Towards the Automation of the Design of Logic Programming Languages

Tatjana Lutovac *James Harland*

Department of Computer Science
Royal Melbourne Institute of Technology
GPO Box 2476V, Melbourne, 3001, Australia
{tanja, jah} @cs.rmit.edu.au

Abstract

Logic programs consist of formulas of mathematical logic and various proof-theoretic techniques can be used to design and analyse execution models for such programs. We briefly review the main problems, which are questions that are still elusive in the design of logic programming languages, from a proof-theoretic point of view. Existing approaches and analyses which lead to the various languages are all rather sophisticated and involve complex manipulations of proofs. All are designed for analysis on paper by a human and many of them are ripe for automation. We aim to perform the automation of some aspects of proof-theoretic analyses, in order to assist in the design of logic programming languages. In this paper we describe the first steps towards the design of such an automatic analysis tool. We investigate the usage of particular proof manipulations for the analysis of logic programming strategies. We propose a more precise specification of sequent calculi inference rules that we use as a framework for the generalisation and automation-oriented specification of the permutation process. We illustrate different problems and proposed solutions through the examples in various formulations of sequent calculi for classical, intuitionistic, linear, modal, and minimal logic, including single-conclusioned and multiple-conclusioned systems.

Keywords: automation, sequent calculus, inference rules, permutation process, proof search, logic programming languages.

Contents

1 Introduction

It is well known that logic programming (LP) may be thought of as the application of the techniques of mathematical logic to programming tasks. Logic programs may be considered as collections of formulas and their computation may be identified as searching for proofs: given a program $\mathcal{P}$ and a goal G we attempt to satisfy G by searching for a proof of $\mathcal{P} \to G$ using the inference rules of a given logic. In addition, logic programs may be analysed declaratively, *i.e.* as statements of logic, without reference to a sequence of operations, and so are more amenable to program transformation and optimisation than many imperative languages. Also, programs written in LP languages tend to be somewhat more concise, and hence more comprehensible to a human, than the same programs written in an imperative language (such as C or Pascal).

There have been various LP systems implemented (Prolog [44, 30], LO[3], Lolli[27, 28], LinLog[1], λProlog[35], Forum[34], Lygon[47, 49], etc). But still there is no precise concept of what constitutes a LP language as well as what is the essential difference between a LP system and a theorem prover. One proof-theoretic characterisation and a criterion for identifying LP language is formalised in [33] by the identification of *goal-directed proofs* which are usually defined in terms of a *sequent calculus* system of proofs. Essentially, a goal-directed proof is a (cut-free) proof in which a goal is decomposed before the program, and hence the program is not used until the goal is reduced to just an atom. The properties of such proofs have been thoroughly investigated for various fragments of first and higher-order intuitionistic logic (classes of formulas called Horn clauses [30] and hereditary Harrop formulas [22, 33]). Whilst this analysis is quite thorough, there remains the question of how to "lift" the concept of goal-directed proofs to other logics (for example multiple-conclusioned logic, linear, light linear logic, affine, relevant linear logic, etc.) Given the evolution of programming languages towards higher and higher level languages, with a corresponding increase in the computational power of the execution models of these languages, a natural demand is a wide range of expressive logics and more powerful inference facilities in which to write programs.

Generally, the choice of logic/logical fragment is a compromise between the expressiveness of the logic and the difficulty of proof search automation. This compromise affects the ability of the language to express simply and naturally the solution to a given problem. In some cases, we are able to select a logic/logical fragment on the basis of knowing problems which should be solved (such as being able to recursively define a binary tree). Still there is no clear technical statement of why restrictions on some classes of formulas are necessary (and what kind of restrictions), in order to use classes of formulas in logic programs.

The proof search strategy should be *complete* (to always find a proof, if one exists) for a given fragment of logic. For example, the strategy of *uniform proofs* [33] can prove all results for a Horn clause fragment or hereditary Harrop fragment of intuitionistic logic. At the same time, in general, we need complete and tractable classes of proofs with a view to automating and optimising the proof search. The relationship between the logical fragment and the proof strategy is fundamental.

There have been various proof-theoretic techniques, approaches and analyses [1, 2, 13, 23, 28, 32, 34, 38, 47] used to design and analyse LP languages. Many of the existing approaches and analyses which lead to some of the answers are all rather sophisticated and involve complex manipulations of proofs. Many are restricted to particular logic or classes of formulae. Almost all are designed for analysis on paper by a human and many of them are ripe for automation, being formally defined in precise detail, and yet somewhat overwhelming for humans. A general answer and formal definition requires a thorough understanding of all the existing LP systems, as well as of the relationship between theorem proving and LP and, in particular, how to use proof-theoretic techniques to design and analyse execution models for LP languages.

We present initial work on the problem of automating the analyses and design of LP languages. We propose a more precise specification of sequent calculi inference rules that we use for the generalisation and automation-oriented specification of permutation process. We illustrate different problems and proposed solutions through the examples in various formulations of sequent calculi for classical, intuitionistic, linear, modal, minimal logic, including single-conclusioned and multiple-conclusioned systems.

This paper is structured as follows: in the rest of this Section we summarise the aims of the research, as well as the notation that we use. In Section 2 we point out the first steps in developing machine representation of proofs and strategies. The most important proof manipulations are illustrated in Section 3. In Section 4 we present our more precise specification of sequent calculi inference rules that we use for the generalisation and automation-oriented specification of the permutation process (in Sections 5 and 6). An application of the permutation process and the permutation analysis is given in Section 7. Conclusions and further work are presented in Section 8.

1.1 Aims and expected outcomes

The design of new LP languages demands detailed and systematic analyses of existing LP proof systems and precise answers to the questions:

What class of formulae can be used in programs?

What strategy can be applied?

What is the influence and the actual impact of selected choices [48] on the characteristics of the corresponding LP language (for example on expressibility, efficiency, supporting modular programming, abstract data types, higher-order functions, simple operational behaviour, providing proofs as *answers* as well as all possible answer substitutions, etc)?

Many of the existing analyses involve some complex manipulations of proofs (see Section 3). While the manipulations are simple enough in principle they are often somewhat overwhelming, particularly when there is a large number of rules, as in linear logic[14]. We plan to perform a systematic study of existing approaches, solutions, strategies and to investigate ways of modifying them to make them more suitable for automated analysis and proof-search. The plan is to design a system, an *automatic analyser,* ie a *prototype language compiler generator* which will be able to:

- *recognise the appropriate logical fragment which may be used as a basis for a logic programming language;*
- *identify what combinations of inference rules are necessary for the execution of a given fragment;*
- *recognise (and perhaps define) proofs which conform to a given strategy;*
- *identify fragments for which a given strategy is complete.*

Many implementations of LP languages have many features in common, such as the use of unification and some form of the resolution rule. The analyser would be a first step towards providing a library of tools such as unification methods and algorithms, top-down and bottom-up proof search procedures and other implementation techniques. But it would be more than just an automatic support tool for the proof-theoretic analyses. It should be an *adaptive system* capable of systematically analysing a given set of inference rules in order to extract and to determine the most appropriate and optimised way to assemble the various pieces into a LP system. It is expected that such systems in the future will be able to determine the appropriately optimised execution model.

Clearly an automated analyser would be an important and convenient tool for proof theorists, useful for the implementation and optimisation of LP languages, improving the programming productivity of implementors of LP languages and allowing very rapid prototyping. It would also represent a good use of computer technology *i.e.* using machines to handle lengthy and intricate instructions, but in a formalism and framework which is designed and intended for humans.

1.2 A brief overview of some important details

We point out some important concepts which must be considered and taken into account in order to contribute to the outlined aims:

1. finding an appropriate formalism for representing proofs in various logics (see Section 2),

2. investigation[1] of different proof systems, which include :

- characteristics of inference rules (such as subformula property, invertability, permutability, non-determinism, types of restrictions for a rule's application, cut elimination property, completeness, soundness of a given set of rules, equivalence between given sets of inference rules, etc.) and extracting the generic properties of inference rules rather than particular instances of them,
- the translations of inference rules and proofs between different proof systems. For example, from natural deduction to sequent calculus and vice versa [8], or from non-analytic to analytic proof systems [39], or from matrix proofs to sequent calculus proofs [42], etc. The example below presents a proof in *propositional intuitionistic logic* using (a) natural deduction, (b) resolution, (c) sequent calculus:

$$\dfrac{\dfrac{}{((A\land B)\to B)\lor B, A\land B \vdash ((A\land B)\to B)\lor B}\,Ax \qquad \boxed{Subgoal\ 1} \qquad \dfrac{((A\land B)\to B)\lor B, A\land B, B\vdash B}{((A\land B)\to B)\lor B, A\land B, B\vdash B\lor C}\,\lor intro}{((A\land B)\to B)\lor B, A\land B\vdash B\lor C}\,\lor elim$$

$\boxed{Subgoal\ 1}$

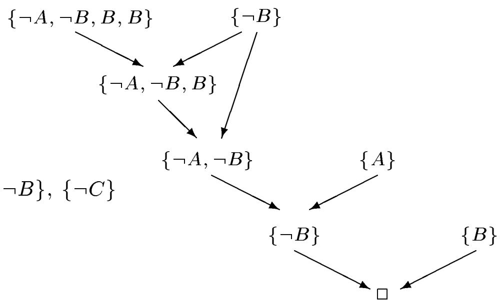
$$\dfrac{\dfrac{\dfrac{}{((A\land B)\to B)\lor B, ((A\land B)\to B), A\land B\vdash (A\land B)\to B}\,Ax \qquad \dfrac{}{((A\land B)\to B)\lor B, A\land B\vdash A\land B}\,Ax}{((A\land B)\to B)\lor B, ((A\land B)\to B), A\land B\vdash B}\,\to elim}{((A\land B)\to B)\lor B, ((A\land B)\to B), A\land B\vdash B\lor C}\,\lor intro$$

(a) proof in natural deduction

assertion: $((A\land B)\to B)\lor B, (A\land B)$

query: $B\lor C$

sets of clauses: $\{\neg A, \neg B, B, B\}$, $\{A\}, \{B\}$, $\{\neg B\}$, $\{\neg C\}$



(b) proof by resolution

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\land B\vdash A\land B}\,Ax \qquad \dfrac{}{B\vdash B}\,Ax}{(A\land B)\to B, A\land B\vdash B}\,\to L \qquad \dfrac{\dfrac{}{B\vdash B}\,Ax}{B, A\land B\vdash B}\,wL}{((A\land B)\to B)\lor B, A\land B\vdash B}\,\lor L}{((A\land B)\to B)\lor B, A\land B\vdash B\lor C}\,\lor R$$

(c) proof in sequent calculus

- methods for introducing new rules (for example as functions from theorems to theorems) or modifying old rules (by merging the existing ones, as shown below by the example from the *sequent calculus* for classical linear logic - see Section 2 and Appendix B),

$$\dfrac{\vdash A,\Gamma \qquad \vdash B,\Delta}{\vdash A\otimes B,\Gamma,\Delta}\,\otimes \;+\; \dfrac{\vdash ?F, ?F, \Sigma}{\vdash ?F, \Sigma}\,?c \;=\; \dfrac{\vdash A,\Gamma, ?\Omega \qquad \vdash B,\Delta, ?\Omega}{\vdash A\otimes B,\Gamma,\Delta, ?\Omega}\,\otimes'$$

- the causes for generating cycles and redundant proofs (explanation and examples are given in part 4. of Section 3) and methods for filtering out redundant structures,

[1] By the term investigation we assume specification and/or generalisation and/or possible automation of a particular problem.

- *meta-level* properties of a proof (such as identification of strategy applied to a given proof and subgoals where more than one strategy is applicable, automating manipulations with, and on a proof (see Section 3), recognising "interesting" and "standard" steps of a proof, as well as families of proofs which contain a common structure, determining the number of witnesses for an existentially quantified (sub)goal, measuring the size of a proof, controlling the resource management in a proof, finding all possible proofs as well as proofs which lead to different answers (different answer substitution), etc.). For example, we present all possible proofs that "formula $\exists x p(x)$ is a logical consequence of the formula $p(a) \wedge p(b)$", by using the *sequent calculus* for classical logic (for the complete list of rules see Appendix B). We classified all proofs in two groups in regard to the answers which proofs provide:

Group 1:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a), p(b) \vdash p(a)}\,\text{w L}}{p(a) \wedge p(b) \vdash p(a)}\,\wedge L}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\exists R \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a), p(b) \vdash p(a)}\,\text{w L}}{p(a), p(b) \vdash \exists x p(x)}\,\exists R}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\wedge L \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a) \vdash \exists x p(x)}\,\exists R}{p(a), p(b) \vdash \exists x p(x)}\,\text{w L}}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\wedge L$$

Group 2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(b) \vdash p(b)}\,Ax}{p(a), p(b) \vdash p(b)}\,\text{w L}}{p(a) \wedge p(b) \vdash p(b)}\,\wedge L}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\exists R \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p(b) \vdash p(b)}\,Ax}{p(a), p(b) \vdash p(b)}\,\text{w L}}{p(a), p(b) \vdash \exists x p(x)}\,\exists R}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\wedge L \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p(b) \vdash p(b)}\,Ax}{p(b) \vdash \exists x p(x)}\,\exists R}{p(a), p(b) \vdash \exists x p(x)}\,\text{w L}}{p(a) \wedge p(b) \vdash \exists x p(x)}\,\wedge L$$

3. investigation of strategies, which includes investigating:
- generic properties of various search strategies in various logic/logic fragments,
- a representation of search strategies,
- methods of adapting proofs to obey certain strategies or identifying cases where this is not possible,
- tools for generalisations, reusing and possible integration of various strategies into one, optimal system for a given logic/rules,
- estimating the applicability of the strategy, the degree of non-determinism which it contains, even the degree of complexity involved in testing for the applicability of the strategy.

1.3 Notation

We use meta-variables as follows: x, y, z, u for variables, a, b, c, p, q for atoms, P, Q, R for atomic formulas, $\phi, \psi, F, A, B, C, D$ for formulas, $\Gamma, \Delta, \Omega, \ldots$ as well as strings of calligraphic letters for (possibly empty) multisets of formulas, $\Gamma\,\mathbf{,}\,\Delta$ $\Gamma \backslash \Delta$ $\Gamma \cap \Delta$ and $\Gamma \subseteq \Delta$ respectively for the *multiset union*, *multiset difference*, *multiset intersection* and the *multi-subset*, $\emptyset$ for the empty multiset, n, i, j, k for natural numbers (including 0) and $\mathbf{\Lambda}$ for the finite set of connectives of a given logical fragment.

Moreover, the meta-variables can be indexed by integers. Additionally, meta-variables which represent multisets can be indexed and/or prefixed by extra parameters: ${}_{top/constant/atom}\Delta_{forbidden}$ where $forbidden$ is the set of connectives that can not appear in formulas of Δ; *top* is the set of connectives that can appear as top-most connective in formulas of Δ; *constant* is the set of logical constants that can appear *as* formulas of Δ, and *atom* is the indicator whether (*atom* $= 1$) or not (*atom* $= 0$) atomic formulas can appear as formulas in Δ.

Remark: For a given meta-variable: ${}_{top/constant/atom}\Delta_{forbidden}$, it is obvious:

$$top \cap forbidden = \emptyset \quad \text{and} \quad forbidden = \Lambda \Rightarrow top = \emptyset\ .$$

We define the following functions from the set of meta-variables to the sets of their allowed top-most connectives, logical constants and forbidden connectives:

$$TOP : {}_{top/constant/atom}\Delta_{forbidden} \mapsto top$$
$$CONSTANT : {}_{top/constant/atom}\Delta_{forbidden} \mapsto constant$$

$$FORBIDDEN : {}_{top/constant/atom}\Delta_{forbidden} \mapsto forbidden$$

We also define functions mapping a given formula F into its top most connective and the set of connectives that appear in the given formula:

$$Top : F \mapsto \mathbf{\Lambda},$$
$$CONNECTIVES : F \mapsto \mathcal{P}(\mathbf{\Lambda})$$

and functions:

$$atom(F) = \begin{cases} 1 & \text{if } F \text{ is atom} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad const(F) = \begin{cases} 1 & \text{if } F \text{ is logical constant} \\ 0 & \text{otherwise} \end{cases}$$

Finally, if $\Gamma = F_1, F_2, \ldots F_n$ and $\circ$ is a unary logical connective, then $\circ\Gamma = \circ F_1, \circ F_2, \ldots \circ F_n$.

2 First steps

The first task is to select and/or develop an appropriate formalism for the machine representation and generalisation of a proof. There are various systems in which inference rules and proofs may be represented and generated, such as Isabelle [37], λProlog[35], ELF[40], Automath[36], LF[25], Lego[41] and various other sequent calculus, natural deduction [10], typed-terms [10], tableau-based, resolution-based, matrix[46], hybrid proof systems. Each approach has its own advantages as well as limitations.

Analytic proof systems [39] such as *resolution, matrix* or *connection method based* systems are very efficient and suited for automatic finding of proofs. But, the generated proofs are not so close to "natural" mathematical reasoning and such results can not directly be used for the construction of computation/programs from the proof. Having in mind the *computation as proof search* or *logic programs as formulas* paradigms it is desirable and important to present proofs in a way that is at once intuitive, simple to analyse and amenable to automated analysis.

It has been demonstrated that proof systems based on *natural deduction* and *sequent calculus* (such as NuPRL [5], Isabelle[37], Lego[41]) can be used quite successfully but the degree of automatic support is very weak.

Natural deduction [10, 17] provides some advantages as a basis for logic programming [19]: comprehensible proofs that are easy understandable for humans; easy provision of control and debugging facilities ("simply by inspecting the state of a proof construction computation" [19]); formulation of various logics as natural deduction systems. In natural deduction, for single-conclusioned logic (as in intuitionistic logic) a proof is a tree-like structure with none, one or more hypotheses but a single conclusion. With such a proof structure it is uniquely determined what was the last rule used, which is something we could not say if there were several conclusions. There is also multiple-conclusioned natural deduction for some logics such as classical logic and linear logic [4]. Many symmetries of logic are not so obvious in natural deduction form. The only symmetry that the system reflects is the introduction - elimination symmetry. "Also, in natural deduction the verification of the correctness of an inferential step may require a global examination of the deduction. In sequent calculus the verification of the correctness of an inferential step requires only local consideration of the sequents involved. "(Bellin, [4]).

For analysis purposes, it is important to present the proof to the user and to the machine (for the automated analysis) that makes it easy to browse and/or edit the proof. *Curry/Howard terms,* or *formulae-as-types principle* imply proofs in a tree form, with each node *i.e.* inference rule being annotate with "history" *i.e.* terms corresponding to deductions being built. This way provides necessary and valuable information at each node but in a manner that is not easy to browse. The balances between necessary and minimal or optimised amount of information, as well as between helpful and unambiguous as possible notation, are very delicate problems. It is also very important for a proof system to be easy modified by elimination of certain rules, mixing of certain rules and adding new rules. Curry/Howard terms do not allow easy defining (or mixing) and adding of new rules.

It is not at all clear what is the best way to analyse proofs for many properties such as, for example, permutations (see subsection 3.1 and Section 5), *i.e.* rearranging logical inferences so that a different order of rules is used but the same conclusion is reached. In particular, the ability to analyse and manipulate various different sets of inference rules could be somehow essential for the choice of the system.

At first we restrict our attention to sequent calculus ($\mathcal{SC}$) as a relatively satisfactory tool for a representation of a chosen logic and a proof-search process. $\mathcal{SC}$ is a simple formalism that allows a proof search to be easy followed, controlled and debugged. It is at once intuitive, simple to analyse and with some structural advantages from the automation point of view. It enables easy modification of forms of rules. It is also interesting to note that some other proof systems (such as *tableaux systems*) which are used in automated theorem proving present just a specialised $\mathcal{SC}$.

