Math. Struct. in Comp. Science: page 1 of 33. © Cambridge University Press 2018
doi:10.1017/S0960129518000117

Subexponentials in non-commutative linear logic

MAX KANOVICH[†], STEPAN KUZNETSOV[‡], VIVEK NIGAM[§,¶] and ANDRE SCEDROV[‖,†]

[†]*National Research University Higher School of Economics, Moscow, Russia*
Email: m.kanovich@qmul.ac.uk
[‡]*Steklov Mathematical Institute of RAS, Moscow, Russia*
Email: skuzn@inbox.ru
[§]*Federal University of Paraíba, João Pessoa, Brazil*
Email: vivek.nigam@gmail.com
[¶]*Fortiss GmbH, Munich, Germany*
[‖]*University of Pennsylvania, Philadelphia, U.S.A.*
E-mail: scedrov@math.upenn.edu

Received 5 July 2017; revised 20 March 2018

Linear logical frameworks with subexponentials have been used for the specification of, among other systems, proof systems, concurrent programming languages and linear authorisation logics. In these frameworks, subexponentials can be configured to allow or not for the application of the contraction and weakening rules while the exchange rule can always be applied. This means that formulae in such frameworks can only be organised as sets and multisets of formulae not being possible to organise formulae as lists of formulae. This paper investigates the proof theory of linear logic proof systems in the non-commutative variant. These systems can disallow the application of exchange rule on some subexponentials. We investigate conditions for when cut elimination is admissible in the presence of non-commutative subexponentials, investigating the interaction of the exchange rule with the local and non-local contraction rules. We also obtain some new undecidability and decidability results on non-commutative linear logic with subexponentials.

To Dale Miller's Festschrift and his contributions to Logic in Computer Science.
Dale's work has been an inspiration to us. He is a great researcher, colleague, advisor, and friend.

1. Introduction

Logic and proof theory have played an important role in computer science. The introduction of linear logic by Girard (1987) is an example of how the beauty of logic can be applied to the principles of computer science. More than 20 years ago, Hodas and Miller (1991, 1994) proposed the intuitionistic linear logical framework, Lolli, which distinguishes between two kinds of formulae: *linear*, that cannot be contracted and weakened, and *unbounded*, that can be contracted and weakened.[†] In contrast to existing intuitionistic/classical logical frameworks, Lolli allowed to express stateful computations using logical connectives. Some years later, Miller (1994, 1996) proposed the classical

[†] The authors received the LICS Test of Time Award for this work.

linear logical framework Forum demonstrating that linear logic can be used among other things to design concurrent systems.[‡]

It has been known, however, since Girard's original linear logic paper (Girard 1987), that the linear logic exponentials $!, ?$ are not canonical. Indeed, proof systems with non-equivalent exponentials (Danos et al. 1993) can be formulated. Nigam and Miller (2009) called them subexponentials and proposed a more expressive linear logical framework called SELL that allows for the specification of any number of non-equivalent subexponentials $!^s, ?^s$. Each subexponential can be specified to behave as linear or as unbounded. This is reflected in the syntax. SELL sequents associate a different context to each subexponential. Thus, formulae may be organised into a number of sets of unbounded formulae and a number of multisets of linear formulae. Nigam and Miller show that SELL is more expressive than Forum, being capable of expressing algorithmic specifications in logic. In the recent years, it has been shown that SELL can also be used to specify linear authorisation logics (Nigam 2012, 2014), concurrent constraint programming languages (Nigam et al. 2013; Olarte et al. 2015) and proof systems (Nigam et al. 2016).

While these logical frameworks have been successfully used for a number of applications, they do not allow sequents to be organised as lists of formulae. This is because all the frameworks above assume that the exchange rule can be applied to any formula. This paper investigates the proof theory of subexponentials in non-commutative linear logic. Our contribution is as follows:

1. We construct general non-commutative linear logic proof systems with subexponentials and investigate conditions for when these systems enjoy cut elimination and when they do not.
2. For systems, in which at least one subexponential obeys the contraction rule in its non-local form, we prove undecidability results.
3. For fragments, in which no subexponential obeys the contraction rule, we prove decidability and establish exact complexity bounds that coincide with the complexity estimations for the corresponding systems without subexponentials: NP for the purely multiplicative system, PSPACE for the system with additive connectives.

The rest of this paper is organised as follows. In Sections 2 and 5, we present two variants of non-commutative linear logic, resp., the multiplicative-additive Lambek calculus (SMALC_Σ) and cyclic linear logic (SCLL_Σ), enriched with subexponential modalities indexed by a subexponential signature Σ. Sections 3 and 4 sketch two possible applications of SMALC_Σ. In Section 6, we establish the cut elimination property for SCLL_Σ using the classical Gentzen's approach with a specific version of the mix rule. In Section 7, we show that SMALC_Σ can be conservatively embedded into SCLL_Σ. This yields, as a side-effect, cut elimination for SMALC_Σ. In Section 8, we explain why we prefer the *non-local* version of the contraction rule by showing that systems with only local contraction fail to enjoy the cut elimination property. Section 9 contains the proof of undecidability for systems with contraction; in Section 10, we prove decidability and

‡ For this work, Miller received yet another LICS Test of Time Award prize.

establish complexity bounds for systems without contraction. Section 11 is for conclusions and directions of future research.

2. The multiplicative-additive Lambek calculus with subexponentials

We start with the Lambek calculus allowing empty antecedents (Lambek 1961), considering it as a non-commutative form of intuitionistic propositional linear logic (Abrusci 1990). The original Lambek calculus includes only multiplicative connectives (multiplication and two implications, called divisions). It is quite natural, however, to equip the Lambek calculus also with additive connectives (conjunction and disjunction), as in linear logic (van Benthem 1991; Buszkowski 2010; Kanazawa 1992; Kuznetsov and Okhotin 2017). We will call this bigger system the *multiplicative-additive Lambek calculus* (MALC). Extended versions of the Lambek calculus have broad linguistical applications, serving as a basis for categorial grammars (Moortgat 1997; Moot and Retoré 2012; Morrill 2017b, 2011); see Section 4 for more details.

In this section, we extend the MALC with a family of subexponential connectives. First, we fix a *subexponential signature* of the form

$$\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle,$$

where $\mathcal{I} = \{s_1, \ldots, s_n\}$ is a set of subexponential labels with a preorder $\preceq$, and $\mathcal{W}$, $\mathcal{C}$ and $\mathcal{E}$ are subsets of $\mathcal{I}$. The sets $\mathcal{W}$, $\mathcal{C}$ and $\mathcal{E}$ are required to be upwardly closed with respect to $\preceq$. That is, if $s_1 \in \mathcal{W}$ and $s_1 \preceq s_2$, then $s_2 \in \mathcal{W}$ and ditto for the sets $\mathcal{E}$ and $\mathcal{C}$. Subexponentials marked with labels from $\mathcal{W}$ allow weakening, $\mathcal{C}$ allows contraction and $\mathcal{E}$ allows exchange (permutation). Since contraction (in the non-local form, see) and weakening yield exchange, here, we explicitly require $\mathcal{W} \cap \mathcal{C} \subseteq \mathcal{E}$.

Formulae are built from variables $p_1, p_2, p_3, \ldots$ and the unit constant $\mathbf{1}$ using five binary connectives: $\cdot$ (product, or multiplicative conjunction), $\backslash$ (left division), $/$ (right division), $\wedge$ (additive conjunction), and $\vee$ (additive disjunction) and a family of unary connectives, indexed by the subexponential signature Σ, denoted by $!^s$ for each $s \in \mathcal{I}$.

The axioms and rules of the MALC with subexponentials, denoted by SMALC_Σ, are as follows:

$$\frac{}{A \to A}\ (\mathrm{ax})$$

$$\frac{\Gamma_1, A, B, \Gamma_2 \to C}{\Gamma_1, A \cdot B, \Gamma_2 \to C}\ (\cdot \to) \qquad \frac{\Gamma_1 \to A \quad \Gamma_2 \to B}{\Gamma_1, \Gamma_2 \to A \cdot B}\ (\to \cdot)$$

$$\frac{\Pi \to A \quad \Gamma_1, B, \Gamma_2 \to C}{\Gamma_1, \Pi, A \backslash B, \Gamma_2 \to C}\ (\backslash \to) \qquad \frac{A, \Pi \to B}{\Pi \to A \backslash B}\ (\to \backslash)$$

$$\frac{\Pi \to A \quad \Gamma_1, B, \Gamma_2 \to C}{\Gamma_1, B / A, \Pi, \Gamma_2 \to C}\ (/ \to) \qquad \frac{\Pi, A \to B}{\Pi \to B / A}\ (\to /)$$

$$\frac{\Gamma_1, \Gamma_2 \to C}{\Gamma_1, \mathbf{1}, \Gamma_2 \to C}\ (\mathbf{1} \to) \qquad \frac{}{\to \mathbf{1}}\ (\to \mathbf{1})$$

$$\frac{\Gamma_1, A_1, \Gamma_2 \to C \quad \Gamma_1, A_2, \Gamma_2 \to C}{\Gamma_1, A_1 \vee A_2, \Gamma_2 \to C}\ (\vee \to) \qquad \frac{\Gamma \to A_i}{\Gamma \to A_1 \vee A_2}\ (\to \vee),\ \text{where } i = 1 \text{ or } 2$$

$$\frac{\Gamma_1, A_i, \Gamma_2 \to C}{\Gamma_1, A_1 \wedge A_2, \Gamma_2 \to C}\ (\wedge \to),\ \text{where } i = 1 \text{ or } 2 \qquad \frac{\Gamma \to A_1 \quad \Gamma \to A_2}{\Gamma \to A_1 \wedge A_2}\ (\to \wedge)$$

$$\frac{\Gamma_1, A, \Gamma_2 \to C}{\Gamma_1, !^s A, \Gamma_2 \to C}\ (! \to) \qquad \frac{!^{s_1} A_1, \dots, !^{s_n} A_n \to B}{!^{s_1} A_1, \dots, !^{s_n} A_n \to !^s B}\ (\to !),\ \text{where } s_j \succeq s \text{ for all } j$$

$$\frac{\Gamma_1, \Gamma_2 \to C}{\Gamma_1, !^s A, \Gamma_2 \to C}\ (\text{weak}),\ \text{where } s \in \mathcal{W}$$

$$\frac{\Gamma_1, !^s A, \Delta, !^s A, \Gamma_2 \to C}{\Gamma_1, !^s A, \Delta, \Gamma_2 \to C}\ (\text{ncontr}_1) \quad \text{and} \quad \frac{\Gamma_1, !^s A, \Delta, !^s A, \Gamma_2 \to C}{\Gamma_1, \Delta, !^s A, \Gamma_2 \to C}\ (\text{ncontr}_2),\ \text{where } s \in \mathcal{C}$$

$$\frac{\Gamma_1, \Delta, !^s A, \Gamma_2 \to C}{\Gamma_1, !^s A, \Delta, \Gamma_2 \to C}\ (\text{ex}_1) \quad \text{and} \quad \frac{\Gamma_1, !^s A, \Delta, \Gamma_2 \to C}{\Gamma_1, \Delta, !^s A, \Gamma_2 \to C}\ (\text{ex}_2),\ \text{where } s \in \mathcal{E}$$

$$\frac{\Pi \to A \quad \Gamma_1, A, \Gamma_2 \to C}{\Gamma_1, \Pi, \Gamma_2 \to C}\ (\text{cut}).$$

Due to the special status of the cut rule, we always explicitly state whether we are using it in our derivations. Namely, we use the notation SMALC_Σ for the cut-free calculus and $\mathrm{SMALC}_\Sigma + (\mathrm{cut})$ for the calculus with the cut rule.

It is sufficient to postulate (ax) only for variables, in the form $p_i \to p_i$. All other instances of $A \to A$ are then derivable in a standard manner, without using (cut). For the subexponential case, derivability of $!^s A \to !^s A$ is due to reflexivity of $\preceq$.

In Section 7, we prove the cut elimination theorem for SMALC_Σ (Corollary 3), that is, SMALC_Σ+(cut) and SMALC_Σ derive the same set of theorems. This yields the subformula property, and therefore it becomes very easy to consider fragments of the system by restricting the language. If we take only rules that operate multiplicative connectives, $\cdot$, $\backslash$, and $/$, and rules that operate subexponentials, $!^s$ ($s \in \mathcal{I}$), we obtain the subexponential extension of the 'pure' Lambek calculus, denoted by SLC_Σ. If we also include the unit constant, $\mathbf{1}$, we get the calculus $\mathrm{SLC}^{\mathbf{1}}_\Sigma$. Finally, removing rules for subexponentials yields, respectively, the Lambek calculus allowing empty antecedents (Lambek 1961) and the Lambek calculus with the unit (Lambek 1969). All these calculi are conservative fragments of SMALC_Σ.

Notice that the version of the Lambek calculus considered in this paper allows the antecedents of sequents to be empty, while the original system by Lambek (1958) does not. This constraint, called *Lambek's restriction,* is motivated by linguistic applications of the Lambek calculus. This restriction, however, appears to be incompatible with (sub)exponential modalities. Namely, as shown by Kanovich et al. (2016a,c), if an extension of the Lambek calculus with an exponential modality equipped with the full set of structural rules (exchange, contraction and weakening) enjoys both cut elimination and substitution property, then Lambek's restriction is essentially violated: $!p, \Pi \to A$ is derivable for any sequent $\Pi \to A$ derivable in the Lambek calculus allowing empty antecedents. A similar result exists for the relevant subexponential modality (which allows contraction and exchange, but not weakening): if $\Pi \to A$ is derivable in the Lambek calculus allowing empty antecedents and contains only one variable p, then $!(p \setminus p), \Pi \to A$ should be derivable in the extension with the relevant modality.

For this reason, throughout this paper, we consider the Lambek calculus allowing empty antecedents (Lambek 1961).

3. Application of SMALC_Σ for system specification

It seems that the proof system SMALC_Σ can be used as a basis for a general logical framework. While existing logical frameworks such as Hodas and Miller (1991, 1994), Miller (1994, 1996), Nigam (2014) and Nigam et al. (2016) can only manipulate logically sets and multisets of formulae, a logical framework based on SMALC_Σ could also manipulate lists of formulae. Moreover, differently from the work of Pfenning and Simmons (2009), where only one non-commutative context can be specified, SMALC_Σ allows for multiple contexts that can all be treated as lists of formulae. This is because SMALC_Σ supports subexponentials.

While the concrete proposal of a logical framework based on SMALC_Σ is left for future work, we illustrate the features of SMALC_Σ (non-commutative formulae and subexponentials) by specifying a computational, distributed system. Consider a system with n machines called $m_1, \ldots, m_n$. Assume that each machine has an input FIFO buffer. Whenever a machine receives a message, it is stored at the beginning of the buffer, and the message at the end of the buffer is processed first by a machine.

