The 'Long Rule' in the Lambek Calculus with Iteration: Undecidability without Meets and Joins

Stepan Kuznetsov [1]

Steklov Mathematical Institute of RAS
8 Gubkina St., Moscow 119991, Russia

Abstract

We consider the Lambek calculus extended with positive iteration as a unary connective. The choice of positive iteration, not Kleene star, is dictated by Lambek's antecedent non-emptiness restriction. Usually iteration is axiomatized either by an inductive schema or by an ω-rule. We consider an intermediate system with a rule which we call the 'long rule,' which reduces iteration of A to explicit treatment of powers of A up to the k-th one, and reusing iteration in the form $A^k \cdot A^+$. In the presence of additive disjunction (union), the 'long rule' is easily derivable. For the 'pure' Lambek calculus without additives this is not the case. For the system with the 'long rule' we prove undecidability. We also investigate connections of this system with the standard inductive-style one.

Keywords: Lambek calculus, iteration, undecidability

1 Introduction

Iteration, or Kleene star, is one of the most basic and at the same time one of the most intriguing algebraic operations appearing in theoretical computer science. Following the line of work by Pratt [23] and Kozen [12], we consider substructural (algebraic) non-commutative logics with two implications (divisions) and iteration as a modality (cf. [24, § 9.5]). The idea of division operations we consider throughout this paper goes back to Krull [14]. From the logical point of view divisions were introduced in the Lambek calculus [19]. The Lambek calculus is a non-commutative intuitionistic variant of Girard's linear logic [7], in the multiplicative-only language (see Abrusci [1]). Thus, the system we are going to consider is *the Lambek calculus* (or non-commutative intuitonistic multiplicative-only linear logic) *extended with iteration.*

[1] sk@mi-ras.ru
This work was performed at the Steklov International Mathematical Center and supported by the Ministry of Science and Higher Education of the Russian Federation (agreement no. 075-15-2019-1614).

Action logic, denoted by **ACT** and introduced by Pratt and Kozen, includes, besides divisions and iteration, also lattice operations: join and meet. Thus, action logic can be viewed as an extension of the multiplicative-additive ('full') Lambek calculus. Following the standard definition of Kleene star as a fixed point, Pratt axiomatizes it using an induction axiom ('pure induction'). In contrast, later works of Buszkowski and Palka [3,21,5] feature a stronger system called *infinitary action logic,* $\mathbf{ACT}_\omega$, with an ω-rule for Kleene star. Buszkowski and Palka show that $\mathbf{ACT}_\omega$ is Π^0_1-complete. Thus, it is undecidable and strictly stronger than action logic with induction axioms/rules of any kind. (As noticed by the author in [16], there exist variations of induction rules which yield systems which are strictly between **ACT** and $\mathbf{ACT}_\omega$.) The question of decidability for **ACT**, posed by Kozen in 1994, was recently solved, by the author of this paper, negatively [18]. This undecidability result applies to the whole range of systems between **ACT** and $\mathbf{ACT}_\omega$. Moreover, its modification [17] gives Σ^0_1-completeness for any logic in this range, provided it is recursively enumerable.

This paper continues the line of [18] and [17]. We now focus on the extension of the Lambek calculus with iteration, but *without* join and meet. Another distinctive feature of the system considered here is the so-called *Lambek's non-emptiness restiction.* Algebraically it means that we allow models without the unit. Lambek's restriction was originally motivated by linguistic applications of the Lambek calculus (see [20, § 2.5]). Here it will help to simplify some of the technicalities in the proofs. We conjecture that our results will also be valid without Lambek's restriction. However, we do not yet claim this, since some technicalities depend on the non-emptiness restriction.

In the presence of Lambek's restriction, we cannot introduce Kleene star itself: one of the axioms for Kleene star, $\mathbf{1} \vdash A^*$, includes the unit (empty antecedent). Instead, we introduce *positive iteration,* A^+. Interestingly, in his pioneering work [10], Kleene himself also avoided using the unit ('empty event') and introduced a binary iteration operation $A * B$, which means $A^* \cdot B$ ("several times A, then B"). In Kleene's notation, A^+ is $A * A$.

2 Preliminaries

Let us formally introduce the Lambek calculus with positive iteration, denoted by $\mathbf{L}^+$. Formulae of $\mathbf{L}^+$ are built from variables using three binary connectives: $\cdot$ (product), $\backslash$ (left division), $/$ (right division), and one unary connective: $^+$ (positive iteration). We formulate $\mathbf{L}^+$ as a sequent calculus, though cut is, unfortunately, not going to be eliminable. Sequents of $\mathbf{L}^+$ are expressions of the form $A_1, \ldots, A_n \vdash B$, where $A_1, \ldots, A_n, B$ are formulae, $n \geq 1$ (empty antecedents are disallowed). Formulae are denoted by capital Latin letters; capital Greek letters stand for sequences of formulae, possibly empty.

The core of $\mathbf{L}^+$ is the Lambek calculus $\mathbf{L}$, with the following axioms and rules of inference:

$$\overline{A \vdash A}$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, \Pi, A \setminus B, \Delta \vdash C} \qquad \frac{A, \Pi \vdash B}{\Pi \vdash A \setminus B}, \text{ where } \Pi \text{ is non-empty}$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, B \mathbin{/} A, \Pi, \Delta \vdash C} \qquad \frac{\Pi, A \vdash B}{\Pi \vdash B \mathbin{/} A}, \text{ where } \Pi \text{ is non-empty}$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \cdot B, \Delta \vdash C} \qquad \frac{\Pi \vdash A \quad \Delta \vdash B}{\Pi, \Delta \vdash A \cdot B} \qquad \frac{\Pi \vdash A \quad \Gamma, A, \Delta \vdash C}{\Gamma, \Pi, \Delta \vdash C}\ (cut)$$

Axioms and rules for iteration reflect the idea that, algebraically, a^+ should be the least (that is, the strongest) b such that $a \vdash b$ and $a \cdot b \vdash b$:

$$\frac{}{A \vdash A^+} \qquad \frac{}{A, A^+ \vdash A^+} \qquad \frac{A \vdash B \quad A, B \vdash B}{A^+ \vdash B}$$

As one can see, iteration here is axiomatized in a non-sequential style; thus, cut is not eliminable in $\mathbf{L}^+$. Unfortunately, no cut-free sequential version for the inductive axiomatization of iteration is known, in the presence of divisions. Unsuccessful attempts were taken by Jipsen [9] and Pentus [22]. For the logic of Kleene algebras, without division (but with join), a cut-free circular hypersequential system was constructed by Das and Pous [6]. This became possible, because for Kleene algebras the inductively axiomatized logic is complete, that is, admits the ω-rule. For systems with division operations, this is not the case due to complexity reasons (a recursively enumerable set of sequents could not coincide with a Π_1^0-hard one).

