LOGIC, ALGEBRA, AND COMPUTER SCIENCE
BANACH CENTER PUBLICATIONS, VOLUME 46
INSTITUTE OF MATHEMATICS
POLISH ACADEMY OF SCIENCES
WARSZAWA 1999

STRONG NORMALIZATION PROOFS FOR CUT ELIMINATION IN GENTZEN'S SEQUENT CALCULI

ELIAS TAHHAN BITTAR

LLAIC1, Université D'Auvergne
BP 86, 63172 Aubière Cedex, France
E-mail: tahhan@llaic.u-clermont1.fr

Abstract. We define an equivalent variant LK_{sp} of the Gentzen sequent calculus LK. In LK_{sp} weakenings or contractions can be performed in *parallel.* This modification allows us to interpret a symmetrical system of mix elimination rules $\mathcal{E}_{LK_{sp}}$ by a finite rewriting system; the termination of this rewriting system can be machine checked. We give also a self-contained strong normalization proof by structural induction. We give another strong normalization proof by a strictly monotone subrecursive interpretation; this interpretation gives subrecursive bounds for the length of derivations. We give a strong normalization proof by applying orthogonal term rewriting results for a confluent restriction of the mix elimination system $\mathcal{E}_{LK_{sp}}$.

1. Introduction. The purpose of this paper is to give a presentation of several techniques of strong normalization proofs for sequent calculi like Gentzen's calculi LJ and LK. Gentzen gave in his Hauptsatz a system of rewriting rules as a cut elimination procedure; he proved that an innermost strategy of reductions, i.e. successive applications of rewriting rules to subderivations containing only one cut, leads to a cut free derivation in a finite number of steps. Therefore cut elimination procedures are *computations* leading to cut free derivations; a main aspect to understanding the computational content of sequent calculi is the definition of cut elimination systems satisfying confluence and strong normalization properties. We focus in this work on the strong normalization properties of cut elimination systems.

We give in section 2 a variant LK_{sp} of the calculus LK. In section 3 we give several cut elimination systems for LK and LK_{sp}. Strong normalization proofs for these cut elimination systems are investigated in section 4. We observe first that under additional confluence properties normalization by an innermost strategy implies strong normalization; this approach is applied in subsection 4.2 to orthogonal restrictions of the calculi

1991 *Mathematics Subject Classification*: Primary 03F05; Secondary 68Q42.

The paper is in final form and no version of it will be published elsewhere.

LK and LK_{sp}. We give in subsection 4.3 proofs for strong normalization by recursive path orderings; one of these strong normalization proofs is machine checkable. Gentzen's innermost normalization proof is performed by structural induction on a rank associated to derivations; in subsection 4.4 we define a rank of a derivation which allows us to prove strong normalization by structural induction. Finally in subsection 4.5 we give strong normalization proofs by interpretation of the inferences by monotone subrecursive functions on natural numbers; these interpretations give upper bounds for the number of steps of reductions required to normalize a derivation. We refer the reader to subsections 3.1 and 4.1 for detailed introductions and historical remarks on mix elimination systems for sequent calculi and strong normalization proofs for such systems.

2. The calculus LK_{sp}

2.1. Preliminaries. We consider a first order language $\mathcal{L}$ without equality with only the connectives: $\neg$, $\vee$, $\exists$. Formulae of $\mathcal{L}$ will be denoted by capital letters, A, B, etc. A formula of $\mathcal{L}$ without any occurrence of any connective is called an *atomic* formula. A *multiset* of formulae is a finite sequence of formulae modulo permutations. Multisets of formulae will be denoted by capital Greek letters, $\Sigma, \Delta, \ldots$. The concatenation of two multisets Σ and Δ will be denoted by Σ, Δ. The set of non-null natural numbers will be denoted by $\mathbb{N}^+$, natural numbers will be denoted by the letters $i, j, k, l, m, n, k_i, k_j \ldots$ The multiset of n copies of a formula A will be denoted by A^n; by definition, A^0 is an empty multiset. The *contraction* of an empty multiset A^0 is A^0 itself, the *contraction* of a non-empty multiset A^n (with $n \neq 0$) is the multiset A. We recall that the *degree* of a formula A, denoted by $|A|$ is defined by:

$$|A| = 1 \text{ if } A \text{ is atomic}, \qquad |\neg A| = |\exists x A| = 1 + |A|, \qquad |A \vee B| = 1 + \max\{|A|, |B|\}\,.$$

A *sequent* is a pair (Σ, Δ) of multisets of formulae, usually denoted by $\Sigma \vdash \Delta$, where Σ (resp. Δ) is called the *antecedent* (resp. *consequent*) of the sequent. The sequent $\vdash$ where both antecedent and consequent are empty multisets is called the *absurd* sequent.

An *inference* is a pair where the first element is a sequence of sequents called the premises of the inference and the second element is a sequent called the conclusion of the inference. We notice that the sequence of premises of an inference may be empty and, in this case, the conclusion is said to be an *axiom*. An inference is usually denoted by:

$$\frac{\text{sequence of premises of the inference}}{\text{conclusion of the inference}}\ \text{inference name,}$$

as in the following example: $\dfrac{\Sigma, B \vdash \Delta \quad \Sigma, C \vdash \Delta}{\Sigma, B \vee C \vdash \Delta}\ \vee_{\mathcal{L}}$.

2.2. LK inference rules. We recall Gentzen's calculus LK in order to fix terminology. The LK inference rules on sequents are defined as follows.

Axiom inference rule - For each atomic formula B of $\mathcal{L}$ we have the axiom inference (for which the set of premises is empty): $\dfrac{}{B \vdash B}\ \mathcal{A}x$; atomic formulae B of both sides are called formulae introduced by an axiom inference.

Structural inference rules -

The left and right weakening inference rules:

$$\frac{\Sigma \vdash \Delta}{\Sigma, A \vdash \Delta}\,\mathcal{W}_{\mathcal{L}} \qquad \text{and} \qquad \frac{\Sigma \vdash \Delta}{\Sigma \vdash A, \Delta}\,\mathcal{W}_{\mathcal{R}}\,.$$

The left and right contraction inference rules:

$$\frac{\Sigma, A, A \vdash \Delta}{\Sigma, A \vdash \Delta}\,\mathcal{C}_{\mathcal{L}} \qquad \text{and} \qquad \frac{\Sigma \vdash A, A, \Delta}{\Sigma \vdash A, \Delta}\,\mathcal{C}_{\mathcal{R}}\,.$$

Logical inference rules -

The left and right negation inference rules:

$$\frac{\Sigma \vdash A, \Delta}{\Sigma, \neg A \vdash \Delta}\,\neg_{\mathcal{L}} \qquad \text{and} \qquad \frac{\Sigma, A \vdash \Delta}{\Sigma \vdash \neg A, \Delta}\,\neg_{\mathcal{R}}\,;$$

any such formulae $\neg A$ are called formulae introduced by a negation inference.

The left and right disjunction inference rules:

$$\frac{\Sigma, B \vdash \Delta \quad \Sigma, C \vdash \Delta}{\Sigma, B \vee C \vdash \Delta}\,\vee_{\mathcal{L}}\,,$$

$$\frac{\Sigma \vdash B, \Delta}{\Sigma \vdash B \vee C, \Delta}\,\vee_{\mathcal{R}} \qquad \text{and} \qquad \frac{\Sigma \vdash B, \Delta}{\Sigma \vdash C \vee B, \Delta}\,\vee_{\mathcal{R}}\,;$$

any such formulae $B \vee C$ are called formulae introduced by a disjunction inference.

The left and right existential inference rules: If a variable y is free in a formula B, (and B in this case is denoted $B(y)$), and is not free in any formula of Σ or Δ, we say that y is an *eigenvariable* of the formula B in the sequent $\Sigma, B(y) \vdash \Delta$ and we define the *left existential inference* by:

$$\frac{\Sigma, B(y) \vdash \Delta}{\Sigma, \exists y B(y) \vdash \Delta}\,\exists_{\mathcal{L}}\,.$$

If t is any term of the language $\mathcal{L}$, if a variable y is free in a formula B, and if $B(y \leftarrow t)$ denotes the substitution of the variable y by the term t in the formula B, we define the *right existential inference* by:

$$\frac{\Sigma \vdash B(y \leftarrow t), \Delta}{\Sigma \vdash \exists y B(y), \Delta}\,\exists_{\mathcal{R}}\,.$$

Any such formulae $\exists y B(y)$ are called formulae introduced by an existential inference.

The *Capture-avoiding substitution* of a variable z by a term t in a formula B is defined recursively by:

- if A is an atomic formula then $A(z \leftarrow t)$ is a capture-avoiding substitution,
- if $B(z \leftarrow t)$ and $C(z \leftarrow t)$ are capture-avoiding substitutions then $(\neg B)(z \leftarrow t)$ and $(B \vee C)(z \leftarrow t)$ are capture-avoiding substitutions;
- and if $B(z \leftarrow t)$ is a capture avoiding substitution and x is a variable which does not occur in t then the substitution $(\exists x B)(z \leftarrow t)$ is capture-avoiding.

Capture-avoiding substitutions on left existential inferences satisfy:

Fact 2.1. *Given an eigenvariable y of a formula B in the sequent $\Sigma, B(y) \vdash \Delta$ and given a left existential inference:* $\dfrac{\Sigma, B(y) \vdash \Delta}{\Sigma, \exists y B(y) \vdash \Delta}\ \exists_{\mathcal{L}}$; *we may rename the eigenvariable y or perform a capture-avoiding substitution.*

Indeed, if a variable z does not occur in any formula of the inference then z is an eigenvariable of the formula $B(y \leftarrow z)$ in the sequent $\Sigma, B(y \leftarrow z) \vdash \Delta$ and we have the inference:

$$\frac{\Sigma, B(y \leftarrow z) \vdash \Delta}{\Sigma, \exists z B(y \leftarrow z) \vdash \Delta}\ \exists_{\mathcal{L}}\,;.$$

If z is a variable different from y and t is a term with no occurrence of the variable y, then the capture-avoiding substitution of the variable z by a term t given by:

$$\frac{\Sigma(z \leftarrow t), B(z \leftarrow t)(y) \vdash \Delta(z \leftarrow t)}{\Sigma(z \leftarrow t), \exists y B(z \leftarrow t)(y) \vdash \Delta(z \leftarrow t)}\ \exists_{\mathcal{L}}\,,$$

is also an inference.

Mix inference rules - A mix of a formula A is the following inference:

$$\frac{\Sigma, A^n \vdash \Delta \quad \Sigma' \vdash A^m, \Delta'}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\ mix_{|A|}\,;$$

where each of the numbers n and m may be equal to zero. Such a formula A is called the principal formula for the mix inference. The multisets of formulae A^n and A^m are called the eliminated multisets of formulae by the mix inference; if $n = m = 1$ the mix inference is also called a *cut* inference.

2.3. LK_{sp} inference rules. The calculus LK_{sp} is a variant of Gentzen's LK calculus with the same non-structural rules and with two new parallel structural inference rules which we now define.

Parallel structural inference rules -

the weakening inference rules: We add simultaneously one multiset in the antecedent and one multiset in the consequent:

$$\frac{\Sigma \vdash \Delta}{\Sigma, \Gamma \vdash \Delta, \Gamma'}\ \mathcal{W}\,;$$

where each of the multisets of formulae Γ and Γ' may be empty. If this is the case the weakening inference is said to be a *dummy weakening.*

the contraction inference rules: We contract simultaneously several multisets occurring in the antecedent and in the consequent:

$$\frac{\Sigma, A_1^{k_1}, \ldots, A_n^{k_n} \vdash \Delta, B_1^{l_1}, \ldots, B_m^{l_m}}{\Sigma, A_1^{\delta(k_1)}, \ldots, A_n^{\delta(k_n)} \vdash \Delta, B_1^{\delta(l_1)}, \ldots, B_m^{\delta(l_m)}}\ \mathcal{C}\,;$$

with $\delta(n) := \min\{1, n\}$.

2.4. LK and LK_{sp} proofs. For the sake of completeness we give in this section a formal definition of proofs.

The concatenation of two sequences of natural numbers p and q is denoted by $p{\cdot}q$. The empty sequence of natural numbers is denoted by Λ. Given three sequences of natural numbers p, p' and q such that the equality $q = p \cdot p'$ holds, the sequence p is said to be a *prefix* of the sequence q and is denoted by $p \leq_{pr} q$ and conversely the sequence q is said to be an *extension* of the sequence p. The prefix relation $\leq_{pr}$ over the set of sequences of natural numbers is a partial order.

Sequences of natural numbers are denoted by $< n_0, \ldots, n_k >$. A (non-empty) *tree* T is a set of finite sequences of natural numbers such that if a sequence p belongs to T then any prefix of the sequence p is also an element of T. If n is a natural number and p is a sequence of natural numbers then the sequence $p{\cdot} < n >$ is said to be a *successor* of p. The elements of a tree are said to be *nodes* or *positions*. The node Λ of a tree T is said to be the *root* of T, and the nodes of maximal length in a tree T are said to be the *leaves* of T. A *tree labeled* by elements of a set S is a mapping from a tree T to the set S.

Let K be either LK or LK_{sp}. A *K-deduction* $\mathcal{D}$ is a tree labeled by K-inferences, the domain of which is denoted by $T(\mathcal{D})$ and such that:

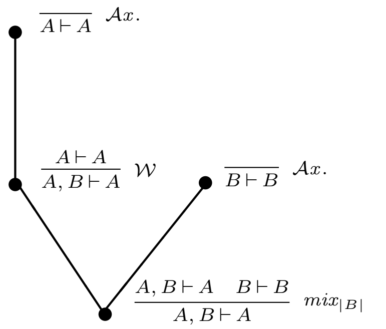
- if a node p is labeled by a K-inference:
$$\frac{\Sigma \vdash \Delta}{\Sigma' \vdash \Delta'} \ ,$$
then either the position p has no successor in $T(\mathcal{D})$ and then the sequent $\Sigma \vdash \Delta$ is said to be a *hypothesis of* $\mathcal{D}$, or the position p has one successor $p{\cdot} < 0 >$ in $T(\mathcal{D})$ labeled by an inference, the conclusion of which is $\Sigma \vdash \Delta$;
- If a node p is labeled by a K-inference:
$$\frac{\Sigma_0 \vdash \Delta_0 \quad \Sigma_1 \vdash \Delta_1}{\Sigma' \vdash \Delta'} \ ,$$
then p has in $T(\mathcal{D})$ at most two successors, respectively $p{\cdot} < 0 >$ and $p{\cdot} < 1 >$, labeled by inferences, the conclusion of which are, respectively, $\Sigma_0 \vdash \Delta_0$ and $\Sigma_1 \vdash \Delta_1$; denoting by ϵ a natural number among $\{0, 1\}$, if the sequence $p{\cdot} < \epsilon >$ is not in $T(\mathcal{D})$ then the sequent $\Sigma_\epsilon \vdash \Delta_\epsilon$ is said to be a *hypothesis of* $\mathcal{D}$.

The deduction tree of a K-deduction, $\mathcal{D}$, is the domain of $\mathcal{D}$; the root of $\mathcal{D}$ is labeled by a K-inference, the conclusion of which is said to be *the conclusion of* $\mathcal{D}$. A *K-proof* Π of a sequent $\Sigma \vdash \Delta$ is a K-deduction, the conclusion of which is the sequent $\Sigma \vdash \Delta$ with no hypothesis. The calculi LK and LK_{sp} satisfy:

FACT 2.2. *A sequent has an LK-proof if and only if it has an LK_{sp}-proof.*

Hence, the calculi LK and LK_{sp} are equivalent. Furthermore, in the absence of ambiguity we omit the mention of whether LK or LK_{sp} is being used.

If Π is a proof and we replace the labels of the proof tree, which are inferences, by their respective names we obtain the *proof name* of Π, this proof name is represented as a term; if we replace the labels of the proof tree of Π by their respective conclusions we obtain what Gentzen called *proof figures* which are the usual representation of proofs in sequent calculi. In the following example we give a proof and its corresponding proof figure and proof name.

Proof	**Proof figure and name**
 $\overline{A\vdash A}\ \mathcal{A}x.$ $\frac{A\vdash A}{A,B\vdash A}\ \mathcal{W}$ $\overline{B\vdash B}\ \mathcal{A}x.$ $\frac{A,B\vdash A\quad B\vdash B}{A,B\vdash A}\ mix_{\|B\|}$	**Proof figure** $\dfrac{\dfrac{\dfrac{}{A\vdash A}\,\mathcal{A}x.}{A,B\vdash A}\,\mathcal{W}\quad \dfrac{}{B\vdash B}\,\mathcal{A}x.}{A,B\vdash A}\,mix_{\|B\|}$ **Proof name** $mix_{\|B\|}(\mathcal{W}(\mathcal{A}x),\mathcal{A}x)$

We remark that in the calculi LK and LK_{sp} several proofs may have the same name; so we cannot recover a proof from its name.

A K-deduction $\mathcal{D}$ of a sequent $\Sigma\vdash\Delta$ is represented by the figure: $\begin{array}{c}\vdots\,\mathcal{D}\\ \Sigma\vdash\Delta\end{array}$.

The *subdeduction of a deduction* $\mathcal{D}$ *at position* p, denoted by $\mathcal{D}|_p$, is defined recursively by:

- the equality $\mathcal{D}_\Lambda := \mathcal{D}$ holds;
- and if $\mathcal{D}$ is a deduction given by:

$$\mathcal{D} = \frac{\begin{array}{c}\vdots\,\mathcal{D}_0\\ \Sigma\vdash\Delta\end{array}}{\Sigma'\vdash\Delta'} \qquad \text{or} \qquad \mathcal{D} = \frac{\begin{array}{c}\vdots\,\mathcal{D}_0\\ \Sigma_0\vdash\Delta_0\end{array}\quad \begin{array}{c}\vdots\,\mathcal{D}_1\\ \Sigma_1\vdash\Delta_1\end{array}}{\Sigma'\vdash\Delta'}\ ;$$

  then $\mathcal{D}|_{<0>\cdot p} := \mathcal{D}_0|_p$ or $\mathcal{D}|_{<0>\cdot p} := \mathcal{D}_0|_p$ and $\mathcal{D}|_{<1>\cdot p} := \mathcal{D}_1|_p$.

A subdeduction $\Pi|_p$ (at position p) of a proof Π is also said to be a *subproof at position* p of the proof Π; the subproofs $\Pi|_{<0>}$ (and possibly $\Pi|_{<1>}$) are said to be *immediate subproofs* of Π. A context of a position p in a proof Π is the subdeduction denoted by $\Pi|^p$ obtained by restricting the proof Π to the positions which do not extend the sequence p. For example, given the proof:

$$\Pi = \cfrac{\cfrac{\cfrac{}{A\vdash A}\,\mathcal{A}x.}{A,B\vdash A}\,\mathcal{W}\quad \cfrac{}{B\vdash B}\,\mathcal{A}x.}{A,B\vdash A}\,mix_{|B|}\ ;$$

the subproof at position $<0>$ of Π and the context of the position $<0>$ in Π are respectively:

$$\Pi|_{<0>} = \cfrac{\cfrac{}{A\vdash A}\,\mathcal{A}x.}{A,B\vdash A}\,\mathcal{W} \qquad \text{and} \qquad \Pi|^{<0>} = \cfrac{A,B\vdash A\quad \cfrac{}{B\vdash B}\,\mathcal{A}x.}{A,B\vdash A}\,mix_{|B|}\,.$$

We adopt the following terminology: a sequent *occurs* in an inference if it is a premise or a conclusion of the inference; a sequent *occurs* in a proof Π if it occurs in an inference labeling a node of the proof tree of Π, a formula *occurs* in a proof Π if it appears in a sequent occuring in Π and a variable *occurs* in a proof Π if it is a variable of a formula occuring in the proof Π. The substitution of a variable z by a term t in each formula of a proof Π is denoted $\Pi(z \leftarrow t)$; if each one of these formulae substitutions is capture-avoiding then the substitution $\Pi(z \leftarrow t)$ is also said to be capture-avoiding and satisfies the:

Lemma 2.1 (Capture-avoiding substitution lemma). *If* Π *is a proof and* $\Pi(z \leftarrow t)$ *is a capture-avoiding substitution then* $\Pi(z \leftarrow t)$ *is a proof; moreover, the proofs* Π *and* $\Pi(z \leftarrow t)$ *have the same name.*

Proof. By structural induction on the construction of the proof Π; the main point is:

Fact 2.3. *Given an eigenvariable* y *of a formula* B *in the sequent* $\Sigma, B(y) \vdash \Delta$ *and given a proof* Π *obtained as follows:*

$$\dfrac{\begin{array}{c}\vdots\ \Pi_1\\ \Sigma, B(y) \vdash \Delta\end{array}}{\Sigma, \exists y B(y) \vdash \Delta}\ \exists_{\mathcal{L}}\,;$$

we can rename in Π *the eigenvariable* y *or perform a capture-avoiding substitution in* Π.

Indeed, these facts hold by the two arguments given below.

If a variable z does not occur in the proof Π then z is an eigenvariable of the formula $B(y \leftarrow z)$ in the sequent $\Sigma, B(y \leftarrow z) \vdash \Delta$ and by inductive hypothesis the capture-avoiding substitution $\Pi_1(y \leftarrow z)$ is a proof, hence $\Pi(y \leftarrow z)$ is a proof given by:

$$\dfrac{\begin{array}{c}\vdots\ \Pi_1(y \leftarrow z)\\ \Sigma, B(y \leftarrow z) \vdash \Delta\end{array}}{\Sigma, \exists z B(y \leftarrow z) \vdash \Delta}\ \exists_{\mathcal{L}}\,;$$

moreover, since by inductive hypothesis the proofs Π_1 and $\Pi_1(y \leftarrow z)$ have the same name then the proofs Π and $\Pi(y \leftarrow z)$ have the same name.