Since SMALC_Σ allows for non-commutative subexponentials, it is possible to specify such a system declaratively using SMALC_Σ, that is, without relying on encoding of lists using terms, but only using logical connectives. Assume a subexponential signature Σ with the indexes $\mathcal{I} = \{m_1, \ldots, m_n, u\}$; $\preceq$ is the reflexive relation, that is, $i \preceq j$ if and only if $i = j$; $\mathcal{E} = \mathcal{C} = \mathcal{W} = \{u\}$. Intuitively, the subexponential m_i will specify the buffer of machine m_i and the subexponential u will specify the theory governing the buffers. Assume a finite set of possible elements.[§]

A buffer at machine m_i with elements $\mathsf{E} = \mathsf{e}_1, \ldots, \mathsf{e}_k$ is specified as the list of formulae in SMALC_Σ

$$!^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end},$$

or $!^{m_i}[\mathsf{start}, \mathsf{E}, \mathsf{end}]$ for short. Thus, a system with n machines is specified as the collection of formulae

$$!^{m_1}[\mathsf{start}, \mathsf{E}_1, \mathsf{end}], !^{m_2}[\mathsf{start}, \mathsf{E}_2, \mathsf{end}], \ldots, !^{m_n}[\mathsf{start}, \mathsf{E}_n, \mathsf{end}].$$

§ This restriction could be overcome by adding quantifiers to our system. This would allow us to quantify over infinite domains such as in $\forall X.e(X)$. This is left, however, for future work.

A sequent specifying such system has the following shape:

$$\Psi, !^{m_1}[\mathsf{start}, \mathsf{E}_1, \mathsf{end}], !^{m_2}[\mathsf{start}, \mathsf{E}_2, \mathsf{end}], \ldots, !^{m_n}[\mathsf{start}, \mathsf{E}_n, \mathsf{end}] \to G,$$

where $\Psi = \Theta_{\text{consume}}, \Theta_{\text{add}}$ are the following formulae specifying the behaviour of the buffers:

— Θ_{consume} is the collection of formulae

$$!^u((!^{m_i}\mathsf{e} \cdot !^{m_i}\mathsf{end}) \setminus !^{m_i}\mathsf{end}),$$

for $1 \leqslant i \leqslant n$ and for all elements e. Such a formula specifies the consumption of an element at the end of a buffer.

This is illustrated by the following derivation, where Δ, Δ' contain the specifications of the buffers of the remaining machines:

$$\dfrac{\dfrac{\dfrac{\overline{\overline{!^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end} \to !^{m_i}\mathsf{e}_k \cdot !^{m_i}\mathsf{end}}}\;(\to \cdot), (\mathrm{ax}) \qquad \Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, \ldots, !^{m_i}\mathsf{e}_{k-1}, !^{m_i}\mathsf{end}, \Delta' \to G}{\Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, \ldots, !^{m_i}\mathsf{e}_{k-1}, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, (!^{m_i}\mathsf{e}_k \cdot !^{m_i}\mathsf{end}) \setminus !^{m_i}\mathsf{end}, \Delta' \to G}\;(\setminus \to)}{\Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, \ldots, !^{m_i}\mathsf{e}_{k-1}, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, !^u((!^{m_i}\mathsf{e}_k \cdot !^{m_i}\mathsf{end}) \setminus !^{m_i}\mathsf{end}), \Delta' \to G}\;(! \to)}{\Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, \ldots, !^{m_i}\mathsf{e}_{k-1}, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}\;(\mathrm{ncontr}_2).$$

In the left branch, the last element of the list is consumed. This can be observed in the right branch. The last element in the list of elements is e_{k-1}. Notice as well, that the only place where the formula $!^u((!^{m_i}\mathsf{e}_k \cdot !^{m_i}\mathsf{end}) \setminus !^{m_i}\mathsf{end})$ can be used (that is introduced) is as in the derivation above. If the last element is not e_k or if it is used in the list of another machine, the derivation above would fail.

— Similarly, Θ_{add} is the collection of

$$!^u((!^{m_i}\mathsf{start} \cdot !^{m_i}\mathsf{e}) \,/\, !^{m_i}\mathsf{start}),$$

for $1 \leqslant i \leqslant n$ and for each element e. Such a formula specifies the addition of an element at the beginning of a buffer.

This is illustrated by the following derivation, where Δ, Δ' contain the specifications of the buffers of the remaining machines:

$$\dfrac{\dfrac{\dfrac{\overline{!^{m_i}\mathsf{start} \to !^{m_i}\mathsf{start}}\;(\mathrm{ax}) \qquad \dfrac{\Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}, !^{m_i}\mathsf{e}_1, !^{m_i}\mathsf{e}_2 \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}{\Psi, \Delta, !^{m_i}\mathsf{start} \cdot !^{m_i}\mathsf{e}, !^{m_i}\mathsf{e}_1, !^{m_i}\mathsf{e}_2 \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}\;(\cdot \to)}{\Psi, \Delta, (!^{m_i}\mathsf{start} \cdot !^{m_i}\mathsf{e}) \,/\, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, !^{m_i}\mathsf{e}_2 \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}\;(/ \to)}{\Psi, \Delta, !^u((!^{m_i}\mathsf{start} \cdot !^{m_i}\mathsf{e}) \,/\, !^{m_i}\mathsf{start}), !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, !^{m_i}\mathsf{e}_2 \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}\;(! \to)}{\Psi, \Delta, !^{m_i}\mathsf{start}, !^{m_i}\mathsf{e}_1, !^{m_i}\mathsf{e}_2 \ldots, !^{m_i}\mathsf{e}_k, !^{m_i}\mathsf{end}, \Delta' \to G}\;(\mathrm{ncontr}_2).$$

The left branch ensures that the formula is contracted at the correct location. Otherwise, it would not be provable. The right branch adds the element e at the beginning of m_i's buffer.

We leave to future work the proposal of a logical framework based on SMALC_Σ. This includes the proposal of a focussed proof system for SMALC_Σ as well as further illustrations on how it can be used for the specification of systems.

4. Linguistic applications of SMALC_Σ

The Lambek calculus, which we now consider as non-commutative intuitionistic linear logic (see Abrusci 1990), was initially introduced as the logical background for type-logical grammars, an approach to interpret natural language syntax as derivability in logical calculi (Ajdukiewicz 1935; Bar-Hillel 1953; Lambek 1958). One of the frameworks developed on the basis of this idea is the framework of Lambek categorial grammars. In Lambek grammars, the two division operations of the Lambek calculus, $\backslash$ and $/$, are interpreted as follows: a syntactic object belongs to type $A \backslash B$ if it lacks an object of type A to be attached on the left side to become an object of type B ($/$ is symmetric). Thus, the Lambek calculus becomes a calculus of syntactic types. Usually, S for 'sentence' and N for 'noun phrase' are used as basic types, and more complicated types are constructed using $\backslash$ and $/$. For example, $N \backslash S$ is something that lacks a noun phrase on the left to become a sentence, i.e., an intransitive verb phrase. This approach allows analysis of many syntactic structures commonly found in English and other natural languages. For example, if *'John'* and *'Mary'* are nouns (N) and the transitive verb *'loves'* is of type $N \backslash (S / N)$, then the sequent $N, (N \backslash S) / N, N \to S$, being a theorem of the Lambek calculus, states that *'John loves Mary'* is a correct sentence. More complex examples include dependent clauses: *'the girl whom John loves'* receives type N, using the following type assignment: $N / CN, CN, (CN \backslash CN) / (S / N), N, (N \backslash S) / N \to N$ (here, CN stands for 'common noun,' a noun without an article), coordination: *'John loves Mary and Pete loves Kate'* is of type S, provided that *'and'* is assigned the type $(S \backslash S) / S$, etc.

Notice that non-commutative linear logic exactly fits to the place of the basic logic for type-logical grammar. Applying contraction and weakening structural rules is generally not allowed. One can neither add extra meaningless words into a text (i.e., apply weakening), nor contract several instances of the same word into one (cf. examples, like *'Buffalo buffalo buffalo . . . '*), without losing grammaticality or changing the meaning of the sentence. Non-commutativity is also important, at least for languages like English: indeed, *'John runs'* is a correct sentence ($N, N \backslash S \to S$ is derivable in the Lambek calculus), but *'runs John'* is not (and $N \backslash S, N \to S$ is not derivable). As we shall see below, however, structural rules are sometimes allowed to be restored in a controlled way.

Unfortunately, the expressive capacity of the 'pure' Lambek calculus is rather limited. This empirical fact is formally justified by Pentus' theorem (Pentus 1993) which states that Lambek grammars can generate only context-free languages; on the other side, natural language syntax can be essentially non-context-free, as discussed by Shieber (1985) on the material of Swiss German. (Pentus' translation of Lambek grammars into context-free ones increases the size of grammar exponentially: thus, Lambek grammars, compared to context-free grammars, can still win in efficiency. For the fragment of the Lambek calculus with only one division, there also exists a polynomial translation (Kuznetsov 2016).)

In order to cover even more sophisticated syntactic phenomena, the Lambek calculus needs to be extended by means of adding new logical connectives and/or using more complicated structure of the sequents. In particular, Morrill and Valentín (2015) suggest to use a ! modality that allows contraction for modelling a syntactic phenomenon called *parasitic extraction* from dependent clauses. Usually, the dependent clause is obtained from an ‘independent’ sentence by omitting one noun phrase (creating a *gap*). For example, in *‘the girl whom John loves’* the dependent clause is *‘John loves . . . ,’* where . . . denotes the gap for noun phrase. In a normal sentence, the gap would have been filled with *Mary,* for example. In Lambek grammar, this corresponds to the fact that *‘John loves’* is of type $S \,/\, N$. Now let us consider examples like *‘the paper that John signed without reading.’* Here, the dependent clause has *two* gaps: *‘John signed . . . without reading . . . ,’* or even more, like in *‘[the paper that] the editor of . . . received . . . , but left . . . in the office without reading . . . ’* Semantically, all these gaps have to be filled with the same instance of *paper.* The contraction rule is capable of handling parasitic extraction. Now the type for *‘that’* becomes $(CN \backslash CN) \,/\, (S \,/\, !N)$ rather than $(CN \backslash CN) \,/\, (S \,/\, N)$, and the contraction rule for $!N$ allows it to branch and fill both gaps. One can see this below in the derivation of the Lambek sequent for *‘the paper that John signed without reading’:*

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
N \to N \quad
\dfrac{
N \to N \quad
\dfrac{
N \backslash S \to N \backslash S \quad
\dfrac{
N \backslash S \to N \backslash S \quad
\dfrac{N \to N \quad S \to S}{N, N \backslash S \to S}
}{N, N \backslash S, (N \backslash S) \backslash (N \backslash S) \to S}
}{N, N \backslash S, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), N \backslash S \to S}
}{N, N \backslash S, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N, N \to S}
}{N, (N \backslash S) / N, N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N, N \to S}
}{N, (N \backslash S) / N, N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N, !N \to S}
}{N, (N \backslash S) / N, !N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N, !N \to S}
}{N, (N \backslash S) / N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N, !N \to S}
}{N, (N \backslash S) / N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N \to S / !N}
\qquad
\dfrac{CN \to CN \quad \dfrac{CN \to CN \quad N \to N}{N / CN, CN \to N}}{N / CN, CN, CN \backslash CN \to N}
}{N / CN, CN, (CN \backslash CN) / (S / !N), N, (N \backslash S) / N, ((N \backslash S) \backslash (N \backslash S)) / (N \backslash S), (N \backslash S) / N \to N.}
$$

Notice that, here, we use the non-local version of the contraction rule. Alternatively, we could have also allowed exchange (cf. Morrill and Valentín), and then the difference between local and non-local contraction disappears. The exchange rule also allows *medial extraction,* i.e., handling situations where the gap is in the middle of the dependent clause: *‘the girl whom John met yesterday.’* Here, *‘John met . . . yesterday’* is of type $S \,/\, !N$, where ! allows exchange, but not $S \,/\, N$ or $N \backslash S$. The weakening rule, however, should not be allowed, since dependent clauses without gaps are ungrammatical. Thus, the ! modality used by Morrill and Valentín is a *relevant* modality (Kanovich et al. 2016b) – in our setting, it becomes $!^s$, where $s \in \mathcal{C}$, $s \in \mathcal{E}$, $s \notin \mathcal{W}$. A variant of the relevant modality, for which the contraction rule is modified to operate together with a controlled non-associativity mechanism called *bracket modalities,* appears in the basic logic used in grammars for the *CatLog* parser (Morrill 2017a).

Finally, the idea of using an indexed family of connectives with different structural properties was developed by Moortgat (1996) in his *multimodal* extension of the non-associative version of Lambek calculus. Multimodal categorial grammars are implemented in the *Grail* parser (Moot 2017).

5. Cyclic linear logic with subexponentials

In this section, we define the second calculus considered in this paper, the extension of *cyclic linear logic* (Yetter 1990) with subexponentials. For a subexponential signature Σ, this calculus is denoted by SCLL_Σ.

We formulate SCLL_Σ in a language with *tight negations*. For a countable set of variables $\mathrm{Var} = \{p_1, p_2, \ldots\}$, we also consider their negations $\bar{p}_1, \bar{p}_2, \ldots$; variables and their negations are called *atoms*. Formulae of SCLL_Σ are built from atoms and constants $\mathbf{1}$ (multiplicative truth), $\bot$ (multiplicative falsity), $\top$ (additive truth) and $\mathbf{0}$ (additive falsity) using four binary connectives: $\otimes$ (multiplicative conjunction), $\parr$ (multiplicative disjunction), $\&$ (additive conjunction) and $\oplus$ (additive disjunction), and also two families of unary connectives, indexed by the subexponential signature Σ: $!^s$ (universal subexponential) and $?^s$ (existential subexponential) for each $s \in \mathcal{I}$ (recall that $\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$, and $\mathcal{I}$ is the set of all subexponential labels).

Negation for arbitrary formulae introduced externally by the following recursive definition ($A^\bot$ means 'not A'):

$$\begin{array}{ll}
p_i^\bot = \bar{p}_i & (!^s A)^\bot = ?^s A^\bot \\
\bar{p}_i^\bot = p_i & (?^s A)^\bot = !^s A^\bot \\
(A \otimes B)^\bot = B^\bot \parr A^\bot & \mathbf{1}^\bot = \bot \\
(A \parr B)^\bot = B^\bot \otimes A^\bot & \bot^\bot = \mathbf{1} \\
(A \oplus B)^\bot = A^\bot \& B^\bot & \mathbf{0}^\bot = \top \\
(A \& B)^\bot = A^\bot \oplus B^\bot & \top^\bot = \mathbf{0}.
\end{array}$$

Sequents of SCLL_Σ are of the form $\vdash \Gamma$, where Γ in SCLL_Σ is a non-empty *cyclically ordered sequence:* sequents $\vdash \Gamma_1, \Gamma_2$ and $\vdash \Gamma_2, \Gamma_1$ are considered graphically equal (i.e., different forms of writing down the same sequent), but other permutations of formulae within $\vdash \Gamma$ are not allowed. One can think that the sequent $\Gamma = A_1, \ldots, A_n$ is actually written on a circle, without any starting point ('$\vdash$').

