

Specifying proof systems in linear logic with subexponentials

Vivek Nigam¹, Elaine Pimentel², and Giselle Reis^{3*}

¹ Department of Mathematics,
University of Pennsylvania, Philadelphia, USA

² Departamento de Matemática,
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

³ Departamento de Ciência da Computação,
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Abstract. In the past years, linear logic has been successfully used as a general logical framework for encoding proof systems. Due to linear logic’s finer control on structural rules, it is possible to match the structural restrictions specified in the encoded logic with the use of linear logic connectives. However, some systems that impose more complicated structural restrictions on its sequents cannot be easily captured in linear logic, since it only distinguishes two types of formulas: classical and linear. In this work we intend to use focused linear logic with fixed points and subexponentials as a logical framework for specifying a variety of proof systems. In particular, we are interested on specifying multi-conclusion intuitionistic logics, such as *mLJ*.

1 Introduction

In the past years [Mil96,Pim01,MP02,MP04,PM05,NM08], linear logic has been successfully used as a general logical framework for specifying many proof systems for different logics. It turns out that, in the sequent calculus, a number of dualities appear directly. For example, the occurrences of a formula on the left or right of a sequent arrow are, in some sense, “dual” occurrences of that formula. The cut and initial inference rules are “dual” inference rules. The introduction rules for a logical connective on the left and right of a sequent arrow generally have dual behaviors, at least if one wants to be able to eliminate both non-atomic cuts and non-atomic initial rules. Using linear logic as a logical framework makes it possible to use its involutive negation to directly capture such dualities of the encoded sequent system and reasoning about them.

On the other hand, when designing a proof system, the structural restrictions imposed to its sequents, usually through structural rules, play a role as important as the logical rules themselves. Already in the first sequent calculus systems designed by Gentzen [Gen69], the system, *LJ*, for intuitionistic logic differed from

* This work wouldn’t be possible without Dale Miller’s advices and the support of CNPq and FAPEMIG.

the system for classical logic, LK , by restricting in former system the right-hand-side of sequents to contain at most one formula. Since then, several other proof systems have been proposed, which differ more on these structural restrictions than on their logical rules.

Differently from classical logic, in linear logic the structural rules of weakening and contraction are not allowed to be used on any formula, but only those marked with the so-called exponentials ($?, !$). In this way, it is possible to distinguish between two different kinds of formulas: linear formulas to which no structural rule is applicable (that is, they cannot be erased nor copied) and unbounded formulas to which it is allowed to apply structural rules. This distinction is normally reflected on syntax by using sequents of the form $\vdash \Theta : \Gamma$ containing two contexts [And92]: Θ contains only unbounded formulas and behaves as a set of formulas, while Γ contains only linear formulas and behaves as a multiset of formulas.

One can, therefore, capitalize on this different treatment of formulas to specify proof systems, such as LK and LJ , whose sequents have at most two different contexts, of which one context can be treated as a multiset and another context be treated as a set of formulas.

It does not seem possible, however, to specify proof systems that impose structural restrictions to its sequents which require *more* than one context to be treated as either a set or a multiset of formulas. For instance, the sequents in Maehara's mLJ system [Mae54] requires that both the left and the right-hand-side of sequents have to be treated as two *distinct* sets of formulas. Focused proof systems are other examples of proof systems that impose structural restrictions that require more than one context. Besides contexts to store the formulas appearing on the left and right-hand-side of sequents, one needs an extra context to keep track of the formula being focused on.

It turns out that the linear logic exponentials are not canonical [DJS93]. In fact, it is possible to construct linear logic proof systems with as many exponential-like operators ($?^a, !^a$), called subexponentials [NM09], as needed. These operators may or may not allow for contraction and/or weakening and are organized in a pre-order that specifies the entailment relation among formulas marked with these operators. Subexponentials, therefore, allows the design of proof systems whose sequents have as many contexts as needed and any of them can be treated as multisets or sets. As this paper shows, the use of subexponentials greatly enhance the proof system's expressiveness allowing the capture of more proof systems.

We propose the use of focused linear logic with subexponentials proof system, called $SELLF$ [NM09], to encode such proof systems that impose more complicated structural restrictions to its sequents. More specifically, we show how to encode the systems GIm , mLJ and LJQ^* in $SELLF$.

Up to this point, we have described how linear logic with subexponentials is suitable for specifying a broad of logical systems. One could ask, on the other hand, if a given linear logic formula is the specification of any sequent calculus system. It turns out that it is possible to classify linear logic formulas in such a

way to determine precisely if it corresponds to any possible object level inference rule.

This paper is organized as follows: Section 2 introduces the system *SELLF*, focused linear logic with subexponentials; Section 3 shows the examples of specifications using *SELLF*; Section 4 shows some more connections between object and meta-level logics and, finally, in Section 5 we conclude by giving some directions for future work.

2 Linear logic with subexponentials

Although we assume that the reader is familiar with linear logic, we review some of its basic proof theory. *Literals* are either atomic formulas or their negations. The connectives \otimes and \wp and their units 1 and \perp are *multiplicative*; the connectives \oplus and $\&$ and their units 0 and \top are *additive* connectives; \forall and \exists are (first-order) quantifiers; and $!$ and $?$ are the exponentials. We shall assume that all formulas are in *negation normal form*, meaning that all negations have atomic scope.

Due to the exponentials, one can distinguish in linear logic two kinds of formulas: the linear ones whose main connective is not a $?$ and the unbounded ones whose main connective is a $?$. The linear formulas can be seen as resources that can only be used once, while the unbounded formulas as unlimited resources which can be used as many times necessary. This distinction is usually reflected in syntax by using two different contexts in linear logic sequents ($\vdash \Theta : \Gamma$) one (Θ) containing only unbounded formulas and another (Γ) only linear formulas [And92]. Such distinction allows to incorporate structural rules, *i.e.*, weakening and contraction, into the introduction rules of connectives, as done in similar presentations for classical logic, *e.g.*, the G3c system in [TS96]. In such presentation, the context (Θ) containing unbounded formulas is treated as a set of formulas, while the other context (Γ) containing only linear formulas is treated as a multiset of formulas.

