Lambek Calculus with Banged Atoms for Parasitic Gaps

Mehrnoosh Sadrzadeh[1][0000−0002−5863−7835]
Lutz Straßburger[2,3][0000−0003−4661−6540]

[1] University College London, UK
[2] INRIA Saclay, Ile-de-France, France

Abstract. Lambek Calculus is a non-commutative substructural logic for formalising linguistic constructions. However, its domain of applicability is limited to constructions with local dependencies. We propose here a simple extension that allows us to formalise a range of relativised constructions with long distance dependencies, notably medial extractions and the challenging case of parasitic gaps. In proof theoretic terms, our logic combines commutative and non-commutative behaviour, as well as linear and non-linear resource management. This is achieved with a single restricted modality. But unlike other extensions of Lambek Calculus with modalities, our logic remains decidable, and the complexity of proof search (i.e., sentence parsing) is the same as for the basic Lambek calculus. Furthermore, we provide not only a sequent calculus, and a cut elimination theorem, but also proof nets.

Keywords: Substructural Logics · Permutation and Contraction · Exponentials · Proof Nets · Polarised Systems · Natural Language · Relativisation · Long Distance Dependencies

1 Introduction

The *Lambek Calculus* [21] is a well-estabished tool in formal linguistics. It is a cut-free proof calculus for a logic with a non-commutative multiplication and two directional divisions. Proof search in the Lambek calculus is **NP**-complete [37]. It has been implemented in automatic parsers—the most recent using deep neural network technology—and applied to verify the grammaticality of sentences and phrases of languages such as English, Dutch and French [29,28,18,19]. However, the non-commutative nature of Lambek Calculus means that it can only model the elementary fragments of these languages.

Natural Languages, on the other hand, witnesses long distance dependencies in a range of constructions such as relativisation, topicalisation, and wh-questions. In these cases, a syntactic element such as an object of a verb e.g. the direct object “articles” of the verb “read” in (a) below, is withdrawn from its original place, i.e. after the verb, and placed at a different part of the sentence, for instance to the head of the relative clause. In such cases, a relative

pronoun such as "that" *relates* the head to the rest of the clause. Traditionally, withdrawal places are called *gaps* and marked with $-$, as in the examples below:
(a) articles that reviewers read $-$ yesterday
(b) articles that reviewers accepted $-_1$ without reading $-_2$
(c) articles that reviewers accepted $-_1$ after skimming $-_2$ without reading $-_3$
(d) the article that a review about $-_1$ in a blog post made $-_2$ famous
(e) articles that chairs persuaded every reviewer of $-_1$ to accept $-_2$
A relative clause can have multiple gaps, where often some of the gaps, for instance $-_2$ in (b) and $-_2$ and $-_3$ in (c), become *parasitic*. As the name suggests, for a gap to be parasitic means that its presence relies on the existence of another gap [5,3,26]. In other words, the second and third gaps did not exist if the first gap was not there in the first place. In order to see this, note that phrases such as "articles that reviewers read them" are not grammatical. In (b), the main gap $-_1$ is after the verb "accepted" and the *parasitic* gap $-_2$ occurs after the gerund "reading". In (c), we have a similar situation, but one main gap $-_1$ and two parasitic ones $-_2$ and $-_3$. The parasitic gap can be the first gap, as is the case for $-_1$ in (d). In this case, both gaps are arguments of the verb "made". Gaps can also be nested, as in (e), where the parasitic gap $-_2$ is the object of "accept", itself nested within a complement of "persuaded".

Naturally, the Lambek calculus has been extended in various ways with different modalities and additional binary connectives to model long distance dependencies, for example [31,2,24,11,12,25,36,32,34,33]. There has been some recent interest in this by adding the linear logic exponentials [6] and subexponentials [35] to the Lambek calculus [13,14]. However, the presence of exponentials or subexponentials with contraction in a system with non-commutative connectives leads immediately to an undecidable proof system [15,16,39]. The existing solution [13,14] is imposing an external bound on the number of copies. Apart from needing *a priori* knowledge about the bound, this solution contradicts the compositional nature of language constructions. Furthermore, even with such a bound, the complexity of parsing is much higher [22].

We propose here an alternative solution that simply restricts the structure of the formulas under the modalities. For the linguistic applications mentioned above, it is enough to allow the exponentials only for atomic formulas. With this insight, we are able to present a cut-free proof system with the same proof search complexity as the original Lambek calculus.

Besides the aforementioned linguistic motivation for our proof system, there is also an independent proof theoretical interest. Since the advent of linear logic, there has been a dichtonomy between linear resources (that can be used exactly once) and classical resources (that can be used unlimited). From the beginning, there have been attempts to merge the two in a single proof system, starting with linear logic itself with the exponentials added, leading to sophisticated systems like Girard's LU [7], merging all the connectives of classical and linear logic. We take here the opposite direction, allowing the duplication only for a selected set of atoms, which leads to a very simple proof system.

Another dichotomy in substructural proof theory is the one between commutativity and non-commutativity—again with various existing proposals to merge

$$
\begin{array}{ccc}
\dfrac{\Gamma_1, a, \Gamma_2 \vdash C}{\Gamma_1, !a, \Gamma_2 \vdash C}\;!_{\mathsf{L}} &
\dfrac{}{!a \vdash !a}\;\mathsf{id}_! &
\dfrac{}{a \vdash a}\;\mathsf{id} \\[2ex]
\dfrac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, A \cdot B, \Gamma_2 \vdash C}\;\cdot_{\mathsf{L}} &
\dfrac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \cdot B}\;\cdot_{\mathsf{R}} &
\dfrac{\Gamma_1, !a, !a, \Gamma_2 \vdash C}{\Gamma_1, !a, \Gamma_2 \vdash C}\;\mathsf{con} \\[2ex]
\dfrac{\Delta \vdash A \quad \Gamma_1, B, \Gamma_2 \vdash C}{\Gamma_1, \Delta, A \backslash B, \Gamma_2 \vdash C}\;\backslash_{\mathsf{L}} &
\dfrac{A, \Gamma \vdash B}{\Gamma \vdash A \backslash B}\;\backslash_{\mathsf{R}} &
\dfrac{\Gamma_1, !a, \Delta, \Gamma_2 \vdash C}{\Gamma_1, \Delta, !a, \Gamma_2 \vdash C}\;\mathsf{perm}_1 \\[2ex]
\dfrac{\Gamma_1, B, \Gamma_2 \vdash C \quad \Delta \vdash A}{\Gamma_1, B/A, \Delta, \Gamma_2 \vdash C}\;/_{\mathsf{L}} &
\dfrac{\Gamma, A \vdash B}{\Gamma \vdash B/A}\;/_{\mathsf{R}} &
\dfrac{\Gamma_1, \Delta, !a, \Gamma_2 \vdash C}{\Gamma_1, !a, \Delta, \Gamma_2 \vdash C}\;\mathsf{perm}_2
\end{array}
$$

Fig. 1. System $\mathsf{L}_!$

the two in a single proof system [38,1,8,9]. All of these follow the idea of having both, commutative and non-commutative connectives in the logic. Our approach here is fundamentally different. We only have non-commutative connectives, but we allow to permute a selected set of atoms.

The result is a very simple proof system, that allows us to obtain a notion of proofs nets that is very similar to the one for plain MLL.

In summary, this paper presents a sequent calculus for a logic that is based on the Lambek calculus, but allows the duplication and permutation of atomic formulas that are marked by the !-modality. We prove cut elimination and provide a notion of proof nets with a simple correctness criterion, and we show decidability and **NP**-completeness of the provability problem. This is achieved by translating our system into a one-sided proof system that is based on cyclic MLL [40] and uses the methods presented in [20]. Finally, we show how our calculus can be applied for linguistic purposes.

