The Complexity of Multiplicative-Additive Lambek Calculus: 25 Years Later

Max Kanovich[1,4], Stepan Kuznetsov[2,4](✉), and Andre Scedrov[3,4]

[1] University College London, London, UK
m.kanovich@ucl.ac.uk
[2] Steklov Mathematical Institute of the RAS, Moscow, Russia
sk@mi-ras.ru
[3] University of Pennsylvania, Philadelphia, USA
scedrov@math.upenn.edu
[4] National Research University Higher School of Economics, Moscow, Russia

Abstract. The Lambek calculus was introduced as a mathematical description of natural languages. The original Lambek calculus is NP-complete (Pentus), while its product-free fragment with only one implication is polynomially decidable (Savateev). We consider Lambek calculus with the additional connectives: conjunction and disjunction. It is known that this system is PSPACE-complete (Kanovich, Kanazawa). We prove, in contrast with the polynomial-time result for the product-free Lambek calculus with one implication, that the derivability problem is still PSPACE-complete even for a very small fragment $(\backslash, \wedge)$, including one implication and conjunction only. PSPACE-completeness is also provided for the $(\backslash, \vee)$ fragment, which includes only one implication and disjunction. Categorial grammars based on the original Lambek calculus generate exactly the class of context-free languages (Gaifman, Pentus). The class of languages generated by Lambek grammars extended with conjunction is known to be closed under intersection (Kanazawa), and therefore includes all finite intersections of context-free languages and, moreover, images of such intersections under alphabetic homomorphisms. We show that the same closure under intersection holds for Lambek grammars extended with disjunction, even for our small $(\backslash, \vee)$ fragment.

Keywords: Lambek calculus · Lambek grammars · Completeness · PSPACE-completeness

1 Introduction

Lambek calculus has been invented to analyze natural and artificial languages by means of categorial grammars [4,17,19,20]. Though the original Lambek calculus can describe only context-free languages [23], it has been proven to be NP-complete [24], even if we confine ourselves to the product-free Lambek calculus

R. Iemhoff et al. (Eds.): WoLLIC 2019, LNCS 11541, pp. 356–372, 2019.
https://doi.org/10.1007/978-3-662-59533-6_22

equipped only with the left implication and the right implication [27]. On the contrary, the product-free Lambek calculus, with only one implication, is known to be decidable in polynomial time [26], see also [15]. It is known [2,3] that already the fragment with only one implication is sufficient to generate all context-free languages.

This paper is focused on the complexity issues for Lambek calculus extended with two additional connectives: additive conjunction and disjunction. This calculus is presented on Table 1 in the form of a sequent calculus. Notice that antecedents of sequents are linearly ordered sequences of formulae, not sets or multisets.

Table 1. The Inference rules of Lambek calculus with conjunction and disjunction

I	$\frac{}{A \vdash A}$		
L$\backslash$	$\frac{\Phi \vdash A \qquad \Sigma_1, B, \Sigma_2 \vdash C}{\Sigma_1, \Phi, (A \backslash B), \Sigma_2 \vdash C}$	**R**$\backslash$	$\frac{A, \Sigma \vdash B}{\Sigma \vdash (A \backslash B)}$ (Σ is not empty)
L$/$	$\frac{\Phi \vdash A \qquad \Sigma_1, B, \Sigma_2 \vdash C}{\Sigma_1, (B / A), \Phi, \Sigma_2 \vdash C}$	**R**$/$	$\frac{\Sigma, A \vdash B}{\Sigma \vdash (B / A)}$ (Σ is not empty)
L$\cdot$	$\frac{\Sigma_1, A, B, \Sigma_2 \vdash C}{\Sigma_1, (A \cdot B), \Sigma_2 \vdash C}$	**R**$\cdot$	$\frac{\Sigma_1 \vdash A \qquad \Sigma_2 \vdash B}{\Sigma_1, \Sigma_2 \vdash (A \cdot B)}$
L$\vee$	$\frac{\Sigma_1, A, \Sigma_2 \vdash C \qquad \Sigma_1, B, \Sigma_2 \vdash C}{\Sigma_1, (A \vee B), \Sigma_2 \vdash C}$	**R**$\vee$	$\frac{\Sigma \vdash A}{\Sigma \vdash (A \vee B)} \qquad \frac{\Sigma \vdash B}{\Sigma \vdash (A \vee B)}$
L$\wedge$	$\frac{\Sigma_1, A, \Sigma_2 \vdash C}{\Sigma_1, (A \wedge B), \Sigma_2 \vdash C}$ $\frac{\Sigma_1, B, \Sigma_2 \vdash C}{\Sigma_1, (A \wedge B), \Sigma_2 \vdash C}$	**R**$\wedge$	$\frac{\Sigma \vdash A \qquad \Sigma \vdash B}{\Sigma \vdash (A \wedge B)}$

As shown above on the example of the Lambek calculus without additive connectives, there are two different ways of measuring complexity for extensions of the Lambek calculus. The first one is the standard notion of algorithmic complexity of the derivability problem for the calculus in question. For the Lambek calculus with additive connectives, 25 years ago, Kanovich [10] and Kanazawa [9] show that its derivability problem is PSPACE-complete. Here we strengthen this result and prove PSPACE-hardness for the smallest possible fragments, with only two connectives: $\mathcal{L}(\backslash, \wedge)$, with only one implication and additive conjunction, and $\mathcal{L}(\backslash, \vee)$, with one implication and disjunction. The first result is presented in Sect. 2. The second result is similar, so we give only a sketch of the proof, in Appendix A. The upper PSPACE bound is known for the whole Lambek calculus with additive connectives [9,10], [13, Sect. 8] and therefore inherited by its fragments, $\mathcal{L}(\backslash, \wedge)$ and $\mathcal{L}(\backslash, \vee)$.

The other complexity measure is the expressive power of categorial grammars based on a given calculus. A categorial grammar $\mathcal{G}$ is a triple $\langle \Sigma, \triangleright, H \rangle$, where Σ is a finite alphabet, $\triangleright$ is a finite binary correspondence between letters of Σ and Lambek formulae (these formulae could also include additive connectives), and H is a formula. A non-empty word $w = a_1 \ldots a_n$ over Σ is accepted by $\mathcal{G}$, if there exist formulae $A_1, \ldots, A_n$ such that $a_i \triangleright A_i$ $(i = 1, \ldots, n)$ and $A_1, \ldots, A_n \vdash H$ is a derivable sequent. The language generated by $\mathcal{G}$ is the set of all accepted words.

For Lambek grammars extended with conjunction, Kanazawa [8] proves that, in addition to context-free languages, they can generate finite intersections of such languages and images of such intersections under alphabetic homomorphisms (*i.e.,* homomorphisms which map letters to letters). In Sect. 3 we prove the dual result, that Lambek grammars enriched with disjunction have the same property. Namely, we show that $\mathcal{L}(\backslash, \vee)$, the product-free fragment with only one implication and disjunction, is already sufficient to generate finite intersections of such languages and images of such intersections under alphabetic homomorphisms.

