From cut-free calculi to automated deduction: the case of bounded contraction [1]

Agata Ciabattoni, Björn Lellmann[2]

Vienna University of Technology. Austria.

Carlos Olarte, Elaine Pimentel[3]

Universidade Federal do Rio Grande do Norte. Brazil

Abstract

The addition of the bounded contraction rules to Full Lambek Calculus with exchange and weakening (FL_{ew}) gives rise to serious complications for proof search. For example, adding to FL_{ew} a naive version of these rules brakes cut-admissibility. While this can be avoided by considering more sophisticated rules, in this work we show that even "good", i.e., focused and cut-free proof systems for FL_{ew} plus bounded contraction do not necessarily lead to good implementations. In order to solve this problem, we propose an extension of the lazy splitting methodology to bounded contractions, showing how to transform our focused, cut-free sequent calculus into a terminating theorem prover. Our system is used to show that the decision problem for FL_{ew} with bounded contraction is in EXPTIME.

Keywords: Proof theory, substructural logics, proof search, bounded contraction.

---

1 Introduction

Cut-free sequent calculi are undoubtedly among the finest tools for proving important theoretical properties such as consistency, decidability or interpolation for a given logic. Unfortunately, when it comes to practical applications, and in particular to automated reasoning tasks, they are not always well-behaved. Seemingly innocuous rules like the contraction rule give rise to a dramatic increase of the search space in backwards proof search procedures. Indeed, a major part of the proof-theoretic effort for proposing good logical proof systems suitable for implementation involves taming exactly this rule. For example, good refinements of sequent systems for classical logic absorb contraction into the logical rules; in intuitionistic logic, contraction can be avoided only after a careful control on the use of the implication left rule [Dyc92]; and all focused systems, e.g. for linear logic, rely on controlling the duplication of classical resources.

---

[1] This work has been supported by the Marie Curie project GetFun (PIRSES-GA-2012-318986 funded by EU-FP7), the FWF START project Y544-N23, EU H2020-MSCA grant 660047, CAPES and CNPq.

[2] Email: `{agata,lellmann}@logic.at`

[3] Email: `{carlos.olarte, elaine.pimentel}@gmail.com`

Here we consider weaker variants of the contraction rule, the so-called *bounded contraction* rules, which for a fixed k, contract $k + 1$ copies of a formula into k copies. These rules are special cases of knotted rules [HOS94] and play an important role, e.g., in finite-valued Łukasiewic logics [Pri96] or varieties of residuated lattices [GJKO07].

Simply adding bounded contraction rules to the sequent calculus for affine intuitionistic additive, multiplicative linear logic aIMALL [Gir87] (or, equivalently, Full Lambek Calculus with exchange and weakening FL_{ew}) brakes cut-admissibility (although not consistency). Equivalent versions of the bounded contraction rules, which preserve cut-elimination when added to FL_{ew}, have been defined, e.g., in [CGT08]. However, although the resulting cut-free systems might suffice for theoretical purposes, the additional rules involve splitting the context into a number of parts. Since there are many ways to do so, this results in an exponential branching in the naive proof search procedure, rendering it unfeasible for all but the most basic practical purposes.

In this paper we propose solve this problem and show how to turn the cut-free sequent calculus for FL_{ew} with bounded contraction into a terminating theorem prover. For this purpose we propose a general form of *lazy splitting* which avoids the splitting of contexts, hence solving completely the problem of the exponential choice inherent in the rules of bounded contraction. This choice is substituted by a much smaller one: from which context, on the leaves, an atomic formula should be taken. This not only turns bounded contractions into harmless structural rules, but also provides the basis for efficient theorem provers for a number of logics. An implementation of the system, available at `http://subsell.logic.at/flew`, is also presented. We further show an EXPTIME upper bound on the complexity of FL_{ew} with bounded contractions. While a PSPACE lower bound for these logics follows from [HT11], their exact complexity is left open.

The paper is structured as follows: The base logical system FL_{ew} with bounded contraction is presented in Section 2, while its focused, cut-free version is described in Section 3. Section 4 shows decidability and complexity results. Section 5 proposes the general lazy splitting procedure. Finally, we conclude and present some future work in Section 6.

2 FL_{ew} with bounded contraction

Providing feasible *automated deduction procedures* for substructural logics, i.e., logics whose sequent calculi lack (or restrict the use of) some of the standard structural rules, is a difficult task. A successful example of such a procedure is the one available under `http://www1.chapman.edu/~jipsen/reslat/`, which implements the algorithm in [OT99] to decide validity of equations in residuated lattices. In this work we provide automated deduction procedures for substructural logics obtained by adding to Full Lambek calculus with exchange and weakening (FL_{ew}, or aIMALL) axioms $\alpha^k \supset \alpha^{k+1}$ (where α^n stands for $\alpha \otimes \cdots \otimes \alpha$, n times, $\alpha^0 = \top$), for $k \geq 1$, or, equivalently, the rules [HOS94,Pri96]

$$\frac{\Delta, A^{k+1} \vdash C}{\Delta, A^k \vdash C}\ (\mathsf{k}+1)-\mathsf{k}$$

The complete list of rules is given in Fig. 1 (note that the Weakening rule is admissible by a standard induction on the depth of the derivation). As shown in [HOS94] such systems enjoy cut elimination if and only if $k = 1$. Consider indeed the sequent

$$((\mathsf{k}+1)-\mathsf{k}) \qquad \frac{}{\Delta, A \vdash A}\,\mathsf{init} \qquad \frac{}{\Delta \vdash \top}\,\top \qquad \frac{}{\Delta, \bot \vdash C}\,\bot$$

$$\frac{\Delta, A, B \vdash C}{\Delta, A \otimes B \vdash C}\,\otimes_L \qquad \frac{\Delta_1 \vdash A \quad \Delta_2 \vdash B}{\Delta_1, \Delta_2 \vdash A \otimes B}\,\otimes_R \qquad \frac{\Delta_1 \vdash A \quad \Delta_2, B \vdash C}{\Delta_1, \Delta_2, A \supset B \vdash C}\,\supset_L \qquad \frac{\Delta, A \vdash B}{\Delta \vdash A \supset B}\,\supset_R$$

$$\frac{\Delta, A_i \vdash C}{\Delta, A_1 \wedge A_2 \vdash C}\,\wedge_{Li} \qquad \frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \wedge B}\,\wedge_R \qquad \frac{\Delta, A \vdash C \quad \Delta, B \vdash C}{\Delta, A \vee B \vdash C}\,\vee_L \qquad \frac{\Delta \vdash A_i}{\Delta \vdash A_1 \wedge A_2}\,\vee_{Ri}$$

Fig. 1. Naive system for FL_{ew} with bounded contraction.

$$C, A \supset C \supset B, (A \supset B)^{k-1}, (A \supset B)^{k+1} \supset D \vdash D \tag{1}$$

A proof with cut in the naive system of Fig.1 is

$$\frac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{init} \quad \dfrac{\dfrac{}{C \vdash C}\,\mathsf{init} \quad \dfrac{}{B \vdash B}\,\mathsf{init}}{C, C \supset B \vdash B}\,\supset_L}{C, A \supset C \supset B, A \vdash B}\,\supset_L}{C, A \supset C \supset B \vdash A \supset B}\,\supset_R \qquad \dfrac{\dfrac{\dfrac{}{D \vdash D}\,\mathsf{init}}{(A \supset B)^{k+1}, (A \supset B)^{k+1} \supset D \vdash D}\,\supset \otimes}{(A \supset B)^{k}, (A \supset B)^{k+1} \supset D \vdash D}\,(\mathsf{k}+1)-\mathsf{k}}{C, A \supset C \supset B, (A \supset B)^{k-1}, (A \supset B)^{k+1} \supset D \vdash D}\,\mathsf{cut}$$

It is straightforward to check that this cut cannot be eliminated, see also [HOS94].

Although this counter-example is interesting, it is based on the fact that some information is hidden inside the implication. This kind of "bureaucracy", where the logical system is not enough for handling all the information given by its formulas, can be easily eliminated using the so called *systems modulo* [DHK03]. Observe that the following equalities hold in in FL_{ew}

$$A \supset B \supset C \quad \equiv \quad B \supset A \supset C \quad \equiv \quad A \otimes B \supset C \tag{2}$$

Hence, one could easily enhance the naive system in Figure 1 by adding the equational theory described above. In this new system, the sequent (1) has a cut-free proof, since $A \supset C \supset B \equiv C \supset A \supset B$:

$$\frac{\dfrac{\dfrac{}{C \vdash C} \quad \dfrac{\dfrac{}{(A \supset B)^{k+1}, (A \supset B)^{k+1} \supset D \vdash D}\,\supset_L, \mathsf{init}\ (k+2\ times)}{(A \supset B)^{k}, (A \supset B)^{k+1} \supset D \vdash D}\,(\mathsf{k}+1)-\mathsf{k}}{C, C \supset A \supset B, (A \supset B)^{k-1}, (A \supset B)^{k+1} \supset D \vdash D}\,\supset}{C, A \supset C \supset B, (A \supset B)^{k-1}, (A \supset B)^{k+1} \supset D \vdash D}\,\equiv$$

But there are some other less trivial counter-examples, such as

$$A \supset \bot, (A \supset B)^{k-1} \vdash (A \supset B)^{k+1} \tag{3}$$

Observe that $A \supset \bot$ implies $A \supset B$. However, using this information in a deduction modulo theory would be equivalent to using cut in our naive system. It is non-trivial to determine which set of reductions modulo are necessary in order to make the system cut-free.