2.1 An introduction to sequent calculus

A *sequent* is a pair denoted by $\Gamma \vdash \Delta$, where both the *antecedent* Γ and the *succedent* Δ are sets (or multisets) of formulas. The intuitive meaning of $\Gamma \vdash \Delta$ is: if all of the formulas in Γ are true, then at least one formula in Δ is true. Sequent calculus rules generally are written as:

$$\frac{sequent_1, sequent_2, \ldots sequent_k}{sequent}\ \text{rule-name} \qquad\qquad k = 0, 1, 2, \ldots$$

The sequents above the line in a rule are called *premises* of the rule and the sequent below the line is called the *conclusion*. The rules with n premises ($n = 0, 1, 2, \ldots$) are called *n-ary* (*axioms*, *unary*, *binary*, ...) rules.

We analyse and take into account various $\mathcal{SC}$ formalisations of various logics that have been developed, including the following (for the complete list of inference rules see the Appendix B):

1. Classical Logic:

1. LK: Two-Sided Gentzen's $\mathcal{SC}$ [10]
2. A One-Sided Formulation [24]
3. C_3: A SC without Structural Rules[2]

2. Intuitionistic Logic:

1. LJ: Gentzen's $\mathcal{SC}$ [10]
2. G_3: A SC without Structural Rules [20]
3. Non-standard SC [42]
4. LJT: Contraction-free Sequent Calculi [7]
5. other variants [42, 46]

3. Classical Linear Logic:

1. CLL: A Two-Sided $\mathcal{SC}$ [14]
2. CLV: A Two-sided Formulation with Zones[2]
3. A One-Sided Formulation [6]
4. One-sided Implicit Weakening Fragment [45]
5. One-sided Formulation without Structural Rules [29]
6. CLL1: One-sided Formulation due to [47]
7. A One-sided Formulation with Zones [1]
8. A One-sided Formulation with Two Zones[2]
9. A Two-sided Formulation due to [12]

4. Intuitionistic Linear Logic:

1. ILL: A $\mathcal{SC}$ [6]
2. JLV: A $\mathcal{SC}$ with Four Zones[2]
3. An specialised Fragment [12]

5. Multiple-conclusioned Intuitionistic Logic:

1. A Standard $\mathcal{SC}$ [42]
2. MLJT: Multiple-conclusioned LJT [7]
3. A variant of multiple-conclusioned intuitionistic logic [46]

6. Minimal Linear Logic:

1. MILL: A $\mathcal{SC}$[2]
2. ILV: A $\mathcal{SC}$ with Three Zones[2]

7. Modal Logic:

1. S4: A $\mathcal{SC}$ for Modal Logic [46]
2. $S4^{\mathcal{S}}$ $\mathcal{SC}$ for Modal Logic [26]
3. $KT^{\mathcal{S}}$ $\mathcal{SC}$ for Modal Logic [26]

8. Light Linear Logic:

1. LLL: A One-sided Formulation [15]

9. A $\mathcal{SC}$ for Logic with Finitely Many Variables [18]

A *proof* of a sequent $\Gamma \vdash \Delta$ is a tree whose nodes are labelled with sequents such that: the root node is labelled with $\Gamma \vdash \Delta$ (which is then called *end-sequent*), the internal nodes are instances of one of inference rules, and the leaf nodes are labelled with axioms [32].

Definition 2.1 Two sets of inference rules Σ_1 and Σ_2 are *equivalent* if for any sequent $\mathcal{S}$, $\mathcal{S}$ is provable in Σ_1 iff $\mathcal{S}$ is provable in Σ_2.

Initially, we will deal with commutative logics (*i.e.* logics with rules for exchanging formulas of sequents). Since the permutations of formulas of sequents are allowed, we will talk about multisets[3] rather than sequences or sets of formulas.

The outlined $\mathcal{SC}$ formalisations are not the only ones. As can be seen, they can be distinguished along a few directions: are they restricted to the left-hand side only, do they incorporate explicit structural rules and how do they control the structural rules, and do they distinguish formula sequences that behave differently (such as *linear* and *non-linear* formulas in linear logic)? As a result, different inference specifications for the same logical fragment can have a different branching factor at a certain search node, different degree of non-determinism, or they are more or less suitable for debugging or automation.

3 Proof Manipulations

As we have already pointed out in subsection 1.1 the design of new LP languages and the automation of such design require analysis of existing LP systems. This process involves the analysis of strategies and logical fragments as well as studying of those characteristics of proofs which are important for the characteristics of the corresponding LP language. *Proof manipulations* present a basis for such analysis. They imply various procedures for modification and adaptation of a proof to different search strategies and to different requirements (such as elimination of redundant steps and loops in a proof, decreasing the number of rules or the amount of branching in a proof, elimination of inference rules with the high level of non-determinism, etc.). In the rest of this Section we briefly describe some interesting proof manipulations.

3.1 Permutations

The *permutation* of two adjacent inference rules of a given proof is reversing their order in the proof but without disturbing the rest of the proof (the part above and below the inferences modulo a duplication of some proof branches and a renaming of certain free variables) as a result of which we get the *equivalent* proof to the given one:

a) ($\mathcal{SC}3.3$)

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash a, c, d, \Gamma\end{matrix} \qquad \begin{matrix}\vdots\\ \vdash b, \Delta\end{matrix}}{\vdash c, d, a \otimes b, \Gamma, \Delta}\,\otimes}{\begin{matrix}\vdash c\wp d, a \otimes b, \Gamma, \Delta\\ \vdots\end{matrix}}\,\wp \quad\Longleftrightarrow\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash c, d, a, \Gamma\end{matrix}}{\vdash c\wp d, a, \Gamma}\,\wp \qquad \begin{matrix}\vdots\\ \vdash b, \Delta\end{matrix}}{\begin{matrix}\vdash c\wp d, a \otimes b, \Gamma, \Delta\\ \vdots\end{matrix}}\,\otimes$$

b) ($\mathcal{SC}2.3$)

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ A(a) \vdash A(a) \vee \exists x A(x)\end{matrix}}{\vdash \neg A(a) \vee A(a) \vee \exists x A(x)}\,\neg}{\begin{matrix}\vdash \neg A(a) \vee \exists x A(x)\\ \vdots\end{matrix}}\,\exists \quad\Longleftrightarrow\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ A(a) \vdash A(a) \vee \exists x A(x)\end{matrix}}{A(a) \vdash \exists A(x)}\,\exists}{\begin{matrix}\vdash \neg A(a) \vee \exists x A(x)\\ \vdots\end{matrix}}\,\neg$$

Permutations are essential proof manipulations. We point out some applications of permutations in Section 7. Here we give a brief overview. Permutations have direct impact on the efficient proof construction in various logics. They allow us to prove that a given search strategy will find all possible

[2] due to Frank Pfenning

[3] for some basic definitions see Appendix A

proofs for a given logic, or to construct an example of a provable sequent which can not be proved with that strategy. It has been shown in [29] how the permutation properties of linear logic can be used to deduce some scooping rules for variable instantiations within a proof. Strategies which "pass" certain permutation tests may be amenable to further analysis, such as minimising the amount of branching in the proof, or delaying certain choices until the optimum amount of information is available. The dynamics of the proof that results from the permutation and strategy analysis clearly has a strong influence on the operational semantics of the resulting language. For example, Lygon[47] is based on the search strategy that *some* permutations of right-hand side rules will lead to a proof, whereas Forum[34] is based on the search strategy that *any* permutation of right-hand rules will lead to a proof. Cut elimination (see next subsection) which is important and non-trivial also involves many tedious permutations and rearranging of inferences in a proof.

3.2 Cut Elimination

The Cut rule is a well known cause for several problems: loops (triggered off by cyclic combinations of Cut and some other rules), permutations (successive applications of the Cut rule can always be permuted) and a high degree of non-determinism. Many logical systems have the Cut elimination property. The proof of this property uses an important and non-trivial construction which involves many tedious permutations and movements of inferences in a proof, as shown below, by the example from $\mathcal{SC}$ 3.3.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp \oplus ?b^\perp}\,\oplus \qquad \dfrac{\dfrac{\dfrac{}{\vdash b, b^\perp}\,Ax}{\vdash b, ?b^\perp}\,?}{\vdash b, a^\perp \oplus ?b^\perp}\,\oplus}{\vdash a \& b, a^\perp \oplus ?b^\perp}\,\& \qquad \dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?f}\,w}{\vdash a, a^\perp \oplus b^\perp, ?f}\,\oplus}{\vdash a^\perp \oplus ?b^\perp, a, ?f}\,cut \quad \longmapsto \quad \dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp \oplus ?b^\perp}\,\oplus \qquad \dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?f}\,w}{\vdash a^\perp \oplus ?b^\perp, a, ?f}\,cut \quad \longmapsto$$

$$\longmapsto \quad \dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \qquad \dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?f}\,w}{\vdash a^\perp, a, ?f}\,cut}{\vdash a^\perp \oplus ?b^\perp, a, ?f}\,\oplus \quad \longmapsto \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \qquad \dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp}\,cut}{\vdash a, a^\perp, ?f}\,w}{\vdash a^\perp \oplus ?b^\perp, a, ?f}\,\oplus \quad \longmapsto \quad \dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?f}\,w}{\vdash a^\perp \oplus ?b^\perp, a, ?f}\,\oplus$$

3.3 Expansion of near proofs or Hypothesis Elimination

This is a transformation of a *near proof* $\mathcal{D}$(derivation where not all leaves are axioms) to a proof which retains (at least relatively) the order of inference rules of $\mathcal{D}$ and whose conclusion is identical to or contains a conclusion of $\mathcal{D}$. This is achieved by the replacement of each non-axiom leaf $\vdash \mathcal{S}$ with a proof of $\vdash \mathcal{S}, \Omega$ (where Ω is possibly empty multiset of formulas dependent on $\mathcal{S}$ and on the structure of inference rules of a given proof system) and then by, eventually, *adaptation* of subproof below $\vdash \mathcal{S}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \qquad \begin{matrix}\nabla \\ \vdash a, b^\perp\end{matrix}}{\vdash a, a^\perp \& b^\perp}\,\&}{\vdash a \oplus b, a^\perp \& b^\perp}\,\oplus}{\vdash (a \oplus b) \wp (a^\perp \& b^\perp)}\,\wp$$

In the resulting proof of the above example the adaptation implies inserting some intermediate steps (inserted inferences are boxed).

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \qquad \vdash a, b^\perp}{\vdash a, a^\perp \& b^\perp}\,\&}{\vdash a \oplus b, a^\perp \& b^\perp}\,\oplus}{\vdash (a \oplus b)\wp(a^\perp \& b^\perp)}\,\wp
\quad \longmapsto \quad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?(a^\perp \otimes b)}\,\boxed{\text{w}} \qquad \dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \quad \dfrac{}{\vdash b, b^\perp}\,Ax}{\vdash a, b^\perp, a^\perp \otimes b}\,\boxed{\otimes}}{\vdash a, b^\perp, ?(a^\perp \otimes b)}\,\boxed{?}}{\vdash a, a^\perp \& b^\perp, ?(a^\perp \otimes b)}\,\&}{\vdash a \oplus b, a^\perp \& b^\perp, ?(a^\perp \otimes b)}\,\oplus}{\vdash (a \oplus b)\wp(a^\perp \& b^\perp), ?(a^\perp \otimes b)}\,\wp
$$

If we allow interpretation of a non-axiom leaf $\vdash \mathcal{S}$ as a hypothesis of, say, a proof $\mathcal{D}$ then the above transformation is known as *hypotheses elimination* [2].

Axioms are absolute truth or static knowledge. *Hypotheses* are auxiliary assumptions or lemmas of a proof which present an extra-logical knowledge in the language of a particular logical system. They are temporary knowledge that is assumed to be true in some stages of the deduction, or belong to a catalogue of already proven results.

There are several potential benefits of allowing hypotheses in proofs. First, for the analysis of meta-properties of proofs it could be useful to reduce the branching factor at some search node or to replace some branches or sub-proofs (whose investigation is not of interest) with sequent(s) that is(are) proven in those branches/sub-proofs. It could be also useful to adapt proof systems to retain the results of a previous proof search and to apply and combine them to a new situation.

The property of whether a given proof can be relieved of hypotheses and what the resulting proof looks like is important. This has a direct impact, for example, on Cut elimination because mostly, in the systems with a Cut elimination property, the Cut can be eliminated from proofs without hypotheses and just reduced to a certain normal form for proofs with hypotheses.

The elimination of a hypothesis $\vdash \mathcal{H}$ from a proof Π can be stated as the problem of searching for a proof (without any hypotheses) of hypothesis $\vdash \mathcal{H}$, which will replace it in Π.

For decidable proof systems it is possible to find, at least, one such proof iff $\vdash \mathcal{H}$ is provable. It seems reasonable to expect that in a decidable proof system every provable hypothesis $\vdash \mathcal{H}$ can be eliminated and replaced by its proof.

The situation is more complicated when a hypothesis $\vdash \mathcal{H}$ (or $\mathcal{H}_2 \vdash \mathcal{H}_1$) is an unprovable sequent, or a given proof system is not decidable. It is interesting, in such situations, to consider the finding of the simplest proof (without hypotheses) whose conclusion incorporates $\mathcal{H}$ $(\mathcal{H}_1, \mathcal{H}_2)$. It is obvious, that this process depends on the hypothesis itself, as well as on the structure of inference rules of a given proof system. Andreoli has shown this on the example of one fragment of propositional classical logic [2].

Developing general methods for hypothesis elimination is research still in progress. It is possible to generalise some dependences and to extract certain, general structures of rules that are *suitable* and *useful* for hypothesis elimination. For example, the structure of weakening rule (in LK), the structure of the mix rule, structure of some multiplicative rules, etc.

Also, given the considerations above we can address another interesting question: what is the minimal extension to provide provability of a given unprovable sequent $\mathcal{S}$ of a given proof system? The example below points out some possible extensions and additions of the unprovable sequent $\vdash p, q$ in different logics:

classical logic	two-sided linear logic	one-sided linear logic
$\dfrac{\dfrac{}{p \vdash p}\,Ax}{p \vdash p, q}\,w$	$\dfrac{\dfrac{}{p \vdash p}\,Ax \quad \dfrac{}{q \vdash q}\,Ax}{p\wp q \vdash p, q}\,\wp \qquad \dfrac{}{\vdash p, q, \top}\,\top \qquad \dfrac{}{0 \vdash p, q}\,0$	$\dfrac{\dfrac{}{\vdash p, p^\perp}\,Ax \quad \dfrac{}{\vdash q, q^\perp}\,Ax}{\vdash p, q, p^\perp \otimes q^\perp}\,\otimes \qquad \dfrac{}{\vdash p, q, \top}\,\top$

3.4 Loops and Redundancy Reduction

This is the construction of a *non-redundant* proof *i.e.* proof with no identical, consecutive, intermediate conclusions. An example of one redundant proof and its non-redundant correction, from $\mathcal{SC}$ 3.5, is given below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash ?((a\oplus b^\perp)\otimes b), a, a^\perp}\,Ax}{\vdash ?((a\oplus b^\perp)\otimes b), a\oplus b^\perp, a^\perp}\,\oplus}{\vdash ?((a\oplus b^\perp)\otimes b), a\oplus b^\perp, a^\perp\oplus b^\perp}\,\oplus \quad \dfrac{\dfrac{}{\vdash ?((a\oplus b^\perp)\otimes b), b, b^\perp}\,Ax}{\vdash ?((a\oplus b^\perp)\otimes b), b, a\oplus b^\perp}\,\oplus}{\vdash ?((a\oplus b^\perp)\otimes b), ((a\oplus b^\perp)\otimes b), a^\perp\oplus b^\perp, a\oplus b^\perp}\,\otimes}{\vdash ?((a\oplus b^\perp)\otimes b), a^\perp\oplus b^\perp, a\oplus b^\perp}\,?D \quad \longmapsto \quad \dfrac{\dfrac{\dfrac{}{\vdash ?((a\oplus b^\perp)\otimes b), a, a^\perp}\,Ax}{\vdash ?((a\oplus b^\perp)\otimes b), a\oplus b^\perp, a^\perp}\,\oplus}{\vdash ?((a\oplus b^\perp)\otimes b), a\oplus b^\perp, a^\perp\oplus b^\perp}\,\oplus$$

3.5 Concatenation of Proofs

A *concatenated* proof is a proof that is a result of concatenation of two given proofs (where hypotheses are allowed). We illustrate that "operation" again on the example from $\mathcal{SC}$ 3.5:

$$\begin{array}{c} \dfrac{\dfrac{\dfrac{}{\vdash A(t), A(t)^\perp}\,Ax \quad \dfrac{}{\vdash A(c), A(c)^\perp}\,Ax}{\vdash A(t), A(c)^\perp, A(t)^\perp\otimes A(c)}\,\otimes}{\vdash A(t), A(c)^\perp, \exists z_1\exists z_2(A(z_1)^\perp\otimes A(z_2))}\,\exists \\ + \\ \dfrac{\dfrac{\vdash A(t), A(c)^\perp}{\vdash \forall y A(y), A(c)^\perp}\,\forall}{\vdash \forall y A(y), \exists x A(x)^\perp}\,\exists \end{array} \quad \longmapsto \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A(t), A(t)^\perp}\,Ax \quad \dfrac{}{\vdash A(c), A(c)^\perp}\,Ax}{\vdash A(t), A(c)^\perp, A(t)^\perp\otimes A(c)}\,\otimes}{\vdash A(t), A(c)^\perp, \exists z_1\exists z_2(A(z_1)^\perp\otimes A(z_2))}\,\exists}{\vdash \forall y A(y), A(c)^\perp, \exists z_1\exists z_2(A(z_1)^\perp\otimes A(z_2))}\,\forall}{\vdash \forall y A(y), \exists x A(x)^\perp, \exists z_1\exists z_2(A(z_1)^\perp\otimes A(z_2))}\,\exists$$

3.6 Some problems

It is our contention that the design of a systematic and automatic analyser for LP languages can be approached by automating manipulations with proofs. It is a necessary step towards the systematic and automatic analysis of LP systems. There are techniques to perform an appropriate analysis and to perform different transformations of a proof but they are all designed for analysis on paper by a human rather than by machine. Also, all existing techniques have been adapted to particular sets of inference rules.

For example, for permutations there is no general framework capable of handling permutabilities of inferences in all systems in Section 2. There are many situations where the permutabilities can not be explained by the existing definitions of active and principal part of a rule and the permutation's conditions on premises of the lower and conclusion of the upper rule [13]. One such situation is illustrated in Subsection 3.1, example b), as well as by the example below, where $\wp'$ presents a mix of $\wp$ and w rules from linear logic. More details are given in subsections 4.1 and 5.3; here we give a brief discussion:

$$\begin{array}{c}\dfrac{\vdash \phi, \psi, \Gamma}{\vdash \phi\wp'\psi, \Gamma, ?\delta}\,\wp' \\ \text{schema-rule}\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash ?A, ?A, B, \Gamma\end{array}}{\vdash ?A, B, \Gamma}\,c?}{\vdash ?A\wp' B, \Gamma, ?A}\,\wp' \quad \Longrightarrow \quad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash ?A, ?A, B, \Gamma\end{array}}{\vdash ?A, ?A\wp' B, \Gamma, ?A}\,\wp'}{\vdash ?A\wp' B, \Gamma, ?A}\,c?$$

The possible permutation of $c?$ and $\wp'$ in the outlined subproof can not be detected using the following specifications ([21] or Definitions 2.1-2.4 of [13]):

- the *active part* of a rule is the multiset of formulas of the premise(s) that do not exist in the conclusion;

- the *principal part* of a rule is the multiset of formulas of the conclusion that did not exist in the premises;

- any pair of permutable rules satisfy *permutation position condition*: "the active part of the lower rule is disjoint of the principal part of the upper rule in the starting subproof" (see Section 3 of [21], or Definition 3.1. and Definition 3.2. of [13]);

- rule $\dfrac{\vdash ?F, ?F, \Gamma}{\vdash ?F, \Gamma}\, c?$ "has two active formulas $?F, ?F$, that are contracted to one formula $?F$" (Definition 2.3 of [13]).