The axioms and rules of inference of SCLL_Σ are as follows:

$$\frac{}{\vdash A, A^\bot}\ (\mathrm{ax})$$

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ (\otimes) \qquad \frac{\vdash A, B, \Gamma}{\vdash A \parr B, \Gamma}\ (\parr)$$

$$\frac{\vdash A_1, \Gamma \quad \vdash A_2, \Gamma}{\vdash A_1 \& A_2, \Gamma}\ (\&) \qquad \frac{\vdash A_i, \Gamma}{\vdash A_1 \oplus A_2, \Gamma}\ (\oplus),\ \text{where } i = 1 \text{ or } 2$$

$$\frac{}{\vdash \mathbf{1}}\ (\mathbf{1}) \qquad \frac{\vdash \Gamma}{\vdash \bot, \Gamma}\ (\bot) \qquad \frac{}{\vdash \top, \Gamma}\ (\top)$$

$$\frac{\vdash B, ?^{s_1} A_1, \ldots, ?^{s_n} A_n}{\vdash !^s B, ?^{s_1} A_1, \ldots, ?^{s_n} A_n}\ (!),\ \text{where } s_j \succeq s \text{ for all } j$$

$$\frac{\vdash A, \Gamma}{\vdash ?^s A, \Gamma}\ (?)$$

$$\frac{\vdash \Gamma}{\vdash ?^s A, \Gamma} \text{ (weak), where } s \in \mathcal{W}$$

$$\frac{\vdash ?^s A, \Gamma, ?^s A, \Delta}{\vdash ?^s A, \Gamma, \Delta} \text{ (ncontr), where } s \in \mathcal{C}$$

$$\frac{\vdash \Gamma, ?^s A, \Delta}{\vdash ?^s A, \Gamma, \Delta} \text{ (ex), where } s \in \mathcal{E}$$

$$\frac{\vdash \Gamma, A^\perp \quad \vdash A, \Delta}{\vdash \Gamma, \Delta} \text{ (cut).}$$

Note that there is no rule for additive falsity, $\mathbf{0}$. The only way to introduce $\mathbf{0}$ is by $(\top)$ or (ax), yielding $\vdash \top, \Gamma_1, \mathbf{0}, \Gamma_2$ (if we use (ax), Γ_1 and Γ_2 are empty).

Also, notice that, we can freely apply cyclic transformations to our sequents, yielding rules of the form

$$\frac{\vdash \Gamma_1, A, B, \Gamma_2}{\vdash \Gamma_1, A \parr B, \Gamma_2} (\parr) \qquad \frac{\vdash \Gamma_1, A, \Gamma_2 \quad \vdash B, \Delta}{\vdash \Gamma_1, A \otimes B, \Delta, \Gamma_2} (\otimes_1) \qquad \frac{\vdash \Delta, A \quad \vdash \Gamma_1, B, \Gamma_2}{\vdash \Gamma_1, \Delta, A \otimes B, \Gamma_2} (\otimes_2).$$

and so on. Due to our conventions, these rules are actually *graphically equal* to the official rules of SCLL_Σ presented above. Sometimes, however, these transformed forms of the rules are more convenient – for example, if we want a specific designated formula to be the rightmost one (see proof of Theorem 2).

As in SMALC_Σ, in SCLL_Σ, it is sufficient to postulate (ax) only for variables, as $\vdash p_i, \bar{p}_i$.

As for the Lambek calculus, we use the notation SCLL_Σ for the cut-free system, and $\mathrm{SCLL}_\Sigma + \text{(cut)}$ for the system with cut. In Section 6, we establish cut elimination, that yields the subformula property. If we remove all additives connectives and rules for them, leaving only $\mathbf{1}$, $\perp$, $\otimes$, $\parr$, and the subexponentials, we get the *multiplicative* fragment of cyclic linear logic with subexponentials, denoted by SMCLL_Σ.

6. Cut elimination in SCLL_Σ

Theorem 1. A sequent is derivable in $\mathrm{SCLL}_\Sigma + \text{(cut)}$ if and only if it is derivable in SCLL_Σ.

The cut elimination strategy we use here goes back to Gentzen (1935), and was applied for linear logic by Girard (1987). We follow the outline of the proof presented in Lincoln et al. (1992, Appendix A), making necessary modifications for the cases where exchange rules are not available.

Since eliminating the cut rule by straightforward induction encounters problems when it comes across the contraction rule, we consider the cut rule together with a more general rule called *mix*, which is a combination of cut and contraction. The two rules can now be eliminated by joint induction (which is impossible for the original cut rule alone).

Another possible cut elimination strategy for SCLL_Σ is 'deep cut elimination' of Braüner and de Paiva (1998). This strategy is applied by Kanovich et al. (2017) to establish cut elimination in a system closely related to SLC_Σ, but with bracket modalities that introduce controlled non-associativity, which makes it hard to formulate the mix rule. In this paper, we follow the more traditional approach.

Since mix needs contraction, it is included only for formulae of the form $?^s A$ with $s \in \mathcal{C}$. Thus, unlike the classic Gentzen's situation, (cut) is not always a particular case of (mix), and in our proof, we eliminate both cut and mix by joint induction.

If $s \in \mathcal{C} \cap \mathcal{E}$ (i.e., $?^s$ also allows exchange – in particular, this is the case for the 'full-power' exponential connective of linear logic), the mix rule can be formulated exactly as in the commutative case

$$\frac{\vdash \Gamma, !^s A^\perp \quad \vdash ?^s A, \ldots, ?^s A, \Delta}{\vdash \Gamma, \Delta}\ (\mathrm{mix}).$$

For $s \in \mathcal{C} - \mathcal{E}$, however, the formulation of mix is more sophisticated, since we are not allowed to gather all instances of $?^s A$ in one area of the sequent

$$\frac{\vdash \Gamma, !^s A^\perp \quad \vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}{\vdash \Gamma, \Delta_1, \Delta_2, \ldots, \Delta_k}\ (\mathrm{mix}).$$

In this rule, one instance of $?^s A$ is replaced with Γ (due to cyclicity, we can suppose that it is the leftmost occurrence), and several (maybe zero) other occurrences of $?^s A$ are removed from the sequent (contracted).

Being equivalent to a consequent application of several (ncontr)'s and (cut), the mix rule is clearly admissible in $\mathrm{SCLL}_\Sigma + (\mathrm{cut})$.

As in the commutative case, cut elimination crucially depends on the fact that the $\preceq$ relation is transitive and that the sets $\mathcal{W}$, $\mathcal{C}$, and $\mathcal{E}$ are upwardly closed w.r.t. $\preceq$. These parts of the definition of the substructural signature Σ come into play when we propagate (cut) or (mix) through the $(!)$ rule that yields $\vdash ?^{s_1} C_1, \ldots, ?^{s_n} C_n, !^s A^\perp$. In this situation, the formula $?^s A$ get replaced by a sequence $?^{s_1} C_1, \ldots, ?^{s_n} C_n$, and we need the same structural rules to be valid for $?^{s_i} C_i$, as for $?^s A$. This is guaranteed by the fact that $s_i \succeq s$ (a prerequisite of the $(!)$ rule) and the closure properties of Σ.

In the non-commutative situation, however, there is another issue one should be cautious about. For cut elimination, it is important that the contraction rule is non-local, i.e., the formulae being contracted can come from distant places of the sequent, with other formulae (Γ) between them. Accordingly, our formulation of (mix) for subexponentials that allow contraction, but not exchange, is also non-local, with Δ_i between the active formulae. In Section 8, we show that for the local version of contraction, that allows contracting only neighbour formulae, cut elimination does not hold.

Proof of Theorem 1. As usual, it is sufficient to eliminate one cut or mix, i.e., to show the following two statements:

— if both $\vdash \Gamma, A^\perp$ and $\vdash A, \Delta$ are derivable in SCLL_Σ, then so is $\vdash \Gamma, \Delta$;
— if $s \in \mathcal{C}$ and both $\vdash \Gamma, !^s A^\perp$ and $\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k$ are derivable in SCLL_Σ, then so is $\vdash \Gamma, \Delta_1, \ldots, \Delta_k$.

We prove both statements by joint nested induction. The outer induction parameter is κ, the total number of connectives in the formula being cut (for (mix), the external $?^s$ also counts). The inner induction parameter is δ, the sum of the heights of the cut-free derivations of two premises, $\vdash \Gamma, A^\perp$ and $\vdash A, \Delta$ for (cut) and $\vdash \Gamma, !^s A^\perp$ and $\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k$ for (mix). At each step, either κ decreases, or δ decreases while κ remains the same.

The cut (mix) elimination procedure is usually very lengthy and tedious, since it requires considering a great number of cases and subcases of which rules are the last rules applied in the (cut-free) derivations of the premises of (cut) or (mix). Here, we try to make it as short as possible by merging similar cases.

6.1. *Cut elimination*

The cut elimination procedure is a rather standard, straightforward induction. When we come across the (ncontr) rule, (cut) becomes (mix), and we jump to the second, more interesting part of the proof.

The last rule applied in the derivation of $\vdash \Gamma, A^\perp$ (or, symmetrically, $\vdash A, \Delta$) is called *principal* either if it is an application of the (!) rule or if it introduces the rightmost $A^\perp$ (symmetrically, the leftmost A) formula. Otherwise, it is called *non-principal.*

Case 1. One of the cut premises is an axiom of the form (ax). Then, the goal sequent coincides with the other premise, and cut disappears.

Case 2. The last rule in the derivation either of $\vdash \Gamma, A^\perp$ or of $\vdash A, \Delta$ is non-principal.

Since $A^{\perp\perp} = A$, the cut (but not mix) rule is *symmetric*. Therefore, we do not have to consider both $\vdash \Gamma, A^\perp$ and $\vdash A, \Delta$; handling only $\vdash \Gamma, A^\perp$ is sufficient.

Let us call ($\mathbin{⅋}$), ($\oplus$), ($\perp$), (?), (weak), (ncontr) and (ex) *easy* rules. An easy rule does not branch the derivation, it only transforms the sequent, and, in the non-principal case, keeps the formula being cut intact. If $\vdash \Gamma, A^\perp$ is derived using an easy rule, the cut application has the following form ('ER' stands for 'easy rule'):

$$\dfrac{\dfrac{\vdash \widetilde{\Gamma}, A^\perp}{\vdash \Gamma, A^\perp}\,(\mathrm{ER}) \qquad \vdash A, \Delta}{\vdash \Gamma, \Delta}\,(\mathrm{cut}),$$

and the cut is propagated

$$\dfrac{\dfrac{\vdash \widetilde{\Gamma}, A^\perp \qquad \vdash A, \Delta}{\vdash \widetilde{\Gamma}, \Delta}\,(\mathrm{cut})}{\vdash \Gamma, \Delta}\,(\mathrm{ER}).$$

The easy rule is still valid in a different context. The new cut has the same κ and a smaller δ parameter, and gets eliminated by induction.

The other non-principal cases, ($\otimes$), ($\&$) and ($\top$), are handled as follows:

$$\dfrac{\dfrac{\vdash \Phi, E \qquad \vdash \Gamma_1, F, \Gamma_2, A^\perp}{\vdash \Gamma_1, \Phi, E \otimes F, \Gamma_2, A^\perp}\,(\otimes) \qquad \vdash A, \Delta}{\vdash \Gamma_1, \Phi, E \otimes F, \Gamma_2, \Delta}\,(\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{\vdash \Phi, E \qquad \dfrac{\vdash \Gamma_1, F, \Gamma_2, A^\perp \qquad \vdash A, \Delta}{\vdash \Gamma_1, F, \Gamma_2, \Delta}\,(\mathrm{cut})}{\vdash \Gamma_1, \Phi, E \otimes F, \Gamma_2, \Delta}\,(\otimes).$$

(The case when $A^\perp$ goes to the branch with E is symmetric)

$$\dfrac{\dfrac{\vdash \Gamma_1, E_1, \Gamma_2, A^\perp \quad \vdash \Gamma_1, E_2, \Gamma_2, A^\perp}{\vdash \Gamma_1, E_1 \mathbin{\&} E_2, \Gamma_2, A^\perp}\ (\&) \qquad \vdash A, \Delta}{\vdash \Gamma_1, E_1 \mathbin{\&} E_2, \Gamma_2, \Delta}\ (\mathrm{cut}),$$

becomes

$$\dfrac{\dfrac{\vdash \Gamma_1, E_1, \Gamma_2, A^\perp \quad \vdash A, \Delta}{\vdash \Gamma_1, E_1, \Gamma_2, \Delta}\ (\mathrm{cut}) \qquad \dfrac{\vdash \Gamma_1, E_2, \Gamma_2, A^\perp \quad \vdash A, \Delta}{\vdash \Gamma_1, E_2, \Gamma_2, \Delta}\ (\mathrm{cut})}{\vdash \Gamma_1, E_1 \mathbin{\&} E_2, \Gamma_2, \Delta}\ (\&)$$

$$\dfrac{\dfrac{}{\vdash \Gamma_1, \top, \Gamma_2, A^\perp}\ (\top) \qquad \vdash A, \Delta}{\vdash \Gamma_1, \top, \Gamma_2, \Delta}\ (\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{}{\vdash \Gamma_1, \top, \Gamma_2, \Delta}\ (\top).$$

For $(\otimes)$ and $(\&)$, the δ parameter decreases with the same κ. For $(\top)$, cut disappears.

Applications of $(\mathbf{1})$ and $(!)$ cannot be non-principal.

Case 3. The last rules in both derivations are principal, and the main connective of A is not a subexponential. Consider the possible pairs of principal rules; due to symmetry of cut, the order in these pairs does not matter.

Subcase 3.1. $(\otimes)$ and $(\invamp)$

$$\dfrac{\dfrac{\vdash \Gamma_2, F^\perp \quad \vdash \Gamma_1, E^\perp}{\vdash \Gamma_1, \Gamma_2, F^\perp \otimes E^\perp}\ (\otimes) \qquad \dfrac{\vdash E, F, \Delta}{\vdash E \invamp F, \Delta}\ (\invamp)}{\vdash \Gamma_1, \Gamma_2, \Delta}\ (\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{\vdash \Gamma_2, F^\perp \qquad \dfrac{\vdash \Gamma_1, E^\perp \quad \vdash E, F, \Delta}{\vdash \Gamma_1, F, \Delta}\ (\mathrm{cut})}{\vdash \Gamma_1, \Gamma_2, \Delta}\ (\mathrm{cut}).$$

The κ parameter for both new cuts is less than κ of the original cut; thus, we can proceed by induction.

Subcase 3.2. $(\&)$ and $(\oplus)$

$$\dfrac{\dfrac{\vdash \Gamma, E_1^\perp \quad \vdash \Gamma, E_2^\perp}{\vdash \Gamma, E_1^\perp \mathbin{\&} E_2^\perp}\ (\&) \qquad \dfrac{\vdash E_i, \Delta}{\vdash E_1 \oplus E_2, \Delta}}{\vdash \Gamma, \Delta}\ (\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{\vdash \Gamma, E_i^\perp \quad \vdash E_i, \Delta}{\vdash \Gamma, \Delta}\ (\mathrm{cut}).$$

Again, κ gets decreased.

Subcase 3.3. $(\mathbf{1})$ and $(\perp)$

$$\dfrac{\dfrac{}{\vdash \mathbf{1}}\ (\mathbf{1}) \qquad \dfrac{\vdash \Delta}{\vdash \perp, \Delta}\ (\perp)}{\vdash \Delta}\ (\mathrm{cut}).$$

Cut disappears, since its goal coincides with the premise of $(\perp)$, which is already derived.

In the principal case, we do not need to consider the $(\top)$ rule, since it has no principal counterpart that introduces $\top^\perp = \mathbf{0}$.

Case 4. Both last rules are principal, $A = ?^s B$ and $A^\perp = !^s B^\perp$. The left premise, $\vdash \Gamma, !^s B^\perp$, is derived using $(!)$ by introducing $!^s B^\perp$. Therefore, $\Gamma = ?^{s_1} C_1, \ldots, ?^{s_n} C_n$, where $s_i \succeq s$ for all i. Consider the possible cases for the last rule in the derivation of the other premise, $\vdash ?^s B, \Delta$.

Subcase 4.1. The last rule is (?)

$$\dfrac{\dfrac{\vdash \Gamma, B^{\perp}}{\vdash \Gamma, !^s B^{\perp}}\,(!) \quad \dfrac{\vdash B, \Delta}{\vdash ?^s B, \Delta}\,(?)}{\vdash \Gamma, \Delta}\,(\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{\vdash \Gamma, B^{\perp} \quad \vdash B, \Delta}{\vdash \Gamma, \Delta}\,(\mathrm{cut}).$$

The κ parameter gets decreased.