Given a term t with no occurrence of the variable y and such that the substitution of a variable z different from the variable y by the term t in the proof Π_1 is capture-avoiding, then by inductive hypothesis $\Pi_1(z \leftarrow t)$ is a proof and therefore the capture avoiding substitution $\Pi(z \leftarrow t)$ is proof given by:

$$\dfrac{\begin{array}{c}\vdots\ \Pi_1(z \leftarrow t)\\ \Sigma(z \leftarrow t), B(z \leftarrow t)(y) \vdash \Delta(z \leftarrow t)\end{array}}{\Sigma(z \leftarrow t), \exists y B(z \leftarrow t)(y) \vdash \Delta(z \leftarrow t)}\ \exists_{\mathcal{L}}\,;$$

moreover, since by inductive hypothesis the proofs Π_1 and $\Pi_1(z \leftarrow t)$ have the same name, the proofs Π and $\Pi(z \leftarrow t)$ have the same name. ∎

Remark 2.1. The above lemma allows one to identify proofs up to bounded variable renaming; henceforth, we will assume this identification.

3. Mix elimination systems

3.1. Historical remarks. In his works Gentzen defined the sequent calculus in order to establish the consistency of logical theories. A logical theory is not consistent if every formula can be deduced from such theory. In particular the LK calculus is not consistent if and only if there is an LK proof of the absurd sequent $\vdash$. In fact if each premise of a non-mix inference is not the absurd sequent then the conclusion of the inference is not the absurd sequent. So, to demonstrate the consistency of LK it is sufficient to show that for each LK proof of a sequent $\Sigma \vdash \Delta$ there is an LK proof of this sequent which does not contain any use of a mix inference. For instance, Gentzen defined several proof transformation systems and for each one of those systems he chose a particular strategy of proof transformation and established that the application of this strategy leads to a mix free proof. These results are called normalization theorems, and the proof transformation systems which satisfy normalization property are called mix elimination systems.

In order to fix terminology for further discussions we distinguish two families of mix elimination systems as follows.

Non-symmetrical mix elimination systems: this family contains the mix elimination system defined by Gentzen in *"Investigation into logical deduction"* [Gen35].

Symmetrical mix elimination systems: this family contains the mix elimination system defined by Gentzen in *"New version of the consistency proof for elementary number theory"* [Gen38]. This family considers mix eliminations which usually called in the literature *cross reductions.*

Recently the question of the algorithmic meaning of mix elimination systems has been raised. As a contribution on this subject we investigate their strong normalization properties, (i.e. any strategy of mix elimination leads to a mix free proof); and we establish (sub)recursive upper bounds for the number of transformation steps needed to obtain mix free proofs. The present contribution is in the tradition of the following works:

- Dragalin gave in [Dra88] a very technical proof of a strong normalization theorem for the non-symmetrical mix elimination system defined in [Gen35]. This proof uses a structural inductive approach; Dragalin proved that if Π_1 and Π_2 are two LK-proofs which are strongly normalizing, then the proof $\Pi = mix(\Pi_1, \Pi_2)$ given by:

$$\dfrac{\begin{array}{c}\vdots\,\Pi_1\\ \Sigma, B^n \vdash \Delta\end{array}\quad\begin{array}{c}\vdots\,\Pi_2\\ \Sigma', \vdash B^m \Delta\end{array}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,mix_{|B|}\,,$$

  is also strongly normalizing.
- In 1990 we gave a constructive proof of a strong normalization theorem for the propositional fragment of the LK-calculus for an orthogonal and confluent restriction of a symmetrical mix elimination system. This proof uses a rewriting representation of mix eliminations and techniques of orthogonal rewriting systems theory; it is based on the fact that if Π_1 and Π_2 are two LK-proofs strongly normalizing – the unique normal forms of which are respectively denoted by $\downarrow(\Pi_1)$ and $\downarrow(\Pi_2)$

– and if a proof $mix(\downarrow(\Pi_1), \downarrow(\Pi_2))$ is also strongly normalizing, then the proof $mix(\Pi_1, \Pi_2)$ is also strongly normalizing. In [Tah92] we gave a new version of this result, using directly a lemma of O'Donnell:

> An orthogonal rewriting system is strongly normalizing if and only if every innermost reduction strategy terminates.

Thus the combination of Gentzen's normalization theorem which stands for any innermost strategy and O'Donnell's lemma implies the strong normalization theorem.

- In [CRS94] Cichon, Rusinowitch and Selhab gave infinite rewriting systems representing classical and intuitionistic sequent calculi and several linear calculi and proved, with recursive path orderings, strong normalization theorems for the linear calculi which they considered. In [CRS96] they defined an infinite rewriting system for a symmetrical mix elimination system, without any confluent restriction; the termination of this rewriting system is obtained through a recursive path ordering.

Another tradition for establishing a strong normalization theorem for mix elimination systems has been developed by Danos &Joinet & Schellinx and by Herbelin:

- In [Joi93] Joinet defined the calculus LKT, an equivalent restriction of the LK-calculus; the calculus LKT can be interpreted in a linear logic calculus satisfying strong normalization; this work has been continued in [DJS95]. Later in [Her95] Herbelin defined an extension $\bar{\lambda}\mu$ of Parigot's $\lambda\mu$ calculus [Par92] and extended the Curry-Howard isomorphism to an isomorphism between the sequent calculus LKT and $\bar{\lambda}\mu$ and gave a strong normalization proof for $\bar{\lambda}\mu$-terms. In the same work Herbelin also defined a $\bar{\lambda}$ calculus and an equivalent restriction LJT of the intuitionistic calculus LJ and established an extension of the Curry-Howard isomorphism to an isomorphism between LJT and $\bar{\lambda}$. Recently, in [DP96] R. Dyckhoff and L. Pinto gave a strong normalization proof by rewriting techniques of the calculus MJ, a reformulation of Herbelin's calculus LJT.

3.2. Illustrative example of a mix elimination rule. Given two proofs Π and $\hat{\Pi}'$ such that the conclusion of the proof $\hat{\Pi}'$ coincides with the conclusion of the subproof $\Pi|_p$ of Π at position p; a *replacement* in a proof Π of its subproof $\Pi|_p$ at position p by the proof $\hat{\Pi}'$ is a proof Π' such that the subproof $\Pi'|_p$ of Π' at position p is the proof $\hat{\Pi}'$ and the contexts of the position p in the proof Π and Π' are equal. A *proof transformation* of a proof Π into a proof Π' is a replacement in a proof Π of a subproof $\Pi|_p$ at a position p by any proof $\hat{\Pi}'$ of the conclusion of $\Pi|_p$. A finite or infinite sequence of proofs $P =< \Pi_0, \Pi_1, \ldots, \Pi_n \ldots >$ is said to be a *sequence of proof transformations* of the proof Π_0 if each proof Π_i in the sequence P admits a proof transformation into the proof Π_{i+1}. A set of proof transformation rules $\mathcal{E}$ is a *mix elimination system* if for any proof Π there is a sequence of proof transformations $< \Pi, \Pi_1, \ldots, \Pi_n >$ following proof transformation rules in $\mathcal{E}$ and such that there is no mix inference in the proof Π_n. A proof transformation rule belonging to a mix elimination system is also said to be a *mix elimination rule.* A proof transformation following a mix elimination rule is said to be a *mix elimination.* A mix elimination system $\mathcal{E}$ satisfies the *strong normalization property* if there is no infinite sequence of proof transformations following mix elimination rules belonging to $\mathcal{E}$. We define in this section a symmetrical mix elimination system (denoted by $\mathcal{E}_{LK_{sp}}$) satisfying the strong normalization property; one of the mix elimination rules of $\mathcal{E}_{LK_{sp}}$ is given in the next example.

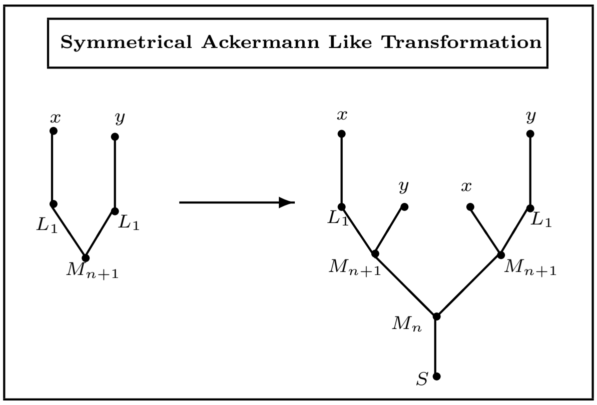
Example 3.1. If a proof Π given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(\neg B)^n \vdash B,\Delta}}{\Sigma,(\neg B)^{n+1}\vdash\Delta}\,\neg_{\mathcal{L}} \qquad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma',B\vdash(\neg B)^m,\Delta'}}{\Sigma'\vdash(\neg B)^{m+1},\Delta'}\,\neg_{\mathcal{R}}}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\; mix_{|\neg B|}\,,$$

occurs as a subproof of a proof $\hat{\Pi}$, then the proof Π can be replaced in the proof $\hat{\Pi}$ by the proof Π' given by:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(\neg B)^n \vdash B,\Delta}}{\Sigma,(\neg B)^{n+1}\vdash\Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma',B\vdash(\neg B)^m,\Delta'}}{\Sigma,\Sigma',B\vdash\Delta,\Delta'}\,mix_{|\neg B|} \qquad \dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(\neg B)^n\vdash B,\Delta} \quad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma',B\vdash(\neg B)^m,\Delta'}}{\Sigma'\vdash(\neg B)^{m+1},\Delta'}}{\Sigma,\Sigma'\vdash B,\Delta,\Delta'}\,mix_{|\neg B|}}{\Sigma,\Sigma',\Sigma,\Sigma'\vdash\Delta,\Delta',\Delta,\Delta'}\,mix_{|B|}}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\,\mathcal{C}\,.$$

We observe that this proof transformation has the following labeled tree transformation scheme:



In this scheme the label M_n stands for a mix principal formula which has degree n, the label L_1 stands for an unary logical rule, the label S stands for a structural rule and labels x and y stand for proof variables (i.e. variables which can be replaced by a proof).

This tree transformation can also be expressed as the term rewriting rule:

$$M_{n+1}\left(L_1(x),L_1(y)\right) \longrightarrow S\left(M_n\left(M_{n+1}\left(L_1(x),y\right),M_{n+1}\left(x,L_1(y)\right)\right)\right)$$

where a term rewriting rule is a directed equation; which means that substitution in equations are always made in the direction of the arrow. In fact, in order to establish strong normalization properties of a mix elimination system, we could skip the interpretation of these mix elimination systems by rewriting systems and give immediately an interpretation, $\mathcal{I} : LK_{sp}$ proofs $\longrightarrow \mathbb{N}^+$, of LK_{sp} proofs such that if a proof u is obtained from a proof t by a mix elimination then $\mathcal{I}(t) > \mathcal{I}(u)$. Roughly speaking:

$$\text{If} \quad t \xrightarrow{\text{mix elim.}} u \quad \text{then} \quad \mathcal{I}(t) > \mathcal{I}(u)\,;$$

such an interpretation is presented in section 4.5. Nevertheless we define in this section a finite rewriting system (denoted by $\mathcal{R}'_{LK_{sp}}$) which interprets the symmetrical mix elimination system $\mathcal{E}_{LK_{sp}}$ and such that termination (i.e. the strong normalization property) of $\mathcal{R}'_{LK_{sp}}$ can be machine checked.

3.3. Interpretation of LK and LK_{sp} proofs. There are in the literature several proof-term calculi which represent sequent calculi such that each proof can be recovered from the term which represents it, see for instance, in [Gal91], [Pfe94], [Pin93] or [Tah92]. These proof-term calculi are useful in implementing sequent calculi in logical frameworks and in logic programming (cf. [Pfe94], [Pin93]), but actually, when we want to deal with termination properties it is sufficient to deal with proof name transformations as is done in [CRS94]; furthermore it is only relevant to distinguish the inference rules according to a classification by types of inferences which we introduce below:

> Axiom inference rules, structural inference rules, unary or binary logical inference rules and mix inference rule of degree $k \in \mathbb{N}^+$, where the degree of a mix is the degree of its principal formula.

Thus we associate to each proof a ground term on the signature:

$$\mathcal{F} := \{\alpha, S, L_1, L_2\} \cup \{M_n \mid n \in \mathbb{N}^+\}\}\,;$$

where the symbol α has arity zero, the symbols S and L_1 have arity one, and L_2 and each M_n have arity two. Ground terms on $\mathcal{F}$ are terms built with the symbols of $\mathcal{F}$ without any variable; they are defined recursively by:

$$\text{Ground terms on } \mathcal{F}\text{:} \qquad t ::= \alpha \mid S(t) \mid L_1(t) \mid L_2(t,t) \mid \{M_n(t,t)\}_{n \in \mathbb{N}^+}\,.$$

If $\mathcal{V}$ is a set of variables then the set of terms on $\mathcal{F}$ is defined by:

$$\text{Terms on } \mathcal{F}\text{:} \qquad t ::= x \in \mathcal{V} \mid \alpha \mid S(t) \mid L_1(t) \mid L_2(t,t) \mid \{M_n(t,t)\}_{n \in \mathbb{N}^+}\,.$$

Denoting by $|A|$ the degree of a formula A, by Π_1, Π_2 two LK_{sp}-proof names, we associate to each LK_{sp}-proof name Π a ground term on $\mathcal{F}$, said to be a *type* of Π in $\mathcal{F}$, this is done through a mapping τ defined recursively by:

$\tau(\mathcal{A}x)$	$=$	α	$\tau(\exists_{\mathcal{L}}(\Pi_1))$	$=$	$L_1(\tau(\Pi_1))$
$\tau(\mathcal{W}(\Pi_1))$	$=$	$S(\tau(\Pi_1))$	$\tau(\exists_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau(\Pi_1))$
$\tau(\mathcal{C}(\Pi_1))$	$=$	$S(\tau(\Pi_1))$	$\tau(\vee_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau(\Pi_1))$
$\tau(\neg_{\mathcal{L}}(\Pi_1))$	$=$	$L_1(\tau(\Pi_1))$	$\tau(\vee_{\mathcal{L}}(\Pi_1, \Pi_2))$	$=$	$L_2(\tau(\Pi_1), \tau(\Pi_2))$
$\tau(\neg_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau(\Pi_1))$	$\tau(mix_{\lvert A\rvert}(\Pi_1, \Pi_2))$	$=$	$M_{\lvert A\rvert}(\tau(\Pi_1), \tau(\Pi_2))$

Convention: We assume that the type associated to a proof is the type of its name, so if Π is an LK_{sp}-proof and $\nu(\Pi)$ is the name of Π then: $\tau(\Pi) := \tau(\nu(\Pi))$.

Now we associate to each LK-proof name Π a ground term on $\mathcal{F}$, said to be the *type* of Π in $\mathcal{F}$, this is done through a recursively defined mapping τ'. Moreover, we choose to impose that the mapping τ' keeps the same interpretation as τ for the non-structural inference rules (which are common to LK and LK_{sp}) so that a convenient recursive definition of τ' is:

$\tau'(\mathcal{A}x)$	$=$	α	$\tau'(\neg_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau'(\Pi_1))$
$\tau'(\mathcal{W}_{\mathcal{L}}(\Pi_1))$	$=$	$S(\tau'(\Pi_1))$	$\tau'(\exists_{\mathcal{L}}(\Pi_1))$	$=$	$L_1(\tau'(\Pi_1))$
$\tau'(\mathcal{W}_{\mathcal{R}}(\Pi_1))$	$=$	$S(\tau'(\Pi_1))$	$\tau'(\exists_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau'(\Pi_1))$
$\tau'(\mathcal{C}_{\mathcal{L}}(\Pi_1))$	$=$	$S(\tau'(\Pi_1))$	$\tau'(\vee_{\mathcal{R}}(\Pi_1))$	$=$	$L_1(\tau'(\Pi_1))$
$\tau'(\mathcal{C}_{\mathcal{R}}(\Pi_1))$	$=$	$S(\tau'(\Pi_1))$	$\tau'(\vee_{\mathcal{L}}(\Pi_1, \Pi_2))$	$=$	$L_2(\tau'(\Pi_1), \tau'(\Pi_2))$
$\tau'(\neg_{\mathcal{L}}(\Pi_1))$	$=$	$L_1(\tau'(\Pi_1))$	$\tau'(mix_{\lvert A\rvert}(\Pi_1, \Pi_2))$	$=$	$M_{\lvert A\rvert}(\tau'(\Pi_1), \tau'(\Pi_2))$

Convention: We assume that the type associated to a proof is the type of its name, so if Π is an LK_{sp}-proof and $\nu(\Pi)$ is the name of Π then: $\tau'(\Pi) := \tau'(\nu(\Pi))$.

3.4. Mix elimination and rewriting systems

Interpretation of mix elimination as rewrite rules. In the illustrative example 3.1 of mix elimination rules, the subproof Π of the proof $\hat{\Pi}$ has the interpretation:

$$\tau(\Pi) = M_{n+1}\left(L_1(\tau(\Pi_1)), L_1(\tau(\Pi_2))\right)\,,$$

which is a subterm of $\tau(\hat{\Pi})$. Once the subproof Π is replaced by the subproof Π' in the proof $\hat{\Pi}$ we obtain a proof the interpretation of which is given by replacing a subterm $\tau(\Pi)$ of the term $\tau(\hat{\Pi})$ by the subterm:

$$\tau(\Pi') = S\left(M_n\left(M_{n+1}\left(L_1(\tau(\Pi_1)), \tau(\Pi_2)\right), M_{n+1}\left(\tau(\Pi_1), L_1(\tau(\Pi_2))\right)\right)\right)\,.$$

This replacement is said to be a *reduction* of the term $\tau(\hat{\Pi})$ following the rewrite rule:

$$M_{n+1}\left(L_1(x), L_1(y)\right) \longrightarrow S\left(M_n\left(M_{n+1}\left(L_1(x), y\right), M_{n+1}\left(x, L_1(y)\right)\right)\right)\,.$$

So we say that the illustrative mix elimination rule interpretations in the algebra of terms over $\mathcal{F}$ follows the above rewrite rule.

We shall define a set of mix eliminations $\mathcal{E}_{LK_{sp}}$ and associate to them a set of rewrite rules, or rewrite system, $\mathcal{R}_{LK_{sp}}$ such that if $\hat{\Pi}'$ is obtained from $\hat{\Pi}$ by a mix elimination then $\tau(\hat{\Pi}')$ is obtained from $\tau(\hat{\Pi})$ by a reduction following the associated rewrite rule. Hence we say that the interpretation of the mix elimination system $\mathcal{E}_{LK_{sp}}$ in the algebra $\mathcal{F}$ *follows* the rewrite system $\mathcal{R}_{LK_{sp}}$. Therefore, a sequence of proofs obtained by successive mix eliminations, $\Pi \longrightarrow \Pi_2 \longrightarrow \cdots \longrightarrow \Pi_m$, has an interpretation as a sequence of ground terms, $\tau(\Pi) \longrightarrow \tau(\Pi_2) \longrightarrow \cdots \longrightarrow \tau(\Pi_m)$, obtained by successive reductions following rewrite rules associated to the mix eliminations. Such a sequence is said to be a *derivation* of the ground term $\tau(\Pi)$. Indeed, in order to prove strong normalization of the calculus LK_{sp} it is enough to give a mix elimination system such that the associated rewrite system $\mathcal{R}_{LK_{sp}}$ terminates.

The system $\mathcal{R}_{LK_{sp}}$. We define a mix elimination system denoted $\mathcal{E}_{LK_{sp}}$, the interpretation of which by τ satisfies the rewriting system of terms on $\mathcal{F}$, denoted by $\mathcal{R}_{LK_{sp}}$ and defined by the following rules:

$$\begin{array}{rcl}
M_n(x, \alpha) & \longrightarrow & S(\alpha)\,, \\
M_n(\alpha, y) & \longrightarrow & S(\alpha)\,, \\
M_n\left(L_1(x), y\right) & \longrightarrow & L_1\left(M_n\left(x, y\right)\right)\,,
\end{array}$$

$$\begin{aligned}
M_n\left(x, L_1(y)\right) &\longrightarrow L_1\left(M_n\left(x, y\right)\right)\ , \\
M_n\left(S(x), y\right) &\longrightarrow S\left(M_n\left(x, y\right)\right)\ , \\
M_n\left(x, S(y)\right) &\longrightarrow S\left(M_n\left(x, y\right)\right)\ , \\
M_n\left(L_2(x, y), z\right) &\longrightarrow L_2\left(M_n(x, z), M_n(y, z)\right)\ , \\
M_n\left(x, L_2(y, z)\right) &\longrightarrow L_2\left(M_n(x, y), M_n(x, z)\right)\ , \\
M_{n+1}\left(L_1(x), L_1(y)\right) &\longrightarrow S\left(M_n\left(M_{n+1}\left(L_1(x), y\right), M_{n+1}\left(x, L_1(y)\right)\right)\right)\ , \\
M_{n+1}\left(L_1(x), L_1(y)\right) &\longrightarrow S\left(M_n\left(M_{n+1}\left(x, L_1(y)\right), M_{n+1}\left(L_1(x), y\right)\right)\right)\ , \\
M_{n+1}\left(L_2(x, y), L_1(z)\right) &\longrightarrow S\left(M_n\left(M_{n+1}\left(x, L_1(z)\right), M_{n+1}\left(L_2(x, y), z\right)\right)\right)\ , \\
\text{and}\quad M_{n+1}\left(L_2(x, y), L_1(z)\right) &\longrightarrow S\left(M_n\left(M_{n+1}\left(y, L_1(z)\right), M_{n+1}\left(L_2(x, y), z\right)\right)\right)\ .
\end{aligned}$$

We notice that there are derivations, of ground terms on $\mathcal{F}$ by the rewrite system $\mathcal{R}_{LK_{sp}}$, which are not interpretations of successive mix eliminations in $\mathcal{E}_{LK_{sp}}$.