Rules $c?$ and $\wp'$ are permutable although they are not in a permutation position in the starting subproof. Formula $?A$ is at the same time the active and principal formula of the $c?$ rule and an active formula of the $\wp'$ rule. It is 'restored' through the principal part of the $\wp'$ rule, and so it is available again for the $c?$, rule after the inversion.

It is our contention that a more precise specification of the structure of sequent calculus rules is needed.

4 Specifying Sequent Calculus Rules

4.1 Structure of sequent calculus rules

Before proceeding further let us recall that a *schema-rule* is a template for defining a rule and an *inference* is an instance of a schema-rule:

name (type) of rule:	schema-rule:	an inference:
$\vee$	$\dfrac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\, \vee$	$\dfrac{p, \neg C \vdash \forall x A, a \to b}{p, \neg C \vdash (\forall x A) \vee (a \to b)}\, \vee$

Definition 4.1 *The structure of sequent calculus inference rules:*

• An *active formula* of rule I is a formula in a premise, that does not exist in the conclusion. The *active part* of the i-th premise of an inference I (denoted by $\mathcal{A}_I^i$) is the (possibly empty) multiset of its active formulas.

• A *quasi-active formula* of an inference I is a formula occurrence whose presence in a premise is necessary (but not sufficient[4]) for the rule application, and which is copied unchanged into the conclusion of the rule. The *quasi-active part* of the i-th premise of an inference I (denoted by $\mathcal{QA}_I^i$) is the (possibly empty) multiset of its quasi-active formulas.

• A *principal formula* of rule I is a formula occurrence of a conclusion that does not exist in premise(s) and that is a result of the disappearance of some active formulas of I. The *principal part* of an inference I (denoted by $\mathcal{P}_I$) is the (possibly empty) multiset of its principal formulas.

• A formula occurrence of rule I that does not exist in a premise and that is not a principal formula, is called an *extra formula*. The *extra part* of an inference I (denoted by $\mathcal{E}_I$) is the (possibly empty) multiset of its extra formulas.

• The *context* of the i-th premise of an inference I (denoted by $Context_I^i$ or Con_I^i) is the (possibly empty) multiset - complement of its active and quasi-active part.

The above definition is illustrated with some examples in Table 1 on page 16 (note that $A_?$ and $B_?$ denote ?-labels and a constant $\perp$, $\Gamma_l = \Delta, ?\Sigma, ?\Theta_l$ and $\Gamma_r = \Delta, ?\Sigma, ?\Theta_r$ where $?\Theta_l \cap ?\Theta_r = \emptyset$ and

[4]except for the exchange rule

$\Gamma_{lr} = \Delta, ?\Sigma, ?\Theta_l, ?\Theta_r$).

Lemma 4.2 For any two inference rules I and J from a given proof system:

1. $\mathcal{A}_I \cap \mathcal{P}_I = \emptyset$
2. $\mathcal{A}_I \neq \emptyset \not\Rightarrow \mathcal{P}_I \neq \emptyset$
3. $\mathcal{A}_I = \emptyset \Rightarrow \mathcal{P}_I = \emptyset$
4. $\mathcal{A}_I \neq \emptyset \Rightarrow I$ is Cut or $\mathcal{QA}_I \neq \emptyset \vee \mathcal{P}_I \neq \emptyset \vee \mathcal{E}_I \neq \emptyset$
5. $\mathcal{P}_I = \mathcal{P}_J \not\Rightarrow type(I) = type(J)$
6. $\mathcal{A}_I = \emptyset \wedge \mathcal{QA}_I = \emptyset \wedge \mathcal{P}_I = \emptyset \wedge \mathcal{E}_I = \emptyset \Rightarrow I$ is a Mix-rule.

Proof:
1.,3. are direct consequences of definition 4.1
2. consider the Cut rule:

$$\frac{\vdash A, \Gamma \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ Cut$$

4. follows directly from the analysis of existing sets of rules (see Appendix B). A generation of a unary rule I such that $\mathcal{A}_I \neq \emptyset$ and $\mathcal{QA}_I = \emptyset \wedge \mathcal{P}_I = \emptyset \wedge \mathcal{E}_I = \emptyset$ is not acceptable because it would allow the deduction of the empty sequent (by choosing $Context_I$ to be $\emptyset$).
5. consider the following example from $\mathcal{SC}3.4$ [45]:

$$\frac{\vdash ?A, \Gamma \qquad \vdash \Delta}{\vdash ?A \otimes ?B, \Gamma, \Delta}\ \otimes_{?r} \qquad\qquad \frac{\vdash \Gamma \qquad \vdash ?B, \Delta}{\vdash ?A \otimes ?B, \Gamma, \Delta}\ \otimes_{?l}$$

6. Any binary rule of the form: $\dfrac{\Gamma \vdash \Delta \qquad \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$ actually presents Mix-rule. There is no unary rule of the form: $\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta}$. The generation and introduction of such a rule is not acceptable because every instance of such a rule would present a redundant step.

□

Table 1

			$\mathcal{A}^1$	$\mathcal{A}^2$	$\mathcal{QA}^1$	$\mathcal{QA}^2$	$\mathcal{P}$	$\mathcal{E}$	Con^1	Con^2
MLJT	$\dfrac{A,\Gamma\vdash B}{\Gamma\vdash A\supset B,\Delta}\supset r$		A, B				$A\supset B$	Δ	Γ	
LK	$\dfrac{\Gamma\vdash\Delta,A_1,A_2,\Delta'}{\Gamma\vdash\Delta,A_2,A_1,\Delta'}$ ex r				A_1, A_2				Γ,Δ,Δ'	
MLJT	$\dfrac{B,A,\Gamma\vdash\Delta}{A\supset B,A,\Gamma\vdash\Delta}\supset_1 l$		B		A		$A\subset B$		Γ,Δ	
4.3	$\dfrac{\Gamma\vdash G}{G\multimap A,\Gamma\vdash A}\multimap l$		G				$G\multimap A, A$		Γ	
	$\dfrac{\Gamma\vdash\Delta\quad\Gamma'\vdash\Delta'}{\Gamma,\Gamma'\vdash\Delta,\Delta'}\ mix$								Γ,Δ	Γ',Δ'
CLL	$\dfrac{\Gamma\vdash A,\Delta\quad\Gamma'\vdash B,\Delta'}{\Gamma,\Gamma'\vdash A\otimes B,\Delta,\Delta'}\otimes$	$*$	A	B			$A\otimes B$		Γ,Δ	Γ',Δ'
3.4	$\dfrac{\vdash\Gamma\quad\vdash B,\Delta}{\vdash A_?\otimes B,\Gamma,\Delta}\otimes_{?l}$	$*$		B			$A_?\otimes B$		Γ	Δ
5.1	$\dfrac{A\supset B,\Gamma\vdash A,\Delta\quad B,\Gamma\vdash\Delta}{A\supset B,\Gamma\vdash\Delta}\supset l$	$\circ$	A	B	$A\supset B$				Γ,Δ	Γ,Δ
MLJT	$\dfrac{D\supset B,\Gamma\vdash C\supset D\quad B,\Gamma\vdash\Delta}{((C\supset D)\supset B),\Gamma\vdash\Delta}\supset_4 l$	$\triangleleft$	$D\supset B,\ C\supset D$	B			$(C\supset D)\supset B$		Γ	Γ,Δ
G3	$\dfrac{A\to B,\Gamma\vdash A\quad B,A\to B,\Gamma\vdash C}{A\to B,\Gamma\vdash C}\to l$	$\triangleleft$	$A, A\to B$	B		$A\to B$			Γ	Γ, C
3.4	$\dfrac{\vdash\Gamma_l\quad\vdash\Gamma_r}{\vdash A_?\&B_?,\Gamma_{lr}}\&_{?lr}$	$\triangleleft$						$A_?\&B_?$	Γ_l	Γ_r
3.9	$\dfrac{\Gamma_1,!\Gamma\vdash G_1\quad\Gamma_2,!\Gamma\vdash G_2}{\Gamma_1,\Gamma_2,!\Gamma\vdash G_1\otimes G_2}\otimes$	$\triangleleft$	G_1	G_2			$G_1\otimes G_2$		$\Gamma_1,!\Gamma$	$\Gamma_1,!\Gamma$
LK	$\dfrac{A,\Gamma\vdash\Delta\quad B,\Gamma\vdash\Delta}{A\vee B,\Gamma\vdash\Delta}\vee l$	$\circ$	A	B			$A\vee B$		Γ,Δ	Γ,Δ

4.2 Restrictions on rule applications

The application of sequent calculus rules may involve some restrictions on the rule premise and/or conclusion. It is possible to distinguish the following types of restrictions:

α) restrictions on multisets of formulas;

β) restrictions on formulas;

γ) restrictions on variables.

α) Restrictions on multisets of formulas

These restrictions are related to binary rules and present requirements that some parts of the rule's premises and conclusion must be identical. We refer to these restrictions as *equality restrictions.*

According to the context of premises and conclusion, binary rules can be distinguished as: *multiplicative* rules where the context of each premise is copied unchanged into the conclusion (rules denoted by $*$ in the Table 1 on page 13); *strong additive* rules where the context of each premise and of the conclusion are identical (denoted by $\circ$); and *weak additive* rules where some part of the context of premises and of the conclusion are identical (denoted by $\triangleleft$). A few more examples follow:

multiplicative rule: $$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \wp B \vdash \Delta, \Delta'}\ \wp$$

strong additive rule: $$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \& B, \Delta}\ \&$$

weak additive rule: $$\frac{\Gamma_1, !\Gamma \vdash G_1 \quad \Gamma_2, !\Gamma \vdash G_2}{\Gamma_1, \Gamma_2, !\Gamma \vdash G_1 \otimes G_2}\ \otimes'$$

$\beta 1$) Restrictions on structure of formulas

These are requirements for a particular structure of some formulas from a rule's premise and/or conclusion, such as:

a) a particular order of connectives in a formula;

b) a particular top-most connective in a formula (this is a special case of a));

c) a particular formula/subformula should be atomic (again special case of a));

d) a particular arrangement of subformulas of a rule's principal formula(s);

We refer to these restrictions as *structure restrictions.* The examples of each kind of restriction follow (the parts of rules which must have particular structure are boxed).

a)

$$\frac{\boxed{C \supset (D \supset B)}, \Gamma \vdash \Delta}{\boxed{(C \& D) \supset B}, \Gamma \vdash \Delta}\ \supset_2^* \qquad \frac{\boxed{!\Gamma} \vdash A, \boxed{?\Delta}}{\boxed{!\Gamma} \vdash \boxed{!A}, \boxed{?\Delta}}\ !R$$

b)

$$\frac{A, \Gamma \vdash \Delta \qquad B, \Gamma \vdash \Delta}{\boxed{A \vee B}, \Gamma \vdash \Delta}\ \vee$$

c)

$$\frac{B, \boxed{a}, \Gamma \vdash \Delta}{\boxed{a \supset B}, \boxed{\mathrm{a}}, \Gamma \vdash \Delta}\ \supset_1^* \quad \textit{a being atomic}$$

d)

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \boxed{A \oplus B}, \Delta}\ \oplus \qquad \frac{\boxed{C \supset B}, \boxed{D \supset B}, \Gamma \vdash \Delta}{\boxed{(C \vee D) \supset B}, \Gamma \vdash \Delta}\ \supset_3^*$$

Note that restrictions on the structure of formulas are always present in the *Principal* part (if it exists) of any rule.

β2) Restrictions on number of formulas

These restrictions include number of formulas in active, quasi-active, principal or extra part of a rule;

Note that the *Context* part (if it exists) of any rule is free of those restrictions.

γ) Restrictions on variables

These restrictions include:

a) finitely many variables that could be used in a proof [18]

b) the constraint that some variables can not be free in particular parts of a rule. We refer to this restrictions as *eigenvariable restrictions*.

$$\textit{eigenvariable condition:} \quad \frac{\Gamma \vdash A[y/x], \Delta}{\Gamma \vdash \boxed{\forall x.A}, \Delta}\ \forall \qquad \text{y is not free in } \Gamma, \Delta$$

Definition 4.3 (Definition 7.3.1 of [9])

A proof-tree is a *pure-variable proof-tree* iff, for every variable y occurring as the eigenvariable in an application of the rule $\forall$: *right* or $\exists$: left, y does not occur both free and bound in any formula in the proof-tree, and y only occurs in the subtree rooted at the sequent constituting the premise of the rule.

Lemma 4.4 (Lemma 7.3.1 of [9])

In LK, G, LJ proof-systems, every proof of a sequent $\Gamma \vdash \Delta$ can be converted to a pure-variable proof of the same sequent, simply by replacing occurrences of variables with new variables.

Lemma 4.5

For every sequent calculus proof system every proof of a sequent $\Gamma \vdash \Delta$ can be converted to a pure-variable proof of the same sequent, simply by replacing occurrences of variables with new variables.

Proof: For a given sequent calculus proof system, the proof is similar to that of lemma 4.4.

$\square$

In Table 2 we summarise possible types of restrictions for different parts of inference rules. The symbol $\surd\!\!\surd$ (respectively $\surd$, $\bullet$) denotes that appropriate part of a rule *always* (respectively *potentially, never*) has the appropriate type of restrictions.

	PART OF RULE	*RESTRICTIONS* A	B.1	B.2	C
unary rules	*active part*	$\bullet$	$\surd$	$\surd\!\!\surd$	$\surd$
	quasi-active part	$\bullet$	$\surd$	$\surd$	$\surd$
	context	$\bullet$	$\surd$	$\bullet$	$\surd$
	principal part	$\bullet$	$\surd\!\!\surd$	$\surd\!\!\surd$	$\bullet$
	extra part	$\bullet$	$\surd$	$\surd$	$\surd$
binary rules	*active part*	$\surd$	$\surd$	$\surd\!\!\surd$	$\surd$
	quasi-active part	$\surd$	$\surd$	$\surd$	$\surd$
	context	$\surd$	$\surd$	$\bullet$	$\surd$
	principal part	$\bullet$	$\surd\!\!\surd$	$\surd\!\!\surd$	$\surd$
	extra part	$\bullet$	$\surd$	$\surd$	$\surd$

Table 2

In the rest of the paper we will restrict our attention, for obvious simplicity, to one-sided sequent calculus inference rules. Generally, they have the following form:

$$\frac{\vdash \mathcal{A}_I \ ; \ \mathcal{QA}_I \ ; \ Context_I}{\vdash \mathcal{P}_I \ ; \ \mathcal{QA}_I \ ; \ Context_I \ ; \ \mathcal{E}_I}\ I \qquad \frac{\vdash \mathcal{A}_I^1 \ ; \ \mathcal{QA}_I^1 \ ; \ Context_I^1 \qquad \vdash \mathcal{A}_I^2 \ ; \ \mathcal{QA}_I^2 \ ; \ Context_I^2}{\vdash \mathcal{P}_I \ ; \ \mathcal{QA}_I^1, \mathcal{QA}_I^2 \ ; \ Context_I^1, Context_I^2 \ ; \ \mathcal{E}_I}\ I$$

For example, here is the representation of the contraction rule from one-sided CLL:

$$\frac{\vdash ?A, ?A, \Delta}{\vdash ?A, \Delta}\ c? \qquad\qquad \frac{\vdash ?A\ ;\ ?A\ ;\ \Delta}{\vdash\quad ;\ ?A\ ;\ \Delta\ ;}\ c?$$

It is obvious that it is possible to instruct a machine *i.e.* "automated analyser" to detect and recognise restrictions on the application of a certain rule, from a given schema-rule. Thus, the *equality restrictions* can be detected by the comparison of the quasi-active as well as the context part of the rule's premises and the conclusion. The *structural restrictions* can be recognised directly from a given schema-formula. Recall that for schema-multisets we add an extra parameter(s): ${}_{top/constant/atom}\Delta_{forbidden}$ to point out the possible top-most connectives (if any), forbidden connectives, allowed logical constants and atoms that can appear as formulas of Δ, in the instantiations of a given schema-multiset.

Thus, for example, for the schema-rule:

$$\frac{\vdash C \supset B_{\oplus}\ \ ;\ \ ;\ \ \Delta_{\exists}, ?\Gamma, !\Sigma \qquad \vdash a \supset B_{\oplus}\ \ ;\ \ ;\ \ \Theta_{\otimes}, ?\Gamma, !\Sigma}{\vdash (C \wp a) \supset B_{\oplus}\ ;\ \ ;\ \ \Delta_{\exists}, \Theta_{\otimes}, ?\Gamma, !\Sigma\ ;}\ I$$

one valid instantiation would be:

$$\frac{\vdash \bot \supset (a \otimes b)\ \ ;\ \ ;\ \ c, !a \qquad \vdash a \supset (a \otimes b)\ \ ;\ \ ;\ \ a \& (b \oplus \top), !a}{\vdash (\bot \wp a) \supset (a \otimes b)\ ;\ \ ;\ \ c, a \& (b \oplus \top), !a\ ;}\ I$$

Restrictions $\beta 2$ must be formalised directly by the explicit quotation of the requested (number of) formulas in the schema-rule. The *eigenvariable restrictions* require additional information and pointers on the eigenvariable variables and parts of rules in which they can not appear free.

5 Permutations

The *permutation* of two adjacent inference rules of a given proof is reversing their order in the proof but without disturbing the rest of the proof (the part above and below the inferences modulo a duplication of some proof branches and a renaming of certain free variables) as a result of which we get the *equivalent* proof to the given one:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(a)\vdash p(a)}\,Ax}{p(a),p(b)\vdash p(a)}\,\mathrm{w\ L}}{p(a)\wedge p(b)\vdash p(a)}\,\wedge L}{p(a)\wedge p(b)\vdash \exists x p(x)}\,\exists R \quad\Leftrightarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{}{p(a)\vdash p(a)}\,Ax}{p(a),p(b)\vdash p(a)}\,\mathrm{w\ L}}{p(a),p(b)\vdash \exists x p(x)}\,\exists R}{p(a)\wedge p(b)\vdash \exists x p(x)}\,\wedge L \quad\Leftrightarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{}{p(a)\vdash p(a)}\,Ax}{p(a)\vdash \exists x p(x)}\,\exists R}{p(a),p(b)\vdash \exists x p(x)}\,\mathrm{w\ L}}{p(a)\wedge p(b)\vdash \exists x p(x)}\,\wedge L$$

Figure 1. shows the possible permutations of given inference rules I and J.

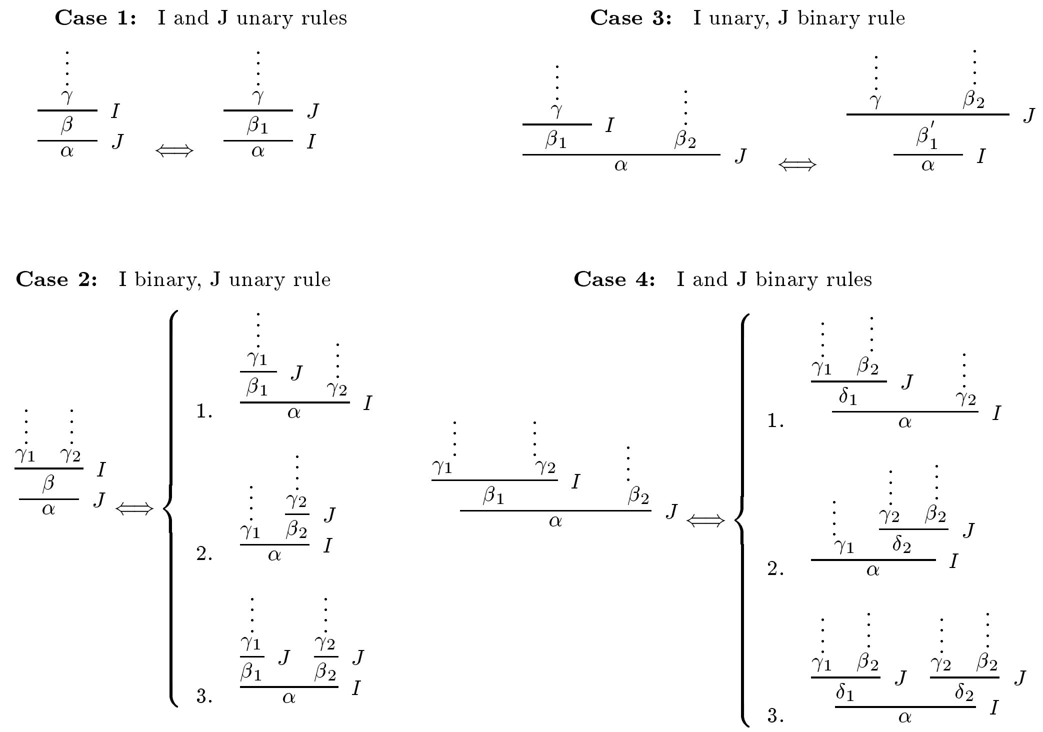


Figure 1.