It turns out that the exponentials are not canonical [DJS93]. In fact, if, for any reason, we decide to define a blue and red conjunctions (\wedge^b and \wedge^r respectively) with the standard classical rules:

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge^b B} \wedge^b R \qquad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge^b B \vdash \Delta} \wedge^b L$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge^r B} \wedge^r R \qquad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge^r B \vdash \Delta} \wedge^r L$$

then it is easy to show that, for any formulas A and B , $A \wedge^b B \equiv A \wedge^r B$. This means that the classical conjunction is *canonical*. However, the same behavior does not hold with the linear logic modals. In fact, suppose we have red $!$, $?$ and blue $!^b$, $?^b$ sets of exponentials with the standard linear logic rules:

$$\frac{\vdash ?^r \Gamma, F}{\vdash ?^r \Gamma, !^r F} !^r \quad \frac{\vdash \Gamma, F}{\vdash \Gamma, ?^r F} D?^r \quad \frac{\vdash ?^b \Gamma, F}{\vdash ?^b \Gamma, !^b F} !^b \quad \frac{\vdash \Gamma, F}{\vdash \Gamma, ?^b F} D?^b$$

We cannot show $!^r F \equiv !^b F$ nor $?^r F \equiv ?^b F$. This opens the possibility of defining classes of exponentials, the *subexponentials* [NM09]. In this way, it is possible to build proof systems containing as many exponential-like operators, $(!^l, ?^l)$ as one needs: they may or may not allow contraction and weakening, and are organized in a pre-order (\preceq) specifying the entailment relation between these operators. Formally, a proof system for linear logic with subexponentials, called $SELL_\Sigma$, is specified by using a subexponential signature of the form $\langle I, \preceq, \mathcal{W}, \mathcal{C} \rangle$, where I is the set of labels for subexponentials, \preceq is a preorder relation among the elements of I , and both \mathcal{W} and \mathcal{C} are subsets of I , specifying which subexponentials allow for, respectively, weakening and contraction. We will require the preorder \preceq to be upwardly closed with respect to the sets \mathcal{W} and \mathcal{C} , that is, if $x \prec y$ and $x \in \mathcal{W}$ ($x \in \mathcal{C}$), then $y \in \mathcal{W}$ ($y \in \mathcal{C}$). $SELL_\Sigma$ contains the same introduction rules as in linear logic for all connectives, except the exponentials. These are specified, on the other hand, by the subexponential signature, Σ , as follows:¹

$$\frac{\vdash C, \Delta}{\vdash ?^x C, \Delta} D, \text{ if } x \in I \quad \frac{\vdash ?^y C, ?^y C, \Delta}{\vdash ?^y C, \Delta} C, \text{ if } y \in \mathcal{C} \quad \frac{\vdash \Delta}{\vdash ?^z C, \Delta} W, \text{ if } z \in \mathcal{W}$$

That is, dereliction can be applied to any subexponential, and contraction (respectively, weakening) only to subexponentials that appear in the set \mathcal{C} (respectively, \mathcal{W}). In this paper, we will assume that $\mathcal{C} = \mathcal{W}$. The promotion rule is given by the following inference rule:

$$\frac{\vdash ?^{x_1} C_1, \dots, ?^{x_n} C_n, C}{\vdash ?^{x_1} C_1, \dots, ?^{x_n} C_n, !^a C} !^a$$

where $a \preceq x_i$ for all $i = 1, \dots, n$. The promotion rule will play an important role here, namely, to specify the structural restrictions of encoded proof systems. In particular, one can use a subexponential bang, $!^c$, to check whether there are only some type of formulas in the context, namely, those that are marked with subexponentials, $?^x$, such that $c \preceq x$. If there is any formula $?^y F$ in the context such that $c \not\preceq y$, then $!^c$ cannot be introduced.

As we show in this paper, the use of subexponentials greatly increases the expressiveness of linear logic by no longer limiting one to use only two contexts, but as many as one needs, namely, one context for each subexponential. Moreover, since one can specify to which subexponentials one is allowed to copy or erase formulas, these contexts can either be treated as multisets or sets of formulas. This will allow us to encode in $SELL$ a vast number of proof systems that do not seem possible to be encoded in *plain* linear logic.

2.1 Focusing

First proposed by Andreoli [And92] for plain linear logic, focused proof systems provide the normal form proofs for cut-free proofs. In this section, we review the focused proof system for $SELL$, called $SELLF$, proposed in [NM09].

¹ Whenever it is clear from the context, we will elide the subexponential signature Σ .

In order to introduce *SELLF*, we first recall some more terminology. We classify as *positive* the formulas whose main connective is either \otimes, \oplus, \exists , the subexponential bang, the unit 1 and positive literals. All other formulas are classified as *negative*. Figure 1 contains the focused proof system *SELLF* that is a rather straightforward generalization of Andreoli’s original system. There are two kinds of arrows in this proof system. Sequents with the \Downarrow belong to the *positive* phase and introduce the logical connective of the “focused” formula (the one to the right of the arrow): building proofs of such sequents may require non-invertible proof steps to be taken. Sequents with the \Uparrow belong to the *negative* phase and decompose the formulas on their right in such a way that only invertible inference rules are applied. The structural rules $D_1, D_l, R \Uparrow$, and $R \Downarrow$ make the transition between a negative and a positive phase.

Similarly as in the usual presentation of linear logic, there is a pair of contexts to the left of \Uparrow and \Downarrow of sequents, written here as $\mathcal{K} : \Gamma$. The second context, Γ , collects the formulas whose main connective is not a question-mark, behaving as the bounded context in linear logic. But differently from linear logic, where the first context is a multiset of formulas whose main connective is a question-mark, we generalize \mathcal{K} to be an *indexed context*, which is a mapping from each index in the set I (for some given and fixed subexponential signature) to a finite multiset of formulas, in order to accommodate for more than one subexponential in *SELLF*. In Andreoli’s focused system for linear logic, the index set contains just ∞ and $\mathcal{K}[\infty]$ contains the set of unbounded formulas. Figure 2 contains different operations used in such indexed contexts. For example, the operation $(\mathcal{K}_1 \otimes \mathcal{K}_2)$, used in the tensor rule, specifies the resulting indexed context obtained by merging two contexts \mathcal{K}_1 and \mathcal{K}_2 .

Focusing allows the composition of a collection of inference rules of the same polarity into a “macro-rule”. Consider, for example, a formula $N_1 \oplus N_2 \oplus N_3$, where all N_1, N_2 , and N_3 are negative formulas. Once focused on, the only way to introduce such a formula is by using one a “macro-rule” of the form:

$$\frac{\frac{\vdash \mathcal{K} : \Gamma \Uparrow N_i}{\vdash \mathcal{K} : \Gamma \Downarrow N_1 \oplus N_2 \oplus N_3}}{\vdash \mathcal{K} : \Gamma \Downarrow N_1 \oplus N_2 \oplus N_3}$$

where $i \in \{1, 2, 3\}$. There is no other alternative. For another example, the back-chaining and forward-chaining rules in logic programming can also be explained in such a way [LM09]. Here, we will encode proof systems in *SELLF* in such a way that the “macro-rules” available using our specifications match exactly the inference rules of the encoded system. This is the strongest level of adequacy possible as described in [NM08].