2 The System

We start by defining the set $\mathcal{L}$ of ***formulas*** of our calculus. Formulas are denoted by capital Latin letters $A, B, \ldots$, and are generated from a countable set $\mathcal{A} = \{a, b, \ldots, s, n, gp, \ldots\}$ of ***atoms*** via the following grammar

$$\mathcal{L} ::= \mathcal{A} \mid !\mathcal{A} \mid \mathcal{L} \cdot \mathcal{L} \mid \mathcal{L} \backslash \mathcal{L} \mid \mathcal{L}/\mathcal{L}$$

Observe that the modality ! operates only on atoms, whereas the binary connectives $\cdot$, $\backslash$, and / operate on arbitrary formulas, the modality ! can only be applied to atoms, and we call those atom ***banged atoms***. We use capital Greek letters $\Gamma, \Delta, \ldots$, to denote finite lists of formulas, separated by comma: $\Gamma = A_1, \ldots, A_n$. A ***sequent*** is a pair $\Gamma \vdash A$. The inference rules of our system, denoted by $\mathsf{L}_!$ and shown in Figure 1, consist of the standard Lambek calculus [21], extended by inference rules dealing with the !-modality. As discussed in the introduction, banged atoms can be duplicated and freely permuted with other formulas. This

$$
\dfrac{
\dfrac{\dfrac{}{n\vdash n}\,\mathsf{id}\quad\dfrac{}{n\vdash n}\,\mathsf{id}}{n, n\backslash n\vdash n}\,\backslash_{\mathsf{L}}
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\dfrac{\dfrac{}{n\vdash n}\,\mathsf{id}\quad\dfrac{}{s\vdash s}\,\mathsf{id}}{n, n\backslash s\vdash s}\,\backslash_{\mathsf{L}}}{n\backslash s\vdash n\backslash s}\,\backslash_{\mathsf{R}}
\quad
\dfrac{\dfrac{}{n\vdash n}\,\mathsf{id}\quad\dfrac{}{s\vdash s}\,\mathsf{id}}{n, n\backslash s\vdash s}\,\backslash_{\mathsf{L}}
}{n, n\backslash s, (n\backslash s)\backslash(n\backslash s)\vdash s}\,\backslash_{\mathsf{L}}
\quad \dfrac{}{n\vdash n}\,\mathsf{id}
}{n, (n\backslash s)/n, n, (n\backslash s)\backslash(n\backslash s)\vdash s}\,/_{\mathsf{L}}
\quad \dfrac{}{gp\vdash gp}\,\mathsf{id}
}{n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp\vdash s}\,/_{\mathsf{L}}
\quad \dfrac{}{n\vdash n}\,\mathsf{id}
}{n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n\vdash s}\,/_{\mathsf{L}}
}{n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n\vdash s}\,!_{\mathsf{L}}
}{n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n\vdash s}\,!_{\mathsf{L}}
}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, !n\vdash s}\,\mathsf{perm}_1
}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n\vdash s}\,\mathsf{con}
}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n\vdash s/!n}\,/_{\mathsf{R}}
}{n, (n\backslash n)/(s/!n), n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n\vdash n}\,/_{\mathsf{L}}
$$

Fig. 2. Derivation in $\mathsf{L}_!$ for example (b)

is achieved by the rules con (*contraction*) and perm_1 and perm_2 (*permutation*)[3] In order to be able to *use* banged atoms in a proof, we need to release the modality eventually. This is done by the $!_{\mathsf{L}}$-rule.

In order to have cut elimination, to be discussed in the next section, we also need a $!_R$-rule. The reader familiar with rules for modalities in the sequent calculus, might immediately think of the following three options:

$$
\dfrac{a\vdash b}{!a\vdash !b}\,!^1_{\mathsf{R}}
\qquad\qquad
\dfrac{a_1,\ldots,a_n\vdash b}{!a_1,\ldots,!a_n\vdash !b}\,!^2_{\mathsf{R}}
\qquad\qquad
\dfrac{!a_1,\ldots,!a_n\vdash b}{!a_1,\ldots,!a_n\vdash !b}\,!^3_{\mathsf{R}}
\tag{1}
$$

All three rules would ensure the cut elimination result. Now observe that $!^1_{\mathsf{R}}$ is a special case of $!^2_{\mathsf{R}}$, and $!^2_{\mathsf{R}}$ is derivable from $!^3_{\mathsf{R}}$ with the $!_{\mathsf{L}}$-rule. Thus, in the general case, the three rules have increasing strength. However, since there is no weakening and the ! is restricted to atoms, all three rules are equivalent. This is easy to see by observing that $!a_1,\ldots,!a_n\vdash b$ is only provable if $n=1$ and $a_1=b$. For this reason, it is sufficient to have $!_{\mathsf{R}} = !^1_{\mathsf{R}}$.

However, as the sequent $a\vdash b$ is only provable if $a=b$, and in that case the proof consists of a single application of the id-rule, we can replace $!_{\mathsf{R}}$ by the $\mathsf{id}_!$-rule.[4]

Finally, we have the so-called *Lambek restriction* which demands that we never have an empty left-hand side in a sequent. Looking at our rules, this is equivalent to asking that in the rules $\backslash_{\mathsf{R}}$ and $/_{\mathsf{R}}$, the Γ must not be empty.[5]

[3] Note that we do not have *weakening*. This has linguistic reasons as well as proof theoretical reasons [10].

[4] We thank an anonymous referee for this simplification. However, if the reader insists on a system with only atomic id-rules, then $\dfrac{}{!a\vdash !a}\,\mathsf{id}_!$ can be replaced by $\dfrac{a\vdash b}{!a\vdash !b}\,!_{\mathsf{R}}$, yielding an equivalent system, not affecting any of the results in this paper.

[5] Observe that since we only allow atoms under the !, we do not have the issue discussed in [17].

The cut-rule has the usual form: $\dfrac{\Delta \vdash A \quad \Gamma_1, A, \Gamma_2 \vdash C}{\Gamma_1, \Delta, \Gamma_2 \vdash C}\,\mathsf{cut}$

Theorem 2.1. *If a sequent is provable in* $\mathsf{L}_! + \mathsf{cut}$*, then it is also provable in* $\mathsf{L}_!$.

We show the proof in the next section, and we end this section with the observation that even though the the id-rule is restricted to atoms, its general form is derivable.

Proposition 2.2. *The rule* $\dfrac{}{A \vdash A}\,\mathsf{id}$ *is derivable in* $\mathsf{L}_!$.

Proof. Straightforward induction on A. □

Example 2.3. Figure 2 shows the derivation for our example (b), which will serve as the running example for this paper. For better readability we highlighted for each inference rule instance the principal formula in the conclusion.

3 Cut Elimination

We prove Theorem 2.1 with the help of the following reduction lemma:

Lemma 3.1. *If we have a proof*

$$\dfrac{\overset{\pi_1}{\Delta \vdash A} \qquad \overset{\pi_2}{\Gamma_1, A, \Gamma_2 \vdash C}}{\Gamma_1, \Delta, \Gamma_2 \vdash C}\,\mathsf{cut}$$

where π_1 *and* π_2 *are proofs in* $\mathsf{L}_!$ *that do not contain any cuts, then* $\Gamma_1, \Delta, \Gamma_2 \vdash C$ *is also provable in* $\mathsf{L}_!$ *without the* cut*-rule.*

For proving this lemma, we need the following notions. For a formula A, we define $|A|$ to be the ***size*** of A, i.e., the number of symbols needed to write A. For a proof π, we define $|\pi|$ to be the ***size*** of π, i.e., the number of inference rule instances used in π.