As we already pointed out at the end of Section 3 the existing definitions and explanations [21, 13] do not cover all possible situations, also they are too implicit from the automation point of view. The proposed specification allows developing more precise and general definitions and notions connected with the inference rules. At first, we develop meta-rules dedicated to systematic and automatic analysis of the permutations of a given set of inference rules.

Remark: In the considerations below for any two adjacent inferences I and J, whenever J is a binary rule and directly follows I we will consider, unless otherwise stated, that I is above the left (*i.e.* first) premise $\vdash \mathcal{A}^1_J, \mathcal{QA}^1_J, Context^1_J,$ of J.

In the rest of this Section we develop more detailed, generalised and automation-oriented specification of a permutation process.

5.1 Basic Definitions

Definition 5.1 *(permutation position)* Two inferences I and J of a given proof Π are:

1. in the *weak permutation position* iff[5]:

i) J follows directly I in Π,

ii) $\mathcal{A}_J, \mathcal{QA}_J \subseteq Context_I, \mathcal{QA}_I$

2. in the *strong permutation position* iff:

$\exists k \in \{1,2\}$

i) J follows directly I in Π,

ii) $\mathcal{A}_J, \mathcal{QA}_J \subseteq Context_I^k, QA_I^k$ and $(\mathcal{A}_J \not\subseteq Context_I^k) \Rightarrow (\mathcal{A}_J \backslash Context_I^k \subseteq \mathcal{E}_J)$

The value $k = 1$ corresponds to *strong-left* while $k = 2$ to *strong-right* permutation position. Note that for a unary rule I the strong permutation position implies the strong-left position ($k = 1$) only.

For example, in the first of the following proofs, the $\otimes$ and $\wp$ rules are in the weak permutation position only, because $\mathcal{A}_\wp, \mathcal{QA}_\wp = \{a^\perp, b^\perp\} \subseteq Context_\otimes, \mathcal{QA}_\otimes = \{a^\perp, b^\perp, c\}$ and the active formulas of the $\wp$ rule (formulas $a^\perp$ and $b^\perp$) belong to different premises of the $\otimes$ rule. In the second proof, the active formulas of the $\wp$ rule belong to same premise of the $\otimes$ rule, and hence the rules are, at the same time, in the weak and strong permutation position.

$$\dfrac{\dfrac{\vdash a, a^\perp \qquad \dfrac{\vdash b, b^\perp}{\vdash ?c, b, b^\perp}\,w?}{\vdash a \otimes b, a^\perp, ?c, b^\perp}\,\otimes}{\vdash b^\perp \wp a^\perp, a \otimes b, ?c}\,\wp \qquad\qquad \dfrac{\dfrac{\vdash a, a^\perp \qquad \dfrac{\vdash b, b^\perp}{\vdash ?c, b, b^\perp}\,w?}{\vdash a \otimes b, a^\perp, ?c, b^\perp}\,\otimes}{\vdash b^\perp \wp ?c, a \otimes b, a^\perp}\,\wp$$

weak perm pos. for $\otimes$ and $\wp$ — strong and weak permutation pos. for $\otimes$ and $\wp$

An explanation of the differences between weak and strong permutation position and their relationship with the existing notion of permutation position are given in the next subsection.

Definition 5.2 *(permutation object/figure, inverted object/figure, permutation result)*

For adjacent inferences I and J (I, L and J where I and L are inferences of the same type) of a given proof Π, being in strong permutation position (each pair I,J and L,J is in strong permutation position):

• The subproofs of Π (if any):

$$(i)\ \begin{matrix} I \\ J \end{matrix} \qquad (ii)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1}}{\beta_1}\,I \quad \dfrac{\overset{\vdots}{\gamma_2}}{\beta_2}\,L}{\alpha}\,J \qquad (iii)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots}{\gamma_2}}{\delta_1}\,I \quad \dfrac{\overset{\vdots}{\beta_1} \quad \overset{\vdots}{\beta_2}}{\delta_2}\,L}{\alpha}\,J \qquad (iv)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots\,\mathcal{D}}{\beta_2}}{\delta_1}\,I \quad \dfrac{\overset{\vdots}{\beta_1} \quad \overset{\vdots\,\mathcal{D}}{\beta_2}}{\delta_2}\,L}{\alpha}\,J$$

we call *permutation object* (*i.e.* I, J permutation object) (i) and *permutation figure* (*i.e.* I, L, J permutation figure) (ii, iii, iv). Note that (iv) is special form of (iii).

• The *inverted object* (*inverted figure*) for I and J is derivation developed from their permutation object (figure) by inverting their order in the following way:

$$(i)\ \begin{matrix} J \\ I \end{matrix} \qquad (ii)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots}{\gamma_2}}{\beta}\,J}{\alpha_1}\,I \qquad (iii)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots}{\beta_1}}{\delta_1}\,J \quad \dfrac{\overset{\vdots}{\gamma_2} \quad \overset{\vdots}{\beta_2}}{\delta_2}\,J}{\alpha_1}\,I \qquad (iv)\ \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots}{\beta_1}}{\beta_1'}\,J \quad \overset{\vdots\,\mathcal{D}}{\beta_2}}{\alpha_1}\,I$$

• The *permutation result* for I and J is inverted object (figure) with the same conclusion and the same premise (modulo a duplication of some proof branches and a renaming of certain free variables) as the permutation object (figure).

[5] if and only if

Above definitions are illustrated, through the examples from one-sided CLL, in the table below:

	perm. object/figure	*perm. position*	*inverted object/figure*	*perm. result*
1)	$\dfrac{\dfrac{\vdash a, b, ?\Gamma}{\vdash a \wp b, ?\Gamma}\,\wp}{\vdash !(a \wp b), ?\Gamma}\,!$	not in permutation position	—	—
2)	$\dfrac{\dfrac{\vdash B, ?\Gamma, A}{\vdash ?B, ?\Gamma, A}\,?}{\vdash !A, ?\Gamma, ?B}\,!$	weak strong-left	does not exist	—
3)	$\dfrac{\dfrac{\vdash p[t/x], a}{\vdash \exists x p(x), a}\,\exists \quad \vdash \exists x p(x), b}{\vdash a \& b, \exists x p(x)}\,\&$	weak strong-left	does not exist	—
4)	$\dfrac{\dfrac{\vdash a, s, m \quad \vdash b, m}{\vdash m, a \otimes b, s, m}\,\otimes}{\vdash m \oplus n, a \otimes b, s, m}\,\oplus$	weak strong-right strong-left	$\dfrac{\dfrac{\vdash a, s, m}{\vdash m \oplus n, s, a}\,\oplus \quad \vdash b, m}{\vdash a \otimes b, s, m \oplus n, m}\,\otimes$	$\surd$
			$\dfrac{\vdash a, s, m \quad \dfrac{\vdash b, m}{\vdash m \oplus n, b}\,\oplus}{\vdash a \otimes b, s, m \oplus n, m}\,\otimes$	$\surd$
			$\dfrac{\dfrac{\vdash a, s, m}{\vdash m \oplus n, s, a}\,\oplus \quad \dfrac{\vdash b, m}{\vdash m \oplus n, b}\,\oplus}{\vdash a \otimes b, s, m \oplus n, m \oplus n}\,\otimes$	$\times$
5)	$\dfrac{\dfrac{\vdash p[t/x], a}{\vdash \exists x p(x), a}\,\exists \quad \dfrac{\vdash p[t/x], b}{\vdash \exists x p(x), b}\,\exists}{\vdash a \& b, \exists x p(x)}\,\&$	weak strong-right strong-left	$\dfrac{\dfrac{\vdash p[t/x], a \quad \vdash p[t/x], b}{\vdash p[t/x], a \& b}\,\&}{\vdash \exists x p(x), a \& b}\,\exists$	$\surd$

Table 3

In the example 2) of the above Table, the presence of linear formulas prevents the ! rule from being applicable on the premise of the ? rule, and hence the inverted object does not exist. In the example 3) the equality restrictions prevent the & rule from being applicable on the premise of the $\exists$ rule. In the example 4) it is possible to invert the $\otimes$ and $\oplus$ rules in three different ways, but only the first two shown inverted objects have the same end-sequent as the permutation object.

Definition 5.3 *(permutability of inferences)*
An inference I is *permutable over* an inference J of a given proof iff they verify the conditions:

- $\boxed{\mathcal{A}}$. I and J are in the strong permutation position,
- $\boxed{\mathcal{B}}$. For their permutation object there exists a permutation result.

In the other cases, we say that I is *not permutable over* J.
In the inverted object and the permutation result we say that inference I is *moved down* and J is *moved up* in regard to their positions in the permutation object.

Intuitively, it is clear that, for a given permutation object $\begin{array}{c}\vdots\,\mathcal{D}\\ I\\ J\end{array}$, the condition $\mathcal{B}$ implies that is possible to reverse the order of I and J, producing the same end-sequent as in the permutation object. $\mathcal{B}$ can be decomposed into the conditions $\mathcal{B}1, \mathcal{B}2$ and $\mathcal{C}$, as follows:

- $\boxed{\mathcal{B}1}$: J is applicable on the appropriate premise(s) of I,

$\boxed{\mathcal{B}2}$: The conclusion-sequent of $\begin{array}{c}\vdots\ \mathcal{D}\\ J\end{array}$ can be a premise of I (*i.e.* meet the restrictions of I),

$\boxed{\mathcal{C}}$: Subproof $\begin{array}{c}\vdots\ \mathcal{D}\\ J\\ I\end{array}$ and the starting permutation object (figure) have the same end-sequent.

Intuitively, the above definitions are easy to understand. In the next subsections we discuss them in more detail and through examples.

5.2 Some characteristic of equivalent proofs modulo inference permutations

In this subsection we give the definition of *equivalent proofs modulo permutations* and point out some relationships between equivalent proofs.

Definition 5.4 ([13])

- A proof $\Pi^{'}$ is *obtained by permutation of I with J in* Π if it is obtained by replacing the permutation object (permutation figure) of I and J (relatively to I and J) by the permutation result modulo a renaming of free variables.
- A proof $\Pi^{'}$ is *equivalent to proof* Π *modulo an inference permutation* (denoted by $\Pi^{'} \sim \Pi$) if there exists two inferences I and J in Π such that $\Pi^{'}$ is obtained by permutation of I with J in Π.
- A proof $\Pi^{'}$ is *equivalent to proof* Π *modulo an inference order* (denoted by $\Pi^{'} \sim^{*} \Pi$) if there exists a finite sequence $\Pi_1, \Pi_2, ..., \Pi_n$ of n proofs ($n \geq 1$) such that:
  1) $\Pi_1 = \Pi$ and $\Pi_n = \Pi^{'}$;
  2) for $i \in [1, n-1]$ $\Pi_i \sim \Pi_{i+1}$

Theorem 5.5 ([13]) The relation $\sim^{*}$ is an equivalence relation.

Let us denote the equivalence class of a proof Π modulo $\sim^{*}$ as $[\Pi]$ and the set of all such equivalence classes as $\Pi/_{\sim^{*}}$.

Lemma 5.6
The number of leaves (axioms and hypothesis) in a given proof-tree π (proof π) is one more than the number of binary nodes of the proof-tree (binary rules of π).

Proof: by induction on the number of nodes (inference rules) in π.

□

Lemma 5.7
1) The number of rules in proofs of $[\Pi]$ is not constant.
2) The number of axioms and hypothesis in proofs of $[\Pi]$ is not constant.

Proof: consequence of duplication of some proof branches during the permutation

□

Lemma 5.8
For any two proofs Π and $\Pi^{'}$ which are equivalent modulo an inference permutation ($\Pi^{'} \sim \Pi$):
1) $|number\_of\_leaves(\Pi) - number\_of\_leaves(\Pi^{'})| \leq 1$;
2) $|number\_of\_rules(\Pi) - number\_of\_rules(\Pi^{'})| \in \{0, 1, k+1\}$, where k is the number of rules in the proof branch which is duplicated during the permutation.

Proof: straightforward

□

5.3 Permutation positions

The strong permutation position takes into account more carefully the distribution of formulas from $\mathcal{A}_J, QA_J$ between the multiset $Context_I, QA_I$. The demand $\mathcal{A}_J, QA_J \subseteq Context_I^k, QA_I^k$ interrupts further analysis in the situation where active and/or quasi-active formulas of J belong to different premises of I. Such a situation can not be "recognised" in the weak permutation position:

i)

$$\dfrac{\dfrac{\vdash a, a^{\perp} \qquad \vdash b, b^{\perp}}{\vdash a \otimes b, a^{\perp}, b^{\perp}}\ \otimes}{\vdash a^{\perp} \wp b^{\perp}, a \otimes b}\ \wp$$

weak ($\not\Rightarrow$ strong) permutation pos.
not permutable rules

ii)

$$\dfrac{\dfrac{\vdash a, a^{\perp} \qquad \dfrac{\vdash b, b^{\perp}}{\vdash ?c, b, b^{\perp}}\ w?}{\vdash a \otimes b, a^{\perp}, ?c, b^{\perp}}\ \otimes}{\vdash b^{\perp} \wp ?c, a \otimes b, a^{\perp}}\ \wp$$

strong ($\Rightarrow$ weak) permutation pos.
permutable rules

iii)

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash A, B, C\end{array}}{\vdash A, B, C \oplus D}\ \oplus}{\vdash A \wp B, C \oplus D}\ \wp$$

strong ($\Leftrightarrow$ weak) permutation pos.
permutable rules

The condition $(\mathcal{A}_J \not\subseteq Context_I^k) \Rightarrow (\mathcal{A}_J \backslash Context_I^k \subseteq \mathcal{E}_J)$ ensures that the situations in which quasi-active formula(s) of upper inference I, which has(have) been "consumed" by lower J (*i.e.* used as the active formulae(s) for J) are acceptable (for the further permutation analysis) iff such formula(s) can be restored again through the extra part $\mathcal{E}_J$ (in that way it(they) remains available for I, after the eventual reversing of the order of I and J). As an example, consider rule $\wp'$ which presents a mix of $\wp$ and w rules from linear logic:

$$\dfrac{\vdash \phi, \psi, \Gamma}{\vdash \phi \wp' \psi, \Gamma, ?\delta}\ \wp'$$

schema-rule

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash ?A, ?A, B, \Gamma\end{array}}{\vdash ?A, B, \Gamma}\ c?}{\vdash ?A \wp' B, \Gamma, ?A}\ \wp' \quad \Longrightarrow \quad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash ?A, ?A, B, \Gamma\end{array}}{\vdash ?A, ?A \wp' B, \Gamma, ?A}\ \wp'}{\vdash ?A \wp' B, \Gamma, ?A}\ c?$$

We could say that the notion of *strong permutation position* and Definition 5.3 are dedicated to the analysis of possible permutations in a given proof *i.e.* when we have the concrete instances of a given inference rule. Such permutations and inference movements in a given proof are essential for rearranging proofs to satisfy or obey particular strategies. It also helps us to detect other proofs of a given sequent which differ only in the order of inference rules.

The notion of *weak permutation position* is suitable for recognition of non-permutable rules in a given proof tree (in regard to part 3. of lemma 5.9 below), as well as for recognition of pairs of schema-rules which represent rules that can never be permutable. Also, because of its simplicity, it is used during the applications of particular strategies for bottom-up construction of proofs (for the choice of formula to be decomposed next). We will discuss this in Section 7.

In [13] Galmiche and Perrier also use the notion of permutation position (Definition 3.1. and Definition 3.2 of [13]) as a precondition for the further checking of permutabilities of given rules. As we already pointed at the end of Section 3 these definitions can not detect and explain some permutabilities. According to our specification of inference rules, Galmiche-Perrier's condition (for permutation position): *"the principal part of upper rule I is disjoint of the active part of the premise of lower rule J"* (condition β) Definition 3.1 of [13]) corresponds to the condition: $\mathcal{A}_J, \mathcal{QA}_J \subseteq Context_I$.

Further in the paper, we will refer to the *Galmiche-Perrier notion of permutation position* as *GP permutation position*. We can say that Galmiche and Perrier are more concerned with permutation analysis on the level of given schema-rules than in the analysis of possible permutabilities directly, in a given proof.

We can establish the following dependencies among the different permutation positions:

Lemma 5.9 For any two adjacent inference rules I and J of a given proof Π:

1. Strong permutation position $\Rightarrow$ weak permutation position.
2. Weak permutation position $\not\Rightarrow$ strong permutation position.
3. I is not in weak permutation position with $J \Rightarrow I$ is not permutable over J.

4. GP permutation position $\Rightarrow$ weak permutation position,
5. GP permutation position $\not\Rightarrow$ strong permutation position,
6. weak permutation position $\not\Rightarrow$ GP permutation position,
7. If $\mathcal{QA}_J = \mathcal{QA}_I = \emptyset$ then weak permutation position $\Leftrightarrow$ GP permutation position,
8. strong permutation position $\not\Rightarrow$ GP permutation position.

Proof:

1., 4., 7. obvious from definition 5.1;

2. see above examples;
3. corollary of 1.;
5. see example i) above;

6., 8. see example at the end of section 3

□

The following lemma give us the necessary condition for the existence of a permutation object/figure:

Lemma 5.10

1. For a I, J permutation object: $Context_J = \mathcal{P}_I, \mathcal{E}_I, (Context_I, \mathcal{QA}_I)\backslash(\mathcal{A}_J, \mathcal{QA}_J)$
2. For a I, I, J permutation figure: $\forall k \in \{1,2\}$ $Context_J^k = \mathcal{P}_I, \mathcal{E}_I, (Context_I, \mathcal{QA}_I)\backslash(\mathcal{A}_J^k, \mathcal{QA}_J^k)$

Proof: straightforward

□

6 Automating Permutations

First steps in the automation of the permutation process include a detailed specification of all pre-conditions ($\mathcal{A}, \mathcal{B}1, \mathcal{B}2$ and $\mathcal{C}$) for a permutation. In our formalisation we strive to be as general as possible.

A specification of conditions $\mathcal{B}1$ and $\mathcal{B}2$ can be based on a certain form of unification. For example, let I and J be in a strong-left permutation position in a given proof:

$$\dfrac{\dfrac{\begin{matrix}\vdots \\ \vdash \Delta_1\end{matrix} \quad \begin{matrix}\vdots \\ \vdash \Delta_2\end{matrix}}{\vdash \Sigma_1}\, I \qquad \begin{matrix}\vdots \\ \vdash \Sigma_2\end{matrix}}{\vdash \Omega}\, J$$

Then, "**J is applicable on the appropriate premise(s) of I**" if there is a distribution of formulas of the multiset Δ_1 between the active, quasi-active and the context part of J *i.e.* if is possible to unify the sequent $\vdash \Delta_1$ with the schema of left premise of J. The naive choice is to try all possible pairs of formulas of the sequents whose unification is considered. This is simple in principle, but implies the details that can often be somewhat overwhelming, particularly in those instances when a large number of formulas is involved.

Also, a specification of one condition can depend on the arity of involved rules, and for binary rules on the type of rules (multiplicative, strong-additive or weak-additive).