In this paper, we will make great use of the promotion rule, $!^l$, in order to specify the structural restrictions of a proof system. In particular, this rule determines two different operations when seeing this introduction rule from the conclusion to premise. The first one arises by its side condition: a bang can be introduced only if the linear contexts that are not greater to l are all empty. This operation is similar to the promotion rule in plain linear logic: a bang can be introduced only if the linear context is empty. Nigam and Miller exploited

$$\begin{array}{c}
\text{NEGATIVE PHASE} \\
\frac{\frac{\vdash \mathcal{K} : \Gamma \uparrow L, A \quad \vdash \mathcal{K} : \Gamma \uparrow L, B}{\vdash \mathcal{K} : \Gamma \uparrow L, A \& B} [\&] \quad \frac{\vdash \mathcal{K} : \Gamma \uparrow L, A, B}{\vdash \mathcal{K} : \Gamma \uparrow L, A \wp B} [\wp]}{\vdash \mathcal{K} : \Gamma \uparrow L, \top} [\top] \quad \frac{\vdash \mathcal{K} : \Gamma \uparrow L}{\vdash \mathcal{K} : \Gamma \uparrow L, \perp} [\perp]} \\
\frac{\vdash \mathcal{K} : \Gamma \uparrow L, A\{c/x\}}{\vdash \mathcal{K} : \Gamma \uparrow L, \forall x.A} [\forall] \quad \frac{\vdash \mathcal{K} +_l A : \Gamma \uparrow L}{\vdash \mathcal{K} : \Gamma \uparrow L, ?^l A} [?^l] \\
\text{POSITIVE PHASE} \\
\frac{\vdash \mathcal{K} : \Gamma \Downarrow A_i}{\vdash \mathcal{K} : \Gamma \Downarrow A_1 \oplus A_2} [\oplus_i] \quad \frac{\vdash \mathcal{K}_1 : \Gamma \Downarrow A \quad \vdash \mathcal{K}_2 : \Delta \Downarrow B}{\vdash \mathcal{K}_1 \otimes \mathcal{K}_2 : \Gamma, \Delta \Downarrow A \otimes B} [\otimes, \text{given } (\mathcal{K}_1 = \mathcal{K}_2)|_{\mathcal{C} \cap \mathcal{W}}] \\
\frac{}{\vdash \mathcal{K} : \cdot \Downarrow 1} [1, \text{given } \mathcal{K}[\mathcal{Z} \setminus \mathcal{W}] = \emptyset] \quad \frac{\vdash \mathcal{K} : \Gamma \Downarrow A\{t/x\}}{\vdash \mathcal{K} : \Gamma \Downarrow \exists x.A} [\exists] \\
\frac{\vdash \mathcal{K} \leq_l : \cdot \uparrow A}{\vdash \mathcal{K} : \cdot \Downarrow !^l A} [!^l, \text{given } \mathcal{K}[\{x \mid l \not\leq x \wedge x \notin \mathcal{W}\}] = \emptyset] \\
\text{INITIAL, REACTION AND DECISION RULES} \\
\frac{}{\vdash \mathcal{K} : \Gamma \Downarrow A_p} [I, \text{given } A_p^\perp \in (\Gamma \cup \mathcal{K}[\mathcal{Z}]) \text{ and } (\Gamma \cup \mathcal{K}[\mathcal{Z} \setminus \mathcal{W}]) \subseteq \{A_p^\perp\}] \\
\frac{\vdash \mathcal{K} +_l P : \Gamma \Downarrow P}{\vdash \mathcal{K} +_l P : \Gamma \uparrow \cdot} [D_l, \text{given } l \in \mathcal{C} \cap \mathcal{W}] \quad \frac{\vdash \mathcal{K} : \Gamma \Downarrow P}{\vdash \mathcal{K} +_l P : \Gamma \uparrow \cdot} [D_l, \text{given } l \notin \mathcal{C} \cap \mathcal{W}] \\
\frac{\vdash \mathcal{K} : \Gamma \Downarrow P}{\vdash \mathcal{K} : \Gamma, P \uparrow \cdot} [D_1] \quad \frac{\vdash \mathcal{K} : \Gamma \uparrow N}{\vdash \mathcal{K} : \Gamma \Downarrow N} [R \Downarrow] \quad \frac{\vdash \mathcal{K} : \Gamma, S \uparrow L}{\vdash \mathcal{K} : \Gamma \uparrow L, S} [R \uparrow]
\end{array}$$

Fig. 1. Focused linear logic system with subexponentials. It is assumed that: $\mathcal{C} \subseteq \mathcal{W}$; L is a list of formulas, Γ is a multi-set of formulas and positive literals, A_p is a positive polarity literal, P is a non-negative literal, S is a positive literal or formula and N is a negative formula.

$$\begin{array}{l}
\bullet (\mathcal{K}_1 \otimes \mathcal{K}_2)[i] = \begin{cases} \mathcal{K}_1[i] \cup \mathcal{K}_2[i] & \text{if } i \notin \mathcal{C} \\ \mathcal{K}_1[i] & \text{if } i \in \mathcal{C} \cap \mathcal{W} \end{cases} \quad \bullet \mathcal{K}[S] = \bigcup \{\mathcal{K}[i] \mid i \in S\} \\
\bullet (\mathcal{K} +_l A)[i] = \begin{cases} \mathcal{K}[i] \cup \{A\} & \text{if } i = l \\ \mathcal{K}[i] & \text{otherwise} \end{cases} \quad \bullet \mathcal{K} \leq_i [l] = \begin{cases} \mathcal{K}[l] & \text{if } i \leq l \\ \emptyset & \text{if } i \not\leq l \end{cases} \\
\bullet (\mathcal{K}_1 \star \mathcal{K}_2) \upharpoonright_S \text{ is true if and only if } (\mathcal{K}_1[j] \star \mathcal{K}_2[j])
\end{array}$$

Fig. 2. Specification of operations on contexts. Here, $i \in I$, $j \in \mathcal{S}$, $\mathcal{S} \subseteq I$, and the binary connective $\star \in \{=, \subseteq, \supseteq\}$.

this in [NM08] to encode *LJ* in Andreoli’s focused system for linear logic. The second operation is specified by using the operation $\mathcal{K}_{\leq l}$: in the premise of the promotion rule all unbounded contexts that are not greater than l are erased. Notice that such operation is not available in plain linear logic.

Finally, to improve readability, we will often show explicitly the formulas appearing in the image of the indexed context, \mathcal{K} , of a sequent. For example, if the set of subexponential indexes is $\{x_1, \dots, x_n\}$, then the following negative sequent

$$\vdash \Theta_1 :_{x_1} \Theta_2 :_{x_2} \cdots \Theta_n :_{x_n} \Gamma \uparrow L$$

denotes the *SELLF* sequent $\vdash \mathcal{K} : \Gamma \uparrow L$, such that $\mathcal{K}[x_i] = \Theta_i$ for all $1 \leq i \leq n$. We will also assume the existence of a maximal subexponential called ∞ for which contraction and weakening are allowed and which is greater than all other subexponentials. This subexponential is used to store linear logic theories specifying proof systems.