2 PSPACE-Hardness of the Fragment $\mathcal{L}(\backslash, \wedge)$

Within our fragment $\mathcal{L}(\backslash, \wedge)$, we intend to encode quantified Boolean statements of the form:

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \dots \exists x_{2n-1} \forall x_{2n} \, (C_1 \vee C_2 \vee \cdots \vee C_m) \tag{1}$$

Here $(C_1 \vee C_2 \vee \cdots \vee C_m)$ is a DNF over the Boolean variables x_1, x_2, $\dots x_{2n}$.

Definition 1. *We express validity of (1) in terms of* **the winning strategy** *given by a binary tree of height* $2n{+}1$*, the nodes of which are labelled as follows.*

The root is labelled by "$\exists x_1$*" and has only one outgoing edge the end of which is labelled by "*$\forall x_2$*". In its turn, this node has two outgoing edges the ends of which are labelled by the same "*$\exists x_3$*".*

By induction, for $1 \leq k \leq n$*, each of the nodes on the level* $2k{-}1$ *is labelled by "*$\exists x_{2k-1}$*", and each of the nodes on the level* $2k$ *is labelled by "*$\forall x_{2k}$*".*

At the node "$\exists x_{2k-1}$*",* **the choice move of the proponent** *is to label the unique outgoing edge either by* t_{2k-1}*,* **meaning** x_{2k-1} **be true***, or by* f_{2k-1}*,* **meaning** x_{2k-1} **be false***. Being at the next node, "*$\forall x_{2k}$*",* **the opponent** *responds by labeling two outgoing edges by* t_{2k} *and* f_{2k}*, resp.*

Lastly, on the final level $2n{+}1$*, each terminal node* v *is labelled by some* C_ℓ *so that, collecting the sequence of* α_1*,* α_2*,* $\dots$*,* α_{2n-1}*,* α_{2n} *that label the respective edges along the branch leading from the root "*$\exists x_1$*" to this leaf* v*, we get:*

$$C_\ell(\alpha_1, \alpha_2, \dots, \alpha_{2n-1}, \alpha_{2n}) = \top \tag{2}$$

We illustrate the challenges we have to answer to with Example 1.

Example 1. We consider the following statement (which is invalid):

$$\exists x_1 \forall x_2 \, (C_1 \vee C_2) = \exists x_1 \forall x_2 \, ((x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)) \tag{3}$$

To provide (2), we express C_1 and C_2 as the formulas E_1 and E_2, resp.

$$E_1 = (f_2 \backslash (t_1 \backslash \top)) \;\equiv\; ((t_1 \cdot f_2) \backslash \top) \tag{4}$$

$$E_2 = (t_2 \backslash (f_1 \backslash \top)) \;\equiv\; ((f_1 \cdot t_2) \backslash \top) \tag{5}$$

Following [9,10], we intend to express the "choice move" $\exists x_1$ as $(t_1 \wedge f_1)$, and the "branching move" $\forall x_2$ as $(t_2 \vee f_2)$, resulting in the following encoding sequent:

$$(t_1 \wedge f_1),\ (t_2 \vee f_2),\ (E_1 \wedge E_2) \vdash \top \tag{6}$$

Taking $(t_1 \wedge f_1),\ (t_2 \vee f_2)$ as a sequence, we assume that these formulas should be executed *in the natural order*. Starting with $(t_1 \wedge f_1)$, we have to prove either

$$t_1,\ (t_2 \vee f_2),\ (E_1 \wedge E_2) \vdash \top \tag{7}$$

or

$$f_1,\ (t_2 \vee f_2),\ (E_1 \wedge E_2) \vdash \top \tag{8}$$

Since both sequents are not derivable, we might have concluded that (6) was not derivable and, hence, it was in a proper correlation with the invalid (3).
However, if we first apply $(t_2 \vee f_2)$, the related sequents turn out to be derivable

$$(t_1 \wedge f_1),\ t_2,\ (E_1 \wedge E_2) \vdash \top \tag{9}$$
$$(t_1 \wedge f_1),\ f_2,\ (E_1 \wedge E_2) \vdash \top \tag{10}$$

which shows that in fact (6) is derivable and, hence, fails to express the invalid (3).

The intuitive remedy proposed by [9,10,18] is to force the correct order of actions by means of "leading" q_i. E.g., here we can express the "choice move" $\exists x_1$ and the "branching move" $\forall x_2$ as the following formulas adjusted

$$(q_0 \backslash ((t_1 \cdot q_1) \wedge (f_1 \cdot q_1))) \tag{11}$$
$$(q_1 \backslash ((t_2 \cdot q_2) \vee (f_2 \cdot q_2))) \tag{12}$$

resulting in the correct non-provable encoding sequent, something like that

$$q_0,\ (q_0 \backslash ((t_1 \cdot q_1) \wedge (f_1 \cdot q_1))),\ (q_1 \backslash ((t_2 \cdot q_2) \vee (f_2 \cdot q_2))),\ (q_2 \backslash (E_1 \wedge E_2)) \vdash \top \tag{13}$$

The challenge of implementing this approach within $\mathcal{L}(\backslash, \wedge)$ consists of two parts:

(a) get rid off the disjunctions, in the absence of the full duality of $\wedge$ and $\vee$;
(b) get rid off the positive products of the form $(A \backslash (B_1 \cdot B_2))$

2.1 The Relative Negation and Double Negation (Non-commutative)

Definition 2. *In our encodings we will use the following abbreviation. We fix an atomic proposition b, and define 'relative negation' A^b by: $A^b = (A \backslash b)$.*

Our relative negation can be seen as a non-commutative analogue of the linear logic negation [5], which is defined by $A^\perp = A \multimap \perp$.

As for the relative "double negation", the novelty of our approach is that we are in favour of the "asymmetric" $A^{bb} = ((A \backslash b) \backslash b)$, because of its nice properties proven in Lemma 1.

We use also the following notation for the towers of double negations:

$$A^{[0]} = A, \quad A^{[k+1]} = (A^{[k]} \backslash b) \tag{14}$$

Remark 1. The "double negation" in the symmetrical form: ${}^{b}A^{b} = (b/(A\backslash b))$, has received recognition as being appropriate and logical within a non-commutative linear logic framework (see [1]).

E.g., the natural $A \vdash {}^{b}A^{b}$ is valid, in contrast to our A^{bb}, see Lemma 1(e).

However, the crucial Lemma 1(a) is destroyed with ${}^{b}A^{b}$, which is the reason for our "non-logical" choice of A^{bb}. ■

For a sequence $\Gamma = A_1, A_2, .., A_s$, by Γ^{bb} we denote the sequence $A_1^{bb}, A_2^{bb}, .., A_s^{bb}$.