There are a number of ways of proposing less naive sequent systems for handling bounded contractions. For instance, the algorithm in [CGT08], transforms (suitable) axioms into structural rules and leads to the following rules for bounded contraction

$$\frac{\Delta, \Delta_1^{k+1} \vdash C \quad \ldots \quad \Delta, \Delta_1^{i_1}, \ldots, \Delta_k^{i_k} \vdash C \quad \ldots \quad \Delta, \Delta_k^{k+1} \vdash C}{\Delta, \Delta_1, \ldots, \Delta_k \vdash C}\,\mathsf{k}'$$

where $\Delta_1, \dots, \Delta_k$ are non-empty multisets of formulas and $i_1 + \dots + i_k = k+1$. It was shown in *op. cit.* that adding these rules to FL_{ew} preserves cut-elimination. Although cut-free, due to the large number of premises of this rule, the above calculus is not good for proof search. Alternatively we propose the following rule, which greatly reduces the number of branches:

$$\frac{\Delta, \Delta_1^{k+1} \vdash C \qquad \dots \qquad \Delta, \Delta_k^{k+1} \vdash C}{\Delta, \Delta_1, \dots, \Delta_k \vdash C}\ \mathsf{k}$$

Lemma 2.1 *The rules* k *and* k' *are equivalent in* FL_{ew} *in the presence of the cut rule.*

Proof. Clearly k derives k'. For the converse direction it is enough to show that each $\Delta, \Delta_1^{i_1}, \dots \Delta_k^{i_k} \vdash C$ (with $i_1 + \dots + i_k = k+1$) can be derived from the premises of k. This follows by cutting the sequent $\Delta, \bigvee_{i=1}^{k}(\bigotimes \Delta_i^{k+1}) \vdash C$, where $\bigotimes \Delta_i$ is the formula obtained by replacing in Δ_i the commas by tensors, with $\Delta_1^{i_1}, \dots \Delta_k^{i_k} \vdash \bigvee_{i=1}^{k}(\bigotimes \Delta_i^{k+1})$ which is a sequent derivable in FL_{ew} extended with k'. □

Using the rule k, the sequents (1) and (3) are cut-free provable: take $\Delta_1 = \{C, A \supset C \supset B\}$, $\Delta_2 = \dots = \Delta_k = \{A \supset B\}$ and $\Delta = (A \supset B)^{k+1} \supset D$ and $\Delta_1 = \{A \supset \bot\}$, $\Delta_2 = \dots = \Delta_k = \{A \supset B\}$ and $\Delta = \emptyset$, respectively.

Observe that when using the k rule there is an exponential number of choices, since one has to cleverly split the context into $k+1$ parts. Hence the difficulty of finding an arbitrary cut-formula is replaced by a hard context splitting.

It is instructive to consider the relationship between the naive system in Fig. 1 and the system obtained by replacing the $(\mathsf{k}+1)-\mathsf{k}$ rule with the k rule. Assume that we use k instances of cut to produce k copies of the formula A, which will be used for applying the $(\mathsf{k}+1)-\mathsf{k}$ rule:

$$\begin{array}{c}
\dfrac{\Delta_k \vdash A \qquad \dfrac{A^{k+1}, \Delta \vdash C}{A^k, \Delta \vdash C}\ (\mathsf{k}+1)-\mathsf{k}}{A^{k-1}, \Delta, \Delta_k \vdash C}\ \mathsf{cut} \\
\vdots \\
\dfrac{\Delta_1 \vdash A \qquad A, \Delta, \Delta_2, \dots, \Delta_k \vdash C}{\Delta, \Delta_1, \dots, \Delta_k \vdash C}\ \mathsf{cut}
\end{array}$$

Observe that choosing the cut formula (in this case, A) is the same as choosing the multisets $\Delta_1, \dots, \Delta_k$ that imply A. It turns out that we can choose, instead of a random A, the formula $\bigvee_{i=1}^{k}(\bigotimes \Delta_i)$ as the cut formula. Indeed, in the presence of $(\mathsf{k}+1)-\mathsf{k}$, we have the equivalence $\bigvee_{i=1}^{k}(\bigotimes \Delta_i)^k \equiv (\bigvee_{i=1}^{k}(\bigotimes \Delta_i))^k$. Hence the proof above can be re-written as

$$\begin{array}{c}
\dfrac{\dfrac{}{\Delta_k \vdash \bigvee_{i=1}^{k}(\bigotimes \Delta_i)}\ \lor_R, \mathsf{init} \qquad \dfrac{\dfrac{\dfrac{(\Delta_1)^{k+1}, \Delta \vdash C \quad \dots \quad (\Delta_k)^{k+1}, \Delta \vdash C}{\bigvee_{i=1}^{k}((\Delta_i))^{k+1}, \Delta \vdash C}\ \lor_L}{(\bigvee_{i=1}^{k}(\Delta_i))^{k+1}, \Delta \vdash C}\ \equiv}{(\bigvee_{i=1}^{k}(\Delta_i))^{k}, \Delta \vdash C}\ (\mathsf{k}+1)-\mathsf{k}}{(\bigvee_{i=1}^{k}(\Delta_i))^{k-1}, \Delta, \Delta_k \vdash C}\ \mathsf{cut} \\
\vdots \\
\dfrac{\dfrac{}{\Delta_1 \vdash \bigvee_{i=1}^{k}(\bigotimes \Delta_i)}\ \lor_R, \mathsf{init} \qquad \bigvee_{i=1}^{k}(\Delta_i), \Delta, \Delta_2, \dots, \Delta_k \vdash C}{\Delta, \Delta_1, \dots, \Delta_k \vdash C}\ \mathsf{cut}
\end{array}$$

showing that the rule k is derivable in the naive system. Conversely, observe that $(\mathsf{k}+1)-\mathsf{k}$

Negative Rules

$$\frac{}{\Gamma;\Delta \vdash \top}\ \top_R \qquad \frac{}{\Gamma;\Delta,\bot \vdash C}\ \bot_L \qquad \frac{\Gamma;\Delta,F,G \vdash C}{\Gamma;\Delta,F \otimes G \vdash C}\ \otimes_L$$

$$\frac{\Gamma;\Delta,F \vdash G}{\Gamma;\Delta \vdash F \supset G}\ \supset_R \qquad \frac{\Gamma;\Delta \vdash F \quad \Gamma;\Delta \vdash G}{\Gamma;\Delta \vdash F \wedge G}\ \wedge_R \qquad \frac{\Gamma;\Delta,F \vdash C \quad \Gamma;\Delta,G \vdash C}{\Gamma;\Delta,F \vee G \vdash C}\ \vee_L$$

$$\frac{\Gamma,F,G;\Delta \vdash C}{\Gamma,F \,\&\, G;\Delta \vdash C}\ \&_L\ \& \in \{\wedge,\otimes\} \qquad \frac{\Gamma,F;\Delta \vdash C \quad \Gamma,G;\Delta \vdash C}{\Gamma,F \vee G;\Delta \vdash C}\ \vee_{LC}$$

$$\frac{\Gamma,G;\Delta \vdash C}{\Gamma,F \supset G,F;\Delta \vdash C}\ \supset_{LCC} \qquad \frac{\Gamma,G;\Delta \vdash C}{\Gamma,F \supset G,G;\Delta \vdash C}\ \supset_{LCG}$$

Positive Rules

$$\frac{\Gamma;\Delta_1 \vdash F \quad \Gamma;\Delta_2 \vdash G}{\Gamma;\Delta_1,\Delta_2 \vdash F \otimes G}\ \otimes_R \qquad \frac{\Gamma;\Delta_1 \vdash F \quad \Gamma;\Delta_2,G \vdash C}{\Gamma;\Delta_1,\Delta_2,F \supset G \vdash C}\ \supset_L$$

$$\frac{\Gamma,F \supset G;\Delta_1 \vdash F \quad \Gamma,F \supset G;\Delta_2,G \vdash C}{\Gamma,F \supset G;\Delta_1,\Delta_2 \vdash C}\ \supset_{LC} \qquad \frac{\Gamma;\Delta,F_i \vdash C}{\Gamma;\Delta,F_1 \wedge F_2 \vdash C}\ \wedge_{Li} \qquad \frac{\Gamma;\Delta \vdash F_i}{\Gamma;\Delta \vdash F_1 \vee F_2}\ \vee_{Ri}$$

Structural Rules

$$\frac{}{\Gamma;\Delta \vdash p}\ \mathsf{init}\ p \in \{\Delta,\Gamma\} \qquad \frac{\Gamma,\Gamma_1;\Delta \vdash C\ \ldots\ \Gamma,\Gamma_k;\Delta \vdash C}{\Gamma;\Delta,\Gamma_1,\ldots,\Gamma_k \vdash C}\ \mathsf{k}$$

Fig. 2. System FL_{ew}^k. In the init axiom, p is atomic.

is an instance of k with $\Delta_1 = \ldots = \Delta_k = A$. Hence, in this case, finding a cut-formula or splitting the context have the same level of difficulty.

Remark 2.2 This destroys the myth that cut-free systems are necessarily well behaved. Sometimes, choosing a cut formula can be as hard as applying a rule, even when the cut can be shown to be analytic.

3 A focused, cut-free system for bounded contractions

We start by observing that, once a formula is contracted using the k rule, it can be contracted an infinite number of times. Hence our sequents have two classes of formulas: the *unbounded* ones, that can be weakened and contracted, and the *bounded* ones, that can be only weakened. Of course, FL_{ew} rules have different behaviors when applied to these different classes of formulas.