As shown by Pratt [23], in the presence of division operations left iteration is also right. This means that the following axiom and rule are derivable in $\mathbf{L}^+$:

$$\frac{}{A^+, A \vdash A^+} \qquad \frac{A \vdash B \quad B, A \vdash B}{A^+ \vdash B}$$

A stronger version of $\mathbf{L}^+$ is obtained by introducing the ω-rule for iteration:

$$\frac{\Gamma, A, \Delta \vdash B \quad \Gamma, A, A, \Delta \vdash B \quad \Gamma, A, A, A, \Delta \vdash B \quad \ldots}{\Gamma, A^+, \Delta \vdash B}$$

Axioms for iteration can also be reformulated in a sequential style:

$$\frac{\Gamma_1 \vdash A \quad \ldots \quad \Gamma_n \vdash A}{\Gamma_1, \ldots, \Gamma_n \vdash A^+}\ (n \geq 1)$$

and in this infinitary system, denoted by $\mathbf{L}^+_\omega$, cut is eliminable. This is essentially due to Palka [21], with necessary modifications connected with Lambek's restriction.

Adding join ($\vee$) and meet ($\wedge$) with the following rules:

$$\frac{\Gamma, A_1, \Delta \vdash C \quad \Gamma, A_2, \Delta \vdash C}{\Gamma, A_1 \vee A_2, \Delta \vdash C} \qquad \frac{\Pi \vdash A_i}{\Pi \vdash A_1 \vee A_2}\ (i = 1, 2)$$

$$\frac{\Gamma, A_i, \Delta \vdash C}{\Gamma, A_1 \wedge A_2, \Delta \vdash C}\ (i = 1, 2) \qquad \frac{\Pi \vdash A_1 \quad \Pi \vdash A_2}{\Pi \vdash A_1 \wedge A_2}$$

to $\mathbf{L}^+$ and $\mathbf{L}^+_\omega$ yields $\mathbf{ACT}^+$ and $\mathbf{ACT}^+_\omega$ respectively. These are positive variants of ordinary and infinitary action logic. Complexity results for $\mathbf{ACT}^+$ and $\mathbf{ACT}^+_\omega$ can be proved by slight modifications of the proofs for systems without Lambek's restriction and with Kleene star instead of positive iteration. Thus, due to Buszkowski [3] and Palka [21] $\mathbf{ACT}^+_\omega$ is Π^0_1-complete; $\mathbf{ACT}^+$ is undecidable [18] (more precisely, Σ^0_1-complete [17]).

In the infinitary case, Buszkowski's Π^0_1-hardness result can be strengthened: $\mathbf{L}^+_\omega$, the system without join and meet, is already Π^0_1-hard [15]. In this paper, we investigate the possibility of performing a similar strengthening of the undecidability result for $\mathbf{ACT}^+$ [18] to $\mathbf{L}^+$. Namely, we prove undecidability not for $\mathbf{L}^+$ itself, but for a system very closely related to $\mathbf{L}^+$.

An important component of the undecidability proof for $\mathbf{ACT}^+$ is the so-called 'long rule' [18], formulated as follows:

$$\frac{A \vdash B \quad A, A \vdash B \quad \ldots \quad A^k \vdash B \quad A^k, A^+ \vdash B}{A^+ \vdash B}$$

Actually, this is a series of rules parametrized by k. In the presence of $\vee$, this rule can be easily derived, for any k, using cut with $A^+ \vdash A \vee A^2 \vee \ldots \vee A^k \vee (A^k \cdot A)$. This can be also performed without $\vee$, but with $\wedge$ and division operations [17]. Notice that the 'long rule' itself includes neither $\vee$, nor $\wedge$, but its derivation in $\mathbf{ACT}^+$ requires one of these connectives.

By $\mathbf{L}^+_\ell$ we denote $\mathbf{L}^+$ with the 'long rule' added as a rule of inference. More precisely, we include instances of the 'long rule' for each k.

The rest of this paper is organized as follows. In Section 3, we prove undecidability of $\mathbf{L}^+_\ell$. In Section 4, we show that, unlike $\mathbf{ACT}^+$ and $\mathbf{L}^+_\omega$, in $\mathbf{L}^+$ the 'long rule' is not *derivable*. The question whether a weaker property, *admissibility* of the 'long rule' in $\mathbf{L}^+$, holds is left open. Section 5 includes some concluding remarks and speculations.

We conclude this section by showing that the 'long rule' is derivable in $\mathbf{L}^+_\omega$ and presenting a contextified (sequent-style) version of the 'long rule.'

Lemma 2.1 *The 'long rule' is derivable in* $\mathbf{L}^+_\omega$*.*

Proof. In $\mathbf{L}^+_\omega$, one can easily derive $A^n \vdash A^+$ for any $n \geq 1$ (just use the right rule for iteration with $\Gamma_1 = \ldots = \Gamma_n = A$).

Now, given the premises of the 'long rule,' let us establish $A^m \vdash B$ for any $m \geq 1$. Indeed, if $m \leq k$, this sequent is explicitly given. If $m > k$, then we use cut:

$$\frac{A^{m-k} \vdash A^+ \quad A^k, A^+ \vdash B}{A^m \vdash B}\ (cut)$$

Now $A^+ \vdash B$ is derived by the ω-rule. $\square$

Lemma 2.2 *The following 'sequential version' of the 'long rule' is derivable in* $\mathbf{L}_\ell^+$*:*

$$\frac{\Gamma, A, \Delta \vdash B \quad \Gamma, A, A, \Delta \vdash B \quad \ldots \quad \Gamma, A^k, \Delta \vdash B \quad \Gamma, A^k, A^+, \Delta \vdash B}{\Gamma, A^+, \Delta \vdash B}$$

Proof. If $\Gamma = G_1, \ldots, G_s$, let $\bullet\Gamma = G_1 \cdot \ldots \cdot G_s$; similarly for $\bullet\Delta$. Now $\Gamma, A^+, \Delta \vdash B$ is derived by cut from $A^+ \vdash \bullet\Gamma \setminus B \,/\, \bullet\Delta$ and $\Gamma, \bullet\Gamma \setminus B \,/\, \bullet\Delta, \Delta \vdash B$. The latter is derivable in $\mathbf{L}$; the derivation for the former is by the 'long rule':

$$\frac{\dfrac{\Gamma, A, \Delta \vdash B}{A \vdash \bullet\Gamma \setminus B \,/\, \bullet\Delta} \quad \ldots \quad \dfrac{\Gamma, A^k, \Delta \vdash B}{A^k \vdash \bullet\Gamma \setminus B \,/\, \bullet\Delta} \quad \dfrac{\Gamma, A^k, A^+, \Delta \vdash B}{A^k, A^+ \vdash \bullet\Gamma \setminus B \,/\, \bullet\Delta}}{A^+ \vdash \bullet\Gamma \setminus B \,/\, \bullet\Delta}$$

□

3 Undecidability of $\mathbf{L}_\ell^+$

Theorem 3.1 *The derivability problem in* $\mathbf{L}_\ell^+$ *is undecidable.*

The proof of Theorem 3.1 combines ideas of the undecidability proof for **ACT** from [18] and the Π_1^0-hardness proof for $\mathbf{L}_\omega^+$ from [15].