Lemma 1. (a) *The following rules are derivable in Lambek calculus, $s \geq 1$:*

$$\frac{A_1,\ A_2,\ \ldots,\ A_s\ \vdash C}{A_1^{bb},\ A_2^{bb},\ \ldots,\ A_s^{bb}\ \vdash C^{bb}} \tag{15}$$

(b) *Though $\vee$ and $\wedge$ are not fully dual: $(A \wedge B)^{b} \not\vdash (A^{b} \vee B^{b})$, the following equivalence fits our purposes:*

$$A^{b} \wedge B^{b} \vdash (A \vee B)^{b} \quad \textit{and} \quad (A \vee B)^{b} \vdash A^{b} \wedge B^{b} \tag{16}$$

(c) *To simulate branching, we will use the derivable rule:*

$$\frac{\Gamma,\, A,\, \Delta \vdash C \qquad \Gamma,\, B,\, \Delta \vdash C}{\Gamma^{bb},\ (A^{b} \wedge B^{b})^{b},\ \Delta^{bb}\ \vdash C^{bb}} \tag{17}$$

(d) *With $G_i = (q_{i-1}\backslash B)$, the crucial rule of "leading" q_{i-1} is given by:*

$$\frac{\dfrac{\Gamma,\, \gamma,\, B,\, \Delta \vdash C}{\Gamma,\ \gamma,\ q_{i-1},\ G_i,\ \Delta \vdash C}}{\Gamma^{bb},\ (\gamma \cdot q_{i-1})^{bb},\ G_i^{bb},\ \Delta^{bb}\ \vdash C^{bb}} \tag{18}$$

(e) Essential complications *are caused by the fact that $A \vdash A^{bb}$ is not valid.*

Lemma 2. *If c does not occur in $A_1, \ldots, A_n, B$, then the sequent $A_1^{cc}, \ldots, A_n^{cc} \vdash B^{cc}$ is equiderivable with $A_1, \ldots, A_n \vdash B$.*

Proof. The right-to-left direction is due to Lemma 1(i). For the left-to-right direction, we use the reversibility of $\mathbf{R}\backslash$:

$$B \backslash c, (A_1 \backslash c) \backslash c, \ldots, (A_n \backslash c) \backslash c \vdash c.$$

By induction on k, let us show derivability of

$$A_{n-k}, \ldots, A_n, B \backslash c, (A_1 \backslash c) \backslash c, \ldots, (A_{n-k-1} \backslash c) \backslash c \vdash c.$$

Induction base ($k = 0$) is given above. For the induction step, apply Lemma 11 below, which yields derivability of

$$A_{n-k}, \ldots, A_n, B \backslash c, (A_1 \backslash c) \backslash c, \ldots, (A_{n-k-2} \backslash c) \backslash c \vdash A_{n-k-1} \backslash c$$

and reverse the $\mathbf{R}\backslash$ rule. Finally, we get $A_1, \ldots, A_n, B \backslash c \vdash c$, and one more application of Lemma 11 yields the necessary $A_1, \ldots, A_n \vdash B$. ■

2.2 Complexity of the fragment $\mathcal{L}(\backslash, \wedge)$

Remark 2. Because of Lemma 1, for the sake of readability, here we will conceive of the formula $((A \cdot B)\backslash C)$ as abbreviation for $(B\backslash (A\backslash C))$. In particular, $(A \cdot B)^b$ is abbreviation for $(B\backslash (A\backslash b))$. The formula $(A \vee B)^b$ is conceived of as abbreviation for $((A\backslash b) \wedge (B\backslash b))$.

Theorem 1. *The fragment $\mathcal{L}(\backslash, \wedge)$ is PSPACE-hard.*

Proof. The direction from winning trees to derivable sequents is provided by Corollary 1.

By running from the leaves of the winning tree, labelled by some C_ℓ, to its root "$\exists x_1$", we have to address the following issues:

(a) With one and the same sequent of *polynomial size*, deal with the exponential number of branches and their sequences of α_1, α_2, ..., α_{2n-1}, α_{2n} that label the respective edges along the branch leading from the root to some leaf v.
(b) In particular, verify "polynomially" the corresponding equalities (2).

■

Remark 3. To guarantee the proper order of the inference rules applied, we use the "leading" q_0, q_1, ..., q_{2n-1}, q_{2n}, and $c_{\ell,2n}$, $c_{\ell,2n-1}$,..., $c_{\ell,2}$, $c_{\ell,1}$, $c_{\ell,0}$. The latter $c_{\ell,i}$ is used to keep one and the same C_ℓ in the process of verifying (2).

2.3 Verifying the Equality (2)

We start with (b), assuming that the sequence α_1, α_2, ..., α_{2n-1}, α_{2n} is *fixed.*

Definition 3. *Let F_ℓ denote: $(q_{2n}\backslash c_{\ell,2n})$, and H_ℓ denote: $(c_{\ell,0}\backslash (e_0\backslash e_0))$.*
For $1 \le i \le 2n$, let $E_{\ell,i}$ denote the formula: $(c_{\ell,i}\backslash (t_i\backslash c_{\ell,i-1}))$, if the conjunct C_ℓ contains the variable x_i; and $E_{\ell,i}$ denote the formula: $(c_{\ell,i}\backslash (f_i\backslash c_{\ell,i-1}))$, if the conjunct C_ℓ contains the variable $\neg x_i$; and $E_{\ell,i}$ denote the formula: $((c_{\ell,i}\backslash (t_i\backslash c_{\ell,i-1})) \wedge (c_{\ell,i}\backslash (f_i\backslash c_{\ell,i-1})))$, if C_ℓ contains neither x_i, nor $\neg x_i$.
We introduce their "closed" versions:

$$\widetilde{F} = \bigwedge_{\ell=1}^{m} F_\ell, \quad \widetilde{H} = \bigwedge_{\ell=1}^{m} H_\ell, \quad \widetilde{E_i} = \bigwedge_{\ell=1}^{m} E_{\ell,i} \qquad (19)$$

Lemma 3. *In case (2) holds, a sequent of the specific form is derivable:*

$$e_0^{bb}, \alpha_1^{bb}, \alpha_2^{bb}, \ldots, \alpha_{2n-2}^{bb}, \alpha_{2n-1}^{bb}, (\alpha_{2n} \cdot q_{2n})^{bb},\ \Delta_n^{bb}\ \vdash\ e_0^{bb} \qquad (20)$$

where Δ_n is a sequence of formulas: $\Delta_n = \widetilde{F}, \widetilde{E}_{2n}, \widetilde{E}_{2n-1}, \ldots, \widetilde{E_2}, \widetilde{E_1}, \widetilde{H}$.

NB: Notice that Δ_n *does not depend* on particular α_1, α_2, ..., α_{2n-1}, α_{2n}.

Proof. Since $\alpha_{i-1}, \alpha_i, c_{\ell,i}, E_{\ell,i,\alpha_i} \vdash (\alpha_{i-1} \cdot c_{\ell,i-1})$, by a simple inverse induction on i, we can "consume" all of the $\alpha_i, c_{\ell,i}$ with getting the sequents derivable:

$$e_0, \alpha_1, \alpha_2, \ldots, \alpha_{2n}, q_{2n}, F_\ell, E_{\ell,2n}, E_{\ell,2n-1}, \ldots E_{\ell,2}, E_{\ell,1}, H_\ell \vdash e_0$$

and (see the rule $\mathbf{L}\wedge$)

$$e_0, \alpha_1, \alpha_2, \ldots, \alpha_{2n-2}, \alpha_{2n-1}, \alpha_{2n}, q_{2n}, \varDelta_n \vdash e_0$$

resulting in (20) with the help of Lemma 1. ■

2.4 Simulating the Opponent's and Proponent's Moves

Now we are ready to simulate the moves in the play.