3 Encoding proof systems

Similar as in Church’s simple type theory [Chu40], we assume that linear logic propositions have type o and that the linear logic quantifiers have type $(\textit{term} \rightarrow \textit{form}) \rightarrow o$, where *term* and *form* are respectively the types for an object-logic term and for object-logic formulas. Moreover, following [PM05,Pim01,NM08], we encode a sequent in *SELLF* by using two meta-level atoms $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ of type $\textit{form} \rightarrow o$. They denote respectively an object logic formula appearing on the left and on the right-hand-side of a sequent. For example, the sequent $B_1, \dots, B_n \vdash C_1, \dots, C_m$ could be encoded by the *SELLF* sequent $\vdash \Theta :_{\infty} \lfloor B_1 \rfloor, \dots, \lfloor B_n \rfloor :_l \lceil C_1 \rceil, \dots, \lceil C_m \rceil :_r \cdot \uparrow \cdot$, where Θ encodes the proof system’s introduction rules. Notice that in *SELLF* we can configure the contexts for the subexponentials l (for left) and r (for right) to either behave like sets or multisets by changing the subexponential signature accordingly. For instance, if we use the subexponentials signature $\langle \{l, r, \infty\}, \preceq, \{l, \infty\}, \{l, \infty\} \rangle$, with some preorder \preceq , the contexts l and ∞ are treated as sets, while the context r is treated as a multiset. Such situation would be useful for any proof systems where the right-hand-side of its sequent behaves as a multiset of formulas and the left-hand-side behaves as a set of formulas. Finally, for convenience, if Γ is a (multi)set of object-logic formulas, then $\lfloor \Gamma \rfloor$ (respectively, $\lceil \Gamma \rceil$) denotes the (multi)set of meta-level atoms $\{\lfloor F \rfloor \mid F \in \Gamma\}$ (respectively, $\{\lceil F \rceil \mid F \in \Gamma\}$).

3.1 G1m

The first logic that we encode in *SELLF* is the proof system for minimal logic called G1m [TS96], where the rules for contraction and weakening are explicit rules in the system and all introduction rules are *multiplicative*.² The rules of

² The exchange rule is still implicitly incorporated in a sequent by assuming its context to be multiset of formulas and not lists. The system G1m used here is called *context-independent* rules in [TS96, Remark 3.1.5].

this system are depicted in Figure 3. There both the left and right-hand-side of sequents are treated as two distinct multisets of formulas. This is particularly relevant for the rules \supset_R and Cut since the formula C on the right-hand-side of their conclusion sequents needs to be moved necessarily to the right premise.

Consequently, in *SELLF*, we will require two subexponentials, l and r , that do not allow neither contraction nor weakening to store, respectively, object-logic formulas appearing on the left and on the right of the sequent. Moreover, we use the theory \mathcal{L}_{G1m} depicted in Figure 4 to specify in *SELLF* the G1m's introduction rules. This theory is, on the other hand, stored in the subexponential ∞ that is greater than both l and r and since an introduction rule can be used in an object-logic proof as many times needed, ∞ is allowed to both contract and weaken. This can be summarized by the subexponential signature $\langle\{\infty, l, r\}, \{l \prec \infty, r \prec \infty\}, \{\infty\}, \{\infty\}\rangle$.

Intuitively, each clause in \mathcal{L}_{G1m} specifies an introduction rule of G1m. To obtain such strong correspondence between focused proofs obtained from \mathcal{L}_{G1m} , we need to capture precisely the structural restrictions in the system. In particular, the use of the $!^l$ in the clauses (\supset_L) , specifying the rule \supset_L , and Id_2 , specifying Cut rules, is necessary. It forces that the side-formula, C , appearing in the right-hand-side of their conclusion is moved to the correct premise. This is illustrated by the following derivation:

$$\frac{\frac{\frac{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_1] \dot{i} [A] \dot{i} \cdot \uparrow}{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_1] \dot{i} \cdot \dot{i} \cdot \downarrow !^{l?r} [A]} \quad !^l, ?^r \quad \frac{\frac{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_2, A] \dot{i} [C] \dot{i} \cdot \uparrow}{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_2] \dot{i} [C] \dot{i} \cdot \downarrow ?^l [A]} \quad R\downarrow, ?^l}{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_1, F_2] \dot{i} [C] \dot{i} \cdot \downarrow !^{l?r} [A] \otimes ?^l [A]} \quad \otimes}{\vdash \mathcal{L}_{G1m} \dot{\infty} [F_1, F_2] \dot{i} [C] \dot{i} \cdot \uparrow} \quad D_{\infty}, \exists$$

When introducing the tensor, the formula $[C]$ cannot go to the left branch since, in that case, the $!^l$ could not be introduced: in order to introduce this connective the r context must be empty. Therefore, the only way to introduce the formula Id_2 is with a derivation such as the one above. Notice that such specification would not be possible in plain linear logic because, there, only one context can be treated as a multiset, while in G1m, one requires two such contexts.

Formally, the following was proved in [Nig09].

Proposition 1. *Let $\Gamma \cup \{C\}$ be a set of object logic formulas, and let the subexponentials, l and r , be specified by the signature $\langle\{\infty, l, r\}, \{l \prec \infty, r \prec \infty\}, \{\infty\}, \{\infty\}\rangle$. Then the sequent $\vdash \mathcal{L}_{G1m} \dot{\infty} [\Gamma] \dot{i} [C] \dot{i} \cdot \uparrow$ is provable in *SELLF* if and only if the sequent $\Gamma \vdash C$ is provable in *G1m*.*

3.2 *mLJ*

We now encode in *SELLF* the multi-conclusion intuitionistic sequent calculus *mLJ*, whose rules appear in Figure 5. In order to specify *mLJ*, two different contexts are also required, but differently from G1m, they need to be treated as sets of formulas. This restriction is a result of the right introduction rules

$$\begin{array}{c}
\frac{\Gamma_1 \rightarrow A \quad \Gamma_2, B \rightarrow C}{\Gamma_1, \Gamma_2, A \supset B \rightarrow C} \supset L \qquad \frac{\Gamma, A \rightarrow B}{\Gamma \rightarrow A \supset B} \supset R \qquad \frac{\Gamma, A_i \rightarrow C}{\Gamma, A_1 \wedge A_2 \rightarrow C} \wedge_i L \\
\frac{\Gamma_1 \rightarrow A \quad \Gamma_2 \rightarrow B}{\Gamma_1, \Gamma_2 \rightarrow A \wedge B} \wedge R \qquad \frac{\Gamma, A\{t/x\} \rightarrow C}{\Gamma, \forall x A \rightarrow C} \forall L \qquad \frac{\Gamma \rightarrow A\{c/x\}}{\Gamma \rightarrow \forall x A} \forall R \\
\frac{\Gamma, A\{c/x\} \rightarrow C}{\Gamma, \exists x A \rightarrow C} \exists L \qquad \frac{\Gamma \rightarrow A\{t/x\}}{\Gamma \rightarrow \exists x A} \exists R \qquad \frac{\Gamma, A \rightarrow C \quad \Gamma, B \rightarrow C}{\Gamma, A \vee B \rightarrow C} \vee L \\
\frac{\Gamma \rightarrow A_i}{\Gamma \rightarrow A_1 \vee A_2} \vee_i R \qquad \frac{\Gamma \rightarrow C}{\Gamma, A \rightarrow C} W_L \qquad \frac{\Gamma, A, A \rightarrow C}{\Gamma, A \rightarrow C} C_L \\
\frac{}{A \rightarrow A} I \qquad \frac{\Gamma_1 \rightarrow A \quad \Gamma_2, A \rightarrow C}{\Gamma_1, \Gamma_2 \rightarrow C} \text{Cut}
\end{array}$$