Since there are infinitely-many degrees, the system $\mathcal{R}_{LK_{sp}}$ is itself infinite. To prove its termination we can use each one of the two methods presented below which allows us to reduce the problem to finite rewriting systems. One of these methods is based on a compactness argument, and the other, called the *internalizing method* works by replacing the infinitely many binary symbols M_n by a new unique (internal to the signature) ternary symbol $M(h, x, y)$ where h is a natural integer term, while x and y are proof terms.

Compactness method : the systems $\mathcal{R}^k_{LKsp}$. The signature $\mathcal{F}$ is a union of finite signatures $\mathcal{F}_k$ such that the restriction $\mathcal{R}^k_{LKsp}$ of $\mathcal{R}_{LK_{sp}}$ to terms on $\mathcal{F}_k$ is finite and such that if t is a ground term on $\mathcal{F}$ and σ is any derivation of t by the rewrite system $\mathcal{R}_{LK_{sp}}$, then there is a natural number k such that t is a ground term on $\mathcal{F}_k$ and σ is a derivation of t by the rewrite system $\mathcal{R}^k_{LK_{sp}}$. The method goes as follows:

The degree of the principal formula of a mix is also called the *degree of the mix inference.* We define the *mix degree of a proof* as the maximum of degrees of mix inferences which appear in the proof. Each mix elimination of $\mathcal{E}_{LK_{sp}}$ transforms a proof Π into a proof Π' such that the mix degree of Π' is less than or equal to the mix degree of Π; thus if Π is a proof of degree less than or equal to the natural number $k \in \mathbb{N}^+$ and if the sequence of proofs, $\Pi \longrightarrow \Pi_2 \longrightarrow \cdots \longrightarrow \Pi_m$, is obtained by successive mix eliminations, then the sequence of respective interpretations, $\tau(\Pi) \longrightarrow \tau(\Pi_2) \longrightarrow \cdots \longrightarrow \tau(\Pi_m)$, is obtained by following the finite rewrite system $\mathcal{R}^k_{LK_{sp}}$ which is the restriction of the rewrite system $\mathcal{R}^k_{LK_{sp}}$ to terms on the signature $\mathcal{F}_k$ defined by:

$$\mathcal{F}_k := \{\alpha, S, L_1, L_2\} \cup \{M_n \mid n \in \{1 \ldots, k\}\}\,.$$

Internalizing method : the system $\mathcal{R}'_{LK_{sp}}$. We consider a sorted signature with two sorts, the natural number sort and the proof sort, this signature $\mathcal{F}'$ is defined by:

$$\mathcal{F}' := \{1, s, \alpha, S, L_1, L_2, M\}\,;$$

such that ground terms on $\mathcal{F}'$ are defined by:

$$\text{Natural Numbers Ground terms:}\qquad n ::= 1 \mid s(n)\,,$$

$$\text{Proof Ground terms on } \mathcal{F}'\text{:}\qquad t ::= \alpha \mid S(t) \mid L_1(t) \mid L_2(t, t) \mid M(n, t, t)\,;$$

and if $\mathcal{V}_\pi$ is a set of proof variables and $\mathcal{V}_\nu$ is a set of natural number variables then the proof terms on $\mathcal{F}'$ are defined by:

$$\text{Natural Number terms:} \qquad n ::= h \in \mathcal{V}_\nu \mid 1 \mid s(n)\,,$$

$$\text{Proof terms on } \mathcal{F}'\text{:} \qquad t ::= x \in \mathcal{V}_\pi \mid \alpha \mid S(t) \mid L_1(t) \mid L_2(t,t) \mid M(n,t,t)\,.$$

We associate a proof ground term on $\mathcal{F}'$ to each ground term on $\mathcal{F}$ in two steps.

1. We map each non-null natural number to a natural number term by the application $\rrbracket \cdot \llbracket$ defined by:

$$\rrbracket 1 \llbracket := 1 \quad \text{and} \quad \rrbracket n+1 \llbracket := s(\rrbracket n \llbracket)\,.$$

2. We define recursively a mapping τ'' from ground terms on $\mathcal{F}$ to proof ground terms on $\mathcal{F}'$ by:

$$\begin{array}{rcl} \tau''(\alpha) & := & \alpha\,, \\ \tau''(S(x)) & := & S(\tau''(x))\,, \\ \tau''(L_1(x)) & := & L_1(\tau''(x))\,, \\ \tau''(L_2(x,y)) & := & L_2(\tau''(x),\tau''(y))\,, \\ \text{and} \quad \tau''(M_n(x,y)) & := & M(\rrbracket n \llbracket, \tau''(x), \tau''(y))\,. \end{array}$$

Under the assumptions of the illustrative example 3.1 of mix elimination and denoting by $\tau'' \circ \tau$ the composition of the mapping τ'' with the mapping τ and denoting for the sake of readability: $\Pi''_1 := \tau'' \circ \tau(\Pi_1)$ and $\Pi''_2 := \tau'' \circ \tau(\Pi_2)$; the subproof Π of the proof $\hat{\Pi}$ has the interpretation:

$$\tau'' \circ \tau(\Pi) = M\left(\rrbracket n+1 \llbracket, L_1(\Pi''_1), L_1(\Pi''_2)\right)\ ,$$

which is a subterm of $\tau'' \circ \tau(\hat{\Pi})$. Once the subproof Π is replaced by the subproof Π' in the proof $\hat{\Pi}$ we obtain a proof $\hat{\Pi}'$. The interpretation of $\hat{\Pi}'$ by $\tau'' \circ \tau$ gives a term that is the result of replacing the subterm $\tau'' \circ \tau(\Pi)$ of the term $\tau'' \circ \tau(\hat{\Pi})$ by the subterm:

$$\tau'' \circ \tau(\Pi') = S\left(M\left(\rrbracket n \llbracket, M\left(\rrbracket n+1 \llbracket, L_1(\Pi''_1), \Pi''_2\right), M\left(\rrbracket n+1 \llbracket, \Pi''_1, L_1(\Pi''_2)\right)\right)\right)\ .$$

This replacement is a reduction of the term $\tau(\hat{\Pi})$ following the rewrite rule:

$$M\left(s(h), L_1(x), L_1(y)\right) \longrightarrow S\left(M\left(h, M\left(s(h), L_1(x), y\right), M\left(s(h), x, L_1(y)\right)\right)\right)\ ;$$

where the symbol h is a natural number variable, and the symbols x and y are proof variables. So we say that the illustrative mix elimination rule interpretations in the algebra of terms $\mathcal{F}'$ follow the former rewrite rule.

Actually, the interpretation by $\tau'' \circ \tau$ of the mix elimination system $\mathcal{E}_{LK_{sp}}$ follows the rewriting system on proof terms on $\mathcal{F}'$ denoted by $\mathcal{R}'_{LK_{sp}}$ and defined by:

$$\begin{array}{rcl} M(h,x,\alpha) & \longrightarrow & S(\alpha)\,, \\ M(h,\alpha,y) & \longrightarrow & S(\alpha)\,, \\ M\left(h, L_1(x), y\right) & \longrightarrow & L_1\left(M\left(h,x,y\right)\right)\ , \\ M\left(h, x, L_1(y)\right) & \longrightarrow & L_1\left(M\left(h,x,y\right)\right)\ , \\ M\left(h, S(x), y\right) & \longrightarrow & S\left(M\left(h,x,y\right)\right)\ , \\ M\left(h, x, S(y)\right) & \longrightarrow & S\left(M\left(h,x,y\right)\right)\ , \\ M\left(h, L_2(x,y), z\right) & \longrightarrow & L_2\left(M(h,x,z), M(h,y,z)\right)\ , \end{array}$$

$$\begin{aligned}
M\left(h,x,L_2(y,z)\right) &\longrightarrow L_2\left(M(h,x,y),M(h,x,z)\right)\ ,\\
M\left(s(h),L_1(x),L_1(y)\right) &\longrightarrow S\left(M\left(h,M\left(s(h),L_1(x),y\right),M\left(s(h),x,L_1(y)\right)\right)\right)\ ,\\
M\left(s(h),L_1(x),L_1(y)\right) &\longrightarrow S\left(M\left(h,M\left(s(h),x,L_1(y)\right),M\left(s(h),L_1(x),y\right)\right)\right)\ ,\\
M\left(s(h),L_2(x,y),L_1(z)\right) &\longrightarrow S\left(M\left(h,M\left(s(h),x,L_1(z)\right),M\left(s(h),L_2(x,y),z\right)\right)\right)\ ,\\
\text{and}\quad M\left(s(h),L_2(x,y),L_1(z)\right) &\longrightarrow S\left(M\left(h,M\left(s(h),y,L_1(z)\right),M\left(s(h),L_2(x,y),z\right)\right)\right)\ ;
\end{aligned}$$

where the symbol h is a natural number variable, and the symbols x, y and z are proof variables.

3.5. The symmetrical mix elimination system $\mathcal{E}_{LK_{sp}}$. In this section we propose a mix elimination system $\mathcal{E}_{LK_{sp}}$ such that its interpretation in the algebra of terms on $\mathcal{F}$ follows the rewriting system $\mathcal{R}_{LK_{sp}}$ and hence its interpretation in the algebra of terms on $\mathcal{F}'$ follows the rewriting system $\mathcal{R}'_{LK_{sp}}$. This mix elimination system $\mathcal{E}_{LK_{sp}}$ is based on a mix elimination system proposed in [Pab90] and is in the tradition of those studied in [Gen38], [Gir87], [GLT89], [Tah92], [Gal93] and [CRS96]. We show in a later section that the set of transformations given in this section is *exhaustive*, which means that each mix inference with non-mix inference premises occurring in a proof matches at least one left hand side of a mix elimination rule. We point out also in this section that a mix inference can match more than one rule and the well known fact that a proof may have several normal forms.

Active and passive premise inferences. Following Gentzen the left premise inference in a proof interpreted by $M_n(\alpha,y)$, $M_n(L(x),y)$ or $M_n(L(x_1,x_2),y)$ is said to be an *active premise inference* if the formula introduced by the axiom inference α or by the logical inference L is eliminated by the mix inference interpreted by M_n; otherwise the left premise inference is said to be a *passive premise inference.* Active and passive right premise inferences are defined in the same way. For example in the proof:

$$\dfrac{\dfrac{\begin{matrix}\vdots\ \Pi_1\\ \Sigma,(\neg B)^n\vdash B,\Delta\end{matrix}}{\Sigma,(\neg B)^n,\neg B\vdash\Delta}\,\neg_{\mathcal{L}}\qquad\dfrac{\begin{matrix}\vdots\ \Pi_2\\ \Sigma',B\vdash(\neg B)^m,\Delta'\end{matrix}}{\Sigma'\vdash(\neg B)^{m+1},\Delta'}\,\neg_{\mathcal{R}}}{\Sigma,\neg B,\Sigma'\vdash\Delta,\Delta'}\,mix_{|\neg B|}\ ,$$

the left premise inference $\neg_{\mathcal{L}}$ is passive and the right premise inference $\neg_{\mathcal{R}}$ is active.

In this section we decorate interpretations of axiom and logical inferences with asterisks indicating when mix premise inferences are active or passive. For instance, the decorated interpretation in $\mathcal{F}$ of the above example is: $M_{n+1}(L_1(\tau(\Pi_1)),L_1^*(\tau(\Pi_2)))$. We remark that this decoration is external since we cannot infer the active or passive character of a mix inference premise from the proof name.

Remark 3.1. Since the introduced formulae of axiom inferences are atomic, any mix inference occuring in a LK_{sp}-proof does not have an active axiom inference premise and an active logical inference premise.

Exhaustivity of the mix elimination system. As usual, the mix elimination rules belonging to $\mathcal{E}_{LK_{sp}}$ are classified as follows.

Immediate reductions - The mix inference can be substituted by non-mix inferences; this is the case if each one of the mix premises are active axioms or one of the mix premises is a passive axiom. The decorated interpretation of the replaced proof in these cases matches either $M_1(\alpha^*, \alpha^*)$ or $M_n(\alpha, y)$ or $M_n(x, \alpha)$.

Commutative reductions - The mix inference can be permuted with one of the premise inferences; this is the case if: either one of the mix premises is a structural inference, the decorated interpretation of the replaced proof in these cases matches either $M_n(S(x), y)$ or $M_n(x, S(y))$; or one of the mix premises is a passive logical inference, the decorated interpretation of the replaced proof in these cases matches either $M_n(L_1(x), y)$ or $M_n(x, L_1(y))$ or $M_n(L_2(x_1, x_2), y)$ or $M_n(x, L_2(y_1, y_2))$.

Symmetrical essential reductions - Both mix premises are logical active inferences. The decorated interpretation of the replaced proof in these cases matches either $M_n(L_1^*(x), L_1^*(y))$ or $M_n(L_2^*(x_1, x_2), L_1^*(y))$. We point out that the symmetrical nature of the sequent calculus LK_{sp} is implemented in the mix elimination system $\mathcal{E}_{LK_{sp}}$ by the fact that the performance of a mix elimination of a mix inference with one active logical premise requires the other premise to also be active; whence the symmetrical qualification of $\mathcal{E}_{LK_{sp}}$.

Any non-mix free LK_{sp}-proof can be reduced by a mix elimination rule belonging to $\mathcal{E}_{LK_{sp}}$; this is a consequence of:

Fact 3.1. *The mix elimination system $\mathcal{E}_{LK_{sp}}$ is exhaustive.*

Proof. Any mix inference employed in a proof, and such that neither of its premises are mix inferences, matches at least one of the mix elimination rules belonging to $\mathcal{E}_{LK_{sp}}$; indeed, either:

- both mix premises are active logical inferences and then we proceed by symmetrical essential reduction,
- or one of the mix premises is a structural inference or a passive logical inference and then we proceed by commutative reduction,
- or one of the mix premises is an axiom inference and then we proceed by immediate reduction; indeed, if the previous two alternatives do not hold, then by remark 3.1 either both premises of the mix inference are active axiom inferences or at least one of the premises is a passive axiom inference. ■

In the next subsections we give the mix eliminations which constitute $\mathcal{E}_{LK_{sp}}$ and their interpretations in $\mathcal{F}$ by rewriting rules belonging to $\mathcal{R}_{LK_{sp}}$.

Immediate reductions

Active Axiom premise inferences - A proof the decorated name of which matches the term $M_1(\alpha^*, \alpha^*)$ and given by:

$$\dfrac{\dfrac{}{A \vdash A}\,\mathcal{A}x. \qquad \dfrac{}{A \vdash A}\,\mathcal{A}x.}{A \vdash A}\,mix_{|A|}\,, \qquad \text{is replaced by} \qquad \dfrac{\dfrac{}{A \vdash A}\,\mathcal{A}x.}{A \vdash A}\,\mathcal{W};$$

where the weakening is a dummy weakening.

Cases $M_n(\alpha, y)$ and $M_n(x, \alpha)$ - A proof the decorated name of which matches the term $M_n(\alpha, y)$ and given by:

$$\dfrac{\dfrac{}{A\vdash A}\,\mathcal{A}x \qquad \overset{\vdots\ \Pi_1}{\Sigma\vdash(C)^m,\Delta}}{\Sigma, A\vdash A,\Delta}\,mix_{|C|}\,, \qquad \text{is replaced by} \qquad \dfrac{\dfrac{}{A\vdash A}\,\mathcal{A}x}{\Sigma, A\vdash A,\Delta}\,\mathcal{W}\,;$$

and a proof the decorated name of which matches the term $M_n(x, \alpha)$ and given by:

$$\dfrac{\overset{\vdots\ \Pi_1}{\Sigma, C^n\vdash\Delta} \qquad \dfrac{}{A\vdash A}\,\mathcal{A}x}{\Sigma, A\vdash A,\Delta}\,mix_{|C|}\,, \qquad \text{is replaced by} \qquad \dfrac{\dfrac{}{A\vdash A}\,\mathcal{A}x}{\Sigma, A\vdash A,\Delta}\,\mathcal{W}\,.$$

Thus the interpretation of each immediate reduction follows one of the rewriting rules: $M_n(\alpha, y) \longrightarrow S(\alpha)$ or $M_n(x, \alpha) \longrightarrow S(\alpha)$.

Commutative reductions

Weakening premise inference - A proof the decorated name of which matches the term $mix(\mathcal{W}(x), y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\ \Pi_1}{\Sigma, A^k\vdash\Delta}}{\Sigma,\Gamma, A^k, A^{k'}\vdash\Delta,\Gamma'}\,\mathcal{W} \qquad \overset{\vdots\ \Pi_2}{\Sigma'\vdash A^m,\Delta'}}{\Sigma,\Gamma,\Sigma'\vdash\Delta,\Gamma',\Delta'}\,mix_{|A|}\,,$$

is replaced by the proof:

$$\dfrac{\dfrac{\overset{\vdots\ \Pi_1}{\Sigma, A^k\vdash\Delta} \qquad \overset{\vdots\ \Pi_2}{\Sigma'\vdash A^m,\Delta'}}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\,mix_{|A|}\,;}{\Sigma,\Gamma,\Sigma'\vdash\Delta,\Gamma',\Delta'}\,\mathcal{W}$$

the interpretation of this reduction follows the rewriting rule: $M_n(S(x), y) \longrightarrow S(M_n(x, y))$. For proofs the decorated names of which match the term $mix(x, \mathcal{W}(y))$ we define by symmetry a mix elimination rule the interpretation of which follows the rewriting rule: $M_n(x, S(y)) \longrightarrow S(M_n(x, y))$.

Contraction premise inference - A proof the decorated name of which matches the term $mix(\mathcal{C}(x), y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\ \Pi_1}{\Sigma,(A_1)^{k_1},\ldots,(A_n)^{k_n}\vdash(B_1)^{l_1},\ldots,(B_m)^{l_m},\Delta}}{\Sigma,(A_1)^{\delta(k_1)},\ldots,(A_n)^{\delta(k_n)}\vdash(B_1)^{\delta(l_1)},\ldots,(B_m)^{\delta(l_m)},\Delta}\,\mathcal{C} \qquad \overset{\vdots\ \Pi_2}{\Sigma'\vdash(A_1)^j,\Delta'}}{\Sigma,(A_2)^{\delta(k_2)},\ldots,(A_n)^{\delta(k_n)},\Sigma'\vdash(B_1)^{\delta(l_1)},\ldots,(B_m)^{\delta(l_m)},\Delta,\Delta'}\,mix_{|A_1|}\,,$$

is replaced by the proof:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(A_1)^{k_1},\dots,(A_n)^{k_n}\vdash(B_1)^{l_1},\dots,(B_m)^{l_m},\Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma'\vdash(A_1)^{j},\Delta'}}{\Sigma,(A_2)^{k_2},\dots,(A_n)^{k_n},\Sigma'\vdash(B_1)^{l_1},\dots,(B_m)^{l_m},\Delta,\Delta'}\;mix_{|A_1|}}{\Sigma,(A_2)^{\delta(k_2)},\dots,(A_n)^{\delta(k_n)},\Sigma'\vdash(B_1)^{\delta(l_1)},\dots,(B_m)^{\delta(l_m)},\Delta,\Delta'}\;\mathcal{C}\ ;$$

the interpretation of this reduction follows the rewriting rule: $M_n(S(x),y) \longrightarrow S(M_n(x,y))$. For proofs the decorated names of which match the term $mix(\mathcal{C}(x),y)$ we define by symmetry a mix elimination rule the interpretation of which follows the rewriting rule: $M_n(x,S(y)) \longrightarrow S(M_n(x,y))$.

Passive negation premise inference - A proof the decorated name of which matches the term $mix(\neg_{\mathcal{L}}(x),y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(C)^k\vdash B,\Delta}}{\Sigma,(C)^k,\neg B\vdash\Delta}\;\neg_{\mathcal{L}} \quad \overset{\vdots\;\Pi_2}{\Sigma'\vdash(C)^m,\Delta'}}{\Sigma,\neg B,\Sigma'\vdash\Delta,\Delta'}\;mix_{|C|}\ ,$$

is replaced by the proof:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(C)^k\vdash B,\Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma'\vdash(C)^m,\Delta'}}{\Sigma,\Sigma'\vdash B,\Delta,\Delta'}\;mix_{|C|}\ ;}{\Sigma,\neg B,\Sigma'\vdash\Delta,\Delta'}\;\neg_{\mathcal{L}}$$

the interpretation of this reduction follows the rewriting rule: $M_n(L_1(x),y) \longrightarrow L_1(M_n(x,y))$. For proofs whose decorated names match the term $mix(\neg_{\mathcal{R}}(x),y)$ we define a mix elimination rule the interpretation of which follows the latter rewriting rule too. For proofs the decorated names of which match the term $mix(x,\neg_{\mathcal{R}}(y))$ or $mix(x,\neg_{\mathcal{L}}(y))$ we define by symmetry mix elimination rules the interpretation of which follow the rewriting rule: $M_n(x,L_1(y)) \longrightarrow L_1(M_n(x,y))$.