Lemma 4. *For any sequence $\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}, \alpha_{2n}$, labeling the branch that leads from the root to "$\forall x_{2n}$", the opponent move at "$\forall x_{2n}$" is to label two outgoing edges by t_{2n} and f_{2n} resp. We simulate the move by the derivable sequent:*

$$e_0^{[6]}, \alpha_1^{[6]}, \alpha_2^{[6]}, \ldots, \alpha_{2n-2}^{[6]}, (\alpha_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]}, G_{2n}^{[2]}, \varDelta_n^{[6]} \vdash e_0^{[6]} \tag{21}$$

where

$$G_{2n} = (q_{2n-1} \backslash \left((t_{2n} \cdot q_{2n})^{[3]} \wedge (f_{2n} \cdot q_{2n})^{[3]}\right)^{[1]}) \tag{22}$$

Proof. Having got two sequences at hand

$$\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}, t_{2n},$$

and

$$\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}, f_{2n},$$

by Lemma 3 we have

$$e_0^{[2]}, \alpha_1^{[2]}, \alpha_2^{[2]}, \ldots, \alpha_{2n-2}^{[2]}, \alpha_{2n-1}^{[2]}, (t_{2n} \cdot q_{2n})^{[2]}, \varDelta_n^{[2]} \vdash e_0^{[2]} \tag{23}$$

and

$$e_0^{[2]}, \alpha_1^{[2]}, \alpha_2^{[2]}, \ldots, \alpha_{2n-2}^{[2]}, \alpha_{2n-1}^{[2]}, (f_{2n} \cdot q_{2n})^{[2]}, \varDelta_n^{[2]} \vdash e_0^{[2]} \tag{24}$$

by Lemma 1(c) we produce

$$e_0^{[4]}, \alpha_1^{[4]}, \alpha_2^{[4]}, \ldots, \alpha_{2n-2}^{[4]}, \alpha_{2n-1}^{[4]}, \left((t_{2n} \cdot q_{2n})^{[3]} \wedge (f_{2n} \cdot q_{2n})^{[3]}\right)^{[1]}, \varDelta_n^{[4]} \vdash e_0^{[4]}$$

and conclude, Lemma 1(d), with the sequent (21) where G_{2n} is given by (22). ■

Lemma 5. *For the shorter sequence $\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}$, labeling the one-edge shorter branch that leads from the root to "$\exists x_{2n-1}$", the proponent move at "$\exists x_{2n-1}$" is to label the outgoing edge by α_{2n-1}.*
We simulate the move by the derivable sequent:

$$e_0^{[8]}, \alpha_1^{[8]}, \alpha_2^{[8]}, \ldots, \alpha_{2n-3}^{[8]}, (\alpha_{2n-2}^{[6]} \cdot q_{2n-2})^{[2]}, G_{2n-1}^{[2]}, G_{2n}^{[4]}, \varDelta_n^{[8]} \vdash e_0^{[8]} \tag{25}$$

where

$$G_{2n-1} = (q_{2n-2} \backslash \left((t_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]} \wedge (f_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]}\right)) \tag{26}$$

Proof. Lemma 4 provides

$$e_0^{[6]}, \alpha_1^{[6]}, \alpha_2^{[6]}, \dots, \alpha_{2n-2}^{[6]},\ (\alpha_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]},\ G_{2n}^{[2]},\ \Delta_n^{[6]}\ \vdash\ e_0^{[6]}$$

and, hence,

$$e_0^{[6]}, \alpha_1^{[6]}, \alpha_2^{[6]}, \dots, \alpha_{2n-2}^{[6]},\ \Big((t_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]} \wedge (f_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]}\Big),\ G_{2n}^{[2]},\ \Delta_n^{[6]}\ \vdash\ e_0^{[6]}$$

By Lemma 1(d) we conclude with the desired (25). ■

Corollary 1. *If the statement (1) is valid then the following sequent is derivable in Lambek:*

$$(e_0^{[6n]} \cdot q_0)^{[2]}, G_1^{[2]}, G_2^{[4]}, \dots, G_{2n-1}^{[4n-2]}, G_{2n}^{[4n]},\ \Delta_n^{[6n+2]}\ \vdash\ e_0^{[6n+2]} \tag{27}$$

where

$$G_1 = (q_0 \backslash \big((t_1^{[6n-2]} \cdot q_1)^{[2]} \wedge (f_1^{[6n-2]} \cdot q_1)^{[2]}\big)) \tag{28}$$

$$G_2 = (q_1 \backslash \big((t_2^{[6n-6]} \cdot q_2)^{[3]} \wedge (f_2^{[6n-6]} \cdot q_2)^{[3]}\big)^{[1]}) \tag{29}$$

$$\dots$$

$$G_{2n-1} = (q_{2n-2} \backslash \big((t_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]} \wedge (f_{2n-1}^{[4]} \cdot q_{2n-1})^{[2]}\big)) \tag{30}$$

$$G_{2n} = (q_{2n-1} \backslash \big((t_{2n} \cdot q_{2n})^{[3]} \wedge (f_{2n} \cdot q_{2n})^{[3]}\big)^{[1]}) \tag{31}$$

Proof. By the bottom-up induction following the previous lemmas. ■

The direction from derivable sequents to winning trees is provided by Lemma 6.

Lemma 6. *If the sequent (27) is derivable in Lambek then the statement (1) is valid.*

Proof Sketch. Being derivable in Lambek calculus, the sequent (27) is derivable in linear logic. Replacing b with $\bot$, we get that $A^{bb} \equiv A$, resulting in that we can confine ourselves to Horn-like formulas, similar to (11) and (12), with the leading propositions from Remark 3. In its turn, such a Horn-like derivation can be transformed into a Horn-like tree program (see [11,12,18]), which in fact happens to be a winning strategy for the statement (1).

This concludes the proof of Lemma 6 and thereby the proof of Theorem 1. ■

In fact, we have proved a more general result.

Corollary 2. *Let L be a calculus that includes $\mathcal{L}(\backslash, \wedge)$, with or without Lambek's restriction, and is in turn included in linear logic. Then the fragment of L, which uses only one implication and conjunction, is PSPACE-hard.*

Proof. Given an instance of quantified Boolean formula (1), we take the sequent (27) and prove that there exists a winning tree if and only if (27) is derivable in L. Namely, if there is a winning tree, that sequent is derivable in $\mathcal{L}(\backslash, \wedge)$ with Lambek's restriction, and thereby in the corresponding fragment of L. On the other hand, if that sequent is derivable in L, then, repeating proof of Lemma 6 for the derivation in linear logic, we conclude that there exists a winning tree. ■

We can also modify this technique to establish PSPACE-hardness for the fragment $\mathcal{L}(\backslash, \vee)$, which includes only one implication and disjunction.

Theorem 2. *The fragment $\mathcal{L}(\backslash, \vee)$ is PSPACE-hard.*

We give a proof sketch in Appendix A.

3 Grammars Based on the Lambek Calculus with Disjunction

Theorem 3. *For any fragment of the Lambek calculus with conjunction and disjunction, which includes at least one division operation, $\backslash$, and disjunction, $\vee$, the class of languages generated by categorial grammars based on this calculus (in particular, the class of languages generated by $\mathcal{L}(\backslash, \vee)$-grammars) is closed under finite intersections.*

This theorem immediately yields the following corollary.