Fig. 3. The sequent calculus system GIm for minimal logic. Here, Γ_1, Γ_2 are multisets of formulas and C is a formula; in the rules $\exists L$ and $\forall R$, the eigenvariable c does not appear free in Γ nor C ; and $i \in \{1, 2\}$.

$$\begin{array}{ll}
(\supset_L) \quad [A \supset B]^\perp \otimes (!^r[A] \otimes ?^l[B]) & (\supset_R) \quad [A \supset B]^\perp \otimes (?^l[A] \wp ?^r[B]) \\
(\wedge_L) \quad [A \wedge B]^\perp \otimes (?^l[A] \oplus ?^l[B]) & (\wedge_R) \quad [A \wedge B]^\perp \otimes (?^r[A] \otimes ?^r[B]) \\
(\vee_L) \quad [A \vee B]^\perp \otimes (?^l[A] \& ?^l[B]) & (\vee_R) \quad [A \vee B]^\perp \otimes (?^r[A] \oplus ?^r[B]) \\
(\forall_L) \quad [\forall B]^\perp \otimes ?^l[Bx] & (\forall_R) \quad [\forall B]^\perp \otimes \forall x ?^r[Bx] \\
(\exists_L) \quad [\exists B]^\perp \otimes \forall x ?^l[Bx] & (\exists_R) \quad [\exists B]^\perp \otimes ?^r[Bx] \\
(\text{Id}_1) \quad [B]^\perp \otimes [B]^\perp & (\text{Id}_2) \quad !^r[B] \otimes ?^l[B] \\
(\text{C}_L) \quad [B]^\perp \otimes (?^l[B] \wp ?^l[B]) & (\text{W}_L) \quad [B]^\perp \otimes \perp
\end{array}$$

Fig. 4. The theory, \mathcal{L}_{GIm} , for GIm .

for implications and universal quantifiers. In these rules, all the side-formulas appearing in the right-hand-side of the conclusion sequent have to be erased in the premise, while the side-formulas appearing in the left-hand-side remain untouched.

The theory \mathcal{L}_{mlj} in Figure 6 specifies mLJ in *SELLF*. As before, we make use of two subexponentials l and r to store, respectively, meta-level atoms $[\cdot]$ and $[\cdot]$, but now we allow both contraction and weakening to these subexponential indexes. That is, we use the subexponential signature $\langle \{\infty, l, r\}; \{l \preceq \infty, r \preceq \infty\}; \{\infty, l, r\}; \{\infty, l, r\} \rangle$. The use of $!$ in the clauses (\supset_r) and (\forall_R) enforces that the formulas in the context r are erased when these formulas are introduced, as illustrates the following derivation introducing the formula (\forall_R) :

$$\frac{\frac{\frac{\frac{\frac{\frac{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} [Ac] \dot{i} \cdot \uparrow}{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} \cdot \dot{i} \cdot \uparrow \forall x ?^r [Ax]} \vee, ?^r}{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} [\Delta, \forall x A] \dot{i} \cdot \downarrow \downarrow !^r \forall x ?^r [Ax]} !^r}{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} [\Delta, \forall x A] \dot{i} \cdot \downarrow \downarrow [\forall x A]^\perp} I_r}{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} [\Delta, \forall x A] \dot{i} \cdot \downarrow \downarrow [\forall A]^\perp \otimes !^r \forall x ?^r [Ax]} \otimes}{\vdash \mathcal{L}_{mlj} \dot{\infty} [\Gamma] \dot{i} [\Delta, \forall x A] \dot{i} \cdot \uparrow} D_{\infty, \exists}$$

$$\begin{array}{c}
\frac{\Gamma, A \supset B \rightarrow A, \Delta \quad \Gamma, A \supset B, B \rightarrow \Delta}{\Gamma, A \supset B \rightarrow \Delta} \supset_l \quad \frac{\Gamma, A \rightarrow B}{\Gamma \rightarrow A \supset B, \Delta} \supset_r \\
\frac{\Gamma, A \wedge B, A, B \rightarrow \Delta}{\Gamma, A \wedge B \rightarrow \Delta} \wedge_l \quad \frac{\Gamma \rightarrow A \wedge B, A, \Delta \quad \Gamma \rightarrow A \wedge B, B, \Delta}{\Gamma \rightarrow A \wedge B, \Delta} \wedge_r \\
\frac{\Gamma, A \vee B, A, \rightarrow \Delta \quad \Gamma, A \vee B, B \rightarrow \Delta}{\Gamma, A \vee B \rightarrow \Delta} \vee_l \quad \frac{\Gamma \rightarrow A \vee B, A, B, \Delta}{\Gamma \rightarrow A \vee B, \Delta} \vee_r \\
\frac{\Gamma, \forall x A, A\{t/x\} \rightarrow \Delta}{\Gamma, \forall x A \rightarrow \Delta} \forall_l \quad \frac{\Gamma \rightarrow A\{c/x\}}{\Gamma \rightarrow \Delta, \forall x A} \forall_r \\
\frac{\Gamma, \exists x A, A\{c/x\} \rightarrow \Delta}{\Gamma, \exists x A \rightarrow \Delta} \exists_l \quad \frac{\Gamma \rightarrow \Delta, \exists x A, A\{t/x\}}{\Gamma \rightarrow \Delta, \exists x A} \exists_r \\
\frac{}{\Gamma, A \rightarrow A, \Delta} \text{I} \quad \frac{\Gamma \rightarrow B, \Delta \quad \Gamma, B \rightarrow \Delta}{\Gamma \rightarrow \Delta} \text{Cut} \quad \frac{}{\Gamma, \perp \rightarrow \Delta} \perp_l
\end{array}$$

Fig. 5. The multi-conclusion intuitionistic sequent calculus, mLJ , with additive rules.

$$\begin{array}{ll}
(\supset_l) & [A \supset B]^\perp \otimes (?^r[A] \& ?^l[A]) \\
(\wedge_l) & [A \wedge B]^\perp \otimes (?^l[A] \wp ?^r[B]) \\
(\vee_l) & [A \vee B]^\perp \otimes (?^l[A] \& ?^r[B]) \\
(\forall_L) & [\forall B]^\perp \otimes ?^l[Bx] \\
(\exists_L) & [\exists B]^\perp \otimes \forall x ?^l[Bx] \\
(\perp_L) & [\perp]^\perp \\
(Id_1) & [B]^\perp \otimes [B]^\perp \\
(\supset_r) & [A \supset B]^\perp \otimes !(?^l[A] \wp ?^r[B]) \\
(\wedge_r) & [A \wedge B]^\perp \otimes (?^r[A] \& ?^r[B]) \\
(\vee_r) & [A \vee B]^\perp \otimes (?^r[A] \wp ?^r[B]) \\
(\forall_R) & [\forall B]^\perp \otimes !\forall x ?^r[Bx] \\
(\exists_R) & [\exists B]^\perp \otimes ?^r[Bx] \\
(Id_2) & ?^l[B] \otimes ?^r[B]
\end{array}$$

Fig. 6. The theory, \mathcal{L}_{mlj} , for the multi-conclusion intuitionistic logic system mLJ .