Passive disjunction premise inference - A proof the decorated name of which matches the term $mix(\vee_{\mathcal{L}}(x_1,x_2),y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(D)^k,B\vdash\Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma,(D)^k,C\vdash\Delta}}{\Sigma,(D)^k,B\vee C\vdash\Delta}\;\vee_{\mathcal{L}} \quad \overset{\vdots\;\Pi_3}{\Sigma'\vdash(D)^m,\Delta'}}{\Sigma,B\vee C,\Sigma'\vdash\Delta,\Delta'}\;mix_{|D|}\ ,$$

is replaced by the proof:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma,(D)^k,B\vdash\Delta} \quad \overset{\vdots\;\Pi_3}{\Sigma'\vdash(D)^m,\Delta'}}{\Sigma,B,\Sigma'\vdash\Delta,\Delta'}\;mix_{|D|} \quad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma,(D)^k,C\vdash\Delta} \quad \overset{\vdots\;\Pi_3}{\Sigma'\vdash(D)^m,\Delta'}}{\Sigma,C,\Sigma'\vdash\Delta,\Delta'}\;mix_{|D|}\ ;}{\Sigma,B\vee C,\Sigma'\vdash\Delta,\Delta'}\;\vee_{\mathcal{L}}$$

the interpretation of this reduction follows the rewriting rule: $M_n(L_2(x,y),z) \longrightarrow L_2(M_n(x,z),M_n(y,z))$.

A proof the decorated name of which matches the term $mix(\vee_{\mathcal{R}}(x), y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, D^k \vdash B, \Delta}}{\Sigma, D^k \vdash B \vee C, \Delta}\,\vee_{\mathcal{R}} \qquad \overset{\vdots\;\Pi_2}{\Sigma' \vdash D^m, \Delta'}}{\Sigma, \Sigma' \vdash B \vee C, \Delta, \Delta'}\; mix_{|D|}\,,$$

is replaced by the proof:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, D^k \vdash B, \Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma' \vdash D^m, \Delta'}}{\Sigma, \Sigma' \vdash B, \Delta, \Delta'}\; mix_{|D|}\,;}{\Sigma, \Sigma' \vdash B \vee C, \Delta, \Delta'}\,\vee_{\mathcal{R}}$$

the interpretation of this reduction follows the rewriting rule: $M_n(L_1(x), y) \longrightarrow L_1(M_n(x, y))$. For proofs the decorated names of which match $mix(x, \vee_{\mathcal{L}}(y_1, y_2))$ or $mix(x, \vee_{\mathcal{R}}(y))$ we define by symmetry mix elimination rules the interpretations of which follow one of the rewriting rules:

$$M_{n+1}(x, L_2(y, z)) \longrightarrow L_2(M_{n+1}(x, y), M_{n+1}(x, z))\,,$$

$$\text{or} \quad M_{n+1}(x, L_1(y)) \longrightarrow L_1(M_{n+1}(x, y))\,.$$

Passive existential premise inference - A proof Π the decorated name of which matches the term $mix(\exists_{\mathcal{L}}(x), y)$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (C)^k, B(y) \vdash \Delta}}{\Sigma, (C)^k, \exists y B(y) \vdash \Delta}\,\exists_{\mathcal{L}} \qquad \overset{\vdots\;\Pi_2}{\Sigma' \vdash (C)^m, \Delta'}}{\Sigma, \exists y B(y), \Sigma' \vdash \Delta, \Delta'}\; mix_{|C|}\,,$$

where by the equivalence of proofs up to bounded variable renaming (c.f. remark 2.1) we can assume without loss of generality that the variable y does not occur in the proof Π_2; and given a variable z which does not occur in Π then by the capture-avoiding substitution lemma, $\Pi(y \leftarrow z)$ is a proof, so we can replace the proof Π by the proof:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1(y \leftarrow z)}{\Sigma, (C)^k, B(y \leftarrow z) \vdash \Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma' \vdash (C)^m, \Delta'}}{\Sigma, \Sigma', B(y \leftarrow z) \vdash \Delta, \Delta'}\; mix_{|C|}\,;}{\Sigma, \exists z B(y \leftarrow z), \Sigma' \vdash \Delta, \Delta'}\,\exists_{\mathcal{L}}$$

the interpretation of this reduction follows the rewriting rule: $M_n(L_1(x), y) \longrightarrow L_1(M_n(x, y))$. For proofs whose decorated names match the term $mix(\exists_{\mathcal{R}}(x), y)$ we define a mix elimination rule the interpretation of which follows the latter rewriting rule too. For proofs the decorated names of which match the term $mix(x, \exists_{\mathcal{R}}(y))$ or $mix(x, \exists_{\mathcal{L}}(y))$ we define by symmetry mix elimination rules the interpretation of which follow the rewriting rule: $M_n(x, L_1(y)) \longrightarrow L_1(M_n(x, y))$.

Symmetrical essential reductions

Active negation premises - A proof the decorated name of which matches $mix(\neg^*_{\mathcal{L}}(x), \neg^*_{\mathcal{R}}(y))$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (\neg B)^n \vdash B, \Delta}}{\Sigma, (\neg B)^n, \neg B \vdash \Delta}\,\neg_{\mathcal{L}} \qquad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma', B \vdash (\neg B)^m, \Delta'}}{\Sigma' \vdash \neg B, (\neg B)^m, \Delta'}\,\neg_{\mathcal{R}}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,mix_{|\neg B|}\,,$$

is replaced by the proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (\neg B)^n \vdash B, \Delta}}{\Sigma, (\neg B)^n, \neg B \vdash \Delta}\,\neg_{\mathcal{L}} \quad \overset{\vdots\;\Pi_2}{\Sigma', B \vdash (\neg B)^m, \Delta'}}{\Sigma, \Sigma', B \vdash \Delta, \Delta'}\,mix_{|\neg B|} \quad \dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (\neg B)^n, \vdash B, \Delta} \quad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma', B \vdash (\neg B)^m, \Delta'}}{\Sigma' \vdash \neg B, (\neg B)^m, \Delta'}\,\neg_{\mathcal{R}}}{\Sigma, \Sigma' \vdash B, \Delta, \Delta'}\,mix_{|\neg B|}}{\Sigma, \Sigma', \Sigma, \Sigma' \vdash \Delta, \Delta', \Delta, \Delta'}\,mix_{|B|}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,\mathcal{C}\;;$$

the interpretation of this reduction follows the rewriting rule:

$$M_{n+1}(L_1(x), L_1(y)) \longrightarrow S(M_n(M_{n+1}(L_1(x), y), M_{n+1}(x, L_1(y))))\,.$$

Active disjunction premises - A proof the decorated name of which matches the term $mix(\vee^*_{\mathcal{L}}(x_1, x_2), \vee^*_{\mathcal{R}}(y))$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (B \vee C)^n, B \vdash \Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma, (B \vee C)^n, C \vdash \Delta}}{\Sigma, (B \vee C)^n, B \vee C \vdash \Delta}\,\vee_{\mathcal{L}} \qquad \dfrac{\overset{\vdots\;\Pi_3}{\Sigma' \vdash B, (B \vee C)^m, \Delta'}}{\Sigma' \vdash B \vee C, (B \vee C)^m, \Delta'}\,\vee_{\mathcal{R}}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,mix_{|B \vee C|}\,,$$

is replaced by the proof:

$$\dfrac{\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (B \vee C)^n, B \vdash \Delta} \quad \dfrac{\overset{\vdots\;\Pi_3}{\Sigma' \vdash B, (B \vee C)^m, \Delta'}}{\Sigma' \vdash B \vee C, (B \vee C)^m, \Delta'}}{\Sigma, \Sigma', B \vdash \Delta, \Delta'} \quad \dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (B \vee C)^n, B \vdash \Delta} \quad \overset{\vdots\;\Pi_2}{\Sigma, (B \vee C)^n, C \vdash \Delta}}{\Sigma, (B \vee C)^n, B \vee C \vdash \Delta} \quad \overset{\vdots\;\Pi_3}{\Sigma' \vdash (B \vee C)^m, B, \Delta}}{\Sigma, \Sigma' \vdash B, \Delta, \Delta'}}{\Sigma, \Sigma', \Sigma, \Sigma' \vdash \Delta, \Delta', \Delta, \Delta'}\,mix_{|B|}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,\mathcal{C}\;;$$

the interpretation of this reduction follows the rewriting rule:

$$M_{n+1}(L_2(x, y), L_1(z)) \longrightarrow S(M_n(M_{n+1}(x, L_1(z)), M_{n+1}(L_2(x, y), z)))\,.$$

Active existential premises - A proof the decorated name of which matches the term $mix(\exists^*_{\mathcal{L}}(x), \exists^*_{\mathcal{R}}(y))$ and given by:

$$\dfrac{\dfrac{\overset{\vdots\;\Pi_1}{\Sigma, (\exists x B)^n, B(x \leftarrow y) \vdash \Delta}}{\Sigma, (\exists x B)^n, \exists x B(x) \vdash \Delta}\,\exists_{\mathcal{L}} \qquad \dfrac{\overset{\vdots\;\Pi_2}{\Sigma' \vdash B(x \leftarrow t), (\exists x B)^m, \Delta'}}{\Sigma' \vdash \exists x B, (\exists x B)^m, \Delta'}\,\exists_{\mathcal{R}}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,mix_{|\exists x B|}\,,$$

is replaced by the proof:

$$
\dfrac{
  \dfrac{
    \begin{array}{c}\vdots\ \Pi'_1(x\leftarrow t)\\ \Sigma,(\exists xB)^n,B(x\leftarrow t)\vdash\Delta\end{array}
    \quad
    \dfrac{\begin{array}{c}\vdots\ \Pi_2\\ \Sigma'\vdash B(x\leftarrow t),(\exists xB)^m,\Delta'\end{array}}{\Sigma'\vdash\exists xB,(\exists xB)^m,\Delta'}
  }{\Sigma,\Sigma',B(x\leftarrow t)\vdash\Delta,\Delta'}
  \quad
  \dfrac{
    \dfrac{\begin{array}{c}\vdots\ \Pi_1\\ \Sigma,(\exists xB)^n,B(x\leftarrow y)\vdash B,\Delta\end{array}}{\Sigma,(\exists xB)^n,\exists xB\vdash\Delta}
    \quad
    \begin{array}{c}\vdots\ \Pi_2\\ \Sigma'\vdash B(x\leftarrow t),(\exists xB)^m,\Delta'\end{array}
  }{\Sigma,\Sigma'\vdash B(x\leftarrow t),\Delta,\Delta'}
  ;
}{
  \dfrac{\Sigma,\Sigma',\Sigma,\Sigma'\vdash\Delta,\Delta',\Delta,\Delta'}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\ \mathcal{C}
}\ mix_{|B|}
$$

where Π'_1 is a proof obtained by renaming the bounded variables in Π_1 by new variable names which do not occur in the proof Π; thus by the capture-avoiding substitution lemma $\Pi'_1(x \leftarrow t)$ is a proof of the sequent $\Sigma, (\exists xB)^n, B(x \leftarrow t) \vdash \Delta$ and has the same name as Π_1. Therefore the interpretation of this reduction follows the rewriting rule:

$$M_{n+1}(L_1(x), L_1(y)) \longrightarrow S(M_n(M_{n+1}(x, L_1(y)), M_{n+1}(L_1(x), y)))\,.$$

3.6. The symmetrical mix elimination $\mathcal{E}_{LK}$. A mix elimination system $\mathcal{E}_{LK}$ for the sequent calculus LK is obtained by a modification of the system $\mathcal{E}_{LK_{sp}}$ following the criteria given below:

> Structural premise mix elimination rules of $\mathcal{E}_{LK}$ are particular cases of the structural mix elimination rules of $\mathcal{E}_{LK_{sp}}$. Non-structural mix elimination rules of $\mathcal{E}_{LK}$ are obtained by replacing in the non-structural mix elimination rules of $\mathcal{E}_{LK_{sp}}$ the use of parallel structural rules by several applications of structural rules in $\mathcal{E}_{LK}$.

For example, the mix elimination rules in $\mathcal{E}_{LK_{sp}}$ interpreted by the rewriting rules:

$$
\begin{array}{rcl}
M(h, x, \alpha) & \longrightarrow & S(\alpha)\,, \\
M\left(s(h), L_2(x, y), L_1(z)\right) & \longrightarrow & S\left(M\left(h, M\left(s(h), y, L_1(z)\right), M\left(s(h), L_2(x, y), z\right)\right)\right)\,;
\end{array}
$$

have to be replaced by mix elimination rules in $\mathcal{E}_{LK}$ interpreted by the infinite sets of rewriting rules:

$$
\begin{array}{rcl}
\{M(h, x, \alpha) & \longrightarrow & S^k(\alpha)\}_{k\in\mathbb{N}}\,, \\
\{M\left(s(h), L_2(x, y), L_1(z)\right) & \longrightarrow & S^k\left(M\left(h, M\left(s(h), y, L_1(z)\right), M\left(s(h), L_2(x, y), z\right)\right)\right)\}_{k\in\mathbb{N}}\,.
\end{array}
$$

The mix elimination system $\mathcal{E}_{LK}$ is interpreted by the infinite rewriting rule system denoted by $\mathcal{R}'_{LK}$ and given by the proof term rewriting rules over the signature $\mathcal{F}'$:

$$
\begin{array}{rcl}
\{M(h, x, \alpha) & \longrightarrow & S^k(\alpha)\}_{k\in\mathbb{N}}\,, \\
\{M(h, \alpha, y) & \longrightarrow & S^k(\alpha)\}_{k\in\mathbb{N}}\,, \\
M\left(h, L_1(x), y\right) & \longrightarrow & L_1\left(M\left(h, x, y\right)\right)\,, \\
M\left(h, x, L_1(y)\right) & \longrightarrow & L_1\left(M\left(h, x, y\right)\right)\,, \\
\{M\left(h, S(x), y\right) & \longrightarrow & S^i\left(M\left(h, x, y\right)\right)\}_{i\in\{0,1\}}\,, \\
\{M\left(h, x, S(y)\right) & \longrightarrow & S^i\left(M\left(h, x, y\right)\right)\}_{i\in\{0,1\}}\,, \\
M\left(h, L_2(x, y), z\right) & \longrightarrow & L_2\left(M(h, x, z), M(h, y, z)\right)\,, \\
M\left(h, x, L_2(y, z)\right) & \longrightarrow & L_2\left(M(h, x, y), M(h, x, z)\right)\,, \\
\{M\left(s(h), L_1(x), L_1(y)\right) & \longrightarrow & S^k\left(M\left(h, M\left(s(h), L_1(x), y\right), M\left(s(h), x, L_1(y)\right)\right)\right)\}_{k\in\mathbb{N}}\,,
\end{array}
$$

$$\begin{array}{rcl}
\{M\left(s(h), L_1(x), L_1(y)\right) & \longrightarrow & S^k\left(M\left(h, M\left(s(h), x, L_1(y)\right), M\left(s(h), L_1(x), y\right)\right)\right)\}_{k\in\mathbb{N}}\,, \\
\{M\left(s(h), L_2(x,y), L_1(z)\right) & \longrightarrow & S^k\left(M\left(h, M\left(s(h), x, L_1(z)\right), M\left(s(h), L_2(x,y), z\right)\right)\right)\}_{k\in\mathbb{N}}\,, \\
\{M\left(s(h), L_2(x,y), L_1(z)\right) & \longrightarrow & S^k\left(M\left(h, M\left(s(h), y, L_1(z)\right), M\left(s(h), L_2(x,y), z\right)\right)\right)\}_{k\in\mathbb{N}}\,.
\end{array}$$

3.7. Ambiguity of the mix elimination systems $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$. The mix elimination rules in the system $\mathcal{E}_{LK_{sp}}$ are not exclusive. Indeed, up to symmetry we have the following *ambiguous patterns*:

$M_n(\alpha, S(y))$	
$M_n(\alpha, L_1(y))$	$M_n(\alpha, L_2(y,z))$
$M_n(S(x), L_1(y))$	$M_n(S(x), L_2(y,z))$
$M_n(L_1(x), L_1(y))$	$M_n(L_1(x), L_2(y,z))$

If the interpretation of a proof matches one of the former ambiguous patterns then we can replace this proof following either:

a left side mix elimination rule - this is the case if the left-hand side of the associated rewriting rule is among the terms: $M_n(\alpha, y)$, $M_n(S(x), y)$, $M_n(L_1(x), y)$ or $M_n(L_2(x,y), z)$;

or a right side elimination rule - this is the case if the left-hand side of the associated rewriting rule is among the terms: $M_n(x, \alpha)$, $M_n(x, S(y))$, $M_n(x, L_1(y))$ or $M_n(x, L_2(y,z))$.

We notice that the mix elimination system $\mathcal{E}_{LK}$ is also ambiguous.

Moreover, this ambiguity is not removed by the mix elimination system $\mathcal{E}_{LK_{sp}}$, for instance, a proof with the interpretation matching the term $M_n(S(x), L_1(y))$ can be reduced in two different ways:

$$M_n(S(x), L_1(y)) \longrightarrow S(M_n(x, L_1(y))) \longrightarrow L_1(S(M_n(x,y)))\,,$$

$$\text{or}\quad M_n(S(x), L_1(y)) \longrightarrow L_1(M_n(S(x), y)) \longrightarrow S(L_1(M_n(x,y)))\,,$$

where the terms $L_1(S(M_n(x,y)))$ and $S(L_1(M_n(x,y)))$ are not equal up to successive mix eliminations. Thus, the mix elimination system $\mathcal{E}_{LK_{sp}}$ is not confluent (where, denoting by $\mathcal{R}^*$ the transitive closure of a binary relation $\mathcal{R}$ over proofs, the binary relation $\mathcal{R}$ satisfies the *confluence property* if for any triplets of proofs Π, Π_1 and Π_2 such that the relations $\Pi\mathcal{R}\Pi_1$ and $\Pi\mathcal{R}\Pi_2$ are satisfied there is a proof Π_3 such that the relations $\Pi_1\mathcal{R}^*\Pi_3$ and $\Pi_2\mathcal{R}^*\Pi_3$ are also satisfied).

An arbitrary way of avoiding the ambiguity of the mix elimination systems $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$ is to give priority in the application of mix elimination rules to the left side mix elimination rules over the right side mix elimination rules. Such restrictions are said to be *left-priority* restrictions of the mix elimination systems $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$, denoted respectively by $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$, and are defined by the criterion:

> if a proof can be replaced following either a left side or a right side mix elimination rule then the replacement must follow the left side mix elimination rule.

In section 4 we show that $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$ are confluent and strongly normalizing.

4. Strong normalization proofs

4.1. Introduction. Let $\mathcal{E}$ be a mix elimination system for a K-calculus. A *redex* is a proof which can be replaced following a mix elimination rule. If the subproof $\Pi|_p$ (at position p) of a proof Π is a redex then $\Pi|_p$ is said to be *the redex at position p of the proof* Π. A subproof of Π which is a redex is said to be a *redex* $\hat{\Pi}$ *of the proof* Π; an *innermost redex* $\hat{\Pi}$ of a proof Π is a minimal redex (with respect to the subdeduction order) of Π, thus any proper subproof of an innermost redex is mix free. A *reduction* (respectively *innermost reduction*) of a redex $\hat{\Pi}$ in a proof Π is the replacement in the proof Π of the redex $\hat{\Pi}$ (respectively innermost redex $\hat{\Pi}$) following a mix elimination rule; let us denote by Π' the proof obtained once $\hat{\Pi}$ has been replaced in Π, we say that the proof Π is reduced by $\mathcal{E}$ to the proof Π'; this reduction relation is denoted $\Pi\mathcal{E}\Pi'$. As usual we denote by $\mathcal{E}^*$ the reflexive and transitive closure of the relation $\mathcal{E}$. A proof Π is *$\mathcal{E}$-normal* if it cannot be reduced by $\mathcal{E}$.

