dots \setminus (p_{k-1} \setminus (p_k \setminus q)) \dots)$ for some $k \in \mathbb{N}$ and some primitive formulas p_i, q .

³We apologize to the reader for such long abbreviations.

- In Section 3, we describe the results of this paper (the main one is Theorem 4, because it is the one that disproves the conjecture of Buszkowski).
- In Section 4, we define the notion of LRBVASSAM and prove that LBC-grammars are equivalent to LRBVASSAMs.
- In Section 5, we present a non-semilinear set generated by an LRBVASSAM.
- In Section 6, we use the results of Sections 4 and 5 to prove that there is a language generated by an LBC-grammar that is not a permutation closure of any context-free language. We prove the main theorems of the work in this section.
- In Section 7, we establish closure properties for the class of languages generated by LBC-grammars.
- In Section 8, we conclude.

2 Preliminaries

In this section, we clarify the notation used throughout the paper and define the calculi of interest along with corresponding categorial grammars.

2.1 Basic Notions

- We shall use the notion of a *multiset* in this paper. We are not going to provide a formal definition of this notion assuming that it is well known (see, e.g., [11]). Intuitively, it is “a collection of objects (called elements) in which elements may occur more than once” [11]. For example, $\{a, a, a, b, b, c\}$ is a multiset. Given two multisets A and B , their disjoint union $A \sqcup B$ is obtained as follows: if a occurs m times in A and n times in B , then a occurs $(m + n)$ times in $A \sqcup B$. E.g., $\{a, b, c\} \sqcup \{a, a, b\} = \{a, a, a, b, b, c\}$.
- Let Σ be a finite set called an *alphabet*. We assume that hereinafter the letter Σ is reserved for the alphabet. Let $|\Sigma| = \kappa \in \mathbb{N}$ be the size of the alphabet; the letter κ is also “global”, it is reserved for the cardinality of Σ throughout the remainder of the paper. The set Σ^* is the set of *words* over Σ including the empty word Λ ; besides, $\Sigma^\oplus = \Sigma^* \setminus \{\Lambda\}$. If $w \in \Sigma^*$ and $a \in \Sigma$, then let $|w|_a$ equal the number of occurrences of the symbol a in w (for example, $|aab|_a = 2$, $|aab|_b = 1$, $|aab|_c = 0$). The length $|w|$ of the word w is the total number of symbols in it.
- If $w = a_1 \dots a_n$, then the same word can be represented using commas as separators between symbols, namely, $w = a_1, \dots, a_n$.
- A *language* is any subset of Σ^* .
- By a^k we denote either the word $a \dots a$ or the multiset $\{a, \dots, a\}$ with a repeated k times (the meaning is always clear from the context).
- A *context-free grammar* is a triple $Gr = \langle N, P, S \rangle$, where N is a finite alphabet of nonterminal symbols ($N \cap \Sigma = \emptyset$), P is a set of productions, and $S \in N$. Each production is of the form $A \rightarrow \alpha$ where $A \in N$ is a nonterminal symbol and $\alpha \in (N \cup \Sigma)^*$ is a word. An application of the production $A \rightarrow \alpha$ is of the form $\eta A \theta \Rightarrow \eta \alpha \theta$ for $\eta, \theta \in (N \cup \Sigma)^*$ being some words. A word $w \in \Sigma^*$ is generated by Gr if and only if there is a sequence of applications of productions of the form $S \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_k = w$. The *language generated by Gr* is the set of all words generated by Gr . Such a language is called a *context-free language*.

- If $w = a_1 \dots a_n$ and $\sigma \in S_n$ is a permutation, then $\sigma(w) = a_{\sigma(1)} \dots a_{\sigma(n)}$.
- If L is a language, then its *permutation closure* L^{perm} consists of words of the form $\sigma(w)$ where $w \in L$ and $\sigma \in S_n$ is a permutation for $n = |w|$. Informally, we consider all possible words that can be obtained from those from L by permuting symbols in them.
- A language L is called *commutative* if $L^{perm} = L$.
- Hereinafter, let us fix an enumeration of symbols of the alphabet Σ , namely, let $\Sigma = \{a_1, \dots, a_\kappa\}$. Then the *Parikh image* of $w \in \Sigma^*$ is defined as $\Psi(w) = (|w|_{a_1}, \dots, |w|_{a_\kappa})$. Thus Ψ is a function that maps Σ^* to \mathbb{N}^κ . This definition is generalized to languages in an obvious way: $\Psi(L) = \{\Psi(w) \mid w \in L\}$. We shall also consider the *inverse Parikh image*:

$$\Psi^{-1}(V) = \{w \in \Sigma^* \mid \Psi(w) \in V\}.$$

- A subset of \mathbb{N}^κ is *semilinear* if and only if it is a finite union of linear sets, i.e. of sets of the form $\{v_0 + n_1v_1 + \dots + n_lv_l \mid n_1, \dots, n_l \in \mathbb{N}\}$ where v_0, \dots, v_l are fixed vectors from \mathbb{N}^κ .
- The size $|v|$ of a vector $v = (x_1, \dots, x_\kappa) \in \mathbb{N}^\kappa$ equals $x_1 + \dots + x_\kappa$. By e_i we denote the standard-basis vector $(0, \dots, 0, 1, 0, \dots, 0)$ where 1 stands at the i -th position.

The following obvious properties of the Parikh image shall be used later:

- $\Psi^{-1}(\Psi(L))$ equals the permutation closure L^{perm} of L .
- $\Psi(L) = \Psi(L^{perm})$.

The fundamental result relating context-free languages and semilinear sets is Parikh's theorem [12, 13], which states that, if L is a context-free language, then its Parikh image $\Psi(L)$ is a semilinear set, and vice versa, each semilinear set is a Parikh image of some context-free language. The following proposition easily follows from Parikh's theorem:

Proposition 1 *For any language L , its permutation closure L^{perm} is the permutation closure of a context-free language if and only if $\Psi(L)$ is semilinear.*

Proof If $L^{perm} = L_0^{perm}$ for some context-free language L_0 , then $\Psi(L) = \Psi(L^{perm}) = \Psi(L_0^{perm}) = \Psi(L_0)$ is semilinear according to Parikh's theorem. Conversely, if $\Psi(L)$ is semilinear, then there exists a context-free language L_0 such that $\Psi(L_0) = \Psi(L)$. This implies that $L_0^{perm} = \Psi^{-1}(\Psi(L_0)) = \Psi^{-1}(\Psi(L)) = L^{perm}$, hence L^{perm} is the permutation closure of the context-free language L_0 . \square

2.2 The Lambek Calculus With Permutation

In this section, we define the Lambek calculus with the permutation rule (or simply the *commutative Lambek calculus*) and its several variants in the Gentzen style.

- Let us fix a countable set of *primitive formulas* Pr . The set of *formulas* is defined as follows: $\text{Fm}_{\text{LP}} := \text{Pr} \mid \text{Fm}_{\text{LP}} \setminus \text{Fm}_{\text{LP}} \mid \text{Fm}_{\text{LP}} \bullet \text{Fm}_{\text{LP}}$.

- A *sequent* is a structure of the form $\Pi \Rightarrow A$ where Π is a finite non-empty multiset consisting of formulas and $A \in \text{Fm}_{\text{LP}}$. The non-emptiness requirement for Π is called *Lambek's restriction*. The multiset of formulas Π is called the *antecedent of the sequent*, and A is called the *succedent of the sequent*.
- If Γ and Δ are two multisets of formulas, then their union $\Gamma \sqcup \Delta$ shall be denoted using the comma: $\Gamma \sqcup \Delta = \Gamma, \Delta$. Besides, the notation Γ, A means $\Gamma \sqcup \{A\}$ for any multiset of formulas Γ and any formula A .
- The axioms and rules of LP are as follows:

$$\frac{}{A \Rightarrow A} \text{ (ax)} \quad \frac{\Gamma, B \Rightarrow C \quad \Pi \Rightarrow A}{\Gamma, \Pi, A \setminus B \Rightarrow C} (\setminus_L) \quad \frac{\Pi, A \Rightarrow B}{\Pi \Rightarrow A \setminus B} (\setminus_R)$$

$$\frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \bullet B \Rightarrow C} (\bullet_L) \quad \frac{\Pi \Rightarrow A \quad \Phi \Rightarrow B}{\Pi, \Phi \Rightarrow A \bullet B} (\bullet_R)$$

Hereinafter capital Latin letters A, B, C, \dots stand for formulas; capital Greek letters $\Gamma, \Delta, \Pi, \Phi, \dots$ stand for words of formulas. Besides, Π, Φ in the rules above must be non-empty according to Lambek's restriction⁴.

This completes the definition of the *commutative Lambek calculus* LP. The *Lambek-van Benthem calculus* is the product-free fragment of LP, i.e. its formulas are defined as follows: $\text{Fm}_{\text{LBC}} := \text{Pr} \mid \text{Fm}_{\text{LBC}} \setminus \text{Fm}_{\text{LBC}}$.

Remark 1 This definition differs from the presentation of LBC in Section 1. In Section 1, antecedents of sequents are defined as words of formulas rather than multisets. The following permutation rule introduced there says that the order of formulas in the antecedent can be freely changed:

$$\frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, B, A, \Delta \Rightarrow C}$$

By defining an antecedent of a sequent as a multiset we make the permutation rule implicit; this is more convenient for further reasonings. Clearly, the two ways of defining LP (using multisets and using words along with the permutation rule) are equivalent.

If Lambek's restriction is removed from the definition of LP, then we come up with the definition of the *commutative Lambek calculus allowing empty antecedents* denoted by L^*P . Besides, one can also add the constant symbol I to the language of L^*P and the following axiomatization for it:

$$\frac{\Pi \Rightarrow A}{\Pi, I \Rightarrow A} (\text{I}_L) \quad \frac{}{\Rightarrow I} (\text{I}_R)$$

This calculus is called the *tensor-implication logic* in [14]. We call it the *commutative Lambek calculus with the unit* and denote it as L_I^*P .

From now on, the set of formulas of a calculus \mathcal{K} of interest is denoted by $\text{Fm}_{\mathcal{K}}$. Note that $\text{Fm}_{\text{LP}} = \text{Fm}_{\text{L}^*\text{P}}$ since these two logics differ in structure of sequents but not in the set of formulas. Clearly, $\text{Fm}_{\text{LBC}} \subsetneq \text{Fm}_{\text{LP}} = \text{Fm}_{\text{L}^*\text{P}} \subsetneq \text{Fm}_{\text{L}_I^*\text{P}}$.

⁴Actually, it suffices to require that Π is non-empty in the rule (\setminus_R) ; then, one can prove that all antecedents of derivable sequents are non-empty.

2.3 Categorical Grammars

Now, let us define the notion of categorical grammars based on the calculi of interest.

Definition 1 Let \mathcal{K} be one of the calculi LP, LBC, L*P, L₁*P.

- A \mathcal{K} -grammar is a pair $G = \langle S, \triangleright \rangle$ where $S \in \text{Fm}_{\mathcal{K}}$ is a distinguished formula, and $\triangleright \subseteq \Sigma \times \text{Fm}_{\mathcal{K}}$ is a finite binary relation between symbols of the alphabet and formulas. In other words, the relation \triangleright assigns a finite number of formulas to each symbol of Σ .
- The language $L(G)$ generated by G is the set of words $a_1 \dots a_n \in \Sigma^*$ such that there exist formulas $T_1, \dots, T_n \in \text{Fm}_{\mathcal{K}}$, for which the following holds:
 1. $a_i \triangleright T_i$ ($i = 1, \dots, n$);
 2. $\mathcal{K} \vdash T_1, \dots, T_n \Rightarrow S$.

In particular, the empty word Λ belongs to $L(G)$ if and only if the sequent $\Rightarrow S$ with the empty antecedent is derivable in \mathcal{K} .

Note that the title of this paper *Commutative Lambek Grammars* does not refer to any particular kind of grammars defined above. The term *commutative Lambek grammar* is a generic notion that means a grammar based on any of the calculi considered in this paper.

Remark 2 Lambek's restriction implies that the empty word Λ does not belong to the language generated by an LP-grammar. However, it may belong to the language generated by an L*P-grammar or by an L₁*P-grammar.

Example 1 Given that $\Sigma = \{a, b\}$, consider the LP-grammar $G_0 = \langle S_0, \triangleright_0 \rangle$ where

- $S_0 = p \backslash p$;
- $a \triangleright_0 E_a = q \backslash (p \backslash p)$; $b \triangleright_0 E_b^1 = q$; $b \triangleright_0 E_b^2 = q \bullet q$.

Then $L(G_0) = \{a^n b^l \mid 0 < l \leq n \leq 2l\}^{perm}$. We leave the proof of this fact as an exercise. As a particular case, let us show, for example, that $aba \in L(G_0)$. To do this, replace the occurrences of a in aba by the formula E_a and the only occurrence of b by E_b^2 . Then the sequent $E_a, E_b^2, E_a \Rightarrow S_0$ is derivable in LP:

$$\frac{\frac{\frac{p \Rightarrow p \quad p \Rightarrow p}{p \backslash p, p \Rightarrow p} (\backslash_L) \quad p \Rightarrow p}{p \backslash p, p \backslash p, p \Rightarrow p} (\backslash_L) \quad q \Rightarrow q}{q \backslash (p \backslash p), q, p \backslash p, p \Rightarrow p} (\backslash_L) \quad q \Rightarrow q}{q \backslash (p \backslash p), q, q, q \backslash (p \backslash p), p \Rightarrow p} (\backslash_L) \quad q \Rightarrow q}{q \backslash (p \backslash p), q, q, q \backslash (p \backslash p) \Rightarrow p \backslash p} (\backslash_R) \quad q \Rightarrow q}{q \backslash (p \backslash p), q \bullet q, q \backslash (p \backslash p) \Rightarrow p \backslash p} (\bullet_L)$$

The grammar G_0 can be considered as an L*P-grammar; let us denote the L*P-grammar $\langle S_0, \triangleright_0 \rangle$ by G_0^* . Then $\Lambda \in L(G_0^*)$ since $L^*P \vdash \Rightarrow p \setminus p$:

$$\frac{p \Rightarrow p}{\Rightarrow p \setminus p} (\setminus_R)$$

As we mentioned in Section 1, in [5, 6], it is proved that LBC-grammars (and, consequently, LP-grammars) generate all permutation closures of context-free languages. Moreover, in [5], it is proved that LBC-grammars of order less than or equal to 1 generate exactly permutation closures of context-free languages.

2.4 Auxiliary Notions

Below we define notions and notation that shall be used from this moment forth.

- In the formula $C \setminus (B \setminus A)$, brackets shall be omitted: $C \setminus B \setminus A$. In general, a formula of the form $A_1 \setminus (\dots \setminus (A_{k-1} \setminus (A_k \setminus B)) \dots)$ shall be written as $A_1 \setminus \dots \setminus A_{k-1} \setminus A_k \setminus B$.
- Since product is associative, we shall omit brackets in the formulas like $A \bullet (B \bullet C)$ or $(A \bullet B) \bullet C$ and write $A \bullet B \bullet C$ instead.
- The formula $A \setminus B$ occurring in the rule applications of (\setminus_L) and (\setminus_R) is called *major* in these rule applications. Similarly, the formula $A \bullet B$ is called *major* in the rule applications of (\bullet_L) and (\bullet_R) .
- Let $A^{\bullet k}$ be a shorthand notation for $\underbrace{A \bullet \dots \bullet A}_{k \text{ times}}$. Formally, $A^{\bullet 1} = A$, and $A^{\bullet(k+1)} = (A^{\bullet k} \bullet A)$. (Recall that A^k is a shorthand notation for $\underbrace{A, \dots, A}_{k \text{ times}}$.)
- The *length* of formulas is defined as follows:
 1. $|p| = 1$ for $p \in \text{Pr}$;
 2. $|A \circ B| = |A| + |B| + 1$, $\circ \in \{\bullet, \setminus\}$.

Informally, the length is the number of symbols in a formula or in a sequent.

Definition 2 Given a \mathcal{K} -grammar G , we denote by $\text{Fm}(G)$ the set of all formulas occurring in G (including S). Formally,

$$\text{Fm}(G) = \{T \in \text{Fm}_{\mathcal{K}} \mid \exists a \in \Sigma (a \triangleright T)\} \cup \{S\}.$$

Clearly, $\text{Fm}(G)$ is finite since so is \triangleright .

The set $\text{SFm}^+(G)$ of *positive subformulas* of G and the set $\text{SFm}^-(G)$ of *negative subformulas* of G are defined as the least pair of sets satisfying the following requirements:

- If there exists $a \in \Sigma$ such that $a \triangleright T$, then $T \in \text{SFm}^-(G)$;
- $S \in \text{SFm}^+(G)$;
- If $B \setminus A \in \text{SFm}^+(G)$, then $A \in \text{SFm}^+(G)$ and $B \in \text{SFm}^-(G)$;
- If $B \setminus A \in \text{SFm}^-(G)$, then $A \in \text{SFm}^-(G)$ and $B \in \text{SFm}^+(G)$;
- If $A \bullet B \in \text{SFm}^+(G)$, then both A and B are in $\text{SFm}^+(G)$;
- If $A \bullet B \in \text{SFm}^-(G)$, then both A and B are in $\text{SFm}^-(G)$.

The set $\text{SFm}(G)$ of all subformulas of G is the union $\text{SFm}^+(G) \cup \text{SFm}^-(G)$.

The intuition behind the definitions of positive and negative subformulas is the following. If one wants to check that a word $a_1 \dots a_n$ is generated by a \mathcal{K} -grammar G , then replace each symbol a_i by a formula T_i such that $a_i \triangleright T_i$ and then check that $\mathcal{K} \vdash T_1, \dots, T_n \Rightarrow S$. If this sequent is derivable, fix its derivation and look at formulas occurring in it. Then formulas in the antecedents of the sequents that are present in this derivation are negative subformulas of G , while formulas occurring in the succedents of the sequents are positive ones.

2.5 Properties

In the remainder of the paper, several properties of the calculi defined above are used.

- The cut rule is admissible in each of the calculi LP, LBC, L*P, L_I*P:

$$\frac{\Pi \Rightarrow A \quad \Gamma, A \Rightarrow B}{\Gamma, \Pi \Rightarrow B} \text{ (cut)}$$

This means that if the sequents $\Pi \Rightarrow A$ and $\Gamma, A \Rightarrow B$ are derivable in LP (or in LBC / in L*P / in L_I*P), then so is $\Gamma, \Pi \Rightarrow B$. This is proved in [6, Chapter 7].

- As a consequence, the following rules are also admissible in each of these logics:

$$\frac{\Gamma, A \bullet B \Rightarrow C}{\Gamma, A, B \Rightarrow C} (\bullet_L^{-1}) \quad \frac{\Pi \Rightarrow A \setminus B}{\Pi, A \Rightarrow B} (\setminus_R^{-1})$$

Indeed, the first rule can be obtained as the application of the cut rule to the sequents $A, B \Rightarrow A \bullet B$ and $\Gamma, A \bullet B \Rightarrow C$; the second rule is the application of the cut rule to the sequents $\Pi \Rightarrow A \setminus B$ and $A \setminus B, A \Rightarrow B$.

- Let us say that formulas A and B are equivalent in a calculus \mathcal{K} (denoted by $\mathcal{K} \vdash A \Leftrightarrow B$) if $\mathcal{K} \vdash A \Rightarrow B$ and $\mathcal{K} \vdash B \Rightarrow A$. For \mathcal{K} being one of the calculi LP, LBC, L*P, L_I*P, if two formulas are equivalent, then they are interchangeable in any context. Interchangeability means the following:
 - Let C be a formula with a unique primitive subformula $x \in \text{Pr}$. By $C[x/D]$ we denote the substitution of D for x . If $\mathcal{K} \vdash A \Leftrightarrow B$, then $\mathcal{K} \vdash C[x/A] \Leftrightarrow C[x/B]$. (The proof is by induction on the length of C .)
 - Let C be a formula of \mathcal{K} and let Γ be a multiset of formulas of \mathcal{K} . If $A \Leftrightarrow B$ is derivable in \mathcal{K} , then the sequent $\Gamma, A, \Delta \Rightarrow C$ is derivable in \mathcal{K} if and only if so is $\Gamma, B, \Delta \Rightarrow C$. (This is proved by applying the cut rule.)
- Let $\langle S, \triangleright \rangle$ be a grammar based on one of the calculi of interest. If $a \triangleright A$ and $\mathcal{K} \vdash A \Leftrightarrow B$, then replacing the pair $a \triangleright A$ in \triangleright by the pair $a \triangleright B$ does not change the language generated by this grammar. This follows from the previous statement.
- Let us agree on the following: if $B = A_1 \bullet \dots \bullet A_n$ is product of several formulas, then $B \setminus C$ denotes the formula $A_1 \setminus \dots \setminus A_n \setminus C$. It can be straightforwardly proved that $\mathcal{K} \vdash B \setminus C \Leftrightarrow B \setminus C$, hence these formulas are interchangeable.

3 The Results of the Work

We aim to disprove Buszkowski’s conjecture that LBC-grammars generate only permutation closures of context-free languages. We shall do this in a constructive way, namely, by presenting a language generated by an LBC-grammar that is not a permutation closure of a context-free language.

Definition 3 Let $\Sigma = \{a, b\}$. Let us define $QL = \{a^l b^n \mid l, n \in \mathbb{N}, 1 \leq n, l \leq n^2\}$. It is the language⁵ consisting of words composed of n symbols b and of l symbols a where the number l is upper bounded by n^2 .

The permutation closure of the language QL is of the main interest for us.

Proposition 2 *The language QL^{perm} is not the permutation closure of a context-free language.*

Proof According to Proposition 1, QL^{perm} is the permutation closure of a context-free language if and only if $\Psi(QL) = \{(l, n) \mid 1 \leq n, l \leq n^2\}$ is semilinear. The proof of the fact that this is not the case is ex falso. Assume that the set $QS = \{(l, n) \mid 1 \leq n, l \leq n^2\}$ is semilinear. It is proved in [15, Theorem 6.1, Theorem 6.2] that semilinear sets are closed under intersection and complement. Consequently, the set $QS' = (\mathbb{N}^2 \setminus QS) \cap \{(l, n) \mid 1 \leq n\}$ is semilinear. Note that $(\mathbb{N}^2 \setminus QS) \cap \{(l, n) \mid 1 \leq n\} = \{(l, n) \mid 1 \leq n, l > n^2\}$.

