A Logical Framework with Commutative and Non-Commutative Subexponentials

Max Kanovich[1,2], Stepan Kuznetsov[3,2], Vivek Nigam[4,5], and Andre Scedrov[6,2]

[1] University College London, London, UK, m.kanovich@ucl.ac.uk
[2] National Research University Higher School of Economics, Moscow, Russia
[3] Steklov Mathematical Institute, Moscow, Russia, sk@mi.ras.ru
[4] Federal University of Paraíba, Brazil
[5] fortiss, Germany, nigam@fortiss.org
[6] University of Pennsylvania, USA, scedrov@math.upenn.edu

Abstract. Logical frameworks allow the specification of deductive systems using the same logical machinery. Linear logical frameworks have been successfully used for the specification of a number of computational, logics and proof systems. Its success lies on the fact that formulas can be distinguished as linear, which behave intuitively as resources, and unbounded, which behave intuitionistically. Commutative subexponentials enhance the expressiveness of linear logic frameworks by allowing the distinction of multiple contexts. These contexts may behave as multisets of formulas or sets of formulas. Motivated by applications in distributed systems and in type-logical grammar, we propose a linear logical framework containing both commutative and non-commutative subexponentials. Non-commutative subexponentials can be used to specify contexts which behave as lists, not multisets, of formulas. In addition, motivated by our applications in type-logical grammar, where the weakenening rule is disallowed, we investigate the proof theory of formulas that can only contract, but not weaken. In fact, our contraction is non-local. We demonstrate that under some conditions such formulas may be treated as unbounded formulas, which behave intuitionistically.

1 Introduction

Logical frameworks [7, 8, 13, 32, 23] have been proposed to specify deductive systems, such as proof systems [7, 13, 26, 24, 32], logics [7, 22] and operational semantics [25, 27, 29, 32]. The systems that can be encoded depend on the expressive power of the logical framework. Linear logical frameworks, based on Linear Logic [6], allow the encoding of, for example, stateful systems [32, 22]. Logical Frameworks with subexponentials allow the encoding of, for example, distributed systems [25, 27], authorization logics [22]. Ordered Logical Frameworks [29] allow the specification of systems whose behavior respects some order, for example, evaluation strategies.

One key idea [2] of logical frameworks is to distinguish formulas according to the structural rules (weakening, contraction and exchange rules) that are applicable. For example, linear logical frameworks distinguish two types of formulas: *Unbounded Formulas* which behave intuitionistically, that is, can be considered as a set of formulas and *Linear Formulas* which behave linearly, that is, should be considered as multiset

of formulas. Ordered logical frameworks also consider *Ordered Formulas* which are non-commutative, that is, can be considered as a list, not multiset, of formulas. This distinction is reflected in syntax. Linear logical frameworks have two contexts $\Theta : \Gamma$, where Θ is a set of unbounded formulas and Γ a multiset of linear[7] formulas. Ordered linear logic, on the other hand, has three contexts $\Theta : \Gamma : \Delta$ where Δ is a list of ordered formulas.

Logical Frameworks with Subexponentials refine Linear Logical Frameworks by distinguishing different types of unbounded and linear formulas. They work, therefore, on sequents with multiple contexts. This increased expressiveness allows for the specification of greater number of proof systems [26] and distributed systems [27] when compared to logical frameworks without subexponentials. However, existing logical frameworks with subexponentials do not allow ordered formulas.

Our main contribution is the logical framework SNILLF which has the following two innovations:

1. **Non-Commutative Subexponentials:** SNILLF allows both commutative and non-commutative subexponentials [10]. This means that SNILLF works not only with multiple contexts for unbounded and linear formulas, but also multiple ordered contexts. As an illustration of the power of this system, we encode a distributed system where machines have FIFO buffers storing messages received from the network;
2. **Proof Search with formulas that can contract, but not weaken:** Motivated by applications in type-logical grammar, where weakening of formulas is not allowed, SNILLF allows formulas to be marked with subexponentials that can contract, but not weaken. We classify such formulas as relevant. Relevant formulas lead to complications for proof search because contracting a formula implies that it should be necessarily used in the proof. Thus the contraction of relevant formulas involves a "don't know" non-determinism. This paper investigates the proof theory of relevant formulas. We demonstrate that in some situations it is safe (sound and complete) to consider relevant formulas as unbounded, that is, formulas that can both weaken and contract. We illustrate the use relevant formulas by using SNILLF in type-logical grammar applications.

In Section 2, we review the basic proof theory of non-commutative proof systems, namely Lambek Calculus, and subexponentials. Then in Section 3 we motivate the use of non-commutatitive subexponentials and relevant formulas with some concrete examples. Section 4 investigates the proof theory of relevant formulas. The Logical Framework SNILLF is introduced in Section 5 as a focused proof system. We revisit our main examples in Section 6. Finally, we comment related and future work in Sections 7 and 8.

2 Lambek Calculus with Subexponentials

While we assume some familiarity with Lambek Calculus [12], we review some of its proof theory. Its rules are depicted in Figure 1 contaning atomic formulas, the unit constant $\mathbf{1}$, universal quantifier $\forall$, and binary connectives: $\cdot$ (product), $\backslash$ (left division) and

[7] Or affine which can be weakened.

$$\frac{}{F \to F}\ I \qquad \frac{\Gamma_1, \Gamma_2 \to C}{\Gamma_1, \mathbf{1}, \Gamma_2 \to C}\ \mathbf{1}_L \qquad \frac{}{\to \mathbf{1}}\ \mathbf{1}_R \qquad \frac{\Pi \to G \quad \Gamma_1, F, \Gamma_2 \to C}{\Gamma_1, F/G, \Pi, \Gamma_2 \to C}\ /_L \qquad \frac{\Pi, F \to G}{\Pi \to G/F}\ /_R \qquad \frac{\Gamma_1, F\{t/x\}, \Gamma_2 \to C}{\Gamma_1, \forall x.F, \Gamma_2 \to C}\ \forall_L$$

$$\frac{\Gamma_1, F, G, \Gamma_2 \to C}{\Gamma_1, F \cdot G, \Gamma_2 \to C}\ \cdot_L \qquad \frac{\Gamma_1 \to F \quad \Gamma_2 \to G}{\Gamma_1, \Gamma_2 \to F \cdot G}\ \cdot_R \qquad \frac{\Pi \to F \quad \Gamma_1, G, \Gamma_2 \to C}{\Gamma_1, \Pi, F \setminus G, \Gamma_2 \to C}\ \setminus_L \qquad \frac{F, \Pi \to G}{\Pi \to F \setminus G}\ \setminus_R \qquad \frac{\Pi \to F\{e/x\}}{\Pi \to \forall x.F}\ \forall_R$$

Fig. 1. Cut-Free Proof System Lambek Proof System. Here $\{t/x\}$ denotes the capture avoiding substitution of x by t. Moreover, e is a fresh eigenvariable, that is, not appearing in Π and F.

/ (right division). The formulas in the sequent should be seen as lists, not multisets, of formulas. For example, the $\Gamma, F_1, F_2, \Delta \longrightarrow G$ and $\Gamma, F_2, F_1, \Delta \longrightarrow G$ are not equivalent in general as there may be a proof for one, but not for the other.

In our previous work [10], we proposed the proof system SNILL_Σ (Subexponential Non-Commutative Intuitionistic Linear Logic)[8] which extends propositional Lambek Calculus with subexponentials. Subexponentials derive from an observation from Linear Logic [5, 6, 23]. Namely, the linear logic exponentials, !, are non-canonical. That is, LL allows for an unbounded number of subexponentials, $!^{\mathsf{s}}$, indexed by elements in a set of indexes $\mathsf{s} \in \mathcal{I}$.