Subcase 4.2. The last rule is (!)

$$\dfrac{\vdash ?^{s_1}C_1, \ldots, ?^{s_n}C_n, !^s B^{\perp} \quad \dfrac{\vdash ?^s B, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}{\vdash ?^s B, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, !^q E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}\,(!)}{\vdash ?^{s_1}C_1, \ldots, ?^{s_n}C_n, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, !^q E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}\,(\mathrm{cut}),$$

becomes

$$\dfrac{\dfrac{\vdash ?^{s_1}C_1, \ldots, ?^{s_n}C_n, !^s B^{\perp} \quad \vdash ?^s B, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}{\vdash ?^{s_1}C_1, \ldots, ?^{s_n}C_n, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}\,(\mathrm{cut}),}{\vdash ?^{s_1}C_1, \ldots, ?^{s_n}C_n, ?^{q_1}D_1, \ldots, ?^{q_i}D_i, !^q E, ?^{q_{i+1}}D_{i+1}, \ldots, ?^{q_m}D_m}\,(!)$$

where the new application of (!) is legal due to transitivity of $\preceq$: $s_i \succeq s \succeq q$. The κ parameter is the same, δ decreases.

Subcase 4.3. The last rule is (weak). In this case, since $s_i \succeq s$ and $s \in \mathcal{W}$, then also $s_i \in \mathcal{W}$ and $\Gamma = ?^{s_1}C_1, \ldots, ?^{s_n}C_n$ can be added to Δ using the weakening rule n times. Cut disappears.

Subcase 4.4. The last rule is (ncontr). In this case, cut is replaced by mix with the same κ and a smaller δ

$$\dfrac{\vdash \Gamma, !^s B^{\perp} \quad \dfrac{\vdash ?^s B, \Delta_1, ?^s B, \Delta_2}{\vdash ?^s B, \Delta_1, \Delta_2}\,(\mathrm{ncontr})}{\vdash \Gamma, \Delta_1, \Delta_2}\,(\mathrm{cut}) \quad \rightsquigarrow \quad \dfrac{\vdash \Gamma, !^s B^{\perp} \quad \vdash ?^s B, \Delta_1, ?^s B, \Delta_2}{\vdash \Gamma, \Delta_1, \Delta_2}\,(\mathrm{mix}).$$

Subcase 4.5. The last rule is (ex). Similarly to Subcase 4.3, $s_i \in \mathcal{E}$, and we can apply the exchange rule for Γ as a whole. This means that (cut) can be interchanged with (ex), decreasing δ with the same κ.

6.2. *Mix elimination*

For the left premise, the definition of principal rule is the same as for (cut). For the right one, a rule is principal if it is (!) or operates with one of the $?^s A$ formulae used in (mix). Eliminating mix is easier, since now principal rules could be only rules for subexponentials, and thus we have to consider a smaller number of cases. Moreover, we can assume that $k \geqslant 2$, since mix with $k = 1$ is actually cut.

Case 1. One of the mix premises is an axiom of the form (ax). Then, as for (cut), the goal coincides with the other premise.

Case 2. The last rule in the derivation of the left premise, $\vdash \Gamma, !^s A^{\perp}$, is non-principal. In this case, we proceed exactly as in the non-principal case for (cut): the mix rule gets propagated to the left, and δ decreases with the same κ.

Case 3. The last rule in the left derivation is principal and the last rule in the right one is non-principal. In this case, the rule on the left is (!), introducing $!^s A^{\perp}$. The interesting situation here is the $(\otimes)$ rule yielding the right premise, $?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k$. The derivation branches, and there are two possibilities: either all instances of $?^s A$ involved in (mix) go to one branch, or they split between branches.

If they do not split, the transformation is again the same as for cut elimination:

$$\dfrac{\vdash \Gamma, !^s A^\perp \qquad \dfrac{\vdash \Phi, E \qquad \vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_i', F, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_i', \Phi, E \otimes F, \Delta_i'', \ldots, ?^s A, \Delta_k}\,(\otimes)}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Phi, E \otimes F, \Delta_i'', \ldots, \Delta_k}\,(\mathrm{mix}),$$

becomes

$$\dfrac{\vdash \Phi, E \qquad \dfrac{\vdash \Gamma, !^s A^\perp \qquad \vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_i', F, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', F, \Delta_i'', \ldots, \Delta_k}\,(\mathrm{mix}).}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Phi, E \otimes F, \Delta_i'', \ldots, \Delta_k}\,(\otimes)$$

The situation with splitting is more involved. In this case, we recall that $\vdash \Gamma, !^s A^\perp$ is obtained by application of !, therefore, $\Gamma = ?^{s_1} C_1, \ldots, ?^{s_n} C_n$, where $s_i \succeq s$ for all i. Hence, $s_i \in \mathcal{C}$, and we can apply the non-local contraction rule for formulae in Γ. Then, we first apply (mix) to both premises of $(\otimes)$, apply $(\otimes)$ and arrive at a sequent with two occurrences of Γ, that are merged by applying the (ncontr) rule n times. An example of such transformation is presented below (the case where the leftmost $?^s A$ goes with E instead of F is symmetric)

$$\dfrac{\vdash \Gamma, !^s A^\perp \qquad \dfrac{\vdash \Delta_j', ?^s A, \ldots, ?^s A, \Delta_i', E \qquad \vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_j'', F, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_j'', \Delta_j', ?^s A, \ldots, ?^s A, \Delta_i', E \otimes F, \Delta_i'', \ldots, ?^s A, \Delta_k}\,(\otimes)}{\vdash \Gamma, \Delta_1, \ldots, \Delta_j'', \Delta_j', \ldots, \Delta_i', E \otimes F, \Delta_i'', \ldots, \Delta_k}\,(\mathrm{mix}),$$

becomes

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma, !^s A^\perp \quad \vdash \Delta_j', ?^s A, \ldots, ?^s A, \Delta_i', E}{\vdash \Delta_j', \Gamma, \ldots, \Delta_i', E}\,(\mathrm{mix}) \quad \dfrac{\vdash \Gamma, !^s A^\perp \quad \vdash ?^s A, \Delta_1, \ldots, ?^s A, \Delta_j'', F, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash \Gamma, \Delta_1, \ldots, \Delta_j'', F, \Delta_i'', \ldots, \Delta_k}\,(\mathrm{mix}).}{\vdash \Gamma, \Delta_1, \ldots, \Delta_j'', \Delta_j', \Gamma, \ldots, \Delta_i', E \otimes F, \Delta_i'', \ldots, \Delta_k}\,(\otimes)}{\vdash \Gamma, \Delta_1, \ldots, \Delta_j'', \Delta_j', \ldots, \Delta_i', E \otimes F, \Delta_i'', \ldots, \Delta_k}\,(\mathrm{ncontr})\ n\ \text{times}$$

Both new applications of (mix) have a smaller δ with the same κ, and we proceed by induction.

All other non-principal cases (easy rules, $(\top)$, and $(\&)$) are handled exactly as in the non-principal case for (cut), only the notation becomes a bit longer.

Case 4. The last rule in both derivations is principal. Then, again, the left premise is (!) introducing $!^s A^\perp$, whence $\Gamma = ?^{s_1} C_1, \ldots, ?^{s_n} C_n$, and we consider subcases on which rule is used on the right.

Subcase 4.1. The last rule is (?). If this rule introduces the leftmost instance of $?^s A$, the transformation is as follows (recall that $k \geqslant 2$):

$$\dfrac{\dfrac{\vdash \Gamma, A^\perp}{\vdash \Gamma, !^s A^\perp}\,(!) \qquad \dfrac{\vdash A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}\,(?)}{\vdash \Gamma, \Delta_1, \Delta_2, \ldots, \Delta_k}\,(\mathrm{mix}),$$

becomes

$$\dfrac{\vdash \Gamma, A^{\perp} \qquad \dfrac{\vdash \Gamma, !^s A^{\perp} \quad \vdash A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}{\vdash A, \Delta_1, \Gamma, \Delta_2, \ldots, \Delta_k}\ (\text{mix}).}{\dfrac{\vdash \Gamma, \Delta_1, \Gamma, \Delta_2, \ldots, \Delta_k}{\vdash \Gamma, \Delta_1, \Delta_2, \ldots, \Delta_k}\ (\text{ncontr}) \text{ several times}}\ (\text{cut})$$

For (mix), κ is the same and δ gets decreased. For (cut), κ gets decreased (A is simpler than $?^s A$), and we do not care for δ (which is uncontrolled). Thus, both cut and mix are eliminable by induction. Finally, $s_i \in \mathcal{C}$ (since $s_i \succeq s$), whence (ncontr) can be applied to formulae from Γ.

If the (?) rule introduces another instance of $?^s A$, the translation is the same, but the second Γ could appear not after Δ_1, but after some other Δ_i.

Subcase 4.2. The last rule is (!). The same as Subcase 4.2 of cut elimination.

Subcase 4.3. The last rule is (ncontr). Our mix rule was specifically designed to subsume (ncontr)

$$\dfrac{\vdash \Gamma, !^s A^{\perp} \qquad \dfrac{\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', ?^s A, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', \Delta_i'', \ldots, ?^s A, \Delta_k}\ (\text{ncontr})}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Delta_i'', \ldots, \Delta_k}\ (\text{mix}),$$

transforms into

$$\dfrac{\vdash \Gamma, !^s A^{\perp} \quad \vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', ?^s A, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Delta_i'', \ldots, \Delta_k}\ (\text{mix}).$$

The δ parameter gets reduced with the same κ.

Subcase 4.4. The last rule is (ex). If this rule did not move the leftmost instance of $?^s A$, then it gets subsumed by (mix) exactly as (ncontr) in the previous subcase. If the leftmost instance of $?^s A$ gets moved, then we recall that $\Gamma = ?^{s_1} C_1, \ldots, ?^{s_n} C_n$ and $s_i \in \mathcal{E}$ for all i by the definition of subexponential signature, since $s_i \succeq s$ and $s \in \mathcal{E}$. This means we can apply the exchange rule for Γ as a whole, and

$$\dfrac{\vdash \Gamma, !^s A^{\perp} \qquad \dfrac{\vdash \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', ?^s A, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', \Delta_i'', \ldots, ?^s A, \Delta_k}\ (\text{ex})}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Delta_i'', \ldots, \Delta_k}\ (\text{mix}),$$

transforms into

$$\dfrac{\dfrac{\vdash \Gamma, !^s A^{\perp} \quad \vdash \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_i', ?^s A, \Delta_i'', \ldots, ?^s A, \Delta_k}{\vdash \Delta_1, \ldots, \Delta_i', \Gamma, \Delta_i'', \ldots, \Delta_k}\ (\text{mix})}{\vdash \Gamma, \Delta_1, \ldots, \Delta_i', \Delta_i'', \ldots, \Delta_k}\ (\text{ex}) \text{ several times.}$$

Here, we first apply (mix) with the same κ and a smaller δ and then move Γ to the correct place by several applications of (ex).

Subcase 4.5. The last rule is (weak). Again, if it introduced an instance of $?^s A$ different from the leftmost one, it is subsumed by (mix). If the leftmost instance gets weakened, then we apply mix to the second $?^s A$ (recall that $k \geqslant 2$, so we do have another instance), and then exchange Γ with Δ_1

$$\dfrac{\vdash \Gamma, !^s A \quad \dfrac{\vdash \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}{\vdash ?^s A, \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}\,(\text{weak})}{\vdash \Gamma, \Delta_1, \Delta_2, \ldots, \Delta_k}\,(\text{mix}),$$

transforms into

$$\dfrac{\dfrac{\vdash \Gamma, !^s A \quad \vdash \Delta_1, ?^s A, \Delta_2, \ldots, ?^s A, \Delta_k}{\vdash \Delta_1, \Gamma, \Delta_2, \ldots, \Delta_k}\,(\text{mix})}{\vdash \Gamma, \Delta_1, \Delta_2, \ldots, \Delta_k}\,(\text{ex})\text{ several times.}$$

Exchange of formulae from Γ is allowed, since, by our definitions, $\mathcal{W} \cap \mathcal{C} \subseteq \mathcal{E}$ and $s_i \succeq s \in \mathcal{W} \cap \mathcal{C}$ (s is in $\mathcal{W}$, since we used (weak), and in $\mathcal{C}$, since we used (mix)). Again, κ is the same and δ gets reduced. □

7. Embedding of SMALC_Σ into SCLL_Σ and cut elimination in SMALC_Σ

In this section, we define an extension to subexponentials of the standard embedding of Lambek formulae into cyclic linear logic. Lambek formula A translates into linear logic formula $\widehat{A}$:

$$\begin{array}{ll}
\widehat{p_i} = p_i & \widehat{\mathbf{1}} = \mathbf{1} \\
\widehat{A \cdot B} = \widehat{A} \otimes \widehat{B} & \widehat{!^s A} = !^s \widehat{A} \\
\widehat{A \setminus B} = \widehat{A}^\perp \parr \widehat{B} & \widehat{A \wedge B} = \widehat{A} \,\&\, \widehat{B} \\
\widehat{B \,/\, A} = \widehat{B} \parr \widehat{A}^\perp & \widehat{A \vee B} = \widehat{A} \oplus \widehat{B}.
\end{array}$$

For convenience, we also recall the definition of negation in SCLL_Σ and present the negative translations (negations of translations) of Lambek formulae

$$\begin{array}{ll}
\widehat{p_i}^\perp = \bar{p}_i & \widehat{\mathbf{1}}^\perp = \perp \\
(\widehat{A \cdot B})^\perp = \widehat{B}^\perp \parr \widehat{A}^\perp & (\widehat{!^s A})^\perp = ?^s \widehat{A}^\perp \\
(\widehat{A \setminus B})^\perp = \widehat{B}^\perp \otimes \widehat{A} & (\widehat{A \wedge B})^\perp = \widehat{A}^\perp \oplus \widehat{B}^\perp \\
(\widehat{B \,/\, A})^\perp = \widehat{A} \otimes \widehat{B}^\perp & (\widehat{A \vee B})^\perp = \widehat{A}^\perp \,\&\, \widehat{B}^\perp.
\end{array}$$

For $\Pi = A_1, \ldots, A_k$, let $\widehat{\Pi}^\perp$ be $\widehat{A}_k^\perp, \ldots, \widehat{A}_1^\perp$ (for left-hand sides of Lambek sequents, we need only the negative translation).

The negative translations are necessary, since in SCLL_Σ all negations are tight: $A^\perp$ is just a metasyntactic shortcut, really one has to propagate the negation downwards to the atoms. On the other side, negation is used in the translation of Lambek's division operations ($\setminus$ and $/$). Thus, positive and negative translations of SMALC_Σ formulae to SCLL_Σ are defined by joint recursion.

Theorem 2. The following statements are equivalent:

1. the sequent $\Pi \to B$ is derivable in SMALC_Σ;
2. the sequent $\Pi \to B$ is derivable in $\mathrm{SMALC}_\Sigma + (\text{cut})$;

3. the sequent $\vdash \widehat{\Pi}^{\perp}, \widehat{B}$ is derivable in $\mathrm{SCLL}_\Sigma + (\mathrm{cut})$;
4. the sequent $\vdash \widehat{\Pi}^{\perp}, \widehat{B}$ is derivable in SCLL_Σ.