For example, for the permutation of a given additive rule I and a unary rule J (producing permutation result of the form F_k $k \in \{1,2\}$), the specification of the condition $\mathcal{B}2$ differs for strong-additive and weak-additive rules I, as shown in Table 4 below. Note that we discuss the outlined specification in the subsection 6.3.2.

Select case:	Specification of the condition $\mathcal{B}2$:
I strong-additive	• $\mathcal{P}_J = \emptyset \ \wedge \ \mathcal{E}_J = \mathcal{A}_J$
I weak-additive	• $\mathcal{LIMIT}_I[\mathcal{P}_J] = \emptyset$ • $\mathcal{LIMIT}_I[\,\mathcal{E}_J \setminus (\mathcal{A}_J \setminus Context_I^k)\,] = \mathcal{A}_J \cap context_I$ • multiset $Cont_I^k \setminus \mathcal{A}_J\ ,\ \mathcal{P}_J, \mathcal{FREE}_I^k[\mathcal{E}_J \setminus (\mathcal{A}_J \setminus Context_I^k)]$ satisfies structure requirements for the schema-multiset $Cont_I^k$

Table 4

Besides formalisation, we try to reduce all the necessary testing in a permutation process on checking of minimal, sufficient conditions. For instance, the analysis shows that, in some cases, the condition $\mathcal{C}$ is fulfilled automatically after the reversal of the rules, and therefore the appropriate checking of this condition is un-necessary in such cases. Thus, for any two unary rules I and J of a given proof, which are in the strong permutation position and which are reversible, the end-sequents of subproofs:

$$\begin{array}{c} \vdots \\ J \\ I \end{array} \qquad\qquad \begin{array}{c} \vdots \\ I \\ J \end{array}$$

are identical. The proof of this claim is given in Theorem 6.2; we illustrate it here, through an example from linear logic:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash A, C, \Sigma\end{array}}{\vdash A \oplus B, C, \Sigma}\,\oplus}{\vdash ?C, A \oplus B, \Sigma}\,? \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash A, C, \Sigma\end{array}}{\vdash A, ?C, \Sigma}\,?}{\vdash A \oplus B, ?C, \Sigma}\,\oplus$$

So, the consideration of arity and types of rules seems reasonable during a formalisation of the permutation process.

Also, the form of the permutation result is another detail which has the influence on the specification of permutation process. The *permutation result* for I and J can have, as Figure 1. on page 17 shows, different forms in regard to possible duplication of some proof branches or duplication of J. This form can be determined in advance, before the direct, final generation of the permutation result. Information about what forms of permutation results are possible for a given pair of rules are valuable in the process such as adapting proofs to obey certain strategies, identifying cases where this is not possible, minimising the amount of branching in the proof, estimating the applicability of the strategy, delaying certain choices until the availability of an optimum amount of information, controlling the degree of non-determinism (if possible, not to duplicate rules or branches with high level of non-determinism).

The specification of the necessary and sufficient conditions for the permutation of given rules, is given in subsection 6.3. Before that, we discuss and generalise some tests.

6.1 Testing strong permutation position

Definition 5.1 provides relatively precise details for an implementation of the condition $\mathcal{A}$. It can be implemented through a decision procedure which: takes as input a given proof Π, the rules in question I and J, and natural number $k \in \{1, 2\}$, and, checks strong permutation position of I and J, corresponding to k:

Test $\boldsymbol{A_{I,J,\Pi}(\,k\,)}$

If J follows directly I in Π **and**

$\mathcal{A}_J, QA_J \subseteq Context_I^k, QA_I^k$ **and**

$(\mathcal{A}_J \not\subseteq Context_I^k) \Rightarrow (\mathcal{A}_J \backslash Context_I^k \subseteq \mathcal{E}_J)$

Then Return true

Else Return false

For rules I and J of a given proof Π, the relations between the condition $\mathcal{A}$ and the test **A** are obvious:

$$A_{I,J,\Pi}(1) \vee A_{I,J,\Pi}(2) \iff \mathcal{A}$$
$$\forall k \in \{1,2\} A_{I,J,\Pi}(k) \iff \text{strong-left and strong-right permutation position}$$
$$\forall k \in \{1,2\} A_{I,J,\Pi}(k) \iff \mathcal{A}$$
$$A_{I,J,\Pi}(1) \implies \mathcal{A}$$
$$A_{I,J,\Pi}(2) \implies \mathcal{A}$$
$$\text{for unary } I: \quad A_{I,J,\Pi}(1) \iff \mathcal{A}$$

6.2 Conditions for Permutation Result

A first step for realizing the automated-oriented formulation of a permutation's conditions involves detailed specification of all changes which may occur during a permutation. The proposed labelling of the structure of sequent calculus inference rules, given in Definition 4.1, enables us to specify part of a rule which can change during a permutation and how it changes:

Proposition 6.1

1) The active, quasi-active, principal and extra part of an inference rule are not changed by its permutation up or down, in a given proof-tree.

2) The context part of an inference can change during inference permutation.

3) Let $\longmapsto$ denote the mapping of the multisets of formula before the permutation that are mapped into the appropriate *Context* part of a rule, after the permutation. Let $\bar{k}$ be defined as $3-k$, and multisets $\mathcal{C}_I$ and $\mathcal{C}_J$ be defined as:

$$\mathcal{C}_I = \begin{cases} \emptyset & \text{if } I \text{ is a strong-additive rule} \\ Context_I^{\bar{k}} & \text{otherwise} \end{cases} \qquad \mathcal{C}_J = \begin{cases} \emptyset & \text{if } J \text{ is a strong-additive rule} \\ Context_J^2 & \text{otherwise} \end{cases}$$

After the permutation, the context of permuted rules I and J are changed as follows:

Case 1: I and J unary rules:

$[Context_I, \mathcal{E}_J] \backslash \mathcal{A}_J \,,\, \mathcal{P}_J \longmapsto Context_I$

$Context_J \backslash [\mathcal{P}_I, \mathcal{E}_I] \,,\, \mathcal{A}_I \longmapsto Context_J$

Case 2: I binary, J unary rule:

$[Context_I^k, \mathcal{E}_J] \backslash \mathcal{A}_J \,,\, \mathcal{P}_J \longmapsto Context_I^k$

$Context_J \backslash [\mathcal{P}_I, \mathcal{E}_I, \mathcal{QA}_I^{\bar{k}}, \mathcal{C}_I] \,,\, \mathcal{A}_I^k \longmapsto Context_J$

Case 3: I unary, J binary rule:

$[Context_I, \mathcal{E}_J] \backslash \mathcal{A}_J^1 \,,\, \mathcal{P}_J \,,\, \mathcal{QA}_J^2, \mathcal{C}_J \longmapsto Context_I$

$Context_J^1 \backslash [\mathcal{P}_I, \mathcal{E}_I] \,,\, \mathcal{A}_I \longmapsto Context_J^1$

Case 4: I and J binary rules:

$[Context_I^k, \mathcal{E}_J] \backslash \mathcal{A}_J^1 \,,\, \mathcal{P}_J \,,\, \mathcal{QA}_J^2 \,,\, \mathcal{C}_J \longmapsto Context_I^k$

$Context_J^1 \backslash [\mathcal{P}_I, \mathcal{E}_I, \mathcal{QA}_I^{\bar{k}}, \mathcal{C}_I] \,,\, \mathcal{A}_I^k \longmapsto Context_J^1$

Proof:

1) Straightforward from the definition of a permutation process.

2) See examples presented in subsection 3.1.

3) The proof of these cases follows the same framework. It is based on a simple inspection of distribution of formulas in the permutation object and the permutation result. We will give the proofs for the cases

1. and 2..

<u>*Remark about the notation:*</u> *In order to point out the distribution of formulas and multisets between the rules in the permutation object and the permutation result we will record sequents of some intermediate results twice. Note that the notation of multisets that appear in the proof is slightly different to the notation introduced in subsection 1.1*

Case 1: The permutation object and permutation result would be respectively:

$$\dfrac{\vdash \mathcal{A}_I \ ; \ \mathcal{QA}_I \ ; \ Context_I \ ;}{\vdash \mathcal{P}_I \ ; \ \underbrace{\mathcal{QA}_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I} \ ; \ \underbrace{Context_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}_I} \ ; \ \mathcal{E}_I}\ I$$

$$\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \ ; \ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \ ; \ \underbrace{\mathcal{C}_I, q\mathcal{A}_I, \mathcal{P}_I, \mathcal{E}_I}_{Context_J} \ ;}{\vdash \mathcal{P}_J \ ; \ \mathcal{QA}'_J, \mathcal{QA}''_J \ ; \ \mathcal{C}_I, q\mathcal{A}_I, \mathcal{P}_I, \mathcal{E}_I \ ; \ \mathcal{E}_J}\ J$$

$$\underbrace{\vdash \mathcal{A}_I \ ; \ \underbrace{\mathcal{QA}_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I} \ ; \ \underbrace{Context_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}_I} \ ;}$$

$$\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \ ; \ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \ ; \ q\mathcal{A}_I, \mathcal{C}_I, \mathcal{A}_I \ ;}{\underbrace{\vdash \mathcal{P}_J \ ; \ \mathcal{QA}'_J, \mathcal{QA}''_J \ ; \ q\mathcal{A}_I, \mathcal{C}_I, \mathcal{A}_I \ ; \ \underbrace{\mathcal{E}_J}_{\mathcal{A}''_J, \mathcal{E}'_J}}}\ J$$

$$\dfrac{\vdash \mathcal{A}_I \ ; \ \underbrace{\mathcal{QA}''_J, \mathcal{A}''_J, q\mathcal{A}_I}_{\mathcal{QA}_I} \ ; \ \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}_I, \mathcal{QA}'_J}_{Context_I} \ ;}{\vdash \mathcal{P}_I \ ; \ \mathcal{QA}''_J, \mathcal{A}''_J, q\mathcal{A}_I \ ; \ \mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}_I, \mathcal{QA}'_J \ ; \ \mathcal{E}_I}\ I$$

Case 2:

$\boxed{\textbf{Subcase 2.A.}}$ I multiplicative binary rule

Assuming the strong-left permutation position, the permutation object and the permutation result would be respectively:

$$\dfrac{\vdash \mathcal{A}^1_I \ ; \ \mathcal{QA}^1_I \ ; \ Context^1_I \ ; \qquad \vdash \mathcal{A}^2_I \ ; \ \mathcal{QA}^2_I \ ; \ Context^2_I \ ;}{\underbrace{\vdash \mathcal{P}_I \ ; \ \underbrace{\mathcal{QA}^1_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}^1_I}, \mathcal{QA}^2_I \ ; \ \underbrace{Context^1_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}^1_I}, Context^2_I \ ; \ \mathcal{E}_I}}\ I$$

$$\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \ ; \ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \ ; \ \underbrace{\mathcal{P}_I, \mathcal{QA}^2_I, q\mathcal{A}^1_I, \mathcal{C}^1_I, Context^2_I, \mathcal{E}_I}_{Context_J} \ ;}{\vdash \mathcal{P}_J \ ; \ \mathcal{QA}'_J, \mathcal{QA}''_J \ ; \ \mathcal{P}_I, \mathcal{QA}^2_I, q\mathcal{A}^1_I, \mathcal{C}^1_I, Context^2_I, \mathcal{E}_I \ ; \ \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}\ J$$

$$\underbrace{\vdash \mathcal{A}^1_I \ ; \ \underbrace{\mathcal{QA}^1_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}^1_I} \ ; \ \underbrace{Context^1_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}^1_I} \ ;}$$

$$\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \ ; \ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \ ; \ \underbrace{\mathcal{A}^1_I, q\mathcal{A}^1_I, \mathcal{C}^1_I}_{Context_J} \ ;}{\underbrace{\vdash \mathcal{P}_J \ ; \ \mathcal{QA}'_J, \mathcal{QA}''_J \ ; \ \mathcal{A}^1_I, q\mathcal{A}^1_I, \mathcal{C}^1_I \ ; \ \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}}\ J$$

$$\dfrac{\vdash \mathcal{A}^1_I \ ; \ \underbrace{\mathcal{QA}''_J, \mathcal{A}''_J, q\mathcal{A}^1_I}_{\mathcal{QA}^1_I} \ ; \ \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}^1_I, \mathcal{QA}'_J}_{Context^1_I} \qquad \vdash \mathcal{A}^2_I \ ; \ \mathcal{QA}^2_I \ ; \ Context^2_I \ ;}{\vdash \mathcal{P}_I \ ; \ \mathcal{QA}^1_I, \mathcal{QA}^2_I \ ; \ Context^1_I, Context^2_I \ ; \ \mathcal{E}_I}\ I$$

$\boxed{\textbf{Subcase 2.B.}}$ I strong additive rule

The permutation object and the permutation result would be respectively:

$$\frac{\vdash \mathcal{A}_I^1 \;;\; qa_I \;;\; context_I \;; \qquad \vdash \mathcal{A}_I^2 \;;\; qa_I \;;\; context_I \;;}{\underbrace{\vdash \mathcal{P}_I \;;\; \underbrace{qa_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I} \;;\; \underbrace{context_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}_I} \;;\; \mathcal{E}_I}}\; I$$

$$\frac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{P}_I, q\mathcal{A}_I, \mathcal{C}_I, \mathcal{E}_I}_{Context_J} \;;}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J, \mathcal{QA}''_J \;;\; \mathcal{P}_I, q\mathcal{A}_I, \mathcal{C}_I, \mathcal{E}_I \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}\; J$$

$$\frac{\dfrac{\underbrace{\vdash \mathcal{A}_I^1 \;;\; qa_I \;;\; context_I \;;}}{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^1, q\mathcal{A}_I, \mathcal{C}_I}_{Context_J}} \qquad \dfrac{\underbrace{\vdash \mathcal{A}_I^2 \;;\; qa_I \;;\; context_I \;;}}{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^2, q\mathcal{A}_I, \mathcal{C}_I}_{Context_J}}}{}$$

$$\frac{\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^1, q\mathcal{A}_I, \mathcal{C}_I}_{Context_J}}{\vdash \mathcal{P}_J \;;\; Context_J \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{A}''_J, \mathcal{E}'_J}}\; J \qquad \dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^2, q\mathcal{A}_I, \mathcal{C}_I}_{Context_J}}{\vdash \mathcal{P}_J \;;\; Context_J \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{A}''_J, \mathcal{E}'_J}}\; J}{}$$

$$\frac{\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I}_{qa_I} \;;\; \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{QA}'_J, \mathcal{C}_I}_{context_I} \;; \qquad \vdash \mathcal{A}_I^2 \;;\; \underbrace{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I}_{qa_I} \;;\; \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{QA}'_J, \mathcal{C}_I}_{context_I} \;;}{\vdash \mathcal{P}_I \;;\; qa_I \;;\; context_I \;;\; \mathcal{E}_I}\; I$$

$\boxed{\textbf{Subcase 2.C.}}$ I weak additive rule

The case when I is a weak additive rule demands a more complicated analysis. Let us denote the conclusion's and premise's part of context which must be identical by $context_I$ and a "free" (unconditional part) part by $Cont_I^k$. So, generally, a weak additive rule I is of the form:

$$\frac{\vdash \mathcal{A}_I^1 \;;\; \mathcal{QA}_I^1 \;;\; context_I, Cont_I^1 \qquad \vdash \mathcal{A}_I^2 \;;\; \mathcal{QA}_I^2 \;;\; context_I, Cont_I^2}{\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I^1, \mathcal{QA}_I^2 \;;\; context_I, Cont_I^1, Cont_I^2 \;;\; \mathcal{E}_I}\; I$$

The permutation object for the strong-left permutation position and the corresponding permutation result would be respectively:

$$\frac{\vdash \mathcal{A}_I^1 \;;\; \mathcal{QA}_I^1 \;;\; context_I, Cont_I^1 \;; \qquad \vdash \mathcal{A}_I^2 \;;\; \mathcal{QA}_I^2 \;;\; context_I, Cont_I^2 \;;}{\underbrace{\vdash \mathcal{P}_I \;;\; \underbrace{\mathcal{QA}_I^1}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I^1}, \mathcal{QA}_I^2 \;;\; \underbrace{context_I, Cont_I^1}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}_I^1}, Cont_I^2 \;;\; \mathcal{E}_I}}\; I$$

$$\frac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{P}_I, \mathcal{QA}_I^2, q\mathcal{A}_I^1, \mathcal{C}_I^1, Cont_I^2, \mathcal{E}_I}_{Context_J} \;;}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J, \mathcal{QA}''_J \;;\; \mathcal{P}_I, \mathcal{QA}_I^2, q\mathcal{A}_I^1, \mathcal{C}_I^1, Cont_I^2, \mathcal{E}_I \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}\; J$$

$$
\cfrac{
\begin{array}{c}
\underbrace{
\cfrac{
\begin{array}{c}
\underbrace{\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}_I^1}_{\mathcal{A}''_J,\mathcal{QA}''_J,q\mathcal{A}_I^1} \;;\; \underbrace{context_I, Cont_I^1}_{\mathcal{A}'_J,\mathcal{QA}'_J,\mathcal{C}_I^1} \;;}\\
\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1}_{Context_J} \;;
\end{array}
}{
\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J, \mathcal{QA}''_J \;;\; \mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1 \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J,\mathcal{A}''_J}
}\,J
}\\
\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}''_J, \mathcal{A}''_J, q\mathcal{A}_I^1}_{\mathcal{QA}_I^1} \;;\; \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}_I^1, \mathcal{QA}'_J}_{context_I, Cont_I^1}
\end{array}
\qquad
\vdash \mathcal{A}_I^2 \;;\; \mathcal{QA}_I^2 \;;\; context_I, Cont_I^2 \;;
}{
\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I^1, \mathcal{QA}_I^2 \;;\; context_I, Cont_I^1, Cont_I^2 \;;\; \mathcal{E}_I
}\,I
$$

The permutation result for the strong-left and the strong-right permutation position (in the permutation object) would be:

$$
\cfrac{
\begin{array}{c}
\underbrace{
\cfrac{
\begin{array}{c}
\underbrace{\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}_I^1}_{\mathcal{A}''_J,\mathcal{QA}''_J,q\mathcal{A}_I^1} \;;\; \underbrace{context_I, Cont_I^1}_{\mathcal{A}'_J,\mathcal{QA}'_J,\mathcal{C}_I^1} \;;}\\
\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1}_{Context_J} \;;
\end{array}
}{
\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J, \mathcal{QA}''_J \;;\; \mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1 \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J,\mathcal{A}''_J}
}\,J
}\\
\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}''_J, \mathcal{A}''_J, q\mathcal{A}_I^1}_{\mathcal{QA}_I^1} \;;\; \underbrace{\mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}_I^1, \mathcal{QA}'_J}_{context_I, Cont_I^1}
\end{array}
\qquad
\begin{array}{c}
\underbrace{
\cfrac{
\begin{array}{c}
\underbrace{\vdash \mathcal{A}_I^2 \;;\; \underbrace{\mathcal{QA}_I^2}_{\mathcal{B}''_J,\mathcal{QB}''_J,q\mathcal{B}_I^1} \;;\; \underbrace{context_I, Context_I^2}_{\mathcal{B}'_J,\mathcal{QB}'_J,\mathcal{CB}_I^1} \;;}\\
\vdash \underbrace{\mathcal{B}'_J, \mathcal{B}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QB}'_J, \mathcal{QB}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^2, q\mathcal{B}_I^1, \mathcal{CB}_I^1}_{Context_J} \;;
\end{array}
}{
\vdash \mathcal{P}_J \;;\; \mathcal{QB}'_J, \mathcal{QB}''_J \;;\; \mathcal{A}_I^2, q\mathcal{B}_I^1, \mathcal{CB}_I^1 \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{EB}'_J,\mathcal{B}''_J}
}\,J
}\\
\vdash \mathcal{A}_I^2 \;;\; \underbrace{\mathcal{QB}''_J, \mathcal{B}''_J, q\mathcal{B}_I^1}_{\mathcal{QA}_I^2} \;;\; \underbrace{\mathcal{P}_J, \mathcal{EB}'_J, \mathcal{CB}_I^1, \mathcal{QB}'_J}_{context_I, Cont_I^2}
\end{array}
}{
\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I^1, \mathcal{QA}_I^2 \;;\; context_I, Cont_I^1, Cont_I^2 \;;\; \mathcal{E}_I
}\,I
$$

□

According to the above result the *Context* part is the only part of a rule which can change during a permutation. Hence, it is sufficient to concentrate on the *Context* parts of given rules and check whether or not the multisets, specified in the above proposition, can be accepted as the contexts of appropriate rules after the eventual permutation.