Formally, SNILL_Σ contains all rules in Figure 1. Furthermore, it is parametrized by a subexponential signature $\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$, where $\mathcal{W}, \mathcal{C}, \mathcal{E} \subseteq \mathcal{I}$ and $\preceq$ is a pre-order over the elements of $\mathcal{I}$ upwardly closed with respect to $\mathcal{W}, \mathcal{C}, \mathcal{E}$, that is, if $\mathsf{s}_1 \in \mathcal{W}$ and $\mathsf{s}_1 \preceq \mathsf{s}_2$, then $\mathsf{s}_2 \in \mathcal{W}$ and similar for $\mathcal{C}, \mathcal{E}$. SNILL_Σ contains the following rules:

– For each $\mathsf{s} \in \mathcal{I}$, SNILL_Σ contains the dereliction and promotion rules:

$$\frac{\Gamma_1, F, \Gamma_2 \to G}{\Gamma_1, !^{\mathsf{s}}F, \Gamma_2 \to G}\ Der \qquad \frac{!^{\mathsf{s}_1}F_1, \ldots, !^{\mathsf{s}_n}F_n \longrightarrow F}{!^{\mathsf{s}_1}F_1, \ldots, !^{\mathsf{s}_n}F_n \longrightarrow !^{\mathsf{s}}F}\ !^{\mathsf{s}}{}_R, \text{provided, } \mathsf{s} \preceq \mathsf{s}_i, 1 \leq i \leq n$$

– For each $\mathsf{w} \in \mathcal{W}$ and $\mathsf{c} \in C$, SNILL_Σ contains the rules:

$$\frac{\Gamma, \Delta \longrightarrow G}{\Gamma, !^{\mathsf{w}}F, \Delta \longrightarrow G}\ W \qquad \frac{\Gamma_1, !^{\mathsf{c}}F, \Delta, !^{\mathsf{c}}F, \Gamma_2 \to G}{\Gamma_1, !^{\mathsf{c}}F, \Delta, \Gamma_2 \to G}\ C_1 \qquad \frac{\Gamma_1, !^{\mathsf{c}}F, \Delta, !^{\mathsf{c}}F, \Gamma_2 \to G}{\Gamma_1, \Delta, !^{\mathsf{c}}F, \Gamma_2 \to G}\ C_2$$

– For each $\mathsf{e} \in \mathcal{E}$, SNILL_Σ contains the rules:

$$\frac{\Gamma_1, \Delta, !^{\mathsf{e}}F, \Gamma_2 \to C}{\Gamma_1, !^{\mathsf{e}}F, \Delta, \Gamma_2 \to C}\ E_1 \qquad \frac{\Gamma_1, !^{\mathsf{e}}F, \Delta, \Gamma_2 \to C}{\Gamma_1, \Delta, !^{\mathsf{e}}F, \Gamma_2 \to C}\ E_2$$

Intuitively, the set $\mathcal{I}$ specifies the subexponential names, $\mathcal{W}$ the subexponentials that are allowed to weaken, C the subexponentials that allow to contract, and $\mathcal{E}$ the subexponentials that allow to exchange.

Notice additionally that contraction is non-local, that is, the contracted formula can appear anywhere in left hand side of the premise.

In our previous work, we proved that the propositional fragment of SNILL_Σ admits cut-elimination. The following extends this result to first-order SNILL_Σ.

Theorem 1. *For any subexponential signature Σ, SNILL_Σ admits cut-elimination.*

[8] In that paper, the system was called SMALC.

For our applications, we will consider subexponential signatures $\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$ with the following restrictions:

$$\mathcal{W} \subseteq \mathcal{E} \qquad \text{and} \qquad \mathcal{C} \subseteq \mathcal{E}$$

That is, all subexponentials that can be weakened or contracted can also be exchanged. This restriction on subexponentials will be used to establish conditions for reducing "don't know" non-determinism as we describe in Section 4. Moreover, they are enough to specify our intended applications as described in Section 6.

In the remainder of this paper, we will elide the subexponential signature Σ whenever it is clear from the context.

Given the restriction above on subexponential signtures, we can classify formulas of the form $!^{\mathsf{s}}F$ according to the structural rules that are applicable to s:

- **Linear Formulas:** These formulas are not allowed to be contracted nor weakened, that is, subexponentials $\mathsf{s} \notin \mathcal{W} \cup \mathcal{C}$. Linear subexponentials range over $\mathsf{l}, \mathsf{l}_1, \mathsf{l}_2, \ldots$. They can be commutative when $\mathsf{l} \in \mathcal{E}$ or non-commutative otherwise;
- **Unbounded Formulas:** These formulas can be both weakened and contracted, that is, subexponentials $\mathsf{s} \in \mathcal{W} \cap \mathcal{C}$. Unbounded subexponentials range over $\mathsf{u}, \mathsf{u}_1, \mathsf{u}_2, \ldots$. As $\mathcal{W} \subseteq \mathcal{E}$, these formulas are always commutative that is $\mathsf{u} \in \mathcal{E}$;
- **Affine Formulas:** These formulas can only be weakened and not contracted, that is, subexponentials $\mathsf{s} \in \mathcal{W}$ and $\mathsf{s} \notin \mathcal{C}$; Affine subexponentials range over $\mathsf{a}, \mathsf{a}_1, \mathsf{a}_2, \ldots$. As $\mathcal{W} \subseteq \mathcal{E}$, these formulas are always commutative that is $\mathsf{a} \in \mathcal{E}$;
- **Relevant Formulas:** These formulas cannot be weakened but can be contracted, that is, subexponentials $\mathsf{s} \in \mathcal{C}, \mathsf{s} \notin \mathcal{W}$. Relevant subexponentials range over $\mathsf{r}, \mathsf{r}_1, \mathsf{r}_2, \ldots$. As $\mathcal{C} \subseteq \mathcal{E}$, these formulas are always commutative that is $\mathsf{r} \in \mathcal{E}$.

Logical frameworks have been proposed with unbounded, linear and affine formulas, but without relevant formulas. To illustrate the difficulty involving relevant formulas, consider the following derivations with an instance of the dot rule and contraction rules. In the derivation to the left, only the formula $!^{\mathsf{u}}F$ is contracted, while in the right the formula $!^{\mathsf{r}}H$ is also contracted.

$$\dfrac{\dfrac{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma \longrightarrow G_1 \quad !^{\mathsf{u}}F, \Delta \longrightarrow G_2}{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma, !^{\mathsf{u}}F, \Delta \longrightarrow G_1 \cdot G_2}\otimes_R}{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma, \Delta \longrightarrow G_1 \cdot G_2}\, C \qquad \dfrac{\dfrac{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma \longrightarrow G_1 \quad !^{\mathsf{u}}F, !^{\mathsf{r}}H, \Delta \longrightarrow G_2}{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma, !^{\mathsf{u}}F, !^{\mathsf{r}}H, \Delta \longrightarrow G_1 \cdot G_2}\otimes_R}{!^{\mathsf{u}}F, !^{\mathsf{r}}H, \Gamma, \Delta \longrightarrow G_1 \cdot G_2}\, 2 \times C$$

As unbounded formulas can always be weakened, it is always safe to contract them. If the contracted formula is needed then it can be used and if it turns out not to be needed, the unbounded formula can be weakened before applying the initial rule. Thus, a collection of unbounded formulas can be safely treated as a set of formulas. *This means that the non-determinism due to unbounded formulas is a don't care non-determinism.*

The same is not the case for relevant formulas. As these formulas cannot be weakened, provability may depend on whether one contracts a relevant formula or not. For example, in the derivation to the right, the formula $!^{\mathsf{r}}H$ has to be necessarily used in both premises, while in the derivation to the left, the formula $!^{\mathsf{r}}H$ can only be used in the left premise. *This means that the choice of contracting a relevant formula or not involves a don't know non-determinism.*

3 Examples

We detail two different domain applications for which SNILLF can be applied. The first is on the specification of distributed systems. The second is on type-logical grammar.

3.1 Distributed Systems Semantics

Computer systems work with data structures which behave as sets, multisets and as lists. As an example, consider a system with n machines called $\mathsf{m}_1, \ldots, \mathsf{m}_n$. 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.

A buffer at machine m_i with elements Γ_i is specified as the list of formulas where start and end mark the start and end of the list $[\mathsf{start}, \Gamma_i, \mathsf{end}]_{\mathsf{m}i}$. Thus a system with n machines is specified as the collection of contexts of the form which are associated to non-commutative subexponentials $\mathsf{m}1, \ldots, \mathsf{m}n$, respectively:

$$[\mathsf{start}, \Gamma_1, \mathsf{end}]_{\mathsf{m}1}\ [\mathsf{start}, \Gamma_2, \mathsf{end}]_{\mathsf{m}2}\ \cdots [\mathsf{start}, \Gamma_n, \mathsf{end}]_{\mathsf{m}n}$$

As we describe in detail in Section 6, since these contexts behave as lists, the order of the elements of the buffers is important allowing one to specify the correct FIFO behavior of such buffers.

3.2 Type-Logical Grammar

The Lambek calculus was initially designed by Joachim Lambek [12] as a basic logic in a framework for describing natural language syntax. The idea of such frameworks goes back to works of Ajdukiewicz [1] and Bar-Hillel [3]; nowadays formal grammars of such sort are called *type-logical,* or *categorial* grammars.