Corollary 3. *Grammars based on $\mathcal{L}(\backslash, \vee)$ can generate arbitrary finite intersections of context-free languages.*

Moreover, $\mathcal{L}(\backslash, \vee)$ also captures images of such intersections under *alphabetic homomorphisms.* A alphabetic homomorphism is a mapping $h \colon \Sigma_1^+ \to \Sigma_2^+$ of words over one alphabet to words of another one, such that $h(\Sigma_1) \subseteq \Sigma_2$ and $h(uv) = h(u)\,h(v)$ for any $u, v \in \Sigma_1^+$. The class of languages generated by $\mathcal{L}(\backslash, \vee)$-grammars is closed under alphabetic homomorphisms. Indeed, if the grammar $\mathcal{G} = \langle \Sigma_1, \rhd, H \rangle$ generates language M, then $\mathcal{G}_h = \langle \Sigma_2, \rhd_h, H \rangle$, where $a \rhd_h A$ iff $b \rhd A$ for some $b \in h^{-1}(a)$, generates $h(M)$. This yields the following stronger corollary.

Corollary 4. *Grammars based on $\mathcal{L}(\backslash, \vee)$ can generate all language of the form $h(M_1 \cap \ldots \cap M_k)$, where M_1, ..., M_k are context-free and h is a alphabetic homomorphism.*

Notice that this extension of Corollary 3 is non-trivial, since $h(M_1 \cap M_2)$ is not always equal to $h(M_1) \cap h(M_2)$. There is an example by Păun [22] of a language which is not a finite intersection of context-free languages, but can be obtained from such an intersection by applying a alphabetic homomorphism: $\{a^{2n^2} \mid n \geq 1\} = h(\{(a^n b^n)^n \mid n \geq 1\})$, where $h(a) = h(b) = a$.

Before proving Theorem 3, we establish several technical lemmata. The first one is a simplified version of Kanazawa's [9] Lemma 13.

Definition 4. *Let the set of variables include two disjoint subsets,* Var_1 *and* Var_2*. A formula is called a P_i-formula if it includes only variables from* Var_i *($i = 1, 2$).*

Lemma 7. *Let Γ and Δ sequences consisting of P_1-formulae and P_2-formulae, in an arbitrary order. Let B be a P_2-formula and C be a P_1-formula. Then the sequent $\Gamma, B, \Delta \vdash C$ is not derivable.*

Proof. Induction on a cut-free derivation. The sequent in question could not be in axiom, because then $B = C$, and P_1-formulae and P_2-formulae do not intersect.

Now consider the last rule applied in the derivation. If it is a one-premise rule, i.e., one of $\mathbf{L}\cdot$, $\mathbf{R}/$, $\mathbf{R}\backslash$, $\mathbf{L}\wedge$, $\mathbf{R}\vee$, then its premise also satisfies the conditions of the lemma, and such a sequent, by induction hypothesis, could not be derivable. Contradiction. The same happens with $\mathbf{L}\vee$ and $\mathbf{R}\wedge$, where both premises are not derivable by induction hypothesis. For $\mathbf{R}\cdot$, induction hypothesis yields non-derivability of the premise into which the B formula goes.

The most tricky cases are $\mathbf{L}\backslash$ and $\mathbf{L}/$. We consider the former; the latter is dual. Recall that $\mathbf{L}\backslash$ is a rule of the form

$$\frac{\Phi \vdash E \quad \Sigma_1, F, \Sigma_2 \vdash C}{\Sigma_1, \Phi, E \setminus F, \Sigma_2 \vdash C}\ \mathbf{L}\backslash$$

Now the question is where comes B. There are three possible cases.

Case 1: B is in Σ_1 or Σ_2. In this case, the right premise satisfies the condition of the lemma, and is therefore not derivable by induction hypothesis.
Case 2: B is in Φ. In this case, let us consider $E \setminus F$, which is either a P_1-formula or a P_2-formula. If $E \setminus F$ is a P_1-formula, then so is E, and the left premise, $\Phi \vdash E$, satisfies the condition of the lemma and is not derivable by induction. If $E \setminus F$ is a P_2-formula, then so is F, and now the right premise $\Sigma_1, F, \Sigma_2 \vdash C$, satisfies the condition of the lemma, and induction hypothesis yields its non-derivability.
Case 3: $B = E \setminus F$. The right premise satisfies the condition of the lemma (F is a P_2-formula and C is a P_1-formula), and is therefore not derivable my induction hypothesis. ■

The next 4 lemmas are proved by straightforward induction on derivation. We put their proofs in Appendix B.

Definition 5. *Define the notion of* strictly positive *occurrence of a subformula inside a formula:*

- *A is strictly positive in itself;*
- *C occurs strictly positively in $A \setminus B$ if and only if it occurs strictly positively in B; the same for $B \,/\, A$;*
- *C occurs strictly positively in $A \cdot B$ if and only if if occurs strictly positively in A or in B; the same for $A \vee B$ and $A \wedge B$.*

Lemma 8 (Disjunctive Property). *Let F_1 and F_2 be arbitrary formulae, and $E_1, \ldots, E_n$ be formulae without $\wedge$ in which subformulae of the form $A \vee B$ do not occur strictly positively. Then the derivability $E_1, \ldots, E_n \vdash F_1 \vee F_2$ implies the derivability of $E_1, \ldots, E_n \vdash F_i$ for $i = 1$ or 2.*

Lemma 9. *If $F_1, \ldots, F_n$ do not include variable b, then $F_1, \ldots, F_n \vdash b$ is not derivable.*

Lemma 10. *If $F_1, \ldots, F_\ell, E_1 \setminus b, \ldots, E_k \setminus b, b \vdash b$ is derivable and $F_1, \ldots, F_\ell$ do not include b, then $k = \ell = 0$.*

Lemma 11. *If $F_1, \ldots, F_\ell, E_1 \setminus b, \ldots, E_k \setminus b \to b$ is derivable and $F_1, \ldots, F_\ell$ do not include b, then $F_1, \ldots, F_\ell, E_1 \setminus b, \ldots, E_{k-1} \setminus b \to E_k$ is derivable.*

The following lemma is the key one for the proof of Theorem 3.

Lemma 12. *Let $A_1, \ldots, A_n, C$ be P_1-formulae, $B_1, \ldots, B_n, D$ be P_2-formulae, and let b be a fresh variable, $b \notin \mathrm{Var}_1 \cup \mathrm{Var}_2$. Also suppose that no formula of the form $E \vee F$ occurs in A_1, ..., A_n, B_1, ..., or B_n strictly positively. Then the sequent*

$$((A_1 \setminus b) \vee (B_1 \setminus b)) \setminus b, \ldots, ((A_n \setminus b) \vee (B_n \setminus b)) \setminus b \vdash ((C \setminus b) \vee (D \setminus b)) \setminus b$$

is derivable if and only if so are $A_1, \ldots, A_n \vdash C$ and $B_1, \ldots, B_n \vdash D$.