This theorem yields both cut elimination for SMALC_Σ and embedding of SMALC_Σ into SCLL_Σ.

Corollary 3. A sequent is derivable in $\mathrm{SMALC}_\Sigma + (\mathrm{cut})$ if and only if it is derivable in SMALC_Σ.

Corollary 4. The sequent $\Pi \to B$ is derivable in SMALC_Σ if and only if the sequent $\vdash \widehat{\Pi}^{\perp}, \widehat{B}$ is derivable in SCLL_Σ.

We prove Theorem 2 by establishing round-robin implications: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$. The last implication, $4 \Rightarrow 1$, is a bit surprising, since the Lambek calculus is in a sense 'intuitionistic,' and CLL is 'classic' (cf. Chaudhuri 2010). However, it becomes possible due to the restricted language used in the Lambek calculus: it includes neither multiplicative disjunction ($\parr$), nor negation, nor existential subexponentials ($?^s$), nor additive constants ($\mathbf{0}$ and $\top$).

In the commutative case, as shown by Schellinx (1991), these are exactly the restrictions under which intuitionistic linear logic is a conservative fragment of classical linear logic. In our non-commutative case, the situation is the same: SMALC_Σ in its restricted language gets conservatively embedded into SCLL_Σ, but extending the language and including some of the forbidden connectives leads to failure of the conservativity claim.

Multiplicative disjunction and negation allow encoding *tertium non datur*, $A \parr A^{\perp}$, which is intuitionistically invalid.

In the implication-only language, there is still a principle that is valid classically, but not intuitionistically, called *Peirce's law* (Peirce 1885): $((X \Rightarrow Y) \Rightarrow X) \Rightarrow X$. Encoding Peirce's law in substructural logic requires explicitly allowing contraction for the rightmost X and weakening for Y, like this: $(x \setminus ?^w y) \setminus x \to ?^c x$, where $w \in \mathcal{W}$ and $c \in \mathcal{C}$. This would be a counter-example for the $4 \Rightarrow 1$ implication; fortunately, formulae of the form $?^s A$ are outside the language of SMALC_Σ. The translation of this substructural form of Peirce's law into cyclic linear logic, $\vdash \bar{x} \otimes (\bar{x} \parr ?^w y), ?^c x$, is derivable in SCLL_Σ with an appropriate substructural signature Σ

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(ax)}
\UnaryInfC{$\vdash x, \bar{x}$}
\RightLabel{(?)}
\UnaryInfC{$\vdash ?^c x, \bar{x}$}
\AxiomC{}
\RightLabel{(ax)}
\UnaryInfC{$\vdash \bar{x}, x$}
\RightLabel{(?)}
\UnaryInfC{$\vdash \bar{x}, ?^c x$}
\RightLabel{(weak)}
\UnaryInfC{$\vdash \bar{x}, ?^w y, ?^c x$}
\RightLabel{($\parr$)}
\UnaryInfC{$\vdash \bar{x} \parr ?^w y, ?^c x$}
\RightLabel{($\otimes$)}
\BinaryInfC{$\vdash ?^c x, \bar{x} \otimes (\bar{x} \parr ?^w y), ?^c x$}
\RightLabel{(ncontr).}
\UnaryInfC{$\vdash \bar{x} \otimes (\bar{x} \parr ?^w y), ?^c x$}
\end{prooftree}
$$

Finally, if one extends the Lambek calculus with the $\mathbf{0}$ constant governed by the following left rule:

$$
\frac{}{\Gamma_1, \mathbf{0}, \Gamma_2 \to C}\ (\mathbf{0} \to)
$$

and no right rule (Lambek 1993), the $4 \Rightarrow 1$ implication (where $\widehat{\mathbf{0}} = \mathbf{0}$) also becomes false. This is established by a non-commutative version of the counter-example by Schellinx (1991)

$$(r \,/\, (\mathbf{0} \setminus q)) \,/\, p, (s \,/\, p) \setminus \mathbf{0} \to r.$$

Since the Lambek calculus with $\mathbf{0}$ still has the cut elimination property (as we do not need subexponentials and additives, one can prove it by simple induction, as in Lambek (1958), one can perform exhaustive proof search and find out that this sequent is not derivable. On the other hand, its translation into cyclic linear logic, $\vdash \top \otimes (s \parr \bar{p}), p \otimes (\top \parr q) \otimes \bar{r}, r$, is derivable in SCLL_Σ

$$\dfrac{\dfrac{}{\vdash \top}\,(\top) \qquad \dfrac{\dfrac{\dfrac{}{\vdash \bar{p}, p}\,(\mathrm{ax}) \qquad \dfrac{\dfrac{\dfrac{}{\vdash s, \top, q}\,(\top)}{\vdash s, \top \parr q}\,(\parr) \qquad \dfrac{}{\vdash \bar{r}, r}\,(\mathrm{ax})}{\vdash s, (\top \parr q) \otimes \bar{r}, r}\,(\otimes)}{\vdash s, \bar{p}, p \otimes ((\top \parr q) \otimes \bar{r}), r}\,(\otimes)}{\vdash s \parr \bar{p}, p \otimes ((\top \parr q) \otimes \bar{r}), r}\,(\parr)}{\vdash \top \otimes (s \parr \bar{p}), p \otimes ((\top \parr q) \otimes \bar{r}), r}\,(\otimes).$$

Our proof of the $4 \Rightarrow 1$ implication for the restricted language is essentially based on the ideas of Schellinx (1991). We show that if a sequent of the form $\vdash \widehat{\Gamma}^\perp, \widehat{B}$ is derivable in SCLL_Σ, then in each sequent in the derivation there is exactly one formula of the form $\widehat{C}$, and all others are of the form $\widehat{C}^\perp$ (see Lemma 6). This means that all sequents in the SCLL_Σ-derivation are actually translations of Lambek sequents, and the derivation as a whole can be mapped onto a derivation in SMALC_Σ.

This technical lemma is proved using an extension of the $\natural$-counter by Pentus (1998) to formulae of SCLL_Σ without $\mathbf{0}$ and $\top$, but possibly using additive and subexponential connectives (Pentus considers only the multiplicative fragment of cyclic linear logic).

The idea behind the $\natural$-counter is as follows. When analysing the derivation of $\vdash \widehat{\Pi}^\perp, \widehat{B}$, in particular, when considering the branching $(\otimes)$ rule, one needs a quick and easy method of understanding whether a SCLL_Σ-formula is of the form $\widehat{A}$ or of the form $\widehat{A}^\perp$. The $\natural$ counter exactly does the job: $\natural(E)$ is 0 if $E = \widehat{A}$ and 1 if $E = \widehat{A}^\perp$ (see Lemma 5 below). For formulae not of the form $\widehat{A}$ or $\widehat{A}^\perp$ (i.e., neither positive, nor negative translations of SMALC_Σ-formulae), the value of $\natural$ can be arbitrary – but such formulae never appear in the (cut-free) derivation of $\vdash \widehat{\Pi}^\perp, \widehat{B}$. Finally, by Lemma 6, in any sequent in this derivation for exactly one formula $\natural$ has value 0 (and for all other formulae, it is 1), which guarantees that this sequent is actually a translation of a SMALC_Σ-sequent.

Notice that the $\natural$-counter cannot be extended to $\mathbf{0}$ constant preserving the properties we need in our proof (due to the counter-example shown above).

The $\natural$-counter is defined recursively as follows:

$$\begin{array}{ll}
\natural(p) = 0 & \natural(A \parr B) = \natural(A) + \natural(B) - 1 \\
\natural(\bar{p}) = 1 & \natural(A \otimes B) = \natural(A) + \natural(B) \\
\natural(\mathbf{1}) = 0 & \natural(A \oplus B) = \natural(A \,\&\, B) = \natural(A) \\
\natural(\perp) = 1 & \natural(?^s A) = \natural(!^s A) = \natural(A).
\end{array}$$

If $\Gamma = E_1, \ldots, E_k$, then let $\natural(\Gamma) = \natural(E_1) + \ldots + \natural(E_k)$.

Then, we establish the following properties of the $\natural$-counter.

Lemma 5.

1. $\natural(\widehat{A}) = 0$;
2. $\natural(\widehat{A}^\perp) = 1$;
3. if $A \oplus B$ is of the form $\widehat{C}$ or $\widehat{C}^\perp$, then $\natural(A) = \natural(B) = \natural(A \oplus B)$;
4. if each A_i for $i = 1, \ldots, n$ is of the form $\widehat{C}$ or $\widehat{C}^\perp$ and the sequent $\vdash A_1, \ldots, A_n$ is derivable in SCLL_Σ, then $\natural(A_1) + \ldots + \natural(A_n) = n - 1$;

Proof. 1. Induction on the structure of A

$$\begin{aligned}
&\natural(\widehat{p}) = \natural(p) = 0\\
&\natural(\widehat{\mathbf{1}}) = \natural(\mathbf{1}) = 0\\
&\natural(\widehat{A \cdot B}) = \natural(\widehat{A} \otimes \widehat{B}) = \natural(\widehat{A}) + \natural(\widehat{B}) = 0 + 0 = 0\\
&\natural(\widehat{A \setminus B}) = \natural(\widehat{A}^\perp \parr \widehat{B}) = \natural(\widehat{A}^\perp) + \natural(\widehat{B}) - 1 = 1 + 0 - 1 = 0\\
&\natural(\widehat{B / A}) = \natural(\widehat{B} \parr \widehat{A}^\perp) = \natural(\widehat{B}) + \natural(\widehat{A}^\perp) - 1 = 0 + 1 - 1 = 0\\
&\natural(\widehat{A \vee B}) = \natural(\widehat{A} \oplus \widehat{B}) = \natural(\widehat{A}) = 0\\
&\natural(\widehat{A \wedge B}) = \natural(\widehat{A} \mathbin{\&} \widehat{B}) = \natural(\widehat{A}) = 0\\
&\natural(\widehat{!^s A}) = \natural(!^s \widehat{A}) = \natural(\widehat{A}) = 0.
\end{aligned}$$

2. Induction on the structure of A

$$\begin{aligned}
&\natural(\widehat{p}^\perp) = \natural(\bar{p}) = 1\\
&\natural(\widehat{\mathbf{1}}^\perp) = \natural(\perp) = 1\\
&\natural((\widehat{A \cdot B})^\perp) = \natural(\widehat{B}^\perp \parr \widehat{A}^\perp) = \natural(\widehat{B}^\perp) + \natural(\widehat{A}^\perp) - 1 = 1 + 1 - 1 = 1\\
&\natural((\widehat{A \setminus B})^\perp) = \natural(\widehat{B}^\perp \otimes \widehat{A}) = \natural(\widehat{B}^\perp) + \natural(\widehat{A}) = 1 + 0 = 1\\
&\natural((\widehat{B / A})^\perp) = \natural(\widehat{A} \otimes \widehat{B}^\perp) = \natural(\widehat{A}) + \natural(\widehat{B}^\perp) = 0 + 1 = 1\\
&\natural((\widehat{A \vee B})^\perp) = \natural(\widehat{A}^\perp \mathbin{\&} \widehat{B}^\perp) = \natural(\widehat{A}^\perp) = 1\\
&\natural((\widehat{A \wedge B})^\perp) = \natural(\widehat{A}^\perp \oplus \widehat{B}^\perp) = \natural(\widehat{A}^\perp) = 1\\
&\natural((\widehat{!^s A})^\perp) = \natural(?^s\, \widehat{A}^\perp) = \natural(\widehat{A}^\perp) = 1.
\end{aligned}$$

3. If $A \oplus B = \widehat{C}$, then $C = C_1 \vee C_2$, $A = \widehat{C}_1$, $B = \widehat{C}_2$, and $\natural(A) = \natural(B) = 0$.
If $A \oplus B = \widehat{C}^\perp$, then $C = C_1 \wedge C_2$, $A = \widehat{C}_1^\perp$, $B = \widehat{C}_2^\perp$, and $\natural(A) = \natural(B) = 1$.
4. Induction on the derivation in SCLL_Σ
Case 1, (ax): $\natural(\widehat{A}) + \natural(\widehat{A}^\perp) = 0 + 1 = 1 = 2 - 1$, $n = 2$.
Case 2, ($\otimes$). Let Γ include n_1 formulae and Δ include n_2 formulae. Then, by induction hypothesis, $\natural(\Gamma) + \natural(A) = n_1 + 1 - 1 = n_1$ and $\natural(B) + \natural(\Delta) = n_2 + 1 - 1 = n_2$. Therefore, $\natural(\Gamma) + \natural(A \otimes B) + \natural(\Delta) = \natural(\Gamma) + \natural(A) + \natural(B) + \natural(\Delta) = n_1 + n_2 = (n_1 + n_2 + 1) - 1 = n - 1$.

Case 3, ($\invamp$). Let Γ include n_1 formulae. Then, by induction hypothesis, $\natural(A) + \natural(B) + \natural(\Gamma) = n_1 + 2 - 1 = n_1 + 1$. Therefore, $\natural(A \invamp B) + \natural(\Gamma) = \natural(A) + \natural(B) - 1 + \natural(\Gamma) = (n_1 + 1) - 1 = n - 1$.

Case 4, ($\&$). Since $\natural(A_1 \& A_2) = \natural(A_1)$, we have $\natural(A_1 \& A_2) + \natural(\Gamma) = \natural(A_1) + \natural(\Gamma)$, which is $n - 1$ by induction hypothesis.

Case 5, ($\oplus$). By Statement 3 of this Lemma, since $A_1 \& A_2$ is of the form $\widehat{C}$ or $\widehat{C}^\perp$, we have $\natural(A_1 \& A_2) = \natural(A_i)$ for both $i = 1$ and $i = 2$. Thus, $\natural(A_1 \oplus A_2) + \natural(\Gamma) = \natural(A_i) + \natural(\Gamma)$, which is $n - 1$ by induction hypothesis.

Case 6, ($\mathbf{1}$): $\natural(\mathbf{1}) = 0 = 1 - 1$, $n = 1$.

Case 7, ($\perp$). In this case, Γ contains $n - 1$ formulae, by induction hypothesis $\natural(\Gamma) = (n - 1) - 1$, and $\natural(\perp) + \natural(\Gamma) = 1 + (n - 1) - 1 = n - 1$.

Case 8, ($\top$). Impossible, since $\top$ is neither of the form $\widehat{C}$, nor of the form $\widehat{C}^\perp$.

Case 9, (!). Adding $!^s$ does not alter the $\natural$-counter.

Case 10, (?). Adding $?^s$ does not alter the $\natural$-counter.

Case 11, (weak). The new formula $?^s A$ could not be of the form $\widehat{C}$, therefore, it is of the form $\widehat{C}^\perp$. Hence, by Statement 2 of this Lemma, $\natural(?^s A) = 1$, and $\natural(?^s A) + \natural(\Gamma) = 1 + (n - 1) - 1 = n - 1$.

Case 12, (ncontr). Again, $\natural(?^s A) = 1$, and $\natural(?^s A) + \natural(\Gamma) + \natural(\Delta) = \natural(?^s A) + \natural(\Gamma) + \natural(?^s A) + \natural(\Delta) - 1 = ((n + 1) - 1) - 1 = n - 1$.

Case 13, (ex). In this case, $\natural(?^s A) + \natural(\Gamma) + \natural(\Delta) = \natural(\Gamma) + \natural(?^s A) + \natural(\Delta) = n - 1$ by induction hypothesis. $\square$

Lemma 6. If each A_i for $i = 1, \ldots, n$ is of the form $\widehat{C}$ or $\widehat{C}^\perp$ and the sequent $\vdash A_1, \ldots, A_n$ is derivable in SCLL_Σ, then exactly one of $A_1, \ldots, A_n$ is of the form $\widehat{C}$, and all other are of the form $\widehat{C}^\perp$.