Proof (of Lemma 3.1). The proof is standard and proceeds by induction on the lexicographic pair $\langle |A|, |\pi_1| + |\pi_2| \rangle$ and a case analysis on the bottom-most rule instances in π_1 and π_2.

1. If the bottommost rule instance in π_1 or in π_2 does not operate on the cut-formula A, then we can do a simple rule permutation and proceed by induction hypothesis (since $|\pi_1| + |\pi_2|$ has decreased).
2. We now consider the cases where the bottommost rules instances in π_1 and in π_2 do operate on the cut-formula A. If $A = a$ for some atom a, then the cut disappears trivially.
3. If $A = A'/A''$, then we can replace

$$\dfrac{\dfrac{\overset{\pi_1'}{\Delta, A'' \vdash A'}}{\Delta \vdash A'/A''}\,/_\mathsf{R} \qquad \dfrac{\overset{\pi_2'}{\Gamma_1, A', \Gamma_2 \vdash C} \quad \overset{\pi_2''}{\Lambda \vdash A''}}{\Gamma_1, A'/A'', \Lambda, \Gamma_2 \vdash C}\,/_\mathsf{L}}{\Gamma_1, \Delta, \Lambda, \Gamma_2 \vdash C}\,\mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\pi_2''}{\Lambda \vdash A''} \quad \overset{\pi_1'}{\Delta, A'' \vdash A'}}{\Delta, \Lambda \vdash A'}\,\mathsf{cut} \qquad \overset{\pi_2'}{\Gamma_1, A', \Gamma_2 \vdash C}}{\Gamma_1, \Delta, \Lambda, \Gamma_2 \vdash C}\,\mathsf{cut}$$

   and proceed by induction hypothesis.

4. If $A = A'\backslash A''$ or $A = A' \cdot A''$, we proceed similarly.
5. Let us now look at the cases when $A = !a$ for some atom a. There are four possibilities for the bottommost rule in π_2. In the case for $!_{\mathsf{L}}$ we have

$$\dfrac{\dfrac{}{!a \vdash !a}\,\mathsf{id}_! \qquad \dfrac{\overset{\pi_2'}{\Gamma_1, a, \Gamma_2 \vdash C}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,!_{\mathsf{L}}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,\mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\overset{\pi_2'}{\Gamma_1, a, \Gamma_2 \vdash C}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,!_{\mathsf{L}}$$

and in the case of the con-rule we have

$$\dfrac{\dfrac{}{!a \vdash !a}\,\mathsf{id}_! \qquad \dfrac{\overset{\pi_2'}{\Gamma_1, !a, !a, \Gamma_2 \vdash C}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,\mathsf{con}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,\mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\overset{\pi_2'}{\Gamma_1, !a, !a, \Gamma_2 \vdash C}}{\Gamma_1, !a, \Gamma_2 \vdash C}\,\mathsf{con}$$

Finally, the cases of the perm_1- and perm_2-rules is similar to case 1 above. □

Proof (of Theorem 2.1). By induction on the number of cuts in the derivation, using Lemma 3.1. □

4 A One-sided System

In this section we present a system equivalent to $\mathsf{L}_!$, but with the (non-commutative versions of the) two connectives $\otimes$ and $\parr$ from multiplicative linear logic (MLL). For this, we closely follow the presentation of [20]. The main ingredient is the distinction of the formulas that would occur on the left-hand side of the sequent turnstile from the ones on the right-hand side, using the polarities $\bullet$ (called ***input***) and $\circ$ (called ***output***).

The *raison d'être* of this exercise is to give the proof nets in the next section, which will essentially be the proof nets for cyclic MLL restricted to the Lambek calculus as presented in [20], with some additional conditions for accommodating our !-atoms.

First, let us recall MLL in its cyclic version [40]. While doing so, we add the necessary modalities that will correspond to the one of $\mathsf{L}_!$. We denote this logic by $\mathsf{CyMLL}_{!?}$,[6]. For defining its set $\mathcal{M}$ of formulas, we introduce a set $\mathcal{A}^\perp = \{a^\perp, b^\perp, \ldots, s^\perp, n^\perp, gp^\perp, \ldots\}$ of dual atoms, isomorphic to $\mathcal{A}$. We let $\mathcal{X} = \mathcal{A} \cup \mathcal{A}^\perp$ and define $\mathcal{M}$ via the following grammar:

$$\mathcal{M} ::= \mathcal{X} \mid !\mathcal{X} \mid ?\mathcal{X} \mid \mathcal{M} \otimes \mathcal{M} \mid \mathcal{M} \parr \mathcal{M}$$

The operation $(\cdot)^\perp$ can be extended to all formulas via $a^{\perp\perp} = a$, and $(!a)^\perp = ?a^\perp$, and $(?a)^\perp = !a^\perp$, and $(A \otimes B)^\perp = B^\perp \parr A^\perp$, and $(A \parr B)^\perp = B^\perp \otimes A^\perp$.[7] Sequents of $\mathsf{CyMLL}_{!?}$ are simply finite lists of formulas $\vdash \Gamma = \vdash A_1, \ldots, A_n$, i.e., everything is one-sided, and we can write $\parr\Gamma$ for $A_1 \parr \cdots \parr A_n$. The inference rules are shown in Figure 3, and as expected, the cut-rule (in its one-sided version) is admissible.[8]

[6] Note that even though the language is similar to cyclic MELL, the logic is very different.

[7] Note the inversion of the order of the arguments, which is crucial.

[8] There is only one perm-rule in $\mathsf{CyMLL}_{!?}$ because the other one is derivable with the help of the cyc-rule.

$$
\dfrac{}{\vdash a, a^{\perp}}\,\mathsf{id}
\qquad
\dfrac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \parr B, \Delta}\,\parr
\qquad
\dfrac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\,\otimes
\qquad
\dfrac{\vdash \Delta, \Gamma}{\vdash \Gamma, \Delta}\,\mathsf{cyc}
$$

$$
\dfrac{}{\vdash ?a, !a^{\perp}}\,\mathsf{id}_{!}
\qquad
\dfrac{\vdash \Gamma_1, a, \Gamma_2}{\vdash \Gamma_1, ?a, \Gamma_2}\,?
\qquad
\dfrac{\vdash \Gamma_1, ?a, ?a, \Gamma_2}{\vdash \Gamma_1, ?a, \Gamma_2}\,\mathsf{con}
\qquad
\dfrac{\vdash \Gamma_1, \Delta, ?a, \Gamma_2}{\vdash \Gamma_1, ?a, \Delta, \Gamma_2}\,\mathsf{perm}
$$