First we encode several kinds of Turing machine behaviour via totality-like properties of context-free grammars. Then we follow the idea of Buszkowski [3] and embed these grammars into the Lambek environment. However, instead of the standard embedding (which goes back to Gaifman [2]) we use Safiullin's [25] construction of Lambek grammars with unique type assignment.

We consider only deterministic Turing machines, and suppose that each Turing machine has a designated *cycling state* q_c in which it gets stuck. (Rules for q_c are as follows: $\langle q_c, a\rangle \to \langle q_c, a, N\rangle$ for any letter a of the inner alphabet; N stands for "no move.") The cycling state can be added to any Turing machine, even if it is not necessary: in this case it can be just made unreachable.

Following the standard way (see [13, Lect. 35]), we encode a configuration of our Turing machine as $b_1 \ldots b_{i-1} q b_i b_{i+1} \ldots b_n$, if the machine is in state q, observing the i-th letter of the word $b_1 \ldots b_n$ in its memory. Protocols are sequences of configurations separated by a special character #, also beginning and ending with #.[2] Let Σ be the alphabet for protocols (including the inner alphabet, the set of states, and #). A protocol is *correct,* if each configuration, starting from the second one, is the successor of the previous configuration. A protocol is a *halting* one, if the last configuration has no successor (the machine cannot proceed one more step forward).

Given a Turing machine $\mathfrak{M}$ and an input word x, one can effectively construct (see [13, Lect. 35], for example) a context-free grammar $\mathcal{G}_{\mathfrak{M},x}$ which

---

[2] In some other presentations of this construction in textbooks, the code of every second configuration is inverted. For our purposes, this is irrelevant.

generates all words over Σ, except the correct halting protocol of $\mathfrak{M}$ on x (if it exists). This construction gives a reduction of the *non-halting* problem for Turing machines to the totality problem for context-free grammars, and thus establishes Π_1^0-hardness of the latter.

We suppose that $\mathcal{G}_{\mathfrak{M},x}$ is in Greibach normal form [8] and extend it by extra rules for capturing the easy case of non-halting—getting stuck in q_c:

$$
\begin{array}{ll}
S \Rightarrow \#CU & \\
U \Rightarrow aU & \text{for any } a \in \Sigma \\
U \Rightarrow a & \text{for any } a \in \Sigma \\
C \Rightarrow aC & \text{for any } a \in \Sigma \\
C \Rightarrow q_c U & \\
C \Rightarrow q_c &
\end{array}
$$

In these rules, non-terminal U generates all non-empty words and C generates all words including q_c. Thus, the rule $S \Rightarrow \#CU$ captures the idea that any word including q_c could not be a correct halting protocol.

We also suppose that $\mathcal{G}_{\mathfrak{M},x}$ has a subgrammar starting with a non-terminal E which generates all words which are incorrect protocols and cannot be fixed by extending to the right. Due to greibachization, the leading # gets removed. For example, such a "bad" protocol could include a configuration which is followed by another configuration which is not its successor. For more details, see [18,17]. We express the idea that such a "bad" protocol cannot be fixed, by adding the following rules:

$$S \Rightarrow \#EU, \qquad S \Rightarrow aU \text{ for } a \neq \#.$$

(The second rule states that a good protocol should always start with #.)

We denote the extended grammar by $\mathcal{G}'_{\mathfrak{M},x}$.

Next, in order to use reasoning in the style of [15], we restrict ourselves to a two-letter alphabet $\{e, f\}$. Let $\Sigma = \{a_1, a_2, \ldots, a_N\}$ and define a homomorphism $h \colon \Sigma^+ \to \{e, f\}^+$ on letters as follows:

$$h(a_i) = ef^i = e\underbrace{f \ldots f}_{i \text{ times}}.$$

(Then h is uniquely propagated to words as a homomorphism.)

By $h(\mathcal{G}'_{\mathfrak{M},x})$ we denote the image of $\mathcal{G}'_{\mathfrak{M},x}$ under homomorphism h. In order to maintain it in Greibach normal form, for each old rule of the form $A \Rightarrow a_i BC$ we introduce a series of rules

$$A \Rightarrow eX_1, \quad X_1 \Rightarrow fX_2, \quad \ldots, \quad X_{i-1} \Rightarrow fX_i, \quad X_i \Rightarrow fBC,$$

where $X_1, \ldots, X_i$ are new non-terminal symbols (different for each rule of the original grammar). Translations for rules of the forms $A \Rightarrow a_i B$ and $A \Rightarrow a_i$ is similar.

Next, let us construct the grammar $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$. We extend $h(\mathcal{G}'_{\mathfrak{M},x})$ with rules generating words with subwords of the form ef^m, where $m > N = |\Sigma|$ (these words are not in the image of h):

$$\begin{array}{ll} S \Rightarrow eF_{\geq N}W & F \Rightarrow fF \\ S \Rightarrow eF_{\geq N} & F \Rightarrow f \\ S \Rightarrow eFS' & F_{\geq N} \Rightarrow fF_{\geq N-1} \\ S' \Rightarrow eFS' & F_{\geq N-1} \Rightarrow fF_{\geq N-2} \\ S' \Rightarrow eF_{\geq N}W & \ldots \\ S' \Rightarrow eF_{\geq N} & F_{\geq 3} \Rightarrow fF_{\geq 2} \\ W \Rightarrow eFW & F_{\geq 2} \Rightarrow fF \\ W \Rightarrow eF & \end{array}$$

Here S', W, F, and $F_{\geq m}$ $(m = 2, \ldots, N)$ are new non-terminal symbols.

Finally, we replace U with W in the 'old' part of the grammar. This will not alter the language, since any word derived from W is either also derived from U, or includes a subword of the form ef^m with $m > N$, which is of course not an h-image of a correct protocol.