Let $\mathcal{E}$ be a mix elimination system for a K-calculus. An *$\mathcal{E}$-derivation*, respectively *innermost $\mathcal{E}$-derivation*, of a proof Π_0 or sequence of $\mathcal{E}$-reductions, respectively innermost $\mathcal{E}$-reductions, is a sequence of K-proofs $P =< \Pi_0, \ldots, \Pi_n, \ldots >$ such that for each pair of successive K-proofs Π_i and Π_{i+1} of P the relation $\Pi_i\mathcal{E}\Pi_{i+1}$ holds. A proof Π is *strongly $\mathcal{E}$-normalizing* if all its $\mathcal{E}$-derivations are finite. An *$\mathcal{E}$-normalization*, respectively *innermost $\mathcal{E}$-normalization*, is a $\mathcal{E}$-derivation, respectively innermost $\mathcal{E}$-derivation, $< \Pi_0, \ldots, \Pi_n, \ldots >$ such that Π_n is an $\mathcal{E}$-normal proof. A mix elimination system $\mathcal{E}$ is *confluent* if for each triplet of K-proofs Π, Π' and Π'' such that $\Pi\mathcal{E}\Pi'$ and $\Pi\mathcal{E}\Pi''$ there is a K-proof Π''' such that $\Pi'\mathcal{E}^*\Pi'''$ and $\Pi''\mathcal{E}^*\Pi'''$; so a confluent mix elimination system satisfies the diagram:

$$\begin{tikzcd}
\Pi \arrow[r, "\mathcal{E}"] \arrow[d, "\mathcal{E}"'] & \Pi' \arrow[d, "\mathcal{E}^*"] \\
\Pi'' \arrow[r, "\mathcal{E}^*"] & \Pi'''
\end{tikzcd}$$

The mix elimination systems $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$ satisfy the cornerstone:

THEOREM 4.1 (Gentzen's Hauptsatz). [Gen35] *Any LK-proof (respectively LK_{sp}-proof) is (innermost) normalizing by the mix elimination systems $\mathcal{E}_{LK}$ (respectively $\mathcal{E}_{LK_{sp}}$).*

PROOF. It is enough to prove this assertion for an innermost redex:

$$\dfrac{\dfrac{\vdots\ \Pi_0}{\Sigma_0, A^n \vdash \Delta_0} \quad \dfrac{\vdots\ \Pi_1}{\Sigma_1 \vdash A^m, \Delta_1}}{\Sigma, \Sigma' \vdash \Sigma, \Delta'}\ mix_{|A|}\ .$$

The proof is carried out by induction on the lexicographic order on $(|A|, |\Pi_0|, |\Pi_1|)$ (given by the degree of the mix formula A, the size of the proof Π_0 and the size of the proof Π_1). ■

The question of strong normalization of the original non-symmetrical mix elimination system proposed in [Gen35] has been solved in [Dra88]; nevertheless the symmetrical mix elimination systems $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$ turn out to be more suitable for strong normalization proofs. We already gave in section 3 historical remarks on strong normalization proofs for sequent calculi; we will give, below, further classification and further details about these proofs and we will show how these techniques are applied in this work to prove strong normalization of $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$.

Structural induction proofs - Using this technique one proves that if the immediate subproofs of a proof Π are strongly normalizing then the proof Π itself is strongly normalizing; it was used in [Dra88] to prove strong normalization of the original non-symmetrical mix elimination system proposed in [Gen35] and by Coquand in [Her95] to prove strong normalization of Herbelin's sequent calculus LJT; we apply it in section 4.4 to prove strong normalization of $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$. (We expect that such structural inductive proofs can be obtained by automated theorem provers.)

Innermost normalization proofs - This technique uses the parallel moves lemma property, see section 4.2, which is a sufficient condition to ensure that innermost normalization stated in the Hauptsatz theorem implies strong normalization for orthogonal mix elimination systems, this is based on results in [O'D77], these results were extended in [Gra92]; we used it in [Tah92] to prove strong normalization of the propositional fragment of $\mathcal{E}^l_{LK}$ and we apply it in section 4.2 to prove a parallel moves lemma and hence the strong normalization for the orthogonal restrictions $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$.

Recursive path ordering monotone interpretations proofs - This technique applies term rewriting recursive path ordering termination proofs developed in [Der82] and [KL80], it was used, building on Okada's lectures, in [CRS94] to prove the strong normalization of fragments of the linear calculus and in [CRS96] to prove strong normalization of $\mathcal{E}_{LK}$; we apply it in section 4.3 to give a *finite* rewriting system interpretation of $\mathcal{E}_{LK_{sp}}$ and hence a totally machine checkable proof of strong normalization of $\mathcal{E}_{LK_{sp}}$.

Natural monotone interpretations proofs - Using this technique one defines monotone homomorphisms from the algebra of proof names (ordered by the rewriting system associated to mix elimination systems) to the algebra of subrecursive functions in the Grzegorczyk hierarchy, for a study in subrecursive functions we refer the reader to [Ros84]; it was used in [CRS96] to give subrecursive upper bounds for the lengths of derivations in some fragments of the linear calculus; we apply it in section 4.5 to give subrecursive upper bounds for the lengths of $\mathcal{E}_{LK}$ and $\mathcal{E}_{LK_{sp}}$ derivations.

4.2. Strong normalization of orthogonal restrictions. In this section we apply orthogonal rewriting systems techniques to constructor based orthogonal mix elimination systems; we refer the reader to [HL92] for a technical study and to [Klo92] and [DJ90] for reviews on this subject.

Constructor orthogonal mix elimination systems. A proof Π *extends* a deduction $\mathcal{D}$ if for each common node p of Π and $\mathcal{D}$ the inference which labels the node p in Π is equal to the inference which labels the node p in $\mathcal{D}$; informally, a proof Π extends a deduction $\mathcal{D}$ if the deduction $\mathcal{D}$ is obtained by erasing subproofs of the proof Π.

The *replaced deduction* of a mix elimination rule is the deduction obtained by skipping the subproofs which are copied in the replaced proof; i.e. The replaced deduction of a mix elimination rule ρ is the maximal deduction (with respect to the subdeduction order) which is extended by all proofs that can be replaced following the mix elimination rule ρ. For instance, the replaced deduction of the mix elimination rule given in the illustrative example 3.1 is the deduction:

$$\dfrac{\dfrac{\Sigma, (\neg B)^n \vdash B, \Delta}{\Sigma, (\neg B)^{n+1} \vdash \Delta}\,\neg_{\mathcal{L}} \qquad \dfrac{\Sigma', B \vdash (\neg B)^m, \Delta'}{\Sigma' \vdash (\neg B)^{m+1}, \Delta'}\,\neg_{\mathcal{R}}}{\Sigma, \Sigma' \vdash \Delta, \Delta'}\,mix_{|\neg B|}\,.$$

A mix elimination is said to be a *constructor mix elimination* if the non-root nodes of the replaced deduction are labeled by non-mix inferences. A mix elimination system is said to be a *constructor mix elimination system* if it is constituted of constructor mix elimination rules. By construction, $\mathcal{E}^l_{LK_{sp}}$ is a constructor mix elimination system.

A mix elimination rule is *left linear* if the non-axiom hypotheses of the associated replaced deduction are pairwise different; a mix elimination system is *left linear* if each one of its mix elimination rules is left linear. Two deductions are *unifiable* if they admit a common proof extension. A constructor mix elimination system is *orthogonal* if it is left linear and there are no two mix elimination rules with the same priority and with unifiable mix replaced deductions.

As a consequence of the left-priority we imposed on $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$ we have:

FACT 4.1. *The mix elimination systems $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$ are constructor orthogonal mix elimination systems.*

Parallel moves lemma. Two sequences of natural numbers are *comparable* if one of them is a prefix of the other. A sequence P of redexes of a proof Π, indexed by their respective positions in the proof Π is said to be a *sequence of parallel redexes of the proof* Π if their respective positions in Π are pairwise incomparable. Given two sequences of parallel redexes P and Q of a proof Π, we denote by $P - Q$ the sequence of parallel redexes in P, the positions of which are not positions of any redex in Q. We remark that a sequence of innermost redexes of a proof Π, indexed by their respective positions in the proof Π is also a sequence of parallel redexes of this proof. A *parallel reduction* of a proof Π at a sequence of parallel redexes P is the replacement of each redex in P following a mix elimination rule; this is denoted by (Π, P, Π') or $\Pi \xrightarrow{P} \Pi'$.

Parallel reductions for constructor orthogonal mix elimination systems satisfy a version of the following lemma proved by Rosen [Ros73] and extended by G. Huet in [Hue80] to orthogonal rewriting systems.

LEMMA 4.1 (Parallel moves lemma [Ros73,Hue80]). *Let $\mathcal{E}$ be a constructor orthogonal mix elimination system. For each pair of parallel reductions (Π_0, Q, Π'_0) and (Π_0, P, Π_1) there is a pair of sequences of parallel redexes P' and Q' such that the following diagram is satisfied:*

$$\begin{tikzcd}
\Pi_0 \arrow[r, "P"] \arrow[d, "Q"'] & \Pi_1 \arrow[d, "Q'"] \\
\Pi'_0 \arrow[r, "P'"] & \Pi'_1 .
\end{tikzcd}$$

We refer the reader to [Hue80] for a proof of this lemma; the only additional delicate point in the adaptation of Rosen and Huet's proof to constructor orthogonal mix elimination systems is the way of handling of capture-avoiding substitutions performed by existential premises mix elimination rules. The confluence property is the particular case of the parallel moves lemma when, under the same assumptions as in this lemma, the sequences P and Q have both only one redex. Hence, we have the:

Corollary 4.1. *The mix elimination systems $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$ are confluent.*

In this work we need only to prove that constructor orthogonal mix elimination systems satisfy the following version of the parallel moves lemma.

Lemma 4.2. *Let $\mathcal{E}$ be a constructor orthogonal mix elimination system. For each pair of parallel reductions (Π_0, Q, Π'_0) and (Π_0, P, Π_1) such that Q is a sequence of innermost redexes of Π_0 there is a sequence of innermost redexes Q'; such that:*

- *the following diagram holds:*

$$\begin{tikzcd}
\Pi_0 \arrow[r, "P"] \arrow[d, "Q"'] & \Pi_1 \arrow[d, "Q'"] \\
\Pi'_0 \arrow[r, "P-Q"] & \Pi'_1 ,
\end{tikzcd}$$

- *and if $P \subseteq Q$ then $Q' = Q - P$.*

Proof. The proof is obtained by induction on the lexicographic order on the pair of cardinalities of the sequences Q and P denoted by $(|Q|, |P|)$; the initial step is carried out in the following lemma. ■

Lemma 4.3. *Given $\mathcal{E}$ a constructor orthogonal mix elimination system, given a redex Π and denoting by Π' the proof obtained after reduction of the redex Π; if $\hat{\Pi}$ is an innermost redex of Π and if $\Pi \neq \hat{\Pi}$ then there is a sequence of innermost redexes Q of the proof Π' and a redex Π'' such that the following diagram holds:*

$$\begin{tikzcd}
\Pi \arrow[r, "\{\Pi\}"] \arrow[d, "\{\hat{\Pi}\}"'] & \Pi' \arrow[d, "Q"] \\
\Pi'' \arrow[r, "\{\Pi''\}"'] & \Pi''' .
\end{tikzcd}$$

Proof. We give a proof sketch for an exemplary case of a mix elimination rule belonging to $\mathcal{E}^l_{LK_{sp}}$. Given a redex Π:

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi_1\\ \hat\Sigma,(\exists\hat x\hat B)^{\hat n},\hat B(\hat x\leftarrow\hat y)\vdash\hat\Delta\end{array}}{\hat\Sigma,(\exists\hat x\hat B)^{\hat n},\exists\hat x\hat B(\hat x)\vdash\hat\Delta}\,\exists_{\mathcal L}\quad\dfrac{\begin{array}{c}\vdots\ \hat\Pi_2\\ \hat\Sigma'\vdash\hat B(\hat x\leftarrow\hat t),(\exists\hat x\hat B)^{\hat m},\hat\Delta'\end{array}}{\hat\Sigma'\vdash\exists\hat x\hat B,(\exists\hat x\hat B)^{\hat m},\hat\Delta'}\,\exists_{\mathcal R}}{\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta'}\,mix_{|\exists\hat x\hat B|}\\ \vdots\ \mathcal{D}_1\\ \Sigma,(\exists xB)^n,B(x\leftarrow y)\vdash\Delta\end{array}}{\Sigma,(\exists B)^n,\exists xB(x)\vdash\Delta}\,\exists_{\mathcal L}\qquad\dfrac{\begin{array}{c}\vdots\ \Pi_2\\ \Sigma'\vdash B(x\leftarrow t),(\exists xB)^m,\Delta'\end{array}}{\Sigma'\vdash\exists xB,(\exists xB)^m,\Delta'}\,\exists_{\mathcal R}}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\,mix_{|\exists xB|}\ ;$$

we denote by $\hat\Pi$ the redex:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi_1\\ \hat\Sigma,(\exists\hat x\hat B)^{\hat n},\hat B(\hat x\leftarrow\hat y)\vdash\hat\Delta\end{array}}{\hat\Sigma,(\exists\hat x\hat B)^{\hat n},\exists\hat x\hat B(\hat x)\vdash\hat\Delta}\,\exists_{\mathcal L}\quad\dfrac{\begin{array}{c}\vdots\ \hat\Pi_2\\ \hat\Sigma'\vdash\hat B(\hat x\leftarrow\hat t),(\exists\hat x\hat B)^{\hat m},\hat\Delta'\end{array}}{\hat\Sigma'\vdash\exists\hat x\hat B,(\exists\hat x\hat B)^{\hat m},\hat\Delta'}\,\exists_{\mathcal R}}{\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta'}\,mix_{|\exists\hat x\hat B|}\ ;$$

and by Π_1 the proof:

$$\begin{array}{c}\vdots\ \hat\Pi(x\leftarrow t)\\ (\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta')(x\leftarrow t)\\ \vdots\ \mathcal{D}_1(x\leftarrow t)\\ \Sigma,(\exists xB)^n,B(x\leftarrow t)\vdash\Delta\,.\end{array}$$

We suppose that the bounded variables of Π_1 and $\hat\Pi_1$ are renamed in such a way that the substitutions $\Pi_1(x\leftarrow t)$ and $\hat\Pi_1(\hat x\leftarrow\hat t)$ are capture-avoiding; hence the variable $\hat x$ does not occur in the term t and therefore we have the equality:

$$\hat\Pi_1(\hat x\leftarrow\hat t)(x\leftarrow t)=\hat\Pi_1(x\leftarrow t)(\hat x\leftarrow\hat t(x\leftarrow t))\,.$$

The proof Π admits at least two possible reductions, either we reduce the redex Π and obtain the proof Π' given by:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi(x\leftarrow t)\\ (\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta')(x\leftarrow t)\\ \vdots\ \mathcal{D}_1(x\leftarrow t)\\ \Sigma,(\exists xB)^n,B(x\leftarrow t)\vdash\Delta\end{array}\quad\dfrac{\begin{array}{c}\vdots\ \Pi_2\\ \Sigma'\vdash B(x\leftarrow t),(\exists xB)^m,\Delta'\end{array}}{\Sigma'\vdash\exists xB,(\exists xB)^m,\Delta'}}{\Sigma,\Sigma',B(x\leftarrow t)\vdash\Delta,\Delta'}\quad\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi\\ \hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta'\\ \vdots\ \mathcal{D}_1\\ \Sigma,(\exists xB)^n,B(x\leftarrow y)\vdash\Delta\end{array}}{\Sigma,(\exists xB)^n,\exists xB\vdash\Delta}\quad\begin{array}{c}\vdots\ \Pi_2\\ \Sigma'\vdash B(x\leftarrow t),(\exists xB)^m,\Delta'\end{array}}{\Sigma,\Sigma'\vdash B(x\leftarrow t),\Delta,\Delta'}}{\Sigma,\Sigma',\Sigma,\Sigma'\vdash\Delta,\Delta',\Delta,\Delta'}\,mix_{|B|}}{\Sigma,\Sigma'\vdash\Delta,\Delta'}\,\mathcal{C}\ ;$$

or we obtain a proof Π'' by replacing in the proof Π the redex $\hat\Pi$ by its reduction $\hat\Pi'$ given by:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi_1(\hat x\leftarrow\hat t)\\ \hat\Sigma,(\exists\hat x\hat B)^{\hat n},\hat B(\hat x\leftarrow\hat t)\vdash\hat\Delta\end{array}\quad\dfrac{\begin{array}{c}\vdots\ \hat\Pi_2\\ \hat\Sigma'\vdash\hat B(\hat x\leftarrow\hat t),(\exists\hat x\hat B)^{\hat m},\hat\Delta'\end{array}}{\hat\Sigma'\vdash\exists\hat x\hat B,(\exists\hat x\hat B)^{\hat m},\hat\Delta'}}{\hat\Sigma,\hat\Sigma',\hat B(\hat x\leftarrow\hat t)\vdash\hat\Delta,\hat\Delta'}\quad\dfrac{\dfrac{\begin{array}{c}\vdots\ \hat\Pi_1\\ \hat\Sigma,(\exists\hat x\hat B)^{\hat n},\hat B(\hat x\leftarrow\hat y)\vdash\hat B,\hat\Delta\end{array}}{\hat\Sigma,(\exists\hat x\hat B)^{\hat n},\exists\hat x\hat B\vdash\hat\Delta}\quad\begin{array}{c}\vdots\ \hat\Pi_2\\ \hat\Sigma'\vdash\hat B(\hat x\leftarrow\hat t),(\exists\hat x\hat B)^{\hat m},\hat\Delta'\end{array}}{\hat\Sigma,\hat\Sigma'\vdash\hat B(\hat x\leftarrow\hat t),\hat\Delta,\hat\Delta'}}{\hat\Sigma,\hat\Sigma',\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta',\hat\Delta,\hat\Delta'}\,mix_{|\hat B|}}{\hat\Sigma,\hat\Sigma'\vdash\hat\Delta,\hat\Delta'}\,\mathcal{C}\ .$$

If we reduce the redex Π'' and on the other hand we reduce the innermost redexes $\hat\Pi$ and $\hat\Pi(x\leftarrow t)$ in the proof Π', and since $\hat\Pi_1(\hat x\leftarrow\hat t)(x\leftarrow t)=\hat\Pi_1(x\leftarrow t)(\hat x\leftarrow\hat t(x\leftarrow t))$, we obtain the same proof denoted by Π'''; so we have the diagram:

$$\begin{tikzcd}
\Pi \arrow[r, "\{\Pi\}"] \arrow[d, "\{\hat{\Pi}\}"'] & \Pi' \arrow[d, "\{\hat{\Pi},\hat{\Pi}(x\leftarrow t)\}"] \\
\Pi'' \arrow[r, "\{\Pi''\}"'] & \Pi'''. \ \blacksquare
\end{tikzcd}$$

O'Donnell's lemma. Following O'Donnell [O'D77] the lemma 4.2 allows us to prove (in lemma 4.5) that if a proof Π admits an infinite sequence of reductions then the proof Π also admits an infinite sequence of innermost reductions.

Definition 4.1 (Derived sequence of parallel reductions). Let $\mathcal{E}$ be a constructor orthogonal mix elimination system. Let (Π_i, P_i, Π_{i+1}) be a finite or infinite sequence of parallel reductions, let Q_0 be a sequence of innermost parallel redexes of Π_0; a *derived sequence of parallel reductions* of (Π_i, P_i, Π_{i+1}) by Q_0 is a sequence of parallel reductions $(\Pi'_i, P'_i, \Pi'_{i+1})$, such that, for each natural number i, there is a sequence Q_{i+1} of innermost parallel redexes of Π_{i+1}, and such that we have the diagram:

$$\begin{tikzcd}
\Pi_i \arrow[r, "P_i"] \arrow[d, "Q_i"'] & \Pi_{i+1} \arrow[d, "Q_{i+1}"] \\
\Pi'_i \arrow[r, "P'_i"] & \Pi'_{i+1} \,.
\end{tikzcd}$$

We notice that the existence of derived sequences of parallel reductions is guaranteed by lemma 4.2.

Derived sequences of parallel reductions satisfy

Lemma 4.4 (O'Donnell [O'D77]). *Under the assumptions of the last definition; if the sequence of parallel reductions* (Π_i, P_i, Π_{i+1}) *is infinite then the derived sequence of parallel reductions* $(\Pi'_i, P'_i, \Pi'_{i+1})$ *of the sequence* (Π_i, P_i, Π_{i+1}) *by the sequence of innermost redexes* Q_0 *is infinite.*

Proof. Notice first that since Q_0 is a sequence of innermost parallel redexes (of Π_0) then, by lemma 4.2, each Q_i is a sequence of innermost parallel node. Suppose that:

$$\forall i \geq k \;\; P'_i = \emptyset\,,$$

thus since the redexes of Q_i are all innermost, by lemma 4.2, we have:

$$\forall i \geq k \qquad (P_i \subseteq Q_i) \wedge (Q_{i+1} = Q_i - P_i);$$

then, beyond k, the sequences P_i are pairwise disjoint and all included in Q_k, but Q_k is finite. ■

Lemma 4.5 (O'Donnell's lemma [O'D77]). *Let* $\mathcal{E}$ *be a constructor orthogonal mix elimination system, then a proof is strongly normalizing if and only if it is innermost normalizing.*

Proof. Let $(\Pi_i, Q_i, \Pi_{i+1})_{i=0,\ldots,n-1}$ be a sequence of innermost reductions such that Π_n is normal. If Π_0 accepts an infinite sequence of reductions, σ^0, then by lemma 4.4

the sequence σ^1 derived from σ^0 by Q_0 is an infinite sequence of reductions of Π_1. By iterating this process we obtain an infinite sequence σ^n of reductions of Π_n. ∎

We remark that the proof of the former proposition is not constructive since it is achieved using reductio ad absurdum; we refer the reader to [Tah94] for a constructive proof of O'Donnell's lemma.

As a corollary of Gentzen's theorem and O'Donnell's lemma we have:

PROPOSITION 4.1. *The mix elimination systems $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$ satisfy the strong normalization property.*

We remark that the conjunction of the lemma 4.2 and the strong normalization property implies the confluence of $\mathcal{E}^l_{LK}$ and $\mathcal{E}^l_{LK_{sp}}$.

REMARK 4.1. In this section we used the fact that, for orthogonal rewriting systems, innermost normalization implies strong normalization. We could also formulate a non-erasing orthogonal mix elimination system and use the fact that, for non-erasing orthogonal rewriting systems, weak normalization implies strong normalization. This latter approach has been successfully used to prove that weak normalization implies strong normalization for λ-typed calculi; we refer the reader to [Ned73], [Klo80], [Gro93], [KW94] and [Hon96] for works in this direction.