Fig. 3. Inference rules for $\mathsf{CyMLL}_{!?}$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash n^{\perp}, n$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash n^{\perp}, n$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash n^{\perp}, n \otimes n^{\perp}, n$}
\RightLabel{cyc}
\UnaryInfC{$\vdash n, n^{\perp}, n \otimes n^{\perp}$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$s \vdash s$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash n^{\perp}, n \otimes s^{\perp}, s$}
\RightLabel{cyc}
\UnaryInfC{$\vdash n \otimes s^{\perp}, s, n^{\perp}$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash n \otimes s^{\perp}, s \parr n^{\perp}$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash n^{\perp}, n$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash s^{\perp}, s$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash n^{\perp}, n \otimes s^{\perp}, s$}
\RightLabel{cyc}
\UnaryInfC{$\vdash n \otimes s^{\perp}, s, n^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash n \otimes s^{\perp}, (s \parr n^{\perp}) \otimes (n \otimes s^{\perp}), s, n^{\perp}$}
\RightLabel{cyc}
\UnaryInfC{$\vdash (s \parr n^{\perp}) \otimes (n \otimes s^{\perp}), s, n^{\perp}, n \otimes s^{\perp}$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash n, n^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash (s \parr n^{\perp}) \otimes (n \otimes s^{\perp}), s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}$}
\RightLabel{cyc}
\UnaryInfC{$\vdash s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}, (s \parr n^{\perp}) \otimes (n \otimes s^{\perp})$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash gp, gp^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp}$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\vdash n, n^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n, n^{\perp}$}
\RightLabel{cyc}
\UnaryInfC{$\vdash n^{\perp}, s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{$?$}
\UnaryInfC{$\vdash ?n^{\perp}, s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, n^{\perp}, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{$?$}
\UnaryInfC{$\vdash ?n^{\perp}, s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, ?n^{\perp}, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{perm}
\UnaryInfC{$\vdash ?n^{\perp}, ?n^{\perp}, s, n^{\perp}, (n \otimes s^{\perp}) \otimes n((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{con}
\UnaryInfC{$\vdash ?n^{\perp}, s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash ?n^{\perp} \parr s, n^{\perp}, (n \otimes s^{\perp}) \otimes n, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash n, n^{\perp}, (n \otimes n^{\perp}) \otimes (?n^{\perp} \parr s), n^{\perp}, (n \otimes s^{\perp}) \otimes n, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n$}
\RightLabel{cyc}
\UnaryInfC{$\vdash n^{\perp}, (n \otimes n^{\perp}) \otimes (?n^{\perp} \parr s), n^{\perp}, (n \otimes s^{\perp}) \otimes n, ((s \parr n^{\perp}) \otimes (n \otimes s^{\perp})) \otimes gp, gp^{\perp} \otimes n, n$}
\end{prooftree}
$$

Fig. 4. Example of a derivation in $\mathsf{CyMLL}_{!?}$

There is the obvious mapping $(\cdot)^{\flat} \colon \mathcal{L} \to \mathcal{M}$, defined via:[9]

$$
a^{\flat} = a \qquad (!a)^{\flat} = !a \qquad (A \cdot B)^{\flat} = B^{\flat} \otimes A^{\flat}
$$
$$
(A \backslash B)^{\flat} = B^{\flat} \parr (A^{\flat})^{\perp} \qquad (B / A)^{\flat} = (A^{\flat})^{\perp} \parr B^{\flat}
$$

Let us now characterize the image of that translation. For this, we define ***polarized formulas***:

$$
\begin{aligned}
\mathcal{M}^{\bullet} &::= \mathcal{A}^{\perp} \mid ?\mathcal{A}^{\perp} \mid \mathcal{M}^{\bullet} \parr \mathcal{M}^{\bullet} \mid \mathcal{M}^{\bullet} \otimes \mathcal{M}^{\circ} \mid \mathcal{M}^{\circ} \otimes \mathcal{M}^{\bullet} \\
\mathcal{M}^{\circ} &::= \mathcal{A} \quad \mid !\mathcal{A} \quad \mid \mathcal{M}^{\circ} \otimes \mathcal{M}^{\circ} \mid \mathcal{M}^{\circ} \parr \mathcal{M}^{\bullet} \mid \mathcal{M}^{\bullet} \parr \mathcal{M}^{\circ}
\end{aligned}
$$

It immediately follows that $\mathcal{M}^{\bullet} \cap \mathcal{M}^{\circ} = \emptyset$ and $\mathcal{M}^{\bullet} \cup \mathcal{M}^{\circ} \subsetneq \mathcal{M}$, and that $(\cdot)^{\perp}$ inverts the polarity.

The purpose of this is to have access to the following two propositions that we can now formulate. For the first one, we define $\mathcal{L}^{\flat}$ to be the image of $\mathcal{L}$ under $(\cdot)^{\flat}$.

Proposition 4.1 ([20]). *We have* $\mathcal{L}^{\flat} = \mathcal{M}^{\circ}$.

[9] Here we also invert the order, even though this is not strictly needed (and also not done in [20]). We do it here to ease the use of the system for the linguistic applications.

Proposition 4.2 ([20]). *A sequent $A_1, \ldots, A_n \vdash B$ is provable in $\mathsf{L}_!$ if and only if the sequent $\vdash (A_1^\flat)^\perp, \ldots, (A_1^\flat)^\perp, B^\flat$ is provable in $\mathsf{CyMLL}_{!?}$.*

Observe that such a translated sequent has exactly one formula with $\circ$-polarity and (because of the non-empty LHS condition of $\mathsf{L}_!$) at least one formula of $\bullet$-polarity.

Example 4.3. In Figure 4 we show the result of applying these two propositions to the derivation in Example 2.3. Observe that the overal structure of the two derivations is the same. This is the reason for the choice of which premise is left and which is right in the $\backslash_\mathsf{L}$- and $/_\mathsf{L}$-rules in Figure 1.[10]

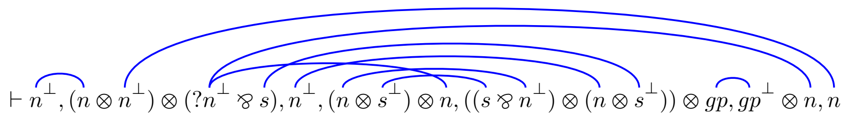
5 Proof Nets

With the work of the previous section, we can now profit from the well developed theory of proof nets for MLL [4], and apply it to our logic, following the presentation in [20].

The ***tree*** of a formula A is the term tree of A, where the inner nodes are labeled by binary connectives and the leaves are labeled by atoms or atoms with a modality.[11] A ***prenet*** π is a pair $\langle \Gamma, \ell \rangle$ where Γ is a finite list of formulas from $\mathcal{M}$ and ℓ is an ***(axiom) linking***, which is a symmetric binary relation on the leaves of the sequent forest Γ, such that

(i) every non-modal atom is related to exactly one other non-modal atom or ?-atom that is dual to it, and
(ii) every !-atom is connected to exactly one ?-atom that is dual to it, and
(iii) every ?-atom is connected only to non-modal atoms or !-atoms that are all dual to it, and it has to be connected to at least one such atom.

Such a related pair of leaves is called an ***axiom link***. We can draw a prenet by simply writing Γ and adding the axiom links on top of it by simply connecting the corresponding atoms by an edge, as in the example below:



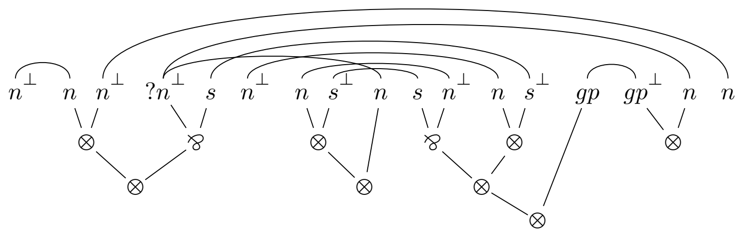
$$\vdash n^\perp, (n \otimes n^\perp) \otimes (?n^\perp \parr s), n^\perp, (n \otimes s^\perp) \otimes n, ((s \parr n^\perp) \otimes (n \otimes s^\perp)) \otimes gp, gp^\perp \otimes n, n \tag{2}$$

The ***graph*** $\mathcal{G}(\pi)$ of the prenet $\pi = \langle \Gamma, \ell \rangle$ is obtained from the sequent forest of Γ by adding an edge between any leaves that are in ℓ-relation. For example,

[10] Furthermore, the order of the formulas in the conclusion is the same in the two derivations in Examples 2.3 and 4.3. This is the reason for inverting the order in the definition of $(\cdot)^\flat$.