The grammar for formulas in FL_{ew} is shown below, and the rules of the proposed cut-free system FL_{ew}^k are presented in Figure 2.

$$F ::= \top \mid \bot \mid A \mid F_1 \otimes F_2 \mid F_1 \vee F_2 \mid F_1 \supset F_2 \mid F_1 \wedge F_2.$$

The connectives $\otimes, \vee$ are called *positive*, while $\supset, \wedge$ are called *negative*. The *sequents* in FL_{ew}^k have the shape

$$\Gamma;\Delta \vdash C$$

where Γ is the *unbounded context* and Δ is the *bounded* (or linear) one.

We now move in the direction of proposing a notion of *focusing* [And92] for systems with bounded contractions. We start by analyzing the possible orderings of applications of rules. The goal is to organize proofs in order to reduce the non determinism, as in [MS07].

Definition 3.1 Let S be a sequent with two formulas A and B such that the rule α (resp. β) can be applied on A (resp. B). We say that a rule β *permutes down* α, notation $\beta \downarrow \alpha$, if for any proof π of S starting with α followed immediately by β (reading proofs bottom up), there exists a proof π' of S where the two rules have been exchanged (considering also the empty case). We write $\beta \updownarrow \alpha$ when $\beta \downarrow \alpha$ and $\alpha \downarrow \beta$.

Given two sets of inference rules $\mathcal{A}$ and $\mathcal{B}$, we say that $\mathcal{A}$ has *weak permutability* if, for any two rules α, α' of $\mathcal{A}$, $\alpha \updownarrow \alpha'$. We say that $\mathcal{A}$ has *full permutability over* $\mathcal{B}$ when $\mathcal{A}$ has weak permutability and, for any pair of rules $(\alpha,\beta) \in \mathcal{A} \times \mathcal{B}$, $\alpha \downarrow \beta$ holds.

The following proposition identifies the classes of rules that have weak or full permutability in FL^k_{ew}. The proof is straightforward by doing small step permutations or using well known equivalences in intuitionistic logic. The list with all the counter-examples for non-permutability cases can be found in Appendix A.

Proposition 3.2 *Let* $\mathcal{N}$ *be the set of inference rules* $\{\top_R, \bot_L, \otimes_L, \supset_R, \wedge_R, \vee_L, \&_L, \vee_{LC}, \supset_{LCC}, \supset_{LCG}\}$ *and* $\mathcal{P}$ *be the set of inference rules* $\{\otimes_R, \supset_L, \supset_{LC}, \wedge_{Li}, \vee_{Ri}\}$. *Then, (1)* $\mathcal{N}$ *has full permutability over* $\mathcal{P}$ *and* k*, (2)* k *has full permutability over* $\mathcal{P}$ *and (3)* $\mathcal{P}$ *has weak permutability.*

We can hence separate the proof steps into two phases. In the *negative phase* all negative rules are applied eagerly until the left context has only atomic negative formulas and the succedent formula is positive or atomic. When this process terminates, we may possibly apply k a number of times and the *positive phase* starts by choosing a negative formula on the left or a positive one on the right. The focusing persists until a positive formula appears in the left context or a negative one appears in the succedent, and we come back to the negative phase.

This determines a focused system for FL^k_{ew}, called FFL^k_{ew} (see Figure B.1 in the Appendix B), which (using Proposition 3.2) is sound and complete w.r.t. FL^k_{ew}.

Theorem 3.3 *The system* FFL^k_{ew} *is sound and complete w.r.t.* FL^k_{ew}.

4 Decidability and Complexity

The unfocused and focused systems above give rise to seemingly the first purely proof-theoretic proofs of decidability and complexity for FL^k_{ew} (a proof of the finite model property is in [GJ13]). Decidability in both systems follows from standard backwards proof search with a history mechanism to avoid loops.

Theorem 4.1 *The problems of deciding whether a sequent is derivable in* FL^k_{ew} *and in* FFL^k_{ew}*, respectively, are decidable.*

Proof. By performing a standard backwards proof search, storing every sequent encountered in the current branch of the attempted derivation in a *history*, i.e., a sequence of sequents, and only applying rules if none of their premises is in the history. Since all the rules have the subformula property, the number of sequents possibly appearing in a derivation is finite, and hence the proof search terminates. □

The complexity bounds provided by the previous theorem are however far from optimal, since in the worst case the procedure needs to visit every possible sequent on a single branch, giving an exponential space bound on naive proof search. This can be improved by considering *forward* instead of backwards proof search as follows.

Theorem 4.2 *The problem of deciding whether a formula is a theorem of* FL^k_{ew} *is in* EXPTIME.

Proof. All the rules of FL^k_{ew} have the subformula property, hence only sequents containing subformulae of the input formula can occur in a derivation of that formula. For an input formula of size n the first component of such a sequent is a set of subformulae of the input, the second component is a multiset containing w.l.o.g. at most $k-1$ copies of each subformula of the input (in case there are k copies of a formula we may apply the k-rule to transfer all of them into the classical context), and the right hand side is a single subformula of the input. So there are at most $2^n{\cdot}k^n{\cdot}n$ relevant sequents, each containing at most $n{+}k{\cdot}n{+}1$ formulae, so of size at most $n\cdot(n+k\cdot n+1)$. To check derivability, we implement a *forward search* procedure: given an input formula A construct the set S_A of relevant sequents as above. Then starting from the initial sequents in S_A apply all possible rules to the already constructed sequents and add the resulting sequents. After at most $\mathsf{card}(\mathsf{S}_A)$-many steps this reaches a fixpoint. Now check whether the sequent $.;. \vdash A$ is among the constructed ones. There is a fixed number of rules in FL^k_{ew}, every such rule has at most k premises, and computing the conclusion of such a rule can be done easily, thus each of the steps of the construction can be done in time exponential in n. Since there are only exponentially many such steps, the whole procedure runs in exponential time. □

While this gives an upper bound for the complexity of FL^k_{ew}, a PSPACE lower bound follows from the results in [HT11]. The exact complexity of FL^k_{ew} still seems to be open.

5 Lazy Splitting

In what we have presented so far, heavy proof theory machinery was used in order to present a "good proof system" for bounded contraction, in the sense that it is cut free and it has a notion of normal forms (via focusing). However, the proof systems proposed in the previous sections are still far from being suitable for implementation.

In fact, once a negative phase finishes, one has to decide either to apply the k rule or a positive rule. But this entails a huge problem for proof search: the logical context can be split in an exponential number of ways due to the rules k, $\otimes_R$, $\supset_L$ and $\supset_{LC}$. Observe that the focusing theorem shows that *if* a sequent is provable and *if* the proof uses the k rule, *then* you may apply it before the positive rules. But if this is not the case, then there will be useless attempts of splitting the context, and this is extremely inefficient. Thus, a naive implementation of FFL^k_{ew} (Figure B.1) does not work even for simple sequents, as shown in Section 5.1.

In linear logic, *lazy splitting* systems have been proposed in order to minimize the non-determinism during proof search (see e.g., [HM94,LP99,CHP00]). The idea is to separate the linear context into two: the formulas that will/will not be used in a branch of a derivation. In this way, one avoids the splitting of contexts: a branch of a derivation goes up with all the resources, consumes what is needed and then allows the "re-use" of the spare resources in the other branch. Hence the name *lazy*, as one postpones the decision of splitting, thus improving proof search.

Formally, in the system proposed in [LP99], the rule $\otimes_R$ becomes:

$$\frac{\Gamma; \cdot; \Delta :: E \vdash F \;/\; \Delta' :: E' \quad \Gamma; \Delta'; E' \vdash G \;/\; E''}{\Gamma; \Delta; E \vdash F \otimes G \;/\; E''} \otimes_R$$

where Γ represents the classical context and Δ the linear one. Moreover, E is a multiset of formulas representing an *excess* of resources (the *input*) that the sequent may *return* as an *output*. The notation $\Psi \vdash F \;/\; \Delta :: E$ means that, in order to prove F from the context Ψ, the formulas in the list of multisets $\Delta :: E$ were not used (*excess*). The distinction between Δ (*non-returnable*) and E (*returnable*) in the above derivation is used to define the scope of the formulas. This can be better understood with the analysis of the right rule for linear implication

$$\frac{\Gamma; \Delta, F; E \vdash G \;/E'}{\Gamma; \Delta; E \vdash F \multimap G \;/E'} \multimap_R$$

Observe that the scope of the formula F is the premise sequent, hence F cannot be returned in the conclusion sequent (via E') and it must be placed into the non-returnable context Δ.

The same techniques can be applied in FL_{ew}^k for the *linear logic connectives*, but the rule k still needs to decide how to split the context. Hence, even for "simple" provable sequents with few extra hypotheses (that may be weakened), a prover using focusing and lazy splitting on linear connectives is still not practical as shown in Section 5.1.

In the following we will present a new and non-trivial extension of lazy splitting to bounded contractions. More interestingly, we shall show that the lazy version of k can be *eagerly* applied without loosing provability. Hence, the results in Proposition 3.2 (k has full permutability over $\mathcal{P}$) lead, in fact, to a perfect implementation strategy.

Informally, the idea is to separate the classical context into 3 contexts (that should be thought as classical "wannabe" contexts), indicating the number of times that a formula was used in a derivation: none, once or many. If a formula was not used at all in an application of k, it is a *returnable excess* or *output*. If it was used exactly once, it is a *non-returnable excess*, that is, it is linear: this formula will be copied to the other branches of k but it will not be returned as an output. And if it was used 2 or more times, it is *classical* and it cannot be used in the other branches of k. The laziness for bounded contractions thus comes in k steps, where formulas are allowed to move from one context to a "more classical" one.