This finishes the construction of $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$. From this construction, one can easily see the following property:

Lemma 3.2 *The grammar $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$ generates all words of the language generated by the regular expression $(ef^+)^+$ if and only if $\mathfrak{M}$ does not halt on x. If $\mathfrak{M}$ does halt on x, then $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$ generates all words of this language, except $h(\pi)$, where π is the halting protocol of $\mathfrak{M}$ on x.*

The next step uses Safiullin's construction of Lambek grammar with unique type assignment. This result was published by Safiullin as a short note [25] without detailed proofs. A complete exposition is presented in the Appendix of [15]. We shall need Safiullin's result for grammars over a two-letter alphabet in the following form.

Theorem 3.3 (Safiullin) *Let $\widetilde{\mathcal{G}}$ be a context-free grammar over alphabet $\{e, f\}$ in Greibach normal form. Then there exist formulae E, F, and H_A for each non-terminal A, such that the following holds:*

(i) *a non-empty word w is generated by $\widetilde{\mathcal{G}}$ if and only if the sequent $\Gamma_w \vdash H_S$ is derivable in* **L***, where Γ_w is a sequence of formulae obtained from w by replacing e with E and f with F (e.g., for $w = effee$ we have $\Gamma_w = E, F, F, E, E$);*

(ii) *for each rule of $\widetilde{\mathcal{G}}$ we have the following sequents derivable in* **L***:*

Rule	**Sequent**
$A \Rightarrow eBC$	$E, H_B, H_C \vdash H_A$
$A \Rightarrow fBC$	$F, H_B, H_C \vdash H_A$
$A \Rightarrow eB$	$E, H_B \vdash H_A$
$A \Rightarrow fB$	$F, H_B \vdash H_A$
$A \Rightarrow e$	$E \vdash H_A$
$A \Rightarrow f$	$F \vdash H_A$

In this theorem, the first statement is essentially the result on transforming a context-free grammar into a Lambek grammar with unique type assignment (E is the type for e, F for f, and H_S is the goal type). The second statement is actually a technical lemma (induction step) for proving the "only if" direction in the first statement. However, we shall need the second statement explicitly. Further details of Safiullin's construction are irrelevant for us, we use it as a black box.

Using induction and statement (ii), one can easily prove a strengthening of the "only if" part of statement (i). Namely,

(iii) *if a word α in the alphabet of both terminal and non-terminal symbols is derivable in $\mathcal{G}$ from a non-terminal A (notation: $A \Rightarrow^* \alpha$), then the sequent $\Gamma_\alpha \vdash H_A$ is derivable in* **L**.

Here Γ_α is obtained from α by replacing e with E, f with F, and each non-terminal B by the corresponding H_B.

Consider the sequent

$$(E \cdot F^+)^+ \vdash H_S,$$

where E, F and H_S are obtained from $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$ by the construction from Theorem 3.3. Now we proceed as in [18], proving one direction for $\mathbf{L}^+_\omega$ and non-halting of $\mathfrak{M}$ on x and the other direction for $\mathbf{L}^+_\ell$ and $\mathfrak{M}$ getting stuck in q_c while running on x.

Lemma 3.4 *The sequent $(E \cdot F^+)^+ \vdash H_S$ is derivable in $\mathbf{L}^+_\omega$ if and only if $\mathfrak{M}$ does not halt on x.*

Proof. The ω-rule is invertible, by cut with $A, \ldots, A \to A^+$. Thus, $(E{\cdot}F^+)^+ \vdash H_S$ is derivable in $\mathbf{L}^+_\omega$ if and only if so is $\Gamma_w \vdash H_S$ for any word w from the language of the regular expression $(ef^+)^+$. This sequent does not include the iteration modality, so its derivability in $\mathbf{L}^+_\omega$ is equivalent to its derivability in **L**. By Theorem 3.3, derivability of all these sequents is equivalent to the fact that $\mathcal{G}_{\mathfrak{M},x}$ generates all words satisfying the regular expression $(ef^+)^+$. By Lemma 3.2, this is equivalent to non-halting of $\mathfrak{M}$ on x. $\square$

Lemma 3.5 *If $\mathfrak{M}$ gets stuck in q_c when running on x, then $(E \cdot F^+)^+ \vdash H_S$ is derivable in $\mathbf{L}^+_\ell$.*

Proof. Here the 'long rule' finally comes into play. Let n be the length (in symbols, not in steps) of the protocol of $\mathfrak{M}$ running on x until it reaches q_c.

Using the 'long rule,' we derive $(E \cdot F^+)^+ \vdash H_S$ from the following sequents:

$$\begin{array}{ll} (E, F^+)^k \vdash H_S & \qquad k \leq n \\ (E, F^+)^n, (E \cdot F^+)^+ \vdash H_S & \end{array}$$

The first series of sequents, $(E, F^+)^k \vdash H_S$, is also derived by exhaustive application of the 'long rule,' in its form with sequential contexts (Lemma 2.2), up to $N = |\Sigma|$. The sequents we now have to derive are of the form $\Pi_1, \ldots, \Pi_k \vdash H_S$, where $k \leq n$ and each Π_i is either E, F^s, where $s \leq N$, or E, F^N, F^+.

If all Π_i's are of the form E, F^s, then $\Pi_1, \ldots, \Pi_k \vdash H_S$ does not include $^+$ and is derivable in $\mathbf{L}$ by applying Lemma 3.4 and inverting the ω-rule.

The more interesting case is when our sequent includes E, F^N, F^+. Let Π_{i_0} be the first Π_i of this form. First we notice that $F^+ \vdash H_F$ is derivable in $\mathbf{L}_\ell^+$:

$$\frac{F \vdash H_F \quad F, H_F \vdash H_F}{F^+ \vdash H_F}$$

Here the premises are derivable by Theorem 3.3(ii), due to the rules $F \Rightarrow f$ and $F \Rightarrow fF$. Thus, by cut, we can replace E, F^N, F^+ by E, F^N, H_F.

Moreover, since $F_{\geq N} \Rightarrow^* f^N F$, we can apply cut with $F^N, H_F \vdash H_{F_{\geq N}}$ and replace Π_{i_0} with $E, H_{F_{\geq N}}$. For $i \neq i_0$ we similarly replace Π_i with E, H_F, using either $F \Rightarrow^* f^N F$ or $F \Rightarrow^* f^k$. Thus, the whole sequent is now of the form

$$E, H_F, \ldots, E, H_F, E, H_{F_{\geq N}}, E, H_F, \ldots, E, H_F \vdash H_S,$$

which is derivable due to the following derivation in $\widetilde{G}_{\mathfrak{M},x}$:

$$S \Rightarrow eFS' \Rightarrow^* eF \ldots eFS' \Rightarrow eF \ldots eFeF_{\geq N}W \Rightarrow^* eF \ldots eFeF_{\geq N}eF \ldots eF$$

for $i_0 \neq 1, k$ and similarly (but using different rules of $\widetilde{G}_{\mathfrak{M},x}$) for $i_0 = 1$ and $i_0 = k$.