In the notations of Subsect. 2.1, the first sequent of this lemma can be shortly written as $(A_1^b \vee B_1^b)^b, \ldots, (A_n^b \vee B_n^b)^b \vdash (C^b \vee D^b)^b$. Though $(A^b \vee B^b)^b$ is not equivalent to $A \wedge B$, and even not equivalent to $(A \wedge B)^{bb}$, this sequent happens to be equiderivable with $A_1 \wedge B_1, \ldots, A_n \wedge B_n \vdash C \wedge D$, which Kanazawa [9] used for his intersection construction with additive conjunction.

Proof. The **"if"** part is straightforwardly established by direct derivation.

For the **"only if"** part we first use the reversibility of $\mathbf{R}\setminus$ and $\mathbf{L}\vee$, which yields derivability of the following two sequents:

$$C \setminus b, ((A_1 \setminus b) \vee (B_1 \setminus b)) \setminus b, \ldots, ((A_n \setminus b) \vee (B_n \setminus b)) \setminus b \vdash b$$
$$D \setminus b, ((A_1 \setminus b) \vee (B_1 \setminus b)) \setminus b, \ldots, ((A_n \setminus b) \vee (B_n \setminus b)) \setminus b \vdash b.$$

Let us analyze the derivation of the first sequent. We claim derivability of $K_1, \ldots, K_n, C \setminus b \vdash b$, where each K_i is either A_i or B_i. In order to prove it, consider a more general statement, the derivability of

$$K_{n-k}, \ldots, K_n, C \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_{n-k-1}^b \vee B_{n-k-1}^b)^b \vdash b.$$

This statement is proved by induction on k. Indeed, for $k = 0$ derivability of this sequent was shown above. For the induction step, suppose that

$$K_{n-k}, \ldots, K_n, C \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_{n-k-1}^b \vee B_{n-k-1}^b)^b \vdash b$$

is derivable and apply Lemma 11, which yields derivability of

$$K_{n-k}, \ldots, K_n, C \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_{n-k-2}^b \vee B_{n-k-2}^b)^b \vdash A_{n-k-1}^b \vee B_{n-k-1}^b.$$

Now apply the Disjunctive Property (Lemma 8) and obtain derivability of

$$K_{n-k}, \ldots, K_n, C \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_{n-k-2}^b \vee B_{n-k-2}^b)^b \vdash K_{n-k-1} \setminus b,$$

where K_{n-k-1} is either A_{n-k-1} or B_{n-k-1}. Reversion of $\mathbf{L}\backslash$ yields the necessary

$$K_{n-(k+1)}, K_{n-k}, \ldots, K_n, C \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_{n-k-2}^b \vee B_{n-k-2}^b)^b \vdash b.$$

In the end of the induction, for $k = n-1$, we get $K_1, \ldots, K_n, C \setminus b \vdash b$, and one more application of Lemma 11 yields $K_1, \ldots, K_n \vdash C$.

Now recall that C is a P_1-formula, and each of $K_1, \ldots, K_n$ is a P_1-formula or a P_2-formula. If $K_i = B_i$ for some i, i.e., it is a P_2-formula, then $K_1, \ldots, K_n \vdash C$ is not derivable by Lemma 7. Thus, for all i we have $K_i = A_i$, and obtain the needed sequent $A_1, \ldots, A_n \vdash C$.

The same reasoning applied to $D \setminus b, (A_1^b \vee B_1^b)^b, \ldots, (A_n^b \vee B_n^b)^b \vdash b$ yields $B_1, \ldots, B_n \vdash D$. ■

Lemma 12, together with Lemma 2 of Subsect. 2.1, yield the following corollary:

Corollary 5. *Let $A_1, \ldots, A_n, C$ be P_1-formulae, $B_1, \ldots, B_n, D$ be P_2-formulae, and let b and c be fresh variables ($b, c \notin \mathrm{Var}_1 \cup \mathrm{Var}_2$, $b \neq c$). Then the sequent*

$$((A_1^{cc})^b \vee (B_1^{cc})^b)^b, \ldots, ((A_n^{cc})^b \vee (B_1^{cc})^b)^b \vdash ((C^{cc})^b \vee (D^{cc})^b)^b$$

is derivable if and only if so are $A_1, \ldots, A_n \vdash C$ and $B_1, \ldots, B_n \vdash D$.

Proof. The only strictly positive subformula or A_i^{cc} and B_j^{cc} is c. Thus, there is no strictly positive subformula the form $E \vee F$, and we can apply Lemma 12. This lemma yields the fact that

$$((A_1^{cc})^b \vee (B_1^{cc})^b)^b, \ldots, ((A_n^{cc})^b \vee (B_1^{cc})^b)^b \vdash ((C^{cc})^b \vee (D^{cc})^b)^b$$

is derivable if and only if so are $A_1^{cc}, \ldots, A_n^{cc} \vdash C^{cc}$ and $B_1^{cc}, \ldots, B_n^{cc} \vdash D^{cc}$. For these two sequents, we apply Lemma 2 and replace these sequents with equiderivable ones, $A_1, \ldots, A_n \vdash C$ and $B_1, \ldots, B_n \vdash D$. ■

Now we are ready to prove the main result of this section.

Proof (of Theorem 3). Consider two categorial grammars over the same alphabet, $\mathcal{G}_1 = \langle \Sigma, \triangleright_1, H_1 \rangle$ and $\mathcal{G}_2 = \langle \Sigma, \triangleright_2, H_2 \rangle$. Without loss of generality we can suppose that all formulae of $\mathcal{G}_i$ are P_i-formulae (otherwise just rename the variables). Construct the new grammar $\mathcal{G} = \langle \Sigma, \triangleright, H \rangle$, where, for each $a \in \Sigma$ we postulate $a \triangleright ((A^{cc})^b \vee (B^{cc})^b)^b$ for any A and B such that $a \triangleright_1 A$ and $a \triangleright_2 A$; $H = ((H_1^{cc})^b \vee (H_2^{cc})^b)^b$. Here b and c are fresh variables: b and c are distinct and do not occur in $\mathcal{G}_1$ or $\mathcal{G}_2$. By Corollary 5 a word $a_1, \ldots a_n$ is accepted by $\mathcal{G}$ if and only if it is accepted by both $\mathcal{G}_1$ and $\mathcal{G}_2$. Therefore, the language generated by $\mathcal{G}$ is exactly the intersection of languages generated by $\mathcal{G}_1$ and $\mathcal{G}_2$. ■

4 Concluding Remarks

In this paper we have proved two refined results on the complexity of the Lambek calculus enriched either with conjunction or disjunction. Namely, we have established PSPACE-completeness for small fragments $\mathcal{L}(\backslash, \wedge)$ and $\mathcal{L}(\backslash, \vee)$. Notice that the encoding used in this paper is more involved than the encodings from [6,8,10,18], because here we were not allowed to use the product (multiplicative conjunction) and one of the divisions. Besides, we have proved that $\mathcal{L}(\backslash, \vee)$-grammars generate all finite intersections of context-free languages and images of such intersections under alphabetic homomorphisms.