[11] To simplify the presentation, we consider the modalities not as unary inner nodes, but as part of the leaves.

below is the graph of the prenet above:

 (3)

$n^\perp$ n $n^\perp$ $?n^\perp$ s $n^\perp$ n $s^\perp$ n s $n^\perp$ n $s^\perp$ gp $gp^\perp$ n n

A ***subnet*** of a prenet π is a prenet π' whose graph $\mathcal{G}(\pi')$ is a subgraph of $\mathcal{G}(\pi)$.

It is straightforward to translate a $\mathsf{CyMLL}_{!?}$ sequent proof into a prenet $\langle \Gamma, \ell \rangle$: Γ is always the endsequent of the proof, and ℓ is constructed inductively as follows: The id-axiom creates a single axiom link between the two atoms in its conclusion. The rules $\parr$, ?, !, cyc, and perm have the same leaves in the premise as in the conclusion and therefore preserve the linking ℓ. The $\otimes$-rule simply takes the union of the linkings of the premises, and the con-rule links the $?a$ in the conclusion to all atoms which are linked to one of the two $?a$ in the premise, and preserves all other links for Γ, Δ (see Fig. 3). A $\mathsf{L}_!$***-proof net*** is a prenet that is obtained in this manner from a $\mathsf{CyMLL}_{!?}$ sequent proof that is the translation of a $\mathsf{L}_!$ sequent proof according to the previous section.

Example 5.1. The prenet in (2) is an $\mathsf{L}_!$-proof net, as it is obtained from the $\mathsf{CyMLL}_{!?}$-proof in Example 4.3, which is the translation of the $\mathsf{L}_!$-proof in Example 2.3.

Now we give a correctness criterion that allows us to distinguish the proof nets without having to resort to the sequent calculus. For this, we are going to adapt the results of [20] to the presence of our modalities. We start with the standard MLL-criterion. A ***DR-switching*** of a prenet π is obtained from $\mathcal{G}(\pi)$ by removing for each $\parr$ node one of the edges connecting it to its childen, and for each ?-leaf all but one axiom edges. Now, a prenet $\pi = \langle \Gamma, \ell \rangle$ is $\mathsf{L}_!$***-correct*** iff the following conditions hold:

1. every DR-switiching of π is a connected and acyclic graph;
2. $\parr\Gamma \in \mathcal{M}^\circ$;
3. the graph obtained from $\mathcal{G}(\pi)$ by removing all axiom edges connected to a ?-leaf is planar; and
4. every subnet of π has at least two conclusions.

Theorem 5.2. *A prenet is a* $\mathsf{L}_!$*-proof net iff it is* $\mathsf{L}_!$*-correct.*

Proof (Sketch). The first condition is the standard MLL-correctness criterion, where our contaction rule con behaves like a $\parr$-rule on atoms. The second condition is the restriction to the intuitionistic setting (exactly one formula on the right-hand side of the $\vdash$). The third condition is about the non-commutativity of the logic (with the exception of the ?-atoms), and the last condition encodes the Lambek restriction that forces a non-empty left-hand side. The details are similar to existing proofs in the literature (e.g., [20]), as the adjustments to the presence of our ?-atoms are straightforward. ☐

Remark 5.3. The four conditions are independent, and their presence or absence leads to different logics. For example, having only conditions 1 and 3 would lead to a correctness criterion for $\mathsf{CyMLL}_{!?}$, which might be of independent proof theoretical interest.

6 Decidability and Complexity

Theorem 6.1. *(i) The logic* $\mathsf{L}_!$ *is decidable. (ii) Moreover, the complexity of proof search is* **NP***-complete.*

Proof. Every axiom link connects a positive and a negative atom. The number of positive atoms is fixed by the endsequent. Only negative atoms can be duplicated by the con-rule. This means that the size of the sequents in a proof is limited, and decidability follows immediately. Furthermore, the size of a sequent is also linear in the size of the endsequent, which means the size of the whole proof is bound by a polynomial and can therefore be guessed by an **NP**-algorithm.[12] Finally, **NP**-hardness follows from **NP**-hardness of the standard Lambek calculus. □

Remark 6.2. The same holds for the logic $\mathsf{CyMLL}_{!?}$, but the proof is a bit more involved, because in $\mathsf{CyMLL}_{!?}$ the con-rule can also duplicate positive atoms. However, the number of possible duplications is still bounded by a polynomial because the DR-switching condition forbids an axiom link between two ?-atoms.

7 Application to Relativisation in Natural Language

In this section, we put $\mathsf{L}_!$ to use and show how it is employed in linguistic reasoning. The simplest cases of relativisation are when the extraction site is either at the front or rear of the original sentence. For instance, at the subject position of a verb in (f) below, or its object position in (g) below.

(f) articles that $-$ disappeared
(g) articles that reviewers read $-$

These are still derivable in Lambek Calculus without permutation or contraction. One starts to face challenges when the extraction sites are medial, as is the case for all of the examples presented in the introduction. In order to demonstrate the differences, we first present a derivation for (g), and then show how our calculus deals with (a)–(c) from the introduction. Derivations for (d) and (e) are provided in the appendix. The proof of (f) is straightforward. In order to model these clauses, we first design a *lexicon*, which is an assignment of $\mathcal{L}$ to the vocabulary used in the examples, where s is a declarative sentence, n a noun phrase, and gp

[12] Note that we can restrict the number of consecutive applications of the cyc- and perm-rules.

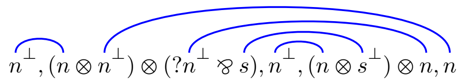
a gerund phrase (or gerundive verb):

articles, reviewers	::	n	
that	::	$(n\backslash n)/(!n\backslash s)$ or $(n\backslash n)/(s/!n)$	
disappeared	::	$(n\backslash s)$	
read, accepted	::	$(n\backslash s)/n$	(4)
reading, skimming	::	gp/n	
yesterday	::	$(n\backslash s)\backslash(n\backslash s)$	
without, after	::	$((n\backslash s)\backslash(n\backslash s))/gp$ or $((n\backslash s)\backslash(n\backslash s))/n$	

The syntactic categories of "that" take care of its roles, as a subject or an object relative pronoun. In its subject role in (f), it needs a sentence with a missing noun somewhere at its front (before its verb), expressed by the \ connective in its subtype $(!n\backslash s)$. In the object role in (g), it is expecting a sentence with a missing noun somewhere at is rear (after its verb), expressed by the use of / in its subtype $(s/!n)$. In either case, $!n$ indicates that the missing noun can be any where (before or after the verb, respectively) and any number of times. This reasoning about the subtype $(s/!n)$ is expressible using the cut-rule, as shown in the derivation of (g) below:

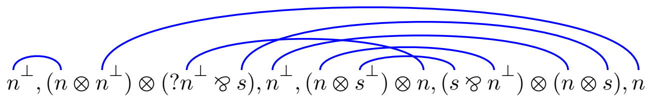
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n\vdash n}\,\mathsf{id} \qquad \dfrac{\dfrac{}{n\vdash n}\,\mathsf{id}\quad \dfrac{}{s\vdash s}\,\mathsf{id}}{n, n\backslash s\vdash s}\,\backslash_{\mathsf{L}}}{n,(n\backslash s)/n,n\vdash s}\,/_{\mathsf{L}}}{n,(n\backslash s)/n,!n\vdash s}\,!_{\mathsf{L}}}{n,(n\backslash s)/n\vdash s/!n}\,/_{\mathsf{R}} \qquad \dfrac{\dfrac{}{s/!n\vdash s/!n}\,\mathsf{id} \qquad \dfrac{\dfrac{}{n\vdash n}\,\mathsf{id}\quad \dfrac{}{n\vdash n}\,\mathsf{id}}{n,n\backslash n\vdash n}\,\backslash_{\mathsf{L}}}{n,(n\backslash n)/(s/!n),(s/!n)\vdash n}\,/_{\mathsf{L}}}{n,(n\backslash n)/(s/!n),n,(n\backslash s)/n\vdash n}\,\mathsf{cut}$$

After eliminating the cut-rule, the first rule that is applied is the $/_{\mathsf{L}}$-rule, which leads to a linguistically unintuitive proof. The reason ! is used in the type of the relative pronoun rather than the type of the noun is since nouns takes other – non contractive – roles in the absence of "that". The proof net corresponding to either derivation, however, is the same and is shown below:


$n^{\perp},(n\otimes n^{\perp})\otimes(?n^{\perp}\parr s),n^{\perp},(n\otimes s^{\perp})\otimes n,n$

In (a), we have an extraction from the middle of the incomplete sentence and need to permute the missing noun, as shown below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{as before}}{n,(n\backslash s)/n,n,(n\backslash s)\backslash(n\backslash s)\vdash s}\,/_{\mathsf{L}}}{n,(n\backslash s)/n,!n,(n\backslash s)\backslash(n\backslash s)\vdash s}\,!_{\mathsf{L}}}{n,(n\backslash s)/n,(n\backslash s)\backslash(n\backslash s),!n\vdash s}\,\mathsf{perm}_2}{n,(n\backslash s)/n,(n\backslash s)\backslash(n\backslash s)\vdash s/!n}\,/_{\mathsf{R}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{n\vdash n}\,id}{!n\vdash !n}\,!_{\mathsf{R}} \quad \dfrac{}{s\vdash s}\,id}{s/!n,!n\vdash s}\,/_{\mathsf{R}} \qquad \dfrac{\dfrac{}{n\vdash n}\,id \quad \dfrac{}{n\vdash n}\,id}{n,n\backslash n\vdash n}\,\backslash_{\mathsf{L}}}{n,(n\backslash n)/(s/!n),s/!n\vdash n}\,/_{\mathsf{L}}}{n,(n\backslash n)/(s/!n),n,(n\backslash s)/n,(n\backslash s)\backslash(n\backslash s)\vdash n}\,\mathsf{cut}$$


$n^{\perp},(n\otimes n^{\perp})\otimes(?n^{\perp}\parr s),n^{\perp},(n\otimes s^{\perp})\otimes n,(s\parr n^{\perp})\otimes(n\otimes s),n$

For (b) we need contraction as well as permutation, see the derivation below. The cut-free variant of this derivation is our running Example 2.3 and its proof net is provided in (2).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n \vdash n}\,id \quad \dfrac{\dfrac{\text{as before}}{n, n\backslash s, ((n\backslash s)\backslash(n\backslash s))/gp, gp \vdash s}\,/_{\mathsf{L}}}{n, n\backslash s, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n \vdash s}\,/_{\mathsf{L}}}{n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n \vdash s}\,/_{\mathsf{L}}}{n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n \vdash s}\,!_{\mathsf{L}}}{n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n \vdash s}\,!_{\mathsf{L}}}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, !n \vdash s}\,\mathsf{perm}_2}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n \vdash s}\,\mathsf{con}}{n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n \vdash s/!n}\,/_{\mathsf{R}} \quad \dfrac{\text{as before}}{n, (n\backslash n)/(s/!n), s/!n \vdash n}\,/_{\mathsf{L}}}{n, (n\backslash n)/(s/!n), n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n \vdash n}\,\mathsf{cut}$$

When there is more than one parasitic extraction, as in (c), we need more applications of the con and $!_{\mathsf{L}}$ rules. See below for an abridged derivation and the appendix for the full tree and the proof net:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{as above}}{\Upsilon, n, \Phi, n, \Psi, n \vdash s}\,/_{\mathsf{L}}}{\Upsilon, !n, \Phi, !n, \Psi, !n \vdash s}\,!_{\mathsf{L}} \times 3}{\Upsilon, \Phi, \Psi, !n, !n, !n \vdash s}\,\mathsf{perm}_2 \times 2}{\Upsilon, \Phi, \Psi, !n \vdash s}\,\mathsf{con} \times 2}{\Upsilon, \Phi, \Psi \vdash s/!n}\,/_{\mathsf{R}} \quad \dfrac{\text{as above}}{n, (n\backslash n)/(s/!n), s/!n \vdash n}\,/_{\mathsf{L}}}{n, (n\backslash n)/(s/!n), \underbrace{n, (n\backslash s)/n}_{\Upsilon}, \underbrace{((n\backslash s)\backslash(n\backslash s))/gp, gp/n}_{\Phi}, \underbrace{((n\backslash s)\backslash(n\backslash s))/gp, gp/n}_{\Psi} \vdash n}\,\mathsf{cut}$$

Relative clauses with parasitic gaps and more adverbs, as in "articles that reviewers accepted $-_1$ immediately without reading $-_2$ properly" are derivable with more, in this case 2, applications of perm_2 rule. Cases when the gap is in the co-argument, as in (d) and (f), are also derivable. Other types of extraction are also derivable, e.g. infinitival, as in "men that John assured Mary to be reliable most of the time". We present some examples in the appendix.

8 Discussion and Conclusion

We developed an extension of Lambek Calculus with the Linear Logic exponential, where it is only applicable to atoms. The system has a cut-elimination theorem and proof nets, and is decidable.

Our system can successfully model a range of parasitic gap phenomena. However, whenever a logic is applied to natural language, there is a risk of over generation and our system is not exempt from it. Traditionally, over generation is overcome by adding key-and-lock *bracket* modalities to modal Lambek calculi [2,24]. Modal Lambek Calculi without bracket modalities [15,13,14,23] will all naturally over generate. The dilemma is that existing systems with both contraction and bracket modalities are undecidable [16]. We conjecture that our system extended with bracket modalities is still decidable. We are also currently investigating the possibility of developing proof nets for this extended system, along the work of [30].