6.2.1 Possible generalisation

It is important to point out that results of part 3) of Proposition 6.1 assume that I *is always above the left (i.e. first) premise of* J *in the permutation object.* They can be generalised to deal with the cases where an inference rule I is above the p-th premise of an inference rule J $(p \in \{1,2\})$.

Let inference rules I and J (I is above the p-th premise of J) form the permutation object, and $k \in \{1,2\}$ correspond to appropriate strong-permutation position of I and J. Let $\bar{p}$ be defined as $3-p$, and multisets $\mathcal{C}_I$ and $\mathcal{C}_J$ as follows:

$$\mathcal{C}_I = \begin{cases} \emptyset & \text{if } I \text{ is a strong-additive rule} \\ Context_I^{\bar{k}} & \text{otherwise} \end{cases} \qquad \mathcal{C}_J = \begin{cases} \emptyset & \text{if } J \text{ is a strong-additive rule} \\ Context_J^{\bar{p}} & \text{otherwise} \end{cases}$$

After the eventual permutation, the new *Context*(s) of permuted rules I and J are defined by the functions $\mathcal{M}_{I,J}(k,p)$ and $\mathcal{N}_{I,J}(k,p)$ respectively, as follows:

Select case: $\mathcal{M}_{I,J}(k,p)$

J unary	$[Context^k_I, \mathcal{E}_J] \backslash \mathcal{A}_J\ ,\ \mathcal{P}_J$
J binary	$[Context^k_I, \mathcal{E}_J] \backslash \mathcal{A}^p_J\ ,\ \mathcal{P}_J\ ,\ \mathcal{QA}^p_J\ ,\ \mathcal{C}_J$

Select case: $\mathcal{N}_{I,J}(k,p)$

I unary	$Context^p_J \backslash [\mathcal{P}_I, \mathcal{E}_I]\ ,\ \mathcal{A}_I$
I binary	$Context^p_J \backslash [\mathcal{P}_I, \mathcal{E}_I, \mathcal{QA}^{\bar{k}}_I, \mathcal{C}_I]\ ,\ \mathcal{A}^k_I$

6.2.2 Implementing restrictions for *Context*

Let us recall (from subsection 4.2), that *Context* part of a rule can have three types of restrictions: *structure restrictions, equality restrictions* and *eigen-variable restrictions.* Intuitively is clear that only multisets which satisfied all restrictions can be accepted as (a part of) the *Context* of a given rule.

Let Γ be a given multiset, I a given schema-rule and Δ a schema-multiset which presents the $Context_I$ or its particular part. A simple way to implement the test for checking whether or not multiset Γ satisfies the *"structure restrictions* of formulas in Δ, is through a decision procedure, denoted **test STRUCT**, as follows:

Test STRUCT(Γ: multiset, Δ : schema-multiset)

If for every formula F from multiset Γ:

$$CONNECTIVES(F) \cap FORBIDDEN(\Delta) = \emptyset$$
$$Top(F) \in TOP(\Delta)$$
$$const(F) \in CONSTANT(\Delta)$$
$$atom(F) \leq atom(\Delta)$$

Then **Return** true

Else **Return** false

For instance, only the first of given multisets:

$$\Gamma_1 = !A, !(B \oplus C), !\forall x C(x), a, !A$$
$$\Gamma_2 = !A, B \oplus C, !\forall x C(x), a, !A$$
$$\Gamma_3 = !A, !(B \otimes C), !\forall x C(x), a, !A$$

can be unified with the given schema-multiset: ${}_{!/\emptyset/1}\Delta_{\otimes}$ *i.e.*

$$\mathbf{STRUCT}(\Gamma_1,\ {}_{!/\emptyset/1}\Delta_{\otimes}\ ) = \text{true}$$
$$\mathbf{STRUCT}(\Gamma_2,\ {}_{!/\emptyset/1}\Delta_{\otimes}\ ) = \text{false}$$
$$\mathbf{STRUCT}(\Gamma_3,\ {}_{!/\emptyset/1}\Delta_{\otimes}\ ) = \text{false}$$

Finally, we define test **D** which we find useful in the implementation on the *equality restrictions* for the *Context* of rules which should be permuted. As we will see in the next subsection, an implementation of the conditions $\mathcal{B}1$ and $\mathcal{B}2$ implies a particular combining of tests **STRUCT** and **D**. For a given inference I and a natural number $k \in \{1, 2\}$, test **D** is defined as follows:

Test D(I: inference, k: number)

If I unary

Then **If** $\mathcal{P}_I = \emptyset \wedge \mathcal{E}_I = \mathcal{A}_I$

Then **Return** true

Else **Return** false

Else **If** I multiplicative rule
  Then **If** $\mathcal{P}_I = \emptyset \wedge \mathcal{E}_I, \mathcal{QA}_I^{\bar{k}}, Context_I^{\bar{k}} = \mathcal{A}_I^k$
    Then **Return** true
    Else **Return** false
  Else **If** $\mathcal{P}_I = \emptyset \wedge \mathcal{E}_I, \mathcal{QA}_I^{\bar{k}} = \mathcal{A}_I^k$
    Then **Return** true
    Else **Return** false

Note that instances of a rule I which verify test **D** actually represent redundant steps. Generally we are interested in elimination of such steps. For some analyses it could be interesting to find and point out (for example, by using permutation analysis) all the positions in a given proof where redundant instances of a given rule can appear. Redundant instances of a rule can be used as markers for some important sub-proofs.

6.3 Specifying the permutation process

In this section we summarise all necessary and sufficient tests for the permutation of given rules, according to arity of the rules in question and the form of the permutation result.

6.3.1 Unary rules

Theorem 6.2
A unary rule I is permutable over an unary rule J iff they verify the tests:
$\mathbf{A}_{I,J,\Pi}(1)$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1{,}1), Context_J^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1{,}1), Context_I^1)$.
Proof:
For unary I and J, having in mind the results of the Proposition 6.1, the following dependencies between the conditions and the tests are immediate:

$$\mathcal{A} \Longleftrightarrow \mathbf{A}_{I,J,\Pi}(1)$$
$$\mathcal{B}1 \Longleftrightarrow \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1,1), Context_J^1),$$
$$\mathcal{B}2 \Longleftrightarrow \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_J^1)$$

($\Rightarrow$): By Definition 5.3, I is permutable over J iff the conditions $\mathcal{A}, \mathcal{B}1, \mathcal{B}2$ and $\mathcal{C}$ hold. Hence, I and J must verify the tests $\mathbf{A}_{I,J,\Pi}(1)$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1{,}1), Context_J^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1{,}1), Context_I^1)$.
($\Leftarrow$): verification of the given tests ensures the transformation of the starting permutation object $i)$ into the derivation $ii)$:

$i)$

$$\underbrace{\dfrac{\vdash \mathcal{A}_I \;;\; \mathcal{QA}_I \;;\; Context_I \;;}{\vdash \mathcal{P}_I \;;\; \underbrace{\mathcal{QA}_I}_{\mathcal{A}''_J,\mathcal{QA}''_J,q\mathcal{A}_I} \;;\; \underbrace{Context_I}_{\mathcal{A}'_J,\mathcal{QA}'_J,\mathcal{C}_I} \;;\; \mathcal{E}_I}\,I}$$
$$\dfrac{\vdash \underbrace{\mathcal{A}'_J,\mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J,\mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{C}_I,q\mathcal{A}_I,\mathcal{P}_I,\mathcal{E}_I}_{Context_J} \;;}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J,\mathcal{QA}''_J \;;\; \mathcal{C}_I,q\mathcal{A}_I,\mathcal{P}_I,\mathcal{E}_I \;;\; \mathcal{E}_J}\,J$$

$ii)$

$$\underbrace{\vdash \mathcal{A}_I \;;\; \mathcal{QA}_I, \;;\; Context_I \;;}$$
$$\underbrace{\dfrac{\vdash \mathcal{A}_J \;;\; \mathcal{QA}_J \;;\; q\mathcal{A}_I,\mathcal{C}_I,\mathcal{A}_I \;;}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J,\mathcal{QA}''_J \;;\; q\mathcal{A}_I,\mathcal{C}_I,\mathcal{A}_I \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{A}''_J,\mathcal{E}'_J}}\,J}$$
$$\dfrac{\vdash \mathcal{A}_I \;;\; \mathcal{QA}_I \;;\; \mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J \;;}{\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I \;;\; \mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J \;;\; \mathcal{E}_I}\,I$$

The equality of end-sequents (condition $\mathcal{C}$) is fulfilled automatically and can be checked directly. □

6.3.2 Binary-unary rules

For the case of a binary rule I and unary J more care is needed. Let us denote their initial structure by F_0 and possible forms of permutation results respectively by F_1, F_2 and F_3:

$$F_0 : \quad \dfrac{\dfrac{\overset{\vdots}{\gamma_1} \quad \overset{\vdots}{\gamma_2}}{\beta}\, I}{\alpha}\, J \qquad\qquad F_1 : \quad \dfrac{\dfrac{\overset{\vdots}{\gamma_1}}{\beta_1}\, J \qquad \overset{\vdots}{\gamma_2}}{\alpha}\, I \qquad\qquad F_2 : \quad \dfrac{\overset{\vdots}{\gamma_1} \qquad \dfrac{\overset{\vdots}{\gamma_2}}{\beta_2}\, J}{\alpha}\, I \qquad\qquad F_3 : \quad \dfrac{\dfrac{\overset{\vdots}{\gamma_1}}{\beta_1}\, J \qquad \dfrac{\overset{\vdots}{\gamma_2}}{\beta_2}\, J}{\alpha}\, I$$

Theorem 6.3
A multiplicative rule I is permutable over a unary rule J, giving permutation result of the form

1) $F_k,\ k \in \{1,2\}$
2) F_3 (even more: all three forms of permutation result are possible)

iff they verify the tests:

1) $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(\mathrm{k},1), Context_J^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(\mathrm{k},1), Context_I^k)$
2) $\forall k \in \{1,2\}$ $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(\mathrm{k},1), Context_J^1)$ and $\mathbf{D}(J,1)$.

Proof:
1) ($\Rightarrow$): since the permutation result has the form $F_k,\ k \in \{1,2\}$, J is applied on the k-th premise of I in the permutation result and the test $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$ must be fulfilled. Since I is a multiplicative rule and J a unary rule, their *Context* parts can have just restrictions on the structure of formulas. According to results of Proposition 6.1, the fulfilment of the conditions $\mathcal{B}1$ and $\mathcal{B}2$ implies verification of test **STRUCT**, as follows:

$$\mathcal{B}1 \Longleftrightarrow \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1)$$
$$\mathcal{B}2 \Longleftrightarrow \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k)$$

($\Leftarrow$): let assume verification of the given tests for $k = 1$. Then it is possible to transform the permutation object F into the derivation F_1

i) F :

$$\dfrac{\vdash \mathcal{A}_I^1 \ ;\ \mathcal{QA}_I^1 \ ;\ Context_I^1 \ ; \qquad \vdash \mathcal{A}_I^2 \ ;\ \mathcal{QA}_I^2 \ ;\ Context_I^2 \ ;}{\underbrace{\vdash \mathcal{P}_I \ ;\ \underbrace{\mathcal{QA}_I^1}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I^1}, \mathcal{QA}_I^2 \ ;\ \underbrace{Context_I^1}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{C}_I^1}, Context_I^2 \ ;\ \mathcal{E}_I}}\ I$$

$$\dfrac{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}_J} \ ;\ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}_J} \ ;\ \underbrace{\mathcal{P}_I, \mathcal{QA}_I^2, q\mathcal{A}_I^1, \mathcal{C}_I^1, Context_I^2, \mathcal{E}_I}_{Context_J} \ ;}{\vdash \mathcal{P}_J \ ;\ \mathcal{QA}'_J, \mathcal{QA}''_J \ ;\ \mathcal{P}_I, \mathcal{QA}_I^2, q\mathcal{A}_I^1, \mathcal{C}_I^1, Context_I^2, \mathcal{E}_I \ ;\ \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}\ J$$

ii) F_1 :

$$\underbrace{\vdash \mathcal{A}_I^1 \ ;\ \mathcal{QA}_I^1 \ ;\ Context_I^1 \ ;}$$

$$\dfrac{\vdash \mathcal{A}_J \ ;\ \mathcal{QA}_J \ ;\ \mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1 \ ;}{\underbrace{\vdash \mathcal{P}_J \ ;\ \mathcal{QA}_J \ ;\ \mathcal{A}_I^1, q\mathcal{A}_I^1, \mathcal{C}_I^1 \ ;\ \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J, \mathcal{A}''_J}}}\ J$$

$$\dfrac{\vdash \mathcal{A}_I^1 \ ;\ \mathcal{QA}_I^1 \ ;\ \mathcal{P}_J, \mathcal{E}'_J, \mathcal{C}_I^1, \mathcal{QA}'_J \qquad \vdash \mathcal{A}_I^2 \ ;\ \mathcal{QA}_I^2 \ ;\ Context_I^2 \ ;}{\vdash \mathcal{P}_I \ ;\ \mathcal{QA}_I^1, \mathcal{QA}_I^2 \ ;\ Context_I^1, Context_I^2 \ ;\ \mathcal{E}_I}\ I$$

The equality of the end-sequents is fulfilled automatically and can be checked directly. Therefore the derivation F_1 is the permutation result for I and J.

2) ($\Rightarrow$): the existence of F_1 and F_2 (forms of) permutation result(s) implies that $\forall k \in \{1,2\}$ the following tests must be verified:

$\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(\mathrm{k},1), Context_J^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(\mathrm{k},1), Context_I^k)$.

The F_3 permutation result has the following form:

$$\dfrac{\underbrace{\dfrac{\underbrace{\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}_I^1}_{\mathcal{A}''_J,\mathcal{QA}''_J,q\mathcal{A}_I^1} \;;\; \underbrace{Context_I^1}_{\mathcal{A}'_J,\mathcal{QA}'_J,\mathcal{C}_I} \;;}_{\displaystyle \vdash \underbrace{\mathcal{A}'_J,\mathcal{A}''_J}_{\mathcal{A}_J} \;;\; \underbrace{\mathcal{QA}'_J,\mathcal{QA}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^1,q\mathcal{A}_I^1,\mathcal{C}_I}_{Context_J} \;;}}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J,\mathcal{QA}''_J \;;\; \mathcal{A}_I^1,q\mathcal{A}_I^1,\mathcal{C}_I \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{E}'_J,\mathcal{A}''_J}}\,J}_{\displaystyle \vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}''_J,\mathcal{A}''_J,q\mathcal{A}_I^1}_{\mathcal{QA}_I^1} \;;\; \underbrace{\mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J}_{Context_I^1}} \qquad \underbrace{\dfrac{\underbrace{\vdash \mathcal{A}_I^2 \;;\; \underbrace{\mathcal{QA}_I^2}_{\mathcal{B}''_J,q\mathcal{B}''_J,q\mathcal{B}_I} \;;\; \underbrace{Context_I^2}_{\mathcal{B}'_J,q\mathcal{B}'_J,\mathcal{CB}_I} \;;}_{\displaystyle \vdash \underbrace{\mathcal{B}'_J,\mathcal{B}''_J}_{\mathcal{A}_J} \;;\; \underbrace{q\mathcal{B}'_J,q\mathcal{B}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^2,q\mathcal{B}_I,\mathcal{CB}_I}_{Context_J} \;;}}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}'_J,\mathcal{QA}''_J \;;\; \mathcal{A}_I^1,q\mathcal{B}_I^1,\mathcal{CB}_I \;;\; \underbrace{\mathcal{E}_J}_{\mathcal{EB}_J,\mathcal{B}''_J}}\,J}_{\displaystyle \vdash \mathcal{A}_I^2 \;;\; \underbrace{q\mathcal{B}''_J,\mathcal{B}''_J,q\mathcal{B}_I^2}_{\mathcal{QA}_I^2} \;;\; \underbrace{\mathcal{P}_J,\mathcal{EB}_J,\mathcal{CB}_I,q\mathcal{B}'_J}_{Context_I^2}}}{\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I^1,\mathcal{QA}_I^2 \;;\; \mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J,\mathcal{P}_J,\mathcal{EB}_J,\mathcal{CB}_I,q\mathcal{B}'_J \;;\; \mathcal{E}_I}\,I$$

where its end-sequent must be the same as the end-sequent of the starting permutation object $\mathbf{F}$ (i). The equality of these end-sequents implies the following equivalences:

$$\begin{array}{ll}
 & \text{end-sequents of } F \text{ and } F_3 \text{ are equal } \underline{(\text{condition } \mathcal{C})} \\
\Leftrightarrow & \mathcal{P}_J,\, \mathcal{EB}_J = \mathcal{B}'_J \\
\Leftrightarrow & (\mathcal{P}_J = \emptyset \wedge \mathcal{EB}_J = \emptyset \wedge \mathcal{B}'_J = \emptyset) \vee (\mathcal{P}_J = \emptyset \wedge \mathcal{EB}_J = \mathcal{B}'_J)^6 \\
\Leftrightarrow & (\mathcal{P}_J = \mathcal{EB}_J = \mathcal{A}'_J = \emptyset) \vee (\mathcal{P}_J = \emptyset \wedge \mathcal{E}_J = \mathcal{A}_J) \\
\Leftrightarrow & \mathcal{P}_J = \emptyset \wedge \mathcal{E}_J = \mathcal{A}_J \;\underline{(\text{test } \mathbf{D}(J,1\,))}
\end{array}$$

($\Leftarrow$): it is obvious that fulfilment of the test $\mathbf{D}(J)$ implies fulfilment of the tests: $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(2,1), Context_I^2)$. According to part **i)** of this theorem there is the permutation result of forms F_1 and F_2. As a consequence of their existence it is possible to apply J on each premise of I and then to apply I on the resulting sequent:

$$\dfrac{\dfrac{\vdash \mathcal{A}_I^1 \;;\; \mathcal{QA}_I^1 \;;\; Context_I^1}{\vdash \mathcal{A}_I^1 \;;\; \underbrace{\mathcal{QA}''_J,\mathcal{A}''_J,q\mathcal{A}_I^1}_{\mathcal{QA}_I^1} \;;\; \mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J}\,J \qquad \dfrac{\underbrace{\vdash \mathcal{A}_I^2 \;;\; \underbrace{\mathcal{QA}_I^2}_{\mathcal{B}''_J,q\mathcal{B}''_J,q\mathcal{B}_I} \;;\; \underbrace{Context_I^2}_{\mathcal{B}'_J,q\mathcal{B}'_J,\mathcal{CB}_I} \;;}_{\displaystyle \vdash \underbrace{\mathcal{B}'_J,\mathcal{B}''_J}_{\mathcal{A}_J} \;;\; \underbrace{q\mathcal{B}'_J,q\mathcal{B}''_J}_{\mathcal{QA}_J} \;;\; \underbrace{\mathcal{A}_I^2,q\mathcal{B}_I,\mathcal{CB}_I}_{Context_J} \;;}}{\vdash \mathcal{A}_I^2 \;;\; \mathcal{QA}_I^2 \;;\; \mathcal{P}_J,\mathcal{EB}_J,\mathcal{CB}_I,q\mathcal{B}'_J}\,J}{\vdash \mathcal{P}_I \;;\; \mathcal{QA}_I^1,\mathcal{QA}_I^2 \;;\; \mathcal{P}_J,\mathcal{E}'_J,\mathcal{C}_I,\mathcal{QA}'_J,\mathcal{P}_J,\mathcal{EB}_J,\mathcal{CB}_I,q\mathcal{B}'_J \;;\; \mathcal{E}_I}\,I$$

As test $\mathbf{D}(J)$ ensures the equality of end-sequents of generated derivation and the corresponding permutation object F, the above derivation is (by definition) a permutation result in F_3 form. □

[6]note that as $\mathcal{B}'_J \subseteq \mathcal{A}_J$ and $\mathcal{P}_J \cap \mathcal{A}_J = \emptyset$, we have $\mathcal{P}_J \cap \mathcal{B}'_J = \emptyset$

Theorem 6.4
A strong additive rule I is permutable over a unary rule J, giving permutation result of the form:

1) F_3

2) $F_k,\ \ k\ \in\ \{1,2\}$

iff they verify the tests:

1) $\forall k \in \{1,2\}$ $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(\mathrm{k},1), \mathit{Context}_J^1)$ and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(\mathrm{k},1), \mathit{Context}_I^k)$

2) $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(\mathrm{k},1), \mathit{Context}_J^1)$ and $\mathbf{D}(J,1)$.