The set $QS'' = \{(l, n) \mid 1 \leq n, 1 \leq l \leq n^2 + 1\}$ is obtained from QS by adding the vector $(1, 0)$ to each its vector. Thus QS'' is semilinear as well. This yields that the set $QS' \cap QS'' = \{(l, n) \mid 1 \leq n, l = n^2 + 1\}$ is semilinear. This implies that $\{n^2 + 1 \mid n \in \mathbb{N}, 1 \leq n\}$ is semilinear (it is the projection of the former set), which is obviously false. \square

Our goal now is to prove the following theorem:

Theorem 3 *There exists an LBC-grammar QG such that:*

1. QG generates the language QL^{perm} .
2. If the grammar QG is considered as an L*P-grammar, then it generates the same language QL^{perm} .

This theorem disproves Buszkowski’s conjecture and also shows us that Pentus’ theorem relating Lambek categorial grammars and context-free grammars cannot be adapted for the case of LP. We shall prove Theorem 3 by constructing an LBC-grammar generating QL^{perm} ; the construction, however, shall be indirect. Firstly, we shall introduce another formalism called *linearly-restricted branching vector addition systems with states and additional memory (LRBVASSAM)*; their definition is presented in Section 4. These systems form a modification of branching vector addition

⁵ QL is an abbreviation for “quadratic language”.

systems with states defined in [16], which in turn extend vector addition systems [17]. We shall prove in a constructive way that, for a set V generated by some LRBVASSAM H , there exists an LBC-grammar (which we denote by $\text{LBCG}(H)$) generating its inverse Parikh image $\Psi^{-1}(V)$. Then we shall introduce the LRBVASSAM QH generating the set $QS = \{(l, n) \mid 1 \leq n, l \leq n^2\}$, which is the Parikh image of QL^{perm} ; hence the LBC-grammar $\text{LBCG}(QH)$ generates QL^{perm} . This shall constitute the proof of the main theorem of this paper:

Theorem 4 *The set of languages generated by LBC-grammars properly contains the set of permutation closures of context-free languages.*

We shall generalize this statement as follows:

Theorem 5 *For \mathcal{K} being any of the calculi LBC, LP, L^*P , or L_1^*P , \mathcal{K} -grammars can generate languages that are not permutation closures of context-free languages.*

The proof uses LRBVASSAMs and the grammar $\text{LBCG}(H)$ (Construction 2). In fact, it turns out that LBC-grammars and LRBVASSAMs are equivalent:

Theorem 6 *A commutative language L is generated by an LBC-grammar if and only if its Parikh image $\Psi(L)$ is generated by an LRBVASSAM.*

This theorem shall be considered in Section 4, after the definition of LRBVASSAM. It can be viewed as an adaptation of Pentus' theorem. Indeed, LRBVASSAMs belong to the rule-based paradigm, and they have similarities with Chomsky grammars. To prove it, one transforms the LRBVASSAM H generating the set $V = \Psi(L)$ into the LBC-grammar $\text{LBCG}(H)$ generating $\Psi^{-1}(V) = L$ (Construction 2). The other way round, given the LP-grammar G generating L , one constructs the LRBVASSAM, which we denote by $\text{LRB}(G)$, generating $\Psi(L)$ (Construction 3).

As a side result, the equivalence of LP-grammars and LRBVASSAMs allows us to prove the following:

Theorem 7 *The class of languages generated by LP-grammars is equal to the class of languages generated by LBC-grammars.*

That is to say, removing product from the commutative Lambek calculus does not decrease expressive power of categorial grammars, so LP-grammars and LBC-grammars are equivalent. Indeed, each LP-grammar G can be converted into an LRBVASSAM $\text{LRB}(G)$ (Theorem 12), which in turn can be converted into an LBC-grammar $\text{LBCG}(\text{LRB}(G))$ (Theorem 10). Therefore G and $\text{LBCG}(\text{LRB}(G))$ generate the same language. Thus LBC-grammars are at least as expressive as LP-grammars (the converse is trivial).

In Section 7, we establish closure properties of the class of languages generated by LP-grammars using their equivalence with LRBVASSAMs.

Theorem 8 *The class of languages generated by LP-grammars is closed under union, intersection, commutative concatenation, and commutative Kleene plus.*

Let us clarify the meaning of the terms *commutative concatenation* and *commutative Kleene plus*.

Definition 4 The *commutative concatenation* of two commutative languages L_1, L_2 is the language $(\{w_1w_2 \mid w_1 \in L_1, w_2 \in L_2\})^{perm}$.

Definition 5 The *commutative Kleene plus* of the commutative language L is $(L^\oplus)^{perm}$.

4 LRBVASSAM

Vector addition systems are introduced in [17]; they represent a very natural and simple formalism equivalent to well-known Petri nets. The simplest definition of a vector addition system is the following one [17]: a *vector addition system (VAS)* is a pair (v, W) where $W \subseteq \mathbb{Z}^d$ is a finite set of integer-valued vectors of dimension d and $v \in \mathbb{N}^d$ is a vector. The reachability set of the VAS (v, W) consists of vectors of the form $v + w_1 + \dots + w_k$ such that $w_i \in W$, and $v + w_1 + \dots + w_i \in \mathbb{N}^d$ for all $i = 1, \dots, k$. Equivalently, let us say that $u \Rightarrow u + w$ is a direct derivation step in a VAS (v, W) if $u \in \mathbb{N}^d$, $u + w \in \mathbb{N}^d$, and $w \in W$; then the reachability set of (v, W) consists of vectors that can be obtained from v by means of several direct derivation steps.

Countless modifications of vector addition systems have been considered in the literature: VASP, VASS, AVASS, BVASS, EBVASS, EVASS, PVASS etc. These extensions are used for different purposes. As a side remark we note that one of them is studying algorithmic complexity of fragments of propositional linear logic [18, 19]. Thus vector addition systems have already been used for analyzing substructural logics (note that L*P is in fact the multiplicative fragment of intuitionistic linear logic). Branching vector addition systems with states (BVASS) is one of such modifications, which is defined in [16]. As the authors of [16] say, BVASSes are developed as a natural extension of both vector addition systems and Parikh images of context-free grammars. Here is the formal definition of BVASS [16]:

Definition 6 A *branching vector addition system with states (BVASS)* is a tuple $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, K \rangle$ where

- $K \in \mathbb{N}$ is the *dimension of H* ;
- Q is a finite set of *states*;
- \mathcal{P}_0 is a finite set of *nullary rules* of the form $q(\nu)$ where $q \in Q$, $\nu \in \mathbb{N}^K$;
- \mathcal{P}_1 is a finite set of *unary rules* of the form $p(x + \delta_2) \leftarrow q(x + \delta_1)$ where $p, q \in Q$, $\delta_1, \delta_2 \in \mathbb{N}^K$;

- \mathcal{P}_2 is a finite set of *binary rules* of the form $p(x + y) \leftarrow q(x), r(y)$ where $p, q, r \in Q$;
- $s \in Q$ is the distinguished *accepting state*.

Note that x, y are variables (just abstract symbols) while ν, δ_i are vectors from \mathbb{N}^K .

A pair $p(v)$ where $p \in Q, v \in \mathbb{N}^K$ is called a *fact*. A *derivation of a fact* $p(v)$ is a sequence of facts $p_1(v_1), \dots, p_n(v_n)$ with $p_n(v_n) = p(v)$ such that one of the following holds for each $i = 1, \dots, n$:

1. $p_i(v_i) \in \mathcal{P}_0$;
2. $v_i = w + \delta_2$ where $w \in \mathbb{N}^K, w + \delta_1 = v_j$, and $p_i(x + \delta_2) \leftarrow p_j(x + \delta_1) \in \mathcal{P}_1$ (for some $j < i$);
3. $v_i = v_j + v_k$, and $p_i(x + y) \leftarrow p_j(x), p_k(y) \in \mathcal{P}_2$ (for some $j, k < i$).

A *fact* $p(v)$ is *derivable in* H if there exists its derivation. The *set* $L(H)$ *generated by* H consists of vectors $v \in \mathbb{N}^K$ such that $s(v)$ is derivable in H .

For further reasonings, we define the notion of a *derivation tree in a BVASS* H :

Definition 7 The notion of a *derivation tree of a fact* $p(u)$ is inductively defined as follows:

1. If $p(u) \in \mathcal{P}_0$, then $p(u)$ is a derivation tree of $p(u)$;
2. If \mathcal{T} is a derivation tree of a fact $q(v + \delta_1)$ and $\zeta = p(x + \delta_2) \leftarrow q(x + \delta_1) \in \mathcal{P}_1$, then the following structure is a derivation tree of $p(u) = p(v + \delta_2)$:

$$\frac{\mathcal{T}}{p(v + \delta_2)} (\zeta)$$

3. If $\mathcal{T}_1, \mathcal{T}_2$ are derivation trees of $q(v)$ and $r(w)$ resp. and $\eta = p(x + y) \leftarrow q(x), r(y)$ belongs to \mathcal{P}_2 , then the following structure is a derivation tree of $p(u) = p(v + w)$:

$$\frac{\mathcal{T}_1 \quad \mathcal{T}_2}{p(v + w)} (\eta)$$

Definition 8 Given a derivation tree \mathcal{T} , let $\text{Rule}(\mathcal{T})$ be the *multiset of rule occurrences in* \mathcal{T} :

1. If $\mathcal{T} = p(u)$, then $\text{Rule}(\mathcal{T}) = \{p(u)\}$;
2. Let \mathcal{T} be of the following form for some \mathcal{T}' :

$$\frac{\mathcal{T}'}{p(u)} (\zeta)$$

Then $\text{Rule}(\mathcal{T}) = \{\zeta\} \sqcup \text{Rule}(\mathcal{T}')$. Put differently, we increase the number of occurrences of ζ in the multiset $\text{Rule}(\mathcal{T}')$ by 1.

3. Let \mathcal{T} be of the following form for some $\mathcal{T}_1, \mathcal{T}_2$:

$$\frac{\mathcal{T}_1 \quad \mathcal{T}_2}{p(u)} (\eta)$$

Then $\text{Rule}(\mathcal{T}) = \text{Rule}(\mathcal{T}_1) \sqcup \text{Rule}(\mathcal{T}_2) \sqcup \{\eta\}$.

Let the *size of* \mathcal{T} be the cardinality of $\text{Rule}(\mathcal{T})$, i.e. the number of rules in \mathcal{T} .

Example 2 Let $B^0 = \langle Q^0, \mathcal{P}_0^0, \mathcal{P}_1^0, \mathcal{P}_2^0, s, K^0 \rangle$ be the following BVASS.

- The dimension K^0 equals 3.
- The set of states Q^0 equals $\{q, s\}$.
- The set \mathcal{P}_0^0 consists of the following nullary rules:

$$\begin{aligned} - \rho_1^0 &= s(1, 0, 1); \\ - \rho_2^0 &= q(1, 1, 1). \end{aligned}$$

- The set \mathcal{P}_1^0 consists of the following unary rules:

$$\begin{aligned} - \rho_1^1 &= s(x + (0, 3, 0)) \leftarrow s(x + (0, 0, 3)); \\ - \rho_2^1 &= s(x + (0, 0, 3)) \leftarrow s(x + (0, 3, 0)). \end{aligned}$$

- The set \mathcal{P}_2^0 equals $\{\rho_1^2\}$ where

$$\rho_1^2 = s(x + y) \leftarrow s(x), q(y)$$

- The accepting state is s .

Then, for instance, $(3, 2, 3) \in L(B^0)$ and $(3, 5, 0) \in L(B^0)$. Below a derivation tree of $s(3, 5, 0)$ is presented; it contains the derivation tree of $s(3, 2, 3)$ as a subtree.

$$\begin{array}{c} \frac{\frac{s(1, 0, 1) \quad q(1, 1, 1)}{s(2, 1, 2)} \quad (\rho_1^2)}{s(3, 2, 3)} \quad (\rho_1^1) \quad \frac{q(1, 1, 1)}{q(1, 1, 1)} \quad (\rho_1^2)}{s(3, 5, 0)} \quad (\rho_1^1) \\ \frac{s(3, 2, 3)}{s(3, 2, 3)} \quad (\rho_2^1) \\ \frac{s(3, 5, 0)}{s(3, 5, 0)} \quad (\rho_1^1) \\ \frac{s(3, 2, 3)}{s(3, 2, 3)} \quad (\rho_2^1) \\ \frac{s(3, 5, 0)}{s(3, 5, 0)} \quad (\rho_1^1) \end{array} \quad (1)$$

The multiset of rule occurrences in this derivation tree equals $\{\rho_1^0, \rho_2^0, \rho_2^0, \rho_1^2, \rho_1^2, \rho_1^1, \rho_2^1, \rho_1^1, \rho_2^1, \rho_1^1, \rho_2^1, \rho_1^1\}$. Its size equals 10.

Remark 3 If a unary rule is of the form $q(x + \delta_2) \leftarrow p(x + \delta_1)$ and, e.g., $\delta_1 = \vec{0}$, then let us write $q(x + \delta_2) \leftarrow p(x)$ instead of the awkward $q(x + \delta_2) \leftarrow p(x + \vec{0})$.

It turns out that branching vector addition systems with states are related to LP-grammars. However, to formulate this relation precisely we have to modify the definition of BVASS. These modifications are as follows:

1. Firstly, we are going to require that the size of a derivation tree of a fact $s(v)$ in a BVASS must be $O(|v|)$. We call this the *linear restriction*. Note that it immediately makes the membership problem for BVASS decidable and even places it in NP. Indeed, if a vector v belongs to the set $L(H)$ generated by a BVASS H with the

linear restriction, then this can be justified by a derivation tree of $s(v)$ which size does not exceed $C|v|$. Consequently, this derivation has polynomial size w.r.t. the size of v , hence the problem can be solved non-deterministically in polynomial time.

2. Secondly, we are going to add *additional memory* to BVASSes. Namely, let $v \in \mathbb{N}^K$ be a vector such that $s(v)$ is derivable in H . Let us call its first k coordinates ($k \leq K$) *main memory coordinates*, and let us call the remaining $(K - k)$ coordinates *additional memory coordinates*. Then, let us remove additional memory coordinates of v and thus obtain a new vector v' . Let the set generated by H consist of such new vectors of dimension k . Informally, we consider the last $K - k$ coordinates as auxiliary ones that are used only for intermediate calculations and that must be removed by the end of a derivation. Moreover, let us require that additional memory coordinates in v must equal 0 by the end of the derivation.

Let us formally introduce both modifications. Given a vector $v' \in \mathbb{N}^k$ and $K \geq k$, we denote by $\iota_K(v')$ the vector $v \in \mathbb{N}^K$ such that $v_i = v'_i$ ($i = 1, \dots, k$) and $v_i = 0$ for $k < i \leq K$. Informally, ι_K appends $(K - k)$ additional memory coordinates to v' , which are equal to 0. For example, $\iota_5(1, 2, 3) = (1, 2, 3, 0, 0)$.

Definition 9 A *linearly-restricted branching vector addition system with states and additional memory (LRBVASSAM)* is a tuple $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, k, K, C \rangle$ where all the components except for k and C are defined as in Definition 6 and $k, C \in \mathbb{N}$, $0 \leq k \leq K$. The set $L(H)$ generated by this LRBVASSAM consists of vectors $u \in \mathbb{N}^k$ such that there exists a derivation tree of $s(\iota_K(u))$ in H of size not greater than $C|u|$.

Remark 4 Note that the zero vector $\vec{0}$ does not belong to $L(H)$ for any H . Indeed, this would imply that there exists a derivation tree of $s(\iota_K(\vec{0}))$ of size not greater than 0, which is impossible since the size of any derivation tree is at least 1.

Example 3 Let $H^0 = \langle Q^0, \mathcal{P}_0^0, \mathcal{P}_1^0, \mathcal{P}_2^0, s, k^0, K^0, C^0 \rangle$ be an LRBVASSAM defined as follows:

- $Q^0, \mathcal{P}_0^0, \mathcal{P}_1^0, \mathcal{P}_2^0, s, K^0$ are the same as in Example 2;
- $k^0 = 2$;
- $C^0 = 1$.

Since $k^0 = 2$ and $K^0 = 3$, there are two main memory coordinates and one additional memory coordinate. Then $(3, 5) \in L(H^0)$. To check this, we need to show that there is a derivation tree of $s(\iota_3(3, 5)) = s(3, 5, 0)$ in H of size less than or equal to $C^0 \cdot |(3, 5)| = 1 \cdot 8 = 8$. Note that (1) is not such a derivation tree because its size is $10 > 8$. However, obviously, it can be reduced to the following one:

$$\frac{\frac{\frac{s(1, 0, 1) \quad q(1, 1, 1)}{s(2, 1, 2)} \quad (\rho_1^2) \quad q(1, 1, 1)}{s(3, 2, 3)} \quad (\rho_1^2)}{s(3, 5, 0)} \quad (\rho_1^1) \quad (2)$$

The size of the new derivation tree equals $6 \leq 8$. Hence $(3, 5) \in L(H^0)$.

Note that $(3, 2) \notin L(H^0)$, although $s(3, 2, 3)$ is derivable in $L(B^0)$ as shown in Example 2. This happens because we do not simply erase the additional memory coordinates but we

check that they equal 0 and then remove them. It is left as an exercise to rigorously prove that $(3, 2) \notin L(H^0)$.

Remark 5 It is not so common to directly restrict sizes of derivations for formal grammars; usually, in order to decrease expressive power of some class of formal grammars, restrictions are imposed on the structure of rules used in them. For example, unrestricted grammars may include rules of the form $\alpha \rightarrow \beta$ where α and β are arbitrary words; however, such grammars are too powerful, and the membership problem for them is undecidable. This makes them computationally unattractive. However, if we allow one to use only rules of the form $A \rightarrow \beta$, then we obtain context-free grammars, which are effectively decidable. This is an example of how the structure of rules in a grammar can be restricted to reduce expressive power of grammars. In the case of LRBVASSAM, we do not change the definition of rules for BVASS or the definition of a derivation but rather explicitly limit size of a derivation.

One example of imposing a linear restriction similar to that from Definition 9 can be found in [20] for multiset-valued linear index grammars; however, we failed to find a simple connection between these grammars and LP-grammars, so we develop our own definitions. It would also be interesting to consider other kinds of restrictions, e.g., the quadratic one or polynomial one and to establish hierarchy results concerning generated classes of sets.

The linear restriction, however, takes a special place among other possible restrictions due to the following reason. For each BVASS H there exists a constant $C_H > 0$ such that, if $s(v)$ is derivable in H , then the size of its derivation must be at least $C_H \cdot |v|$. Indeed, if D_H is the maximum of $|\delta_2|$ such that $p(x + \delta_2) \leftarrow q(x + \delta_1) \in \mathcal{P}_1$ and of all numbers $|\nu|$ for $p(\nu) \in \mathcal{P}_0$, then a derivation tree of the size n always results in a fact $q(v)$ such that $|v| \leq D_H \cdot n$ (the proof is by induction on n). Finally, let C_H equal $1/D_H$. Ergo, the linear restriction is the least possible one that does not make the resulting language finite; this property distinguishes it among other restrictions.

The main result relating LRBVASSAMs and LBC-grammars is the following theorem:

Theorem 6 A commutative language L is generated by an LBC-grammar if and only if its Parikh image $\Psi(L)$ is generated by an LRBVASSAM.

It turns out that for further reasonings it is important to require that all the unary rules in an LRBVASSAM are of the form $q(x + \delta_2) \leftarrow p(x + \delta_1)$ for δ_1 being non-zero.

Definition 10 An LRBVASSAM $\langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, k, K, C \rangle$ is *input-sensitive* if, for each $\varphi_1 = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$, it holds that $\delta_1 \neq \vec{0}$.

Proposition 9 For each LRBVASSAM $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, k, K, C \rangle$ there exists an input-sensitive LRBVASSAM \tilde{H} generating the same set $L(H)$.

Proof Given a vector $v = (v_1, \dots, v_K) \in \mathbb{N}^K$, let $v \frown a$ denote the vector (v_1, \dots, v_K, a) .

Let us define $\tilde{H} = \langle \tilde{Q}, \tilde{\mathcal{P}}_0, \tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \tilde{s}, k, \tilde{K}, \tilde{C} \rangle$ as follows:

- $\tilde{Q} = Q$; $\tilde{s} = s$; $\tilde{C} = C$;
- $\tilde{K} = K + 1$;

- $\widetilde{\mathcal{P}}_0$ consists of nullary rules of the form $\varphi^{(0)} = q(\nu \frown 0)$ and $\varphi^{(1)} = q(\nu \frown 1)$ for $\varphi = q(\nu) \in \mathcal{P}_0$;
- $\widetilde{\mathcal{P}}_1$ consists of unary rules of the form $\varphi^{(0)} = q(x + \delta_2 \frown 0) \leftarrow p(x + \delta_1 \frown 1)$ and $\varphi^{(1)} = q(x + \delta_2 \frown 1) \leftarrow p(x + \delta_1 \frown 1)$ for $\varphi = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$;
- $\widetilde{\mathcal{P}}_2 = \mathcal{P}_2$;

The LRBVASSAM \widetilde{H} is clearly input-sensitive.

Given a derivation tree $\widetilde{\mathcal{T}}$ of a fact $s(\iota_{\widetilde{K}}(v))$ in \widetilde{H} for $v \in \mathbb{N}^k$, one can obtain a derivation tree of $s(\iota_K(v))$ in H from it by removing the last coordinate from each vector in each fact occurring in $\widetilde{\mathcal{T}}$ and by replacing each rule of the form $\varphi^{(i)}$ by φ . The size of the new derivation tree equals that of $\widetilde{\mathcal{T}}$. Consequently, $L(\widetilde{H}) \subseteq L(H)$.