The idea of a type-logical grammar is simple: the central part of the grammar is the *lexicon,* a finite binary correspondence $\triangleright$ between words of the language and formulae of the basic logic (such as Lambek Calculus) . These formulae are also called *syntactic categories,* or *types.* Thus, in this framework the grammar is fully *lexicalised, i.e.,* all syntactic information is kept in the types associated to words, and one does not need to formulate "global" syntactic rules like "a sentence is a combination of a noun phrase and a verb phrase." The second component of a type-logical grammar is the *goal type.* Usually it is a designated variable (*primitive type*) S (meaning "sentence").

A sentence $w = a_1\, a_2\, \ldots\, a_n$ is accepted by the grammar, if there exist such formulae $F_1, F_2, \ldots, F_n$ that $a_i \triangleright F_i$ for $1 \leq i \leq n$ and the sequent $F_1, F_2, \ldots, F_n \to S$ is derivable. The language generated by the grammar is defined as the set of all accepted sentences.

As shown by Pentus [28], grammars based on the Lambek calculus can generate only context-free languages. It is known, however, that certain natural language structures are beyond the context-free formalism (as discussed, for example, by Shieber [31] on Swiss German material). This also served as motivation for extending the Lambek calculus with extra connectives, in particular, subexponential modalities.

In order to show how a subexponential connective can be useful in type-logical grammar, let us consider the following series of examples. The syntactic analysis shown

in these examples is due to Morrill and Valentín [19]. In our toy grammar for a small fragment of English we associate the following types to words:

John, *Mary* $\triangleright N$	(noun phrase)
loves, *signed* $\triangleright N \setminus S / N$	(transitive verb)
girl, *paper* $\triangleright CN$	(common noun)
the $\triangleright N / CN$	(article: transforms a common noun into a noun phrase)
without $\triangleright (N \setminus S) \backslash (N \setminus S) / GC$	
reading $\triangleright GC / N$	("*reading the paper*" is a gerund clause, GC)
that, *whom* $\triangleright (CN \setminus CN) / (S / !^{\mathsf{s}} N)$	(dependent clause coordinator)

The simplest example, *"John loves Mary,"* is justified as a correct sentence (of type S) by the following derivation in Lambek calculus:

$$\dfrac{N \to N \quad \dfrac{N \to N \quad S \to S}{N, N \setminus S \to S}}{N, N \setminus S / N, N \to S}$$

There are more sophisticated syntactic constructions for which the *contraction* rule is used. First consider the following sentence: *"John signed the paper without reading it"* (of type S), supported by the following Lambek derivation:

$$\dfrac{CN \to CN \quad \dfrac{N \to N \quad \dfrac{GC / N, N \to GC \quad N, N \setminus S, (N \setminus S) \backslash (N \setminus S) \to S}{N, N \setminus S, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, N \to S}}{N, N \setminus S / N, N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, N \to S}}{N, N \setminus S / N, N / CN, CN, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, N \to S}$$

Now let us transform this sentence into a dependent clause: *"the paper that John signed without reading"* (this phrase should be of type N, noun phrase). Notice that here we removed not only *"the paper,"* but also *"it,"* forming two gaps which should be filled with the same $!^{\mathsf{s}} N$. This phenomenon is called *parasitic extraction* and can be handled using dereliction, exchange and contraction:

$$\dfrac{\dfrac{\dfrac{\dfrac{N, N \setminus S / N, N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, N \to S}{N, N \setminus S / N, !^{\mathsf{s}} N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, !^{\mathsf{s}} N \to S}\, Der}{N, N \setminus S / N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N, !^{\mathsf{s}} N \to S}\, C_L}{N, N \setminus S / N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N \to S / !^{\mathsf{s}} N} \qquad N / CN, CN, CN \setminus CN \to N}{N / CN, CN, (CN \setminus CN) / (S / !^{\mathsf{s}} N), N, N \setminus S / N, (N \setminus S) \backslash (N \setminus S) / GC, GC / N \to N}$$

Contraction can be used several times, generating examples like *"the paper that the editor of received, but left in the office without reading."*

Finally, the last example shows that *weakening* should not be allowed. Consider *"the girl whom John loves Mary."* This should not be a legal noun phrase, but can be derived using weakening:

$$\dfrac{\dfrac{\dfrac{N, N \setminus S / N, N \to S}{N, N \setminus S / N, N, !^{\mathsf{s}} N \to S}\, W_L}{N, N \setminus S / N, N \to S / !^{\mathsf{s}} N} \qquad N / CN, CN, CN \setminus CN \to N}{N / CN, CN, (CN \setminus CN) / (S / !^{\mathsf{s}} N), N, N \setminus S / N, N \to N}$$

Thus, the subexponential used for type-logical grammar is a *relevant* one; in other words, $\mathsf{s} \in \mathcal{E}$, $\mathsf{s} \in \mathcal{C}$, $\mathsf{s} \notin \mathcal{W}$.

4 Treating Relevant Formulas as Unbounded Formulas

Given that contraction of relevant formulas involves "don't know non-determinism", during proof search, we would like to postpone (from a bottom-up perspective) as much as possible the application of contraction of relevant formulas. The following lemma provides us with insight on which rules are problematic:

Lemma 1. *Contraction rules permute over all rules except rules* $\cdot_R, \backslash_L, /_L$ *and Der.*

For proof search, this means that for rules R other than $\cdot_R, \backslash_L, /_L$ and *Der*, it is safe to not contract relevant formulas. This is because from the lemma above, if there is a proof where a formula is contracted before the application of R, then there is also a proof where the formula is contracted after R.

However, the same is not the case for $\cdot_R, \backslash_L, /_L$ and *Der*. For example, it is not possible to permute contraction over $\backslash_L$ in the following derivation as the occurrences of $!^{\mathsf{r}}F$ are split among the premises:

$$\dfrac{\dfrac{\Pi_1, !^{\mathsf{r}}F, \Pi_2 \longrightarrow F_1 \qquad \Gamma_1, !^{\mathsf{r}}F, \Gamma_2, F_2, \Gamma_3 \longrightarrow G}{\Gamma_1, !^{\mathsf{r}}F, \Gamma_2, \Pi_1, !^{\mathsf{r}}F, \Pi_2, F_1 \backslash F_2, \Gamma_3 \longrightarrow G}\, \backslash_L}{\Gamma_1, \Gamma_2, \Pi_1, !^{\mathsf{r}}F, \Pi_2, F_1 \backslash F_2, \Gamma_3 \longrightarrow G}\, C_L$$

We analyse the rules $\cdot_R, \backslash_L, /_L$ and *Der* individually and investigate how to reduce don't know non-determinism.

Consider the following derivation to the left containing an instance of $\cdot_R$ rule where r is a relevant formula and the relevant formula $!^{\mathsf{r}}H$ is moved to the right premise. The symmetric reasoning applies if $!^{\mathsf{r}}H$ is moved to the left premise.

$$\dfrac{\Gamma_1 \to F \quad \Gamma_2, !^{\mathsf{r}}H, \Gamma_3 \to G}{\Gamma_1, \Gamma_2, !^{\mathsf{r}}H, \Gamma_3 \to F \cdot G}\, \cdot_R \qquad \dfrac{\dfrac{\Gamma_1' \to F \quad \Gamma_2, !^{\mathsf{r}}H, \Gamma_3 \to G}{\Gamma_1', \Gamma_2, !^{\mathsf{r}}H, \Gamma_3 \to F \cdot G}\, \cdot_R}{\Gamma_1, \Gamma_2, !^{\mathsf{r}}H, \Gamma_3 \to F \cdot G}\, n \times C_L$$

As $!^{\mathsf{r}}H$ cannot be weakened, it should be necessarily used in the right premise. That is, it behaves as a linear formula. How about the left premise? Since contraction is not local, it is possible to contract $!^{\mathsf{r}}H$ as many times such that the contracted formulas are moved to the left premise. This means that during proof search, it is safe to consider the formula H unbounded in the left premise. If n copies of H are used in the proof of the left premise, where $n \geq 0$, we can contract it as illustrates the derivation above to the right where Γ_1' contains the contracted occurrences of the formula $!^{\mathsf{r}}H$.