4.3. Strong normalization proofs by recursive path orderings. In this section we describe and apply a recursive strong normalization criterion based on partial well-order theory. We refer the reader to [Der82, KL80] for the initial studies on this subject or to [MZ94] for a new formulation of this matter. Strong normalization proofs for sequent calculi by recursive path orderings have been achieved in [CRS94, CRS96, DP96].

A *partial order* on a set S, denoted by $(S,<)$, is a non reflexive transitive binary relation on the set S; the reflexive closure of the partial order $(S,<)$ is denoted $(S,\leq)$. A finite or infinite sequence $(s_0,\ldots,s_i,\ldots)$ of elements of a partially ordered set $(S,<)$ is *increasing* if for each pair (s_i,s_{i+1}) of successive elements in σ the inequality $s_i < s_{i+1}$ holds. A partial order $(S,<)$ on a set S is a *partial well-order* if every infinite sequence of elements of the set S has an infinite increasing subsequence. A partial order $(S,<_2)$ is an *extension* of a partial order $(S,<_1)$ if for each pair (t,u) of elements of S such that the inequality $t <_1 u$ holds the inequality $t <_2 u$ holds too. Extensions of partial well-orders satisfy:

LEMMA 4.6 (Extension lemma). *Every partial order extending a partial well-order is a partial well-order.*

PROOF. Given a partial order $(S,<_2)$ extending a partial well-order $(S,<_1)$ and given a sequence σ of elements in S then by the partial well-order property of $(S,<_1)$ there is an infinite $<_1$-increasing subsequence of σ which is also an infinite $<_2$-increasing subsequence of σ since $(S,<_2)$ extends $(S,<_1)$. ∎

Given a partially ordered set $(S,<)$, the *lexicographic order* induced by $(S,<)$ on the set of finite sequences of elements of S, denoted by $(S^*,<^{lex})$, is defined by:

$$(s_0,\ldots,s_m) <^{lex} (t_0,\ldots,t_n) \qquad \Longleftrightarrow \qquad \exists i_0, \quad \forall i < i_0 \; s_i = t_i \quad \text{and} \quad s_{i_0} < t_{i_0}\,.$$

The lexicographic order $(S^*, <^{lex})$ induced by $(S, <)$ satisfies the well known properties:

PROPOSITION 4.2. *For a* totally *ordered set* $(S, <)$*, the lexicographic order* $(S^*, <^{lex})$ *induced by* $(S, <)$ *on the set of finite sequences of elements of* S *is also a total order.*

If $(S, <)$ *is a* well founded *order then for each natural number* n *the restriction of the lexicographic order to finite sequences of length less or equal to* n *denoted by* $(S^{\leq n}, <^{lex})$ *is also a* well founded *order.*

A *signature* $\mathcal{F}$ is a union of *disjoint* sets $\mathcal{F}_n$ indexed by natural numbers; the *arity* of elements in the set $\mathcal{F}_n$ is the natural number n, (we remark that each element of the signature has a *fixed* arity); the elements of $\mathcal{F}$ whose arity is zero are said to be *constants.* A signature is said to be *with bounded arities* if the set of arities of its elements is bounded. Given a signature $\mathcal{F}$ and an enumerable set of variable symbols $\mathcal{X}$, the set *of terms* $\mathcal{T}(\mathcal{F}, \mathcal{X})$ *over* $\mathcal{F}$ *and* $\mathcal{X}$ is the smallest set including $\mathcal{X}$ and constants such that the term $u := d(t_0, \ldots, t_m)$ belongs to $\mathcal{T}(\mathcal{F}, \mathcal{X})$ whenever $d \in \mathcal{F}_{m+1}$ and each t_i belongs to $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Each term t_i is said to be *the immediate subterm* at *coordinate* i of the term u and the letter d is said to be the *head* of the term u. A term without any occurrence of variables is said to be a *closed* or *ground* term over $\mathcal{F}$; the set of ground terms over $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F})$. Given a signature $\mathcal{F}$ and a set of variables $\mathcal{X}$, the *subterm order* on terms over $\mathcal{F}$ and $\mathcal{X}$, denoted by $(\mathcal{T}(\mathcal{F}, \mathcal{X}), \lhd)$ is defined recursively by:

$$s \lhd t = g(t_0, \ldots, t_n) \qquad \Longleftrightarrow \qquad \exists j \; s \lhd t_j \quad \text{or} \quad \exists j \; s = t_j \,.$$

A term s is a *subterm* of a term t if either s is equal to t or the inequality $s \lhd t$ is satisfied. Subterm orders satisfy:

FACT 4.2. *Given a signature* $\mathcal{F}$ *and a set of variables* $\mathcal{X}$*, the subterm order on terms over* $\mathcal{F}$ *and* $\mathcal{X}$*, denoted by* $(\mathcal{T}(\mathcal{F}, \mathcal{X}), \lhd)$ *is* well founded*.*

The induction on terms using the well founded subterm order is said to be a *structural induction.* The induction on n-tuples of terms using the lexicographic order on n-tuples of terms induced by the structural order on terms is said to be *the lexicographical structural induction on* n*-tuples of terms.*

A useful partial well-order for strong normalization theory of term rewriting system is:

DEFINITION 4.2 (Kruskal order). Let $(\mathcal{F}, <)$ be an ordered signature; the *Kruskal order* induced by $(\mathcal{F}, <)$ on ground terms over $\mathcal{F}$, denoted by $(\mathcal{T}(\mathcal{F}), <_K)$, is defined recursively as follows:

$$s = f(s_0, \ldots, s_m) <_K t = g(t_0, \ldots, t_n) \iff$$
$$\begin{cases} \exists j \; s \leq_K t_j \,, \\ \text{or } f < g \text{ and } \exists j_0 < \ldots < j_m \text{ such that } \forall i \; s_i \leq_K t_{j_i} \,, \\ \text{or } f = g \text{ and } \forall i \; s_i \leq_K t_i \,, \text{and } \exists i_0 \; s_{i_0} <_K t_{i_0} \,. \end{cases}$$

This Kruskal order is a partial well-order:

THEOREM 4.2 (Kruskal's tree theorem [Kru60]). *Let* $(\mathcal{F}, <)$ *be a* partial well-ordered *signature; the Kruskal order* $(\mathcal{T}(\mathcal{F}), <_K)$ *induced by* $(\mathcal{F}, <)$ *on ground terms over* $\mathcal{F}$ *is a* partial well-order.

Finite versions of this theorem turn out not to be provable in arithmetic theory and even in more powerful mathematical theories, for a presentation on this matter see [Gal91]; Kruskal's original proof is not constructive; for a constructive proof see [Wei94].

As a consequence of Kruskal's tree theorem the Kruskal order on ground terms over $\mathcal{F}$ admits a total well founded extension defined by Kamin and Levy:

Definition 4.3 (Lexicographic path order). Let $(\mathcal{F}, <)$ be an ordered signature; the *lexicographic path order* induced by $(\mathcal{F}, <)$ on ground terms over $\mathcal{F}$, denoted by $(\mathcal{T}(\mathcal{F}), <_{lpo})$, is defined recursively as follows:

$$s = f(s_0, \ldots, s_m) <_{lpo} t = g(t_0, \ldots, t_n) \iff$$
$$\begin{cases} \exists j \; s \leq_{lpo} t_j \,, \\ \text{or } f < g \text{ and } \forall i \; s_i <_{lpo} t \,, \\ \text{or } f = g \text{ and } \exists i_0, \forall i < i_0 \; s_i = t_i \,, \; s_{i_0} <_{lpo} t_{i_0} \text{ and } \forall i > i_0 \; s_i <_{lpo} t \,. \end{cases}$$

Lexicographic path orderings satisfy:

Proposition 4.3 ([KL80]). *Let $(\mathcal{F}, <)$ be a* total *ordered signature; the lexicographic path order $(\mathcal{T}(\mathcal{F}), <_{lpo})$ induced by $(\mathcal{F}, <)$ on ground terms over $\mathcal{F}$ is a* total *order.*

Proof. We prove here the transitivity, the non-reflexivity and the totality of the relation $(\mathcal{T}(\mathcal{F}), <_{lpo})$.

Transitivity. We proceed by lexicographic structural induction on triplets of terms. A triplet of terms (s, t, u) satisfies *transitivity* if the inequalities $s <_{lpo} t <_{lpo} u$ imply the inequality $s <_{lpo} u$. Suppose that, for each triplet smaller in the lexicographical structural induction order than a triplet (s, t, u), transitivity holds, we show that transitivity also holds for (s, t, u). Suppose that $s = f(s_0, \ldots, s_m)$, $t = g(t_0, \ldots, t_n)$ and $u = h(u_0, \ldots, u_p)$ and that the inequalities $s <_{lpo} t <_{lpo} u$ hold; we proceed by case analysis.

If there is an immediate subterm t_j of t such that the inequality $s \leq_{lpo} t_j$ holds, it is easy to establish that the inequality $t_j <_{lpo} u$ holds; so by inductive hypothesis the inequality $s <_{lpo} u$ is satisfied.

If there is an immediate subterm u_k of u such that the inequality $t \leq_{lpo} u_k$ holds, then $s <_{lpo} t \leq_{lpo} u_k$ and so by inductive hypothesis $s <_{lpo} u_k$; hence, by definition, $s <_{lpo} u$.

If the two previous cases do not hold then necessarily the heads of the terms s, t and u satisfy the inequalities $f \leq g \leq h$. It is easy to establish that for each immediate subterm s_i of the term s we have the inequality $s_i <_{lpo} t$, so by inductive hypothesis the inequality $s_i <_{lpo} u$ holds.

- If the heads of the terms s and t satisfy the inequality $f < g$ then the inequality $f < h$ also holds and since for each immediate subterm s_i of s the inequality $s_i <_{lpo} u$ holds, then, by definition, the inequality $s <_{lpo} u$ is satisfied.
- If the heads of the terms t and u satisfy the inequality $g < h$ we proceed as in the former case.
- If the heads of the terms s, t and u satisfy the equalities $f = g = h$ then we have the following lexicographical inequalities:

$$(s_0, \ldots, s_m) <_{lpo}^{lex} (t_0, \ldots, t_n) <_{lpo}^{lex} (u_0, \ldots, u_p)\,;$$

which imply by inductive hypothesis that $(s_0, \ldots, s_m) <_{lpo}^{lex} (u_0, \ldots, u_p)$, and since for each immediate subterm s_i of the term s the inequality $s_i <_{lpo} u$ holds then, by definition, the inequality $s <_{lpo} u$ is satisfied.

Non-reflexivity. We proceed by structural induction on terms. A term t satisfies *non-reflexivity* if the inequality $t <_{lpo} t$ does not hold. We proceed by *reductio ad absurdum*; suppose that each immediate subterm s_i of a term s is non-reflexive and that $s <_{lpo} s$; then we have to discard only the possibility of the existence of an immediate subterm s_i of s such that the inequality $s <_{lpo} s_i$ holds; if this is the case and since by definition $s_i <_{lpo} s$ then by transitivity the subterm s_i also would be non-reflexive.

We notice that this proof of non-reflexivity is not constructive, since it uses the reductio ad absurdum reasoning; for a constructive proof of non-reflexivity, it is enough to prove constructively that the relation $(\mathcal{T}(\mathcal{F}), <_{lpo})$ is well founded, this is done later in the proof of the corollary 4.2 for the particular case of signatures with bounded arities.

Totality. We proceed by lexicographic structural induction on pairs of terms. A pair of terms (s, t) satisfies *totality* if either $s = t$ or one of the inequalities $s <_{lpo} t$ or $t <_{lpo} s$ hold. Suppose that for each pair smaller in the lexicographical structural induction order than a pair (s, t) totality holds, we show that totality also holds for (s, t). Suppose that $s = f(s_0, \ldots, s_m)$ and $t = g(t_0, \ldots, t_n)$ and $s \neq t$; we proceed by case analysis.

If for some immediate subterm t_j of the term t the inequality $s \leq_{lpo} t_j$ holds then by transitivity the inequality $s <_{lpo} t$ also holds.

If for some immediate subterm s_i of the term s the inequality $t \leq_{lpo} s_i$ holds then by transitivity the inequality $t <_{lpo} s$ also holds.

If the previous two cases do not hold then by inductive hypothesis we have, for each immediate subterm t_j of the term t, the inequality $t_j <_{lpo} s$ holds *and*, for each immediate subterm s_i of the term s, the inequality $s_i <_{lpo} t$ holds. Moreover:

- either the inequality $f < g$ holds and then the inequality $s <_{lpo} t$ also holds;
- or the equality $f = g$ holds and since the non-equality $s \neq t$ holds there exists a least coordinate i_0 such that the respective immediate subterms s_{i_0} and t_{i_0} of s and t are different, then, by inductive hypothesis, either the inequality $s_{i_0} <_{lpo} t_{i_0}$ holds and hence the inequality $s <_{lpo} t$ also holds or the inequality $t_{i_0} <_{lpo} s_{i_0}$ and hence $t <_{lpo} s$ also holds. ■

As claimed above the lexicographic path orders satisfy:

Lemma 4.7 (Extension lemma). *Given an ordered signature $(\mathcal{F}, <)$; the lexicographic path order $(\mathcal{T}(\mathcal{F}), <_{lpo})$ on ground terms over $\mathcal{F}$ is an* extension order *of the Kruskal order $(\mathcal{T}(\mathcal{F}), <_K)$ on ground terms over $\mathcal{F}$.*

Proof. We proceed by structural induction on terms. Given a term t we denote by $<_K \subset_t <_{lpo}$ the property asserting that each term s satisfying the inequality $s <_K t$ also satisfies the inequality $s <_{lpo} t$. Given two terms $s = f(s_0, \ldots, s_m)$ and $t = g(t_0, \ldots, t_n)$, suppose that for each immediate subterm t_j of the ground term t the property $<_K \subset_{t_j} <_{lpo}$ is satisfied. We prove that the property $<_K \subset_{t_j} <_{lpo}$ is also satisfied; for this suppose that the inequality $s <_K t$ holds. We prove by cases that the inequality $s <_{lpo} t$ also holds.

If there is some immediate subterm t_j of the term t such that the inequality $s <_K t_j$ holds then, since by inductive hypothesis the property $<_K \subset_{t_j} <_{lpo}$ is satisfied, the inequality $s <_{lpo} t_j$ also holds and so by transitivity the inequality $s <_{lpo} t$ holds.

If the heads of the terms s and t satisfy the inequality $f < g$ and there is a subsequence $(t_{j_0}, \ldots, t_{j_m})$ of immediate subterms of t such that for each coordinate i the immediate subterms s_i and t_{j_i} of the respective terms s and t satisfy the inequality $s_i <_K t_{j_i}$ then by inductive hypothesis the inequalities $s_i <_{lpo} t_{j_i} <_{lpo} t$ also hold and so by transitivity the inequality $s_i <_{lpo} t$ holds; therefore by definition the inequality $s <_{lpo} t$ holds.

If the heads of the terms s and t are equal, if for each coordinate i the immediate subterms s_i and t_i of the respective terms s and t satisfy the inequality $s_i \leq_K t_i$ and if i_0 is the least coordinate such that $s_{i_0} <_K t_{i_0}$, then: $(*)$ the inequality $s_{i_0} <_{lpo} t_{i_0}$ holds by inductive hypothesis, $(**)$ and by inductive hypothesis for each coordinate i the immediate subterm s_i and t_i of the respective terms s and t satisfy the inequality $s_i \leq_{lpo} t_i <_{lpo} t$ and so by transitivity the inequality $s_i <_{lpo} t$ also holds; therefore by definition $s <_{lpo} t$. ∎

As a consequence of Kruskal's tree theorem, the extension lemma and Kamin-Levy's proposition we have:

Corollary 4.2. *Given a* total well founded *ordered signature* $(\mathcal{F}, <)$*; then the lexicographic path order* $<_{lpo}$ *on ground terms over* $\mathcal{F}$ *is also a* total and well founded *order.*

For the sake of completeness (and constructivity) we give a proof of the last corollary for the particular case of signatures with bounded arities. See also [Buch95] for a similar proof.

Proof (*for signatures with bounded arities*). A term t is *well founded* if there is no infinite $<_{lpo}$-decreasing sequences starting with t, in this case the relation $<_{lpo}$ restricted to terms $<_{lpo}$-smaller than t is a well founded total order, we denote by $\delta(t)$ the order type of this set ordered by $<_{lpo}$. A term t is *admissible* if each one of its immediate subterms is well founded. We associate to each admissible ground term $t = g(t_0, \ldots, t_n)$ a rank defined by $\rho(t) = (g, \delta(t_0), \ldots, \delta(t_i))$; we remark that since $(\mathcal{F}, <)$ is a well founded signature with bounded arity, then the class of ranks of admissible terms is also well founded.

We notice first that if each admissible term is well founded and since constants are admissible then each term is well founded (a straightforward proof can be done by structural induction); therefore we prove that each admissible term is well founded by induction on the rank of admissible terms. Given an admissible term $t = g(t_0, \ldots, t_m)$, suppose that each admissible term of rank less than the rank of the term t is well founded, we prove that t is also well founded; for this it is enough to show that each term s satisfying the inequality $s <_{lpo} t$ is well founded; moreover, we prove by structural induction that each subterm s' of s is well founded.

Suppose that $s' = f(s'_0, \ldots, s'_m)$ is a subterm of s such that each immediate subterm s'_i of s' is well founded (if s' is a constant then we convene that $s' = f$); since the inequality $s <_{lpo} t$ holds the inequality $s' <_{lpo} t$ also holds, therefore:

- either there is an immediate subterm t_j of the term t such that the inequality

$s' <_{lpo} t_j$ holds. Since, by admissibility of the term t, the subterm t_j of the term t is well founded, so s' is also well founded;

- or $f < g$ and thus:

$$\begin{array}{rcl}\rho(s') & = & (f, \delta(s'_0), \ldots, \delta(s'_m)) \\ & <^{lex} & (g, \delta(t_0), \ldots, \delta(t_n))\,,\end{array}$$

  so by inductive hypothesis s' is well founded;

- or $f = g$ and for each coordinate i such that $i < i_0$ the equality $s'_i = t_i$ holds, the inequality $s'_{i_0} <_{lpo} t_{i_0}$ holds and for each coordinate i such that $i > i_0$ the inequality $s'_i <_{lpo} t$ holds, thus:

$$\begin{array}{rcl}\rho(s') & = & (f, \delta(t_0), \ldots, \delta(t_{i_0-1}), \delta(s'_{i_0}), \delta(s'_{i_0+1}), \delta(s'_m)) \\ & <^{lex} & (f, \delta(t_0), \ldots, \delta(t_{i_0-1}), \delta(t_{i_0}), \delta(t_{i_0+1}), \delta(t_n))\,,\end{array}$$

  so by inductive hypothesis s' is well founded. (We notice that this case does not happen if s' is a constant.) ■

We state the analogous definitions of Kruskal and lexicographical path orderings for many sorted signatures; Kruskal's Tree Theorem and Kamin & Levy's proposition also hold in this case.

Given an order on ground terms $(\mathcal{T}(\mathcal{F}), <)$ over a (many sorted) signature $\mathcal{F}$, a rewrite system $\mathcal{R}$ over ground terms is *decreasing* if any $\mathcal{R}$-derivation is decreasing with respect to $(\mathcal{T}(\mathcal{F}), <)$. Roughly speaking:

$$\text{if} \quad t \xrightarrow{\mathcal{R}} u \quad \text{then} \quad t > u\,.$$

Therefore, a decreasing rewrite system with respect to a well founded order over ground terms is terminating. The termination of the rewriting systems $\mathcal{R}_{LK_{sp}}$, $\mathcal{R}'_{LK_{sp}}$, and $\mathcal{R}'_{LK}$ can be achieved by lexicographical path orderings. For instance, the rewriting systems $\mathcal{R}'_{LK_{sp}}$, and $\mathcal{R}'_{LK}$ are decreasing with respect to the total and well founded lexicographical path ordering on ground terms over $\mathcal{F}'$ induced by the total order on the signature $\mathcal{F}'$ defined by:

$$1 < s < \alpha < S < L_1 < L_2 < M\,.$$

The termination of the rewriting system $\mathcal{R}'_{LK_{sp}}$ and for each integer k of the rewriting system $\mathcal{R}^k_{LK_{sp}}$ is totally machine checkable. For instance, the rewriting laboratory ORME [Les90] checked that each rule in $\mathcal{R}'_{LK_{sp}}$ is decreasing with respect to the lexicographic path ordering defined in the former paragraph. We remark that the termination of the rewriting system $\mathcal{R}'_{LK}$ interpreting the mix elimination system $\mathcal{E}_{LK}$ for the sequent calculus LK is not totally machine checkable since $\mathcal{R}'_{LK}$ is infinite; nevertheless, the fact that any given rule in $\mathcal{R}'_{LK}$ is decreasing with respect to the lexicographic path ordering defined previously is machine checkable.

In the next sections we give two direct proofs of termination; the first one uses the structural induction approach and the second one uses monotone interpretations on the natural numbers and gives recursive upper bounds for the length of derivations of a term.