There are some questions left for future work. First, we see that in our constructions for proving PSPACE-hardness involve formulae of unbounded implication depth. On the other hand, for the original Lambek calculus without additive connectives, which is NP-complete, Pentus [25], nevertheless, a polynomial time decision procedure for the case where the order (a complexity measure similar to implication depth) of formulae is bounded by a constant d, fixed in advance. The degree of the polynomial, of course, depends on d. For the Lambek calculus with additives, we plan to show that it is not the case. Following the basic ideas of our encoding, with the formulas of the implication nesting depth bounded by some constant, we intend to simulate at least co-NP-hardness of our small fragment $\mathcal{L}(\backslash, \wedge)$ with one implication and conjunction.

Another open question is to describe the class of languages generated by Lambek grammars with additive connectives. In particular, Kuznetsov and Okhotin [14,16] show that such grammars can generate languages described by conjunctive grammars [21]. Such grammars can be quite powerful, for example, can generate $\{a^{4^n} \mid n \geq 1\}$ [7]. It is yet unknown whether all such languages can be generated by $\mathcal{L}(\backslash, \vee)$-grammars.

Acknowledgments. We would like to thank the anonymous referees for their helpful comments.

Financial Support

The work of Max Kanovich and Andre Scedrov was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Moscow, Russia. The work of Stepan Kuznetsov was supported by the Young Russian Mathematics award, by the grant MK-430.2019.1 of the President of Russia, and by the Russian Foundation for Basic Research grant 18-01-00822. Section 2 was contributed by Kanovich and Scedrov. Section 3 was contributed by Kuznetsov. Sections 1 and 4 were contributed jointly and equally by all co-authors.

A PSPACE-completeness of the fragment $\mathcal{L}(\backslash, \vee)$

In this section we will modify Sect. 2 to establish PSPACE-completeness for the fragment $\mathcal{L}(\backslash, \vee)$, which includes only one implication and disjunction.

Remark 4. For the sake of readability, we conceive of the formula $((A \cdot B) \backslash C)$ as abbreviation for $(B \backslash (A \backslash C))$. In particular, $(A \cdot B)^b$ is abbreviation for

$(B\backslash(A\backslash b))$. Because of Lemma 1, the formula $(A^b \wedge B^b)$ is conceived of as abbreviation for $((A \vee B)\backslash b)$ within this section.

Theorem 4. *The fragment $\mathcal{L}(\backslash, \vee)$ is PSPACE-complete.*

Proof Sketch. We start with the equality (2), assuming that $\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}$, α_{2n} are given.

To prove Lemma 14, the "disjunction analog" of Lemma 3, we modify the basic material given in the "conjunction" Definition 3 by means of Definitions 6 and 7 working within the $\mathcal{L}(\backslash, \vee)$ fragment.

Definition 6. *For $1 \leq i \leq 2n$, let $E_{\ell,i,\beta}$ denote the formula: $(c_{\ell,i}\backslash(\beta\backslash c_{\ell,i-1}))$. Let F_ℓ denote: $(q_{2n}\backslash c_{\ell,2n})$, and H_ℓ denote: $(c_{\ell,0}\backslash(e_0\backslash e_0))$.*

Lemma 13. *The following "verifying" sequent is derivable in Lambek calculus*

$$e_0, \alpha_1, \alpha_2, \ldots, \alpha_{2n}, q_{2n}, F_\ell, E_{\ell,2n,\alpha_{2n}}, E_{\ell,2n-1,\alpha_{2n-1}}, \ldots E_{\ell,2,\alpha_2}, E_{\ell,1,\alpha_1}, H_\ell \ \vdash \ e_0$$

Proof. By the inverse induction on i: $\alpha_{i-1}, \alpha_i, c_{\ell,i}, E_{\ell,i,\alpha_i} \ \vdash \ (\alpha_{i-1} \cdot c_{\ell,i-1})$ ■

Definition 7. *We introduce the following formulas:*

$$\begin{cases} \widetilde{F} = \left(\bigvee_{\ell=1}^{m} (F_\ell)^b\right)^b, \quad \widetilde{H} = \left(\bigvee_{\ell=1}^{m} (H_\ell)^b\right)^b \\ \widetilde{E_i} = \left(\bigvee_{1\leq\ell\leq m,\ E_{\ell,i,\beta}\in\mathcal{E}_{\ell,i}} (E_{\ell,i,\beta})^b\right)^b \end{cases} \tag{32}$$

where a one- or two-element set of formulas, $\mathcal{E}_{\ell,i}$, is defined as follows:

(1) $\mathcal{E}_{\ell,i} = \{ E_{\ell,i,t_i} \}$, if the conjunct C_ℓ contains the variable x_i,
(2) $\mathcal{E}_{\ell,i} = \{ E_{\ell,i,f_i} \}$, if the conjunct C_ℓ contains $\neg x_i$,
(3) $\mathcal{E}_{\ell,i} = \{ E_{\ell,i,t_i},\ E_{\ell,i,f_i} \}$, if C_ℓ contains neither x_i, nor $\neg x_i$.

By applying (2) and Lemma 1, we get the desired verification:

Lemma 14. *The following sequent is derivable in Lambek*

$$e_0^{bb}, \alpha_1^{bb}, \alpha_2^{bb}, \ldots, \alpha_{2n-1}, (\alpha_{2n} \cdot \ q_{2n})^{bb},\ \varDelta_n \ \vdash \ e_0^{bb}$$

where $\varDelta_n$ is a sequence of formulas: $\varDelta_n = \widetilde{F}, \widetilde{E}_{2n}, \widetilde{E}_{2n-1}, \ldots, \widetilde{E_2}, \widetilde{E_1}, \widetilde{H}$

Corollary 6. *It suffices to follow the line of reasoning in Sect. 2 to find appropriate G_1, G_2, $\ldots$, G_{2n-1}, G_{2n}, such that the following sequent is derivable in Lambek calculus if and only if the statement (1) is valid:*

$$(e_0^{[4n]} \cdot q_0)^{[2]}, G_1^{[2]}, G_2^{[4]}, \ldots, G_{2n-1}^{[4n-2]}, G_{2n}^{[4n]},\ \varDelta_n^{[4n]} \ \vdash \ e_0^{[4n+2]} \tag{33}$$

B Proofs of Technical Lemmas for Section 3

Proof (of Lemma 8). Induction on derivation. The sequent in question could not be an axiom, since the antecedent of $F_1 \vee F_2 \vdash F_1 \vee F_2$ includes $F_1 \vee F_2$ in a strictly positive position. Consider the last rule applied in the derivation. It could be $\mathbf{L}\backslash$, $\mathbf{L}/$, $\mathbf{L}\cdot$, or $\mathbf{R}\vee$. Rules with $\wedge$ cannot be used, since there are no $\wedge$'s in the antecedent, and the main connective of the succedent is $\vee$. If the last rule is $\mathbf{R}\vee$, we immediately reach our goal.

If the derivation ends with an application of $\mathbf{L}\cdot$:

$$\frac{E_1, \dots, E_i', E_i'', \dots, E_n \vdash F_1 \vee F_2}{E_1, \dots, E_i' \cdot E_i'', \dots, E_n \vdash F_1 \vee F_2}\ \mathbf{L}\cdot$$

then we apply the induction hypothesis, get $E_1, \dots, E_i', E_i'', \dots, E_n \vdash F_i$ ($i = 1$ or 2) and apply $\mathbf{L}\cdot$ to this sequent, which yields our goal.