In particular, since $l \not\leq r$, all formulas should be erased in the context r in the premise of the promotion rule. The derivation above also illustrates how one can specify fresh values with the use of the universal quantifier. As in mLJ , the eigenvariable c cannot appear in Δ nor Γ .

The following result is proved by induction on the height of focused proofs [Nig09].

Proposition 2. *Let $\Gamma \cup \Delta$ be a set of object-logic formulas, and let the subexponentials l and r be specified by the signature $\langle \{\infty, l, r\}; \{l \leq \infty, r \leq \infty\}; \{\infty, l, r\}; \{\infty, l, r\} \rangle$. Then the sequent $\vdash \mathcal{L}_{mlj} \dot{\circ} [\Gamma] \dot{\circ} [\Delta] \dot{\circ} \uparrow$ is provable in $SELLF$ if and only if the sequent $\Gamma \vdash \Delta$ is provable in mLJ .*

3.3 LJQ^*

The systems in the previous sections always required two contexts, that were treated either as multisets or as sets. There are systems, however, that require more than two contexts to be specified, such as the focused multi-conclusion system for intuitionistic logic LJQ^* depicted in Figure 7. This system is a variant of the system proposed by Herbelin [Her95, page 78] and was used by Dyckhoff

$$\begin{array}{c}
\frac{\Gamma, A \supset B \rightarrow A; \cdot \quad \Gamma, A \supset B, B \vdash \Delta}{\Gamma, A \supset B \vdash \Delta} \supset_l \quad \frac{\Gamma, A \vdash B}{\Gamma \rightarrow A \supset B; \Delta} \supset_r \\
\frac{\Gamma, A \vee B, A \vdash \Delta \quad \Gamma, A \vee B, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \vee_l \quad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \rightarrow A \vee B; \Delta} \vee_r \\
\frac{\Gamma, A \wedge B, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge_l \quad \frac{\Gamma \rightarrow A; \Delta \quad \Gamma \rightarrow B; \Delta}{\Gamma \rightarrow A \wedge B; \Delta} \wedge_r \\
\frac{}{\Gamma, A \rightarrow A; \Delta} I \quad \frac{\Gamma \rightarrow C; \Delta}{\Gamma \vdash C, \Delta} D \quad \frac{}{\Gamma, \perp \vdash \Delta} \perp_l
\end{array}$$

Fig. 7. The focused multi-conclusion system for intuitionistic logic - LJQ^* .

$$\begin{array}{ll}
(Id_1) \quad [A]^\perp \otimes [A]^\perp & (\perp_L) \quad [\perp]^\perp \\
(\supset_L) \quad [A \supset B]^\perp \otimes (!^l ?^f [A] \otimes !^r ?^l [B]) & (\supset_R) \quad [A \supset B]^\perp \otimes !^l (?^l [A] \wp ?^r [B]) \\
(\vee_L) \quad [A \vee B]^\perp \otimes (!^r ?^l [A] \otimes !^r ?^l [B]) & (\vee_R) \quad [A \vee B]^\perp \otimes !^r (?^r [A] \wp ?^r [B]) \\
(\wedge_L) \quad [A \wedge B]^\perp \otimes !^r (?^l [A] \wp ?^l [B]) & (\wedge_R) \quad [A \wedge B]^\perp \otimes (!^r ?^f [A] \otimes !^r ?^f [B])
\end{array}$$

Fig. 8. The theory \mathcal{L}_{ljq} used to encode the system LJQ^* .

& Lengrand in [DL06]. LJQ^* has two types of sequents: unfocused sequents of the form $\Gamma \vdash \Delta$ and focused sequents of the form $\Gamma \rightarrow A; \Delta$ where the formula A , in the *stoup*, is focused on. Proofs are restricted as follows: the logical right introduction rules introduce only focused sequents, while the left introduction rules introduce only unfocused sequents.

We use the theory \mathcal{L}_{ljq} depicted in Figure 8 to specify the system LJQ^* in *SELLF*. Besides the subexponential ∞ , we make use of three subexponentials: the first two, l and r , are as before, used to encode, respectively, the left and the right-hand-side of object-logic sequents, while the third subexponential, f , is new and used to encode the stoup of object-logic focused sequents. Formally, they are specified by the signature $\langle \{f, l, r, \infty\} \{r \prec l \prec \infty\}; \{l, r, \infty\}; \{l, r, \infty\} \rangle$. Notice that, differently from the previous encoding, the subexponentials r , and l are related in the pre-order and moreover contraction and weakening are not available only to f . As before, the restrictions to sequents imposed by the focusing discipline are encoded implicitly by the use of subexponentials. One is able to correctly specify the restrictions that positive rules can only be applied to the focused formula and that negative rules can only be applied when the stoup is empty.

To illustrate the fact that negative rules are only applicable when the stoup is empty, consider the following derivation introducing the clause (\wedge_L) :

$$\frac{\frac{\frac{}{\vdash \mathcal{K} \Downarrow [A \wedge B]^\perp} I_l \quad \frac{\vdash \mathcal{L}_{ljq} \dot{i} [\Gamma', A, B] \dot{i} [\Delta] \dot{i} \cdot \cdot \uparrow}{\vdash \mathcal{L}_{ljq} \dot{i} [\Gamma'] \dot{i} [\Delta] \dot{i} \cdot \cdot \Downarrow !^r (?^l [A] \wp ?^l [B])} !^r, \wp, 2 \times ?^l}{\vdash \mathcal{L}_{ljq} \dot{i} [\Gamma'] \dot{i} [\Delta] \dot{i} \cdot \cdot \Downarrow [A \wedge B]^\perp \otimes !^r (?^l [A] \wp ?^l [B])} \otimes}{\vdash \mathcal{L}_{ljq} \dot{i} [\Gamma'] \dot{i} [\Delta] \dot{i} \cdot \cdot \uparrow} D_\infty, 2 \times \exists}$$

where \mathcal{K} is an abbreviation for the context $\mathcal{L}_{ljq} \overset{\circ}{\circ} [I'] \dot{i} [\Delta] \dot{i} \cdot \dot{i} \cdot$, and Γ' is the set $\Gamma \cup \{A \wedge B\}$. Since $r \not\prec f$, the context f must be empty in order to introduce the $!$ in the right branch. On the other hand, since $r \prec l$, the l context is left untouched in the premise of this derivation, thus specifying precisely the \wedge_L introduction rule.

As expected, each clause in \mathcal{L}_{ljq} specifies precisely an inference rule in LJQ^* [Nig09]:

Proposition 3. *Let $\Gamma \cup \Delta \cup \{C\}$ be a set of object logic formulas, and let the subexponentials l, r and f be specified by the signature $\langle \{f, l, r, \infty\} \{r \prec l \prec \infty\}; \{l, r, \infty\}; \{l, r, \infty\} \rangle$. Then the sequent $\vdash \mathcal{L}_{ljq} \dot{i} [\Gamma] \dot{i} [\Delta] \dot{i} \cdot \dot{i} \cdot \uparrow$ is provable in $SELLF$ if and only if the sequent $\Gamma \vdash \Delta$ is provable in LJQ^* .*

3.4 Implementation of linear logic with subexponentials

The system depicted in Figure 1 and all examples of encodings listed in this paper were implemented using λ -Prolog. In this scenario, focused linear logic with subexponentials is the object-level logic and a fragment of the intuitionistic logic is the meta-level logic. Most of the inference rules of Figure 1 have a straightforward specification in λ -Prolog, differing only in the context management. The difference between additive and multiplicative operators, and the presence of subexponentials are all related to context control. The source code is available at <http://kontesti.me/~giselle/SELLF/>.