References

1. Vito Michele Abrusci and Paul Ruet. Non-commutative logic I: The multiplicative fragment. *Annals of Pure and Applied Logic*, 101:29–64, 2000.
2. Guy Barry, Mark Hepple, Neil Leslie, and Glyn Morrill. Proof figures and structural operators for categorial grammar. In Jürgen Kunze and Dorothee Reimann, editors, *EACL 1991, 5th Conference of the European Chapter of the Association for Computational Linguistics, April 9-11, 1991, Congress Hall, Alexanderplatz, Berlin, Germany*, pages 198–203. The Association for Computer Linguistics, 1991.
3. P. Culicover and P. Postal, editors. *Parasitic Gaps*. MIT Press, 2001.
4. Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Arch. Math. Log.*, 28(3):181–203, 1989.
5. Elisabet Engdahl. Parasitic gaps. *Linguistics and Philosophy*, 6(1):5–34, 1983.
6. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
7. Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.
8. Alessio Guglielmi. A system of interaction and structure. *ACM Trans. on Computational Logic*, 8(1):1–64, 2007.
9. Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag, 2001.
10. Willem Heijltjes and Robin Houston. Proof equivalence in MLL is PSPACE-complete. *Logical Methods in Computer Science*, Volume 12, Issue 1, March 2016.
11. Gerhard Jäger. *Anaphora and type logical grammar*, volume 24. Springer Science &amp; Business Media, 2006.
12. Makoto Kanazawa. The lambek calculus enriched with additional connectives. *Journal of Logic, Language, and Information*, 1(2):141–171, 1992.
13. Max I. Kanovich, Stepan L. Kuznetsov, Vivek Nigam, and Andre Scedrov. Subexponentials in non-commutative linear logic. *Math. Struct. Comput. Sci.*, 29(8):1217–1249, 2019.
14. Max I. Kanovich, Stepan L. Kuznetsov, Vivek Nigam, and Andre Scedrov. Soft subexponentials and multiplexing. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I*, volume 12166 of *Lecture Notes in Computer Science*, pages 500–517. Springer, 2020.
15. Max I. Kanovich, Stepan L. Kuznetsov, and Andre Scedrov. Undecidability of the lambek calculus with a relevant modality. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla, editors, *Formal Grammar - 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*, volume 9804 of *Lecture Notes in Computer Science*, pages 240–256. Springer, 2016.
16. Max I. Kanovich, Stepan L. Kuznetsov, and Andre Scedrov. Undecidability of the lambek calculus with subexponential and bracket modalities. In Ralf Klasing and Marc Zeitoun, editors, *Fundamentals of Computation Theory - 21st International Symposium, FCT 2017, Bordeaux, France, September 11-13, 2017, Proceedings*, volume 10472 of *Lecture Notes in Computer Science*, pages 326–340. Springer, 2017.
17. Max I. Kanovich, Stepan L. Kuznetsov, and Andre Scedrov. Reconciling lambek's restriction, cut-elimination and substitution in the presence of exponential modalities. *J. Log. Comput.*, 30(1):239–256, 2020.

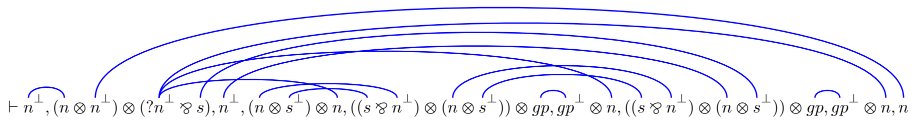
18. Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. Æthel: Automatically extracted typelogical derivations for dutch. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5257–5266. European Language Resources Association, 2020.
19. Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. SPINDLE: spinning raw text into lambda terms with graph attention. In Danilo Croce and Luca Soldaini, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. EACL 2023 - System Demonstrations, Dubrovnik, Croatia, May 2-4, 2023*, pages 128–135. Association for Computational Linguistics, 2023.
20. François Lamarche and Christian Retoré. Proof nets for the Lambek-calculus — an overview. In V. Michele Abrusci and Claudia Casadio, editors, *Proceedings of the Third Roma Workshop "Proofs and Linguistic Categories"*, pages 241–262. CLUEB, Bologna, 1996.
21. Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.
22. Ranko Lazić and Sylvain Schmitz. Nonelementary complexities for branching vass, mell, and extensions. *ACM Trans. Comput. Logic*, 16(3), jun 2015.
23. Lachlan Mcpheat, Gijs Wijnholds, Mehrnoosh Sadrzadeh, Adriana Correia, and Alexis Toumi. Anaphora and ellipsis in lambek calculus with a relevant modality: Syntax and semantics. *Journal of Cognitive Science*, 22:1–34, 2021.
24. Michael Moortgat. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3-4):349–385, 1996.
25. Michael Moortgat and Richard Oehrle. Structural abstractions. In *Proofs and Linguistic Categories: Application of Logic to the Analysis and Implementation of Natural Language, Proceedings of the 1996 Roma Workshop*, 1996.
26. Michael Moortgat, Mehrnoosh Sadrzadeh, and Gijs Wijnholds. A frobenius algebraic analysis for parasitic gaps. *FLAP*, 7(5):823–852, 2020.
27. Michael Moortgat, Mehrnoosh Sadrzadeh, and Gijs Wijnholds. A frobenius algebraic analysis for parasitic gaps. *Journal of Applied Logics*, 7:823–852, 2020.
28. Richard Moot. Semi-automated extraction of a wide-coverage type-logical grammar for french. In Philippe Langlais and Michel Gagnon, editors, *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts, TALN 2010, Montréal, Canada, July 2010*, pages 189–194. ATALA, 2010.
29. Richard Moot. The grail theorem prover: Type theory for syntax and semantics. *CoRR*, abs/1602.00812, 2016.
30. Richard Moot and Quintijn Puite. Proof nets for the multimodal lambek calculus. *Studia Logica*, 71:415–442, 2002.
31. Glyn Morrill, Neil Leslie, Mark Hepple, and Guy Barry. Categorial deductions and structural operations. *Studies in Categorial Grammar. Edinburgh Working Papers in Cognitive Science*, 1990.
32. Glyn Morrill and Oriol Valentín. Computational coverage of TLG: Nonlinearity. In *Proceedings of Third Workshop on Natural Language and Computer Science*, volume 32, pages 51–63. EasyChair Publications, 2015.
33. Glyn Morrill and Oriol Valentín. On the logic of expansion in natural language. In *Logical Aspects of Computational Linguistics*, pages 228–246. Springer, 2016.
34. Glyn Morrill, Oriol Valentín, and Mario Fadda. The displacement calculus. *Journal of Logic, Language and Information*, 20(1):1–48, 2011.

35. Vivek Nigam and Dale Miller. Algorithmic specifications in linear logic with subexponentials. In *ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP)*, pages 129–140, 2009.
36. Richard T. Oehrle. Resource-sensitivity—a brief guide. In Geert-Jan M. Kruijff and Richard T. Oehrle, editors, *Resource-Sensitivity, Binding and Anaphora*, pages 231–255. Springer Netherlands, Dordrecht, 2003.
37. Mati Pentus. Lambek calculus is np-complete. *Theor. Comput. Sci.*, 357(1-3):186–201, 2006.
38. Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Ph. de Groote and J. R. Hindley, editors, *Typed Lambda Calculus and Applications, TLCA'97*, volume 1210 of *LNCS*, pages 300–318. Springer, 1997.
39. Lutz Straßburger. System NEL is undecidable. In Ruy De Queiroz, Elaine Pimentel, and Lucília Figueiredo, editors, *10th Workshop on Logic, Language, Information and Computation (WoLLIC)*, volume 84 of *Electronic Notes in Theoretical Computer Science*, pages 166–177, 2003.
40. David N. Yetter. Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic*, 55(1):41–64, 1990.