Proof. Let our sequent include k formulae of the form $\widehat{C}$ and $(n - k)$ formulae of the form $\widehat{C}^\perp$. Then, on one hand, $\natural(A_1) + \ldots + \natural(A_n) = n - 1$ by Statement 4 of the previous Lemma. On the other hand, by Statements 1 and 2, $\natural(A_1) + \ldots + \natural(A_n) = k \cdot 0 + (n - k) \cdot 1 = n - k$. Thus, $n - k = n - 1$, whence $k = 1$. $\square$

Now, we are ready to prove Theorem 2.

Proof of Theorem 2. $\boxed{4 \Rightarrow 1}$ We proceed by induction on the derivation of $\vdash \widehat{\Pi}^\perp, \widehat{B}$ in SCLL_Σ. In our notation, we shall always put the formula of the form $\widehat{B}$ into the rightmost position (and use the cyclically transformed versions of the rules, as shown above, see Section 5).

The most interesting case is the ($\otimes$) rule. If it yields the rightmost formula, $\widehat{B} = \widehat{E \cdot F} = \widehat{E} \otimes \widehat{F}$, then the ($\otimes$) rule application transforms into ($\to \cdot$)

$$\frac{\vdash \widehat{\Delta}^\perp, \widehat{E} \quad \vdash \widehat{\Gamma}^\perp, \widehat{F}}{\vdash \widehat{\Gamma}^\perp, \widehat{\Delta}^\perp, \widehat{E} \otimes \widehat{F}} \, (\otimes_2) \quad \rightsquigarrow \quad \frac{\Delta \to E \quad \Gamma \to F}{\Gamma, \Delta \to E \cdot F} \, (\to \cdot).$$

If the ($\otimes$) rule yields a formula of the form $\widehat{A}^\perp$ from $\widehat{\Pi}^\perp$, there are two possibilities: $\widehat{A}^\perp$ is either $\widehat{E} \otimes \widehat{F}^\perp = \widehat{(F \,/\, E)}^\perp$ or $\widehat{F}^\perp \otimes \widehat{E} = \widehat{(E \setminus F)}^\perp$. Also, one can apply the ($\otimes$) either in the ($\otimes_1$) or in the ($\otimes_2$) form. This leads to four possible cases. Two of them are handled as follows:

$$\dfrac{\vdash \widehat{\Delta}^{\perp}, \widehat{E} \quad \vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{\Delta}^{\perp}, \widehat{E} \otimes \widehat{F}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\otimes_2) \quad \leadsto \quad \dfrac{\Delta \to E \quad \Gamma_2, F, \Gamma_1 \to B}{\Gamma_2, F \,/\, E, \Delta, \Gamma_1 \to B}\,(/ \to)$$

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B} \quad \vdash \widehat{E}, \widehat{\Delta}^{\perp}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}^{\perp} \otimes \widehat{E}, \widehat{\Delta}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\otimes_1) \quad \leadsto \quad \dfrac{\Delta \to E \quad \Gamma_2, F, \Gamma_1 \to B}{\Gamma_2, E \setminus F, \Delta, \Gamma_1 \to B}\,(\setminus \to).$$

In the other two cases, we have the following:

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{E}, \widehat{\Gamma}_2^{\perp}, \widehat{B} \quad \vdash \widehat{F}^{\perp}, \widehat{\Delta}^{\perp}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{E} \otimes \widehat{F}^{\perp}, \widehat{\Delta}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\otimes_1) \quad \text{or} \quad \dfrac{\vdash \widehat{\Delta}^{\perp}, \widehat{F}^{\perp} \quad \vdash \widehat{\Gamma}_1^{\perp}, \widehat{E}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{\Delta}^{\perp}, \widehat{F}^{\perp} \otimes \widehat{E}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\otimes_2).$$

These situations violate Lemma 6, since in $\vdash \widehat{\Gamma}_1^{\perp}, \widehat{E}, \widehat{\Gamma}_2^{\perp}, \widehat{B}$ there are two formulae of the form $\widehat{C}$, and therefore this premise could not be derivable in SCLL_{Σ}. Thus, these two cases are impossible.

All other rules are translated straightforwardly

$$\dfrac{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}^{\perp}, \widehat{F}}{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}^{\perp} \parr \widehat{F}}\,(\parr) \quad \leadsto \quad \dfrac{E, \Gamma \to F}{\Gamma \to E \setminus F}\,(\to \setminus)$$

$$\dfrac{\vdash \widehat{E}^{\perp}, \widehat{\Gamma}^{\perp}, \widehat{F}}{\vdash \widehat{\Gamma}^{\perp}, \widehat{F} \parr \widehat{E}^{\perp}}\,(\parr) \quad \leadsto \quad \dfrac{\Gamma, E \to F}{\Gamma \to F \,/\, E}\,(\to \setminus)$$

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{E}^{\perp}, \widehat{F}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{E}^{\perp} \parr \widehat{F}^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\parr) \quad \leadsto \quad \dfrac{\Gamma_2, F, E, \Gamma_1 \to B}{\Gamma_2, F \cdot E, \Gamma_1 \to B}\,(\cdot \to)$$

$$\dfrac{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}_1 \quad \vdash \widehat{\Gamma}^{\perp}, \widehat{E}_2}{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}_1 \,\&\, \widehat{E}_2}\,(\&) \quad \leadsto \quad \dfrac{\Gamma \to E_1 \quad \Gamma \to E_2}{\Gamma \to E_1 \wedge E_2}\,(\to \wedge)$$

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}_1^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B} \quad \vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}_2^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}_1^{\perp} \,\&\, \widehat{F}_2^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\&) \quad \leadsto \quad \dfrac{\Gamma_2, F_1, \Gamma_1 \to B \quad \Gamma_2, F_2, \Gamma_1 \to B}{\Gamma_2, F_1 \vee F_2, \Gamma_1 \to B}\,(\vee \to)$$

$$\dfrac{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}_i}{\vdash \widehat{\Gamma}^{\perp}, \widehat{E}_1 \oplus \widehat{E}_2}\,(\oplus) \quad \leadsto \quad \dfrac{\Gamma \to E_i}{\Gamma \to E_1 \vee E_2}\,(\to \vee)$$

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}_i^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{F}_1^{\perp} \oplus \widehat{F}_2^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\oplus) \quad \leadsto \quad \dfrac{\Gamma_2, F_i, \Gamma_1 \to B}{\Gamma_2, F_1 \wedge F_2, \Gamma_1 \to B}\,(\wedge \to)$$

$$\dfrac{}{\vdash \mathbf{1}}\,(\mathbf{1}) \quad \leadsto \quad \dfrac{}{\to \mathbf{1}}\,(\to \mathbf{1})$$

$$\dfrac{\vdash \widehat{\Gamma}_1^{\perp}, \widehat{\Gamma}_2^{\perp}, \widehat{B}}{\vdash \widehat{\Gamma}_1^{\perp}, \bot, \widehat{\Gamma}_2^{\perp}, \widehat{B}}\,(\bot) \quad \leadsto \quad \dfrac{\Gamma_2, \Gamma_1 \to B}{\Gamma_2, \mathbf{1}, \Gamma_1 \to B}\,(\mathbf{1} \to).$$

The $(\top)$ rule cannot be applied, since $\top$ is neither of the form $\widehat{C}$, nor of the form $\widehat{C}^{\perp}$.

$$\dfrac{\vdash ?^{s_1}\widehat{A}_1^{\perp}, \ldots, ?^{s_n}\widehat{A}_n^{\perp}, \widehat{B}}{\vdash ?^{s_1}\widehat{A}_1^{\perp}, \ldots, ?^{s_n}\widehat{A}_n^{\perp}, !^{s}\widehat{B}}\,(!) \quad \leadsto \quad \dfrac{!^{s_n}A_n, \ldots, !^{s_1}A_1 \to B}{!^{s_n}A_n, \ldots, !^{s_1}A_1 \to !^{s}B}\,(\to !).$$

$\boxed{1 \Rightarrow 2}$ Trivial: allowing the cut rule does not invalidate cut-free derivations.

$\boxed{2 \Rightarrow 3}$ Straightforward induction on the derivation in $\mathrm{SMALC}_\Sigma + (\mathrm{cut})$. The cut rule is translated as follows:

$$\frac{\Pi \to A \quad \Gamma_1, A, \Gamma_2 \to C}{\Gamma_1, \Pi, \Gamma_2 \to C}\ (\mathrm{cut}) \qquad \rightsquigarrow \qquad \frac{\vdash \widehat{\Pi}^\perp, \widehat{A} \quad \vdash \widehat{A}^\perp, \widehat{\Gamma}_1^\perp, \widehat{C}, \widehat{\Gamma}_2^\perp}{\vdash \widehat{\Pi}^\perp, \widehat{\Gamma}_1^\perp, \widehat{C}, \widehat{\Gamma}_2^\perp}\ (\mathrm{cut}).$$

For translating other rules, one simply reverses arrows in the proof of the $4 \Rightarrow 1$ implication (see above).

$\boxed{3 \Rightarrow 4}$ Follows from cut elimination in SCLL_Σ (Theorem 1). □

8. Cut vs. contraction

The contraction rules of SMALC_Σ and SCLL_Σ are *non-local*, i.e., they can take formulae for contraction from distant places of the sequent. In the presence of exchange (permutation) rules, non-local contraction rules are equivalent to *local* ones, that contract two neighbour copies of the same formula marked with an appropriate subexponential

$$\frac{\Gamma_1, !^s A, !^s A, \Gamma_2 \to C}{\Gamma_1, !^s A, \Gamma_2 \to C}\ (\mathrm{contr}),\ \text{for } \mathrm{SMALC}_\Sigma; \qquad \frac{\vdash ?^s A, ?^s A, \Gamma}{\vdash ?^s A, \Gamma}\ (\mathrm{contr}),\ \text{for } \mathrm{SCLL}_\Sigma.$$

If the subexponential does not allow exchange ($s \in \mathcal{C}_{(\mathrm{local})} - \mathcal{E}$), however, cut elimination with the local contraction rule *fails.*

Theorem 7. The extension of the Lambek calculus with a unary connective ! axiomatised by rules $(! \to)$, $(\to !)$, (contr), and optionally, (weak) does not admit (cut).

Proof. One can take the following sequent as a counter-example:

$$r \,/\, q, !p, !(p \setminus q), q \setminus s \to r \cdot s.$$

This sequent has a proof with (cut)

$$\dfrac{\dfrac{\dfrac{\dfrac{p \to p \quad q \to q}{p, p \setminus q \to q}\ (\setminus \to)}{!p, !(p \setminus q) \to q}\ (! \to)\ \text{twice}}{!p, !(p \setminus q) \to !q}\ (\to !) \qquad \dfrac{\dfrac{\dfrac{q \to q \quad \dfrac{q \to q \quad \dfrac{r \to r \quad s \to s}{r, s \to r \cdot s}\ (\to \cdot)}{r, q, q \setminus s \to r \cdot s}\ (\setminus \to)}{r \,/\, q, q, q, q \setminus s \to r \cdot s}\ (/ \to)}{r \,/\, q, !q, !q, q \setminus s \to r \cdot s}\ (! \to)\ \text{twice}}{r \,/\, q, !q, q \setminus s \to r \cdot s}\ (\mathrm{contr})}{r \,/\, q, !p, !(p \setminus q), q \setminus s \to r \cdot s}\ (\mathrm{cut}),$$

(here, we just replace $!p, !(p \setminus q)$ with $!q$ and use it twice, by contraction), but has no cut-free proof. In order to verify the latter, we notice that due to subformula and polarity properties, the only rules that can be applied are $(/ \to)$, $(\setminus \to)$, $(\to \cdot)$, $(! \to)$ and (contr). Moreover, since ! appears only on the top level, the rules operating ! can be moved to the very bottom of the proof (this is actually a small *focusing* instance here). These rules would be applied to a (pure Lambek) sequent of the form

$$r \,/\, q, p, \ldots, p, (p \setminus q), \ldots, (p \setminus q), q \setminus s \to r \cdot s,$$

but an easy proof search attempt shows that none of these sequents is derivable in the Lambek calculus. □

This failure of cut elimination of the calculus with (contr) motivates the usage of the non-local version of contraction, ($\mathrm{ncontr}_{1,2}$). A crucial point in the cut elimination procedure was the fact that if $\Pi \to {!A}$ was obtained by application of $(\to !)$, then all structural rules that can be applied to $!A$, can be also applied to Π *as a whole*. Local contraction does not satisfy this condition: e.g., if $\Pi = {!B_1}, {!B_2}$, then $\Pi, \Pi = {!B_1}, {!B_2}, {!B_1}, {!B_2}$ cannot be contracted to ${!B_1}, {!B_2} = \Pi$, even if ! allows local contraction. Non-local contraction, however, does the job. Thus, with non-local contraction the sequent used in the proof of Theorem 7 obtains a cut-free proof

$$
\begin{prooftree}
\AxiomC{$p \to p$}
\AxiomC{$p \to p$}
\AxiomC{$q \to q$}
\AxiomC{$q \to q$}
\AxiomC{$r \to r$}
\AxiomC{$s \to s$}
\RightLabel{$(\to \cdot)$}
\BinaryInfC{$r, s \to r \cdot s$}
\RightLabel{$(\backslash \to)$}
\BinaryInfC{$r, q, q \backslash s \to r \cdot s$}
\RightLabel{$(/ \to)$}
\BinaryInfC{$r / q, q, q, q \backslash s \to r \cdot s$}
\doubleLine
\RightLabel{$(\backslash \to)$ twice}
\TrinaryInfC{$r / q, p, p \backslash q, p, p \backslash q, q \backslash s \to r \cdot s$}
\doubleLine
\RightLabel{$(! \to)$ 4 times}
\UnaryInfC{$r / q, !p, !(p \backslash q), !p, !(p \backslash q), q \backslash s \to r \cdot s$}
\RightLabel{(ncontr_2)}
\UnaryInfC{$r / q, !p, !p, !(p \backslash q), q \backslash s \to r \cdot s$}
\RightLabel{(contr).}
\UnaryInfC{$r / q, !p, !(p \backslash q), q \backslash s \to r \cdot s$}
\end{prooftree}
$$

This counter-example can also be translated into SCLL_Σ using the embedding of SLC_Σ into SCLL_Σ (see Section 7).

9. Undecidability of SLC_Σ

In the view of Corollary 4, we prove *lower* complexity bounds for fragments of SLC_Σ and *upper* ones for fragments of SCLL_Σ.

Theorem 8. If $\mathcal{C} \neq \varnothing$ (i.e., at least one subexponential allows the non-local contraction rule), then the derivability problem in $\mathrm{SLC}_\Sigma^{\mathbf{1}}$ is undecidable.

The proof follows the line presented in Kanovich et al. (2017), using ideas from Lincoln et al. (1992), Kanazawa (1999) and de Groote (2005). In the latter three papers, undecidability is established for non-commutative propositional linear logic systems equipped with an exponential that allows all structural rules (contraction, weakening and exchange), as ELC defined below. The difference of our setting is that here only contraction is guaranteed and exchange and weakening are optional.