Let us show the converse inclusion. Let $v \in L(H)$; this means, by the definition, that $s(\iota_K(v))$ has a derivation tree \mathcal{T} in H of the size $\leq C|v|$.

We define functions $\chi^{(0)}$ and $\chi^{(1)}$ such that $\chi^{(i)}$ transforms a derivation tree \mathcal{T} of a fact $t(u)$ into a derivation tree of the fact $t(u \frown i)$ in \widetilde{H} (for $i \in \{0, 1\}$). The joint inductive definition is as follows:

- If \mathcal{T} is of the form $t(u)$ (hence $t(u) \in \mathcal{P}_0$), then $\chi^{(i)}(\mathcal{T}) := t(u \frown i) \in \widetilde{\mathcal{P}}_0$.
- Let \mathcal{T} be of the following form for $\zeta = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$:

$$\frac{\mathcal{T}'}{q(w + \delta_2)} (\zeta)$$

Here \mathcal{T}' is a derivation tree of $p(w + \delta_1)$. Then let us define $\chi^{(i)}(\mathcal{T})$ as the following derivation tree:

$$\frac{\chi^{(1)}(\mathcal{T}')}{q((w + \delta_2) \frown i)} (\zeta^{(i)})$$

Note that $\chi^{(1)}(\mathcal{T}')$ is the derivation tree of $q((w + \delta_1) \frown 1) = q(\iota_{\widetilde{K}}(w) + \delta_1 \frown 1)$, and $\zeta^{(i)} = q(x + \delta_2 \frown i) \leftarrow p(x + \delta_1 \frown 1) \in \widetilde{\mathcal{P}}_1$.

- Let \mathcal{T} be of the following form for $\mathcal{T}_1, \mathcal{T}_2$ being derivation trees of facts $p_1(w_1)$ and $p_2(w_2)$ resp. and for $\eta = q(x_1 + x_2) \leftarrow p_1(x_1), p_2(x_2)$:

$$\frac{\mathcal{T}_1 \quad \mathcal{T}_2}{q(w_1 + w_2)} (\eta)$$

Then $\chi^{(i)}(\mathcal{T})$ is of the following form:

$$\frac{\chi^{(0)}(\mathcal{T}_1) \quad \chi^{(i)}(\mathcal{T}_2)}{q((w_1 + w_2) \frown i)} (\eta)$$

Clearly, $\chi^{(i)}$ translates a correct derivation tree into a correct one and, moreover, it preserves the structure and the size of the initial derivation tree. Thus, given a derivation tree \mathcal{T} of $s(\iota_K(v))$, $\chi^{(0)}(\mathcal{T})$ is the derivation tree of $s(\iota_K(v) \frown 0) = s(\iota_{\widetilde{K}}(v))$ in \widetilde{H} of the same size. This proves that $L(H) \subseteq L(\widetilde{H})$. \square

Remark 6 From now on, without loss of generality, we usually consider all LRBVASSAMs to be input-sensitive.

4.1 From LRBVASSAMs to LBC-Grammars

Our goal is to prove that LRBVASSAMs are equivalent to LBC-grammars. Firstly, we shall show that the latter are at least as expressive as the former. To do this, we would like to encode nullary, unary and binary rules of LRBVASSAMs using formulas of LBC. Let us simplify this task a bit and use LP instead of LBC; so, we allow ourselves to use product.

Construction 1

Assume that we are given the input-sensitive LRBVASSAM $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$.

- Let us consider states from Q as primitive formulas; besides, we introduce new primitive formulas g_1, \dots, g_K, f (g_i correspond to standard-basis vectors e_i in \mathbb{N}^K , and f is a distinguished formula with the informal meaning of being the finish state). Let us agree on the following notation: if $v \in \mathbb{N}^K$, then $g^{\bullet v} := g_1^{\bullet v_1} \bullet \dots \bullet g_K^{\bullet v_K}$ and $g^v := g_1^{v_1} \sqcup \dots \sqcup g_K^{v_K}$. Hence $g^{\bullet v}$ consists of g_i -s combined using product \bullet while g^v is a multiset consisting of g_i -s.
- For each $\varphi_0 = q(\nu) \in \mathcal{P}_0$, let $T(\varphi_0) := q \setminus g^{\bullet \nu} \setminus f$;
- For each $\varphi_1 = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$, let $T(\varphi_1) := q \setminus g^{\bullet \delta_2} \setminus (p \bullet g^{\bullet \delta_1})$;
- For each $\varphi_2 = q(x + y) \leftarrow p(x), r(y) \in \mathcal{P}_2$, let $T(\varphi_2) := q \setminus (p \setminus f) \setminus (r \setminus f) \setminus f$.
- Let \mathcal{P} denote $\mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2$.

Finally, let $\text{LPG}(H)$ be the following LP-grammar: $\text{LPG}(H) := \langle s \setminus f, \triangleright \rangle$ where $a_i \triangleright A$ iff $A = g_i \bullet T(\varphi_1) \bullet \dots \bullet T(\varphi_j)$ for some $0 \leq j \leq C$ and for some φ_l from \mathcal{P} (i here is iterated over the range $1, \dots, \kappa$).

Remark 7 Formally, the definition of $A^{\bullet l}$ is not correct if $l = 0$ (what is the product of A with itself zero times?); consequently, $g^{\bullet v}$ can also be undefined. A way of defining $A^{\bullet 0}$ could be $A^{\bullet 0} = I$ (where I is the unit). While there is the unit constant I in L_1^*P , we do not have it in LP. Nevertheless, let us notice that formulas of the form $g^{\bullet v}$ appear in Construction 1 only in certain positions, namely, within formulas of the form $q \setminus g^{\bullet \nu} \setminus f$ and $q \setminus g^{\bullet \delta_2} \setminus (p \bullet g^{\bullet \delta_1})$. This suggests the following treatment of the problematic cases:

- If $v = (v_1, \dots, v_K)$ and $v_{i_1} \neq 0, \dots, v_{i_j} \neq 0$ ($j \geq 1, i_1 < \dots < i_j$) while $v_t = 0$ for $t \notin \{i_1, \dots, i_j\}$, then we define $g^{\bullet v}$ as $g_{i_1}^{\bullet v_{i_1}} \bullet \dots \bullet g_{i_j}^{\bullet v_{i_j}}$.
- If $\nu = \vec{0}$ in the formula $q \setminus g^{\bullet \nu} \setminus f$, then we simply take a formula of the form $q \setminus f$ instead.
- If $\delta_2 = \vec{0}$ in the formula $q \setminus g^{\bullet \delta_2} \setminus (p \bullet g^{\bullet \delta_1})$, then we take a formula of the form $q \setminus (p \bullet g^{\bullet \delta_1})$ instead.

Note that it is impossible that $\delta_1 = \vec{0}$ since the LRBVASSAM is input-sensitive.

Example 4 Let us take the LRBVASSAM H^0 from Example 3. Then we can transform it into $\text{LPG}(H^0)$ (note that $\kappa = 2$, therefore $\Sigma = \{a_1, a_2\}$). To do this, let us introduce primitive formulas q, s, f, g_1, g_2, g_3 and construct the following formulas:

- $T(\rho_1^0) = s \setminus (g_1 \bullet g_3) \setminus f$;
- $T(\rho_2^0) = q \setminus (g_1 \bullet g_2 \bullet g_3) \setminus f$;

- $T(\rho_1^1) = s \backslash (g_2 \bullet g_2 \bullet g_2) \backslash (s \bullet g_3 \bullet g_3 \bullet g_3)$;
- $T(\rho_2^1) = s \backslash (g_3 \bullet g_3 \bullet g_3) \backslash (s \bullet g_2 \bullet g_2 \bullet g_2)$;
- $T(\rho_1^2) = s \backslash (s \backslash f) \backslash (q \backslash f) \backslash f$.

Then $\text{LPG}(H^0) = \langle s \backslash f, \triangleright^0 \rangle$ where

1. $a_1 \triangleright^0 g_1$; $a_1 \triangleright^0 g_1 \bullet T(\rho)$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$;
2. $a_2 \triangleright^0 g_2$; $a_2 \triangleright^0 g_2 \bullet T(\rho)$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$.

Since $C^0 = 1$, we consider only products with at most one formula of the form $T(\rho)$.

The idea behind Construction 1 is to represent the fact $t(v_1, v_2, v_3)$ by the multiset $g_1^{v_1}, g_2^{v_2}, g_3^{v_3}, t$ in the antecedent of a sequent. Formulas $T(\rho)$ encode rules applied in a derivation of the fact. In fact, we shall show that, if the fact $t(v_1, v_2, v_3)$ is derivable in an LRBVASSAM, then $g_1^{v_1}, g_2^{v_2}, g_3^{v_3}, t, T(\zeta_1), \dots, T(\zeta_m) \Rightarrow f$ is derivable in LP where ζ_1, \dots, ζ_m are the rules applied in the derivation of $t(v_1, v_2, v_3)$. For example, consider the following derivation of $s(2, 1, 2)$:

$$\frac{s(1, 0, 1) \quad q(1, 1, 1)}{s(2, 1, 2)} (\rho_1^2)$$

Its multiset of rule occurrences is $\rho_1^0, \rho_2^0, \rho_1^2$. This derivation corresponds to the following derivation in LP (we omit some its parts to fit in the page width):

$$\frac{\frac{f \Rightarrow f \quad g_1, g_3 \Rightarrow g_1 \bullet g_3}{g_1, g_3, (g_1 \bullet g_3) \backslash f \Rightarrow f} (\backslash_L) \quad \frac{s \Rightarrow s}{s \Rightarrow s} (\backslash_L)}{\frac{g_1, g_2, g_3, q, T(\rho_2^0) \Rightarrow f}{g_1, g_2, g_3, T(\rho_2^0) \Rightarrow q \backslash f} (\backslash_R) \quad \frac{g_1, g_3, s, T(\rho_1^0) \Rightarrow f}{g_1, g_3, T(\rho_1^0) \Rightarrow s \backslash f} (\backslash_R)}{\frac{g_1, g_2, g_3, T(\rho_2^0), (q \backslash f) \backslash f \Rightarrow f}{g_1, g_2, g_3, T(\rho_2^0), g_1, g_3, T(\rho_1^0), (s \backslash f) \backslash (q \backslash f) \backslash f \Rightarrow f} (\backslash_L) \quad \frac{s \Rightarrow s}{s \Rightarrow s} (\backslash_L)}{\frac{g_1, g_1, g_2, g_3, g_3, s, T(\rho_2^0), T(\rho_1^0), T(\rho_1^2) \Rightarrow f}{g_1, g_1, g_2, g_3, g_3, T(\rho_2^0), T(\rho_1^0), T(\rho_1^2) \Rightarrow s \backslash f} (\backslash_R)}$$

Similarly, one can transform the derivation (2) into the derivation of the sequent $g_1, g_1, g_1, g_2, g_2, g_2, g_2, g_2, T(\rho_2^0), T(\rho_1^0), T(\rho_1^2), T(\rho_1^1), T(\rho_1^1) \Rightarrow s \backslash f$. From this sequent we can derive the one $g_1 \bullet T(\rho_2^0), g_1 \bullet T(\rho_1^0), g_1 \bullet T(\rho_1^2), g_2 \bullet T(\rho_1^2), g_2 \bullet T(\rho_1^1), g_2, g_2, g_2 \Rightarrow s \backslash f$. Its derivability proves that $a_1 a_1 a_1 a_2 a_2 a_2 a_2 a_2$ belongs to $L(\text{LPG}(H^0))$.

The constructed grammar allows one to simulate derivations in H by derivations in LP using the LP-grammar $\text{LPG}(H)$. However, the latter grammar uses many formulas with product, so it is not an LBC-grammar. Our goal now is to enhance Construction 1 by eliminating product. The idea of reaching this goal is to use the equivalence $(A \bullet B) \backslash C \Leftrightarrow A \backslash B \backslash C$. If $\varphi_0 = q(\nu) \in \mathcal{P}_0$, then $T(\varphi_0)$ can be re-defined as $q \backslash g^{\bullet \nu} \parallel f$ (see the definition of \parallel in Section 2.5). However, this trick can be used only if product stands under division, so it does not work with formulas $T(\varphi_1)$ for $\varphi_1 \in \mathcal{P}_1$. Namely, the subformula $p \bullet g^{\bullet \varphi_1}$ is not under division. Even worse, in the definition of $\text{LPG}(H)$, we introduce formulas $g_i \bullet T(\varphi_1) \bullet \dots \bullet T(\varphi_j)$ where product is the outermost operation.

To handle these cases, we shall use *type-raising* considered in various papers on the Lambek calculus and on combinatory categorial grammars (see, e.g., [21]). The idea is, given a formula $B = A_1 \bullet \dots \bullet A_n$, to replace it by a formula of the form $(B \backslash C) \backslash C$ for some C ; this places B under division, hence we can replace the latter formula by the equivalent one $(B \parallel C) \backslash C$. The same trick is extensively used in [22]:

there $C = b$ is a fresh variable, and $A \setminus b$ is called *relative negation of A*. Thus $(B \setminus b) \setminus b$ is a “pseudo-double-negation” of B .

Let us modify Construction 1 as follows:

Construction 2

- For each $\varphi_0 = q(\nu) \in \mathcal{P}_0$ we define the formula $T'(\varphi_0) := q \setminus g^{\bullet\nu} \setminus f$;
- For each $\varphi_1 = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$ we define the formula $T'(\varphi_1) := q \setminus g^{\bullet\delta_2} \setminus ((p \bullet g^{\bullet\delta_1}) \setminus f) \setminus f$;
- For each $\varphi_2 = q(x + y) \leftarrow p(x), r(y) \in \mathcal{P}_2$ we define the formula $T'(\varphi_2) := T(\varphi_2) = q \setminus (p \setminus f) \setminus (r \setminus f) \setminus f$.

Finally, let us define the LBC-grammar $\text{LBCG}(H)$ as follows: $\text{LBCG}(H) := \langle s \setminus f, \triangleright' \rangle$ where $a_i \triangleright' B$ if and only if $B = s \setminus ((s \bullet A) \setminus f) \setminus f$ where $A = g_i \bullet T'(\varphi_1) \bullet \dots \bullet T'(\varphi_j)$ for some $0 \leq j \leq C$ and for some φ_l from $\mathcal{P} = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2$ (i is iterated over the range $1, \dots, \kappa$).

Clearly, $\text{LBCG}(H)$ is an LBC-grammar since all the formulas are defined using only the division operation.

Remark 8 Unfortunately, the number of pairs in $\text{LBCG}(H)$ is not polynomial w.r.t. the size of the LRBVASSAM H , so the construction is not polynomial in size. However, it is polynomial if the parameter C in the LRBVASSAM is fixed: the number of pairs in the binary relation of $\text{LBCG}(H)$ has the upper bound $|\Sigma|(1 + |\mathcal{P}| + \dots + |\mathcal{P}|^C) \leq \kappa |\mathcal{P}|^{C+1}$ for $|\mathcal{P}| > 1$.

Example 5 Let us take the LRBVASSAM H^0 from Example 3 and transform it into $\text{LBCG}(H^0)$. The following formulas are constructed:

- $T'(\rho_1^0) = s \setminus g_1 \setminus g_3 \setminus f$;
- $T'(\rho_2^0) = q \setminus g_1 \setminus g_2 \setminus g_3 \setminus f$;
- $T'(\rho_1^1) = s \setminus g_2 \setminus g_2 \setminus g_2 \setminus (s \setminus g_3 \setminus g_3 \setminus g_3 \setminus f) \setminus f$;
- $T'(\rho_2^1) = s \setminus g_3 \setminus g_3 \setminus g_3 \setminus (s \setminus g_2 \setminus g_2 \setminus g_2 \setminus f) \setminus f$;
- $T'(\rho_1^2) = s \setminus (s \setminus f) \setminus (q \setminus f) \setminus f$.

Then $\text{LBCG}(H^0) = \langle s \setminus f, \triangleright' \rangle$ where

1. $a_1 \triangleright' s \setminus (s \setminus g_1 \setminus f) \setminus f$; $a_1 \triangleright' s \setminus (s \setminus g_1 \setminus T'(\rho) \setminus f) \setminus f$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$;
2. $a_2 \triangleright' s \setminus (s \setminus g_2 \setminus f) \setminus f$; $a_2 \triangleright' s \setminus (s \setminus g_2 \setminus T'(\rho) \setminus f) \setminus f$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$.

Another example of applying Construction 2 is Example ??.

The main result of this section is the following theorem.

Theorem 10 *Let H be an input-sensitive LRBVASSAM. Then:*

1. $L(\text{LBCG}(H)) = \Psi^{-1}(L(H))$.
2. *The grammar $\text{LBCG}(H)$ being considered as an L*P-grammar generates the same language $\Psi^{-1}(L(H))$.*

Firstly, we want to introduce yet another construction for the sake of simplifying proof analysis. Namely, let us define the LP-grammar $\text{LBCG}_1(H)$.

1. For each $\varphi_0 = q(\nu) \in \mathcal{P}_0$ we define the formula $T'_1(\varphi_0) := q \backslash g^{\bullet \nu} \backslash f$;
2. For each $\varphi_1 = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$ we define the formula $T'_1(\varphi_1) := q \backslash g^{\bullet \delta_2} \backslash ((p \bullet g^{\bullet \delta_1}) \backslash f) \backslash f$;
3. For each $\varphi_2 = q(x + y) \leftarrow p(x), r(y) \in \mathcal{P}_2$ we define the formula $T'_1(\varphi_2) := q \backslash (p \backslash f) \backslash (r \backslash f) \backslash f$.

Then, $\text{LBCG}_1(H) := \langle s \backslash f, \triangleright'_1 \rangle$ where $a_i \triangleright'_1 B$ if and only if

$$B = s \backslash ((s \bullet A) \backslash f) \backslash f$$

where $A = g_i \bullet T'_1(\varphi_1) \bullet \dots \bullet T'_1(\varphi_j)$ for some $0 \leq j \leq C$ such that all φ_l are from $\mathcal{P} = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2$.

Example 6 Let us take the LRBVASSAM H^0 from Example 3 and transform it into $\text{LBCG}_1(H^0)$. The following formulas are constructed:

- $T'_1(\rho_1^0) = s \backslash (g_1 \bullet g_3) \backslash f$;
- $T'_1(\rho_2^0) = q \backslash (g_1 \bullet g_2 \bullet g_3) \backslash f$;
- $T'_1(\rho_1^1) = s \backslash (g_2 \bullet g_2 \bullet g_2) \backslash ((s \bullet g_3 \bullet g_3 \bullet g_3) \backslash f) \backslash f$;
- $T'_1(\rho_2^1) = s \backslash (g_3 \bullet g_3 \bullet g_3) \backslash ((s \bullet g_2 \bullet g_2 \bullet g_2) \backslash f) \backslash f$;
- $T'_1(\rho_1^2) = s \backslash (s \backslash f) \backslash (q \backslash f) \backslash f$.

Then $\text{LBCG}_1(H^0) = \langle s \backslash f, \triangleright'_1 \rangle$ where

1. $a_1 \triangleright' s \backslash ((s \bullet g_1) \backslash f) \backslash f$; $a_1 \triangleright' s \backslash ((s \bullet g_1 \bullet T'_1(\rho)) \backslash f) \backslash f$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$;
2. $a_2 \triangleright' s \backslash ((s \bullet g_2) \backslash f) \backslash f$; $a_2 \triangleright' s \backslash ((s \bullet g_2 \bullet T'_1(\rho)) \backslash f) \backslash f$ for $\rho \in \{\rho_1^0, \rho_2^0, \rho_1^1, \rho_2^1, \rho_1^2\}$.

Remark 9 The grammar $\text{LBCG}_1(H)$ is obtained from $\text{LBCG}(H)$ by replacing each \parallel by \backslash . The resulting grammar is not an LBC-grammar anymore, since it includes formulas with products. Note that $L(\text{LBCG}(H)) = L(\text{LBCG}_1(H))$; this directly follows from the fact that replacing a formula in a grammar by an equivalent one does not change the language generated by the grammar (see Subsection 2.5).

So, our goal is to prove that $L(\text{LBCG}_1(H)) = \Psi^{-1}(L(H))$. The proof consists of the main lemma supported by several technical definitions, propositions and lemmas.

Definition 11 Let $\Theta = \{\theta_1, \dots, \theta_l\}$ be a multiset consisting of rules from \mathcal{P} with a fixed enumeration of its elements. Then $F^{(i)}(\Theta)$ is defined as the formula $s \backslash ((s \bullet A^{(i)}(\Theta)) \backslash f) \backslash f$ where $A^{(i)}(\Theta) = g_i \bullet T'_1(\theta_1) \bullet \dots \bullet T'_1(\theta_l)$.

Note that, if Θ' is the same set Θ with some other enumeration of its elements, then $\mathcal{K} \vdash F^{(i)}(\Theta) \Leftrightarrow F^{(i)}(\Theta')$. Therefore, it is not important how elements of Θ are enumerated.

This definition introduces a short notation for formulas appearing in the grammar $\text{LBCG}_1(H)$. Indeed, the definition of $\text{LBCG}_1(H)$ says that $a_i \triangleright_1^l F^{(i)}(\Theta)$ for all possible multisets Θ of cardinality less than or equal to C that consist of rules from \mathcal{P} .

Definition 12 Let $u = (u_1, \dots, u_K) \in \mathbb{N}^K$ be a vector such that $|u| = M$. We transform it into the following tuple $\text{tuple}(u) = (i_1, \dots, i_M)$: firstly, $i_1 = \dots = i_{u_1} = 1$; secondly, $i_{u_1+1} = \dots = i_{u_1+u_2} = 2$; in general, $i_{u_1+\dots+u_{l-1}+1} = \dots = i_{u_1+\dots+u_{l-1}+u_l} = l$ (for $l = 2, \dots, K$). For example, if $u = (2, 3, 0, 1)$, then the tuple is $i_1 = 1, i_2 = 1, i_3 = 2, i_4 = 2, i_5 = 2, i_6 = 4$.