Similarly, consider the following instance of $\backslash_L$ to the left where the relevant formula $!^{\mathsf{r}}H$ is moved to the left premise. A symmetric observation can be carried out for $/_L$.

$$\dfrac{\Pi_1, !^{\mathsf{r}}H, \Pi_2 \to F \quad \Gamma_1, G, \Gamma_2 \to C}{\Gamma_1, \Pi_1, !^{\mathsf{r}}H, \Pi_2 F \setminus G, \Gamma_2 \to C}\, \backslash_L \qquad \dfrac{\dfrac{\Pi_1, !^{\mathsf{r}}H, \Pi_2, \to F \quad \Gamma_1', G, \Gamma_2' \to C}{\Gamma_1', \Pi_1, !^{\mathsf{r}}H, \Pi_2, F \setminus G, \Gamma_2' \to C}\, \backslash_L}{\Gamma_1, \Pi_1, !^{\mathsf{r}}H, \Pi_2, F \setminus G, \Gamma_2 \to C}\, n \times C_L$$

As before, since $!^{\mathsf{r}}H$ cannot be weakened, it should be necessarily used in the left premise. That is, it behaves like a linear non-commutative formula. Following a similar

reasoning as for $\cdot$, we can treat this formula as unbounded in the right premise. Since contractions are non-local, we can copy $!^{\mathsf{r}}H$ so that they are moved to the right premise as illustrates the derivation above to the right where Γ_1', Γ_2' contain the contracted occurrences of the formula $!^{\mathsf{r}}H$.

The same reasoning applies for relevant formulas moved to the right premise It is safe to consider the formula H as unbounded in the left premise.

The leads to the our first key observation:

Key Observation 1: *During proof search, any relevant formula moved to one premise of $\cdot_R, \backslash_L, /_L$ can be considered unbounded in the other premise.*

Finally, consider the following instance of Der_L on a relevant formula:

$$\frac{\Gamma_1, H, \Gamma_2 \longrightarrow G}{\Gamma_1, !^{\mathsf{r}}H, \Gamma_2 \longrightarrow G}\ Der$$

Applying the same reasoning as above, the formula $!^{\mathsf{r}}H$ can be treated as unbounded as one can make as many copies as needed before the dereliction. This leads to the following key observation:

Key Observation 2: *During proof search, any relevant formula derelicted by Der can be considered unbounded in its premise.*

Example 1. Consider the derivation below left with the relevant formula $!^{\mathsf{r}}A$:

$$\frac{\dfrac{}{!^{\mathsf{r}}A \longrightarrow A}\ Der, I \qquad A' \longrightarrow A \cdot A' \cdot A}{!^{\mathsf{r}}A, A \setminus A' \longrightarrow A \cdot A' \cdot A}\ \backslash_L \qquad \frac{\dfrac{}{!^{\mathsf{r}}A \longrightarrow A}\ Der, I \qquad \underline{!^{\mathsf{r}}A}, A' \longrightarrow A \cdot A' \cdot A}{!^{\mathsf{r}}A, A \setminus A' \longrightarrow A \cdot A' \cdot A}\ \backslash_L$$

Following the Key Observation 1 above, as $!^{\mathsf{r}}A$ is moved to the left premise, we can treat $!^{\mathsf{r}}A$ as unbounded in the right premise. This is denoted by the formula $\underline{!^{\mathsf{r}}A}$ as shown in the derivation to the right. We can now prove the right premise using $\underline{!^{\mathsf{r}}A}$ as illustrated by the derivation Ξ below. (Recall unbounded formulas can be contracted safely):

$$\Xi = \frac{\dfrac{}{\underline{!^{\mathsf{r}}A} \longrightarrow A}\ Der, I \qquad \dfrac{\dfrac{}{A' \longrightarrow A'}\ I}{\underline{!^{\mathsf{r}}A}, A' \longrightarrow A'}\ W_L \qquad \dfrac{}{\underline{!^{\mathsf{r}}A} \longrightarrow A}\ Der, I}{\underline{!^{\mathsf{r}}A}, A' \longrightarrow A \cdot A' \cdot A}\ 2 \times \cdot_R$$

Notice that it may seem unsound to weaken $\underline{!^{\mathsf{r}}A}$ in the middle branch. However, as we can control the number of times $!^{\mathsf{r}}A$ is contracted, we can transform this derivation into a SNILL proof: In particular, we can infer from Ξ that we require two copies of $!^{\mathsf{r}}A$. Thus the corresponding SNILL proof starts with two contractions:

$$\frac{\dfrac{\dfrac{}{!^{\mathsf{r}}A \longrightarrow A}\ Der, I \qquad !^{\mathsf{r}}A, A', !^{\mathsf{r}}A \longrightarrow A \cdot A' \cdot A}{!^{\mathsf{r}}A, !^{\mathsf{r}}A, A \setminus A', !^{\mathsf{r}}A \longrightarrow A \cdot A' \cdot A}\ \backslash_L}{!^{\mathsf{r}}A, A \setminus A' \longrightarrow A \cdot A' \cdot A}\ 2 \times C_L$$

It is now a matter to construct a proof based on Ξ.

Example 2. Given that we allow non-local contractions, one could imagine whether Key Observation 1 would also work for non-commutative relevant subexponentials s such that $\mathsf{s} \in C$ and $\mathsf{s} \notin \mathcal{E} \cup \mathcal{W}$. However this is not true in general. Consider the following derivation where we attempt to use Key Observation 1, that is, where $\underline{!^{\mathsf{s}}A}$ is treated as an unbounded formula:

$$\dfrac{\dfrac{}{!^{\mathsf{s}}A \longrightarrow A} \quad \dfrac{\dfrac{\dfrac{}{\underline{!^{\mathsf{s}}A}, A_1, A_2 \longrightarrow A_1 \cdot A \cdot A_2}}{\underline{!^{\mathsf{s}}A}, A_1 \cdot A_2 \longrightarrow A_1 \cdot A \cdot A_2}}{\underline{!^{\mathsf{s}}A} \longrightarrow (A_1 \cdot A_2 \,/\, A_1 \cdot A \cdot A_2)}}{!^{\mathsf{s}}A \longrightarrow A \cdot (A_1 \cdot A_2 \,/\, A_1 \cdot A \cdot A_2)}$$

In the open premise, it would be tempting to move $\underline{!^{\mathsf{s}}A}$ to the place between A_1 and A_2 and finish the "proof". However, the resulting derivation would not correspond to a valid SNILL proof as it is not possible to contract the original $!^{\mathsf{s}}A$ so that it is placed exactly between A_1 and A_2. While we conjecture that this could be solved by also recalling the places where relevant formulas can be contracted, we leave this investigation for future work. Moreover, such non-commutative relevant formulas are not needed for our applications here.

5 Focused Proof System for SNILL

Logical frameworks are defined proof theoretically by a focused proof system. This section introduces the focused proof system SNILLF for SNILL. We prove that SNILLF is sound and complete with respect to SNILL.

First proposed by Andreoli [2] for Linear Logic, focused proof systems reduce proof search space by distinguishing rules which have don't know non-determinism, classified as *positive*, from rules which have don't care non-determinism, classified as *negative*. For SNILL, the rules $\cdot_R, \backslash_L, /_L, \forall_L$ are positive rules and the rules $\cdot_L, \backslash_R, /_R, \forall_R$ are negative. Formulas of the form $F \cdot G$ and $!^{\mathsf{s}}F$ and 1 are classified as positive while the remaining formulas as negative.

SNILLF sequents are constructed using the following four types of contexts:

- **Commutative Contexts ($\mathcal{K}$):** A commutative context $\mathcal{K}$ maps a commutative subexponentials $\mathsf{s} \in \mathcal{E}$ to a set of formulas if $\mathsf{s} \in \mathcal{W} \cap C$, that is, it is unbounded, and to a multiset of formula otherwise. Intutively, such a context $\mathcal{K}$ denotes the formulas: $\mathcal{K}[\mathsf{s}_1], \mathcal{K}[\mathsf{s}_2], \ldots, \mathcal{K}[\mathsf{s}_n]$ where $\{\mathsf{s}_1, \ldots, \mathsf{s}_n\} = \mathcal{E}$;
- **Unrestricted Relevant Context ($\mathcal{R}^u$):** An unrestricted context $\mathcal{R}^u$ maps relevant subexponentials $\mathsf{r} \in C$ and $\mathsf{r} \notin \mathcal{W}$ to sets of formulas. Intuitively, this context stores the relevant formulas which can be treated as unbounded. Using the notation in Section 4, $\mathcal{R}^u$ represents the formulas $\underline{\mathcal{R}^u[\mathsf{r}_1]}, \ldots, \underline{\mathcal{R}^u[\mathsf{r}_n]}$, where $\{\mathsf{r}_1, \ldots, \mathsf{r}_n\}$ is the set of all relevant subexponentials;
- **Subexponential Boxes:** $[F_1, \ldots, F_k]_{\mathsf{s}}$ where $\mathsf{s} \notin \mathcal{E}$ and $F_1, \ldots, F_k$ is list, not multiset, of formulas. This box should be interpreted as the list of formulas $!^{\mathsf{s}}F_1, \ldots, !^{\mathsf{s}}F_k$;
- **Unmarked Boxes:** $[F_1, \ldots, F_k \Uparrow G_1, \ldots, G_m]$, where $F_1, \ldots, F_k$ and $G_1, \ldots, G_m$ are both lists, not multisets, of formulas. This box should be interpreted as the list of formulas $F_1, \ldots, F_k, G_1, \ldots, G_m$. When $m = 0$, we write such box as $[F_1, \ldots, F_k]_\star$.