The rules for LFL_{ew}^k, the lazy system for FL_{ew}^k, are depicted in Figure 3 and explained in the following. Sequents in LFL_{ew}^k have the shape

$$\boxed{L}; \Delta; E \vdash G \;/\; \boxed{L'}; E'$$

where G is a formula, E, E' are lists of multisets of formulas, Δ is a multiset of formulas and L, L' are (possibly empty) lists of triples of the form $\boxed{\Gamma}; \boxed{\Omega_\Delta :: \Omega_E}; \boxed{N_\Delta :: N_E}$, where the subscripts Δ and E refer to the returnable /non returnable part of the contexts. When Δ, E are not important, we simply write Ω, N instead of $\Omega_\Delta :: \Omega_E$ and $N_\Delta :: N_E$. Moreover, when Γ, Ω and N are not important we write $\boxed{[A]}$.[4]

The context $\boxed{\Gamma}; \boxed{\Omega}; \boxed{N}$ is classical but each part obeys different rules, after the application of the lazy version of k:

- Γ contains all the formulas that have been used two or more times.

---

[4] A useful mnemonic. We use capital letters to denote (Γ)lassical, used (Ω)nce, (N)ot used, (E)xcess formulas and (A)ll.

- The context Ω contains the formulas that were used only once.
- N contains all the formulas that have not been used, that is, the *output*.

The contexts Δ and E follow the design principles for the lazy system for linear logic in [LP99]. The next example clarifies better the use of the new contexts.

Example 5.1 Here we will consider the system LFL^2_{ew} (that is, with with $k = 2$). For proving the sequent $a^2, b \vdash a^3 \otimes b$, we start by applying the lazy version of the k rule:

$$\dfrac{\begin{matrix}\pi_1\\ \boxed{\boxed{\boxdot;\boxdot;\boxdot} :: \boxdot;\boxdot;\boxed{a^2, b :: \cdot}}; \cdot; \cdot \vdash a^3 \otimes b \ / \ \boxed{\boxed{\boxdot;\boxdot;\boxdot} :: \boxed{\Gamma^1}; \boxed{\Omega^1}; \boxed{N^1}}; \cdot\end{matrix} \qquad \Upsilon_2}{\boxed{\boxdot;\boxdot;\boxdot}; a^2, b; \cdot \vdash a^3 \otimes b \ / \ \boxed{\boxdot;\boxdot;\boxdot}; \cdot}\ \mathsf{k}$$

As usual, the "$/[A]; E$" part (the output) is computed by rules in a top-down fashion. Hence, Γ^1, Ω^1, N^1 will be determined once π_1 is finished. Since E and $\boxed{L}$ are empty in the conclusion, all the returned output will also be empty. Observe that, in the left premise, we assume that the whole linear context $\{a^2, b\}$ is going to be used as part of the bounded contraction ($\boxed{N}$ context). The proof π_1 continues as follows (omitting the "$\boxed{\boxdot;\boxdot;\boxdot}$ ::" part):

$$\dfrac{\dfrac{}{\boxed{\boxdot;\boxdot;\boxed{a^2, b :: \cdot}}; \cdot; \cdot \vdash a \ / \ \boxed{\boxdot;\boxed{a :: \cdot};\boxed{a, b :: \cdot}}; \cdot}\ init \qquad \begin{matrix}\pi'_1\\ \boxed{\boxdot;\boxed{a :: \cdot};\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a^2 \otimes b \ / \ \boxed{\boxed{\Gamma^1}; \boxed{\Omega^1}; \boxed{N^1}}; \cdot\end{matrix}}{\boxed{\boxdot;\boxdot;\boxed{a^2, b :: \cdot}}; \cdot; \cdot \vdash a^3 \otimes b \ / \ \boxed{\boxed{\Gamma^1}; \boxed{\Omega^1}; \boxed{N^1}}; \cdot}\ \otimes_R$$

Note that, in the left derivation, one a is used in the initial axiom and hence it is moved to the $\boxed{\Omega}$ context in the output. The other formulas a, b are set in the N context and both Ω, N are passed to the right premise in the $\otimes_R$ rule – they are the excess.

Derivation π'_1 proceeds as follows:

$$\dfrac{\dfrac{}{\boxed{\boxdot;\boxed{a :: \cdot};\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a \ / \ \boxed{a};\boxdot;\boxed{a, b :: \cdot}; \cdot}\ I \qquad \dfrac{\Psi_1 \qquad\qquad \Psi_2}{\boxed{\boxed{a};\boxdot;\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a \otimes b \ / \ \boxed{\boxed{\Gamma^1}; \boxed{\Omega^1}; \boxed{N^1}}; \cdot}\ \otimes_R}{\boxed{\boxdot;\boxed{a :: \cdot};\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a^2 \otimes b \ / \ \boxed{\boxed{\Gamma^1}; \boxed{\Omega^1}; \boxed{N^1}}; \cdot}\ \otimes_R$$

where

$$\Psi_1 = \dfrac{}{\boxed{\boxed{a :: \cdot};\boxdot;\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a \ / \ \boxed{\boxed{a :: \cdot};\boxdot;\boxed{a, b :: \cdot}}; \cdot}\ I \qquad \Psi_2 = \dfrac{}{\boxed{\boxed{a :: \cdot};\boxdot;\boxed{a, b :: \cdot}}; \cdot; \cdot \vdash b \ / \ \boxed{\boxed{a :: \cdot};\boxed{b :: \cdot};\boxed{a :: \cdot}}; \cdot}\ I$$

Hence now we know that $\Gamma^1 = \{a\}, \Omega^1 = \{b :: \cdot\}$ and $N^1 = \{a :: \cdot\}$.

Let's come back to the right premise Υ_2. Since the output of the Υ_1 premise was $\boxed{\boxed{a :: \cdot};\boxed{b :: \cdot};\boxed{a :: \cdot}}$, the classical input for Υ_2 will be $\boxed{\boxdot;\boxdot;\boxed{a :: \cdot}}$ and the linear input will be $\{b :: \cdot\}$. Hence, $\Upsilon_2 = \boxed{\boxdot;\boxdot;\boxed{a :: \cdot}}; b; \cdot \vdash a^3 \otimes b \ / \ \boxed{\boxed{\Gamma^2}; \boxed{\Omega^2}; \boxed{N^2}}; \cdot$. With a similar derivation as the one just presented, it is easy to see that $\Gamma^2 = \{a\}$, $\Omega^2 = \{b :: \cdot\}$ and $N^2 = \emptyset$. Hence everything is consumed and the output of the rule k will be empty.

This mimics the following derivation in the system FL^2_{ew}

$$\dfrac{\overline{\overline{a; b \vdash a^3 \otimes b}} \qquad \overline{\overline{a; b \vdash a^3 \otimes b}}}{\cdot; a^2, b \vdash a^3 \otimes b}\ \mathsf{k}$$

where the a's are split by the bounded contraction and b remains in the linear context.

As we could see, the system "counts" the number of times a formula was used during a proof: none, once or several times. If the formula was not used at all (i.e., it remains in the $\boxed{N}$ context), then it will be placed in the classical context ($\boxed{N}$) of the next premise of the k rule. If it was used once ($\boxed{\Omega}$ context), then it must be part of the linear context in all premises. Finally, if it was used twice (or more), it will not be passed to the remaining premises.

Now let us explain the other rules in the LFL^k_{ew} system. The positive rules are similar to those in [LP99]. For instance, the rule $\otimes_R$ uses both Δ and E ($\Delta :: E$) in order to prove F. The formulas not used in the proof of F, i.e, $\Delta' :: E'$, are used in the proof of G. The final output of the proof of $F \otimes G$ is E'' (since Δ'' cannot be returned). Consider now the classical context in $\otimes_R$. The proof of F may move some formulas in $\boxed{L}$ producing $\boxed{L'}$ and, from $\boxed{L'}$, the proof of G may also move some other formulas producing $\boxed{L''}$. This "moving of formulas" is determined by the `take` procedure, as stated in Definition 5.2 below.

The negative rules $\wedge_R$ and $\vee_L$ need some adjustments w.r.t. the system in [LP99] due to the presence of weakening. Consider the following rule form [LP99]:

$$\frac{\Gamma; \Delta; \Pi \vdash F \;/\; E' \qquad \Gamma; \Delta; \Pi \vdash G \;/\; E'}{\Gamma; \Delta; E \vdash F \wedge G \;/\; E'}$$

In the case of linear logic, the proof of F and G must output exactly the same excess of formulas E'. Note that this is not the case in FL_{ew} since the rules internalize weakening. For instance, the sequent $a, b \vdash a \wedge b$ is provable and, while the proof of a outputs b, the proof of b outputs a. Hence, what we need as output is the multiset intersection (⩀) of the outputs of the two derivations. A similar situation happens with the classical context. For example, in a proof of $a \wedge a^2$ from the context $\boxed{\cdot}; \boxed{\cdot}; \boxed{a :: \cdot}$, the proof of a outputs $\boxed{\cdot}; \boxed{a :: \cdot}; \boxed{\cdot}$ and the proof of a^2 outputs $\boxed{a}; \boxed{\cdot}; \boxed{\cdot}$. Then, the proof of $a \wedge a^2$ outputs $\boxed{\cdot}; \boxed{a :: \cdot}; \boxed{\cdot} \;\widetilde{⩀}\; \boxed{a :: \cdot}; \boxed{\cdot}; \boxed{\cdot} = \boxed{a :: \cdot}; \boxed{\cdot}; \boxed{\cdot}$ where $\widetilde{⩀}$ chooses the left most position of a. Formally,

Definition 5.2 Let F be a formula, $\boxed{L}$ be a classical context and E and Δ be multisets of formulas. Assume that F occurs either in $\boxed{L}$, E or Δ. If F occurs in Δ, then $\mathtt{take}(F, \Delta, \boxed{L}; E) = \boxed{L}; E$. If F does not occur in Δ, then `take` chooses one of the occurrences of F and produces $\boxed{L'}; E'$ as follows:

- If F is taken from E, then $\boxed{L'} = \boxed{L}$ and $E' = E \setminus \{F\}$
- Let $\boxed{L} = \boxed{L_h :: \boxed{\Gamma}; \boxed{\Omega}; \boxed{N} :: L_t}$ where L_h and L_t are (possibly empty) lists of triples. If F is taken from $\boxed{\Gamma}; \boxed{\Omega}; \boxed{N}$, then $E = E'$ and we have three choices. If F is taken from Γ, then $\boxed{L'} = \boxed{L}$. If F is taken from Ω then $\boxed{L'} = \boxed{L_h :: \boxed{\Gamma \cup \{F\}}; \boxed{\Omega \setminus \{F\}}; \boxed{N} :: L_t}$. Finally, if F is taken from N, then $\boxed{L'} = \boxed{L_h :: \boxed{\Gamma}; \boxed{\Omega \uplus \{F\}}; \boxed{N \setminus \{F\}} :: L_t}$.[5]

Let $\boxed{[A_1]} = \boxed{\Gamma_1}; \boxed{\Omega_1}; \boxed{N_1}$ and $\boxed{[A_2]} = \boxed{\Gamma_2}; \boxed{\Omega_2}; \boxed{N_2}$ be classical contexts containing the same multiset of formulas, where each formula F may occur at possibly different positions (1 for N, 2 for Ω and 3 for Γ). The context $\boxed{[A_3]} = \boxed{[A_1]} \;\widetilde{⩀}\; \boxed{[A_2]}$ is obtained as follows.

(i) Start with $\boxed{[A_3]} = \boxed{\cdot}; \boxed{\cdot}; \boxed{\cdot}$.

---

[5] We stress out that we always take/add a formula F from/to the context respecting the Δ, E contexts. Hence, for example, if $F \in N_\Delta$, then $N \setminus \{F\}$ actually means $N \setminus \{F :: \cdot\}$.

Negative Rules

$$\frac{}{\boxed{L}; \Delta, \bot; E \vdash C \ / \ \boxed{L}; E}\ \bot_L \qquad \frac{}{\boxed{L}; \Delta; E \vdash \top \ / \ \boxed{L}; E}\ \top_R$$

$$\frac{\boxed{L}; \Delta, F, G; E \vdash C \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta, F \otimes G; E \vdash C \ / \ \boxed{L'}; E'}\ \otimes_L \qquad \frac{\boxed{L}; \Delta, F; E \vdash G \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta; E \vdash F \supset G \ / \ \boxed{L'}; E'}\ \supset_R$$

$$\frac{\boxed{L}; \Delta; E \vdash F \ / \ \boxed{L_1}; E_1 \qquad \boxed{L}; \Delta; E \vdash G \ / \ \boxed{L_2}; E_2}{\boxed{L}; \Delta; E \vdash F \wedge G \ / \ \boxed{L_1 \widetilde{\Cap} L_2}; E_1 \Cap E_2}\ \wedge_R$$

$$\frac{\boxed{L}; \Delta, F; E \vdash C \ / \ \boxed{L_1}; E_1 \qquad \boxed{L}; \Delta, G; E \vdash C \ / \ \boxed{L_2}; E_2}{\boxed{L}; \Delta, F \vee G; E \vdash C \ / \ \boxed{L_1 \widetilde{\Cap} L_2}; E_1 \Cap E_2}\ \vee_L$$

Positive Rules

$$\frac{\boxed{L}; \cdot; \Delta :: E \vdash F \ / \ \boxed{L'}; \Delta' :: E' \qquad \boxed{L'}; \cdot; \Delta' :: E' \vdash G \ / \ \boxed{L''}; \Delta'' :: E''}{\boxed{L}; \Delta; E \vdash F \otimes G \ / \ \boxed{L''}; E''}\ \otimes_R$$

$$\frac{\boxed{L}; \cdot; \Delta :: E \vdash F \ / \ \boxed{L'}; \Delta' :: E' \qquad \boxed{L'}; G; \Delta' :: E' \vdash C \ / \ \boxed{L''}; \Delta'' :: E''}{\boxed{L}; \Delta, F \supset G; E \vdash C \ / \ \boxed{L''}; E''}\ \supset_L$$

$$\frac{\boxed{L}; \Delta, F_i; E \vdash C \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta, F_1 \wedge F_2; E \vdash C \ / \ \boxed{L'}; E'}\ \wedge_{Li} \qquad \frac{\boxed{L}; \Delta; E \vdash F_i \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta; E \vdash F_1 \vee F_2 \ / \ \boxed{L'}; E'}\ \vee_{Ri}$$

Structural Rules

$$\frac{\mathtt{take}(p, \Delta, \boxed{L}; E) = \boxed{L'}; E'}{\boxed{L}; \Delta; E \vdash p \ / \ \boxed{L'}; E'}\ \mathsf{init}$$

$$\frac{\boxed{L \setminus F}; \Delta, F; E \vdash C \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta; E \vdash C \ / \ \boxed{L'}; E'}\ D_L \qquad \frac{\boxed{L}; \Delta, F; E \vdash C \ / \ \boxed{L'}; E'}{\boxed{L}; \Delta; E, F \vdash C \ / \ \boxed{L'}; E'}\ D_E$$

$$\frac{\Upsilon_1 \qquad \Upsilon_2 \qquad \cdots \quad \Upsilon_k}{\boxed{L}; \Delta; E \vdash C \ / \ \boxed{L^k}; E^k}\ \mathsf{k}$$

$$\Upsilon_1 = \boxed{L :: \boxed{\cdot}; \boxed{\cdot}; \boxed{\Delta :: E}}; \cdot; \cdot \vdash C \ / \ \boxed{L^1 :: \boxed{\Gamma^1}; \boxed{\Omega_\Delta :: \Omega_E}; \boxed{\Delta^1 :: E^1}}; \cdot$$

$$\Upsilon_i = \boxed{L^{i-1} :: \boxed{\cdot}; \boxed{\cdot}; \boxed{\Delta^{i-1} :: E^{i-1}}}; \Omega_\Delta; \Omega_E \vdash C \ / \ \boxed{L^i :: \boxed{\Gamma^i}; \boxed{\Omega^i}; \boxed{\Delta^i :: E^i}}; \Omega^i_E, \ 2 \leq i \leq k$$

Fig. 3. Lazy-splitting LFL^k_{ew} system. In the init axiom, p is atomic. Rules $\&_L$, $\vee_{LC}$, $\supset_{LCC}$, $\supset_{LCG}$ in Figure 2 apply also on the $\boxed{\Gamma}$ context. $\Cap$ represents multiset intersection. The definitions of $\widetilde{\Cap}$ and $\mathtt{take}(F, \Delta, \boxed{L}; E)$ are given in Definition 5.2. In rule D_L, $L \setminus F$ means $\mathtt{take}(F, \cdot, \boxed{L}; \cdot)$. Rule init assumes that p occurs either in Δ, $\boxed{L}$ or E. In k, $\{\Delta, E\}$ must be non-empty.

(ii) Let F be a formula in $\boxed{[A_1]}$ and let $n, m \in \{1, 2, 3\}$ be the left most positions of F in $\boxed{[A_1]}$ and $\boxed{[A_2]}$ respectively.

(iii) Add F at position $max(n, m)$ in $\boxed{[A_3]}$.

(iv) Delete one occurrence of F at position n (resp. m) in $\boxed{[A_1]}$ (resp. $\boxed{[A_2]}$).

(v) If $\boxed{[A_1]} \neq \boxed{\cdot};\boxed{\cdot};\boxed{\cdot}$, goto to (ii).

Given two lists of triples $\boxed{L}$ and $\boxed{L'}$ of the same length and with each correspondent triple containing the same multiset of formulas, we define $\boxed{L} \mathbin{\widetilde{⩀}} \boxed{L'}$ by point-wise applying $\widetilde{⩀}$ to each correspondent triple.

Observe that the changes in the context $\boxed{L}$ are governed by the initial rule, due to the `take` procedure, as follows: if p occurs in Δ (which is non returnable), p is taken from that context and the input and output coincide; otherwise, p is removed from E or it is shifted to the left in the $\boxed{L}$ context.

Theorem 5.3 (Soundness and Completeness) *The sequent* $\cdot; \Delta \vdash G$ *is provable in* FL_{ew}^{k} *if and only if* $\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\cdot}}; \Delta; \cdot \vdash G$ *is provable in* LFL_{ew}^{k}.

Proof. The proof follows by mutual induction on sequents. We will show here how to mimic a proof of $\cdot; \Delta \vdash G$ into LFL_{ew}^{k} and vice-versa.

Starting from the sequent $\cdot; \Delta \vdash G$ in FL_{ew}^{k}, one applies negative rules until only k or positive rules can be applied. These will match exactly the moves for applying negative rules for proving $\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\cdot}}; \Delta; \cdot \vdash G$ in LFL_{ew}^{k}, since only the non returnable linear context is considered.

Consider now that the rule k is applied (for the first time):

$$\dfrac{\begin{array}{c}\pi_1\\ \Gamma_1; \Delta \vdash G\end{array} \quad \cdots \quad \begin{array}{c}\pi_k\\ \Gamma_k; \Delta \vdash G\end{array}}{\cdot; \Delta, \Gamma_1, \ldots, \Gamma_k \vdash G}\ \mathsf{k}$$

We may assume, for every $1 \leq i \leq k$ and without loss of generality, that formulas in Γ_i (resp. Δ) were used two or more times (resp. at most once) in π_i, since Δ is linear (with weakening). Moreover, if $F \in \Gamma_i$ was used at most once, it can be moved to Δ (again, due to weakening).