9 Appendix

Let us first show the full proof tree of (c) "articles that reviewers accepted after skimming without reading", and the resulting proof net:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\RightLabel{$\backslash_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n\backslash s \vdash n\backslash s$}
\RightLabel{$\backslash_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s), ((n\backslash s)\backslash(n\backslash s)) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n\backslash s \vdash n\backslash s$}
\RightLabel{$\backslash_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s), ((n\backslash s)\backslash(n\backslash s)), ((n\backslash s)\backslash(n\backslash s)) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\RightLabel{$/_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s)), ((n\backslash s)\backslash(n\backslash s)) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\BinaryInfC{$n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp, ((n\backslash s)\backslash(n\backslash s)) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\RightLabel{$/_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n, ((n\backslash s)\backslash(n\backslash s)) \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$gp \vdash gp$}
\RightLabel{$/_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp \vdash s$}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$n \vdash n$}
\RightLabel{$/_{\mathsf{L}}$}
\BinaryInfC{$n, (n\backslash s)/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n \vdash s$}
\RightLabel{$!_{\mathsf{L}}$}
\UnaryInfC{$n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n \vdash s$}
\RightLabel{$!_{\mathsf{L}}$}
\UnaryInfC{$n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, n \vdash s$}
\RightLabel{$!_{\mathsf{L}}$}
\UnaryInfC{$n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n \vdash s$}
\RightLabel{perm_2}
\UnaryInfC{$n, (n\backslash s)/n, !n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, !n \vdash s$}
\RightLabel{perm_2}
\UnaryInfC{$n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, !n, !n \vdash s$}
\RightLabel{con}
\UnaryInfC{$n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n, !n \vdash s$}
\RightLabel{con}
\UnaryInfC{$n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, !n \vdash s$}
\RightLabel{$/_{\mathsf{R}}$}
\UnaryInfC{$n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n \vdash s/!n$}
\AxiomC{as before}
\RightLabel{$/_{\mathsf{L}}$}
\UnaryInfC{$n, (n\backslash n)/(s/!n), s/!n \vdash n$}
\RightLabel{cut}
\BinaryInfC{$n, (n\backslash n)/(s/!n), n, (n\backslash s)/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n, ((n\backslash s)\backslash(n\backslash s))/gp, gp/n \vdash n$}
\end{prooftree}
$$

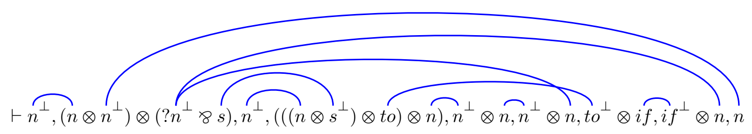

$\vdash n^\perp, (n \otimes n^\perp) \otimes (?n^\perp \parr s), n^\perp, (n \otimes s^\perp) \otimes n, ((s \parr n^\perp) \otimes (n \otimes s^\perp)) \otimes gp, gp^\perp \otimes n, ((s \parr n^\perp) \otimes (n \otimes s^\perp)) \otimes gp, gp^\perp \otimes n, n$

In order to model (e) and (f) we need new atoms. We use the atoms used in [27] for adjective phrases and infinitives of verbs. A few shortcuts are also implemented to keep the size of the proof trees manageable. For instance instead of typing "a" separately and "review" and "blog post" separately, we assign the type n to "a review", similarly to "a blog post" in one step. The new assignments are presented below:

chairs, a blog post, a review	::	n	
famous	::	ap	
accept	::	if/n	
persuaded	::	$((n\backslash s)/\mathit{to})/n$	
made	::	$((n\backslash s)/ap)/n$	(5)
every	::	(n/n)	
to	::	to/if	
in, about	::	$(n\backslash n)/n$	

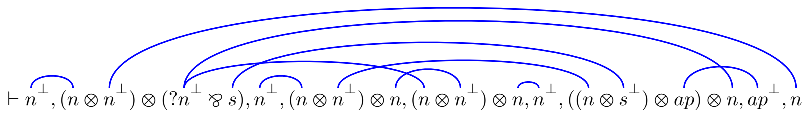
Here are the proof tree and proof net for (d) "articles that chairs persuaded every reviewer of to accept":

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n\vdash n}\,\mathit{id} \quad \dfrac{}{s\vdash s}\,\mathit{id}}{n, ((n\backslash s)\vdash s}\,\backslash_{\mathsf{L}} \quad \dfrac{}{\mathit{to}\vdash \mathit{to}}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to}), \mathit{to}\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to})/n), n, \mathit{to}\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n, \mathit{to}\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, n, \mathit{to}\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{\mathit{if}\vdash \mathit{if}}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, n, \mathit{to}/\mathit{if}, \mathit{if}\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, n, \mathit{to}/\mathit{if}, \mathit{if}/n, n\vdash s}\,/_{\mathsf{L}}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, !n, \mathit{to}/\mathit{if}, \mathit{if}/n, !n\vdash s}\,!_{\mathsf{L}}\times 2}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, \mathit{to}/\mathit{if}, \mathit{if}/n, !n, !n\vdash s}\,\mathsf{perm}_2}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, \mathit{to}/\mathit{if}, \mathit{if}/n, !n\vdash s}\,\mathsf{con}}{n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, \mathit{to}/\mathit{if}, \mathit{if}/n\vdash s/!n}\,/_{\mathsf{R}} \quad \dfrac{\text{as before}}{n, (n\backslash n)/(s/!n), s/!n\vdash n}\,/_{\mathsf{L}}}{n, (n\backslash n)/(s/!n), n, ((n\backslash s)/\mathit{to})/n), n/n, n/n, \mathit{to}/\mathit{if}, \mathit{if}/n\vdash n}\,\mathsf{cut}$$


$\vdash n^{\perp}, (n\otimes n^{\perp})\otimes(?n^{\perp}\parr s), n^{\perp}, (((n\otimes s^{\perp})\otimes \mathit{to})\otimes n), n^{\perp}\otimes n, n^{\perp}\otimes n, \mathit{to}^{\perp}\otimes \mathit{if}, \mathit{if}^{\perp}\otimes n, n$

Finally, here are the proof tree and prof net for (e) "articles that a review about in a blog post made famous":

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n\vdash n}\,\mathit{id} \quad \dfrac{\dfrac{}{n\vdash n}\,\mathit{id} \quad \dfrac{}{s\vdash s}\,\mathit{id}}{n, (n\backslash s)\vdash s}\,\backslash_{\mathsf{L}}}{n, (n\backslash n), (n\backslash s)\vdash s}\,\backslash_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, (n\backslash n)/n, n, (n\backslash s)\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, (n\backslash n)/n, n, n\backslash n, (n\backslash s)\vdash s}\,\backslash_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, (n\backslash n)/n, n, (n\backslash n)/n, n, (n\backslash s)\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{ap\vdash ap}\,\mathit{id}}{n, (n\backslash n)/n, n, (n\backslash n)/n, n, ((n\backslash s)/ap), ap\vdash s}\,/_{\mathsf{L}} \quad \dfrac{}{n\vdash n}\,\mathit{id}}{n, (n\backslash n)/n, n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, n, ap\vdash s}\,/_{\mathsf{L}}}{n, (n\backslash n)/n, !n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, !n, ap\vdash s}\,!_{\mathsf{L}}\times 2}{n, (n\backslash n)/n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, ap, !n, !n\vdash s}\,\mathsf{perm}_2}{n, (n\backslash n)/n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, ap, !n\vdash s}\,\mathsf{con}}{n, (n\backslash n)/n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, ap\vdash s/!n}\,/_{\mathsf{R}} \quad \dfrac{\text{as before}}{n, (n\backslash n)/(s/!n), s/!n\vdash n}\,/_{\mathsf{L}}}{n, (n\backslash n)/(s/!n), n, (n\backslash n)/n, (n\backslash n)/n, n, ((n\backslash s)/ap)/n, ap\vdash n}\,\mathsf{cut}$$



$$\vdash n^{\perp}, (n \otimes n^{\perp}) \otimes (?n^{\perp} \parr s), n^{\perp}, (n \otimes n^{\perp}) \otimes n, (n \otimes n^{\perp}) \otimes n, n^{\perp}, ((n \otimes s^{\perp}) \otimes ap) \otimes n, ap^{\perp}, n$$