Let $\Theta_1, \dots, \Theta_M$ be finite ordered multisets of rules from \mathcal{P} and let $u \in \mathbb{N}^K$ such that $|u| = M$. Then $F^{(u)}(\Theta_1, \dots, \Theta_M) := F^{(i_1)}(\Theta_1), \dots, F^{(i_M)}(\Theta_M)$ where $(i_1, \dots, i_M) = \text{tuple}(u)$.

Lemma 1 (main) Let $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$ be an input-sensitive LRBVASSAM. Consider the sequent of the form

$$g^u, F^{(u')}(\Theta_1, \dots, \Theta_M), t \Rightarrow f \quad (3)$$

where $u, u' \in \mathbb{N}^K$, $m, M \in \mathbb{N}$, $M = |u'|$, $t \in Q$, and Θ_i is a finite ordered multiset of rules from \mathcal{P} for each $i = 1, \dots, M$.

The following statements are equivalent:

1. The sequent (3) is derivable in LP;
2. The sequent (3) is derivable in L*P;
3. The fact $t(u + u')$ has a derivation tree in H such that the multiset of its rule occurrences is $\Theta_1 \sqcup \dots \sqcup \Theta_M$.

This lemma establishes the equivalence of proofs in LP, L*P and in the given LRBVASSAM H . Obviously, statement 1 implies statement 2 (because L*P extends LP). We shall show that statement 2 implies statement 3 and that statement 3 implies statement 1. The former implication is harder than the latter one. To prove it we need to analyze proofs of sequents in L*P. The core of the proof is the following proposition formulated for formulas of the form $T(\zeta)$ from the grammar $\text{LPG}(H)$.

Proposition 11 Let $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$ be an input-sensitive LRBVASSAM. Consider a sequent of the form $g^u, T(\zeta_1), \dots, T(\zeta_m), t \Rightarrow f$ where $u \in \mathbb{N}^K$, $m \in \mathbb{N}$, $\zeta_i \in \mathcal{P}$, $t \in Q$. If it is derivable in L*P, then the fact $t(u)$ has a derivation tree in H such that the multiset of its rule occurrences is $\{\zeta_1, \dots, \zeta_m\}$.

Its proof is not straightforward, it involves several technical lemmas, so we delay it to Section 4.1.2. To prove that statement 2 of Lemma 1 implies statement 3 we need another lemma.

Lemma 2

1. $\text{LP} \vdash T(\zeta) \Rightarrow T'_1(\zeta)$.
2. $\text{LP} \vdash g^i, T(\theta_1), \dots, T(\theta_l) \Rightarrow F^{(i)}(\Theta)$ where $\Theta = \{\theta_1, \dots, \theta_l\}$.

Proof Note that $T(\zeta) = T'_1(\zeta)$ whenever $\zeta \in \mathcal{P}_0 \cup \mathcal{P}_2$. If $\zeta \in \mathcal{P}_1$, then

- $T(\zeta) = q \backslash g^{\bullet \delta_2} \backslash (p \bullet g^{\bullet \delta_1})$,
- $T'_1(\zeta) = q \backslash g^{\bullet \delta_2} \backslash ((p \bullet g^{\bullet \delta_1}) \backslash f) \backslash f$.

Derivability of $q \backslash g^{\bullet \delta_2} \backslash (p \bullet g^{\bullet \delta_1}) \Rightarrow q \backslash g^{\bullet \delta_2} \backslash ((p \bullet g^{\bullet \delta_1}) \backslash f) \backslash f$ follows from the fact that $\text{LP} \vdash A \Rightarrow (A \backslash B) \backslash B$ for all formulas A, B .

To prove the second statement, firstly notice that the following structure represents a correct derivation in LP:

$$\frac{\frac{f \Rightarrow f \quad s, g^i, T'_1(\theta_1), \dots, T'_1(\theta_l) \Rightarrow s \bullet g_i \bullet T'_1(\theta_1) \bullet \dots \bullet T'_1(\theta_l)}{g^i, T'_1(\theta_1), \dots, T'_1(\theta_l), s, (s \bullet g_i \bullet T'_1(\theta_1) \bullet \dots \bullet T'_1(\theta_l)) \backslash f \Rightarrow f} (\backslash_L)}{g^i, T'_1(\theta_1), \dots, T'_1(\theta_l) \Rightarrow s \backslash ((s \bullet g_i \bullet T'_1(\theta_1) \bullet \dots \bullet T'_1(\theta_l)) \backslash f) \backslash f} (\backslash_R) \times 2$$

Thus $\text{LP} \vdash g^i, T'_1(\theta_1), \dots, T'_1(\theta_l) \Rightarrow F^{(i)}(\Theta)$. Applying the cut rule several times to this sequent and to the sequents $T(\theta_1) \Rightarrow T'_1(\theta_1), \dots$, and to $T(\theta_l) \Rightarrow T'_1(\theta_l)$, which have been proved to be derivable in LP, we conclude that $\text{LP} \vdash g^i, T(\theta_1), \dots, T(\theta_l) \Rightarrow F^{(i)}(\Theta)$, as desired. \square

When proving that statement 3 of Lemma 1 implies statement 1 we shall use the following lemmas.

Lemma 3 *If the fact $t(u)$ has a derivation tree in H such that the multiset of its rule occurrences is $\{\varphi_1, \dots, \varphi_N\}$, then $\text{LP} \vdash g^u, T'_1(\varphi_1), \dots, T'_1(\varphi_N), t \Rightarrow f$.*

Proof The proof is by induction on N , which is the size of a derivation tree.

The base case is $N = 0$. Then there is no derivation tree of $t(u)$ with the empty multiset of rule occurrences (there must be at least one nullary rule), ergo the “if” statement is false.

To prove the induction step, consider a derivation tree of $t(u)$. There are three possible cases how it might look like (corresponding to three items of Definition 7):

Case A. The derivation tree of $t(u)$ is $t(u)$. Then this fact must be a nullary rule $t(u) = \varphi_1 \in \mathcal{P}_0$. Then $T'_1(\varphi_1) = T(\varphi_1) = t \backslash g^{\bullet u} \backslash f$. Below a derivation of $g^u, T(\varphi_1), t \Rightarrow f$ is presented:

$$\frac{\frac{f \Rightarrow f \quad g^u \Rightarrow g^{\bullet u}}{g^u, g^{\bullet u} \backslash f \Rightarrow f} (\backslash_L) \quad t \Rightarrow t}{g^u, t \backslash g^{\bullet u} \backslash f, t \Rightarrow f} (\backslash_L)$$

The sequent $g^u \Rightarrow g^{\bullet u}$ is derivable by means of the rule (\bullet_R) .

Formally, we must also consider the case where $u = \bar{0}$. In this case, $T(\varphi_1) = t \backslash f$, and the derivation is of the form:

$$\frac{f \Rightarrow f \quad t \Rightarrow t}{t \backslash f, t \Rightarrow f} (\backslash_L)$$

Case B. The derivation tree of $t(u)$ is of the form

$$\frac{\mathcal{T}}{t(v + \delta_2)} (\varphi_i)$$

for some $\varphi_i = t(x + \delta_2) \leftarrow q(x + \delta_1) \in \mathcal{P}_1$; in particular, $u = v + \delta_2$. Without loss of generality, assume that $i = 1$. Then \mathcal{T} is a derivation tree of the fact $q(v + \delta_1)$, and $\text{Rule}(\mathcal{T}) =$

$\{\varphi_2, \dots, \varphi_N\}$. By the induction hypothesis, the sequent $g^{v+\delta_1}, T'_1(\varphi_2), \dots, T'_1(\varphi_N), q \Rightarrow f$ is derivable. From it, one can derive the sequent $g^u, T'_1(\varphi_1), \dots, T'_1(\varphi_N), t \Rightarrow f$ as follows:

$$\frac{\frac{\frac{g^{v+\delta_1}, T'_1(\varphi_2), \dots, T'_1(\varphi_N), q \Rightarrow f}{g^v, q \bullet g^{\bullet\delta_1}, T'_1(\varphi_2), \dots, T'_1(\varphi_N) \Rightarrow f} (\backslash_R)}{f \Rightarrow f \quad g^v, T'_1(\varphi_2), \dots, T'_1(\varphi_N) \Rightarrow (q \bullet g^{\bullet\delta_1}) \backslash f} (\backslash_L)}{\frac{g^v, T'_1(\varphi_2), \dots, T'_1(\varphi_N), ((q \bullet g^{\bullet\delta_1}) \backslash f) \backslash f \Rightarrow f}{g^{v+\delta_2}, g^{\bullet\delta_2} \backslash ((q \bullet g^{\bullet\delta_1}) \backslash f) \backslash f, T'_1(\varphi_2), \dots, T'_1(\varphi_N) \Rightarrow f} (\backslash_L)} (\backslash_L)}{g^u, t \backslash g^{\bullet\delta_2} \backslash ((q \bullet g^{\bullet\delta_1}) \backslash f) \backslash f, T'_1(\varphi_2), \dots, T'_1(\varphi_N), t \Rightarrow f} (\backslash_L)} t \Rightarrow t$$

This completes the proof for this case.

Case C. The derivation tree of $t(u)$ is of the form

$$\frac{\mathcal{T}_1 \quad \mathcal{T}_2}{t(v_1 + v_2)} (\varphi_i)$$

for some $\varphi_i = t(x + y) \leftarrow q_1(x), q_2(y) \in \mathcal{P}_2$. Without loss of generality, $i = 1$. Then \mathcal{T}_j is a derivation tree of the fact $q_j(v_j)$ for $j = 1, 2$ (such that $u = v_1 + v_2$), and $\text{Rule}(\mathcal{T}_1) \sqcup \text{Rule}(\mathcal{T}_2) = \{\varphi_2, \dots, \varphi_m\}$. By the induction hypothesis, the sequents $g^{v_j}, \Theta_j, q_j \Rightarrow f$ are derivable for $j = 1, 2$ where Θ_1, Θ_2 are multisets such that $\Theta_1, \Theta_2 = T'_1(\varphi_2), \dots, T'_1(\varphi_m)$. From these two sequents, one can derive the sequent $g^u, T'_1(\varphi_1), \dots, T'_1(\varphi_m), t \Rightarrow f$ as follows:

$$\frac{\frac{\frac{g^{v_2}, \Theta_2, q_2 \Rightarrow f}{g^{v_2}, \Theta_2 \Rightarrow q_2 \backslash f} (\backslash_R)}{f \Rightarrow f \quad g^{v_2}, \Theta_2, (q_2 \backslash f) \backslash f \Rightarrow f} (\backslash_L)}{\frac{g^{v_1}, \Theta_1, q_1 \Rightarrow f}{g^{v_1}, \Theta_1 \Rightarrow q_1 \backslash f} (\backslash_R)}{g^{v_1+v_2}, \Theta_1, \Theta_2, (q_1 \backslash f) \backslash (q_2 \backslash f) \backslash f \Rightarrow f} (\backslash_L)} (\backslash_L)}{g^u, t \backslash (q_1 \backslash f) \backslash (q_2 \backslash f) \backslash f, T'_1(\varphi_2), \dots, T'_1(\varphi_m), t \Rightarrow f} (\backslash_L)} t \Rightarrow t$$

It remains to note that $T'_1(\varphi_1) = T(\varphi_1) = t \backslash (q_1 \backslash f) \backslash (q_2 \backslash f) \backslash f$. \square

Lemma 4 *Let $\text{LP} \vdash \Gamma, A, s \Rightarrow B$. Then $\text{LP} \vdash \Gamma, s \backslash ((s \bullet A) \backslash B) \backslash B, s \Rightarrow B$.*

Proof

$$\frac{\frac{\frac{\Gamma, A, s \Rightarrow B}{\Gamma, s \bullet A \Rightarrow B} (\bullet_L)}{B \Rightarrow B \quad \Gamma \Rightarrow (s \bullet A) \backslash B} (\backslash_R)}{\frac{\Gamma, ((s \bullet A) \backslash B) \backslash B \Rightarrow B}{\Gamma, s \backslash ((s \bullet A) \backslash B) \backslash B, s \Rightarrow B} (\backslash_L)} (\backslash_L)} s \Rightarrow s$$

\square

Corollary 1 *If $\text{LP} \vdash \Gamma, A^{(i)}(\Theta), s \Rightarrow f$, then $\text{LP} \vdash \Gamma, F^{(i)}(\Theta), s \Rightarrow f$.*

4.1.1 Proofs of Lemma 1 and Theorem 10

Finally, we are ready to prove the main lemma (Lemma 1) and to show how Theorem 7 follows from it.

Proof of Lemma 1

Let $\text{tuple}(u') = (i_1, \dots, i_M)$. Then $F^{(u')}(\Theta_1, \dots, \Theta_M) = F^{(i_1)}(\Theta_1), \dots, F^{(i_M)}(\Theta_M)$. Let $\Theta_j = \{\theta_j^1, \dots, \theta_j^{l_j}\}$. Let $\mathcal{R} = \Theta_1 \sqcup \dots \sqcup \Theta_M$.

Statement 1 obviously implies 2 since each sequent derivable in LP is derivable in L*P.

To show that statement 2 implies statement 3 let us prove that derivability of the sequent

$$g^u, F^{(u')}(\Theta_1, \dots, \Theta_M), t \Rightarrow f \quad (4)$$

in L*P implies that there is a derivation tree of $t(u)$ in H with the multiset of rule occurrences equal to \mathcal{R} . According to Lemma 2, $g^{i_j}, T(\theta_j^1), \dots, T(\theta_j^{l_j}) \Rightarrow F^{(i_j)}(\Theta_j)$ is derivable in LP. Applying the cut rule to (4) and to these sequents, we conclude that the following sequent is derivable in L*P as well:

$$g^u, g_{i_1}, \dots, g_{i_M}, T(\theta_1^1), \dots, T(\theta_1^{l_1}), \dots, T(\theta_M^1), \dots, T(\theta_M^{l_M}), t \Rightarrow f$$

Note that $g^u, g_{i_1}, \dots, g_{i_M} = g^{u+u'}$. According to Proposition 11, $t(u+u')$ has a derivation tree in H with the multiset of rule occurrences equal to $\{\theta_1^1, \dots, \theta_1^{l_1}, \dots, \theta_M^1, \dots, \theta_M^{l_M}\} = \mathcal{R}$. The implication is proved.

Finally, let us prove that 3 implies 1. By assumption, there exists a derivation tree in H of $t(u+u')$ with the multiset \mathcal{R} of rule occurrences. It follows from Lemma 3 that the sequent

$$g^{u+u'}, T_1'(\theta_1^1), \dots, T_1'(\theta_1^{l_1}), \dots, T_1'(\theta_M^1), \dots, T_1'(\theta_M^{l_M}), t \Rightarrow f$$

is derivable in LP. Applying the rule (\bullet_L) several times we come up with the sequent

$$g^u, A^{(i_1)}(\Theta_1), \dots, A^{(i_M)}(\Theta_M), t \Rightarrow f.$$

Applying Corollary 1 several times we conclude that the sequent

$$g^u, F^{(i_1)}(\Theta_1), \dots, F^{(i_M)}(\Theta_M), t \Rightarrow f$$

is derivable in LP. This is exactly statement 1 of the lemma. \square

Proof of Theorem 10 Let $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$. Consider the grammar $\text{LBCG}_1(H)$, which is an LP-grammar. Assume that $w = b_1 \dots b_M$ where $b_j = a_{i_j} \in \Sigma$ for $j = 1, \dots, M$. Then $w \in L(\text{LBCG}_1(H))$ if and only if there exist multisets $\Theta_1, \dots, \Theta_M$ consisting of rules from \mathcal{P} such that $|\Theta_j| \leq C$ and such that the sequent

$$F^{(i_1)}(\Theta_1), \dots, F^{(i_M)}(\Theta_M) \Rightarrow s \setminus f$$

is derivable in LP. It is equiderivable with the sequent

$$F^{(i_1)}(\Theta_1), \dots, F^{(i_M)}(\Theta_M), s \Rightarrow f.$$

Lemma 1 tells us that the latter sequent is derivable in LP if and only if the fact $s(u')$ has a derivation tree in H such that the multiset of its rule occurrences is $\Theta_1 \sqcup \dots \sqcup \Theta_M$; here $u' \in \mathbb{N}^K$ is a vector such that $\text{tuple}(u')$ and $\{i_1, \dots, i_M\}$ coincide as multisets. The multiset $\{i_1, \dots, i_M\}$ contains $|w|_l$ occurrences of the number l (for $l = 1, \dots, \kappa$) and 0 occurrences of l if $\kappa < l \leq K$. Thus u' such that $\text{tuple}(u')$ coincides with i_1, \dots, i_M must be $\iota_K(\Psi(w))$.

Summing up, $w \in L(\text{LBCG}_1(H))$ if and only if there exist multisets $\Theta_1, \dots, \Theta_M$ consisting of rules from \mathcal{P} such that $|\Theta_j| \leq C$ and such that the fact $s(\iota_K(\Psi(w)))$ has a derivation tree in H with the multiset of its rule occurrences being $\Theta_1 \sqcup \dots \sqcup \Theta_M$. The latter is equivalent to the fact that $s(\iota_K(\Psi(w)))$ has a derivation tree of the size less than or equal to $CM = C|w|$. This is exactly equivalent to the statement that $\Psi(w) \in L(H)$, hence $L(\text{LBCG}_1(H)) = \Psi^{-1}(L(H))$.

It remains to notice that $L(\text{LBCG}_1(H)) = L(\text{LBCG}(H))$ since the difference between the two grammars is that some formulas in them are replaced by equivalent ones.

If $\text{LBCG}_1(H)$ is considered as an L*P-grammar, then the same reasonings hold. The only difference is that we exploit equivalence of statements 2 and 3 of Lemma 1 instead of that of statements 1 and 3. \square

4.1.2 Proof of Proposition 11

To prove Proposition 11 we need several technical lemmas.

Lemma 5 *If $L^*P \vdash \Pi \Rightarrow q$ where $\Pi \in \text{SFm}(\text{LPG}(H))^*$ consists of formulas from $\text{SFm}(\text{LPG}(H))$ and where $q \in Q$, then $\Pi = q$.*

Proof The proof is by induction on the size of a derivation in L^*P . The base case is trivial ($q \Rightarrow q$ is an axiom). To prove the induction step consider the last rule applied in a derivation of $\Pi \Rightarrow q$.

Case (\backslash_L) . The last rule application is of the form:

$$\frac{\Gamma, A \Rightarrow q \quad \Delta \Rightarrow B}{\Gamma, \Delta, B \backslash A \Rightarrow q} (\backslash_L)$$

Here $\Pi = \Gamma, \Delta, B \backslash A$. Let us apply the induction hypothesis to $\Gamma, A \Rightarrow q$ and thus conclude that $\Gamma = \emptyset, A = q$. Therefore, $B \backslash A = B \backslash q \in \text{SFm}(\text{LPG}(H))$. However, there are no formulas in $\text{SFm}(\text{LPG}(H))$ that are of the form $B \backslash q$. This is proved by inspection of Construction 1. Indeed, for each formula $D \backslash C$ from $\text{SFm}(\text{LPG}(H))$ it is the case that C includes either f or g^{δ_1} where $\delta_1 \neq \bar{0}$. Here we use the fact that H is input-sensitive. This leads us to a contradiction. Consequently, it is not the case that (\backslash_L) is the last rule application.

Case (\bullet_L) . The last rule application is of the form:

$$\frac{\Gamma, A, B \Rightarrow q}{\Gamma, A \bullet B \Rightarrow q} (\bullet_L)$$

By the induction hypothesis, $\Gamma, A, B = q$; however, there are at least two formulas in Γ, A, B . This is a contradiction, hence is not the case that (\bullet_L) is the last rule application.

Cases (\backslash_R) and (\bullet_R) are impossible since the formula in the succedent of the sequent is q , which is primitive. \square

This lemma implies the following one:

Lemma 6 (the lock and key lemma) *Let the sequent $\Pi \Rightarrow C$ be derivable in L^*P where Π consists of formulas from $\text{SFm}(\text{LPG}(H))$ and C belongs to this set as well. Let the last rule application be that of (\backslash_L) and let the major formula be of the form $q \backslash A$ for $q \in Q$. Then Π must contain q as a separate formula (i.e. $\Pi = \Pi', q$).*

Proof The last rule application in the derivation of $\Pi \Rightarrow C$ must be of the form:

$$\frac{\Gamma, A \Rightarrow C \quad \Delta \Rightarrow q}{\Gamma, \Delta, q \backslash A \Rightarrow C} (\backslash_L)$$

Here $\Gamma, \Delta, q \backslash A = \Pi$. Since Δ consists of formulas from $\text{SFm}(\text{LPG}(H))$, one can apply Lemma 5 and conclude that $\Delta = q$. In what follows, q is one of the formulas in Π . \square

We call this lemma the lock and key lemma because the primitive formula q reminds us a key that opens “a lock” $q \backslash A$; without the key, the lock cannot be opened.

We need another lemma of the same kind. To formulate it, let us introduce a function \mathcal{D} , which takes a multiset of formulas of LP and returns a multiset of formulas.

Definition 13 The function \mathcal{D} is defined as follows:

- $\mathcal{D}(p) = p$ ($p \in Pr$);
- $\mathcal{D}(B \setminus A) = B \setminus A$;
- $\mathcal{D}(A \bullet B) = \mathcal{D}(A), \mathcal{D}(B)$;
- $\mathcal{D}(\Gamma, \Delta) = \mathcal{D}(\Gamma), \mathcal{D}(\Delta)$.

Informally, we replace all outermost products \bullet in a multiset of formulas by commas. Let us also say that, if $g^{\bullet u} = g_{i_1} \bullet \dots \bullet g_{i_n}$ is a formula where $n = |u|$ and if $\pi \in S_n$, then $\pi(g^{\bullet u})$ is the formula $g_{i_{\sigma(1)}} \bullet \dots \bullet g_{i_{\sigma(n)}}$. In other words, we simply change the order of factors in $g^{\bullet u}$. Since the product is commutative, this leads to an equivalent formula $\pi(g^{\bullet u}) \Leftrightarrow g^{\bullet u}$ interchangeable with $g^{\bullet u}$.