On mimicking π_1 in order to have a proof π_1' of the sequent

$$\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\Delta, \Gamma_1, \ldots, \Gamma_k :: \cdot}}; \cdot; \cdot \vdash G \ / \ \boxed{\boxed{\Gamma^1};\boxed{\Omega^1};\boxed{N^1}}; \cdot$$

we have that: $\Gamma^1 = \Gamma_1$ since formulas in Γ_1 were used two or more times, Ω^1 are formulas in Δ used exactly once in π_1, and $N^1 = \Delta_2, \ldots, \Delta_k, (\Delta - \Omega^1) :: \cdot$, since those formulas were never used in π_1. Due to weakening, we may adapt π_1' so that $\Omega^1 = \Delta$.

Following the same argument, we can mimic π_i to a proof of the sequent

$$\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\Gamma_i, \ldots, \Gamma_k :: \cdot}}; \Delta; \cdot \vdash G \ / \ \boxed{\boxed{\Gamma_i};\boxed{\Omega^i};\boxed{\Gamma_{i+1}, \ldots, \Gamma_k :: \cdot}}; \cdot$$

Hence the output of $\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\cdot}}; \Delta, \Gamma_1, \ldots, \Gamma_k :: \cdot; \cdot \vdash G$ will be $\boxed{\boxed{\cdot};\boxed{\cdot};\boxed{\cdot}}; \cdot$.

The positive rules are handled in the exact same way as in [LP99], so we will omit the details.

For the converse, let's take a closer look at the (lazy) rule k

$$\frac{\Upsilon_1 \qquad \Upsilon_2 \qquad \cdots \quad \Upsilon_k}{\boxed{L}; \Delta; E \vdash C \;/\; \boxed{L^k}; E^k}\ \mathsf{k}$$

On trying to prove the formula C, we move the whole linear context Δ, E to the "classical wannabe" context $\boxed{L :: \boxdot; \boxdot; \boxed{\Delta :: E}}$. Some formulas in Δ, E are linearly consumed $(\Omega_\Delta, \Omega_E)$, some are classically consumed (Γ^1) and some are not used at all (Δ^1, E^1). We now try to prove the same C with the same resources *but* the classical formulas in Γ^1.

$$\boxed{L :: \boxdot; \boxdot; \boxed{\Delta^1 :: E^1}}; \Omega_\Delta, \Omega_E; \cdot \vdash C$$

This will consume formulas in $\Omega_\Delta, \Omega_E, \Delta^1, E^1$: some classically (Γ^2), some linearly (Ω^2) and some will not be used at all $(\Delta^2, E^2, \Omega^2_E$, where $\Omega^2_E = \Omega_\Delta, \Omega_E - \Omega^i)$. And so on. This process mimics perfectly the application of the rule k in FL^k_{ew}, where part of the linear context becomes classical and it is not delivered to the other branches of derivation. □

In Appendix C we present some examples of derivations in both systems, to illustrate better the use of contexts and rules.

5.1 Eager lazy k rule and implementation

We have implemented in Maude `http://maude.cs.illinois.edu` the system FFL^2_{ew} as it is in Figure B.1 (for $k = 2$), FFL^2_{ew} with lazy splitting for the linear logic connectives only (*lazy-*FL^2_{ew} system) and the system LFL^2_{ew} in Figure 3. In this section, we report some experiments on these systems. All the tests were performed on an iMac, 2.9 GHz Intel Core i5 with 8 GB of RAM running Maude 2.7. The source code and examples can be downloaded from `http://subsell.logic.at/flew`. We stress out that, although the examples were implemented for $k = 2$, the implementation for any k is a trivial adaptation of the systems implemented so far.

Table 5.1 summarizes the results of the tests. We tested the canonical examples for cut-elimination as those described in Section 2. As expected, the lazy strategies improved considerably the efficiency of the system. The advantages of using the lazy version of k can be appreciated in the last entries of the table. In those experiments, we added into the context some "useless" formulas, that can be weakened. It can be noted that the uncontrolled use of k in the system *lazy-*FL^2_{ew} affects considerable the performance of the solver.

Some engineering can be used in order to improve the proof search procedure. For instance, the rule init in System LFL^k_{ew} introduces a non-determinism (predicate `taken`). As pointed out in Definition 5.2, we can eliminate some of the choices by taking the atom p from the context Δ that cannot be returned. If the atom is not in Δ, it should be taken from the context Γ and the output coincides with the input. In other case, it must be taken from the other contexts.

Other improvements can be also applied. For instance, it is easy to prove that the following context transformations are safe and improve the search procedure (since they reduce the number of formulas in the context):

- $\boxed{\Gamma}; \boxed{\Omega}; \boxed{N, F, F} \rightsquigarrow \boxed{\Gamma \cup \{F\}}; \boxed{\Omega}; \boxed{N, F}$.
- $\boxed{\Gamma}; \boxed{\Omega}; \boxed{N}; \Delta, F, F; E \rightsquigarrow \boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta; E$.

Sequent	FFL^2_{ew}	$lazy\text{-}FL^2_{ew}$	LFL^2_{ew}
$p \supset \bot, (p \supset q) \vdash (p \supset q)^3$	2ms	2ms	7 ms
$r, p \supset r \supset q, p \supset q \vdash (p \supset q)^3$	47ms	18ms	8ms
$r, p \supset r \supset q, (p \supset q)^3 \supset s \vdash s$	77ms	33ms	8ms
$a, r, p \supset r \supset q, (p \supset q)^3 \supset s \vdash s$	4.2s	168s	55ms
$a, b, r, p \supset r \supset q, (p \supset q)^3 \supset s \vdash s$	-	2.8m	55ms

Table 1
Tests of the systems FFL^k_{ew} (Figure B.1), $lazy\text{-}FL^2_{ew}$ (FFL^k_{ew} + lazy rules for linear logic connectives) and $LFFL^k_{ew}$ (Figure 3). "-" means more than 5 minutes.

- $\boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta, F; E \rightsquigarrow \boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta; E.$
- $\boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta; E, F \rightsquigarrow \boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta; E.$
- $\boxed{\Gamma, F, F}; \boxed{\Omega}; \boxed{N}; \Delta; E \rightsquigarrow \boxed{\Gamma, F}; \boxed{\Omega}; \boxed{N}; \Delta; E.$

More interestingly and actually quite surprising, Proposition 5.5 below shows a sufficient condition for determining if an application of the rule k can be avoided. For that, we introduce the system $KLFL_{ew}$ that differs from LFL^k_{ew} only in the rule k.

Definition 5.4 [System $KLFL_{ew}$] The system $KLFL_{ew}$ shares with LFL^k_{ew} all the proof rules but k, that is substituted by the following rules

$$\dfrac{\Upsilon_1 \quad \cdots \quad \Upsilon_k \quad \star}{\boxed{L}; \Delta; E \vdash C \;/\; \boxed{L^k}; E^k}\ \mathsf{k}_1 \qquad \dfrac{\Upsilon_1 \quad \overline{\star}}{\boxed{L}; \Delta; E \vdash C \;/\; \boxed{L^1}; E^1}\ \mathsf{k}_2$$

where $\Upsilon_1, \ldots, \Upsilon_k$ are as in Figure 3 and $\star$ (resp. $\overline{\star}$) means that $\Gamma_1 \neq \cdot$ (resp. $\Gamma_1 = \cdot$). Similar to LFL^k_{ew}, $\{\Delta, E\}$ must be non-empty.

Rule k_1 is the same as k in the LFL^k_{ew} system. Rule k_2 detects whether the application of k is "useless". This happens when the formulas were used at most once and then, the $\boxed{\Gamma^1}$ context remains empty. In that case, a proof of Υ_1 is indeed a proof of the sequent in the conclusion. For instance, consider a proof of the sequent $a, b \vdash a \otimes b$ where k (in LFL^k_{ew}) is not needed. If we (eagerly) apply k, we obtain the following (omitting the redundant premises):

$$\dfrac{\overset{\pi_1}{\boxed{\boxdot; \boxdot; \boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a \otimes b \;/\; \boxed{\boxdot; \boxed{a, b :: \cdot}; \boxdot}} \qquad \overset{\pi_2}{\boxed{\boxdot; \boxdot; \boxdot}; a, b; \cdot \vdash a \otimes b \;/\; \boxed{[L^2]}; E^2} \qquad \ldots}{\boxed{[\cdot]}; a, b; \cdot \vdash a \otimes b \;/\; \boxed{[\cdot]}; \cdot}\ \mathsf{k}$$

Note that π_1 (up to some minor syntactic conversions) is indeed the proof of the sequent in the conclusion (which is the same as the ones on the other premises). Since π_1 only uses once the formulas a, b, they are returned in the $\boxed{\Omega}$ context and $\boxed{\Gamma}$ remains empty. Then, we have the following proof in $KLFL_{ew}$

$$\dfrac{\overset{\pi_1}{\boxed{\boxed{[\cdot]} :: \boxdot; \boxdot; \boxed{a, b :: \cdot}}; \cdot; \cdot \vdash a \otimes b \;/\; \boxed{\boxed{[\cdot]} :: \boxdot; \boxed{a, b :: \cdot}; \boxdot}}}{\boxed{[\cdot]}; a, b; \cdot \vdash a \otimes b \;/\; \boxed{[\cdot]}; \cdot}\ \mathsf{k}_2$$

that ignores the useless application of k.