Proof: similar to proof of Theorem 6.3.

□

For instance, in linear logic, for the strong additive rule $\dfrac{\vdash \phi\ ;\ ;\ \Sigma \qquad \vdash \phi\ ;\ ;\ \Sigma}{\vdash \phi\&\psi\ ;\ ;\ \Sigma\ ;}\ \&$ and unary $\dfrac{\vdash ?\psi\ ;\ ?\psi\ ;\ \Delta}{\vdash\ \ ;\ ?\psi\ ;\ \Delta\ ;\ ?F}\ cw$, (which is a mix of contraction c and weakening rule w), we have:

$$F_0.\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash B,\Gamma,?A,?A\end{matrix} \qquad \begin{matrix}\vdots\\ \vdash C,\Gamma,?A,?A\end{matrix}}{\vdash ?A,?A,\Gamma,B\&C}\ \&}{\vdash ?A,\Gamma,B\&C,?A}\ cw \quad \Longrightarrow \left\{ \begin{array}{l} F_{2.1}.\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash ?A,?A,\Gamma,B\end{matrix}}{\vdash B,\Gamma,?A,?A}\ cw \qquad \begin{matrix}\vdots\\ \vdash C,\Gamma,?A,?A\end{matrix}}{\vdash ?A,\Gamma,B\&C,?A}\ \& \\[3em] F_{2.2}.\quad \dfrac{\begin{matrix}\vdots\\ \vdash B,\Gamma,?A,?A\end{matrix} \qquad \dfrac{\begin{matrix}\vdots\\ \vdash ?A,?A,\Gamma,C\end{matrix}}{\vdash C,\Gamma,?A,?A}\ cw}{\vdash ?A,\Gamma,B\&C,?A}\ \& \\[3em] F_{2.3}.\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash ?A,?A,\Gamma,B\end{matrix}}{\vdash B,\Gamma,?A,?A}\ cw \qquad \dfrac{\begin{matrix}\vdots\\ \vdash ?A,?A,\Gamma,C\end{matrix}}{\vdash C,\Gamma,?A,?A}\ cw}{\vdash ?A,\Gamma,B\&C,?A}\ \& \end{array}\right.$$

For situations where weak additive rules are involved more care is needed. Let us recall our notation for weak-additive rules. We denote the conclusion's and premise's part of the context which must be identical by $\mathit{context}_I$ and a "free" (unconditional part) part by Cont_I^k. So, generally, a weak additive rule I is of the form:

$$\dfrac{\vdash \mathcal{A}_I^1\ ;\ \mathcal{QA}_I^1\ ;\ \mathit{context}_I, \mathit{Cont}_I^1 \qquad \vdash \mathcal{A}_I^2\ ;\ \mathcal{QA}_I^2\ ;\ \mathit{context}_I, \mathit{Cont}_I^2}{\vdash \mathcal{P}_I\ ;\ \mathcal{QA}_I^1, \mathcal{QA}_I^2\ ;\ \mathit{context}_I, \mathit{Cont}_I^1, \mathit{Cont}_I^2\ ;\ \mathcal{E}_I}\ I$$

According to $\mathcal{SC}$ proof systems outlined in Subsection 1.2, it is possible to distinguish two types of specific criteria for parts of the context which must be the same in premises and conclusion. So we distinguish *two* types of weak additive rules in the analysed $\mathcal{SC}$ proof systems:

type 1:

$$\dfrac{\vdash \mathcal{A}_I^1\ ;\ \mathcal{QA}_I^1\ ;\ \Gamma, \Delta_1 \qquad \vdash \mathcal{A}_I^2\ ;\ \mathcal{QA}_I^2\ ;\ \Gamma, \Delta_2}{\vdash \mathcal{P}_I\ ;\ \mathcal{QA}_I^1, \mathcal{QA}_I^2\ ;\ \Gamma, \Delta_1, \Delta_2\ ;\ \mathcal{E}_I}\ I$$

where $\Delta_1 \cap \Delta_2\ =\ \emptyset$

type 2:

$$\dfrac{\vdash \mathcal{A}_I^1\ ;\ \mathcal{QA}_I^1\ ;\ \Gamma,\ {}_{t_1/a/c_1}\Delta_{f_1} \qquad \vdash \mathcal{A}_I^2\ ;\ \mathcal{QA}_I^2\ ;\ \Sigma,\ {}_{t_1/a/c_1}\Delta_{f_1}}{\vdash \mathcal{P}_I\ ;\ \mathcal{QA}_I^1, \mathcal{QA}_I^2\ ;\ \Gamma, \Sigma,\ {}_{t_1/a/c_1}\Delta_{f_1}\ ;\ \mathcal{E}_I}\ I$$

For rules of type 1. the *Context* part which is common for premises and conclusion is detected only at the moment of the rule's instantiation. So, for a given schema of such a weak-additive rule and a concrete multiset Ω it is not possible to determine, in advance, a multi-subset of Ω which can belong to (be an element of, be unified with) $\mathit{context}_I$ and a multi-subset which can belong to Cont_I^k. For rules of type 2. belonging to particular part of *Context* is determined directly and only by the form of its formulas.

Let us illustrate these types of weak additive rules by examples:

$$\frac{\vdash\ ;\ ;\ \Delta, ?\Sigma, ?\Theta_1 \qquad \vdash B\ ;\ ;\ \Delta, ?\Sigma, ?\Theta_2}{\vdash_{?/0/\bot} A\&B\ ;\ ;\ \Delta, ?\Sigma, ?\Theta_1, ?\Theta_2\ ;}\ \&_l \qquad\qquad \frac{\vdash A\ ;\ ;\ ?\Delta, \Gamma \qquad \vdash B\ ;\ ;\ ?\Delta, \Sigma}{\vdash A \otimes B\ ;\ ;\ ?\Delta, \Gamma, \Sigma\ ;}\ \otimes$$

$$?\Theta_1 \cap ?\Theta_2 = \emptyset$$

Weak-additive rules demand more details to be checked in the permutation process. The following theorem illustrates the specification of the permutation process which involves weak-additive rules of type 2. For a given weak-additive rule I and a multiset Ω, we denote a multi-subset of Ω which can belong to (to be an element of, to unified with) $context_I$ by $\mathcal{LIMIT}_I[\Omega]$ and a multi-subset which can belong to $Cont_I^k$ by $\mathcal{FREE}_I^k[\Omega]$.

Theorem 6.5

A weak additive rule I is permutable over a unary rule J giving permutation result of the form $F_k,\ \ k \in \{1,2\}$ iff they verify the tests:

$\mathbf{A}_{I,J,\Pi}(\mathrm{k})$,

$\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1)$

$\mathbf{STRUCT}(\ Cont_I^k \backslash \mathcal{A}_J\ ,\ \mathcal{P}_J, \mathcal{FREE}_I^k[\mathcal{E}_J \backslash\ (\mathcal{A}_J \backslash Context_I^k)],\ \ Cont_I^k)$,

$\mathcal{LIMIT}_I[\mathcal{P}_J] = \emptyset$, and

$\mathcal{LIMIT}_I[\ \mathcal{E}_J \backslash\ (\mathcal{A}_J \backslash Context_I^k)\ ] = \mathcal{A}_J\ \cap\ context_I$

Proof: similar to proof of Theorem 6.3. □

As an illustration, consider the permutation of a weak-additive and a unary rule through examples from $\mathcal{SC}$ 6.5 and $\mathcal{SC}$ 6.7 respectively:

1.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash \Gamma, C\end{matrix} \quad \begin{matrix}\vdots\\ \vdash \Gamma, C\end{matrix}}{\vdash ?A \otimes ?B, \Gamma, C}\ \otimes_{lr}}{\vdash ?A \otimes ?B, \Gamma, C \wp ?D}\ \wp \quad \Longleftrightarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash \Gamma, C\end{matrix}}{\vdash \Gamma, C \wp ?D}\ \wp \quad \dfrac{\begin{matrix}\vdots\\ \vdash \Gamma, C\end{matrix}}{\vdash \Gamma, C \wp ?D}\ \wp}{\vdash ?A \otimes ?B, \Gamma, C \wp ?D}\ \otimes_{lr}$$

2.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash A, C, \Gamma, ?\Delta\end{matrix} \quad \begin{matrix}\vdots\\ \vdash B, \Sigma, ?\Delta\end{matrix}}{\vdash A \otimes B, C, \Gamma, \Sigma, ?\Delta}\ \otimes'}{\vdash C \oplus D, A \otimes B, \Gamma, \Sigma, ?\Delta}\ \oplus \quad \Longleftrightarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash C, A, \Gamma, ?\Delta\end{matrix}}{\vdash C \oplus D, A, \Gamma, ?\Delta}\ \oplus \quad \begin{matrix}\vdots\\ \vdash B, \Sigma, ?\Delta\end{matrix}}{\vdash C \oplus D, A \otimes B, \Gamma, \Sigma, ?\Delta}\ \otimes'$$

As we have seen, weak-additive rules demand more details to be checked in the permutation process. For the reason of obvious simplicity we will restrict our attention, in the rest of the paper, to unary, multiplicative and strong-additive binary rules only. This is done to simplify the technicalities, and in principle there seems be no reasons to restrict additive rules to strong-additive type only.

6.3.3 Unary-binary rules

Theorem 6.6

A unary rule I is permutable over a multiplicative rule J iff they verify the tests:

$\mathbf{A}_{I,J,\Pi}(1)$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1,1), Context_J^1)$, and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I^1)$

Proof: similar to proof of Theorem 6.3 □

Theorem 6.7
A unary rule I is permutable over a strong additive rule J iff they verify the tests:

$\mathbf{A}_{I,J,\Pi}(1)$, $\mathbf{D}(I,1)$, and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context^1_I)$

Proof: similar to proof of Theorem 6.3 □

For instance, in classical logic, for strong additive rule $\dfrac{\vdash \phi\,;\,;\,\Sigma \quad \vdash \phi\,;\,;\,\Sigma}{\vdash \phi\wedge\psi\,;\,;\,\Sigma\,;}\wedge$ and a unary : $\dfrac{\vdash F\,;\,F\,;\,\Delta}{\vdash\ ;\,F\,;\,\Delta\,;G}cw$ which is a mix of contraction c and weakening rule w, we will have:

$$\dfrac{\dfrac{\vdash C,C,A}{\vdash C,A,C}cw \qquad \vdash B,C,C}{\vdash A\wedge B,C,C}\wedge \quad\Longrightarrow\quad \dfrac{\dfrac{\vdash A,C,C \qquad \vdash B,C,C}{\vdash A\wedge B,C,C}\wedge}{\vdash C,A\wedge B,C}cw$$

As we already pointed out instances of a given rule which satisfy the test **D** present redundant steps in a proof. An eventual elimination of redundant steps in a given proof prior to any other proof transformation will result in the proof with no inferences which meet the requirements of the test **D**. According to Theorem 6.7, fulfilment of the test $\mathbf{D}(I)$ is a necessary and sufficient condition for the permutation of a given unary rule I over a given strong-additive rule J. If we adopt the strategy of *"elimination redundant steps in a given proof prior to any other proof transformation"*, then as a consequence of Theorem 6.7, we come to the statement:

None of the unary rules is permutable over a strong-additive rule.

But, this non-permutability is relative and can be overcome by special treatment (see Subsection 4.2 of [13]). For a unary rule I and a strong-additive rule J of a given proof Π, the idea is to notice the permutation figure I, L, J (if it exists):

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \alpha\end{matrix}}{\gamma_1}I \qquad \dfrac{\begin{matrix}\vdots\\ \beta\end{matrix}}{\gamma_2}L}{\delta}J$$

and to try to transform it in a permutation result of the form: $\dfrac{\dfrac{\begin{matrix}\vdots & \vdots\\ \alpha & \beta\end{matrix}}{\gamma}J}{\delta}I$ We denote this way of permutation as *special permutation of I over J*. As an example, consider the following permutations from $\mathcal{SCKT}^{\mathcal{S}}$ [26] and linear logic, respectively:

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash A,\Diamond A,B,\Gamma\end{matrix}}{\vdash \Diamond A,B,\Gamma}\Diamond \qquad \dfrac{\begin{matrix}\vdots\\ \vdash A,\Diamond A,C,\Gamma\end{matrix}}{\vdash \Diamond A,C,\Gamma}\Diamond}{\vdash B\wedge C,\Diamond A,\Gamma}\wedge \quad\Longrightarrow\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash A,\Diamond A,B,\Gamma\end{matrix} \qquad \begin{matrix}\vdots\\ \vdash A,\Diamond A,C,\Gamma\end{matrix}}{\vdash B\wedge C,A,\Diamond A,\Gamma}\wedge}{\vdash B\wedge C,\Diamond A,\Gamma}\Diamond$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash A,C,D\end{matrix}}{\vdash A,C\wp D}\wp \qquad \dfrac{\begin{matrix}\vdots\\ \vdash B,C,D\end{matrix}}{\vdash B,C\wp D}\wp}{\vdash A\&B,C\wp D}\& \quad\Longrightarrow\quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash A,C,D\end{matrix} \qquad \begin{matrix}\vdots\\ \vdash A,C,D\end{matrix}}{\vdash A\&B,C,D}\&}{\vdash A\&B,C\wp D}\wp$$

The following definition is an generalisation of Definition 5.3 to deal with special permutability.

Definition 6.8 *(special-permutability of inferences)*
An inference I is *special-permutable over* an inference J of a given proof iff they verify the conditions:

$\boxed{\mathcal{A}.}$ there exist an inference L such that I, L, J form a permutation figure;

$\boxed{\mathcal{B}.}$ For their permutation figure there exists a permutation result.

In the other cases, we say that I is *not special-permutable over J*.

As in the case of ordinary permutability, the condition $\mathcal{B}$ can be decomposed into the conditions $\mathcal{B}1, \mathcal{B}2$ and $\mathcal{C}$, as follows:

$\boxed{\mathcal{B}1}$: J is applicable on the appropriate premise(s) of the I, L, J permutation figure;
$\boxed{\mathcal{B}2}$: The generated conclusion-sequent can be a premise of I,

$\boxed{\mathcal{C}}$: Subproof $\overset{\vdots}{\underset{I}{J}}$ and the starting permutation figure have the same end-sequent.

The following theorem gives a possible specification of a special-permutation.

Theorem 6.9
A unary rule I is special-permutable over a strong-additive rule J, iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$$\begin{array}{l} type(I) = type(L) \\ \mathbf{A}_{I,J,\Pi}(1), \\ \mathbf{A}_{L,J,\Pi}(1) \\ \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1,1), Context_J), \\ \mathcal{N}_{I,J,\Pi}(1,1) \ = \ \mathcal{N}_{L,J,\Pi}(1,2), \\ \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I) \end{array}$$

Proof:
($\Rightarrow$:) follows from Definition 5.3 and a comparison of the permutation figure and the permutation result.
($\Leftarrow$:) verification of the tests: $\mathbf{A}_{I,J,\Pi}(1)$, $type(I) = type(L)$ and $\mathbf{A}_{L,J,\Pi}(1)$ ensures the existence of the permutation figure of the form:

$$\dfrac{\underbrace{\dfrac{\vdash \mathcal{A}_I \ ; \ \mathcal{QA}_I \ ; \ Context_I \ ;}{\vdash \mathcal{P}_I \ ; \ \underbrace{\mathcal{QA}_I}_{\mathcal{A}''_J, \mathcal{QA}''_J, q\mathcal{A}_I} \ ; \ \underbrace{Context_I}_{\mathcal{A}'_J, \mathcal{QA}'_J, \mathcal{CA}_I} \ ; \ \mathcal{E}_I}\, I}_{\vdash \underbrace{\mathcal{A}'_J, \mathcal{A}''_J}_{\mathcal{A}^1_J} \ ; \ \underbrace{\mathcal{QA}'_J, \mathcal{QA}''_J}_{\mathcal{QA}^1_J} \ ; \ \underbrace{\mathcal{CA}_I, q\mathcal{A}_I, \mathcal{P}_I, \mathcal{E}_I}_{Context_J} \ ;} \qquad \underbrace{\dfrac{\vdash \mathcal{A}_L \ ; \ \mathcal{QA}_L \ ; \ Context_L \ ;}{\vdash \mathcal{P}_L \ ; \ \underbrace{\mathcal{QA}_L}_{\mathcal{B}''_J, \mathcal{QB}''_J, q\mathcal{B}_L} \ ; \ \underbrace{Context_L}_{\mathcal{B}'_J, \mathcal{QB}'_J, \mathcal{CB}_L} \ ; \ \mathcal{E}_L}\, L}_{\vdash \underbrace{\mathcal{B}'_J, \mathcal{B}''_J}_{\mathcal{A}^2_J} \ ; \ \underbrace{\mathcal{QB}'_J, \mathcal{QB}''_J}_{\mathcal{QA}^2_J} \ ; \ \underbrace{\mathcal{CB}_L, q\mathcal{B}_L, \mathcal{P}_L, \mathcal{E}_L}_{Context_J} \ ;}}{\vdash \mathcal{P}_J \ ; \ \mathcal{QA}^1_J, \mathcal{QA}^2_J \ ; \ \mathcal{C}_I, q\mathcal{A}_I, \mathcal{P}_I, \mathcal{E}_I \ ; \ \mathcal{E}_J}\, J$$

Further, verification of the tests: $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(1,1), Context_J)$ and $\mathcal{N}_{I,J,\Pi}(1,1) \ = \ \mathcal{N}_{L,J,\Pi}(1,2)$ enables application of J on the premises of the permutation figure as follows:

$$\dfrac{\underbrace{\vdash \mathcal{A}_I \ ; \mathcal{QA}_I \ ; \ Context_I}_{\vdash \mathcal{A}^1_J \ ; \mathcal{QA}^1_J \ ; \ \mathcal{A}_I, q\mathcal{A}_I, \mathcal{CA}_I} \qquad \underbrace{\vdash \mathcal{A}_L \ ; \mathcal{QA}_L \ ; \ Context_L}_{\vdash \mathcal{A}^2_J \ ; \mathcal{QA}^2_J \ ; \ \underbrace{\mathcal{A}_L, q\mathcal{B}_L, \mathcal{CB}_L}_{\mathcal{A}_I, q\mathcal{A}_I, \mathcal{CA}_I}}}{\vdash \underbrace{\mathcal{P}_J \ ; \ \mathcal{QA}^1_J, \mathcal{QA}^2_J \ ; \ \mathcal{A}_I, q\mathcal{A}_I, \mathcal{CA}_I \ ; \ \mathcal{E}_J}_{\mathcal{A}_I ; \mathcal{QA}_I ; \mathcal{P}_J, e_J, \mathcal{QA}'_J, \mathcal{QA}^2_J, \mathcal{CA}_I}}\, J$$

Let us denote the above subproof by $\mathcal{D}$. Further, the application of I on the result of $\mathcal{D}$ is provided by verification of the test $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I)$: multiset $[Context_I, \mathcal{E}_J] \backslash \mathcal{A}^1_J$, $\mathcal{P}_J, \mathcal{QA}^2_J, Context^2_J =$ $\mathcal{P}_J, e_J, \mathcal{QA}'_J, \mathcal{QA}^2_J, \mathcal{CA}'_J$ can be unified with $Context_I$.

$$\dfrac{\left.\begin{array}{c} \vdots \\ \vdash \mathcal{A}_I \ ; \ \mathcal{QA}_I \ ; \ \mathcal{P}_J, e_J, \mathcal{QA}'_J, \mathcal{QA}^2_J, \mathcal{CA}_I \end{array}\right\} \mathcal{D}}{\vdash \mathcal{P}_I \ ; \ \mathcal{QA}_I \ ; \ \mathcal{P}_J, e_J, \mathcal{QA}'_J, \mathcal{QA}^2_J, \mathcal{CA}'_J \ ; \ \mathcal{E}_I}\, I$$

The equality of the end-sequents is fulfilled automatically and can be checked directly.