Lemma 7 Let $L^*P \vdash \Pi \Rightarrow \pi(g^{\bullet h})$ where $\Pi \in \text{SFm}(\text{LPG}(H))^*$ consists of formulas from $\text{SFm}(\text{LPG}(H))$, $\vec{0} \neq h \in \mathbb{N}^K$, $n = |h|$, and π is a permutation from S_n . Then $\mathcal{D}(\Pi) = g^{\bullet h}$.

Proof The proof is again by induction on the size of a derivation of the sequent. The base case is trivial since in such a case we have a sequent of the form $g_i \Rightarrow g_i$ for some i . To prove the induction step let us consider the last rule applied. Again, there are several cases:

Case (\setminus_L). In this case, the last rule application has the form:

$$\frac{\Gamma, A \Rightarrow \pi(g^{\bullet h}) \quad \Delta \Rightarrow B}{\Gamma, \Delta, B \setminus A \Rightarrow \pi(g^{\bullet h})} (\setminus_L)$$

Applying the induction hypothesis we conclude that $\mathcal{D}(\Gamma, A) = g^v$ for some $v \in \mathbb{N}^K$. Consequently, A must be a product of several primitive formulas of the form g_i , i.e. $A = \tau(g^{\bullet w})$ for some $w \in \mathbb{N}^K$, $w \neq \vec{0}$, and for some permutation τ . Therefore, $B \setminus \tau(g^{\bullet w})$ belongs to $\text{SFm}(\text{LPG}(H))$. However, there are no formulas of such form in $\text{SFm}(\text{LPG}(H))$; indeed, for each formula $D \setminus C$ from $\text{SFm}(\text{LPG}(H))$ it is the case that C includes either f or some $q \in Q$. In what follows, the last rule application cannot be of the form (\setminus_L).

Case (\setminus_R). This is impossible since there are no divisions in $\pi(g^{\bullet h})$.

Case (\bullet_L). In this case, the last rule application is of the form

$$\frac{\Gamma, E_1, E_2 \Rightarrow \pi(g^{\bullet h})}{\Gamma, E_1 \bullet E_2 \Rightarrow \pi(g^{\bullet h})} (\bullet_L)$$

It suffices to notice that $\mathcal{D}(\Gamma, E_1 \bullet E_2) = \mathcal{D}(\Gamma, E_1, E_2)$. The induction hypothesis completes the proof for this case.

Case (\bullet_R). In this case, the last rule application is of the form

$$\frac{\Gamma_1 \Rightarrow \pi_1(g^{\bullet h_1}) \quad \Gamma_2 \Rightarrow \pi_2(g^{\bullet h_2})}{\Gamma_1, \Gamma_2 \Rightarrow \pi(g^{\bullet h})}$$

where $\pi(g^{\bullet h}) = \pi_1(g^{\bullet h_1}) \bullet \pi_2(g^{\bullet h_2})$ for some vectors h_1, h_2 (note that they belong to \mathbb{N}^K as well as h , they are not h 's coordinates); consequently, $h_1 + h_2 = h$. Applying the induction hypothesis we conclude that $\mathcal{D}(\Gamma_1) = g^{h_1}$, $\mathcal{D}(\Gamma_2) = g^{h_2}$. This implies that $\mathcal{D}(\Gamma_1, \Gamma_2) = g^{\bullet h}$. This concludes the proof. \square

We are ready to prove Proposition 11.

Proof of Proposition 11 We prove that derivability of the sequent $g^u, T(\zeta_1), \dots, T(\zeta_m), t \Rightarrow f$ implies that there is a derivation tree of $t(u)$ in H with the multiset of rule occurrences $\{\zeta_1, \dots, \zeta_m\}$. The proof is by induction on m .

The base case of the proof is trivial. Indeed, the sequent of interest is definitely not an axiom since we have t in the antecedent and $f \neq t$ in the succedent.

We proceed with proving the induction step. As usually, let us consider the last rule application in the derivation. Note that only the rule (\setminus_L) can be applied at the last step of the derivation of $g^u, T(\zeta_1), \dots, T(\zeta_m), t \Rightarrow f$ because all the formulas in this sequent that are not primitive are of the form $B \setminus A$. Without loss of generality, let $T(\zeta_1)$ be the major formula in the last rule application. There are three subcases:

Subcase a: $\zeta_1 = q(\nu) \in \mathcal{P}_0$. Then the last step of the derivation must be of the form

$$\frac{g^{u_2}, g^{\bullet\nu} \setminus f, \Theta_2 \Rightarrow f \quad g^{u_1}, \Theta_1 \Rightarrow q}{g^u, q \setminus g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m), t \Rightarrow f} (\setminus_L)$$

for some $u_1, u_2 \in \mathbb{N}^K$ and some Θ_1, Θ_2 such that $g^{u_1}, g^{u_2} = g^u$ (equivalently, $u_1 + u_2 = u$) and $\Theta_1, \Theta_2 = T(\zeta_2), \dots, T(\zeta_m), t$. This is simply how the rule (\setminus_L) looks like in general. Lemma 5 implies that $g^{u_1}, \Theta_1 = q$, hence $u_1 = \vec{0}$ (there are no formulas of the form g_i in the antecedent) and $\Theta_1 = q$. Since $\Theta_1, \Theta_2 = T(\zeta_2), \dots, T(\zeta_m), t$, it must be the case that $t = q$ and $\Theta_2 = T(\zeta_2), \dots, T(\zeta_m)$. Note also that $u_2 = u$ because $u_1 = \vec{0}$. Summarizing, the last rule application is of the form

$$\frac{g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f \quad q \Rightarrow q}{g^u, q \setminus g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m), q \Rightarrow f} (\setminus_L)$$

Now, let us investigate how the last rule application can look like in the derivation of the sequent $g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f$. Clearly, it must be an application of (\setminus_L) since all the formulas in the sequent are either primitive or of the form $B \setminus A$; besides the succedent is a primitive formula. Two possibilities exist:

1. The formula $T(\zeta_i)$ for some $i \in \{2, \dots, m\}$ is the major one in the last rule application of $g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f$. The formula $T(\zeta_i)$ is of the form $r \setminus A$ for some $r \in Q$ and for some A . Then, according to Lemma 6, the formula r must be contained in $g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m)$ as a separate formula, which is not the case. This leads us to a contradiction.
2. The formula $g^{\bullet\nu} \setminus f$ is major. Then the last rule application in the derivation of $g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f$ must be of the following form:

$$\frac{g^{u_3}, f, \Theta_3 \Rightarrow f \quad g^{u_4}, \Theta_4 \Rightarrow g^{\bullet\nu}}{g^u, g^{\bullet\nu} \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f} (\setminus_L)$$

Here $g^{u_3}, g^{u_4} = g^u$ (equivalently, $u_3 + u_4 = u$), and $\Theta_3, \Theta_4 = T(\zeta_2), \dots, T(\zeta_m)$.

Let us examine the sequent $g^{u_3}, f, \Theta_3 \Rightarrow f$. We want to prove that it is an axiom. Otherwise, it would be the conclusion of some rule application; more precisely, this rule application must be that of (\setminus_L) . Notice that all the formulas with division in this sequent are of the form $r \setminus A$ for some $r \in Q$. If the last rule application was (\setminus_L) , then its major formula would be of this form. According to Lemma 6, this would imply that r is contained in g^{u_3}, f, Θ_3 ; however, this is not the case. As a consequence, $g^{u_3}, f, \Theta_3 \Rightarrow f$ is the axiom $f \Rightarrow f$, hence $u_3 = \vec{0}$, $\Theta_3 = \emptyset$.

Now it remains to apply Lemma 7 to $g^{u_4}, \Theta_4 \Rightarrow g^{\bullet\nu}$, using which we conclude that $g^{u_4} = g^\nu$ and that $\Theta_4 = \emptyset$. Finally, we have $u = u_4 = \nu$, $q = t$, and

$T(\zeta_2), \dots, T(\zeta_m) = \emptyset$. Hence there is a derivation tree of $t(u)$ in H with the multiset of rule occurrences $\{\zeta_1\}$: $t(u) = q(\nu) = \zeta_1$ belongs to \mathcal{P}_0 .

Subcase b: $\zeta_1 = q(x + \delta_2) \leftarrow p(x + \delta_1) \in \mathcal{P}_1$. Then the last step of the derivation of $g^u, T(\zeta_1), \dots, T(\zeta_m), t \Rightarrow f$ must be of the form:

$$\frac{g^u, g^{\bullet\delta_2} \setminus (p \bullet g^{\bullet\delta_1}), T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f \quad q \Rightarrow q}{g^u, q \setminus g^{\bullet\delta_2} \setminus (p \bullet g^{\bullet\delta_1}), T(\zeta_2), \dots, T(\zeta_m), t \Rightarrow f} (\setminus_L)$$

and $t = q$. This is proved in the same way as for Subcase a (using Lemma 5).

The lock and key lemma implies that the major formula in the last rule application in the derivation of

$$g^u, g^{\bullet\delta_2} \setminus (p \bullet g^{\bullet\delta_1}), T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f \quad (5)$$

must be $g^{\bullet\delta_2} \setminus (p \bullet g^{\bullet\delta_1})$. Indeed, the remaining formulas are either primitive or they are of the form $r \setminus A$ for some $r \in Q$. However, there is no formula of the form $r \in Q$ in the antecedent. Ergo, the last rule application in the derivation of (5) must be of the form

$$\frac{g^{u_1}, p \bullet g^{\bullet\delta_1}, \Theta_1 \Rightarrow f \quad g^{u_2}, \Theta_2 \Rightarrow g^{\bullet\delta_2}}{g^u, g^{\bullet\delta_2} \setminus (p \bullet g^{\bullet\delta_1}), T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f} (\setminus_L)$$

for some $u_1, u_2 \in \mathbb{N}^K$ and Θ_1, Θ_2 such that $g^{u_1}, g^{u_2} = g^u$ (equivalently, $u_1 + u_2 = u$) and $T(\zeta_2), \dots, T(\zeta_m) = \Theta_1, \Theta_2$. According to Lemma 7, $g^{u_2}, \Theta_2 = g^{\delta_2}$, hence $u_2 = \delta_2$, $\Theta_2 = \emptyset$.

Since $g^{u_1}, p \bullet g^{\bullet\delta_1}, \Theta_1 \Rightarrow f$ is derivable, then so is $g^{u_1}, p, g^{\delta_1}, \Theta_1 \Rightarrow f$; indeed, the latter is obtained from the former by using the rule (\bullet_L^{-1}) several times, which is admissible in L*P. Since $T(\zeta_2), \dots, T(\zeta_m) = \Theta_1, \Theta_2 = \Theta_1$, the sequent $g^{u_1}, p, g^{\delta_1}, \Theta_1 \Rightarrow f$ equals $g^{u_1 + \delta_1}, T(\zeta_2), \dots, T(\zeta_m), p \Rightarrow f$. Let us apply the induction hypothesis to it; it yields that there is a derivation tree of $p(u_1 + \delta_1)$ in H with the multiset of rule occurrences $\{\zeta_2, \dots, \zeta_m\}$. Finally, applying the rule ζ_1 to $p(u_1 + \delta_1)$ we come up with the fact $q(u_1 + \delta_2) = q(u_1 + u_2) = q(u)$. Consequently, there is a derivation tree in H of the latter fact with the multiset of rule occurrences $\{\zeta_1, \dots, \zeta_m\}$. This concludes the proof for this case.

Subcase c: $\zeta_1 = q(x + y) \leftarrow p(x), r(y)$. Then $T(\zeta_1) = q \setminus (p \setminus f) \setminus (r \setminus f) \setminus f$, the last step of the derivation is of the form

$$\frac{g^u, (p \setminus f) \setminus (r \setminus f) \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f \quad q \Rightarrow q}{g^u, q \setminus (p \setminus f) \setminus (r \setminus f) \setminus f, T(\zeta_2), \dots, T(\zeta_m), t \Rightarrow f} (\setminus_L)$$

and $t = q$. This is proved in the same way as for Subcases a and b (using Lemma 5). Now, let us consider the last rule application in the derivation of the sequent $g^u, (p \setminus f) \setminus (r \setminus f) \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f$. We infer from the lock and key lemma (Lemma 6) that the major formula in this rule application cannot be $T(\zeta_i)$ for any $i = 2, \dots, m$. Thus the major formula must be $(p \setminus f) \setminus (r \setminus f) \setminus f$, ergo the last rule application in the derivation of $g^u, (p \setminus f) \setminus (r \setminus f) \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f$ must be of the form

$$\frac{g^{u_2}, (r \setminus f) \setminus f, \Theta_2 \Rightarrow f \quad g^{u_1}, \Theta_1 \Rightarrow p \setminus f}{g^u, (p \setminus f) \setminus (r \setminus f) \setminus f, T(\zeta_2), \dots, T(\zeta_m) \Rightarrow f} (\setminus_L)$$

for some $u_1, u_2 \in \mathbb{N}^K$ and some Θ_1, Θ_2 such that $u_1 + u_2 = u$ and $\Theta_1, \Theta_2 = T(\zeta_2), \dots, T(\zeta_m)$. Again, the lock and key lemma implies that the last rule application in the derivation of $g^{u_2}, (r \setminus f) \setminus f, \Theta_2 \Rightarrow f$ must be of the form

$$\frac{g^{u_4}, f, \Theta_4 \Rightarrow f \quad g^{u_3}, \Theta_3 \Rightarrow r \setminus f}{g^{u_2}, (r \setminus f) \setminus f, \Theta_2 \Rightarrow f} (\setminus_L)$$

for some $u_3, u_4 \in \mathbb{N}^K$ and some Θ_3, Θ_4 such that $u_3 + u_4 = u_2$ and $\Theta_3, \Theta_4 = \Theta_2$.

Now, let us look at the sequent $g^{u_4}, f, \Theta_4 \Rightarrow f$. If it is obtained after some rule application, then its major formula must be contained in Θ_4 , hence it must equal $T(\zeta_i)$ for some i . Each $T(\zeta_i)$ is of the form $x \setminus A$ for some $x \in Q$. The lock and key lemma (Lemma 6) implies that x must be in g^{u_4}, f, Θ_4 , which is not the case. Consequently, $g^{u_4}, f, \Theta_4 \Rightarrow f$ must be the axiom $f \Rightarrow f$. Therefore, $u_4 = \vec{0}$, $\Theta_4 = \emptyset$.

As we know, the sequents $g^{u_1}, \Theta_1 \Rightarrow p \setminus f$ and $g^{u_3}, \Theta_3 \Rightarrow r \setminus f$ are derivable in L*P. The rule (\setminus_R^{-1}) allows us to infer that the sequents $g^{u_1}, \Theta_1, p \Rightarrow f$ and $g^{u_3}, \Theta_3, r \Rightarrow f$ are derivable as well. Let us apply the induction hypothesis to these sequents and conclude that $p(u_1)$ and $r(u_3)$ can be derived in H ; moreover, the induction hypothesis tells us that the multisets of rule occurrences in some their derivation trees are $\{\zeta \mid T(\zeta) \in \Theta_1\}$ and $\{\zeta \mid T(\zeta) \in \Theta_3\}$ resp. Finally, from $p(u_1)$ and $r(u_3)$ we can obtain the fact $q(u_1 + u_3)$ using ζ_1 . The derivation tree of $q(u_1 + u_3)$ consists of two subtrees that are derivation trees for $p(u_1)$ and $r(u_3)$ connected together via the rule application of ζ_1 . It remains to notice that, since $u_4 = \vec{0}$, it holds that $q(u_1 + u_3) = q(u_1 + u_3 + u_4) = q(u_1 + u_2) = q(u)$. Hence we have proved that $q(u)$ is derivable in H . Moreover, we have proved that the multiset of rule occurrences in its derivation tree equals $\{\zeta \mid T(\zeta) \in \Theta_1\} \sqcup \{\zeta \mid T(\zeta) \in \Theta_3\} \sqcup \{\zeta_1\} = \{\zeta \mid T(\zeta) \in \Theta_1 \sqcup \Theta_3\} \sqcup \{\zeta_1\} = \{\zeta_1, \dots, \zeta_m\}$, which is what we wanted to prove. \square

4.2 From LP-Grammars to LRBVASSAMs

Section 4.1 was devoted to the proof of the fact that LBC-grammars are at least as expressive as LRBVASSAMs. In this subsection, we show that actually the two formalisms are equivalent. Namely, given an LP-grammar G , we shall present an LRBVASSAM $\text{LRB}(G)$ generating $\Psi(L(G))$. Note that LBC-grammars form a subclass of LP-grammars, so this result together with Theorem 10 imply the equivalence of LBC-grammars, LP-grammars and LRBVASSAMs.

The idea behind the construction of $\text{LRB}(G)$ we are going to introduce is the following. Let \mathcal{F} be a finite set of formulas such that $|\mathcal{F}| = c$. Let e_i be the i -th standard-basis vector in \mathbb{N}^c of the form $(0, \dots, 0, 1, 0, \dots, 0)$ where the i -th coordinate equals 1. We fix an enumeration $\text{num} : \mathcal{F} \rightarrow \{1, \dots, c\}$ and, given a sequent $A_1, \dots, A_n \Rightarrow B$ of LP where all $A_i, B \in \mathcal{F}$, we transform it into the fact $B(e_{\text{num}(A_1)} + \dots + e_{\text{num}(A_n)})$. For example, if $\mathcal{F} = \{p, q, r\}$ and $\text{num}(p) = 1$, $\text{num}(q) = 2$, then the sequent $p, p, q \Rightarrow r$ is transformed into the fact $r(2, 1, 0)$. Thus succedents of sequents become states of an LRBVASSAM and antecedents are encoded as vectors. It remains to translate each rule of LP into that of an LRBVASSAM. For example, the rule

$$\frac{\Pi \Rightarrow A \quad \Phi \Rightarrow B}{\Pi, \Phi \Rightarrow A \bullet B} (\bullet_R)$$

is transformed into the binary rule $(A \bullet B)(x + y) \leftarrow A(x), B(y)$ of LRBVASSAM. Indeed, the sum of vectors corresponds to the concatenation of antecedents. The same can be done with all the remaining rules except for the rule (\setminus_L) . It has two premises, so it must be encoded as a binary rule of LRBVASSAM; however, the antecedent of the conclusion of this rule is not simply the result of concatenation of the antecedents of the premises. To overcome this issue, we shall slightly modify the calculus LP in order to make the rule (\setminus_L) “truly binary”.

4.2.1 A Modification of the Commutative Lambek Calculus

Given a categorial grammar G which is either an LP-grammar or an L*P-grammar, let us define a variant of the calculus L*P denoted by L^*P_G and a variant of LP denoted by LP_G . The set Fm_{LP_G} of formulas of both L^*P_G and LP_G equals $Fm_{LP_G} = Fm_{LP} \mid \Sigma$. Recall that Σ is the alphabet; so its symbols are now treated as formulas. Sequents of L^*P_G are of one of the three forms: $\Pi \Rightarrow C$ or $\Pi \Rightarrow_1 C$, or $\Pi \Rightarrow_2 C$ (i.e. we introduce two more sequential arrows) where $C \notin \Sigma$.

The axioms and rules of L^*P_G are as follows:

$$\begin{array}{c} \frac{}{A \Rightarrow A} \text{ (ax)} \quad \frac{\Gamma, A \Rightarrow S}{\Gamma, a \Rightarrow S} \text{ (} a \triangleright A \text{)} \quad \text{for } a \triangleright A \\ \\ \frac{\Delta, D \Rightarrow C}{\Delta \Rightarrow_1 D \setminus C} \text{ (\widehat{1})} \quad \frac{\Pi \Rightarrow_1 C \setminus A \quad \Phi \Rightarrow D}{\Pi, \Phi \Rightarrow_2 (D \setminus C) \setminus A} \text{ (\widehat{L})} \quad \frac{\Delta \Rightarrow_2 D \setminus C}{\Delta, D \Rightarrow C} \text{ (\widehat{2})} \\ \\ \frac{\Pi, D \Rightarrow C}{\Pi \Rightarrow D \setminus C} \text{ (\setminus_R)} \quad \frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \bullet B \Rightarrow C} \text{ (\bullet_L)} \quad \frac{\Pi \Rightarrow A \quad \Phi \Rightarrow B}{\Pi, \Phi \Rightarrow A \bullet B} \text{ (\bullet_R)} \end{array}$$

It is required that all the formulas from $\Gamma, \Pi, \Delta, \Phi$ as well as formulas A, B, C, D must be from Fm_{LP} , i.e. they must not belong to Σ . The formula S is not an arbitrary one but it is the distinguished formula of the grammar G .

The role of new sequential arrows \Rightarrow_1 and \Rightarrow_2 is technical. In a nutshell, they help us to replace the rule (\setminus_L) with the “truly binary” one (\widehat{L}) and also to control the size of a derivation in L^*P_G .

The calculus LP_G is obtained from L^*P_G by requiring that $\Pi \neq \emptyset$ in the rule (\setminus_R) (Lambek’s restriction for LP_G). Equivalently, we can present the rule (\setminus_R) in the following form:

$$\frac{\Gamma, A, D \Rightarrow C}{\Gamma, A \Rightarrow D \setminus C} \text{ (\setminus_R)}$$

Presence of the formula A is needed only to guarantee that there is something in the antecedent of the conclusion. The multiset Γ now can be empty.

Let $\Xi = a_{i_1} \dots a_{i_m} \in \Sigma^*$. If $\Theta = T_1, \dots, T_m$ is a multiset of formulas such that $a_{i_j} \triangleright T_j$ for $j = 1, \dots, m$, then we write $\Xi \triangleright \Theta$. Clearly, if $\Xi_1 \triangleright \Theta_1$ and $\Xi_2 \triangleright \Theta_2$, then $\Xi_1 \Xi_2 \triangleright \Theta_1, \Theta_2$. Recall that we allow one to use commas as separators between symbols in a word. So, we consider $\Xi = a_{i_1}, \dots, a_{i_m}$ both as a word and as a multiset, depending on the context.