For $\mathbf{L}\backslash$, we get the following

$$\frac{E_{i+1}, \dots, E_{j-1} \vdash E_j' \quad E_1, \dots, E_i, E_j'', \dots, E_n \vdash F_1 \vee F_2}{E_1, \dots, E_i, E_{i+1}, \dots, E_{j-1}, E_j' \setminus E_j'', \dots, E_n \vdash F_1 \vee F_2}\ \mathbf{L}\backslash$$

and notice that the antecedent of the right premise still satisfies the conditions of the lemma, thus we can apply induction hypothesis. The induction hypothesis yields $E_1, \dots, E_i, E_j'', \dots, E_n \vdash F_i$. Applying $\mathbf{L}/$ with the same left premise, $E_{i+1}, \dots, E_{j-1} \vdash E_j'$, yields our goal.

The $\mathbf{L}/$ case is symmetric. ■

Proof (of Lemma 9). Induction on derivation. The axiom should be of the form $b \vdash b$, which violates the condition. For each inference rule, we apply the induction hypothesis for the premise from which the succedent b comes. ■

Proof (of Lemma 10). Induction on derivation. Induction base is axiom $b \vdash b$. Consider the last rule applied. If it is one of the one-premise rules, then we use the induction hypothesis for the only premise. For applications of $\mathbf{L}/$ or $\mathbf{L}\backslash$, if the rightmost occurrence of b goes to the right premise, we again directly use the induction hypothesis. Notice that for $\mathbf{L}\backslash$ this is always the case. The other rule, $\mathbf{L}/$, however, can decompose one of the F_i and take the rightmost b to the left premise:

$$\frac{F_{i+1}, \dots, F_\ell, E_1 \setminus b, \dots, E_k \setminus b, b \vdash F_i'' \quad F_1, \dots, F_i' \vdash b}{F_1, \dots, F_i' \,/\, F_i'', F_{i+1}, \dots, F_\ell, E_1 \setminus b, \dots, E_k \setminus b, b \vdash b}$$

The right premise, however, now is not derivable by Lemma 9. Contradiction. ■

Proof (of Lemma 11). Induction on derivation again. Any one-premise rule applied for one of the F_i, as well as $\mathbf{L}/$ or $\mathbf{L}\backslash$ which keeps $E_k \setminus b$ in the right premise, is handled by directly using the induction hypothesis and applying the same rule. The situation where $\mathbf{L}/$ takes $E_k \setminus b$ to the left premise leads to contradiction with Lemma 9, exactly as in the proof of the previous lemma. ■

References

1. Abrusci, V.M.: A comparison between Lambek syntactic calculus and intuitionistic linear logic. Zeitschr. Math. Logik Grundl. Math. (Math. Logic Q.) **36**, 11–15 (1990)
2. Bar-Hillel, Y., Gaifman, C., Shamir, E.: On categorial and phrase-structure grammars. Bull. Res. Council Israel **9F**, 1–16 (1960)
3. Buszkowski, W.: The equivalence of unidirectional Lambek categorial grammars and context-free grammars. Zeitschr. Math. Log. Grundl. Math. **31**, 369–384 (1985)
4. Carpenter, B.: Type-Logical Semantics. MIT Press (1998)
5. Girard, J.-Y.: Linear logic. Theor. Comput. Sci. **50**(1), 1–101 (1987)
6. Horčík, R., Terui, K.: Disjunction property and complexity of substructural logics. Theor. Comput. Sci. **412**(31), 3992–4006 (2011)
7. Jeż, A.: Conjunctive grammars can generate non-regular unary languages. Internat. J. Found. Comput. Sci. **19**(3), 597–615 (2008)
8. Kanazawa, M.: The Lambek calculus enriched with additional connectives. J. Log. Lang. Inform. **1**(2), 141–171 (1992)
9. Kanazawa, M.: Lambek calculus: recognizing power and complexity. In: Gerbrandy, J., et al. (eds.) JFAK. Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday. Amsterdam University Press, Vossiuspers (1990)
10. Kanovich, M.I.: Horn fragments of non-commutative logics with additives are PSPACE-complete. In: Proceedings 1994 Annual Conference of the European Association for Computer Science Logic, Kazimierz, Poland (1994)
11. Kanovich, M.: The direct simulation of Minsky machines in linear logic. In: Girard, J.-Y., Lafont, Y., Regnier, L. (eds.) Advances in Linear Logic. London Mathematical Society Lecture Notes, vol. 222, pp. 123–145. Cambridge University Press, Cambridge (1995)
12. Kanovich, M.I.: The undecidability theorem for the Horn-like fragment of linear logic (Revisited). Math. Struct. Comput. Sci. **26**(5), 719–744 (2016)
13. Kanovich, M., Kuznetsov, S., Nigam, V., Scedrov, A.: Subexponentials in non-commutative linear logic. Math. Struct. Comput. Sci. (2018). Part of Dale Miller's Festschrift
14. Kuznetsov, S.: Conjunctive grammars in Greibach normal form and the Lambek calculus with additive connectives. In: Morrill, G., Nederhof, M.-J. (eds.) FG 2012-2013. LNCS, vol. 8036, pp. 242–249. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39998-5_15
15. Kuznetsov, S.L.: On translating Lambek grammars with one division into context-free grammars. Proc. Steklov Inst. Math. **294**(1), 129–138 (2016)
16. Kuznetsov, S., Okhotin, A.: Conjunctive categorial grammars. In: Proceedings of Mathematics of Language (2017)
17. Lambek, J.: The mathematics of sentence structure. Amer. Math. Monthly **65**, 154–170 (1958)
18. Lincoln, P., Mitchell, J., Scedrov, A., Shankar, N.: Decision problems for propositional linear logic. Ann. Pure Appl. Logic **56**, 239–311 (1992)
19. Moot, R., Retoré, C.: The Logic of Categorial Grammars. A Deductive Account of Natural Language Syntax and Semantics. LNCS, vol. 6850. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31555-8
20. Morrill, G.V.: Categorial Grammar: Logical Syntax, Semantics, and Processing. Oxford University Press (2011)

21. Okhotin, A.: Conjunctive grammars. J. Autom. Lang. Combin. **6**(4), 519–535 (2001)
22. Păun, G.: A note on the intersection of context-free languages. Fundam. Inform. **3**(2), 135–139 (1980)
23. Pentus, M.: Lambek grammars are context-free. In: Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science (LICS 1993), pp. 429–433. IEEE Computer Society Press (1993)
24. Pentus, M.: Lambek calculus is NP-complete. Theor. Comput. Sci. **357**(1–3), 186–201 (2006)
25. Pentus, M.: A polynomial-time algorithm for Lambek grammars of bounded order. Linguist. Anal. **36**(1–4), 441–471 (2010)
26. Savateev, Yu.: Unidirectional Lambek grammars in polynomial time. Theory Comput. Syst. **46**(4), 662–672 (2010)
27. Savateev, Yu.: Product-free Lambek calculus is NP-complete. Ann. Pure Appl. Logic **163**(7), 775–788 (2012)