Finally, the second sequent, $(E, F^+)^n, (E \cdot F^+)^+ \vdash H_S$, is derived in a similar fashion. We applying the 'long rule' with N exhaustively to the instances of F^+ in $(E, F^+)^n$ and consider two cases for premises. If at least one of the instances of E, F^+ becomes E, F^N, F^+, then we again reduce to

$$E, H_F, \ldots, E, H_F, E, H_{F_{\geq N}}, E, H_F, \ldots, E, H_F, (E \cdot F^+)^+ \vdash H_S$$

Next, we notice derivability of $(E \cdot F^+)^+ \vdash H_W$:

$$\dfrac{\dfrac{\dfrac{F^+ \vdash H_F \quad E, H_F \vdash H_W}{E, F^+ \vdash H_W}\,(cut)}{E \cdot F^+ \vdash H_W} \qquad \dfrac{\dfrac{F^+ \vdash H_F \quad E, H_F, H_W \vdash H_W}{E, F^+, H_W \vdash H_W}\,(cut)}{E \cdot F^+, H_W \vdash H_W}}{(E \cdot F^+)^+ \vdash H_W}$$

The premises are derivable by Theorem 3.3(ii) due to $W \Rightarrow eF$ and $W \Rightarrow eFW$; $F^+ \vdash H_F$ was established above.

Thus, we reduce to

$$E, H_F, \dots, E, H_F, E, H_{F_{\geq N}}, E, H_F, \dots, E, H_F, H_W \vdash H_S,$$

which is derivable by statement (iii) below Theorem 3.3 due to

$$S \Rightarrow^* eF \dots eFeF_{\geq N}eF \dots eFW.$$

The second, more interesting case is when each instance of F^+ becomes F^{s_i} for some i:

$$E, F^{s_1}, \dots, E, F^{s_n}, (E \cdot F^+)^+ \vdash H_S.$$

Recalling $(E \cdot F^+)^+ \vdash H_W$ (see above), we reduce to

$$E, F^{s_1}, \dots, E, F^{s_n}, H_W \vdash H_S.$$

Next, this sequent can be rewritten in the form

$$\Gamma_{h(w)}, H_W \vdash H_S,$$

where $w = a_{s_1} \dots a_{s_n}$. Since n is the number of letters in the protocol sufficient for $\mathfrak{M}$ on x to reach the cycling state q_c, the word w either includes q_c, or is an incorrect ("bad") protocol, or does not start with $\#$.

In the first case, we have $w = \#w'$ and $C \Rightarrow^* w'$ in $\mathcal{G}'_{\mathfrak{M},x}$. Thus, we get $C \Rightarrow^* h(w')$ in $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$, and by statement (iii) derive

$$\Gamma_{h(w')} \vdash H_C.$$

Gathering things together and cutting, we get

$$\Gamma_{h(\#)}, H_C, H_W \vdash H_S,$$

which is derivable via statement (iii) and $S \Rightarrow^* h(\#)CW$.

The case where w is a "bad" protocol is similar, using $S \Rightarrow^* h(\#)EW$. Finally, if w starts with $a_{s_1} \neq \#$ we have

$$\Gamma_{h(a_{s_1})}, \Gamma_{h(w')}, H_W \vdash H_S,$$

which is derivable by cut from $\Gamma_{h(w')}, H_W \vdash H_W$ and $\Gamma_{h(a_{s_1})}, H_W \vdash H_S$. These are derivable by statement (iii), using $W \Rightarrow^* h(w')W$ and $S \Rightarrow^* a_{s_1}W$.

This finishes the proof of our key lemma. □

Now we proceed exactly as in [18]. Let

$\mathcal{H} = \{\langle \mathfrak{M}, x \rangle \mid \mathfrak{M} \text{ halts on } x\}$

$\overline{\mathcal{H}} = \{\langle \mathfrak{M}, x \rangle \mid \mathfrak{M} \text{ does not halt on } x\}$

$\mathcal{C} = \{\langle \mathfrak{M}, x \rangle \mid \mathfrak{M} \text{ gets stuck in } q_c \text{ while running on } x\}$

$\mathcal{K} = \{\langle \mathfrak{M}, x \rangle \mid (E \cdot F^+)^+ \vdash H_S$, where E, F, and H_S come from $\widetilde{\mathcal{G}}_{\mathfrak{M},x}$, is derivable in $\mathbf{L}^+_\ell\}$

By Lemma 3.4 $\mathcal{K} \subseteq \overline{\mathcal{H}}$ (recall that $\mathbf{L}^+_\ell$ is a subsystem of $\mathbf{L}^+_\omega$ by Lemma 2.1); by Lemma 3.2 $\mathcal{C} \subseteq \mathcal{K}$. Since $\mathcal{C}$ and $\mathcal{H}$ are recursively inseparable, $\mathcal{K}$ is undecidable, thus so is the derivability problem for $\mathbf{L}^+_\ell$. Theorem 3.1 proved.

Following the reasoning with effective inseparability of $\mathcal{C}$ and $\mathcal{H}$, presented in [17], we can show Σ_1-completeness of $\mathbf{L}^+_\ell$ and, moreover, any recursively enumerable logic in the range between $\mathbf{L}^+_\ell$ and $\mathbf{L}^+_\omega$. This is performed exactly as for action logic with meet and join.

4 Non-derivability of the 'long rule' in $\mathbf{L}^+$

As one can see from the previous section, the 'long rule' is a crucial component of the undecidability proof. If we could derive this rule in $\mathbf{L}^+$, as it can be done for **ACT** [18], we would get undecidability for $\mathbf{L}^+$.

Unfortunately, as we show in this section, this is not the case: the 'long rule' is not derivable in $\mathbf{L}^+$.

Before proceeding further, let us notice a subtle difference between *derivability* and a weaker notion of *admissibility* of a new rule in a calculus. A rule is called *derivable,* if there exists a derivation of the conclusion of this rule with its premises as hypotheses. This derivation is allowed to use cut. On the other hand, a rule (rule scheme) is *admissible,* if, for any substitution of concrete formulae for meta-variables A, B, C, ..., derivability of its premises implies derivability of its conclusion.