4 Relating Meta-level Formulas and Object-level Rules

So far, we have shown how to use linear logic in order to encode logical systems, and how subexponentials can enhance dramatically the amount of systems that can be encoded. On the other hand, it is natural to ask whenever a linear logic formula corresponds to a specification of an inference rule.

In [CGT08], Agata et al introduced a systematic procedure to relate large classes of linear logic formulas into equivalent structural inference rules in sequent and hypersequent calculi. In that work, the classes \mathcal{N}_i and \mathcal{P}_i are defined so that $\mathcal{N}_i \subset \mathcal{N}_{i+1}$, $\mathcal{N}_i \subset \mathcal{P}_{i+1}$, $\mathcal{P}_i \subset \mathcal{P}_{i+1}$, $\mathcal{P}_i \subset \mathcal{N}_{i+1}$ and \mathcal{P}_i is built using *positive* connectives, while \mathcal{N}_i is built using *negative* ones. Although the definition is over intuitionistic linear logic without exponentials, it is straightforward to extend these classes to the whole linear logic.

Definition 1. *A monopole formula is a linear logic formula that contains only occurrences of the asynchronous connectives, namely $\wp, \&, \perp, \top, \forall$ together with the modal $?$, which can only have atomic scope. A bipolar is a formula in which no synchronous connective is in the scope of an asynchronous connective and where $?$ has atomic scope.*

Hence monopoles are in \mathcal{N}_1 while bipolars are in \mathcal{P}_2 , both with the restriction that $?$ must have atomic scope.

In [CGT08] there are two main results concerning this classification: 1) every axiom in \mathcal{N}_2 is equivalent to a finite set of structural rules; 2) every axiom in

\mathcal{P}_3 is equivalent to a finite set of hyperstructural rules. It turns out that, in our approach, we can completely characterize formulas in \mathcal{P}_2 . In fact, since bipolars define *synthetic connectives*, every one of these formulas determines a sequent calculus inference rule, trivially. For example, the linear logic formula $A \otimes (B \& C)$ determines a macro-rule:

$$\frac{\vdash \mathcal{K}_1 : \Gamma_1 \Downarrow A \quad \vdash \mathcal{K}_2 : \Gamma_2 \Uparrow B \quad \vdash \mathcal{K}_2 : \Gamma_2 \Uparrow C}{\vdash \mathcal{K}_1 \otimes \mathcal{K}_2 : \Gamma_1, \Gamma_2 \Downarrow A \otimes (B \& C)}$$

In order to relate this to object level logics, we introduce the concept of *introduction clauses*.

Definition 2. Let \mathcal{Q} be a fixed set of unary meta-level predicates. An introduction clause is a closed formula of the form

$$\forall x_1 \dots \forall x_n [q(\diamond(x_1, \dots, x_n))^\perp \otimes B]$$

where \diamond is an object-level connective of arity n ($n \geq 0$), $q \in \mathcal{Q}$ and B is a bipolar. Furthermore, an atom occurring in B is either of the form $p(x_i)$ or $p(x_i(y))$ where p is a meta-level predicate and $1 \leq i \leq n$. In the first case, x_i has a type of order 0 while in the second case x_i has a type of order 1 and y is a variable quantified (universally or existentially) in B (in particular, y is not in $\{x_1, \dots, x_n\}$).

The next result (proved by a straightforward case analysis on the structure of bipolar formulas) states that introduction clauses naturally yield object-level inference rules.

Proposition 4. Every introduction clause (hence in \mathcal{P}_2) corresponds to a specification of a sequent calculus introduction rule.

There are some interesting questions related to this subject. It is not clear, for example, how to use linear logic in order to model hypersequents (hence we do not know how to deal with the class \mathcal{P}_3). For the class $\mathcal{N}_i, i \geq 3$, we have the following result.

Theorem 1. There exist introduction clauses in $\mathcal{N}_i, i \geq 3$ which does not correspond to any specification of object level inference rules.

Proof. The following proves the above theorem for $n = 3$ and, since $\mathcal{N}_i \subset \mathcal{N}_j$ for $i < j$, the result comes. Consider the following introduction clauses with non-bipolar bodies:³

$$[\diamond(A, B, C)]^\perp \otimes ([A] \& ([B] \otimes [C])) \quad [\diamond(A, B, C)]^\perp \otimes ([A] \oplus ([B] \wp [C]))$$

If they are to correspond to the encoding of sequent inference rules, the *natural* candidates would be

$$\frac{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, A \quad \Gamma_1 \vdash \Delta_1, B \quad \Gamma_2 \vdash \Delta_2, C}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \diamond(A, B, C)} \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, \diamond(A, B, C) \vdash \Delta} \quad \frac{\Gamma, B, C \vdash \Delta}{\Gamma, \diamond(A, B, C) \vdash \Delta}$$

³ Note that these clauses are in \mathcal{N}_3 since they are equivalent to the formulas $[\diamond(A, B, C)] \multimap ([A] \& ([B] \otimes [C]))$ and $[\diamond(A, B, C)] \multimap ([A] \oplus ([B] \wp [C]))$.

$$\begin{array}{ll}
(\Rightarrow L) & \llbracket A \Rightarrow B \rrbracket^\perp \otimes \llbracket A \rrbracket \& \llbracket B \rrbracket. & (\Rightarrow R) & \llbracket A \Rightarrow B \rrbracket^\perp \otimes \llbracket A \rrbracket \wp \llbracket B \rrbracket. \\
(\wedge L) & \llbracket A \wedge B \rrbracket^\perp \otimes \llbracket A \rrbracket \wp \llbracket B \rrbracket. & (\wedge R) & \llbracket A \wedge B \rrbracket^\perp \otimes \llbracket A \rrbracket \& \llbracket B \rrbracket. \\
(\vee R) & \llbracket A \vee B \rrbracket^\perp \otimes \llbracket A \rrbracket \wp \llbracket B \rrbracket. & (\vee L) & \llbracket A \vee B \rrbracket^\perp \otimes \llbracket A \rrbracket \& \llbracket B \rrbracket. \\
(f_c L) & \llbracket f_c \rrbracket^\perp \otimes \top. & & &
\end{array}$$

Fig. 9. Specification of **G3c**.