Proposition 5.5 *The sequent* $\boxed{L}; \Delta; E \vdash G$ *is provable in* LFL^k_{ew} *iff it is provable in* $KLFL_{ew}$.

Proof. ($\Rightarrow$) If there is a proof of a sequent Υ in $KLFL_{ew}$, it is easy to produce a proof of Υ in LFL^k_{ew} by simply dropping the applications of k_2 and substituting any instance of k_1 with k in LFL^k_{ew}.

($\Leftarrow$) We proceed by induction on the length of the derivation of the sequent Υ in LFL^k_{ew}. We need to show that k_i can be always introduced before any other connective (belonging to the positive phase). Consider for instance the following derivation where k is not used in π_i – since k permutes down due to Proposition 3.2 (we omit some of the outputs):

$$\dfrac{\overset{\pi_1}{\boxed{L}; \cdot; \Delta :: E \vdash G \;/\; \boxed{L'}; \Delta' :: E'} \qquad \overset{\pi_2}{\boxed{L'}; \cdot; \Delta' :: E' \vdash G'}}{\boxed{L}; \Delta; E \vdash G \otimes G'}\ \otimes_R$$

In $KLFL_{ew}$, we have the following derivation (we also omit some of the outputs):

$$\dfrac{\dfrac{\overset{\pi_1'}{\boxed{L :: \boxed{\cdot}; \boxed{\cdot}; \boxed{\Delta :: E}}; \cdot; \cdot \vdash G} \qquad \overset{\pi_2'}{\boxed{L' :: \boxed{\cdot}; \boxed{\Delta_1 :: E_1}; \boxed{\Delta_2 :: E_2}}; \cdot; \cdot \vdash G'}}{\boxed{L :: \boxed{\cdot}; \boxed{\cdot}; \boxed{\Delta :: E}}; \cdot; \cdot \vdash G \otimes G'}\ \otimes_R}{\boxed{L}; \Delta; E \vdash G \otimes G'}\ \mathsf{k}_2$$

Derivations π_1' and π_2' can be easily built from π_1 and π_2 (up to some simple syntactic conversions). We note that $\Delta_1 \cup \Delta_2 = \Delta$ and similarly for E. The interesting part is that the output of the derivation on the left must be $\boxed{L'}; \Delta :: E$ since none of the formulas in Δ, E was moved into the Γ contexts. This justifies the use of k_2. □

6 Conclusion and future work

In this work we introduced focused proof systems FFL^k_{ew} ($k > 1$) for Full Lambek Calculus with exchange and weakening extended with bounded contractions. The system was used to show that the validity problem is in EXPTIME. Although, proof theoretically speaking, these are satisfactory results, they do not lead to good implementation strategies.

We then proposed a new and non-trivial extension of lazy splitting to bounded contractions, showing soundness and completeness of the resulting systems LFL^k_{ew}. This notion of laziness for bounded contractions is crucial for implementing good provers for all the logics considered in this paper.

There are a number of ways of continuing this work. The first goal is to determine the exact complexity of FL_{ew} plus bounded contractions. That is, knowing that checking validity in such systems is PSPACE-hard [HT11], Theorem 4.2 implies that such complexity can be either PSPACE-complete or EXPTIME-complete.

Also, all the results presented in this paper strongly rely on the fact that the base logical system allows weakening. Without it, the situation is more complicated. For example, there is no more a clear separation between formulas behaving classically or linearly, and hence the notion of lazy splitting has to be completely reformulated.

We are currently working on implementing the system *KLFL*$_{ew}$ (Definition 5.4) (a prototypical version can be found at `http://subsell.logic.at/flew`). The main challenge is guaranteeing termination (note that one implication on the $\boxed{\Gamma}$ context may generate a loop). The implementation of the system in Figure 3 guarantees termination by keeping the history of sequents and detecting loops (see Theorem 4.1). Loop detection in *KLFL*$_{ew}$ is harder: after applying $\multimap_L$, some formulas can be moved into the linear context and then, the lazy version of k can be eagerly applied again. Hence, the sequents in the history are not exactly the same since the list of $\boxed{[A]}$ contexts grows after applying k. We can rely on Theorem 4.2 to bound the size of such list. However, we believe that something better can be done: if a new $\boxed{[A]}$ context is added and it shares with the previous ones the same formulas in the $\boxed{\Omega}$ and $\boxed{N}$ contexts, a loop is detected.

The bounded contraction rules are a proper subset of the sequent rules that can be automatically extracted from the Hilbert axioms belonging to the class $\mathcal{N}_2$ in the hierarchy of [CGT08] (see `http://www.logic.at/people/lara/axiomcalc.html` for an implementation of this algorithm). A great step forward would be to transform all sequent calculi for substructural logics generated by the algorithm in [CGT08] into efficient automatic provers.

References

[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.

[CHP00] Iliano Cervesato, Joshua S. Hodas, and Frank Pfenning. Efficient resource management for linear logic proof search. *Theoretical Computer Science*, 232(1-2):133–163, 2000.

[DHK03] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *J. of Automated Reasoning*, 31(1):31–72, 2003.

[Dyc92] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *J. of Symbolic Logic*, 57(3):795–807, September 1992.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[GJ13] N. Galatos and P. Jipsen. Residuated frames with applications to decidability. *Transactions of the AMS*, 365(3), 2013.

[GJKO07] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: an algebraic glimpse at substructural logics*. Elsevier, 2007.

[HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.

[HOS94] R. Hori, H. Ono, and H. Schellinx. Extending intuitionistic linear logic with knotted structural rules. *Notre Dame Journal of Formal Logic*, 35(2):219–242, 1994.

[HT11] Rostislav Horcik and Kazushige Terui. Disjunction property and complexity of substructural logics. *Theoretical Computer Science*, 412:3992–4006, 2011.

[LP99] Pablo López and Ernesto Pimentel. Resource management in linear logic search revisited. In Harald Ganzinger, David A. McAllester, and Andrei Voronkov, editors, *Logic Programming and Automated Reasoning, 6th International Conference, LPAR'99, Tbilisi, Georgia, September 6-10, 1999, Proceedings*, volume 1705 of *Lecture Notes in Computer Science*, pages 304–319. Springer, 1999.

[MS07] Dale Miller and Alexis Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of *LNCS*, pages 405–419. Springer, 2007.

[OT99] M. Okada and K. Terui. The finite model property for various fragments of intuitionistic linear logic. *J. Symbolic Logic*, 64(2):790–802, 1999.

[Pri96] Andreja Prijatelj. Bounded contraction and gentzen-style formulation of lukasiewicz logics. *Studia Logica*, 57(2/3):437–456, 1996.

A Permutability of Rules

We present here all the counter-examples for the non-permutability cases between rules in FL^k_{ew} (see Proposition 3.2).

$\mathcal{P} \not\downarrow \mathsf{k}$:

- $\otimes_R \not\downarrow \mathsf{k}$:

$$\cfrac{\overline{\overline{A^{k+1} \vdash A^{k+1}}}\ \otimes_R\ k+1 \text{ times, init}}{A^k \vdash A^{k+1}}\ \mathsf{k}$$

- $\supset_L \not\downarrow \mathsf{k}$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1}}} \quad \overline{\overline{A; B \vdash A^{k+1} \otimes B}}}{A; A^{k+1} \supset B \vdash A^{k+1} \otimes B}\ \supset_L}{A^k, A^{k+1} \supset B \vdash A^{k+1} \otimes B}\ \mathsf{k}$$

- $\wedge_L \not\downarrow \mathsf{k}$:

$$\cfrac{\cfrac{\overline{\overline{A \supset D; A^{k+1} \vdash D^{k+1} \vee C^{k+1}}}}{A \supset D; A^{k+1} \wedge B^{k+1} \vdash D^{k+1} \vee C^{k+1}}\ \wedge_L \quad \cfrac{\overline{\overline{B \supset C; B^{k+1} \vdash D^{k+1} \vee C^{k+1}}}}{B \supset C; A^{k+1} \wedge B^{k+1} \vdash D^{k+1} \vee C^{k+1}}\ \wedge_L}{A^{k+1} \wedge B^{k+1}, A \supset D, B \supset C \vdash D^{k+1} \vee C^{k+1}}\ \mathsf{k}$$

- $\vee_R \not\downarrow \mathsf{k}$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1}}}}{A; \cdot \vdash A^{k+1} \vee B^{k+1}}\ \vee_{L1} \quad \cfrac{\overline{\overline{B; \cdot \vdash B^{k+1}}}}{B; \cdot \vdash A^{k+1} \vee B^{k+1}}\ \vee_{L2}}{A^k, B^k \vdash A^{k+1} \vee B^{k+1}}\ \mathsf{k}$$

$\mathsf{k} \not\downarrow \mathcal{N}$:

- $\mathsf{k} \not\downarrow \supset_R$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1}}}}{A^k \vdash A^{k+1}}\ \mathsf{k}}{A^{k-1} \vdash A \supset A^{k+1}}\ \supset_R$$

- $\mathsf{k} \not\downarrow \vee_L$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1} \vee (A \otimes B)}}}{A^k \vdash A^{k+1} \vee (A \otimes B)}\ \mathsf{k} \quad \cfrac{}{A^{k-1}, B \vdash A^{k+1} \vee (A \otimes B)}}{A^{k-1}, A \vee B \vdash A^{k+1} \vee (A \otimes B)}\ \vee_L$$

- $\mathsf{k} \not\downarrow \wedge_R$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1}}} \quad \overline{\overline{A \wedge B; \cdot \vdash A^{k+1}}}}{A, A \wedge B \vdash A^{k+1}}\ \mathsf{k} \quad \cfrac{\overline{\overline{A \wedge B \vdash B}}\ \wedge_{2L}, \text{init} \quad \cfrac{}{A \vdash A}\ \text{init}}{A, A \wedge B \vdash B \otimes A}\ \otimes_R}{A, A \wedge B \vdash A^{k+1} \wedge (B \otimes A)}\ \wedge_R$$

- $\mathsf{k} \not\downarrow \otimes_L$:

$$\cfrac{\cfrac{\overline{\overline{A; \cdot \vdash A^{k+1}}}}{A, A, A^{k-2} \vdash A^{k+1}}\ \mathsf{k}}{A \otimes A, A^{k-2} \vdash A^{k+1}}\ \otimes_L$$

B Focused knotted system FFL^{k}_{ew}

Focusing [And92] is a discipline on proofs aiming at reducing the non-determinism during proof search. Focused proofs can be interpreted as the normal form proofs.