Lemma 8 *If the last rule application in a given derivation of a sequent is that of $(\widehat{2})$, then this derivation must be of the form*

$$\frac{\frac{\frac{\dots}{\Sigma, A \Rightarrow C}}{\Sigma \Rightarrow_1 A \setminus C} \text{ (\widehat{1})} \quad \frac{\dots}{\Phi \Rightarrow B}}{\frac{\Sigma, \Phi \Rightarrow_2 (B \setminus A) \setminus C}{\Sigma, \Phi, B \setminus A \Rightarrow C} \text{ (\widehat{2})}} \text{ (\widehat{L})} \tag{6}$$

Proof Let the last rule application look as follows:

$$\frac{\Delta \Rightarrow_2 D \setminus C}{\Delta, D \Rightarrow C} \widehat{(2)}$$

So, the sequent of interest is of the form $\Delta, D \Rightarrow C$. The only rule after which $\Delta \Rightarrow_2 D \setminus C$ can appear is $(\widehat{\setminus}_L)$. Hence $D = B \setminus A$ and the last step in the derivation of $\Delta \Rightarrow_2 D \setminus C$ is

$$\frac{\Sigma \Rightarrow_1 A \setminus C \quad \Phi \Rightarrow B}{\Sigma, \Phi \Rightarrow_2 (B \setminus A) \setminus C} \widehat{(\setminus}_L)$$

for some Σ, Φ such that $\Delta = \Sigma, \Phi$. Finally, the sequent $\Sigma \Rightarrow_1 A \setminus C$ can appear only as the result of a rule application of $(\widehat{1})$:

$$\frac{\Sigma, A \Rightarrow C}{\Sigma \Rightarrow_1 A \setminus C} \widehat{(1)}$$

□

Lemma 9 *Let $\Pi, \Xi \Rightarrow C$ be a sequent such that Π is a multiset with elements from Fm_{LP} ; $\Xi = a_{i_1}, \dots, a_{i_m}$; and $C \in \text{Fm}_{\text{LP}}$. Then:*

1. $\text{L}^*P_G \vdash \Pi, \Xi \Rightarrow C$ if and only if $\text{L}^*P \vdash \Pi, \Theta \Rightarrow C$ for some Θ such that $\Xi \triangleright \Theta$.
2. $\text{LP}_G \vdash \Pi, \Xi \Rightarrow C$ if and only if $\text{LP} \vdash \Pi, \Theta \Rightarrow C$ for some Θ such that $\Xi \triangleright \Theta$.

Proof Firstly, let us prove that $\text{L}^*P_G \vdash \Pi, \Xi \Rightarrow C$ implies $\text{L}^*P \vdash \Pi, \Theta \Rightarrow C$ for some appropriate Θ . This is done by induction on the size of a derivation in L^*P_G . The base case is trivial. To prove the induction step, consider the last rule application. It cannot be that of $(\widehat{1})$ or of $(\widehat{\setminus}_L)$ since the sequential arrow is \Rightarrow . If the last rule applied is either (\bullet_L) , (\bullet_R) or (\setminus_R) , then it suffices to apply the induction hypothesis to the premises and use the same rule application in L^*P .

Assume that the last rule application is that of $(\widehat{2})$. Lemma 8 says that then the derivation must be of the form

$$\frac{\frac{\frac{\dots}{\Sigma, A \Rightarrow C} \widehat{(1)} \quad \frac{\dots}{\Phi \Rightarrow B} \widehat{(\setminus}_L)}{\Sigma, \Phi \Rightarrow_2 (B \setminus A) \setminus C} \widehat{(\setminus}_L)}{\Sigma, \Phi, B \setminus A \Rightarrow C} \widehat{(2)}$$

for some Σ and Φ such that $\Sigma, \Phi, B \setminus A = \Pi, \Xi$. Obviously, $B \setminus A$ belongs to Π but not to Ξ . Thus $\Pi = \Pi', B \setminus A$.

Let $\Pi' = \Pi_\Sigma, \Pi_\Phi$ and $\Xi = \Xi_\Sigma, \Xi_\Phi$ such that $\Sigma = \Pi_\Sigma, \Xi_\Sigma$ and $\Phi = \Pi_\Phi, \Xi_\Phi$. By the induction hypothesis, $\text{L}^*P \vdash \Pi_\Sigma, A, \Theta_\Sigma \Rightarrow C$ and $\text{L}^*P \vdash \Pi_\Phi, \Theta_\Phi \Rightarrow B$ for $\Theta_\Sigma, \Theta_\Phi$ for which it is the case that $\Xi_\Sigma \triangleright \Theta_\Sigma$ and $\Xi_\Phi \triangleright \Theta_\Phi$.

From these sequents derivable in L^*P , we derive the following one:

$$\frac{\Pi_\Sigma, A, \Theta_\Sigma \Rightarrow C \quad \Pi_\Phi, \Theta_\Phi \Rightarrow B}{\Pi_\Sigma, \Pi_\Phi, \Theta_\Sigma, \Theta_\Phi, B \setminus A \Rightarrow C} \widehat{(\setminus}_L)$$

The resulting sequent equals $\Pi, \Theta_\Sigma, \Theta_\Phi \Rightarrow C$, and $\Xi = \Xi_\Sigma, \Xi_\Phi \triangleright \Theta_\Sigma, \Theta_\Phi = \Theta$. This is what we aimed to prove.

Another case is the last rule application being that of $(a \triangleright A)$. Without loss of generality, let the last rule application be of the form

$$\frac{\Pi, A, \Xi' \Rightarrow S}{\Pi, a_{i_1}, \Xi' \Rightarrow S} (a_{i_1} \triangleright A)$$

where $\Xi = a_{i_1}, \Xi'$, hence $\Xi' = a_{i_2}, \dots, a_{i_m}$. The induction hypothesis tells us that $L^*P \vdash \Pi, A, \Theta' \Rightarrow S$ for Θ' such that $\Xi' \triangleright \Theta'$. Define Θ as A, Θ' . Then $\Xi \triangleright \Theta$ and $L^*P \vdash \Pi, \Theta \Rightarrow S$. This completes the first part of the proof.

The converse statement, namely, that $L^*P_G \vdash \Pi, \Xi \Rightarrow C$ is implied by $L^*P \vdash \Pi, \Theta \Rightarrow C$, is proved as follows. Firstly, given a derivation of $\Pi, \Theta \Rightarrow C$ we transform each rule application in it in a sequence of rule applications in L^*P_G . The only non-trivial case is a rule application of (\backslash_L) . Given a rule application

$$\frac{\Sigma, A \Rightarrow C \quad \Phi \Rightarrow B}{\Sigma, \Phi, B \backslash A \Rightarrow C} (\backslash_L)$$

of the rule (\backslash_L) , we transform it into the sequence (6) of rule applications in L^*P_G . Hence we prove $\Pi, \Theta \Rightarrow C$ in L^*P_G . Secondly, it remains to apply rules of the form $(a \triangleright A)$ in order to derive $\Pi, \Xi \Rightarrow C$ from $\Pi, \Theta \Rightarrow C$.

The proof of the second statement is the same. We only have to take care about non-emptiness of sequents when applying the induction hypothesis. \square

4.2.2 Transformation of Commutative Lambek Grammars into LRBVASSAMs

Let us now proceed with the definition of the main construction that transforms an L^*P -grammar (or an LP-grammar) into an equivalent LRBVASSAM.

Construction 3

Assume that we are given the L^*P -grammar $G = \langle S, \triangleright \rangle$. We construct the LRBVASSAM $LRB^*(G) := \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, S, \kappa, K, F \rangle$ as follows:

- $Q := SFm^+(G) \cup \{(A, C)_j \mid C \in SFm^+(G), A \in SFm^-(G), j \in \{1, 2\}\}$.
- K equals $\kappa + |SFm^-(G)|$ (recall that $\kappa = |\Sigma|$). Hereinafter we fix a bijection $ind : \Sigma \cup SFm^-(G) \rightarrow \{1, \dots, K\}$ such that $ind(a_i) = i$ for $i = 1, \dots, \kappa$. Informally, ind enumerates negative subformulas of G by numbers from $\kappa + 1$ up to K and it enumerates elements of Σ according to their original numbering (namely, the number i is assigned to $a_i \in \Sigma$).
- \mathcal{P}_0 consists of nullary rules $A(e_{ind(A)})$ for $A \in SFm^-(G) \cap SFm^+(G)$.
- \mathcal{P}_1 consists of the following rules:
 1. $\rho_{A \bullet B, C} = C(x + e_{ind(A \bullet B)}) \leftarrow C(x + e_{ind(A)} + e_{ind(B)})$ for $A \bullet B \in SFm^-(G)$, $C \in SFm^+(G)$;
 2. $\rho_{D \backslash C}^* = (D \backslash C)(x) \leftarrow C(x + e_{ind(D)})$ for $D \backslash C \in SFm^+(G)$;
 3. $\rho_{D, C}^1 = (D, C)_1(x) \leftarrow C(x + e_{ind(D)})$ for $C \in SFm^+(G)$, $D \in SFm^-(G)$;
 4. $\rho_{D, C}^2 = C(x + e_{ind(D)}) \leftarrow (D, C)_2(x)$ for $C \in SFm^+(G)$, $D \in SFm^-(G)$;
 5. $\rho_{a_i}^A = S(x + e_i) \leftarrow S(x + e_{ind(A)})$ for A such that $a_i \triangleright A$.
- \mathcal{P}_2 consists of the following rules:
 1. $\rho_{A \bullet B} = (A \bullet B)(x + y) \leftarrow A(x), B(y)$ for $A \bullet B \in SFm^+(G)$;
 2. $\rho_{D \backslash C, A} = ((D \backslash C), A)_2(x + y) \leftarrow (C, A)_1(x), D(y)$ for $D \backslash C \in SFm^-(G)$ and $A \in SFm^+(G)$.

- S , which is the distinguished formula in G , is also the distinguished state of the new grammar.
- $F = 6 \cdot \max_{A \in \text{Fm}(G)} |A| + 1$.

If G is an LP-grammar, then let us define the LRBVASSAM $\text{LRB}(G)$ in the same way as $\text{LRB}^*(G)$ with the only difference that we replace the rules $\rho_{D \setminus C}^*$ by the rules

$$\rho_{D \setminus C}^A = (D \setminus C)(x + e_{\text{ind}(A)}) \leftarrow C(x + e_{\text{ind}(D)} + e_{\text{ind}(A)})$$

for $D \setminus C \in \text{SFm}^+(G)$ and $A \in \text{SFm}^-(G)$.

Remark 10 The size of the grammar $\text{LRB}^*(G)$ (or $\text{LRB}(G)$) is polynomial w.r.t. the size of G .

Example 7 Consider the LP-grammar G_0 from Example 1. Firstly, let us look at the sets of its positive and negative subformulas:

- $\text{SFm}^+(G_0) = \{p \setminus p, p, q\}$;
- $\text{SFm}^-(G_0) = \{q \setminus p \setminus p, q, q \bullet q, p \setminus p, p\}$.

The LRBVASSAM $\text{LRB}^*(G) := \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, S, \kappa, K, F \rangle$ is defined as follows.

- $Q = \{p \setminus p, p, q\} \sqcup \{(A, C)_i \mid C \in \text{SFm}^+(G_0), A \in \text{SFm}^-(G_0), i \in \{1, 2\}\}$. The total number of states equals $3 + 2 \cdot 3 \cdot 5 = 33$. For example, $(q \bullet q, p \setminus p)_1 \in Q$.
- $\kappa = 2$ (recall that the alphabet Σ equals $\{a, b\}$ in Example 1).
- $K = 2 + 5 = 7$.
- Let ind have the following definition: $\text{ind}(q \setminus p \setminus p) = 3$, $\text{ind}(q) = 4$, $\text{ind}(q \bullet q) = 5$, $\text{ind}(p \setminus p) = 6$, $\text{ind}(p) = 7$.
- $S = S_0 = p \setminus p$.
- $F = 6 \cdot 5 + 1 = 31$.
- Nullary rules are listed below.
 - $(p \setminus p)(0, 0, 0, 0, 0, 1, 0)$;
 - $p(0, 0, 0, 0, 0, 0, 1)$;
 - $q(0, 0, 0, 1, 0, 0, 0)$.
- Unary rules are listed below.
 - $C(x + (0, 0, 0, 0, 1, 0, 0)) \leftarrow C(x + (0, 0, 0, 2, 0, 0, 0))$ for $C \in \{p \setminus p, p, q\}$;
 - $(p \setminus p)(x) \leftarrow p(x + (0, 0, 0, 0, 0, 0, 1))$;
 - $(D, C)_1(x) \leftarrow C(x + e_{\text{ind}(D)})$ for $C \in \text{SFm}^+(G_0), D \in \text{SFm}^-(G_0)$; for example, the rule $(q \bullet q, p \setminus p)_1(x) \leftarrow (p \setminus p)(x + (0, 0, 0, 0, 1, 0, 0))$ is one of these unary rules;
 - $C(x + e_{\text{ind}(D)}) \leftarrow (D, C)_2(x)$ for $C \in \text{SFm}^+(G_0), D \in \text{SFm}^-(G_0)$; for example, the rule $(p \setminus p)(x + (0, 0, 0, 0, 1, 0, 0)) \leftarrow (q \bullet q, p \setminus p)_2(x)$ is one of these unary rules;
 - $(p \setminus p)(x + (1, 0, 0, 0, 0, 0, 0)) \leftarrow (p \setminus p)(x + (0, 0, 1, 0, 0, 0, 0))$;
 - $(p \setminus p)(x + (0, 1, 0, 0, 0, 0, 0)) \leftarrow (p \setminus p)(x + (0, 0, 0, 1, 0, 0, 0))$;
 - $(p \setminus p)(x + (0, 1, 0, 0, 0, 0, 0)) \leftarrow (p \setminus p)(x + (0, 0, 0, 0, 1, 0, 0))$.
- Binary rules are of the form $((D \setminus C), A)_2(x + y) \leftarrow (C, A)_1(x), D(y)$ for $D \setminus C \in \text{SFm}^-(G_0)$ and $A \in \text{SFm}^+(G_0)$. For example, one of these binary rules is $(q \setminus p \setminus p, p)_2(x + y) \leftarrow (p \setminus p, p)_1(x), q(y)$.

Theorem 12

1. Let G be an L^*P -grammar. Then $L(\text{LRB}^*(G)) = \Psi(L(G)) \setminus \{\vec{0}\}$.
2. Let G be an LP -grammar. Then $L(\text{LRB}(G)) = \Psi(L(G))$.

Definition 14 Given the multiset $\Pi = A_1, \dots, A_n$, let $\text{vec}[\Pi] := e_{\text{ind}(A_1)} + \dots + e_{\text{ind}(A_n)}$. Clearly, this definition does not depend on the order of formulas in A_1, \dots, A_n .

Definition 15 In this definition, we are going to describe a translation denoted by τ^* . Its input is a derivation tree of the sequent $\Pi \Rightarrow C$ in L^*P_G ; the translation τ^* transforms it into a derivation tree of the fact $C(\text{vec}[\Pi])$ in $\text{LRB}^*(G)$ as follows:

1. Each sequent $\Phi \Rightarrow B$ within this derivation tree is translated into the fact $B(\text{vec}[\Phi])$.
2. We replace each axiom or rule application in L^*P_G within this derivation tree by a rule application in $\text{LRB}(G)$ as follows:

$$\begin{array}{l}
\overline{X \Rightarrow X} \text{ (ax)} \qquad X(\text{vec}[X]) = X(e_{\text{ind}(X)}) \\
\\
\frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \bullet B \Rightarrow C} \text{ (}\bullet_L\text{)} \qquad \frac{C(\text{vec}[\Gamma] + e_{\text{ind}(A)} + e_{\text{ind}(B)})}{C(\text{vec}[\Gamma] + e_{\text{ind}(A \bullet B)})} (\rho_{A \bullet B, C}) \\
\\
\frac{\Pi, D \Rightarrow C}{\Pi \Rightarrow D \setminus C} (\setminus_R) \qquad \frac{C(\text{vec}[\Pi] + e_{\text{ind}(D)})}{(D \setminus C)(\text{vec}[\Pi])} (\rho_{D \setminus C}^*) \\
\\
\frac{\Delta, D \Rightarrow C}{\Delta \Rightarrow_1 D \setminus C} (\widehat{1}) \qquad \frac{C(\text{vec}[\Delta] + e_{\text{ind}(D)})}{(D, C)_1(\text{vec}[\Delta])} (\rho_{D, C}^1) \\
\\
\frac{\Delta \Rightarrow_2 D \setminus C}{\Delta, D \Rightarrow C} (\widehat{2}) \qquad \frac{(D, C)_2(\text{vec}[\Delta])}{C(\text{vec}[\Delta] + e_{\text{ind}(D)})} (\rho_{D, C}^2) \\
\\
\frac{\Gamma, A \Rightarrow S}{\Gamma, a_i \Rightarrow \overline{S}} (a_i \triangleright A) \qquad \frac{S(\text{vec}[\Gamma] + e_{\text{ind}(A)})}{S(\text{vec}[\Gamma] + e_i)} (\rho_{a_i}^A) \\
\\
\frac{\Pi \Rightarrow A \quad \Phi \Rightarrow B}{\Pi, \Phi \Rightarrow A \bullet B} (\bullet_R) \qquad \frac{A(\text{vec}[\Pi]) \quad B(\text{vec}[\Phi])}{A \bullet B(\text{vec}[\Pi] + \text{vec}[\Phi])} (\rho_{A \bullet B}) \\
\\
\frac{\Pi \Rightarrow_1 C \setminus A \quad \Phi \Rightarrow D}{\Pi, \Phi \Rightarrow_2 (D \setminus C) \setminus A} (\widehat{L}) \qquad \frac{(C, A)_1(\text{vec}[\Pi]) \quad D(\text{vec}[\Phi])}{((D \setminus C), A)_2(\text{vec}[\Pi] + \text{vec}[\Phi])} (\rho_{D \setminus C, A})
\end{array}$$

Similarly, we can define a translation τ of a derivation tree of the sequent $\Pi \Rightarrow C$ in LP_G into a derivation tree of the fact $C(\text{vec}[\Pi])$ in $\text{LRB}(G)$. Since L^*P_G and LP_G differ only in the definition of the rule (\setminus_R) , the only difference in the definition of τ comparing to τ^* is that each rule application of (\setminus_R) is translated into the following rule application:

$$\frac{\Gamma, A, D \Rightarrow C}{\Gamma, A \Rightarrow D \setminus C} (\setminus_R) \quad \frac{C \left(\text{vec}[\Gamma] + e_{\text{ind}(A)} + e_{\text{ind}(D)} \right)}{(D \setminus C) \left(\text{vec}[\Gamma] + e_{\text{ind}(A)} \right)} (\rho_{D \setminus C}^A)$$

Clearly, the rule applications presented in the rightmost column are correct according to the definition of the LRBVASSAM $\text{LRB}^*(G)$ (or $\text{LRB}(G)$).

Lemma 10

1. The translation τ^* translates a correct derivation in L^*P_G into a correct one in $\text{LRB}^*(G)$. Vice versa, each correct derivation in $\text{LRB}^*(G)$ is a translation of a correct derivation in L^*P_G .
2. The translation τ translates a correct derivation in LP_G into a correct one in $\text{LRB}(G)$. Vice versa, each correct derivation in $\text{LRB}(G)$ is a translation of a correct derivation in LP_G .

Proof As we noticed, the fact that the translation τ^* (the translation τ) transforms a correct derivation in L^*P_G (in LP_G) into a correct one in $\text{LRB}^*(G)$ (in $\text{LRB}(G)$) is straightforward. To prove the converse, note that, since each fact derivable in $\text{LRB}^*(G)$ (or in $\text{LRB}(G)$) is of the form $A(v)$ where $v \in \mathbb{N}^K$, it holds that $v = \text{vec}[\Pi, \Xi]$ for some Π consisting of formulas from Fm_{LP_G} and some Ξ consisting of elements of Σ . Then, it suffices to compare Construction 3 with Definition 15 and observe that each application of a rule from $\text{LRB}^*(G)$ (from $\text{LRB}(G)$ resp.) must be of one of the forms present in its rightmost column. Thus each rule application in $\text{LRB}^*(G)$ (in $\text{LRB}(G)$) is a translation of a correct rule application in L^*P_G (in LP_G). \square

Lemma 11 *Let $\text{L}^*\text{P}_G \vdash A_1, \dots, A_n \Rightarrow B$ for some formulas $A_i, B \in \text{Fm}_{\text{LP}}$. Then the size of any derivation tree of this sequent in L^*P_G is less than or equal to $3(|A_1| + \dots + |A_n| + |B|)$.*

Proof The proof is by induction on the size of a given derivation tree of $A_1, \dots, A_n \Rightarrow B$.

If the sequent is an axiom, then the statement obviously holds. To prove the induction step, consider the last rule application in the derivation.

If it is that of (\setminus_R) , then it is of the form

$$\frac{A_1, \dots, A_n, D \Rightarrow C}{A_1, \dots, A_n \Rightarrow D \setminus C} (\setminus_R)$$

The size of the derivation subtree starting from the premise $A_1, \dots, A_n, D \Rightarrow C$ is, by induction hypothesis, of the size not greater than $3(|A_1| + \dots + |A_n| + |D| + |C|)$. Thus the size of the whole derivation tree does not exceed $3(|A_1| + \dots + |A_n| + |D| + |C|) + 1 \leq 3(|A_1| + \dots + |A_n| + |D| + |C| + 1) = 3(|A_1| + \dots + |A_n| + |D \setminus C|)$. Similarly, one considers the rules (\bullet_L) and (\bullet_R) .