We use $\mathcal{NC}$ and its variants to denote a sequence of boxed formulas (Subexponential Boxes and Unmarked Boxes). We write $\mathcal{NC}^\star$ whenever all unmarked boxes are of the form $[F_1, \ldots, F_k]_\star$. We define the set $\mathcal{NC}[\mathsf{s}] = \{F \mid [\Gamma_1, F, \Gamma_2]_\mathsf{s} \in \mathcal{NC}\}$. Also, if $\mathcal{NC}_1 = [\Gamma_1]_{\mathsf{s}_1} \cdots [\Gamma]_{\mathsf{s}_i}$ and $\mathcal{NC}_2 = [\Delta]_{\mathsf{s}_i} \cdots [\Gamma]_{\mathsf{s}_n}$, then $\mathcal{NC}_1 \cdot \mathcal{NC}_2$ is defined to be $[\Gamma_1]_{\mathsf{s}_1} \cdots [\Gamma_i, \Delta]_{\mathsf{s}_i} \cdots [\Gamma_n]_{\mathsf{s}_n}$. Empty boxes $[\cdot]_\mathsf{s}, [\cdot]_\star$ are always elided. These also act as identity elements, that is $[F_1, \ldots, F_n]_\mathsf{s} \cdot []_\mathsf{s} = [F_1, \ldots, F_n]_\mathsf{s}$ and similarly for unmarked boxes.

Finally, we define the following auxiliary operations on commutative contexts:

$$\mathcal{K}[\mathcal{S}] = \bigcup_{\mathsf{s} \in \mathcal{S}} \mathcal{K}[\mathsf{s}] \qquad (\mathcal{K} +_\mathsf{s} F)[\mathsf{s}'] = \begin{cases} \mathcal{K}[\mathsf{s}'] \uplus \{F\} & \text{if } \mathsf{s}' = \mathsf{s} \\ \mathcal{K}[\mathsf{s}'] & \text{otherwise} \end{cases}$$

$$(\mathcal{K}_1 \otimes \mathcal{K}_2)[\mathsf{s}] = \begin{cases} \mathcal{K}_1[\mathsf{s}] \uplus \mathcal{K}_2[\mathsf{s}] & \text{if } \mathsf{s} \notin \mathcal{W} \cap C \\ \mathcal{K}_1[\mathsf{s}] & \text{otherwise} \end{cases}$$

$(\mathcal{K}_1 \star \mathcal{K}_2) \mid_\mathcal{S}$ is true if and only if for all $\mathsf{s} \in \mathcal{S}, \mathcal{K}_1[\mathsf{s}] \star \mathcal{K}_2[\mathsf{s}]$, for $\star \in \{\subset, \subseteq, =\}$

$$\mathcal{K} \leq_\mathsf{s} = \begin{cases} \mathcal{K}[\mathsf{s}_1] & \text{if } \mathsf{s} \preceq \mathsf{s}_1 \\ \emptyset & \text{otherwise} \end{cases}$$

Similar operations are also defined (mutatis mutandis) for Unrestricted Relevant Contexts ($\mathcal{R}^u$). These operations are similar to the ones proposed in [23] used in the formalization of the side conditions of the rules for proof systems with subexponentials.

The rules for the focused proof system SNILLF for SNILL are depicted in Figure 2. They contain the following types of sequents:

- **Negative:** $\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}$ and $\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow F]$. Here $\mathcal{G}$ can be either $[\Uparrow F]$ or $[F]$. Moreover, Γ, Δ are lists of formulas.
- **Positive:** $\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [\Downarrow F]$ and $\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1^\star\ [\Downarrow F]\ \mathcal{NC}_2^\star \longrightarrow [G]_\mathsf{s}$. In the former, the formula F on the r.h.s. is focused on and the latter on the l.h.s.;
- **Decision:** $\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [G]$: Sequents at the border of negative and positive phases.

During the negative phase, formulas (Δ) to the right of Unmarked Boxes ($[\Gamma \Uparrow \Delta]$) are introduced or moved to the left (Γ) or to other contexts using the Reaction rules $\Uparrow_L, \Uparrow_R$. Notice the negative rule $!^{\mathsf{ne}}$. There since the formulas Δ are all not marked with subexponentials, the rule creates a new box $[\Delta]_\star$.

Once a negative phase ends, that is, all unmarked boxes are of the form $[\Gamma]_\star$, one should decide in a formula to focus on using one of the Decide Rules. Decide rules implicitly apply the Dereliction rule whenever applicable. The rules $D_\mathsf{u}, D_\mathsf{nc}, D_\mathsf{r}$ choose a formula marked with a subexponential for which exchange rule applies. Therefore, one can place F any where in the context. This D_nc which forces the formula F to be where it is. It also causes the box where the formula is to be split. Finally, notice that if an unbounded formula is focused on then it is contracted (as in Andreoli's original system). Moreover following Key Observation 2 described Section 4, whenever a relevant formula is added to the context $\mathcal{R}^u$ and is treated as an unbouded formula.

In the positive phase, one can only introduce the formula that is focused on. The rules $\backslash_L, /_L, \cdot_R$ implement the Key Observation 1 described in Section 4 . That is, all relevant formula moved to one premise are added to the $\mathcal{R}^u$ context of the other premise

Negative Phase

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow F_1, F_2, \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow F_1 \cdot F_2, \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}} \cdot_L \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow 1, \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}} 1_L$$

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}\,[\cdot \Uparrow F] \longrightarrow [\Uparrow G]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow G \,/\, F]} /_R \quad \frac{\mathcal{K} : \mathcal{R}^u : [\cdot \Uparrow F]\,\mathcal{NC} \longrightarrow [\Uparrow G]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow F \,/\, G]} \backslash_R \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow F\{x/e\}]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow \forall x.F]} \forall_R$$

$$\frac{\mathcal{K} +_{\mathsf{e}} F : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow !^{\mathsf{e}}F, \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}} !^{\mathsf{e}} \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta]_\star\, [F]_{\mathsf{ne}}\, [\Uparrow \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1, [\Delta \Uparrow !^{\mathsf{ne}}F, \Gamma], \mathcal{NC}_2 \longrightarrow \mathcal{G}} !^{\mathsf{ne}}$$

Positive Phase

$$\frac{\mathcal{K}_1 : \mathcal{R}^u \otimes \mathcal{R}_1 : \mathcal{NC}_2^\star \longrightarrow [\Downarrow F] \quad \mathcal{K}_2 : \mathcal{R}^u \otimes \mathcal{R}_2 : \mathcal{NC}_1^\star\, [\Downarrow G]\, \mathcal{NC}_3^\star \longrightarrow [H]}{\mathcal{K}_1 \otimes \mathcal{K}_2 : \mathcal{R}^u : \mathcal{NC}_1^\star \cdot \mathcal{NC}_2^\star\, [\Downarrow F \setminus G]\, \mathcal{NC}_3^\star \longrightarrow [H]} \backslash_L$$

$$\frac{\mathcal{K}_1 : \mathcal{R}^u \otimes \mathcal{R}_1 : \mathcal{NC}_2^\star \longrightarrow [\Downarrow G] \quad \mathcal{K}_2 : \mathcal{R}^u \otimes \mathcal{R}_2 : \mathcal{NC}_1^\star\, [\Downarrow F]\, \mathcal{NC}_3^\star \longrightarrow [H]}{\mathcal{K}_1 \otimes \mathcal{K}_2 : \mathcal{R}^u : \mathcal{NC}_1^\star\, [\Downarrow F \,/\, G]\, \mathcal{NC}_2^\star \cdot \mathcal{NC}_3^\star \longrightarrow [H]} /_L$$

where $\mathcal{R}_1[\mathsf{r}] = \mathcal{K}_2[\mathsf{r}]$ and $\mathcal{R}_2[\mathsf{r}] = \mathcal{K}_1[\mathsf{r}]$ for all $\mathsf{r} \in C$ and $\mathsf{r} \notin \mathcal{W}$.