Clearly, any derivable rule is admissible. The converse implication, however, does not hold. For example, the rule $\frac{A \vdash A \cdot A}{B \vdash C}$ is admissible, but not derivable in $\mathbf{L}^+$. The reason is that $A \vdash A \cdot A$ cannot be derivable for any A. This can be proved by interpretation on language models, see [4]. Indeed, consider cofinite languages over an alphabet. Product (pairwise concatenation) and divisions (defined according to the rules of the Lambek calculus) of cofinite languages yield again cofinite languages. Thus, if we interpret all variables as cofinite languages, then the interpretation of A will be also cofinite, thus, non-empty. But then the shortest word of A does not belong to $A \cdot A$ (the empty word is not allowed due to Lambek's non-emptiness condition). Thus, the rule in question is admissible *ex falso.* On the other hand, it is clearly non-derivable, since $B \vdash C$ is absolutely foreign to $A \vdash A \cdot A$. Unfortunately, the author is not aware of more interesting examples of admissible non-derivable rules—that is, in which there exist derivable instances of the premises.

We claim only non-derivability of the 'long rule.' Its admissibility in $\mathbf{L}^+$ is left as an open question.

Theorem 4.1 *The special case of the 'long rule' for $k = 1$,*[3]

$$\frac{A \vdash B \quad A, A^+ \vdash B}{A^+ \vdash B}$$

is not derivable in $\mathbf{L}^+$.

[3] We could call it 'short rule.'

Proof. We prove non-derivability of this rule by presenting an algebraic counter-model. The appropriate class of algebraic models for $\mathbf{L}^+$ is formed by *residuated semigroups with iteration (RSGI),* defined as follows.

An RSGI is a partially ordered algebraic structure $(S, \preceq, \cdot, \backslash, /, ^+)$, such that:

(i) $\preceq$ is a partial order on S;

(ii) $(S, \cdot)$ is a semigroup;

(iii) $\backslash$ and $/$ are residuals of $\cdot$ w.r.t. $\preceq$:

$$x \setminus y = \max_{\preceq}\{z \mid x \cdot z \preceq y\}, \qquad y \,/\, x = \max_{\preceq}\{z \mid z \cdot x \preceq y\};$$

(iv) for each $x \in S$, $x^+ = \min_{\preceq}\{y \mid x \preceq y \text{ and } x \cdot y \preceq y\}$.

An interpretation function v is just a mapping of variables to elements of S; then it is propagated to formulae. A sequent $A_1, \ldots, A_n \vdash B$ is true under v, if $v(A_1) \cdot \ldots \cdot v(A_n) \preceq v(B)$.

Clearly, the following strong form of soundness holds for $\mathbf{L}^+$ w.r.t. RSGI: if a sequent is derivable from a set of hypotheses, and under a given v all these hypotheses are true, then so is the goal sequent. (The proof of soundness involves using monotonicity of $\cdot$ w.r.t. $\preceq$, which is due to Lambek [19]. Completeness also holds, by a Lindenbaum – Tarski argument, but we shall not need it.)

We shall present an RSGI and its two elements $a, b \in S$, such that $a \preceq b$, $a \cdot a^+ \preceq b$, but $a^+ \not\preceq b$. This will do the job, since if the rule in question were derivable, then, in particular, one could derive $p^+ \vdash q$ from $p \vdash q$ and $p, p^+ \vdash q$ (p and q are variables). This conflicts soundness, by taking $v(p) = a$, $v(q) = b$.

Let us start with a standard example of RSGI, which reflects Lambek's original linguistic motivations,—the algebra of formal languages. For us, it is sufficient to consider languages without the empty word over a one-letter alphabet $\Sigma = \{s\}$. Such languages are in one-to-one correspondence with sets of non-zero natural numbers (the word $\underbrace{s \ldots s}_{n}$ is represented by n). We denote the set of all such sets by $\mathcal{P}(\mathbb{N}_+)$. The elements $\varnothing$ and $\mathbb{N}_+$ of $\mathcal{P}(\mathbb{N}_+)$ (the empty and the total language) will play special rôles in our construction. The set of all other languages is $\mathcal{P}_0(\mathbb{N}_+) = \mathcal{P}(\mathbb{N}_+) - \{\varnothing, \mathbb{N}_+\}$.

Our RSGI will be $\mathcal{P}(\mathbb{N}_+)$ extended by two extra elements:

$$S = \mathcal{P}(\mathbb{N}_+) \cup \{\xi, \eta\} = \mathcal{P}_0(\mathbb{N}_+) \cup \{\varnothing, \mathbb{N}_+, \xi, \eta\}.$$

The partial order $\preceq$ on S is defined as follows:

- on $\mathcal{P}(\mathbb{N}_+)$, the partial order is the subset relation;
- for any $x \in \mathcal{P}_0(\mathbb{N}_+) \cup \{\varnothing\}$, we have $x \prec \xi$; ξ and $\mathbb{N}_+$ are incomparable;
- η is the maximal element: for any $x \in \mathcal{P}(\mathbb{N}_+) \cup \{\xi\}$, we have $x \prec \eta$.

The product operation on S is commutative and defined as follows:

- for $x, y \in \mathcal{P}(\mathbb{N}_+)$, product is defined as pairwise addition:
$$x \cdot y = \{n + m \mid n \in x, m \in y\};$$
- $\varnothing \cdot \xi = \varnothing \cdot \eta = \varnothing$;
- $\xi \cdot x = \eta$ for any $x \neq \varnothing$;
- $\eta \cdot x = \eta$ for any $x \neq \varnothing$.

Associativity of product, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$, is proved as follows. The interesting case is when at least one of x, y, z is ξ or η: otherwise we refer to associativity of formal language multiplication. If one of x, y, z is ξ or η and another one is $\varnothing$, then $(x \cdot y) \cdot z = x \cdot (y \cdot z) = \varnothing$; otherwise $(x \cdot y) \cdot z = x \cdot (y \cdot z) = \eta$.

Now let us define residuals, that is, prove existence of the corresponding maxima. Since our semigroup is commutative, we shall always have $x \setminus y = y \,/\, x$, so it is sufficient to prove existence of $x \setminus y$.