FL^{k}_{ew} connectives are separated into two classes, the *negative*: $\supset, \wedge$ and the *positive*: $\otimes, \vee$. The polarity of non-atomic formulas is inherited from its outermost connective and *any* bias is assigned to atomic formulas.

Observe, in Figure 2, that the negative connectives have invertible *right* rules, while the positive connectives have invertible *left* rules. This separation induces a two phase proof construction: a *negative*, where no *backtracking* on the selection of inference rules is necessary, and a *positive*, where choices within inference rules can lead to failures for which one may need to backtrack. The other rules in the negative phase are derived from well known equivalences in intuitionistic logic.

We separate the left context of sequents in FFL^{k}_{ew} in three: the set Δ will always denote the unbounded context; Γ is a linear context containing only negative or atomic formulas; and Θ is a general linear context. We will differentiate focused and unfocused sequents by using different arrow symbols: "$\Rightarrow$" for unfocused and "$\rightarrow$" for focused. In this way, FFL^{k}_{ew} contains four types of sequents:

i. $\Delta; \Gamma; \Theta \Rightarrow G$ is an unfocused sequent.

ii. $\Delta; \Gamma; \cdot \Rightarrow G$ is an unfocused sequent representing the end of a negative phase.

iii. $\Delta; \Gamma \rightarrow [F]$ is a sequent focused on the right.

iv. $\Delta; \Gamma, [F] \rightarrow G$ is a sequent focused on the left.

In the negative phase, sequents have the shape (*i*) above and all the negative formulas on the right and all the positive non-atomic formulas on the left are introduced. Also, atomic and negative formulas on the left are moved to the left linear context Γ using the `store` rule. When this phase ends, sequents have the form (*ii*).

Then either the rule k is applied or the positive phase begins. Observe that, after applying the *k* rule, it is mandatory to chose, via one of the decide rules D_l, D_{lc} or D_r, a formula on which to focus, enabling sequents of the forms (*iii*) or (*iv*). Rules are then applied on the focused formula until either an axiom is reached (in which case the proof ends) or a negative subformula on the right or a positive subformula on the left is derived (and the proof switches to the negative phase again). This means that focused proofs can be seen (bottom-up) as a sequence of alternations between negative and positive phases. We will call a *focused phase* a positive phase followed by a negative one.

C Some interesting examples involving lazy splitting

Example C.1 This is an example showing the role of the $\Delta; E$ contexts. Let $L = \boxed{\cdot}; \boxed{\cdot}; \boxed{\cdot}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\boxed{L}; \cdot; c :: \{\{a,b\} :: \cdot\} \vdash a \,/\, \boxed{L}; c :: \{b :: \cdot\}}\,\mathsf{init} \quad \dfrac{}{\boxed{L}; \cdot; c :: \{b :: \cdot\} \vdash c \,/\, \boxed{L}; \cdot :: \{b :: \cdot\}}\,\mathsf{init}}{\boxed{L}; c; \{a,b\} :: \cdot \vdash (a \otimes c) \,/\, \boxed{L}; \cdot :: \{b :: \cdot\}}\,\otimes_R}{\boxed{L}; \cdot; \{a,b\} :: \cdot \vdash (c \supset (a \otimes c)) \,/\, \boxed{L}; \cdot :: \{b :: \cdot\}}\,\supset_R \quad \dfrac{}{\boxed{L}; \cdot; b :: \cdot \vdash b \,/\, \boxed{L}; \cdot}\,\mathsf{init}}{\boxed{L}; a, b; \cdot \vdash (c \supset (a \otimes c)) \otimes b \,/\, \boxed{L}; \cdot}\,\otimes_R$$

Example C.2 Assume that $F \& F'$ is at the Ω_E context of $\boxed{[L]}$ and $\boxed{[L']}$ moves that formula to the Γ context. If we decide to use/focus on that formula (D_L), then we observe:

Negative Phase

$$\frac{}{\Delta;\Gamma;\Theta,\bot \Rightarrow C}\ \bot_L \qquad \frac{}{\Delta;\Gamma;\Theta \Rightarrow \top}\ \top_R \qquad \frac{\Delta;\Gamma;\Theta,F,G \Rightarrow C}{\Delta;\Gamma;\Theta,F \otimes G \Rightarrow C}\ \otimes_L$$

$$\frac{\Delta;\Gamma;\Theta,F \Rightarrow G}{\Delta;\Gamma;\Theta \Rightarrow F \supset G}\ \supset_R \qquad \frac{\Delta;\Gamma;\Theta \Rightarrow F \quad \Delta;\Gamma;\Theta \Rightarrow G}{\Delta;\Gamma;\Theta \Rightarrow F \wedge G}\ \wedge_R \qquad \frac{\Delta;\Gamma;\Theta,F \Rightarrow C \quad \Delta;\Gamma;\Theta,G \Rightarrow C}{\Delta;\Gamma;\Theta,F \vee G \Rightarrow C}\ \vee_L$$

$$\frac{\Delta,F,G;\Gamma;\Theta \Rightarrow C}{\Delta,F \,\&\, G;\Gamma;\Theta \Rightarrow C}\ \&_L\ \& \in \{\wedge,\otimes\} \qquad \frac{\Delta,F;\Gamma;\Theta \Rightarrow C \quad \Delta,G;\Gamma;\Theta \Rightarrow C}{\Delta,F \vee G;\Gamma;\Theta \Rightarrow C}\ \vee_{LC}$$

$$\frac{\Delta,G;\Gamma;\Theta \Rightarrow C}{\Delta,F \supset G,F;\Gamma;\Theta \Rightarrow C}\ \supset_{LCC} \qquad \frac{\Delta,G;\Gamma;\Theta \Rightarrow C}{\Delta,F \supset G,G;\Gamma;\Theta \Rightarrow C}\ \supset_{LCG}$$

Positive Phase

$$\frac{\Delta;\Gamma_1 \rightarrow [F] \quad \Delta;\Gamma_2 \rightarrow [G]}{\Delta;\Gamma_1,\Gamma_2 \rightarrow [F \otimes G]}\ \otimes_R$$

$$\frac{\Delta;\Gamma_1 \rightarrow [F] \quad \Delta;\Gamma_2,[G] \rightarrow C}{\Delta;\Gamma_1,\Gamma_2,[F \supset G] \rightarrow C}\ \supset_L \qquad \frac{\Delta;\Gamma,[F_i] \rightarrow C}{\Delta;\Gamma,[F_1 \wedge F_2] \rightarrow C}\ \wedge_{Li} \qquad \frac{\Delta;\Gamma \rightarrow [F_i]}{\Delta;\Gamma \rightarrow [F_1 \vee F_2]}\ \vee_{Ri}$$

Structural Rules

$$\frac{\Delta,\Delta_1;\Gamma;\cdot \Rightarrow P_a \ \ldots\ \Delta,\Delta_k;\Gamma;\cdot \Rightarrow P_a}{\Delta;\Gamma,\Delta_1,\ldots,\Delta_k;\cdot \Rightarrow P_a}\ \mathtt{k}$$

$$\frac{\Delta;\Gamma,[N] \rightarrow P_a}{\Delta;N,\Gamma;\cdot \Rightarrow P_a}\ D_l \qquad \frac{F \supset G,\Delta;\Gamma,[F \supset G] \rightarrow P_a}{F \supset G,\Delta;\Gamma;\cdot \Rightarrow P_a}\ D_{lc} \qquad \frac{\Delta;\Gamma \rightarrow [P]}{\Delta;\Gamma;\cdot \Rightarrow P}\ D_r$$

$$\frac{\Delta;\Gamma;P \Rightarrow P_a}{\Delta;\Gamma,[P] \rightarrow P_a}\ R_l \qquad \frac{\Delta;\Gamma;\cdot \Rightarrow N}{\Delta;\Gamma \rightarrow [N]}\ R_r \qquad \frac{\Delta;\Gamma,N_a;\Theta \Rightarrow R}{\Delta;\Gamma;\Theta,N_a \Rightarrow R}\ \mathtt{store}$$

$$\frac{}{\Delta;[A_n] \rightarrow A_n}\ I_l \qquad \frac{}{\Delta;\Gamma \rightarrow [A_p]}\ I_r \text{ given } A_p \in \Delta \cup \Gamma \text{ and } \Gamma \subseteq \{A_p\}$$

Fig. B.1. Focused knotted system FFL^k_{ew}. Here P_a is positive or atomic, P is positive, N is negative, A_n a negative atom and A_p a positive atom.

$$\frac{\dfrac{\boxed{[L']};\Delta,F;E \vdash G}{\boxed{[L']};\Delta,F \,\&\, F';E \vdash G}\ \&_L}{\boxed{[L]};\Delta;E \vdash G}\ D_L$$

Note that F cannot be returned to the conclusion of D_L and, for that reason, D rules move formulas to Δ.