4.4. Strong normalization proof by structural induction. In this section we use the notation $t \xrightarrow{\mathcal{R}'} u$ when a proof term t is rewritten to a term u by a rule belonging to the rewrite system $\mathcal{R}'_{LK_{sp}}$. A finite or infinite sequence $\sigma = (t_0, \dots t_i, t_{i+1}, \dots)$ of proof ground terms over $\mathcal{F}'$ is an *$\mathcal{R}'$-derivation* of a proof ground term t_0 if for each pair of immediate terms (t_i, t_{i+1}) in the sequence σ the relation $t_i \xrightarrow{\mathcal{R}'} t_{i+1}$ is satisfied. The set of all $\mathcal{R}'$-derivations of a proof ground term t, denoted by $\Delta(t)$, is called the *$\mathcal{R}'$-derivations tree* of the proof ground term t. The *length* of a finite $\mathcal{R}'$-derivation $\sigma = (t_0, \dots t_n)$ is the natural number n.

A proof ground term t is $\mathcal{R}'$-strongly normalizing, (S.N.), if there is no infinite $\mathcal{R}'$-derivation of t. Note that the $\mathcal{R}'$-derivations tree $\Delta(t)$ is finitely branching. By König's lemma this implies that if if a proof ground term t is S.N. then $\Delta(t)$ is finite; we denote by $\delta(t)$ the least upper bound for the lengths of the derivations of a strongly normalizing term t.

If two proof ground terms t_1 and t_2 are S.N. and n is a natural number ground term, and denoting by $|m|$ and $|u|$ the respective sizes of a natural term m and of a proof term u, the *rank* of the proof ground term $t = M(n, t_1, t_2)$ is defined by:

$$\text{if} \quad t = M(n, t_1, t_2) \qquad \text{then} \quad \rho_t := (|n|, \delta(t_1), \delta(t_2), |t_1|, |t_2|)\,.$$

Proposition 4.4. *If n is a natural number ground term and t_1 and t_2 are two proof ground terms $\mathcal{R}'$-strongly normalizing, then the ground term $M(n, t_1, t_2)$ is also $\mathcal{R}'$-strongly normalizing; roughly speaking:*

If t_1 is S.N. and t_2 is S.N. then $M(n, t_1, t_2)$ is S.N.

Proof. Let $t = M(n, t_1, t_2)$. It is enough to show that if $t \xrightarrow{\mathcal{R}'} t'$ then t' is strongly normalizing. We prove this by induction on the rank ρ_t of the term t.

Initial step - if $\rho_t = (1, 1, 1, 1, 1)$ then the term t satisfies the equality $t = M(1, \alpha, \alpha)$ and has a unique reduction: $M(1, \alpha, \alpha) \xrightarrow{\mathcal{R}'} S(\alpha)$, and $S(\alpha)$ is S.N.

Inductive step - inductive hypothesis: the proposition holds for all terms $u = M(m, u_1, u_2)$ such that $\rho_u <^{lex} \rho_t$, where $<^{lex}$ is the lexicographical order on $(\mathbb{N}^+)^5$.

Reduction of a subterm case - if:

$$t = M(n, t_1, t_2) \xrightarrow{\mathcal{R}'} t' = M(n, t_1', t_2) \qquad \text{with} \qquad t_1 \xrightarrow{\mathcal{R}'} t_1'\,,$$

since t_1 is S.N. then t_1' is S.N. and $\delta(t_1') < \delta(t_1)$; so:

$$\begin{array}{rcl} \rho_{t'} & = & (|n|, \delta(t_1'), \delta(t_2), |t_1'|, |t_2|) \\ & <^{lex} & (|n|, \delta(t_1), \delta(t_2), |t_1|, |t_2|) \\ & = & \rho_t\,, \end{array}$$

thus by inductive hypothesis t' is S.N.

Root reduction case - we consider three generic sub-cases:

Immediate reductions - if $t = M(n, t_1, \alpha)$ and $t = M(n, t_1, \alpha) \xrightarrow{\mathcal{R}'} S(\alpha) = t'$; then t' is S.N.;

Permutation reductions - If $t = M(n, L_1(u_1), t_2)$ and

$$t = M(n, L_1(u_1), t_2) \xrightarrow{\mathcal{R}'} L_1(M(n, u_1, t_2)) = t'\,,$$

it is enough to prove S.N. for $u' = M(n, u_1, t_2)$. Since $\delta(L(u_1)) = \delta(u_1)$ and $|u_1| < |L(u_1)|$ we have:

$$\begin{array}{rcl} \rho_{u'} & = & (|n|, \delta(u_1), \delta(t_2), |u_1|, |t_2|) \\ & <^{lex} & (|n|, \delta(u_1), \delta(t_2), |L_1(u_1)|, |t_2|) \\ & = & \rho_t\,, \end{array}$$

and so by inductive hypothesis $u' = M(n, u_1, t_2)$ is S.N.

Essential reductions - If $t = M(s(m), L_1(u_1), L_1(u_2))$ with:

$$\begin{array}{rcl} t & = & M(s(m), L_1(u_1), L_1(u_2)) \\ & \xrightarrow{\mathcal{R}'} & S(M(m, M(s(m), L_1(u_1), u_2), M(s(m), u_1, L_1(u_2))) \\ & = & t'\,, \end{array}$$

and if we take the notations:

$$\begin{array}{rcl} u'_1 & := & M(s(m), L_1(u_1), u_2)\,, \\ u'_2 & := & M(s(m), u_1, L_1(u_2))\,, \\ u' & := & M(m, u'_1, u'_2)\,, \end{array}$$

then $t' = S(u')$ and t' is S.N. if and only if u' is S.N. but u'_1 is S.N. since:

$$\begin{array}{rcl} \rho_{u'_1} & = & (|s(m)|, \delta(L_1(u_1)), \delta(u_2), |L_1(u_1)|, |u_2|) \\ & <^{lex} & (|s(m)|, \delta(L_1(u_1)), \delta(u_2), |L_1(u_1)|, |L_1(u_2)|) \\ & = & (|s(m)|, \delta(L_1(u_1)), \delta(L_1(u_2)), |L_1(u_1)|, |L_1(u_2)|) \\ & = & \rho_t\,, \end{array}$$

and u'_2 is S.N. by the same arguments; so u' is also S.N. since:

$$\begin{array}{rcl} \rho_{u'} & = & (|m|, \delta(u'_1), \delta(u'_2), |u'_1|, |u'_2|) \\ & <^{lex} & (|s(m)|, \delta(L_1(u_1)), \delta(L_1(u_2)), |L_1(u_1)|, |L_1(u_2)|) \\ & = & \rho_t\,. \ \blacksquare \end{array}$$

We remark that the induction also works on a rank $\bar{\rho}$ defined by:

$$\text{if} \quad t = M(n, t_1, t_2) \qquad \text{then} \quad \bar{\rho}_t := (|n|, \delta(t_1) + \delta(t_2), |t_1| + |t_2|)$$

We notice that, since the $\mathcal{R}'_{LK}$-derivations tree of a term is not finite, the technique used in the former proof does not work for the rewriting system $\mathcal{R}'_{LK}$ which interprets the mix elimination system $\mathcal{E}_{LK}$; nevertheless, the $\mathcal{E}_{LK}$-derivations tree of an LK-proof is finite, and using the same technique as in the former proof we can establish strong normalization of LK-proofs by $\mathcal{E}_{LK}$-mix eliminations.

4.5. Strong normalization proofs by natural interpretations

Introduction. Weiermann has shown in [Wei93] Cichon's claim which states that each finite rewrite system whose termination is proved by a lexicographic path ordering induced by an order over a finite signature yields a subrecursive (multiple recursive) function on the depth of terms which bound the lengths of derivations of terms. A similar and earlier result was achieved by Hofbauer in [Hof92]. These bounding functions are defined recursively in the extended Grzegorczyk-hierarchy; we refer the reader to [Ros84] for a study on subrecursive hierarchies. Weiermann's result can be applied to bound the lengths of ${\mathcal{R}'}_{LK_{sp}}$-derivations since termination of the finite rewriting system ${\mathcal{R}'}_{LK}$ on proof ground terms over the *finite* signature $\mathcal{F}'$ can be proved by a lexicographic path ordering. Our goal is to obtain directly subrecursive upper bounds using monotone algebraic interpretation tools mainly studied by Zantema in [Zan92].

In this section we use the notation $t \overset{\mathcal{R}'}{\longrightarrow} u$ when a proof ground term t over the signature $\mathcal{F}'$ is rewritten to a proof ground term u over the signature $\mathcal{F}'$ by a rule belonging to the rewrite system ${\mathcal{R}'}_{LK_{sp}}$. The $\mathcal{R}'$ *reduction ordering* is the transitive closure of the relation $\overset{\mathcal{R}'}{\longrightarrow}$. We define a homomorphism from the algebra of proof terms over the signature $\mathcal{F}'$ with the $\mathcal{R}'$ reduction ordering to the algebra of totally defined functions on natural numbers with the domination ordering. (We say that a function on natural numbers f, of arity k, *dominates* another function on natural numbers g of the same arity if for each k-tuple of natural numbers $\vec{n}$, the inequality $f(\vec{n}) > g(\vec{n})$ holds). Such homomorphism is said to be a *monotone interpretation on the natural numbers* and a proof of existence of such homomorphism is said to be a termination proof by (monotone) interpretation on natural numbers. In this section we prove termination for the rewrite system $\mathcal{R}'$ by a multiple recursive monotone interpretation.

Natural monotone interpretation for $\mathcal{R}'$. We define in this section a monotone interpretation on the natural numbers denoted by $[\![\cdot]\!]$. This interpretation is from the set of terms over $\mathcal{F}'$ to the set of non-null natural numbers $\mathbb{N}^+$; therefore, if a proof ground term t rewrites by a rule in $\mathcal{R}'$ to a term u, (i.e. $t \overset{\mathcal{R}'}{\longrightarrow} u$), then the inequality $[\![t]\!] > [\![u]\!]$ holds.

Definition 4.4 (strict monotonicity and expansiveness). A function f on natural numbers of arity k is *strictly monotone* with respect to each argument if:

for each pair of k-tuples $(\vec{x}, y_1, \vec{z})$ and $(\vec{x}, y_2, \vec{z})$, if $y_1 < y_2$, then $f(\vec{x}, y_1, \vec{z}) < f(\vec{x}, y_2, \vec{z})$.

A function f on natural numbers of arity k is *expansive* with respect to each argument if:

$$\text{for each } k\text{-tuple } (\vec{x}, y, \vec{z}) , \quad y < f(\vec{x}, y, \vec{z}) .$$

We remark that since the interpretation of each non-constant function is strictly monotone then if a term t rewrites to a term u by a rewrite rule $l \longrightarrow r$ in $\mathcal{R}'$ and if the interpretation of l dominates the interpretation of r, (i.e. the inequality $[\![l]\!] > [\![r]\!]$ holds), then $[\![t]\!] > [\![u]\!]$ holds. Hence, in order to prove the termination of the rewrite system $\mathcal{R}'$ it is enough to prove that for each rewrite rule $l \longrightarrow r$ in $\mathcal{R}'$ the interpretation of l dominates the interpretation of r.

We interpret the non-mix symbols of $\mathcal{F}'$ following the criteria given below:

- The interpretation of natural number ground terms is $[\![1]\!] = 1$, and $[\![s(h)]\!] = [\![h]\!] + 1$.

- The interpretation of the constant α is a constant natural number, for example $[\![\alpha]\!] = 1$.
- The interpretation of unary symbols S and L_1 is a strictly monotone expansive function from $\mathbb{N}^+$ to $\mathbb{N}^+$.
- The interpretation of the binary symbol L_2 is a strictly monotone expansive function with respect to each argument from $\mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$.
- The interpretation of the symbol M is a strictly monotone expansive function with respect to each argument from $\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$.

We take the following monotone interpretations on natural numbers for the non-mix symbol:

$$\begin{array}{rcl} [\![\alpha]\!] & = & 1\,, \\ [\![S(x)]\!] & = & [\![x]\!] + 1\,, \\ [\![L_1(x)]\!] & = & [\![x]\!] + 1\,, \\ \text{and} \quad [\![L_2(x,y)]\!] & = & [\![x]\!] + [\![y]\!]\,. \end{array}$$

It is clear that these interpretations are strictly monotone and expansive with respect to each argument.

Interpretation of the mix symbol M. We interpret the mix symbol M by a strictly monotone and expansive function μ with respect to each argument from $\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$.

Since for each rewrite rule $l \longrightarrow r$ in $\mathcal{R}'$ we must have the dominance property $[\![l]\!] > [\![r]\!]$, the function μ must satisfy the inequalities:

$$\begin{array}{rcl} \mu(h,x,1) & > & 1+1\,, \\ \mu(h,1,y) & > & 1+1\,, \\ \mu\left(h,x+1,y\right) & > & \mu\left(h,x,y\right)+1\,, \\ \mu\left(h,x,y+1\right) & > & \mu\left(h,x,y\right)+1\,, \\ \mu\left(h,x+1,y\right) & > & \mu\left(h,x,y\right)+1\,, \\ \mu\left(h,x,y+1\right) & > & \mu\left(h,x,y\right)+1\,, \\ \mu\left(h,x+y,z\right) & > & \mu(h,x,z)+\mu(h,y,z)\,, \\ \mu\left(h,x,y+z\right) & > & \mu(h,x,y)+\mu(h,x,z)\,, \\ \mu\left(h+1,x+1,y+1\right) & > & \mu\left(h,\mu\left(h+1,x+1,y\right),\mu\left(h+1,x,y+1\right)\right)+1\,, \\ \mu\left(h+1,x+1,y+1\right) & > & \mu\left(h,\mu\left(h+1,x,y+1\right),\mu\left(h+1,x+1,y\right)\right)+1\,, \\ \mu\left(h+1,x+y,z+1\right) & > & \mu\left(h,\mu\left(h+1,x,z+1\right),\mu\left(h+1,x+y,z\right)\right)+1\,, \\ \text{and} \quad \mu\left(h+1,x+y,z+1\right) & > & \mu\left(h,\mu\left(h+1,y,z+1\right),\mu\left(h+1,x+y,z\right)\right)+1\,. \end{array}$$

If we require the function μ to be symmetrical with respect to the second and third arguments (i.e. $\mu(h,x,y) = \mu(h,y,x)$), then the former set of inequalities can be reduced to the inequalities:

$$\begin{array}{rcl} \mu(h,x,1) & > & 1+1\,, \\ \mu\left(h,x+1,y\right) & > & \mu\left(h,x,y\right)+1\,, \end{array}$$

$$\begin{aligned}
\mu\,(h, x+y, z) &> \mu(h, x, z) + \mu(h, y, z)\,,\\
\mu\,(h+1, x+1, y+1) &> \mu\,(h, \mu\,(h+1, x+1, y)\,, \mu\,(h+1, x, y+1)) + 1\,,\\
\mu\,(h+1, x+y, z+1) &> \mu\,(h, \mu\,(h+1, x, z+1)\,, \mu\,(h+1, x+y, z)) + 1\,,\\
\text{and}\quad \mu\,(h+1, x+y, z+1) &> \mu\,(h, \mu\,(h+1, y, z+1)\,, \mu\,(h+1, x+y, z)) + 1\,.
\end{aligned}$$

If we require the function μ to be also strictly monotone and expansive with respect to each argument then, by lemma 4.8 stated below, the former set of inequalities can be reduced to the inequalities:

$$\begin{aligned}
\mu(h, x, 1) &> 1+1\,, &(1)\\
\mu\,(h+1, x+1, y+1) &> \mu\,(h, \mu\,(h+1, x+1, y)\,, \mu\,(h+1, x, y+1)) + 1\,, &(2)\\
\text{and}\quad \mu\,(h, x+y, z) &> \mu(h, x, z) + \mu(h, y, z)\,. &(3)
\end{aligned}$$

Remark 4.2. A monotone interpretation on natural numbers of the mix symbol M for the rewriting system $\mathcal{R}'_{LK}$ for LK-proofs should satisfy the following variant of the inequalities (1) to (3):

$$\begin{aligned}
\mu(h, x, 1) &> 1+x\,,\\
\mu\,(h+1, x+1, y+1) &> \mu\,(h, \mu\,(h+1, x+1, y)\,, \mu\,(h+1, x, y+1)) + x\,,\\
\text{and}\quad \mu\,(h, x+y, z) &> \mu(h, x, z) + \mu(h, y, z)\,.
\end{aligned}$$

Lemma 4.8. *If a function μ from $\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$ is strictly monotone with respect to each argument, symmetrical with respect to the second and third argument and satisfies* (1) *to* (3) *then the following inequalities are satisfied:*

$$\begin{aligned}
\mu\,(h, x+1, y) &> \mu\,(h, x, y) + 1\,,\\
\mu\,(h+1, x+y, z+1) &> \mu\,(h, \mu\,(h+1, x, z+1)\,, \mu\,(h+1, x+y, z)) + 1\,,\\
\textit{and}\quad \mu\,(h+1, x+y, z+1) &> \mu\,(h, \mu\,(h+1, y, z+1)\,, \mu\,(h+1, x+y, z)) + 1\,.
\end{aligned}$$

Proof. By the inequalities (3) and (1) we have:

$$\begin{aligned}
\mu\,(h, x+1, y) &> \mu\,(h, x, y) + \mu\,(h, 1, y)\\
&> \mu\,(h, x, y) + 1\,.
\end{aligned}$$

We prove the inequality:

$$\mu\,(h+1, x+y, z+1) > \mu\,(h, \mu\,(h+1, x, z+1)\,, \mu\,(h+1, x+y, z)) + 1\,,$$

for $y = 1$ and for $y = y' + 1$.

By inequality (2) and symmetry we have:

$$\begin{aligned}
\mu(h+1, x+1, z+1) &> \mu(h, \mu(h+1, x+1, z), \mu(h+1, x, z+1)) + 1\\
&= \mu(h, \mu(h+1, x, z+1), \mu(h+1, x+1, z)) + 1\,.
\end{aligned}$$

By inequality (2), symmetry and expansiveness we have:

$$\begin{aligned}
\mu(h+1, x+y'+1, z+1) &> \mu(h, \mu(h+1, x+y'+1, z), \mu(h+1, x+y', z+1)) + 1\\
&= \mu(h, \mu(h+1, x+y', z+1), \mu(h+1, x+y'+1, z)) + 1\\
&> \mu(h, \mu(h+1, x, z+1), \mu(h+1, x+y'+1, z)) + 1\,. \ \blacksquare
\end{aligned}$$

Thus, in order to obtain a monotone interpretation it is sufficient to define a function μ strictly monotone and expansive with respect to each argument, symmetrical with respect to the second and third argument and such that it satisfies the inequalities (1) to (3).

The function μ. We define the function μ by the following equalities:

$$\begin{array}{rcl} \mu(1,x,y) & := & 3^{x+y}\,, \\ \mu(h,x,1) & := & h\cdot 3^{x+1}\,, \\ \mu(h,1,y) & := & h\cdot 3^{1+y}\,, \\ \text{and}\quad \mu\left(h+1,x+1,y+1\right) & := & 2\cdot\mu\left(h,\mu\left(h+1,x+1,y\right),\mu\left(h+1,x,y+1\right)\right)\,. \end{array}$$

We shall prove step by step that:

1. the function μ is well defined,
2. the function μ is symmetrical with respect to the second and third argument,
3. the function μ is expansive with respect to each argument,
4. the function μ is strictly monotone with respect to each argument,
5. the function μ satisfies the inequalities (1) to (3),
6. and the function μ satisfies the LK variant, stated in the remark 4.2, of the inequalities (1) to (3).

Since the first three equations defining the function μ are overlapping, we prove the:

LEMMA 4.9. *The function μ is well defined for each triplet in $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$.*

PROOF. By induction on the lexicographic order on $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$.

Initial step - The function μ is defined on the triplet $(1,1,1)$ by $\mu(1,1,1)=3^2$.

Inductive step - If for each triplet (h',x',y') such that $(h',x',y')<(h,x,y)$ the value $\mu(h',x',y')$ is well defined, then either:

- if $h=1$, then $\mu(1,x,y)=3^{x+y}$,
- if $h>1$ and $x=1$, then $\mu(h,1,y)=h\cdot 3^{1+y}$,
- if $h>1$ and $y=1$, then $\mu(h,x,1)=h\cdot 3^{x+1}$,
- or if $h>1$, $x>1$ and $y>1$, then $\mu(h,x,y)=2\cdot\mu(h-1,\mu(h,x,y-1),\mu(h,x-1,y))$, which is well defined by lexicographic induction. ∎

LEMMA 4.10. *The function μ is symmetrical with respect to the second and third argument.*

PROOF. By induction on the lexicographic order on $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$.

Initial step - The equation $\mu(1,1,1)=\mu(1,1,1)$ holds.

Inductive step - If for each triplet (h',x',y') such that the inequality $(h',x',y')<^{lex}(h,x,y)$ holds, the equation $\mu(h',x',y')=\mu(h',y',x')$ holds, then either:

- if $h=1$, then $\mu(1,x,y)=3^{x+y}=\mu(1,y,x)$,
- if $h>1$, $x>1$ and $y=1$, then the equation $\mu(h,x,1)=\mu(h,1,x)$ holds by definition,
- if $h>1$, $x=1$ and $y>1$, we proceed as in the preceding case,

- or if $h > 1$, $x > 1$ and $y > 1$, then:

$$\begin{aligned}\mu(h,x,y) &= 2\cdot\mu(h-1,\mu(h,x,y-1),\mu(h,x-1,y))\\ &\overset{I.H.}{=} 2\cdot\mu(h-1,\mu(h,x-1,y),\mu(h,x,y-1))\\ &\overset{I.H.}{=} 2\cdot\mu(h-1,\mu(h,y,x-1),\mu(h,y-1,x))\\ &= \mu(h,y,x)\,. \blacksquare\end{aligned}$$

Lemma 4.11. *The function μ is expansive with respect to each argument.*

Proof. By induction on the lexicographic order on $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$.