- For $x, y \in \mathcal{P}(\mathbb{N}_+)$, if $x \neq \varnothing$, we have
$$x \setminus y = \{n \in \mathbb{N}_+ \mid (\forall m \in x)\, n + m \in y\},$$
as in the algebra of formal languages. Indeed, inside $\mathcal{P}_0(\mathbb{N}_+)$ this is the maximal z such that $x \cdot z \preceq y$. As for ξ and η, we have (since $x \neq \varnothing$) $x \cdot \xi = x \cdot \eta = \eta \not\preceq y$.
- For any y we have $\varnothing \setminus y = \eta$. Indeed, $\varnothing \cdot z \preceq y$ holds for any z (since $\varnothing \cdot z = \varnothing$), so we just take the maximum of the whole S.
- For any x, we have $x \setminus \eta = \eta$. Indeed, $x \cdot z \preceq \eta$ holds for any z (since η is the maximum).
- For any $x \in \mathcal{P}_0(\mathbb{N}_+)$, we have $x \setminus \xi = \mathbb{N}_+$. Indeed, $x \cdot \mathbb{N}_+$ belongs to $\mathcal{P}_0(\mathbb{N}_+)$ and therefore is below ξ in the sense of $\preceq$. On the other hand, the only two elements, which are not below $\mathbb{N}_+$, are ξ and η. For them we have $x \cdot \xi = x \cdot \eta = \eta \not\preceq \xi$.
- We also have $\mathbb{N}_+ \setminus \xi = \mathbb{N}_+$. This happens because of the lack of the empty word (zero in $\mathbb{N}_+$): $\mathbb{N}_+ \cdot \mathbb{N}_+ = \{n \mid n \geq 2\} \preceq \xi$. For ξ and η we have, again, $\mathbb{N}_+ \cdot \xi = \mathbb{N}_+ \cdot \eta = \eta \not\preceq \xi$.
- For any $y \neq \eta$, we have $\eta \setminus y = \varnothing$, since $\eta \cdot \varnothing = \varnothing \prec y$ and $\eta \cdot z = \eta \not\preceq y$ for any $z \neq \varnothing$. (As shown above, $\eta \setminus \eta = \eta$.)
- Similarly, $\xi \setminus \eta = \eta$ (shown above), and for any $y \neq \eta$ we have $\xi \setminus y = \varnothing$ (in particular, $\xi \setminus \xi = \varnothing$). Indeed, $\xi \cdot \varnothing = \varnothing \prec y$ and $\xi \cdot z = \eta \not\preceq y$ for any $z \neq \varnothing$.

Finally, let us define iteration, that is, prove that for any x there exists $x^+ = \min_{\preceq}\{y \mid x \preceq y \text{ and } x \cdot y \preceq y\}$.

- For $x \in \mathcal{P}(\mathbb{N}_+)$, its iteration x^+ is defined traditionally: $x^+ = \{n_1 + \ldots + n_k \mid k \geq 1, n_i \in x\}$. If $x^+ \neq \mathbb{N}_+$, then it is indeed the necessary minimum: it is the minimum in $\mathbb{N}_+$, and two other candidates, ξ and η, are above x^+. The case of $x^+ = \mathbb{N}_+$ is a bit more interesting. Again, $\eta \succ x^+$, so it is not a rival; but ξ is incomparable with $x^+ = \mathbb{N}_+$. Fortunately, ξ fails to satisfy the second condition on y to be considered as a candidate for x^+. If $x \neq \varnothing$, then $x \cdot \xi = \eta \not\preceq \xi$ (if $x = \varnothing$, then $x^+ = \varnothing \neq \mathbb{N}_+$).

- $\xi^+ = \eta$. Indeed, ξ^+ should be ξ or η, and ξ does not suffice, since $\xi \cdot \xi = \eta \not\preceq \xi$. For η, everything is all right: $\xi \preceq \eta$ and $\xi \cdot \eta = \eta \preceq \eta$.
- $\eta^+ = \eta$. Indeed, $\eta \preceq \eta$ and $\eta \cdot \eta = \eta \preceq \eta$. Smaller y's are out of the game, since $\eta \not\preceq y$.

Having defined our specific RSGI S, now let $a = \{1\}$ and $b = \xi$. We have: $a \preceq b$; $a^+ = \mathbb{N}_+$, so $a \cdot a^+ = \{n \mid n \geq 2\} \preceq b$; but $a^+ \not\preceq b$ ($\mathbb{N}_+$ and ξ are incomparable). This finishes our proof. ◻

An important observation on our RSGI S is that its partial order does not form a lattice structure. Namely, $\mathbb{N}_+$ and ξ have no meet: any element of $\mathcal{P}_0(\mathbb{N}_+)$ is below both, and among them there is no maximal one. Dually, $a = \{1\}$ and $a \cdot a^+ = \{n \mid n \geq 2\}$ have no join: ξ and $\mathbb{N}_+$ are above both and are incomparable. This is by design: once we have a lattice, or at least we have a join of a and $a \cdot a^+$, we can apply the derivation of the 'long rule' in $\mathbf{ACT}^+$.

We also notice that in S iteration a^+ is defined as a fixed point, not as a supremum (that is, S is not *-continuous). Indeed, for $a = \{1\}$ its iteration $a^+ = \mathbb{N}_+$ is the smallest y such that $a \preceq y$ and $a \cdot y \preceq y$. However, a^+ is not $\sup_{\preceq}\{a^n \mid n \geq 1\}$. Indeed, there are two incomparable upper bounds for $a^n = \{n\}$, namely, $\mathbb{N}_+$ and ξ. The latter is a 'fake' iteration, since it is not a fixpoint: $a \cdot \xi = \eta \not\preceq \xi$. The non-*-continuity of S is also for a good reason: otherwise, S would model $\mathbf{L}^+_\omega$, and in this system the 'long rule' is derivable (Lemma 2.1).

5 Concluding Remarks

We have proved undecidability (and Σ_1-completeness) of the Lambek calculus with an inductively axiomatized positive iteration modality, extended with the so-called 'long rule' of the form

$$\frac{A \vdash B \quad A, A \vdash B \quad A^k \vdash B \quad A^k, A^+ \vdash B}{A^+ \vdash B}$$

This result refines the undecidability result for action logic [18], since now we obtain undecidability for a system without additive connectives, meet and join ($\wedge$ and $\vee$).

Another distinctive feature of this paper is the Lambek's non-emptiness restriction imposed on the calculus. We conjecture that the same results hold without this restriction. However, this is left as an open question for further research, since some technicalities, namely, Safiullin's Theorem 3.3 and the counter-model construction in Theorem 4.1, in their current state, depend on Lambek's restriction.

In action logic with meet and join, the 'long rule' is derivable; for the multiplicative-only system $\mathbf{L}^+$ studied in this paper, this is not the case (Theorem 4.1). The question of whether the 'long rule' is admissible in $\mathbf{L}^+$ is still open. If the answer happens to be positive, we shall immediately get undecidability of $\mathbf{L}^+$ (since in this case $\mathbf{L}^+$ and $\mathbf{L}^+_\ell$ derive the same set of sequents). If the answer is negative, then $\mathbf{L}^+_\ell$ is strictly stronger than $\mathbf{L}^+$, and complexity of the latter remains a separate open problem.