$$\frac{\mathcal{K}_1 : \mathcal{R}^u \otimes \mathcal{R}_1 : \mathcal{NC}_1^\star \longrightarrow [\Downarrow F] \quad \mathcal{K}_2 : \mathcal{R}^u \otimes \mathcal{R}_2 : \mathcal{NC}_2^\star \longrightarrow [\Downarrow G]}{\mathcal{K}_1 \otimes \mathcal{K}_2 : \mathcal{R}^u : \mathcal{NC}_1^\star \cdot \mathcal{NC}_2^\star \longrightarrow [\Downarrow F \cdot G]} \cdot_R$$

where $\mathcal{R}_1[\mathsf{r}] = \mathcal{K}_2[\mathsf{r}]$ and $\mathcal{R}_2[\mathsf{r}] = \mathcal{K}_1[\mathsf{r}]$ for all $\mathsf{r} \in C$ and $\mathsf{r} \notin \mathcal{W}$.

$$\frac{}{\mathcal{K} : \mathcal{R}^u : \cdot \longrightarrow [\Downarrow 1]}\, 1_R \quad \frac{}{\mathcal{K} : \mathcal{R}^u : [\Downarrow A] \longrightarrow [A]}\, I \quad \text{where } \mathcal{K}[\mathsf{s}] = \emptyset \text{ for all } \mathsf{s} \notin \mathcal{W}$$

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1^\star\, [F\{t/x\}]\, \mathcal{NC}_2^\star \longrightarrow [H]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}_1^\star\, [\Downarrow \forall x.F]\, \mathcal{NC}_2^\star \longrightarrow [H]} \quad \frac{\mathcal{K} \leq_{\mathsf{s}} : \mathcal{R}^u \leq_{\mathsf{s}} : \mathcal{NC}^\star \longrightarrow [\Uparrow F]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [\Downarrow\, !^{\mathsf{s}}F]}\, !^{\mathsf{s}}{}_R, \text{if } \mathcal{K}[x] = \emptyset = \mathcal{NC}^\star[x] \text{ for all } \mathsf{s} \not\leq x$$

Decide Rules

$$\frac{\mathcal{K} +_{\mathsf{u}} F : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1]_{\mathsf{s}}\, [\Downarrow F]\, [\Gamma_2]_s\, \mathcal{NC}_2^\star \longrightarrow [G]}{\mathcal{K} +_{\mathsf{u}} F : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1, \Gamma_2]_{\mathsf{s}}\, \mathcal{NC}_2^\star \longrightarrow [G]}\, D_{\mathsf{u}} \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1]_{\mathsf{s}}\, [\Downarrow F]\, [\Gamma_2]_s\, \mathcal{NC}_2^\star \longrightarrow [G]}{\mathcal{K} +_{\mathsf{nc}} F : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1, \Gamma_2]_{\mathsf{s}}\, \mathcal{NC}_2^\star \longrightarrow [G]}\, D_{\mathsf{nc}}$$

$$\frac{\mathcal{K} : \mathcal{R}^u +_{\mathsf{r}} F : \mathcal{NC}^\star\, [\Gamma_1]_{\mathsf{s}}\, [\Downarrow F]\, [\Gamma_2]_s\, \mathcal{NC}_2^\star \longrightarrow [G]}{\mathcal{K} +_{\mathsf{r}} F : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1, \Gamma_2]_{\mathsf{s}}\, \mathcal{NC}_2^\star \longrightarrow [G]}\, D_{\mathsf{r}}$$

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1]_{\mathsf{s}}\, [\Downarrow F]\, [\Gamma_2]_{\mathsf{s}}\, \mathcal{NC}_2^\star \longrightarrow [G]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star\, [\Gamma_1, F, \Gamma_2]_{\mathsf{s}}\, \mathcal{NC}_2^\star \longrightarrow [G]}\, D_{\mathsf{s}} \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [\Downarrow G]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [G]}\, D_R$$

Reaction Rules

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star\, [\cdot \Uparrow P]\, \mathcal{NC}_2^\star \longrightarrow [G]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star\, [\Downarrow P]\, \mathcal{NC}_2^\star \longrightarrow [G]}\, R_L \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [\Uparrow N_a]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}^\star \longrightarrow [\Downarrow N_a]}\, R_R$$

$$\frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}\, [\Delta, P_a :\Uparrow \Gamma]\, \mathcal{NC}_2 \longrightarrow \mathcal{G}}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC}\, [\Delta :\Uparrow P_a, \Gamma]\, \mathcal{NC}_2 \longrightarrow \mathcal{G}}\, \Uparrow_L \quad \frac{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [P_a]}{\mathcal{K} : \mathcal{R}^u : \mathcal{NC} \longrightarrow [\Uparrow P_a]}\, \Uparrow_R$$

Fig. 2. SNILLF: Focused Proof System for SNILL. Here P is a positive formula; N_a is a negative or atomic formula; P_a is a positive or atomic formula; e is a fresh eigenvariable, not appearing in $\mathcal{K}, \mathcal{R}^u, \mathcal{NC}, F$; $\mathsf{e} \in \mathcal{E}$; $\mathsf{ne} \notin \mathcal{E}$; $\mathsf{u} \in \mathcal{W} \cap C \cap \mathcal{E}$; $\mathsf{nc} \notin C$; $\mathsf{r} \in C$ and $\mathsf{r} \notin \mathcal{W}$.

and treated as unbounded formulas in that premise. This is specified by the side conditions of that rule.

For soundness of SNILLF with respect to SNILL, we rely on the transformations described in Section 4, namely, that is sound to consider relevant formulas as unbounded in some premises. Given this result, soundness just amounts to erasing the focusing annotations and replacing contexts by formulas. For completeness of focusing, we use the modular technique proposed in [14] based on the following permutation lemmas. Lemma 2 justifies the eager application of negative rules (negative phase). Lemma 3 justifies the preservation of focusing in the positive phase.

Lemma 2. *All positive rules permute over all negative rules.*

Lemma 3. *All positive rules permute over all positive rules.*

Theorem 2. *Let $\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$ be a subexponential signature with $\mathcal{C}, \mathcal{W} \subseteq \mathcal{E}$. Let $\mathcal{K}_\emptyset$ and $\mathcal{R}^u_\emptyset$ be the empty contexts, that is, $\mathcal{K}[\mathsf{s}] = \mathcal{R}^u[\mathsf{s}] = \emptyset$ for all s. For any subexponential signature, the sequent $\Gamma \longrightarrow G$ is provable in SNILL_Σ if and only if the sequent $\mathcal{K}_\emptyset : \mathcal{R}^u_\emptyset : [\cdot \Uparrow \Gamma] \longrightarrow [\Uparrow G]$ is provable in SNILLF_Σ.*

6 Applications

We illustrate the power of SNILLF by revisiting the examples described in Section 3.

6.1 Distributed Systems

Assume a subexponential signature $\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$ where $\mathcal{I} = \{\mathsf{u}, \mathsf{N}, \mathsf{m}_1, \ldots, \mathsf{m}_n\}$, $\preceq$ is the reflexive relation, that is $i \preceq j$, then $i = j$, $\mathcal{E} = \{u, \mathsf{N}\}$ and $\mathcal{C} = \mathcal{W} = \{u\}$. Intuitively, we use the subexponential m_i to specify machine m_i's buffer, N to specify the messages sent on the network and u to specify the behavior of the system. Notice that as there are no relevant formulas $\mathcal{R}^u$ is always empty and therefore elided.

A buffer at machine m_i with elements Γ_i is specified as the list of formulas where start and end mark the start and end of the list $[\mathsf{start}, \Gamma_i, \mathsf{end}]_{\mathsf{m}i}$. Thus a system with n machines is specified as the collection of formulas:

$$\mathcal{NC} = [\mathsf{start}, \Gamma_1, \mathsf{end}]_{\mathsf{m1}}\ [\mathsf{start}, \Gamma_2, \mathsf{end}]_{\mathsf{m2}}\ \cdots [\mathsf{start}, \Gamma_n, \mathsf{end}]_{\mathsf{m}n}$$

For a better presentation, instead of using the context $\mathcal{K}$, we show the formulas in the sequent explicitly where $\mathcal{K}[\mathsf{u}] = \mathcal{U}$ and $\mathcal{K}[\mathsf{N}] = \mathcal{N}$:

$$\mathcal{U} : \mathcal{N} : \mathcal{NC} \longrightarrow \mathcal{G}$$

Notice that since buffers are lists of formulas, we use non-commutative subexponentials to specify them. However, messages on the network are not necessarily delivered in a particular order. Moreover, messages should be consumed exactly once. Therefore, we use the commutative subexponential N to mark these messages.