Assume that the last application is that of the rule $(\widehat{2})$. Lemma 8 implies that the last steps in the derivation are as follows:

$$\frac{\frac{\Sigma, C \Rightarrow B}{\Sigma \Rightarrow_1 C \setminus B} (\widehat{1}) \quad \Phi \Rightarrow D}{\frac{\Sigma, \Phi \Rightarrow_2 (D \setminus C) \setminus B}{\Sigma, \Phi, D \setminus C \Rightarrow B} (\widehat{L})} (\widehat{2})$$

Here $\Sigma, \Phi, D \setminus C = A_1, \dots, A_n$. Let $\Sigma = B_1, \dots, B_x$, $\Phi = C_1, \dots, C_y$. By the induction hypothesis, the size of the derivation tree of $\Sigma, C \Rightarrow B$ is less than or equal to $3(|B_1| + \dots + |B_x| + |C| + |B|)$ and that of the derivation tree of $\Phi \Rightarrow D$ is less than or equal to $3(|C_1| + \dots + |C_y| + |D|)$. Hence the size of the whole derivation tree does not exceed $3(|B_1| + \dots + |B_x| + |C| + |B| + |C_1| + \dots + |C_y| + |D|) + 3 = 3(|A_1| + \dots + |A_n| + |B|)$. \square

Lemma 12 *The sequent $\Gamma, a_i \Rightarrow R$ has a derivation tree in L^*P_G (in LP_G) of the size $n + 1$ if and only if there is A such that $a_i \triangleright A$ and such that the sequent $\Gamma, A \Rightarrow R$ has a derivation tree in L^*P_G (in LP_G) of the size n .*

Proof The proof of the “only if” direction is by induction on the size $n + 1$ of a derivation tree of $\Gamma, a_i \Rightarrow R$ in the calculus of interest. This sequent cannot be an axiom because axioms do not involve constants from Σ . Assume that the last rule application is that of (\bullet_L) , or (\bullet_R) , or (\setminus_R) , or $(\widehat{1})$, or $(\widehat{2})$, or (\widehat{L}) . In general, this rule application can be represented as follows:

$$\frac{\Delta, a_i \Rightarrow T \quad \Omega}{\Gamma, a_i \Rightarrow R} (r) \quad (7)$$

Here $\Delta, a_i \Rightarrow T$ is some sequent with the derivation tree of the size $l + 1$ for some l ; (r) is the rule name; Ω is either another premise or nothing. The induction hypothesis for $\Delta, a_i \Rightarrow T$ tells us that $\Delta, A \Rightarrow T$ has a derivation tree of the size l for some A such that $a_i \triangleright A$. Then let us construct a derivation tree of $\Gamma, A \Rightarrow R$ as follows:

$$\frac{\Delta, A \Rightarrow T \quad \Omega}{\Gamma, A \Rightarrow R} (r)$$

Hence we simply change a_i by A both in the premise and in the conclusion. Clearly, the size of this derivation tree is that of (7) decreased by 1.

The remaining case is where the last rule application is that of $(a_i \triangleright A)$:

$$\frac{\Gamma, A \Rightarrow S}{\Gamma, a_i \Rightarrow S} (a_i \triangleright A)$$

Its premise is derivable, and it has a desired form. The size of its derivation tree is n .

To prove the “if” direction, it suffices to take a derivation tree \mathcal{T} of $\Gamma, A \Rightarrow R$ of the size n and apply the rule $(a_i \triangleright A)$. \square

Proof of Theorem 12 Let us denote $\max_{A \in \mathbb{F}_m(G)} |A|$ by μ . Then $F = 6\mu + 1$.

A vector $v \in \mathbb{N}^k$ belongs to $L(\text{LRB}^*(G))$ (to $L(\text{LRB}(G))$) if and only if $S(\iota_K(v))$ has a derivation tree in $\text{LRB}^*(G)$ (in $\text{LRB}(G)$ resp.) with the size not greater than $F|v|$. Note that $v = \Psi(a_{i_1} \dots a_{i_m})$ where $i_1, \dots, i_m = \text{tuple}(v)$. According to Definition 15, $S(\iota_K(v))$ is the translation of the sequent $a_{i_1}, \dots, a_{i_m} \Rightarrow S$.

Firstly, it follows from Lemma 10 that there is one-to-one correspondence between derivations of $a_{i_1}, \dots, a_{i_m} \Rightarrow S$ in L^*P_G (in LP_G) and those of $S(\iota_K(v))$ in $LRB^*(G)$ (in $LRB(G)$ resp.). Moreover, both τ^* and τ preserve sizes of derivations. Ergo, $S(\iota_K(v))$ has a derivation tree of the size $d \leq F|v| = F \cdot m$ if and only if $a_{i_1}, \dots, a_{i_m} \Rightarrow S$ has a derivation tree of the size $d \leq F \cdot m$ (in corresponding formalisms). Lemma 12 entails that the latter is equivalent to the fact that $A_1, \dots, A_m \Rightarrow S$ has a derivation tree of the size $d - m \leq (F - 1)m = 6\mu \cdot m$ for some A_1, \dots, A_m such that $a_{i_j} \triangleright A_j$. But note that, if $A_1, \dots, A_m \Rightarrow S$ is derivable for such A_1, \dots, A_m , then, according to Lemma 11, each its derivation does not exceed $3(|A_1| + \dots + |A_m| + |B|) \leq 3\mu(m + 1) \leq 6\mu \cdot m$ (if $m \geq 1$); hence the restriction on the size of a derivation tree is always satisfied and thus redundant. If $m = 0$, then $v = \vec{0}$, so it belongs neither to $L(LRB^*(G))$ nor to $L(LRB(G))$. Note that $\Psi^{-1}(\vec{0}) = \Lambda$ is the empty word, and it does not belong to $L(G)$ if G is an LP-grammar.

Summarizing the above reasonings, $v \in \mathbb{N}^\kappa$, $v \neq \vec{0}$ belongs to $L(LRB^*(G))$ (to $L(LRB(G))$) if and only if $A_1, \dots, A_m \Rightarrow S$ is derivable in L^*P_G (in LP_G resp.) for some A_1, \dots, A_m such that $a_{i_j} \triangleright A_j$. According to Lemma 9, this is equivalent to derivability of the same sequent in L^*P (in LP resp.) and thus is equivalent to the fact that $a_{i_1} \dots a_{i_m} = \Psi^{-1}(v) \in L(G)$. □

5 LRBVASSAM Generating a Non-Semilinear Set

In the previous section, we have shown that LBC-grammars are equivalent to LRBVASSAMs (Theorems 10, 12). This shall be used to prove Theorem 3 and Theorem 4. In this section, we prepare the second ingredient: we present an LBC-grammar generating the language QL^{perm} where $QL = \{a^l b^n \mid l, n \in \mathbb{N}, 1 \leq n, l \leq n^2\}$. The Parikh image $\Psi(QL^{perm})$ of this language equals $QS = \{(l, n) \mid 1 \leq n, l \leq n^2\}$. If we present an LRBVASSAM generating QS , then we shall prove both theorems of interest.

Construction 4 Let $QH = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$ be the following LRBVASSAM:

- $K = 4$, $\kappa = 2$.
- $Q = \{q, r_1, r_2, s\}$.
- \mathcal{P}_0 includes only $\zeta_0 = q(0, 0, 0, 0)$.
- \mathcal{P}_1 consists of unary rules
 1. $\zeta_1 = q(x + (0, 1, 0, 1)) \leftarrow q(x + (0, 0, 0, 0))$,
 2. $\zeta_2 = r_1(x + (0, 0, 0, 0)) \leftarrow q(x + (0, 0, 0, 0))$,
 3. $\zeta_3 = r_1(x + (1, 0, 1, 0)) \leftarrow r_1(x + (0, 1, 0, 0))$,
 4. $\zeta_4 = r_2(x + (0, 0, 0, 0)) \leftarrow r_1(x + (0, 0, 0, 1))$,
 5. $\zeta_5 = r_2(x + (0, 1, 0, 0)) \leftarrow r_2(x + (0, 0, 1, 0))$,
 6. $\zeta_6 = r_1(x + (0, 0, 0, 0)) \leftarrow r_2(x + (0, 0, 0, 0))$,
 7. $\zeta_7 = s(x + (0, 0, 0, 0)) \leftarrow r_1(x + (0, 0, 0, 0))$.
- \mathcal{P}_2 is empty.
- $C = 5$.

Note that we do not use binary rules.

Theorem 13 *The set $L(QH)$ equals QS .*

Proof First of all, let us agree on the notation used in the proof. We put $\alpha = (1, 0, 0, 0)$, $\beta = (0, 1, 0, 0)$, $\rho = (0, 0, 1, 0)$, $\sigma = (0, 0, 0, 1)$. Be aware that the first and the second coordinates in this LRBVASSAM are the main memory coordinates, while the third and the fourth ones are the additional memory coordinates.

The LRBVASSAM QH has only nullary and unary rules. Consequently, any derivation tree in this grammar can be represented simply as a sequence of facts $t_1(u_1), \dots, t_n(u_n)$ such that each $t_1(u_1)$ is the nullary rule $q(\vec{0})$ and for $i = 2, \dots, n$ the fact $t_i(u_i)$ is obtained from $t_{i-1}(u_{i-1})$ by means of some unary rule from \mathcal{P}_1 . A derivation tree is completely defined by a sequence of rule applications.

Let us say that a derivation in QH is *completely typical* if it is of the following form for some $n \in \mathbb{N}$:

1. It starts with the axiom $\zeta_0 = q(\vec{0})$. The rule ζ_1 is applied n times. The result is

$$q(n(\beta + \sigma)) = q(0, n, 0, n).$$

2. The rule ζ_2 is applied. The result is

$$r_1(n(\beta + \sigma)) = r_1(0, n, 0, n).$$

3. The following part of the derivation consists of n blocks. Below the index i is iterated over the range of integers from 1 to n (it is incremented by 1). Let $i = 1$, $x_1 := 0, y_1 := n$; then $r_1(n(\beta + \sigma)) = r_1(x_i\alpha + y_i\beta + (n - y_i)\rho + (n + 1 - i)\sigma)$. The definition of x_i for arbitrary i shall be presented a bit later.

(a) At the beginning of the i -th iteration we have the fact

$$r_1(x_i\alpha + y_i\beta + (n - y_i)\rho + (n + 1 - i)\sigma).$$

(b) The rule ζ_3 is applied $l_i \leq y_i$ times. The result is

$$r_1((x_i + l_i)\alpha + (y_i - l_i)\beta + (n - y_i + l_i)\rho + (n + 1 - i)\sigma).$$

(c) The rule ζ_4 is applied. The result is

$$r_2((x_i + l_i)\alpha + (y_i - l_i)\beta + (n - y_i + l_i)\rho + (n - i)\sigma).$$

(d) The rule ζ_5 is applied $l'_i \leq n - y_i + l_i$ times. The result is

$$r_2((x_i + l_i)\alpha + (y_i - l_i + l'_i)\beta + (n - y_i + l_i - l'_i)\rho + (n - i)\sigma).$$

(e) The rule ζ_6 is applied. The result is

$$r_1((x_i + l_i)\alpha + (y_i - l_i + l'_i)\beta + (n - y_i + l_i - l'_i)\rho + (n - i)\sigma).$$

This is the last step of the iteration, so let $x_{i+1} := x_i + l_i$, $y_{i+1} := y_i - l_i + l'_i$.
 Finally, i is increased by 1 and the next iteration of step 3 starts.
 The result of all the n iterations is the fact

$$r_1(x_{n+1}\alpha + y_{n+1}\beta + (n - y_{n+1})\rho).$$

4. The rule ζ_7 is applied. The result is

$$s(x_{n+1}\alpha + y_{n+1}\beta + (n - y_{n+1})\rho).$$

Note that, since $r_1(x_i\alpha + y_i\beta + (n - y_i)\rho + (n + 1 - i)\sigma)$ is a correct fact, all its coordinates are non-negative, hence $y_i \geq 0$ and $n - y_i \geq 0$; equivalently, $0 \leq y_i \leq n$.

We say that a derivation is *typical* if it is an initial part (a prefix) of a completely typical derivation. Our claim is that each derivation in G is typical. This is straightforwardly proved by induction on the length of a derivation; the proof is a simple consideration of which rule can be the next one at each step of a completely typical derivation. In fact, the description of a completely typical derivation exhausts all possible sequences of rule applications in H .

Let $(l, m) \in L(QH)$; equivalently, $s(\iota_4(l, m)) = s(l, m, 0, 0) = s(l\alpha + m\beta)$ has a derivation in QH of the size less than or equal to $C(m + l) = 5(m + l)$. If this derivation exists, it is typical. In fact, it must be completely typical since we reach the state s only at the last step of a completely typical derivation. Consider this typical derivation. For it, it must be the case that $y_{n+1} = m$ and $n - y_{n+1} = 0$, hence $y_{n+1} = n = m$. Most importantly, $l = x_{n+1} = l_1 + \dots + l_n \leq y_1 + \dots + y_n \leq n \cdot n = n^2 = m^2$. Be aware that $1 \leq m$, since otherwise $l = m = 0$ and the size of a derivation of $s(l\alpha + m\beta)$ would be less than or equal to 0, which is impossible.

Conversely, if $1 \leq n$ and $l \leq n^2$, then let us show that the fact $s(l\alpha + n\beta)$ has a derivation in QH of size $\leq 5(n + l)$. Indeed, a derivation of interest is the completely typical one with parameters $l_i = n$ (for $i \leq \lfloor l/n \rfloor$), $l_i = l - \lfloor l/n \rfloor n$ (for $i = \lfloor l/n \rfloor + 1$), $l_i = 0$ (for $i > \lfloor l/n \rfloor + 1$); $l'_i = l_i$ (for all i). In this case, the result of the derivation is $s(x_{n+1}\alpha + y_{n+1}\beta + (n - y_{n+1})\rho)$ where $x_{n+1} = l_1 + \dots + l_n = n\lfloor l/n \rfloor + (l - \lfloor l/n \rfloor n) = l$, $y_{n+1} = n$. The size of this derivation can be computed by summing the number of rule applications at each stage.

1. At stage 1, there are n rule applications.
2. At stage 2, there is 1 rule application.
3. At the i -th iteration of stage 3, the number of rule applications equals $l_i + 1 + l'_i + 1 = 2l_i + 2$. The sum of these numbers over all iterations equals $2l + 2n$.
4. At stage 4, there is 1 rule application.

In total, the number of rule applications equals $3n + 2l + 2$. Since $1 \leq n$, the inequality $3n + 2l + 2 \leq 5(n + l)$ holds as expected. \square

6 Proofs of Theorems from Section 3

Finally, we prove all the theorems from Section 3 (except for Theorem 8, which shall be proved in Section 7). Before proofs of the theorems, we recall their statements.

Theorem 3 There exists an LBC-grammar QG such that:

1. QG generates the language QL^{perm} .

2. If the grammar QG is considered as an L^*P -grammar, then it generates the same language QL^{perm} .

Proof of Theorem 3

The desired LP-grammar QG is the grammar $LBCG(\widetilde{QH})$ where \widetilde{QH} is the input-sensitive LRBVASSAM obtained from QH by applying Proposition 9; QH is defined by Construction 4. Theorem 13 says that $L(QH) = QS$; then, by Theorem 10, the language $L(LBCG(\widetilde{QH}))$ equals $\Psi^{-1}(QS) = QL^{perm}$.

The second statement of Theorem 10 implies the second part of the statement of the present theorem. \square

Theorem 4 The set of languages generated by LBC-grammars properly contains the set of permutation closures of context-free languages.

Proof of Theorem 4

Buszkowski in [5] proves that languages generated by LBC-grammars contain permutation closures of context-free languages (without the empty word, to be precise).

Given the LRBVASSAM QH , let us consider the LBC-grammar $LBCG(QH)$. According to Theorem 10, it generates the language $\Psi^{-1}(L(QH)) = \Psi^{-1}(QS) = QL^{perm}$, which is not the permutation closure of a context-free language according to Proposition 2. \square

Theorem 5 For \mathcal{K} being any of the calculi LBC, LP, L^*P , or L_1^*P , \mathcal{K} -grammars can generate languages that are not permutation closures of context-free languages.

Proof of Theorem 5 Since LBC-grammars is a subclass of LP-grammars, the statement of the theorem follows from Theorem 4 for $\mathcal{K} = LP$. The second statement of Theorem 3 proves the statement of the theorem for $\mathcal{K} = L^*P$. Finally, note that L^*P -grammars is a subclass of L_1^*P -grammars, so the statement of the theorem holds for them as well. \square

Theorem 6 A commutative language L is generated by an LBC-grammar if and only if its Parikh image $\Psi(L)$ is generated by an LRBVASSAM.

Proof of Theorem 6

Given an LRBVASSAM H generating $\Psi(L)$, one constructs the input-sensitive LRBVASSAM \widetilde{H} generating the same set (using Proposition 9). Then, $LBCG(\widetilde{H})$ is an LBC-grammar such that $L(LBCG(\widetilde{H})) = \Psi^{-1}(L(H)) = L$, according to Theorem 10. This proves the “if” statement of the theorem. Conversely, given an LBC-grammar G generating L , one constructs the LRBVASSAM $LRB(G)$. Then, according to Theorem 12, $L(LRB(G)) = \Psi(L(G)) = \Psi(L)$. This proves the “only if” statement of the theorem. \square

Theorem 7 The class of languages generated by LP-grammars is equal to the class of languages generated by LBC-grammars.

Proof of Theorem 7

Clearly, the class of languages generated by LBC-grammars is contained in the class of languages generated by LP-grammars. To show the converse, take an LP-grammar G . Firstly, convert it into an LRBVASSAM $LRB(G)$; Theorem 12 implies that $L(LRB(G)) = \Psi(L(G))$. Secondly, convert $LRB(G)$ into the LBC-grammar $LBCG(LRB(G))$. Theorem 10 implies that $L(LBCG(LRB(G))) = \Psi^{-1}(L(LRB(G))) = \Psi^{-1}(\Psi(L(G))) = L(G)$. \square

7 Closure Properties

We have proved that the class of languages generated by LP-grammars is wider than permutation closures of context-free languages. This makes the former class quite mysterious since we do not know much about its structure. Let us establish some closure properties of this class using the equivalence of LP-grammars and LRBVASSAMs. Namely, we are going to prove Theorem 8 by showing that sets generated by LRBVASSAMs are closed under intersection, union, sum, and Kleene star. Let us recall the statement of this theorem:

The class of languages generated by LP-grammars is closed under union, intersection, commutative concatenation, and commutative Kleene plus.

Definition 16 The sum of two sets $A, B \subseteq \mathbb{N}^k$ is the set $A + B = \{u + v \mid u \in A, v \in B\}$.

Definition 17 The Kleene plus of the set $A \subseteq \mathbb{N}^k$ is the set $A^\oplus = \{u_1 + \dots + u_l \mid l > 0, u_i \in A \text{ for } i = 1, \dots, l\}$.

Theorem 14 *The class of sets generated by LRBVASSAMs is closed under union, intersection, sum and Kleene plus.*

According to Theorem 6, this theorem is equivalent to Theorem 8.

Firstly, let us prove that sets generated by LRBVASSAMs are closed under union and sum. To do this, let us introduce a special property of LRBVASSAMs:

Definition 18 An LRBVASSAM $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, k, K, C \rangle$ is *totally restricted* if any derivation tree of any fact of the form $s(\iota_K(v))$ in the BVASS $\langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, K \rangle$ has the size less than or equal to $C|v|$.

Compare this with the definition of LRBVASSAM. In that definition, we do not impose any restriction on a BVASS; instead, we just disregard large derivations. In the definition of a totally restricted LRBVASSAM, it is required that *any possible derivation* is not large.

Proposition 15 *For any LP-grammar $G = \langle S, \triangleright \rangle$ the LRBVASSAM $\text{LRB}(G)$ is totally restricted.*

Proof Look at the proof of Theorem 12. It follows from that proof that $S(\iota_K(v))$ has a derivation tree of size d if and only if $a_{i_1}, \dots, a_{i_m} \Rightarrow S$ has a derivation tree of the size d . It also follows from that proof that any derivation tree of $a_{i_1}, \dots, a_{i_m} \Rightarrow S$ has a size that does not exceed $F \cdot m = F|v|$. Therefore, d must not exceed $F|v|$ for any derivation tree of $S(\iota_K(v))$, as desired. \square

Proposition 16 For each LRBVASSAM H there exists a totally-restricted LRBVASSAM generating $L(H)$.

Proof Take an equivalent input-sensitive LRBVASSAM \tilde{H} (using Proposition 9) and consider the LRBVASSAM $\text{LRB}(\text{LBCG}(\tilde{H}))$. It generates $L(H)$ according to Theorems 10 and 12 and it is totally restricted. \square

Let us prove Theorem 14. To do this, let us take two LRBVASSAMs H^1, H^2 . Assume without loss of generality that they are totally restricted. Let $H^i = \langle Q^i, \mathcal{P}_0^i, \mathcal{P}_1^i, \mathcal{P}_2^i, s^i, \kappa, K^i, C^i \rangle$; we modify these LRBVASSAMs as follows.

1. Let us make Q^1 and Q^2 disjoint by defining $Q = \{(q, i) \mid i = 1, 2, q \in Q^i\}$ and replacing the set of states Q^i by $\bar{Q}^i = \{(q, i) \mid q \in Q^i\}$. Let $\bar{s}^i = (s^i, i)$.
2. Let us define K as $\max\{K^1, K^2\}$. Then, given H^i , let us replace
 - each its nullary rule $q(\nu)$ by $(q, i)(\iota_K(\nu))$,
 - each its unary rule $p(x + \delta_2) \leftarrow q(x + \delta_1)$ by

$$(p, i)(x + \iota_K(\delta_2)) \leftarrow (q, i)(x + \iota_K(\delta_1)),$$

- each its binary rule $q(x + y) \leftarrow p(x), r(y)$ by

$$(q, i)(x + y) \leftarrow (p, i)(x), (r, i)(y).$$

The new sets of nullary, unary and binary rules are denoted by $\bar{\mathcal{P}}_0^i, \bar{\mathcal{P}}_1^i, \bar{\mathcal{P}}_2^i$ resp.