Moreover, non-derivability and potential non-admissibility of the 'long rule' brings some light upon the old question on constructing a cut-free calculus for action logic with inductive axiomatizations for iteration. As noticed in the Preliminaries, for systems with inductive-style rules for iteration no cut-free sequential calculi are known. The issues with the 'long rule' discussed in this paper are actually conservativity issues. Since the 'long rule' is not derivable in $\mathbf{L}^+$, this calculus is not a *strongly conservative* fragment of $\mathbf{ACT}^+$. Namely, consider three sequents $p \vdash q$, $\quad p, p^+ \vdash q$, and $p^+ \vdash q$ (premises and conclusion of the 'long rule'). These sequents are formulated in the language of $\mathbf{L}^+$, without $\vee$ and $\wedge$. Actually, they use only one connective, $^+$. However, one can derive the third one from the first and the second ones only in $\mathbf{ACT}^+$ (via a detour through $\vee$), not in $\mathbf{L}^+$. If the 'long rule' happens to be non-admissible, ordinary conservativity would also fail. In this case, in particular, it would be an open question which sequents without $\vee$ and $\wedge$ are derivable in $\mathbf{ACT}^+$—are these sequents exactly theorems of $\mathbf{L}^+_\ell$, or do they form a larger set?

However, if $\mathbf{ACT}^+$ were axiomatized by a sequent calculus (even with a non-standard notion of proof, like a circular one), it would enjoy conservativity. Thus, in view of the issues with the 'long rule,' it looks reasonable to extend our approaches for axiomatizing $\mathbf{ACT}^+$ and search for hypersequential formalisms where $\vee$ or $\wedge$ is incorporated into the meta-syntax (cf. Kozak's system for distributive full Lambek calculus [11]). Notice that the sequents appearing in the 'long rule' do not include division operations (only product and iteration). Thus, the same conservativity issues could potentially appear in the logics of Kleene algebras and lattices without residuals.

These considerations are quite coherent with the complete cut-free circular proof system for Kleene algebras presented by Das and Pous [6]. Their calculus is hypersequential, introducing join ($\vee$) on the meta-syntactic level to the right-hand sides of sequents. The counter-example for cut-free cyclic provability in a system with traditional sequents given by Das and Pous is $A \cdot A^* \vdash A^* \cdot A$, which is quite close to our 'short rule' in Theorem 4.1.

Acknowledgements The author is grateful to Lev Beklemishev, Anupam Das, Max Kanovich, Fedor Pakhomov, Andre Scedrov, Daniyar Shamkanov, and Stanislav Speranski for fruitful discussions. Being a Young Russian Mathematics award winner, the author thanks its jury and sponsors for this high honour. (The work on this particular paper was funded from another source, as mentioned on the first page.)

References

[1] Abrusci, V. M., *A comparison between lambek syntactic calculus and intuitionistic linear logic*, Zeitschrift für mathematische Logik und Grundlagen der Mathematik **36** (1990),

pp. 11–15.

[2] Bar-Hillel, Y., C. Gaifman and E. Shamir, *On the categorial and phrase-structure grammars*, Bulletin of the Research Council of Israel **9F** (1960), pp. 1–16.

[3] Buszkowski, W., *On action logic: equational theories of action algebras*, Journal of Logic and Computation **17** (2007), pp. 199–217.

[4] Buszkowski, W., *Lambek calculus and substructural logics*, Linguistic Analysis **36** (2010), pp. 15–48.

[5] Buszkowski, W. and E. Palka, *Infinitary action logic: complexity, models and grammars*, Studia Logica **89** (2008), pp. 1–18.

[6] Das, A. and D. Pous, *A cut-free cyclic proof system for Kleene algebra*, in: R. Schmidt and C. Nalon, editors, *Automated Reasoning with Analytic Tableaux and Related Methods. TABLEAUX 2017*, Lecture Notes in Computer Science **10501** (2017), pp. 261–277.

[7] Girard, J.-Y., *Linear logic*, Theoretical Computer Science **50** (1987), pp. 1–102.

[8] Greibach, S. A., *A new normal-form theorem for context-free phrase structure grammars*, Journal of the ACM **12** (1965), pp. 42–52.

[9] Jipsen, P., *From residuated semirings to Kleene algebras*, Studia Logica **76** (2004), pp. 291–303.

[10] Kleene, S. C., *Representation of events in nerve nets and finite automata*, in: C. E. Shannon and J. McCarthy, editors, *Automata Studies*, Princeton University Press, 1956 pp. 3–41.

[11] Kozak, M., *Distributive full Lambek calculus has the finite model property*, Studia Logica **91** (2009), p. 201–216.

[12] Kozen, D., *On action algebras*, in: J. van Eijck and A. Visser, editors, *Logic and Information Flow*, MIT Press, 1994 pp. 78–88.

[13] Kozen, D., "Automata and Complexity," Springer-Verlag, New York, 1997.

[14] Krull, W., *Axiomatische Begründung der algemeinen Idealtheorie*, Sitzungsberichte der physikalischmedizinischen Societät zu Erlangen **56** (1924), pp. 47–63.

[15] Kuznetsov, S., *The Lambek calculus with iteration: two variants*, in: J. Kennedy and R. de Queiroz, editors, *Logic, Language, Information, and Computation. WoLLIC 2017*, Lecture Notes in Computer Science **10388**, 2017, pp. 182–198.

[16] Kuznetsov, S., **-continuity vs. induction: divide and conquer*, in: G. Bezhanishvili, G. D'Agostino, G. Metcalfe and T. Studer, editors, *Proceedings of AiML '18*, Advances in Modal Logic **12** (2018), pp. 493–510.

[17] Kuznetsov, S., *Action logic is undecidable*, arXiv preprint 1912.11273 (2019).

[18] Kuznetsov, S., *The logic of action lattices is undecidable*, in: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (2019), pp. 1–9.

[19] Lambek, J., *The mathematics of sentence structure*, American Mathematical Monthly **65** (1958), pp. 154–170.

[20] Moot, R. and C. Retoré, "The logic of categorial grammars: a deductive account of natural language syntax and semantics," Lecture Notes in Computer Science **6850**, Springer, 2012.

[21] Palka, E., *An infinitary sequent system for the equational theory of *-continuous action lattices*, Fundamenta Informaticae **78** (2007), pp. 295–309.

[22] Pentus, M., *Residuated monoids with Kleene star* (2010), unpublished manuscript.

[23] Pratt, V., *Action logic and pure induction*, in: J. van Eijck, editor, *JELIA 1990: Logics in AI*, Lecture Notes in Artificial Intelligence **478** (1991), pp. 97–120.

[24] Restall, G., "An introduction to substructural logics," Routledge, 2000.

[25] Safiullin, A. N., *Derivability of admissible rules with simple premises in the Lambek calculus*, Moscow University Mathematics Bulletin **62** (2007), pp. 168–171.