The undecidability proof is based on encoding word rewriting (semi-Thue) systems (Thue 1914). A word rewriting system over alphabet $\mathfrak{A}$ is a finite set P of pairs of words over $\mathfrak{A}$. Elements of P are called *rewriting rules* and are applied as follows: if $\langle \alpha, \beta \rangle \in P$, then $\eta\, \alpha\, \theta \Rightarrow \eta\, \beta\, \theta$ for arbitrary (possibly empty) words η and θ over $\mathfrak{A}$. The relation $\Rightarrow^*$ is the reflexive transitive closure of $\Rightarrow$.

The following classical result appears in works of Markov (1947) and Post (1947).

Theorem 9. There exists a word rewriting system P such that the set $\{\langle \gamma, \delta \rangle \mid \gamma \Rightarrow^* \delta\}$ is r.e.-complete (and therefore undecidable) (Markov 1947; Post 1947).

In our encoding, we actually need the weakening rule. However, our subexponential does not necessarily enjoy it. To simulate weakening, we use the unit constant: actually, the $(\mathbf{1} \rightarrow)$ rule is weakening, but for $\mathbf{1}$ rather than $!A$.

Let P be the word rewriting system from Theorem 9 and consider all elements of $\mathfrak{A}$ as variables of the Lambek calculus. We convert rewriting rules of P into Lambek formulae in the following way:

$$\mathcal{B} = \{(u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m) \mid \langle u_1 \ldots u_k, v_1 \ldots v_m \rangle \in P\}.$$

If $\mathcal{B} = \{B_1, \ldots, B_n\}$ (we can take any ordering of $\mathcal{B}$), let

$$\Phi = \mathbf{1} \,/\, !^s B_1, !^s B_1, \ldots, \mathbf{1} \,/\, !^s B_n, !^s B_n.$$

Finally, we consider a *theory* (finite set of sequents) $\mathcal{T}$ associated with P

$$\mathcal{T} = \{v_1, \ldots, v_m \rightarrow u_1 \cdot \ldots \cdot u_k \mid \langle u_1 \ldots u_k, v_1 \ldots v_m \rangle \in P\}.$$

When talking about *derivability from theory* $\mathcal{T}$, we use the rules of the original Lambek calculus, including cut.

Now let $s \in \mathcal{C}$ be the label of the subexponential that allows non-local contraction (and, possibly, also weakening and/or exchange). We also consider, as a technical tool, the extension of the Lambek calculus with an exponential modality ! that allows all three structural rules, contraction, weakening and exchange. We denote this auxiliary calculus by $\mathrm{ELC}^{\mathbf{1}}$.

In our framework, $\mathrm{ELC}^{\mathbf{1}}$ is $\mathrm{SLC}^{\mathbf{1}}_{\Sigma_0}$ with a trivial subexponential signature $\Sigma_0 = \langle \mathcal{I}_0, \preceq_0, \mathcal{W}_0, \mathcal{C}_0, \mathcal{E}_0 \rangle$, where $\mathcal{I}_0 = \mathcal{W}_0 = \mathcal{C}_0 = \mathcal{E}_0 = \{s_0\}$, $\preceq_0$ is trivial, and $!^{s_0}$ is denoted by !. Thus, $\mathrm{ELC}^{\mathbf{1}}$ enjoys all proof-theoretical properties of $\mathrm{SLC}^{\mathbf{1}}_{\Sigma}$, in particular, cut elimination (Corollary 3).

For $\mathcal{B} = \{B_1, \ldots, B_n\}$, let $\Gamma = !B_1, \ldots, !B_n$.

Lemma 10. Let $\gamma = a_1 \ldots a_l$ and $\delta = b_1 \ldots b_k$ be arbitrary words over $\mathfrak{A}$. Then, the following are equivalent:

1. $\gamma \Rightarrow^* \delta$;
2. the sequent $\Phi, b_1, \ldots, b_k \rightarrow a_1 \cdot \ldots \cdot a_l$ is derivable in $\mathrm{SLC}^{\mathbf{1}}_{\Sigma}$;
3. the sequent $\Gamma, b_1, \ldots, b_k \rightarrow a_1 \cdot \ldots \cdot a_l$ is derivable in $\mathrm{ELC}^{\mathbf{1}}$;
4. the sequent $b_1, \ldots, b_k \rightarrow a_1 \cdot \ldots \cdot a_l$ is derivable from $\mathcal{T}$.

Proof. $\boxed{1 \Rightarrow 2}$ Proceed by induction on $\Rightarrow^*$. The base case ($\gamma \Rightarrow^* \gamma$) is handled as follows:

$$\dfrac{!^s B_1 \rightarrow !^s B_1 \quad \ldots \quad !^s B_n \rightarrow !^s B_n \qquad \dfrac{\dfrac{a_1 \rightarrow a_1 \quad \ldots \quad a_m \rightarrow a_l}{a_1 \ldots a_m \rightarrow a_1 \cdot \ldots \cdot a_l}\,(\rightarrow \cdot)\ (l-1) \text{ times}}{\mathbf{1}, \ldots, \mathbf{1}, a_1, \ldots, a_l \rightarrow a_1 \cdot \ldots \cdot a_l}\,(\mathbf{1} \rightarrow)\ n \text{ times}}{\mathbf{1} \,/\, !^s B_1, !^s B_1, \ldots, \mathbf{1} \,/\, !^s B_n, !^s B_n, a_1, \ldots, a_l \rightarrow a_1 \cdot \ldots \cdot a_l}\,(/ \rightarrow)\ n \text{ times.}$$

For the induction step, consider the last step of $\Rightarrow^*$

$$\gamma \Rightarrow^* \eta\, u_1 \dots u_k\, \theta \Rightarrow \eta\, v_1 \dots v_m\, \theta.$$

Then, since $!^s((u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m))$ is in Φ and $s \in \mathcal{C}$, we enjoy the following derivation:

$$\dfrac{\dfrac{\dfrac{v_1 \to v_1 \quad \dots \quad v_m \to v_m}{v_1, \dots, v_m \to v_1 \cdot \ldots \cdot v_m}\ (\to \cdot)\ (m-1) \text{ times} \qquad \dfrac{\Phi, \eta, u_1, \dots, u_k, \theta \to a_1 \cdot \ldots \cdot a_l}{\Phi, \eta, u_1 \cdot \ldots \cdot u_k, \theta \to a_1 \cdot \ldots \cdot a_l}\ (\cdot \to)\ (k-1) \text{ times}}{\dfrac{\Phi, \eta, (u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m), v_1, \dots, v_m, \theta \to a_1 \cdot \ldots \cdot a_l}{\Phi, \eta, !^s((u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m)), v_1, \dots, v_m, \theta \to a_1 \cdot \ldots \cdot a_l}\ (! \to)}\ (/ \to)}{\Phi, \eta, v_1, \dots, v_m, \theta \to a_1 \cdot \ldots \cdot a_l}\ (\mathrm{ncontr}_1).$$

The sequent $\Phi, \eta, u_1, \dots, u_k, \theta \to a_1 \cdot \ldots \cdot a_l$ is derivable by induction hypothesis.

$\boxed{2 \Rightarrow 3}$ For each formula $B_i \in \mathcal{B}$ the sequent $\to \mathbf{1} / !B_i$ is derivable in $\mathrm{ELC}^{\mathbf{1}}$ using the weakening rule

$$\dfrac{\dfrac{\to \mathbf{1}}{!B_i \to \mathbf{1}}\ (\mathrm{weak})}{\to \mathbf{1} / !B_i}\ (\to /).$$

Then, we notice that, since $!$ in $\mathrm{ELC}^{\mathbf{1}}$ obeys all the rules for $!^s$ in $\mathrm{SLC}^{\mathbf{1}}_{\Sigma}$, the sequent $\Phi', b_1, \dots, b_k \to a_1 \cdot \ldots \cdot a_l$, where Φ' is the result of replacing $!^s$ by $!$ in Φ, is derivable in $\mathrm{ELC}^{\mathbf{1}}$. Then, we apply (cut) to remove formulae of the form $\mathbf{1} / !B_i$ from Φ'. This transforms Φ' into Γ and yields derivability of $\Gamma, b_1, \dots, b_k \to a_1 \cdot \ldots \cdot a_l$ in $\mathrm{ELC}^{\mathbf{1}}$.

$\boxed{3 \Rightarrow 4}$ Consider the cut-free derivation of $\Gamma, b_1, \dots, b_k \to a_1 \cdot \ldots \cdot a_l$ (as shown above, $\mathrm{ELC}^{\mathbf{1}}$ enjoys cut elimination). Remove all formulae of the form $!E$ from the left-hand sides of the sequents in this derivation. This transformation does not affect rules not operating with !, they remain valid. Applications of structural rules ($(\mathrm{ncontr}_{1,2})$, $(\mathrm{ex}_{1,2})$, (weak)) do not alter the sequent. The only non-trivial case is $(! \to)$. Since all formulae of the form $!E$ come from Γ (due to the subformula property of the cut-free derivation), the only possible case is the following one:

$$\dfrac{\Delta_1, (u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m), \Delta_2 \to C}{\Delta_1, \Delta_2 \to C},$$

$((u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m)$ transforms into an invisible $!((u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m)))$. This application is simulated using an extra axiom from the theory $\mathcal{T}$ that we are allowed to use

$$\dfrac{\dfrac{\dfrac{v_1, \dots, v_m \to u_1 \cdot \ldots \cdot u_k}{v_1 \cdot \ldots \cdot v_m \to u_1 \cdot \ldots u_k}\ (\cdot \to)\ (k-1) \text{ times}}{\to (u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m)}\ (\to /) \qquad \Delta_1, (u_1 \cdot \ldots \cdot u_k) / (v_1 \cdot \ldots \cdot v_m), \Delta_2 \to C}{\Delta_1, \Delta_2 \to C}\ (\mathrm{cut}).$$

The sequent $v_1, \dots, v_m \to u_1 \cdot \ldots \cdot u_k$ belongs to $\mathcal{T}$.

$\boxed{4 \Rightarrow 1}$ Derivations from $\mathcal{T}$ essentially need the cut rule. However, if one tries to apply the standard cut elimination procedure, all the cuts move directly to new axioms from $\mathcal{T}$ (this procedure is called *cut normalisation*). This yields a weak form of subformula property: any formula appearing in a normalised derivation is a subformula either of $\mathcal{T}$, or of the goal sequent. Since both $\mathcal{T}$ and $b_1, \dots, b_k \to a_1 \cdot \ldots \cdot a_l$ include only variables and the product operation, $\cdot$, rules for other connectives are never applied in the normalised derivation. For simplicity, we omit parentheses and the '$\cdot$' symbols, and the rules get formulated in the following way:

$$\frac{\beta_1 \to \alpha_1 \quad \beta_2 \to \alpha_2}{\beta_1\beta_2 \to \alpha_1\alpha_2}\ (\to \cdot) \qquad \frac{\beta \to \alpha \quad \eta\alpha\theta \to \gamma}{\eta\beta\theta \to \gamma}\ (\mathrm{cut}),$$

(the $(\cdot \to)$ rule becomes trivial), and the axioms are $\alpha \to \alpha$ and rewriting rules from P with the arrows inversed.

One can easily check the following:

— if $\alpha_1 \Rightarrow^* \beta_1$ and $\alpha_2 \Rightarrow^* \beta_2$, then $\alpha_1\alpha_2 \Rightarrow^* \beta_1\beta_2$;
— if $\alpha \Rightarrow^* \beta$ and $\gamma \Rightarrow^* \eta\alpha\theta$, then $\gamma \Rightarrow^* \eta\beta\theta$.

Then, by induction on the derivation, we get $a_1 \ldots a_l \Rightarrow^* b_1 \ldots b_k$, i.e., $\gamma \Rightarrow^* \delta$. □

One could get rid of the unit constant, using the technique from Kuznetsov (2011).

Lemma 11. Let q be a fresh variable and let $\widetilde{\Gamma} \to \widetilde{C}$ be the sequent $\Gamma \to C$ with $\mathbf{1}$ replaced with $q \,/\, q$ and every variable p_i replaced with $(q \,/\, q) \cdot p_i \cdot (q \,/\, q)$. Then, $\Gamma \to C$ is derivable if and only if $\widetilde{\Gamma} \to \widetilde{C}$ is derivable.

Proof. The $(\mathbf{1} \to)$ rule can be interchanged with any rule applied before. Thus, one can place all applications of $(\mathbf{1} \to)$ directly after axioms. All other rules, except $(\mathbf{1} \to)$, remain valid after the replacements. Axioms with $(\mathbf{1} \to)$ applied are sequents of the form $\mathbf{1}, \ldots, \mathbf{1}, p_i, \mathbf{1}, \ldots, \mathbf{1} \to p_i$ or $\mathbf{1}, \ldots, \mathbf{1} \to \mathbf{1}$. After the replacements, they become derivable sequents $q \,/\, q, \ldots, q \,/\, q, (q \,/\, q) \cdot p_i \cdot (q \,/\, q), q \,/\, q, \ldots, q \,/\, q \to q \,/\, q$ and $q \,/\, q, \ldots, q \,/\, q \to q \,/\, q$. This justifies the 'only if' part.

For the 'if' part, we start with $\widetilde{\Gamma} \to \widetilde{C}$ and substitute $\mathbf{1}$ for q (substitution of arbitrary formulae for variables is legal in the $\mathrm{SLC}^{\mathbf{1}}_{\Sigma}$). Since $(\mathbf{1} \,/\, \mathbf{1})$ is equivalent to $\mathbf{1}$ and $(\mathbf{1} \,/\, \mathbf{1}) \cdot p_i \cdot (\mathbf{1} \,/\, \mathbf{1})$ is equivalent to p_i, the result of this substitution is equivalent to $\Gamma \to C$, whence this sequent is derivable. □

This yields the following theorem.

Theorem 12. If $\mathcal{C} \neq \varnothing$, then the derivability problem in SLC_{Σ} is undecidable.

Finally, SMALC_{Σ} and SCLL_{Σ}, being conservative extensions of SLC_{Σ}, is also undecidable.

Corollary 13. If $\mathcal{C} \neq \varnothing$, then the derivability problem in SMALC_{Σ} is undecidable.

Corollary 14. If $\mathcal{C} \neq \varnothing$, then the derivability problem in SCLL_{Σ} is undecidable.

10. Decidability of systems without contraction

The non-local contraction rule plays a crucial role in our undecidability proof presented in the previous section. If there is no subexponential that allows contraction (i.e., $\mathcal{C} = \varnothing$), the derivability problem becomes decidable.

Theorem 15. If $\mathcal{C} = \varnothing$, then the decidability problem for SCLL_{Σ} belongs to PSPACE and the decidability problem for SMCLL_{Σ} belongs to NP. Hence, both problems are algorithmically decidable.

(Recall that SCLL_Σ is the full cyclic linear logic with subexponentials and SMCLL_Σ is the system without additive constants and connectives, $\top$, $\mathbf{0}$, $\&$ and $\oplus$.)

Proof. By Theorem 1, we consider only cut-free derivations. Since contraction is never applied, each rule, except exchange, introduces at least one new connective into the sequent (weakening and the $(\top)$ axiom can introduce whole subformulae at once, all other rules introduce exactly one connective per rule). Thus, in the situation without additive conjunction (in SMCLL_Σ) these connectives can be disjointly traced down to the goal sequent, and each rule application can be associated with a unique connective occurrence in the goal sequent. For exchange rules, we consider several consequent applications of (ex), possibly for different $?^s A$, as one rule. Correctness of such a joint exchange rule application can still be checked in polynomial time. After this joining, each exchange rule is preceded by another rule or an axiom occurrence, therefore, applications of (ex) give not more than a half of the total number of rules applied in the derivation. Thus, the size of a cut-free derivation in SMCLL_Σ, for Σ with $\mathcal{C} = \varnothing$, is linearly bounded by the size of the goal sequent. Since checking correctness of a derivation can be done in polynomial time, this derivation serves as an NP witness, so the derivability problem for SMCLL_Σ, for Σ with $\mathcal{C} = \varnothing$, belongs to the NP class.