Initial step - The inequality $\mu(1,1,1) = 3^2 > 1$ holds.
Inductive step - If for each triplet (h',x',y') such that the inequality $(h',x',y') <^{lex} (h,x,y)$ holds the inequality $\mu(h',x',y') > \max\{h',x',y'\}$ also holds, then either:

- if $h = 1$, then $\mu(1,x,y) = 3^{x+y} > \max\{1,x,y\}$,
- if $h > 1$ and $y = 1$, then $\mu(h,x,1) = h\cdot 3^{x+1} > \max\{h,x,1\}$,
- if $h > 1$ and $x = 1$, then $\mu(h,1,y) = h\cdot 3^{1+y} > \max\{h,1,y\}$,
- or if $h > 1$, $x > 1$ and $y > 1$, then by lexicographic induction:

$$\begin{aligned}\mu(h,x,y) &= 2\cdot\mu(h-1,\mu(h,x,y-1),\mu(h,x-1,y))\\ &\overset{I.H.}{>} 2\cdot\max\{h-1,\mu(h,x,y-1),\mu(h,x-1,y)\}\\ &\overset{I.H.}{>} 2\cdot\max\{h-1,\max\{h,x,y-1\},\max\{h,x-1,y\}\}\\ &> \max\{h,x,y\}\,. \blacksquare\end{aligned}$$

We say that a natural number function μ on $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$ is strictly monotone up to a triplet (h,x,y) if the restriction of the function μ to the cube $\{(h',x',y') \in \mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+ \mid h' \leq h, x' \leq x, y' \leq y\}$ is strictly monotone.

Lemma 4.12. *The function μ is strictly monotone with respect to each argument.*

Proof. By induction on the lexicographic order on $\mathbb{N}^+\times\mathbb{N}^+\times\mathbb{N}^+$.

Initial step - The function μ is monotone up to $(1,1,1)$.
Inductive step - If for each triplet (h',x',y') such that $(h',x',y') < (h,x,y)$ strict monotonicity of μ up to (h',x',y') holds, then either:

- if $h = 1$, then $\mu(1,x,y) = 3^{x+y}$ is monotone up to $(1,x,y)$,
- if $h > 1$ and $y = 1$, then $\mu(h,x,1) = h\cdot 3^{x+1}$ is monotone up to $(h,x,1)$,
- if $h > 1$, $x = 1$ and $y > 1$, we proceed as in the preceding case,
- or if $h > 1$, $x > 1$ and $y > 1$, then by expansiveness:

$$\begin{aligned}\mu(h,x,y) &= 2\cdot\mu(h-1,\mu(h,x,y-1),\mu(h,x-1,y))\\ &> \max\{\mu(h,x,y-1),\mu(h,x-1,y)\}\,;\end{aligned}$$

and by expansiveness, by inductive hypothesis and lexicographic induction:

$$\begin{aligned}\mu(h,x,y) &= 2\cdot\mu(h-1,\mu(h,x,y-1),\mu(h,x-1,y))\\ &>_{I.H} 2\cdot\mu(h-1,x,\mu(h,x-1,y))\\ &\geq_{I.H} 2\cdot\mu(h-1,x,y)\,. \blacksquare\end{aligned}$$

Lemma 4.13. *The function μ satisfies the inequalities* (1) *to* (3).

Proof. By definition the function μ satisfies the inequalities 1 and 2, it remains to prove the inequality $\mu(h, x+y, z) > \mu(h, x, z) + \mu(h, y, z)$; we prove it by case analysis:

- if $h = 1$, then:

$$\begin{array}{rcl} \mu(1, x+y, z) & = & 3^{x+y+z} \\ & > & 3^{x+1} + 3^{y+z} \\ & = & \mu(1, x, z) + \mu(1, y, z)\,, \end{array}$$

- if $h > 1$ and $z = 1$, then:

$$\begin{array}{rcl} \mu(h, x+y, 1) & = & h \cdot 3^{x+y+1} \\ & > & h \cdot 3^{x+z} + h \cdot 3^{y+1} \\ & = & \mu(h, x, 1) + \mu(h, y, 1)\,, \end{array}$$

- if $h > 1$ and $z > 1$, then by expansiveness:

$$\begin{array}{rcl} \mu(h, x+y+1, z) & = & 2 \cdot \mu(h, \mu(h-1, x+y+1, z-1), \mu(h, x+y, z)) \\ & > & \mu(h, x, z) + \mu(h, y+1, z)\,. \blacksquare \end{array}$$

By expansivity of the function μ and lemma 4.13, it is easy to prove that the function μ satisfies the LK variant, stated in remark 4.2, of the inequalities (1) to (3).

Upper bounds for the lengths of derivations. We discuss in this section the accuracy of bounding functions of the lengths of derivations and of the size of normal proofs obtained from the monotone interpretation on natural numbers defined above.

A function λ from $\mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$, on the depth of proofs and the mix degree of proofs, is said to be a *derivations lengths bounding function* if each derivation of a proof of depth d and mix degree n has a length bounded by $\lambda(d, n)$. A function σ from $\mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$, on the depth of proofs and the mix degree of proofs, is said to be a *normal forms sizes bounding function* if each normal form of a proof of depth d and mix degree n has a size bounded by $\sigma(d, n)$.

We consider a function κ from $\mathbb{N}^+ \times \mathbb{N}^+$ to $\mathbb{N}^+$ obtained by the diagonalisation of the function μ with respect to the second and third variable; i.e.: $\kappa(n, m) := \mu(n, m, m)$. For each non-null natural number n we define the first section of the function κ by: $\kappa_n(m) := \kappa(n, m)$. We use exponential notation for the iteration of composition, i.e.: $g^{k+1} := g \circ g^k$.

A useful lemma to bound the length of derivations is:

Lemma 4.14 ([Hof92]). *Given a finite rewrite system $\mathcal{R}$ on a finite signature $\mathcal{F}$ and an interpretation $\mathcal{I}$ of elements of $\mathcal{F}$ such that the interpretation $\mathcal{I}$ prove termination of $\mathcal{R}$. Given $\Phi : \mathbb{N} \longrightarrow \mathbb{N}$ a strictly monotonic function such that for each $f \in \mathcal{F}$ and for every $k \in \mathbb{N}$ the inequality $\mathcal{I}(f)(k, \ldots, k) \leq \Phi(k)$, then for every term t the inequality $\mathcal{I}(t) \leq \Phi^{depth(t)}(1)$.*

We choose an interpretation of proof types decreasing with mix eliminations; so, the lengths of the derivations starting with a proof Π are bounded by the interpretation

$[\![\tau'' \circ \tau(\Pi)]\!]$; but, if the depth of Π is d and the mix degree of Π is n, it is easy to establish, by lemma 4.14, the inequality $[\![\tau'' \circ \tau(\Pi)]\!] < \kappa_n^d(1)$. Therefore:

LEMMA 4.15. *Given a proof* Π*, the depth of which is* d *and the mix degree of which is* n*, the length of the derivations starting with the proof* Π *is bounded by* $\kappa_n^d(1)$.

For each non-null natural number n the function κ_n is primitive recursive; so, for a fixed natural number n the lengths of derivations of proofs of mix degree less than or equal to n is bounded by a primitive recursive function on the depth of proofs. But we claim that the function κ dominates the Ackerman function and hence κ is not primitive recursive. We recall first the definitions of the Kalmar-exponential function and the Ackerman function:

DEFINITION 4.5. The *Kalmar-exponential function* is defined by:

$$\begin{array}{rrcl} & exp(m,0,p) & := & p\,, \\ \text{and} & exp(m,n+1,p) & := & m^{exp(m,n,p)}\,. \end{array}$$

The *Ackerman function* is defined by:

$$\begin{array}{rrcl} & Ack(0,y) & := & y+1\,, \\ & Ack(x+1,0) & := & Ack(x,1)\,, \\ \text{and} & Ack(x+1,y+1) & := & Ack(x,Ack(x+1,y))\,. \end{array}$$

The function κ is not primitive recursive since it satisfies:

LEMMA 4.16. *The function* κ *satisfies the property:*

$$\forall n \in \mathbb{N}^+,\ \forall m \in \mathbb{N}^+, \qquad \kappa(n+2,m+2) \geq Ack(n,m)+2\,.$$

PROOF. We establish first the inequality $(*)$: $\kappa(n+1,m+1) > \kappa(n,\kappa(n+1,m))$. Indeed, by symmetry and monotonicity of μ we have:

$$\begin{array}{rcl} \kappa(n+1,m+1) & = & \mu(n+1,m+1,m+1) \\ & = & 2\mu(n,\mu(n+1,m+1,m),\mu(n+1,m,m+1)) \\ & = & 2\mu(n,\mu(n+1,m+1,m),\mu(n+1,m+1,m)) \\ & > & 2\mu(n,\mu(n+1,m,m),\mu(n+1,m,m)) \\ & = & 2\kappa(n,\kappa(n+1,m))\,. \end{array}$$

We prove the lemma by lexicographical induction on pairs of natural number.

Initial step - By expansivity the inequality $\kappa(2,2) \geq 3$ holds, and $Ack(0,0) = 1$.

Inductive step - We proceed by cases. By expansivity of μ, the inequality $\kappa(2,m+2) \geq m+3$ holds, and $Ack(0,m) = m+1$.

By the inequality $(*)$, expansivity and monotonicity of μ, and inductive hypothesis we have:

$$\begin{array}{rcl} \kappa((n+1)+2,2) & > & \kappa(n+2,\kappa((n+1)+2,1)) \\ & \geq & \kappa(n+2,2+1) \end{array}$$

$$
\begin{aligned}
&\overset{I.H.}{\geq} \quad Ack(n,1)+2 \\
&= \quad Ack(n+1,0)+2\,.
\end{aligned}
$$

By the inequality $(*)$ and inductive hypothesis we have:

$$
\begin{aligned}
\kappa((n+1)+2,(m+1)+2) \quad &> \quad \kappa(n+2,\kappa((n+1)+2,m+2)) \\
&\overset{I.H.}{\geq} \quad \kappa(n+2,Ack(n+1,m)+2) \\
&\overset{I.H.}{\geq} \quad Ack(n,Ack(n+1,m))+2 \\
&= \quad Ack(n+1,m+1)+2\,. \ \blacksquare
\end{aligned}
$$

We notice that, since each mix elimination step increases the size of a proof at most by a factor 2, and since the length of derivations is bounded by $\kappa_n^d(1)$, hence the size of normal forms is bounded by $2^{\kappa_n^d(1)+d+1}$; nevertheless, Schütte proved in [Sch51]

Proposition 4.5. *Each proof, the depth of which is d and the mix degree of which is n, admits a normal proof, the size of which is bounded by the Kalmar-exponential $exp(2,n,d)$.*

We refer the reader to [Per82], [Gir87] and [Gal93] for proofs of similar results. Following [Gen38], [Gir87] and [Gal93], we give hereafter an extension of the mix elimination system $\mathcal{E}_{LK_{sp}}$ and an algorithm to obtain normal forms of proofs.

> We add to the system $\mathcal{E}_{LK_{sp}}$ commutative mix elimination rules interpreted by the following term rewriting rules:
>
> $$\{M_n(M_k(x,y),z) \longrightarrow S(M_k(M_n(x,z),M_n(y,z)))\}_{k<n\in\mathbb{N}}\,;$$
>
> we denote this system by $\mathcal{E}^p_{LK_{sp}}$, since it permutes some mix elimination inferences. *The fast reduction algorithm* is defined as follows:
>
> > At each step, using $\mathcal{E}^p_{LK_{sp}}$ mix elimination rules, replace a minimal subproof containing a mix inference of maximal degree.

In [Gir87], it is provable in primitive recursive arithmetic, that the fast reduction algorithm yields normal forms, the depth of which are a Kalmar-exponential function of the depth and mix degree of the original proofs.

5. Conclusion.

The use of parallel structural rules and symmetrical mix elimination systems allowed us to give simple proofs of strong normalization for sequent calculi. We give hereafter the main goals achieved in this work.

- We adapt Huet's Parallel moves lemma and O'Donnell's lemma from orthogonal rewriting systems theory to orthogonal mix elimination systems.
- We expose classical proofs of the well foundedness of lexicographical path orderings (via Kruskal's Tree Theorem); and we give a direct proof, by transfinite induction, of the well foundedness of lexicographical path orderings (for the case of well founded total signatures with bounded arities). This well foundedness proof technique by transfinite induction can also be applied to other recursive path orderings. We apply

lexicographical path ordering criteria to prove and machine check the termination of a *finite* rewriting system interpreting mix elimination systems.

- We gave a direct proof (via König's lemma) of strong normalization of mix elimination systems, by structural induction on the ranks of proofs we defined.
- We prove strong normalization of mix elimination systems by monotone interpretations on natural numbers, this proof is constructive and finitary. These interpretations are not primitive recursive; thus there remains the question:

> **Open Question:** Is the strong normalization of the mix elimination system $\mathcal{E}_{LK_{sp}}$ provable in primitive recursive arithmetic?

References

[Buch95] W. Buchholz, *Proof-theoretic analysis of termination proofs*, Annals of Pure and Applied Logic 75, 57–65, (1995).

[CRS94] E. A. Cichon, M. Rusinowitch and S. Selhab, *Cut elimination in sequent calculus and rewriting,* Rapport CRIN 94-R-038, (1994).

[CRS96] E. A. Cichon, M. Rusinowitch and S. Selhab, *Cut elimination and rewriting: termination proofs,* Submitted.

[Der82] N. Dershowitz, *Orderings for term rewriting systems,* Theoretical Computer Science, Vol. 17 (5), 279-301, (1982).

[DJ90] N. Dershowitz and J. P. Jouannaud, *Rewrite Systems,* in: Handbook of Theoretical Computer Science, J. Van Leeween (ed.), Vol. B, 243–320, Elsevier, Amsterdam, (1990).

[DJS95] V. Danos, J.-B. Joinet and H. Schellinx, A new deconstructive logic: Linear Logic, Preprint 936, Department of Mathematics, Utrecht University, (1995).

[DP96] R. Dyckhoff and L. Pinto, *Cut-elimination in a permutation-free sequent calculus for intuitionistic logic,* Submitted, (August-1996).

[Dra88] A. G. Dragalin, *Mathematical intuitionism, introduction to proof theory,* Vol. 67 of Translations of Mathematical Monographs, 185–199, American Mathematical Society, (1988), (translation of an article that appeared in 1978).

[Hon96] Hongwei Xi, *On Weak and Strong Normalisations,* Research report 96–189, Dept. of Mathematical Sciences Carnegie Mellon University, (1996).

[Gal91] J. Gallier, *What's so Special about Kruskal's Theorem and the Ordinal* Γ_0 *?, A Survey of Some Results in Proof Theory,* Annals of Pure and Applied Logic, Vol. 53, 199–260, (1991).

[Gal93] J. Gallier, *Constructive logics part I: a tutorial on proof systems and typed λ-calculi,* Theoretical Computer Science, Vol. 110, 249–339, (1993).

[Gen35] G. Gentzen, *Investigations into logical deduction,* in: *Gentzen Collected Works,* E. Szabo (ed.), North Holland, (1969).

[Gen38] G. Gentzen, *New version of the consistency proof of elementary number theory,* in: *Gentzen Collected Works,* E. Szabo (ed.), North Holland, (1969).

[Gir87] J. Y. Girard, *Proof theory and logical complexity,* Bibliopolis, (1987).

[GLT89] J. Y. Girard, Y. Lafont and P. Taylor, *Proofs and types,* Vol. 7 of Cambridge Tracts in Theoretical Computer Science, Cambridge university press, (1989).

[Gra92] B. Gramlich, *Relating innermost, weak, uniform and modular termination of term rewriting systems,* in: International Conf. on Logic Programming and Automated Reasoning, St. Petersburg, A. Voronkov (ed.), Vol. 624, Lecture Notes in Artificial Intelligence, 285–296, Springer, (1992).

[Gro93] Ph. de Groote, *The conservation theorem revisited,* in: International Conf. on Typed Lambda Calculi and applications, Vol. 664, Lecture Notes in Artificial Intelligence, 163–178, Springer, (1993).

[Her95] H. Herbelin, *Séquents qu'on calcule,* Thèse de doctorat, Université Paris VII, (1995).

[HL92] G. Huet and J.-J. Lévy. *Computations In Orthogonal Rewrites Systems I,* in: Computational Logic: Essays in Honour of Alan Robinson, J. Lassez and G. Plotkin (eds.), chapter 11, 395–414, MIT Press, Cambridge, Massachussets. (1992).

[Hof92] D. Hofbauer, *Termination Proofs by Multiset Path Orderings Imply Primitive Recursive Derivation Lengths,* Theoretical Computer Science, Vol. 105 (1), 129–140, (1992).

[Hue80] G. Huet, *Confluent reductions: Abstract properties and applications to term rewriting systems,* J. Assoc. Comput. Mach. Vol. 27 (4), 797–821, (1980).

[Joi93] J. B. Joinet, *Etude de la normalisation du calcul des séquents classique à travers la logique linéaire,* Thèse de doctorat, Université Paris VII, (1993).

[KW94] A. J. Kfoury and J. B. Wells, *New notions of reduction and non-semantic proofs of β-strong normalization in typed λ-calculi,* Technical Report 94–014, Computer Science Department, Boston University, (1994).

[KL80] S. Kamin and J. J. Lévy, *Two generalizations of the recursive path ordering,* unpublished note, Dept. of Computer Science, Univ. of Illinois, Urbana, IL, (1980).

[Klo80] J. W. Klop, *Combinatory reduction systems,* PH.D. thesis, CWI, Amsterdamm Mathematical Center Tracts N^o. 127.

[Klo92] J. W. Klop, *Term Rewriting Systems,* in: Handbook of Logic in Computer Science, S. Abramsky, D. Gabbay, and T. Maibaum (eds.), Vol. 2, chapter 1, 2–117, Clarendon Press, Oxford, (1992).

[Kru60] J. B. Kruskal, *Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture,* Trans. Amer. Society, Vol. 95 210–225, (1960).

[Les90] P. Lescanne, *Implementation of completion by transition rules + control: ORME,* in: Proc. 2nd International Workshop on Algebraic and Logic Programming, Nancy, Vol. 463, Lectures Notes in Computer Science, 262–269, Springer, (1990).

[MZ94] A. Middeldorp and H. Zantema, *Simple Termination Revisited,* in: Proc. of the 12th International Conference on Automated Deduction, CADE'94, Nancy, Lecture Notes in Artificial Intelligence 814, pp. 451–465, (1994).

[Ned73] R. P. Nederpelt, *Strong Normalization for a typed lambda calculus with lambda structured types,* Ph.D. thesis, Tecnische Hogeschool Eindhoven, (1973).

[O'D77] M. J. O'Donnell, *Computing in systems described by equations,* Lecture Notes in Computer Science, Vol. 58. Springer, Berlin, (1977).

[Pab90] J.-F. Pabion, *Cours de logique,* D.E.A. Université Claude-Bernard, Lyon-1, (1990).

[Par92] M. Parigot, *$\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction,* Lecture Notes in Computer Science, Vol. 624, 190–201, Springer, (1992).

[Per82] L. C. P. D. Pereira, *On the estimation of the length of normal derivations,* Philosophical Studies 4, Akademilitteratur, Stockholm (1982).

[Pfe94] F. Pfenning, *A Structural proof of cut elimination and its representation in a logical* *framework,* report CMU-CS-94-218, Carnegie Mellon University, (1994).

[Pin93] L. PINTO, *Cut formulae and logic programming,* in: Extensions of Logic Programming, R. Dyckhoff (ed.), Lecture Notes in Artificial Intelligence, Vol. 798, 282–300, Springer, (1994).

[Pra65] D. PRAWITZ, *Natural deduction, a proof-theoretical study,* Almquist & Wiskell, Stockholm (1965).

[Ros84] H. E. ROSE, Subrecursion: functions and hierarchies, Clarendon Press, Oxford, (1984).

[Ros73] B.K. ROSEN, *Tree-manipulating systems and Church-Rosser theorems,* Journal of the ACM, Vol. 20, 160–187, (1973).

[Sch51] K. SCHÜTTE, *Beweistheoretische Erfassung der Unendlichen Induktion in der Zahlentheorie,* Matematische Annalen, Vol. 122, 369–380, (1951).

[Tah92] E. TAHHAN BITTAR, *Gentzen cut elimination for propositional sequent calculus by rewriting derivations,* preprint Laboratoire de Logique, d'Algorithmique et d'Informatique de Clermont 1, N^o. 16, (1992).

[Tah94] E. TAHHAN BITTAR, *Bornes recursives pour la terminaison d'algorithmes,* thèse de doctorat, Université Lyon1, (1994).

[Wei93] A. WEIERMANN, *Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths,* Theoretical Computer Science, Vol. 139(1&2), 355–362, (1995)

[Wei94] A. WEIERMANN, *Complexity Bounds for Some Finite Forms of Kruskal's Theorem,* Journal of Symbolic Computation, Vol. 18 (5), 489–495, (1994).

[Zan92] H. ZANTEMA, *Termination of Term Rewriting by Interpretation,* technical report, Utrecht University, RUU-CS-92-14, (April-1992).