We now describe how to specify the transmission of messages between machines. For our example, assume two collections of messages $\mathsf{syn}_{\mathsf{mj}}, \mathsf{ack}_{\mathsf{mj}}$ specifying, respectively, a synchronization message from mj and an acknowledgement message to mj. Whenever a machine mi processes the message $\mathsf{syn}_{\mathsf{mj}}$, it sends the message $\mathsf{ack}_{\mathsf{mj}}$ to mj.

The following two clauses specifies this behavior:

$$\mathsf{Deq}(\mathsf{i},\mathsf{j}) = !^{\mathsf{mi}}\mathsf{syn}_{\mathsf{mj}} \cdot !^{\mathsf{mi}}\mathsf{end} \setminus !^{\mathsf{mi}}\mathsf{end} \cdot !^{\mathsf{N}}\,\mathsf{ack}_{\mathsf{mj}}$$
$$\mathsf{Enq}(\mathsf{i},\mathsf{j}) = !^{\mathsf{mj}}\mathsf{start} \cdot !^{\mathsf{mj}}\mathsf{ack}_{\mathsf{mj}} \,/\, !^{\mathsf{N}}\mathsf{ack}_{\mathsf{mj}} \cdot !^{\mathsf{mj}}\mathsf{start}$$

$\mathsf{Deq}(\mathsf{i},\mathsf{j})$ specifies the processing of $\mathsf{syn}_{\mathsf{mj}}$ sending $\mathsf{ack}_{\mathsf{mj}}$ to the network and $\mathsf{Enq}(\mathsf{i},\mathsf{j})$ the receival of $\mathsf{ack}_{\mathsf{mj}}$.

The correctness of this encoding can be easily visualized using focusing. Consider two machines $1, 2$. The focused derivation introducing $\mathsf{Deq} = \mathsf{Deq}(1,2)$ is necessarily of the following form where $\mathcal{M}_2 = [\mathsf{start}, \Gamma_2, \mathsf{end}]_{\mathsf{m2}}$ and $\Theta = \mathsf{Deq}(1,2), \mathsf{Enq}(1,2)$:

$$\dfrac{\dfrac{}{\Theta : \cdot : [\mathsf{syn}_{\mathsf{m2}}, \mathsf{end}]_{\mathsf{m1}} \longrightarrow [\Downarrow !^{\mathsf{m1}}\mathsf{syn}_{\mathsf{m2}} \cdot !^{\mathsf{m1}}\mathsf{end}]} \quad \dfrac{\dfrac{\dfrac{}{\Theta : \mathcal{N}, \mathsf{ack}_{\mathsf{m2}} : [\mathsf{start}, \Gamma_1, \mathsf{end}]_{\mathsf{m1}}\ \mathcal{M}_2 \longrightarrow [G]}}{\Theta : \mathcal{N} : [\mathsf{start}, \Gamma_1]_{\mathsf{m1}}, [\Uparrow !^{\mathsf{m1}}\mathsf{end} \cdot !^{\mathsf{N}}\,\mathsf{ack}_{\mathsf{m2}}]\ \mathcal{M}_2 \longrightarrow [G]}}{\Theta : \mathcal{N} : [\mathsf{start}, \Gamma_1]_{\mathsf{m1}}, [\Downarrow !^{\mathsf{m1}}\mathsf{end} \cdot !^{\mathsf{N}}\,\mathsf{ack}_{\mathsf{m2}}]\ \mathcal{M}_2 \longrightarrow [G]}}{\dfrac{\Theta : \mathcal{N} : [\mathsf{start}, \Gamma_1, \mathsf{syn}_{\mathsf{m2}}, \mathsf{end}]_{\mathsf{m1}}, [\Downarrow \mathsf{Deq}]\ \mathcal{M}_2 \longrightarrow [G]}{\Theta : \mathcal{N} : [\mathsf{start}, \Gamma_1, \mathsf{syn}_{\mathsf{m2}}, \mathsf{end}]_{\mathsf{m1}}, \mathcal{M}_2 \longrightarrow [G]}}$$

Notice that the messages in the network $\mathcal{N}$ are necessarily moved to the right premise, that is, no message is lost. Otherwise, the introduction of $!^{\mathsf{m1}}$ in the left premise would fail since N does not allow weakening and $\mathsf{m1} \not\leq \mathsf{N}$. Moreover, notice that Deq can only be focused on at the location shown above (to the left of $\mathcal{M}_2$). Otherwise, the formula $!^{\mathsf{m1}}\mathsf{end}$ would not be provable: if it is focused not adjacent to a end atom then it would not be provable, and if it is focused to the right of $\mathcal{M}_2$, then one would not be able to introduce $!^{\mathsf{m1}}$. Finally, the message $\mathsf{syn}_{\mathsf{m2}}$ should necessarily appear at the end $\mathsf{m1}$'s' buffer

A similar exercise can be carried out when focusing on $\mathsf{Enq} = \mathsf{Enq}(1,2)$. In this case, the message $\mathsf{ack}_{\mathsf{m2}}$ should be necessarily in $\mathcal{N}$ and moreover, an element is added to the beginning of the buffer of $\mathsf{m2}$. The corresponding derivation is elided.

6.2 Type-Logical Grammar

We return to the sentence *"the paper that John signed without reading"* described in Section 3. The focused proof system SNILLF considerably reduces the proof search space for proving of the typing of this sentence. Assume just a single relevant subexponential r. The corresponding focused proof is as follows where $\Gamma = CN, (CN \setminus CN) \,/\, (S \,/\, !^{\mathsf{r}}N), \Gamma_1$ and $\Gamma_1 = N, N \setminus S \,/\, N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N$. Moreover, we write explicitly the elements of $\mathcal{K}$ and $\mathcal{R}^u$ as in the previous section.

$$\dfrac{\dfrac{\dfrac{}{\cdot : \cdot : [\Downarrow N] \ \rightarrow [N]}\ I \qquad \dfrac{\dfrac{}{\cdot : \cdot : [\Gamma_1]_\star \ \rightarrow [\Downarrow S \,/\, !^{\mathsf{r}}N]} \quad \dfrac{\dfrac{}{\cdot : \cdot : [CN]_\star \ [\Downarrow (CN \setminus CN) \,/\, (S \,/\, !^{\mathsf{s}}N)]\ [\Gamma_1]_\star \ \rightarrow [CN]}\ D_L}{\cdot : \cdot : [\Gamma]_\star \ \rightarrow [CN]}\ /_L}{\cdot : \cdot : [\Downarrow N \setminus CN]\ [\Gamma]_\star \ \rightarrow [N]}\ /_L}{\cdot : \cdot : [\Uparrow N \setminus CN, \Gamma] \rightarrow [N]}\ 7\times \Uparrow_L, D_L$$

Continuing the left premise, we obtain the following derivation, we release focus and apply $/_R$. At this point, the relevant formula $!^{\mathsf{r}}N$ is moved to the commutative context:

$$\dfrac{\dfrac{}{N : \cdot : [GC \,/\, N] \to [GC]} \quad \dfrac{\dfrac{}{\cdot : \underline{N} : [N \setminus S \,/\, N]_\star \to [\Downarrow (N \setminus S)]} \quad \dfrac{}{\cdot : \underline{N} : [N]_\star [\Downarrow N \setminus S] \to [S]}\, 2\times/_L}{\cdot : \underline{N} : [N, N \setminus S \,/\, N]_\star\, [\Downarrow (N \setminus S) \setminus (N \setminus S)] \to [S]}}{\dfrac{N : \cdot : [N, N \setminus S \,/\, N]_\star\, [\Downarrow (N \setminus S) \setminus (N \setminus S) \,/\, GC]\, [GC \,/\, N]_\star \to [S]}{\dfrac{N : \cdot : [\Gamma_1]_\star \to [S]}{\cdot : \cdot : [\Gamma_1]_\star\, [\Uparrow\, !^{\mathsf{r}} N] \to [S]}\, !^{\mathsf{r}}_L}\, D_L}$$

When compared to the derivation in Section 3, focusing reduces proof search in two different ways. First, the proof follows a "back-chaining" strategy [8]. This means that one decides on a formula that can immediately prove the goal. For example, decide on the formula $N \setminus GC$. Search fails immediately if one decides on other formulas. The second way is on deciding when to contract the formula $!^{\mathsf{r}} N$. Indeed, in the derivation above, when the formula N is moved to the left-most branch, it is treated as unbounded in the remaining two branches. This means that one can freely use it as in the middle branch or not as in the right branch.