As it turns out, at the meta level the sequent $\vdash (Id_1)^\perp : \cdot \uparrow \llbracket A \rrbracket \& (\llbracket B \rrbracket \otimes \llbracket C \rrbracket), \llbracket A \rrbracket \oplus (\llbracket B \rrbracket \wp \llbracket C \rrbracket)$ is provable, while at the object level, the two sequent rules listed above cannot be used to prove $\diamond(A, B, C) \vdash \diamond(A, B, C)$. This sequent can be proved only with an immediate application of the initial axiom. Thus, the meta-level encoding of the object-level inference rules is not adequate in this non-bipolar example.

4.1 Invertibility of rules

Another property that has been studied in the sequent calculus setting is the invertibility of rules. We say that a rule is invertible if the provability of the conclusion sequent implies the provability of all the premises.

This property is very much of interest to proof search since invertible rules permute down with the other rules of a proof, reducing the non-determinism. In particular, in systems with only invertible rules, the bottom-up search for a proof can stop as soon as a non provable sequent is reached.

For example, it is well known that all rules in **G3c** (see [TS96]) are invertible. This system is specified in Figure 9. Observe that the meta level connectives in the bodies are asynchronous. In general, the following is a straightforward result.

Theorem 2. *An monopole introduction clause corresponds to an invertible object level rule.*

5 Conclusion and Future Work

We have introduced the logical framework *SELLF*, and showed how to specify different logical systems. Also, we addressed the problem of characterizing linear logic formulas as inference rules, as well as invertible rules.

There are several ways of continuing this work. In fact, in [MP02], a necessary condition was presented in order to guarantee that specified systems had the cut-elimination property. The result was based on the fact that cut elimination is often proved via case analysis, where the duality of inference rules play an important role in the elimination of non-atomic cuts. This is translated to the meta-level by “dual” linear logic formulas. It would be interesting to extend this result to linear logic with subexponentials, since, in most of the systems specified in this paper, the proof of cut-elimination at the meta-level is really hard.

Another direction to pursue would be the relation between *multifocusing* with exponentials and parallel computing. In this way, extend the concept that proof search corresponds to sequential algorithms, as stated in [Nig09].

References

- [AL01] Arnon Avron and Iddo Lev. Canonical propositional gentzen-type systems. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR 2001*, volume 2083 of *LNAI*, pages 529–544. Springer-Verlag, 2001.
- [And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.
- [CGT08] Agata Ciabattoni, Nikolaos Galatos, and Kazushige Terui. From axioms to analytic rules in nonclassical logics. In *23th Symp. on Logic in Computer Science*, pages 229–240. IEEE Computer Society Press, 2008.
- [Chu40] Alonzo Church. A formulation of the simple theory of types. *J. of Symbolic Logic*, 5:56–68, 1940.
- [CP02] Iliano Cervesato and Frank Pfenning. A Linear Logical Framework. *Information & Computation*, 179(1):19–75, November 2002.
- [CT06] Agata Ciabattoni and Kazushige Terui. Towards a semantic characterization of cut-elimination. *Studia Logica*, 82(1):95–119, 2006.
- [DJS93] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 159–171. Springer, 1993.
- [DL06] R. Dyckhoff and S. Lengrand. LJQ: a strongly focused calculus for intuitionistic logic. In A. Beckmann and *et al.*, editors, *Computability in Europe 2006*, volume 3988 of *LNCS*, pages 173–185. Springer, 2006.
- [Dyc92] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *J. of Symbolic Logic*, 57(3):795–807, September 1992.
- [FM88] Amy Felty and Dale Miller. Specifying theorem provers in a higher-order logic programming language. In *Ninth International Conference on Automated Deduction*, pages 61–80, Argonne, IL, May 1988. Springer-Verlag.
- [Gen69] Gerhard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.
- [Gir99] Jean-Yves Girard. On the meaning of logical rules I: syntax vs. semantics. In Berger and Schwichtenberg, editors, *Computational Logic*, pages 215–272. Springer, 1999.
- [Her95] Hugo Herbelin. *Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de lambda-termes et comme calcul de stratégies gagnantes*. PhD thesis, Université Paris 7, 1995.
- [HHP87] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. In *Second Annual Symposium on Logic in Computer Science*, pages 194–204, Ithaca, NY, June 1987.
- [HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, 1993.
- [HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.
- [LM09] Chuck Liang and Dale Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, 410(46):4747–4768, 2009.
- [Mae54] S. Maehara. Eine darstellung der intuitionistischen logik in der klassischen. *Nagoya Mathematical Journal*, pages 45–64, 1954.

- [Mil96] Dale Miller. Forum: A multiple-conclusion specification logic. *Theoretical Computer Science*, 165(1):201–232, September 1996.
- [MN87] Dale Miller and Gopalan Nadathur. A logic programming approach to manipulating formulas and programs. In Seif Haridi, editor, *IEEE Symposium on Logic Programming*, pages 379–388, San Francisco, September 1987.
- [MP02] Dale Miller and Elaine Pimentel. Using linear logic to reason about sequent systems. In Uwe Egly and Christian G. Fermüller, editors, *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, volume 2381 of *LNCS*, pages 2–23. Springer, 2002.
- [MP04] Dale Miller and Elaine Pimentel. Linear logic as a framework for specifying sequent calculus. In Jan van Eijck, Vincent van Oostrom, and Albert Visser, editors, *Logic Colloquium '99: Proceedings of the Annual European Summer Meeting of the Association for Symbolic Logic*, Lecture Notes in Logic, pages 111–135. A K Peters Ltd, 2004.
- [Nig09] Vivek Nigam. *Exploiting non-canonicity in the sequent calculus*. PhD thesis, Ecole Polytechnique, September 2009.
- [NM88] Gopalan Nadathur and Dale Miller. An Overview of λ Prolog. In *Fifth International Logic Programming Conference*, pages 810–827, Seattle, August 1988. MIT Press.
- [NM08] Vivek Nigam and Dale Miller. Focusing in linear meta-logic. In *Proceedings of IJCAR: International Joint Conference on Automated Reasoning*, volume 5195 of *LNAI*, pages 507–522. Springer, 2008.
- [NM09] 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.
- [NPW02] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002. LNCS Tutorial 2283.
- [Pau89] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–397, September 1989.
- [Pfe89] Frank Pfenning. Elf: A language for logic definition and verified metaprogramming. In *Fourth Annual Symposium on Logic in Computer Science*, pages 313–321, Monterey, CA, June 1989.
- [Pfe95] Frank Pfenning. Structural cut elimination. In *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 156–166, San Diego, California, 1995. IEEE Computer Society Press.
- [Pim01] Elaine Gouvêa Pimentel. *Lógica linear e a especificação de sistemas computacionais*. PhD thesis, Universidade Federal de Minas Gerais, Belo Horizonte, M.G., Brasil, December 2001. Written in English.
- [PM05] Elaine Pimentel and Dale Miller. On the specification of sequent systems. In *LPAR 2005: 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, number 3835 in *LNAI*, pages 352–366, 2005.
- [PS99] Frank Pfenning and Carsten Schürmann. System description: Twelf — A meta-logical framework for deductive systems. In H. Ganzinger, editor, *16th Conference on Automated Deduction (CADE)*, number 1632 in *LNAI*, pages 202–206, Trento, 1999. Springer.
- [TS96] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.