Let $\bar{H}^i = \langle \bar{Q}^i, \bar{\mathcal{P}}_0^i, \bar{\mathcal{P}}_1^i, \bar{\mathcal{P}}_2^i, \bar{s}^i, \kappa, K, C^i \rangle$. Clearly, $L(\bar{H}^i) = L(H^i)$; indeed, new additional memory coordinates always equal 0, so they do not affect derivations. It is also clear that the new grammars are also totally restricted.

Construction 5 (LRBVASSAM generating $L(H^1) \cup L(H^2)$)

Let $H^\cup = \langle Q^\cup, \mathcal{P}_0^\cup, \mathcal{P}_1^\cup, \mathcal{P}_2^\cup, s^\cup, \kappa, K, C^\cup \rangle$ where

- $Q^\cup = \bar{Q}^1 \cup \bar{Q}^2 \cup \{s^\cup\}$ where s^\cup is a new state that does not belong to $\bar{Q}^1 \cup \bar{Q}^2$;
- $\mathcal{P}_j^\cup = \bar{\mathcal{P}}_j^1 \cup \bar{\mathcal{P}}_j^2$ for $j = 0, 2$;
- $\mathcal{P}_1^\cup = \bar{\mathcal{P}}_1^1 \cup \bar{\mathcal{P}}_1^2 \cup \{\omega_1, \omega_2\}$ such that $\omega_j = s^\cup(x) \leftarrow \bar{s}^j(x)$;
- $C^\cup = \max\{C^1, C^2\} + 1$.

Proposition 17 Assume that $t(u)$ has a derivation tree in H^\cup for $u \in \mathbb{N}^K$ and $t \in \bar{Q}^j$. Then all the rules in this derivation tree are from $\bar{\mathcal{P}}^j = \bar{\mathcal{P}}_0^j \cup \bar{\mathcal{P}}_1^j \cup \bar{\mathcal{P}}_2^j$.

Proof The proof is by induction on the size of the derivation tree. Without loss of generality, let $j = 1$. It suffices to notice that, if the fact $t(u)$ has the state $t \in \bar{Q}^1$, then it cannot be obtained as the result of any rule from $\bar{\mathcal{P}}^2$ (since there only states from \bar{Q}^2 are involved) or of one of the rules ω_1, ω_2 . The remaining possibility is that $t(u)$ is obtained by means of a

rule from $\bar{\mathcal{P}}^1$. It is either an axiom or is the result of an application of a unary or a binary rule. In the second case, it remains to apply the induction hypothesis to its premises. \square

Lemma 13 *The LRBVASSAM H^\cup generates the set $L(H^1) \cup L(H^2)$.*

Proof Assume that $s^\cup(\iota_K(v))$ has a derivation tree in H^\cup for $v \in \mathbb{N}^\kappa$. The last rule application in this derivation tree must be ω_j for some $j \in \{1, 2\}$; hence $\bar{s}^j(\iota_K(v))$ is derivable in H^\cup . Without loss of generality, let $j = 1$. According to Proposition 17, the derivation tree contains rules only from $\bar{\mathcal{P}}^1$. Therefore $\bar{s}^1(\iota_K(v))$ is derivable in H^1 , hence $v \in L(H^1)$ (since any derivation in H^1 satisfies the linear restriction). Similar reasonings should be made if $j = 2$. This shows that $L(H^\cup) \subseteq L(H^1) \cup L(H^2)$.

The other way round, given a derivation tree of $\bar{s}^j(\iota_K(v))$ in H^j of the size $d \leq C^j|v|$, let us apply the rule ω_j to it and obtain $s^\cup(\iota_K(v))$. The size of its derivation tree equals $d + 1 \leq 1 + C^j|v| \leq C|v|$. Here we use the fact that $|v| \neq 0$ (see Remark 4). Therefore, $L(H^1) \cup L(H^2) \subseteq L(H^\cup)$. \square

Construction 6 (LRBVASSAM generating $L(H^1) + L(H^2)$)

Let $H^+ = \langle Q^+, \mathcal{P}_0^+, \mathcal{P}_1^+, \mathcal{P}_2^+, s^+, \kappa, K, C^+ \rangle$ where

- $Q^+ = \bar{Q}^1 \cup \bar{Q}^2 \cup \{s^+\}$ where s^+ is a new state, which does not belong to $\bar{Q}^1 \cup \bar{Q}^2$;
- $\mathcal{P}_j^+ = \bar{\mathcal{P}}_j^1 \cup \bar{\mathcal{P}}_j^2$ for $j = 0, 1$;
- $\mathcal{P}_2^+ = \bar{\mathcal{P}}_2^1 \cup \bar{\mathcal{P}}_2^2 \cup \{\sigma\}$ such that $\sigma = s^+(x) \leftarrow \bar{s}^1(x), \bar{s}^2(y)$;
- $C^+ = C^\cup = \max\{C^1, C^2\} + 1$.

Proposition 18 *Assume that $t(u)$ has a derivation tree in H^+ for $u \in \mathbb{N}^K$ and $t \in \bar{Q}^j$. Then all the rules in this derivation tree are from $\bar{\mathcal{P}}^j$.*

The proof is the same as that of Proposition 17.

Lemma 14 *The LRBVASSAM H^+ generates the set $L(H^1) + L(H^2)$.*

Proof Assume that $s^+(\iota_K(v))$ has a derivation tree in H^+ for $v \in \mathbb{N}^\kappa$. The last rule application in this derivation tree must be σ :

$$\frac{\bar{s}^1(\iota_K(v_1)) \quad \bar{s}^2(\iota_K(v_2))}{s^+(\iota_K(v))} (\sigma)$$

Thus $\bar{s}^j(\iota_K(v_j))$ is derivable in H^+ for some $v_j \in \mathbb{N}^\kappa$ (here $j = 1, 2$). It holds that $\iota_K(v_1) + \iota_K(v_2) = \iota_K(v)$; equivalently, $v_1 + v_2 = v$. According to Proposition 18, the fact $\bar{s}^j(\iota_K(v_j))$ is derivable in H^j . Consequently, $v_j \in L(H^j)$ since any derivation in H^j satisfies the linear restriction. This shows that $L(H^+) \subseteq L(H^1) + L(H^2)$.

Conversely, given $v_j \in L(H^j)$, consider a derivation tree \mathcal{T}^j of $\bar{s}^j(\iota_K(v_j))$ in H^j ; it must be of the size $d^j \leq C^j|v_j|$. Consider the following derivation tree of $s^+(\iota_K(v_1 + v_2))$:

$$\frac{\mathcal{T}_1 \quad \mathcal{T}_2}{s^+(\iota_K(v_1 + v_2))} (\sigma)$$

Its size equals $d^1 + d^2 + 1 \leq C^1|v_1| + C^2|v_2| + 1 \leq \max\{C^1, C^2\}(|v_1| + |v_2|) + 1 = \max\{C^1, C^2\}|v_1 + v_2| + 1 \leq C^+|v_1 + v_2|$. Here we use the fact that $v_1, v_2 \in \mathbb{N}^\kappa$, which implies that $|v_1| + |v_2| = |v_1 + v_2|$. We also use the fact that $v_1 + v_2 \neq \vec{0}$ (otherwise $v_1 = v_2 = 0$, which is impossible as explained in Remark 4). In what follows, $L(H^+) \subseteq L(H^1) + L(H^2)$. \square

Now, let us show how to generate the commutative Kleene plus $L(H)^\oplus$.

Construction 7 (LRBVASSAM generating $L(H)^\oplus$)

Given the totally-restricted LRBVASSAM $H = \langle Q, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, s, \kappa, K, C \rangle$, let $H^\oplus = \langle Q^\oplus, \mathcal{P}_0^\oplus, \mathcal{P}_1^\oplus, \mathcal{P}_2^\oplus, s^\oplus, \kappa, K, C^\oplus \rangle$ where

- $Q^\oplus = Q \cup \{s^\oplus\}$ where s^\oplus is a new state;
- $\mathcal{P}_0^\oplus = \mathcal{P}_0$;
- $\mathcal{P}_1^\oplus = \mathcal{P}_1 \cup \{\chi\}$ where $\chi = s^\oplus(x) \leftarrow s(x)$;
- $\mathcal{P}_2^\oplus = \mathcal{P}_2 \cup \{\pi\}$ where $\pi = s^\oplus(x + y) \leftarrow s^\oplus(x), s^\oplus(y)$;
- $C^\oplus = C + 2$.

Lemma 15 *The LRBVASSAM H^\oplus generates the set $L(H)^\oplus$.*

The proof is similar to those presented earlier in this section.

Proof Assume that $s^\oplus(\iota_K(v))$ has a derivation tree in H^\oplus for $v \in \mathbb{N}^\kappa$. Let us prove that v belongs to $L(H)^\oplus$ by induction on the size of this derivation. The base case is trivial because there is no axiom with the state s^\oplus .

Consider the last rule application in this derivation.

Case 1. It is π :

$$\frac{s^\oplus(\iota_K(v_1)) \quad s^\oplus(\iota_K(v_2))}{s^\oplus(\iota_K(v))} (\pi)$$

Here v_1, v_2 are some vectors. By induction hypothesis, $v_1, v_2 \in L(H)^\oplus$, and thus $v = v_1 + v_2 \in L(H)^\oplus$ (since the Kleene plus of a set is closed under sums).

Case 2. It is χ :

$$\frac{s(\iota_K(v))}{s^\oplus(\iota_K(v))} (\pi)$$

Derivability of $s(\iota_K(v))$ in H^\oplus is equivalent to derivability of the same fact in H (the proof is the same as for Proposition 17). Thus $v \in L(H) \subseteq L(H)^\oplus$.

To prove the converse, assume that $v = v_1 + \dots + v_l$ where $v_i \in L(H)$; we want to prove that $v \in L(H)^\oplus$. We know that $s(\iota_K(v_i))$ has a derivation tree in H of the size not greater than $C|v_i|$ for $i = 1, \dots, l$; then, $s^\oplus(\iota_K(v_i))$ can be obtained from $s(\iota_K(v_i))$ using π , and the size of its derivation is thus not greater than $C|v_i| + 1$. After that, we apply the rule χ ($l - 1$) times to obtain $s^\oplus(\iota_K(v_1) + \dots + \iota_K(v_l)) = s^\oplus(\iota_K(v))$. The total number of rule applications does not exceed $C|v_1| + 1 + \dots + C|v_l| + 1 + l - 1 = C|v| + 2l - 1$. It remains to note that, since $|v_i| \geq 1$, it holds that $|v| \geq l$. Therefore, the size of the derivation of $s^\oplus(\iota_K(v))$ does not exceed $(C + 2)|v|$, as desired. \square

Speaking of closure under intersection, let us prove this property using LP-grammars instead of LRBVASSAMs (although an argument similar to the above ones can be provided to prove that the class of sets generated by LRBVASSAMs is closed

under intersection). The proof of this fact is inspired by Kanazawa's one from [23] where it is proved that languages generated by grammars over the multiplicative-additive Lambek calculus are closed under intersection. The main idea is that, in the commutative Lambek calculus, we can use multiplicative conjunction instead of the additive one. Let us present the proof.

Construction 8 Let $G^i = \langle S^i, \triangleright^i \rangle$ ($i = 1, 2$) be two LP-grammars. Assume without loss of generality that $\text{SFm}(G^1) \cap \text{SFm}(G^2) = \emptyset$, or, equivalently, that primitive subformulas of formulas from G^1 and G^2 are pairwise disjoint. Having this in mind, let us define $G^\cap = \langle S, \triangleright \rangle$ as follows:

- $S^\cap := S^1 \bullet S^2$;
- $a \triangleright T$ if and only if T is of the form $T_1 \bullet T_2$ where $a \triangleright^i T_i$ ($i = 1, 2$).

Proposition 19

1. Let $\text{LP} \vdash A_1, \dots, A_n \Rightarrow B$ where A_i are from $\text{SFm}(G^\cap)$ and where B is from $\text{SFm}(G^k)$ (for some $k \in \{1, 2\}$). Then all A_i are also from $\text{SFm}(G^k)$.
2. Let $\text{LP} \vdash A_1, \dots, A_n, B_1, \dots, B_m \Rightarrow A \bullet B$ where A_i and A are from $\text{SFm}(G^1)$, and B_i, B are from $\text{SFm}(G^2)$. Then $\text{LP} \vdash A_1, \dots, A_n \Rightarrow A$ and $\text{LP} \vdash B_1, \dots, B_m \Rightarrow B$.

Both statements are proved by straightforward induction on the size of a derivation.

Lemma 16 The LP-grammar G^\cap generates the language $L(G^1) \cap L(G^2)$.

Proof The word $a_1 \dots a_n$ belongs to $L(G^\cap)$ if and only if there exist formulas T_j^i for $i = 1, 2$, $j = 1, \dots, n$ such that $a_j \triangleright T_j^1 \bullet T_j^2$ and

$$\text{LP} \vdash T_1^1 \bullet T_1^2, \dots, T_n^1 \bullet T_n^2 \Rightarrow S^1 \bullet S^2.$$

The latter is equivalent to the fact that $\text{LP} \vdash T_1^1, \dots, T_n^1, T_1^2, \dots, T_n^2 \Rightarrow S^1 \bullet S^2$ (use the rules (\bullet_L) and (\bullet_L^{-1}) to justify this). Using Proposition 19 we conclude that this is equivalent to the fact that $\text{LP} \vdash T_1^1, \dots, T_n^1 \Rightarrow S^1$ and $\text{LP} \vdash T_1^2, \dots, T_n^2 \Rightarrow S^2$. Summarizing, $a_1 \dots a_n$ belongs to $L(G^\cap)$ if and only if there exist formulas T_1^1, \dots, T_n^1 and T_1^2, \dots, T_n^2 such that $a_i \triangleright_1 T_i^1$, $a_i \triangleright_2 T_i^2$ and $\text{LP} \vdash T_1^1, \dots, T_n^1 \Rightarrow S^1$, $\text{LP} \vdash T_1^2, \dots, T_n^2 \Rightarrow S^2$. This means exactly that $a_1 \dots a_n \in L(G^1)$ and $a_1 \dots a_n \in L(G^2)$. This completes the proof. \square

Interestingly, both the class of languages generated by LP-grammars and the class of permutation closures of context-free languages turn out to be closed under intersection. Indeed, each permutation closure of a context-free language is the inverse Parikh image of some semilinear set, and semilinear sets are closed under intersection [24]. If this was not the case, we might have a simpler way of proving that LP-grammars are not equivalent to permutation closures of context-free languages.

8 Conclusion

We have shown that LP-grammars are not context free in the sense that they generate more than permutation closures of context-free languages. This result contrasts with that for Lambek grammars, which are context free [3]. Our proof relies on establishing the equivalence of LP-grammars and LRBVASSAMs, which represent yet another extension of vector addition systems.

Several open questions remain:

1. Is the set of languages generated by LP-grammars closed under complement? Note that semilinear sets are. If the answer to this question is negative, then this would give us another proof of the fact that the class of languages generated by LP-grammars is wider than permutation closures of context-free languages.
2. Can one generate a language like $\{a^{f(n)} \mid n > 0\}$ where $f(n)$ is some nonlinear function (e.g., $f(n) = n^2$) by an LP-grammar? We conjecture that the answer is negative. In our opinion, this problem as well as the previous one can be approached by studying LRBVASSAMs and establishing their properties. Namely, we expect that the pumping lemma or some its version can be proved for LRBVASSAMs.
3. The linear restriction can be imposed on BVASS as well resulting in LRBVASS. Then one might ask whether LRBVASS generate the same class of sets as LRBVASSAMs, i.e. whether additional memory is essential in LRBVASSAMs⁶. Similarly, it would be interesting to answer the question if BVASS are equivalent to BVASSAMs.
4. In general, we are not aware that there is no other formalism existing in the literature, which would appear to be equivalent to LRBVASSAMs. Finding more connections with other formalisms would be interesting.

We would like to emphasize the importance of the *linear restriction* in the definition of LRBVASSAMs. It proved to be extremely useful for relating LP-grammars and BVASS. Moreover, the idea of considering such a restriction can be successfully exploited for other kinds of grammars as well, e.g., for hypergraph Lambek grammars. In [10], we prove that any DPO hypergraph grammar⁷ with the linear restriction similar to the one used in the present paper can be transformed into an equivalent hypergraph Lambek grammar. The main construction in [10] is very close to Construction 1. We assume that using the linear restriction for other kinds of formal grammars could also be fruitful for investigating expressive power of categorial grammars.

Acknowledgments. I thank Stepan L. Kuznetsov for bringing my attention to this problem and for suggesting valuable ideas to explore. I also thank the anonymous reviewer for his careful attention and remarks that helped me to improve my work.

⁶As an exercise, the reader might check that LRBVASSAMs generate the same class of languages as LRBVASSAMs, which are LRBVASSAMs with only one state.

⁷DPO grammars extend unrestricted Chomsky grammars to hypergraphs.

Financial Support

The study was supported by the Theoretical Physics and Mathematics Advancement Foundation “BASIS”; by RFBR, project number 20-01-00670; and by the Interdisciplinary Scientific and Educational School of Moscow University “Brain, Cognitive Systems, Artificial Intelligence”.

References

- [1] Joachim Lambek: The mathematics of sentence structure. *Amer. Math. Mon.* **65**(3), 154–170 (1958)
- [2] Bar-Hillel, Y., Gaifman, H., Shamir, E.: On categorial and phrase structure grammars. *Bull. Res. Counc. Israel* **9**, 1–6 (1960)
- [3] Mati Pentus: Lambek grammars are context free. In: *Proceedings of the Eighth Annual Symposium on Logic in Computer Science*, pp. 429–433. IEEE Computer Society, Montreal, Canada (1993). <https://doi.org/10.1109/LICS.1993.287565>
- [4] Johan van Benthem: The Semantics of Variety in Categorial Grammar. *Linguistic and Literary Studies in Eastern Europe*, vol. 25, pp. 37–55. John Benjamins, Amsterdam (1983)
- [5] Wojciech Buszkowski: A note on the Lambek-van Benthem calculus. *Bulletin of the Section of Logic* **13**(1), 31–35 (1984)
- [6] Johan van Benthem: Language in action. *J. Philos. Log.* **20**(3), 225–263 (1991). <https://doi.org/10.1007/BF00250539>
- [7] Richard Moot and Christian Retoré: *The Logic of Categorial Grammars - A Deductive Account of Natural Language Syntax and Semantics*. Lecture Notes in Computer Science, vol. 6850. Springer, Berlin, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-31555-8>
- [8] Oriol Valentín: *Theory of discontinuous Lambek calculus*. PhD thesis, Universitat Autònoma de Barcelona. Departament de Filologia Catalana (2012)
- [9] Stepan L. Kuznetsov: Personal communication (2021)
- [10] Tikhon Pshenitsyn: From double pushout grammars to hypergraph lambek grammars with and without exponential modality. In: Grabmayer, C. (ed.) *Proceedings TERMGRAPH 2022*, Technion, Haifa, Israel. EPTCS, vol. 377, pp. 9–26 (2023). <https://doi.org/10.4204/EPTCS.377.1>
- [11] Wayne D. Blizard: Multiset theory. *Notre Dame Journal of Formal Logic* **30**(1), 36–66 (1988). <https://doi.org/10.1305/ndjfl/1093634995>

- [12] Parikh, R.J.: Language generating devices. Quarterly Progress Report **60**, 199–212 (1961)
- [13] J. Goldstine: A simplified proof of Parikh’s theorem. Discrete Mathematics **19**(3), 235–239 (1977). [https://doi.org/10.1016/0012-365X\(77\)90103-0](https://doi.org/10.1016/0012-365X(77)90103-0)
- [14] Martin Hyland, Valeria de Paiva: Full intuitionistic linear logic. Ann. Pure Appl. Logic **64**(3), 273–291 (1993). [https://doi.org/10.1016/0168-0072\(93\)90146-5](https://doi.org/10.1016/0168-0072(93)90146-5)
- [15] Seymour Ginsburg, Edwin H. Spanier: Bounded algol-like languages. Transactions of the American Mathematical Society **113**(2), 333–368 (1964)
- [16] Kumar Neeraj Verma and Jean Goubault-Larrecq: Karp-Miller trees for a branching extension of VASS. Discret. Math. Theor. Comput. Sci. **7**(1), 217–230 (2005)
- [17] Richard M. Karp, Raymond E. Miller: Parallel program schemata. Journal of Computer and System Sciences **3**(2), 147–195 (1969). [https://doi.org/10.1016/S0022-0000\(69\)80011-5](https://doi.org/10.1016/S0022-0000(69)80011-5)
- [18] Max I. Kanovich: Petri nets, Horn programs, linear logic and vector games. Ann. Pure Appl. Log. **75**(1-2), 107–135 (1995). [https://doi.org/10.1016/0168-0072\(94\)00060-G](https://doi.org/10.1016/0168-0072(94)00060-G)
- [19] de Groote, P. and Guillaume, B. and Salvati, S.: Vector addition tree automata. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004., pp. 64–73 (2004). <https://doi.org/10.1109/LICS.2004.1319601>
- [20] Owen Rambow: Multiset-valued linear index grammars: Imposing dominance constraints on derivations. In: ACL, pp. 263–270. Morgan Kaufmann Publishers / ACL, Las Cruces, New Mexico (1994). <https://doi.org/10.3115/981732.981768>
- [21] Wojciech Buszkowski: In: Loukanova, R. (ed.) Lambek Calculus with Classical Logic, pp. 1–36. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-63787-3_1
- [22] Max I. Kanovich, Stepan L. Kuznetsov, Andre Scedrov: The complexity of multiplicative-additive lambek calculus: 25 years later. In: Iemhoff, R., Moortgat, M., de Queiroz, R.J.G.B. (eds.) WoLLIC 2019, Utrecht, The Netherlands, Proceedings. Lecture Notes in Computer Science, vol. 11541, pp. 356–372. Springer, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59533-6_22
- [23] Makoto Kanazawa: The Lambek calculus enriched with additional connectives. J. Log. Lang. Inf. **1**(2), 141–171 (1992)
- [24] Seymour Ginsburg, Edwin H. Spanier: Semigroups, Presburger formulas, and languages. Pac. Jour. of Math. **16**(2), 285–296 (1966). <https://doi.org/pjm/1102994974>