7 Related Work

Logical Frameworks When compared to existing logical frameworks, SNILLF has an increased expressiveness. When compared to Intuitionistic Linear Logical (ILL) Frameworks [8, 32], SNILLF also allows ordered and relevant formulas. It also seem possible to encode Ordered Logical Frameworks [30, 29] in SNILLF. In particular, one should only consider three subexponentials, one unbounded, one linear (or affine) and another non-commutative. The resulting system behaves similarly to Ordered Logical Frameworks. Moreover, ILL frameworks with subexponentials do not consider relevant formulas. It seems possible to apply the ideas here for reducing "don't know nondeterminism" in the same way as done here.

Finally, as SNILLF is intuitionistic, it cannot be directly compared to classical logical frameworks such as Forum [13] and Classical Linear Logic with Subexponentials [21]. We leave the proposal of a classical version of SNILLF to future work.

Type-Logical Grammar A structural modality closely related to the relevant subexponential discussed above is used in the *CatLog* theorem prover and type-logical grammar parser, which is an ongoing project of Glyn Morrill and his group in Barcelona [17, 18]. The difference of the calculus used in CatLog in comparison to our system is the use of *bracket modalities* that introduce controlled non-associativity and also interact with the relevant subexponential in a non-trivial fashion (see [19] for more details). Bracket modalities are used to block unwanted derivations like *"the girl whom John loves Mary and Pete loves"* or *"the paper that John signed the article without reading."* (Both examples are incorrect from the point of view of English grammar, but accepted by the grammar discussed above.) As shown by Kanovich *et al.* [9], derivability problem in the Lambek calculus with bracket and subexponential modalities is undecidable. There exists, however, a natural decidable fragment, which is actually used in CatLog. This fragment belongs to the NP class, and CatLog utilises several techniques and heuristics in order to speed-up the parsing procedure. In particular, it uses count-invariants for pruning proof search [11] (which generalise multiplicative count-invariants by van Benthem [4]) and focusing for reducing spurious ambiguity. For the multiplicative-additive

fragment focusing for the system used in CatLog is discussed in detail in [20]; completeness of focusing for the full set of connectives used in CatLog, including subexponential, is left by Morrill as a topic for further research [18].

There also exist other type-logical grammar frameworks based on different variants of the Lambek calculus. A notable one is the *Grail* system developed by Moot [16] on the basis of Moortgat's *multi-modal* extension of the non-associative Lambek calculus [15]. Like the subexponential extension of the Lambek calculus discussed in this paper, Moortgat's system uses an indexed family of structural connectives.

8 Conclusions

This paper introduced the logical framework SNILLF which allows for both commutative and non-commutative subexponentials. We demonstrate the power of SNILLF by specifying the structural semantics of distributed systems with buffers and specifying type-logical grammars. For the latter, SNILLF uses commutative relevant formulas, that is, formulas $!^{\mathsf{s}}F$ that can contract, but not weaken. We investigate the proof theory of such formulas in order to reduce "don't know non-determinism" involved demonstrating that under some conditions, these formulas can be treated as unbouded. We believe that this paper lays the foundations for the development of concrete systems for, *e.g.*, type-logical grammars.

We are currently investigating a number of future work directions. We intend to investigate through prototype implementations the impact of SNILLF for categorial parsers. Such an implementation will help us investigate possible further uses of subexponentials for capturing other grammatical constructions. From the proof theory, we are investigating how to reduce the "don't know non-determism" of non-commutative relevant formulas. We are also investigating classical versions for SNILLF following our previous work [10].

References

1. K. Ajdukiewicz. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27, 1935.
2. Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
3. Y. Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58, 1953.
4. Johan van Benthem. *Language in action: categories, lambdas and dynamic logic*. North Holland, Amsterdam, 1991.
5. Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In *Kurt Gödel Colloquium*, pages 159–171, 1993.
6. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
7. Robert Harper, Furio Honsell, and Gordon D. Plotkin. A framework for defining logics. In LICS. 1987.
8. Joshua S. Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic: Extended abstract. In LICS. 1991.

9. M. Kanovich, S. Kuznetsov, and A. Scedrov. Undecidability of the Lambek calculus with subexponential and bracket modalities. In *Fundamentals of Computation Theory*, volume 10472 of *LNCS*, pages 326–340. Springer, 2017.
10. Max Kanovich, Stepan Kuznetsov, Vivek Nigam, and Andre Scedrov. Subexponentials in non-commutative linear logic. *CoRR*, arXiv:1709.03607, 2018. Accepted to Mathematical Structures in Computer Science.
11. S. Kuznetsov, G. Morrill, and O. Valentín. Count-invariance including exponentials. In *Proc. MoL '17*, volume W17-3413 of *ACL Anthology*, pages 128–139, 2017.
12. J. Lambek. The mathematics of sentence structure. *Amer. Math. Monthly*, 65:154–170, 1958.
13. Dale Miller. Forum: A multiple-conclusion specification logic. *Theor. Comput. Sci.*, 165(1):201–232, 1996.
14. Dale Miller and Alexis Saurin. From proofs to focused proofs: A modular proof of focalization in linear logic. In CSL, pages 405–419, 2007.
15. M. Moortgat. Multimodal linguistic inference. *J. Log. Lang. Inform*, 5(3–4):349–385, 1996.
16. R. Moot. The Grail theorem prover: type theory for syntax and semantics. In *Modern Perspectives in Type-Theoretical Semantics*, volume 98 of *Studies in Linguistics and Philosophy*, pages 247–277. Springer, 2017.
17. G. Morrill. CatLog: a categorial parser/theorem-prover. System demostration. In *LACL*, 2012.
18. G. Morrill. Parsing logical grammar: CatLog3. In *Proc. LACompLing2017*, pages 107–131, Stockholm University, 2017.
19. G. Morrill and O. Valentín. Computation coverage of TLG: Nonlinearity. In *Proc. NLCS '15*, volume 32 of *EPiC Series*, pages 51–63, 2015.
20. G. Morrill and O. Valentín. Multiplicative-additive focusing for parsing as deduction. In *First International Workshop on Focusing*, volume 197 of *EPTCS*, pages 29–54, 2015.
21. Vivek Nigam. *Exploiting non-canonicity in the Sequent Calculus*. PhD thesis, Ecole Polytechnique, September 2009.
22. Vivek Nigam. A framework for linear authorization logics. *Theor. Comput. Sci.*, 536:21–41, 2014.
23. Vivek Nigam and Dale Miller. Algorithmic specifications in linear logic with subexponentials. In *PPDP*, pages 129–140, 2009.
24. Vivek Nigam and Dale Miller. A framework for proof systems. *J. Autom. Reasoning*, 45(2):157–188, 2010.
25. Vivek Nigam, Carlos Olarte, and Elaine Pimentel. A general proof system for modalities in concurrent constraint programming. In *CONCUR*, volume 8052 of *LNCS*, pages 410–424. Springer, 2013.
26. Vivek Nigam, Elaine Pimentel, and Giselle Reis. An extended framework for specifying and reasoning about proof systems. *J. Log. Comput.*, 26(2):539–576, 2016.
27. Carlos Olarte, Elaine Pimentel, and Vivek Nigam. Subexponential concurrent constraint programming. *Theor. Comput. Sci.*, 606:98–120, 2015.
28. M. Pentus. Lambek grammars are context-free. In *LICS*, pages 429–433. 1993.
29. Frank Pfenning and Robert J. Simmons. Substructural operational semantics as ordered logic programming. In LICS, pages 101–110, 2009.
30. Jeff Polakow. Linear logic programming with an ordered context. In *PPDP*, pages 68–79, 2000.
31. S. M. Shieber. Evidence against the context-freeness of natural languages. *Linguistics and Philosophy*, 8:333–343, 1985.
32. Kevin Watkins, Iliano Cervesato, Frank Pfenning, and David Walker. A concurrent logical framework: The propositional fragment. In *TYPES*, pages 355–377, 2003.