For the whole SCLL_Σ system, we follow the strategy by Lincoln et al. (1992, Section 2.1). Namely, we show that the height of a cut-free derivation tree (again, with joined exchange rules) is linear w.r.t. the size of the goal sequent. This follows from the fact that on a path from the goal sequent to an axiom leaf in the derivation tree each rule either introduces new connectives into the goal sequent or is an exchange rule. Therefore, the length of such a path is linearly bounded by the size of the goal sequent. (On the other hand, the size of the whole derivation tree could be exponential, because the $(\&)$ rule copies the same formulae into different branches.) A derivation tree of polynomial height can be guessed and checked by a non-deterministic Turing machine with polynomially bounded space, using the depth-first procedure (Lincoln et al. 1992, Section 2.1). This establishes the fact that the derivability problem for SCLL_Σ, for Σ with $\mathcal{C} = \varnothing$, belongs to NPSPACE, which is equal to deterministic PSPACE by Savitch's theorem (Savitch 1970). $\square$

By Corollary 4, we also get decidability results for the corresponding Lambek systems.

Corollary 16. If $\mathcal{C} = \varnothing$, then the decidability problem for SMALC_Σ belongs to PSPACE and the decidability problem for SLC_Σ belongs to NP. Hence, both systems are algorithmically decidable.

Notice that these complexity bounds are exact, since even without subexponentials the derivability problem in the purely multiplicative Lambek calculus is NP-complete (Pentus 2006) and the derivability problem in the MALC is PSPACE-complete (Kanovich 1994) (see also Kanazawa 1999).

11. Conclusions and future work

In this paper, we have considered two systems of non-commutative linear logic – the MALC and cyclic propositional linear logic – and extended them with subexponentials. For these extended systems, we have proved cut elimination and shown that the first system can be conservatively embedded into the second one. We have also shown that, for cut elimination to hold, the contraction rule should be in the non-local form. Finally, we have established exact algorithmic complexity estimations. Namely, at least one subexponential that allows contraction makes the system undecidable. On the other hand, subexponentials that do not allow contraction do not increase complexity in comparison with the original system without subexponentials: it is still NP for multiplicative systems and PSPACE for multiplicative-additive ones.

A natural step to take from here is to investigate focused (Andreoli 1992) proof systems with non-commutative subexponentials. This would open a number of possibilities such as the development of logical frameworks with non-commutative subexponentials. Such frameworks have been used, for example, by Pfenning and Simmons (2009) for the specification of evaluation strategies of functional programs. While their focused proof system contained a single unbounded modality, a single bounded modality and a single non-commutative modality, focused proof systems with commutative and non-commutative subexponentials would allow for any number of modalities allowing the encoding of an even wider range of systems. Such investigation is left for future work.

In our undecidability proof, we encoded semi-Thue systems in SLC_Σ, using only three connectives, $/$ (one can dually use $\backslash$, of course), $\cdot$ and $!^s$ (where $s \in \mathcal{C}$). The language can be further restricted to $/$ and $!^s$, without $\cdot$, by using a more sophisticated encoding by Buszkowski (1982), see Kanovich et al. (2016b). The number of variables used in the construction could be also reduced to one variable using the technique by Kanovich (1995). We leave the details of these restrictions for future work.

On the other hand, if we allow subexponentials with contraction to be applied only to variables ($!^s p$) or to formulae without $\cdot$ of implication depth 1 (for example, $!^s(p \,/\, q)$), the derivability problem probably becomes decidable, which would be quite nice for linguistic applications. We leave this as an open question for future studies.

For extensions of the Lambek calculus, another interesting question, besides decidability and algorithmic complexity, is the generative power of categorial grammars based on these extensions. Original Lambek grammars generate precisely context-free languages (Pentus 1993). On the other hand, it actually follows from our undecidability proof that grammars based on SLC_Σ, where at least one subexponential in Σ allows contraction ($\mathcal{C} \neq \varnothing$), can generate an arbitrary recursively enumerable language. For decidable fragments (e.g., when $\mathcal{C} = \varnothing$, or subexponentials allowing contraction are somehow restricted syntactically), however, determining the class of languages generated by corresponding grammars is left for future research.

Acknowledgments

We are grateful to Dale Miller, Glyn Morrill, Frank Pfenning, the organisers of Dale Fest, and the anonymous referees of the Journal for their very helpful questions, comments and suggestions, thanks to which this paper is now substantially different from our original presentation at the Dale Fest. The work of Max Kanovich and Andre Scedrov was performed at the International Laboratory for Intelligent Systems and Structural Analysis at the Faculty of Computer Science, National Research University Higher School of Economics, Moscow. They would like to thank Sergei O. Kuznetsov for providing a very pleasant place for productive work. After the first draft of this paper was prepared, Stepan Kuznetsov was selected as one of the Young Russian Mathematics award winners, and he would like to thank its sponsors and jury. (The work of Stepan Kuznetsov on this particular paper was funded from another source, as mentioned below.)

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 Russian Science Foundation under Grant 16-11-10252 and performed at Steklov Mathematical Institute of the Russian Academy of Sciences, Moscow, Russia. The work of Vivek Nigam was supported by CNPq Grant number 304193/2015-1.

Sections 4, 9, 10, and 11 were contributed by Max Kanovich and Andre Scedrov; Sections 6–8 were contributed by Stepan Kuznetsov; Sections 1 and 3 were contributed by Vivek Nigam. Sections 2 and 5 were contributed jointly and equally by all coauthors.

References

Abrusci, V.M. (1990). A comparison between Lambek syntactic calculus and intuitionistic linear logic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **36** 11–15.

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studia Philosophica* **1** 1–27.

Andreoli, J.-M. (1992). Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation* **2**(3) 297–347.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language* **29** 47–58.

van Benthem, J. (1991). *Language in Action: Categories, Lambdas and Dynamic Logic*, North Holland, Amsterdam.

Braüner, T. and de Paiva, V. (1998). A formulation of linear logic based on dependency relations. In: *Proceedings of the International Workshop on Computer Science Logic (CSL '97)*, Lecture Notes in Computer Science, vol. 1414, Springer, 129–148.

Buszkowski, W. (1982). Some decision problems in the theory of syntactic categories. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **28** 539–548.

Buszkowski, W. (2010). Lambek calculus and substructural logics. *Linguistic Analysis* **36** (1–4) 15–48.

Chaudhuri, K. (2010). Classical and intuitionistic subexponential logics are equally expressive. In: *Proceedings of the International Workshop on Computer Science Logic (CSL '10)*, Lecture Notes in Computer Science, vol. 6247, Springer, 185–199.

Danos, V., Joinet, J.-B. and Schellinx, H. (1993). The structure of exponentials: Uncovering the dynamics of linear logic proofs. In: *Proceedings of the Kurt Gödel Colloquium on Computational Logic and Proof Theory (KGC '93)*, Lecture Notes in Computer Science, vol. 713, Springer, 159–171.

Gentzen, G. (1935). Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift* **39** 176–210.

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

de Groote, P. (2005). On the expressive power of the Lambek calculus extended with a structural modality. In: *Language and Grammar*, CSLI Lecture Notes, vol. 168, CSLI, 95–111.

Hodas, J. and Miller, D. (1991). Logic programming in a fragment of intuitionistic linear logic. Extended abstract. In: *Proceedings of the 6th Annual Symposium on Logic in Computer Science (LICS '91)* 32–42.

Hodas, J. and Miller, D. (1994). Logic programming in a fragment of intuitionistic linear logic. *Information and Computation* **110**(2) 327–365.

Kanazawa, M. (1992). The Lambek calculus enriched with additional connectives. *Journal of Logic, Language and Information* **1**(2) 141–171.

Kanazawa, M. (1999). Lambek calculus: Recognizing power and complexity. In: *Johannes Franciscus Abraham Karel (JFAK). Essays dedicated to Johan van Benthem to the occasion of his 50th birthday*. Vossiuspers, Amsterdam Univ. Press.

Kanovich, M. (1994). Horn fragments of non-commutative logics with additives are PSPACE-complete. In: *Extended abstract, presented at the 8th Annual Conference of the European Association for Computer Science Logic (CSL '94)*, Kazimierz, Poland (unpublished manuscript).

Kanovich, M.I. (1995). The complexity of neutrals in linear logic. In: *Proceedings of the 10th Annual Symposium on Logic in Computer Science (LICS '95)*, IEEE, 486–495.

Kanovich, M., Kuznetsov, S. and Scedrov, A. (2016a). Reconciling Lambek's restriction, cut-elimination, and substitution in the presence of exponential modalities. arXiv 1608.02254.

Kanovich, M., Kuznetsov, S. and Scedrov, A. (2016b). Undecidability of the Lambek calculus with a relevant modality. In: *Proceedings of the International Conference on Formal Grammar (FG '15 and '16)*, Lecture Notes in Computer Science, vol. 9804, Springer, 240–256.

Kanovich, M., Kuznetsov, S. and Scedrov, A. (2016c). On Lambek's restriction in the presence of exponential modalities. In: Artemov, S. and Nerode, A. (eds.) *Proceedings of the International Symposium on Logical Foundations of Computer Science (LFCS '16)*, Lecture Notes in Computer Science, vol. 9537, Springer, 146–158.

Kanovich, M., Kuznetsov, S. and Scedrov, A. (2017). Undecidability of the Lambek calculus with subexponential and bracket modalities. In: *Fundamentals of Computation Theory*, Lecture Notes in Computer Science, vol. 10472, Springer, 326–340.

Kuznetsov, S. and Okhotin, A. (2017). Conjunctive categorial grammars. In: *Proceedings of the 15th Meeting on the Mathematics of Language (MoL '17)*, ACL Anthology, vol. W17-3414, 140–151.

Kuznetsov, S.L. (2011). On the Lambek calculus with a unit and one division. *Moscow University Mathematics Bulletin* **66**(4) 173–175.

Kuznetsov, S.L. (2016). On translating Lambek grammars with one division into context-free grammars. *Proceedings of the Steklov Institute of Mathematics* **294** 129–138.

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

Lambek, J. (1961). On the calculus of syntactic types. In: *Structure of Language and Its Mathematical Aspects*, *Proceedings of the Symposia Applied Mathematics*, vol. 12, AMS, 166–178.

Lambek, J. (1969). Deductive systems and categories II. Standard constructions and closed categories. In: *Category Theory, Homology Theory and Their Applications I*, Lecture Notes in Mathematics, vol. 86, Springer, 76–122.

Lambek, J. (1993). From categorial grammar to bilinear logic. In: *Substructural Logics*, Studies in Logic and Computations, vol. 2, Clarendon Press, Oxford, 207–237.

Lincoln, P., Mitchell, J., A. Scedrov and Shankar, N. (1992). Decision problems for propositional linear logic. *Annals of Pure and Applied Logic* **56**(1) 239–311.

Markov, A. (1947). On the impossibility of certain algorithms in the theory of associative systems. *Doklady Academy of Sciences USSR (N. S.)* **55** 583–586.

Miller, D. (1994). A multiple-conclusion meta-logic. In: *Proceedings of the 9th Annual Symposium on Logic in Computer Science(LICS '94)*, IEEE, 272–281.

Miller, D. (1996). Forum: A multiple-conclusion specification logic. *Theoretical Computer Science* **165**(1) 201–232.

Moortgat, M. (1996). Multimodal linguistic inference. *Journal of Logic, Language and Information* **5**(3–4) 349–385.

Moortgat, M. (1997). Categorial type logics. In: van Benthem, J. and ter Meulen, A. (eds.), *Handbook of Logic and Language*, Elsevier.

Moot, R. (2017). The Grail theorem prover: Type theory for syntax and semantics. In: *Modern Perspectives in Type-Theoretical Semantics*, Studies in Linguistics and Philosophy, vol. 98, Springer, 247–277.

Moot, R. and Retoré, C. (2012). *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*, Lecture Notes in Computer Science, vol. 6850, Springer.

Morrill, G. (2017a). Parsing logical grammar: CatLog3. In: *Proceedings of the Workshop on Logic and Algorithms in Computational Linguistics (LACompLing '17)*, Stockholm University, 107–131.

Morrill, G. (2017b). Grammar logicised: Relativisation. *Linguistics and Philosophy* **40**(2) 119–163.

Morrill, G. and Valentín, O. (2015). Computation coverage of TLG: Nonlinearity. In: *Proceedings of the Natural Language and Computer Science (NLCS '15)*, EPiC Series, vol. 32, 51–63.

Morrill, G.V. (2011). *Categorial Grammar: Logical Syntax, Semantics, and Processing*, Oxford Univ. Press.

Nigam, V. (2012). On the complexity of linear authorization logics. In: *Proceedings of the 27th Annual Symposium on Logic in Computer Science (LICS '12)*, IEEE, 511–520.

Nigam, V. (2014). A framework for linear authorization logics. *Theoretical Computer Science* **536**(0) 21–41.

Nigam, V. and Miller, D. (2009). Algorithmic specifications in linear logic with subexponentials. In: *Proceedings of the Principles and Practice of Declarative Programming (PPDP '09)* 129–140.

Nigam, V., Olarte, C. and Pimentel, E. (2013). A general proof system for modalities in concurrent constraint programming. In: *Proceedings of the International Conference on Concurrency Theory (CONCUR '13)*, Lecture Notes in Computer Science, vol. 8052, Springer, 410–424.

Nigam, V., Pimentel, E. and Reis, G. (2016). An extended framework for specifying and reasoning about proof systems. *Journal of Logic and Computation* **26**(2) 539–576.

Olarte, C., Pimentel, E. and Nigam, V. (2015). Subexponential concurrent constraint programming. *Theoretical Computer Science* **606** 98–120.

Peirce, C. S. (1885). On the algebra of logic: A contribution to the philosophy of notation. *American Journal of Mathematics* **7** 180–202.

Pentus, M. (1993). Lambek grammars are context-free. In: *Proceedings of the 8th Annual Symposium on Logic in Computer Science (LICS '93)*, IEEE, 429–433.

Pentus, M. (1998). Free monoid completeness of the Lambek calculus allowing empty premises. In: *Proceedings of the Logic Colloquium '96*, Lecture Notes Logic, vol. 12, Springer, 171–209.

Pentus, M. (2006). Lambek calculus is NP-complete. *Theoretical Computer Science* **357**(1) 186–201.

Pfenning, F. and Simmons, R.J. (2009). Substructural operational semantics as ordered logic programming. In: *Proceedings of the 24th Annual Symposium on Logic In Computer Science (LICS '09)*, IEEE Computer Society, 101–110.

Post, E.L. (1947). Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic* **12** 1–11.

Savitch, W.J. (1970). Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences* **4**(2) 177–192.

Schellinx, H. (1991). Some syntactical observations on linear logic. *Journal of Logic and Computation* **1**(4) 537–559.

Shieber, S. M. (1985). Evidence against the context-freeness of natural languages. *Linguistics and Philosophy* **8** 333–343.

Thue, A. (1914). Probleme über Veränderungen von Zeichenreihen nach gegebener Regeln. *Christiana Videnskabs-Selskabets Skrifter* **10** 1–34.

Yetter, D.N. (1990). Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic* **55**(1) 41–64.