□

6.3.4 Binary-binary rules

For binary rules I and J let us denote their permutation object by F_0 and possible forms of permutation results respectively by $F_{4.1}$, $F_{4.2}$ and $F_{4,3}$ [7]:

$$F_0: \dfrac{\dfrac{\stackrel{\vdots}{\gamma_1} \quad \stackrel{\vdots}{\gamma_2}}{\beta_1}\,I \qquad \stackrel{\vdots}{\beta_2}}{\alpha}\,J \qquad F_{4.1}: \dfrac{\dfrac{\stackrel{\vdots}{\gamma_1} \quad \stackrel{\vdots}{\beta_2}}{\delta_1}\,J \qquad \stackrel{\vdots}{\gamma_2}}{\alpha}\,I \qquad F_{4.2}: \dfrac{\stackrel{\vdots}{\gamma_1} \qquad \dfrac{\stackrel{\vdots}{\gamma_2} \quad \stackrel{\vdots}{\beta_2}}{\delta_2}\,J}{\alpha}\,I \qquad F_{4.3}: \dfrac{\dfrac{\stackrel{\vdots}{\gamma_1} \quad \stackrel{\vdots}{\beta_2}}{\delta_1}\,J \qquad \dfrac{\stackrel{\vdots}{\gamma_2} \quad \stackrel{\vdots}{\beta_2}}{\delta_2}\,J}{\alpha}\,I$$

Theorem 6.10
A multiplicative rule I is permutable over a multiplicative rule J, giving a permutation result of the form: 1) $F_{4.k}$, $k \in \{1,2\}$
2) $F_{4.3}$

iff they verify the tests:

1) $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1)$, and $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k)$
2) $\forall k \in \{1,2\}$ $\mathbf{A}_{I,J,\Pi}(\mathrm{k})$, $\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1)$, $\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k)$, and $\mathbf{D}(J,1)$

Proof: similar to proof of Theorem 6.3.

□

As an illustration, consider the following examples from linear logic:
1.

$$\dfrac{\dfrac{\vdash A,\Gamma,C \quad \vdash B,\Delta}{\vdash A\otimes B,\Gamma,C,\Sigma}\otimes \qquad \vdash C^{\perp},\Sigma}{\vdash A\otimes B,\Gamma,\Delta,\Sigma}\,Cut \qquad \Longleftrightarrow \qquad \dfrac{\dfrac{\vdash A,\Gamma,C \quad \vdash C^{\perp},\Sigma}{\vdash A,\Gamma,\Sigma}\,Cut \qquad \vdash B,\Delta}{\vdash A\otimes B,\Gamma,\Delta,\Sigma}\otimes$$

2.

$$\dfrac{\dfrac{\vdash A,B \quad \vdash A,C}{\vdash A,A,B\otimes C}\otimes \qquad \vdash A,A^{\perp}}{\vdash A,B\otimes C,A}\,Cut \qquad \Longleftrightarrow \qquad \dfrac{\dfrac{\vdash A,B \quad \vdash A,A^{\perp}}{\vdash A,B}\,Cut \qquad \dfrac{\vdash A,C \quad \vdash A,A^{\perp}}{\vdash A,C}\,Cut}{\vdash A,B\otimes C,A}\otimes$$

Theoretically, special-permutation of a multiplicative rule I over a multiplicative rule J (transformation of the permutation figure into the permutation result as shown below)

$$\begin{array}{c} I\ L \\ J \end{array} \quad \mapsto \quad \begin{array}{c} J \\ I \end{array}$$

is possible in a certain situations, as specified in the theorem below.

[7] recall that we assume that I is above the first premise of J in the permutation object(figure)

Theorem 6.11
A multiplicative rule I is special-permutable over a multiplicative rule J iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$$\begin{array}{l}
type(I) = type(L) \\
\mathbf{A}_{I,J,\Pi}(1), \\
\mathbf{A}_{L,J,\Pi}(1), \\
\forall k \in \{1,2\}: \\
\quad \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1\ ) \\
\quad \mathbf{STRUCT}(\mathcal{N}_{L,J,\Pi}(k,2), Context_J^2\ ) \\
\quad \mathbf{STRUCT}([Context_I^k, \mathcal{E}_J]\backslash\mathcal{A}_J^1, \mathcal{P}_J, \mathcal{Q}\mathcal{A}_J^2, \mathcal{A}_L^k, (\mathcal{Q}\mathcal{A}_L^k, Context_L^k)\backslash\mathcal{A}_J^2,\ Context_I^k) \\
\mathcal{A}_J \ = \ \mathcal{A}_L, \mathcal{E}_J \quad \wedge \quad \mathcal{P}_L, \mathcal{E}_L = \mathcal{P}_J .
\end{array}$$

Proof: similar to proof of Theorem 6.9.

□

Theorem 6.12
A multiplicative rule I is permutable over a strong-additive rule J, giving a permutation result of the form: 1) $F_{4.k},\ k \in \{1,2\}$
2) $F_{4.3}$
iff they verify the tests:

$$\begin{array}{lll}
1) & \mathbf{A}_{I,J,\Pi}(\mathrm{k}), & \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J),\ \ \mathcal{N}_{I,J,\Pi}(k,1) \ = \ Context_J \\
 & & \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k),\ \ \mathbf{D}(I,k) \\
2) & \forall k \in \{1,2\} & \\
 & \mathbf{A}_{I,J,\Pi}(\mathrm{k}), & \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J),\ \ \mathcal{N}_{I,J,\Pi}(1,1) \ = \ \mathcal{N}_{I,J,\Pi}(2,1), \\
 & & \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k),\ \ \mathbf{D}(I,k), \qquad \mathbf{D}(J,1)
\end{array}$$

Proof: similar to proof of Theorem 6.3.

□

Example:

$$\dfrac{\dfrac{\vdash A,\Gamma,C \qquad \vdash A,A^{\perp}}{\vdash A,\Gamma,C}\,Cut \qquad \vdash D,A,\Gamma}{\vdash C\&D,A,\Gamma}\,\& \qquad \Longleftrightarrow \qquad \dfrac{\dfrac{\vdash A,\Gamma,C \qquad \vdash D,A,\Gamma}{\vdash C\&D,A,\Gamma}\,\& \qquad \vdash A,A^{\perp}}{\vdash C\&D,A,\Gamma}\,Cut$$

As we have seen the “normal” permutation (permutation object $\mapsto$ permutation result) of a multiplicative rule I over a strong-additive rule J requires redundant steps of I and/or J. According to Theorem 6.12 such redundant application(s) of rules in question, are among the necessary and sufficient conditions for the permutation. If we eliminate redundant steps prior any further proof transformation then, as a consequence of Theorem 6.12, we come to the statement :

None of the multiplicative rules is permutable over a strong-additive rule.

But, again this non-permutability is relative and can be overcome by special permutations (permutation figure $\mapsto$ permutation result). It is specified and formalised in the theorems that follow:

Theorem 6.13
A multiplicative rule I is special-permutable over a strong-additive rule J iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$\forall k \in \{1,2\}$:
$\mathbf{A}_{I,J,\Pi}(\mathrm{k})$,
$\mathbf{A}_{L,J,\Pi}(\mathrm{k})$,
$\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context_J^1)$,
$\mathbf{STRUCT}(\mathcal{N}_{L,J,\Pi}(k,2), Context_J^2)$,
$\mathcal{N}_{I,J,\Pi}(k,1) = \mathcal{N}_{L,J,\Pi}(k,2)$,
$\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k,1), Context_I^k)$,
$\mathbf{D}(J,1)$

Proof: similar to proof of Theorem 6.9.

□

The last test in the above Theorem implies that the corresponding instance of J must be a redundant step. By eliminating redundant steps prior any other proof transformation, as a consequence of the above Theorem, we come to the statement:

None of the permutation figures of the form: $\dfrac{\dfrac{\overset{\vdots}{\alpha_1}\quad\overset{\vdots}{\alpha_2}}{\gamma_1}\,I \qquad \dfrac{\overset{\vdots}{\beta_1}\quad\overset{\vdots}{\beta_2}}{\gamma_2}\,L}{\delta}\,J$ **can be transformed (by permutation) into the permutation result:** $\dfrac{\dfrac{\overset{\vdots}{\alpha_1}\quad\overset{\vdots}{\beta_1}}{\gamma_1'}\,J \qquad \dfrac{\overset{\vdots}{\alpha_2}\quad\overset{\vdots}{\beta_2}}{\gamma_2'}\,J}{\delta}\,I$.

But, according to the next Theorem, it is possible to transform a permutation figure of the form: $\dfrac{\dfrac{\overset{\vdots}{\alpha_1}\quad\overset{\vdots\,\mathcal{D}}{\alpha}}{\gamma_1}\,I \qquad \dfrac{\overset{\vdots}{\beta_1}\quad\overset{\vdots\,\mathcal{D}}{\alpha}}{\gamma_2}\,L}{\delta}\,J$ into the permutation result: $\dfrac{\dfrac{\overset{\vdots}{\alpha_1}\quad\overset{\vdots}{\beta_1}}{\gamma}\,J \qquad \overset{\vdots\,\mathcal{D}}{\alpha}}{\delta}\,I$, as illustrated by the example below:

$$\dfrac{\dfrac{\overset{\vdots}{\vdash A,C,\Delta}\quad\overset{\vdots\,\mathcal{D}}{\vdash B,\Gamma}}{\vdash A\otimes B,C,\Delta,\Gamma}\otimes \qquad \dfrac{\overset{\vdots}{\vdash D,A,\Delta}\quad\overset{\vdots\,\mathcal{D}}{\vdash B,\Gamma}}{\vdash A\otimes B,D,\Delta,\Gamma}\otimes}{\vdash C\&D,A\otimes B,\Delta,\Gamma}\& \qquad \Longleftrightarrow \qquad \dfrac{\dfrac{\overset{\vdots}{\vdash A,C,\Delta}\quad\overset{\vdots}{\vdash D,A,\Delta}}{\vdash C\&D,A,\Delta}\& \qquad \overset{\vdots\,\mathcal{D}}{\vdash B,\Gamma}}{\vdash C\&D,A\otimes B,\Delta,\Gamma}\otimes$$

The illustrated special-permutation can be specified as follows:

Theorem 6.14
A multiplicative rule I is special-permutable over a strong-additive rule J iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$\exists s,t \in \{1,2\}$
$\mathbf{A}_{I,J,\Pi}(\mathrm{s})$,
$\mathbf{A}_{L,J,\Pi}(\mathrm{t})$,
$\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(s,1), Context_J^1)$,
$\mathbf{STRUCT}(\mathcal{N}_{L,J,\Pi}(t,2), Context_J^2)$,
$\mathcal{N}_{I,J,\Pi}(s,1) = \mathcal{N}_{L,J,\Pi}(t,2)$,
$\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(s,1), Context_I^s)$.

Proof: similar to proof of Theorem 6.9.

□

Theorem 6.15
A strong-additive rule I is permutable over a multiplicative rule J giving a permutation result of the form: 1) $F_{4.3}$

2) $F_{4.k}, \ k \in \{1,2\}$

iff they verify the tests:

1) $\forall k \in \{1,2\}$

$\mathbf{A}_{I,J,\Pi}(\mathrm{k}), \quad \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context^1_J), \quad \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I)$

2) $\mathbf{A}_{I,J,\Pi}(\mathrm{k}), \quad \mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context^1_J), \quad \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I), \quad \mathbf{D}(J,1)$

Proof: similar to proof of Theorem 6.3.

□

Example:

$$\dfrac{\dfrac{\vdash A, C, \Gamma \quad \vdash A, D, \Gamma}{C\&D, A, \Gamma}\,\& \qquad \vdash B, \Delta}{\vdash A \otimes B, C\&D, \Gamma, \Delta}\,\otimes \quad \Longleftrightarrow \quad \dfrac{\dfrac{\vdash A, C, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, C, \Gamma, \Delta}\,\otimes \qquad \dfrac{\vdash A, D, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, D, \Gamma, \Delta}\,\otimes}{\vdash A \otimes B, C\&D, \Gamma, \Delta}\,\&$$

As the following theorems show, special-permutation requires redundant steps:

Theorem 6.16
A strong-additive rule I is special-permutable over a multiplicative rule J iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$\forall k \in \{1,2\}$

$\mathbf{A}_{I,J,\Pi}(\mathrm{k})$,

$\mathbf{A}_{L,J,\Pi}(\mathrm{k})$,

$\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k,1), Context^1_J)$,

$\mathbf{STRUCT}(\mathcal{N}_{L,J,\Pi}(k,2), Context^2_J)$,

$\mathbf{STRUCT}(\ [Context^k_I, \mathcal{E}_J]\backslash\mathcal{A}^1_J,\ \mathcal{P}_J, \mathcal{QA}^2_J, \mathcal{A}^k_L, (\mathcal{QA}^k_L, context_L)\backslash(\mathcal{A}^2_J, \mathcal{QA}^2_J),\ Context^k_I)$

$\mathcal{A}^1_L\ ,\ (\mathcal{QA}^1_L, context_L)\backslash(\mathcal{A}^2_J, \mathcal{QA}^2_J)\ =\ \mathcal{A}^2_L\ ,\ (\mathcal{QA}^2_L, context_L)\backslash(\mathcal{A}^2_J, \mathcal{QA}^2_J)$

$\mathbf{D}(L,1)$

Proof: similar to proof of Theorem 6.9.

□

Theorem 6.17
A strong-additive rule I is permutable over a strong-additive rule J giving a permutation result of the form: 1) $F_{4.3}$

2) $F_k, \ k \in \{1,2\}$

iff they verify the tests :

1) $\forall k \in \{1,2\}$ $\mathbf{A}_{I,J,\Pi}(\mathrm{k}), \quad \mathbf{D}(I,k) \quad \mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(1,1), Context_I)$

2) $\mathbf{A}_{I,J,\Pi}(\mathrm{k}), \quad \mathbf{D}(I,k) \quad \mathbf{D}(J,1)$

Proof: similar to proof of Theorem 6.3.

□

As we have seen "normal" permutation (permutation object $\mapsto$ permutation result) of a strong-additive rule I over a strong-additive rule J requires redundant steps of I and/or J. By eliminating redundant steps before any further proof transformation, as a consequence of the above theorem then we have the statement :

None of the strong-additive rules is permutable over a strong-additive rule.

But, again this non-permutability is relative and can be overcome by special permutations (permutation figure $\mapsto$ permutation result). It is specified and formalised in the theorem below, and illustrated by the example from linear logic. By labelling distinct applications of the & rule, we can see the "special permutation":

$$\dfrac{\dfrac{\vdash A, C, D \quad \vdash B, C, D}{\vdash A \wedge B, C, D}\,\&_1 \qquad \dfrac{\vdash A, D, D \quad \vdash B, D, D}{\vdash A \wedge B, D, D}\,\&_2}{\vdash A \wedge B, C\&D, D}\,\&_3 \quad \Longleftrightarrow \quad \dfrac{\dfrac{\vdash A, C, D \quad \vdash A, D, D}{\vdash C\&D, A, D}\,\&_3 \qquad \dfrac{\vdash B, C, D \quad \vdash B, D, D}{\vdash C\&D, B, D}\,\&_3}{\vdash C\&D, A \wedge B, D}\,\&_1$$

Theorem 6.18
A strong-additive rule I is special-permutable over a strong-additive rule J iff they verify the tests (let L denotes the other rule which precedes J in a given proof):

$\forall k \in \{1, 2\}$

$\mathbf{A}_{I,J,\Pi}(\mathrm{k})$,

$\mathbf{A}_{L,J,\Pi}(\mathrm{k})$,

$\mathbf{STRUCT}(\mathcal{N}_{I,J,\Pi}(k, 1), Context_J)$,

$\mathcal{N}_{I,J,\Pi}(k, 1) = \mathcal{A}_L^k \ , \ (\mathcal{Q}\mathcal{A}_L^k, Context_L)\backslash\mathcal{A}_J^2$

$\mathbf{STRUCT}(\mathcal{M}_{I,J,\Pi}(k, 1), Context_I)$

Proof: similar to proof of Theorem 6.9.

□

According to the above theorems, possible forms of the permutation result for a given pair of rules I, J can be summarised as in Table 5.

Table 5

	I		
J	unary	multiplicative	strong-additive
unary	F	$F_{2.1}$ or $F_{2.2}$ $F_{2.3}$ requires redundant steps in perm.object	$F_{2.3}$ $F_{2.1}$ or $F_{2.2}$ require redundant steps in perm.object
multiplicative	F	$F_{4.1}$ or $F_{4.2}$ $F_{4.3}$ requires redundant steps in perm.object	$F_{4.3}$ $F_{4.1}$ or $F_{4.2}$ require redundant steps in perm.object
strong-additive	special permutation otherwise there must be redundant steps in perm.object	special permutation otherwise there must be redundant steps in perm.object	special permutation otherwise there must be redundant steps in perm.object

7 Some applications of permutations and permutation analysis

7.1 Defining proof search procedures

The permutation properties of the inference rules of a given logical fragment have a direct impact on defining efficient proof search strategies which deal with the order of inference rules. We explain this impact through the so-called *"Normal proof concept"*[13], a strategy for proof construction that are applicable in logic programming [33], [38], [46], [47]. It imposes constraints for the choice of formula to be decomposed in the next step of the proof construction. These constraints are based on the permutability of inference rules and contribute to reduction of some sources of non-determinism during the proof construction. In a *bottom-up proof construction* (in which search starts at the root of the tree) this approach always first applies rule for which there is a "guarantee" that in the case of a provable sequent there is a proof with that rule in that particular place in the proof tree. Also, it is important to detect groups of rules that are easy to apply from the conclusion (when are applying in a bottom up direction). It is desirable to apply such rules *as soon as possible* in a proof construction and to postpone the others (which introduce don't know non-determinism when applied in up direction) *as late as possible.* The degree of the realization of this aim *to apply a rule I as soon as possible* or *to apply a rule I as late as possible* is directly dependent on the permutability relationships between I and other inferences in a given logical fragment *i.e.* in a given proof. Let us illustrate that by an example.

Let us consider the sequent $\vdash A \otimes B, C \wp D, \Gamma$. During proof construction we must have a complete strategy for selecting the next formula to be decomposed. This selection is crucial for bottom-up proof construction. On the basis of the permutation results given in [13] we have the following table, where the case (t_1, t_2) in the table contains p iff for any inferences I_1 and I_2 of type t_1 and t_2 being in weak permutation position in a proof, I_1 is permutable over I_2. The case (t_1, t_2) in the table contains np iff there exists two inferences I_1 and I_2 of type t_1 and t_2 being in weak permutation position in a proof, which are not permutable:

	t_1		
t_2	$\otimes$	$\oplus$	$\wp$
$\otimes$	p	p	p
$\oplus$	p	p	p
$\wp$	np	p	p

As a consequence of np for the case $(\otimes, \wp)$ and p for the case $(\wp, \otimes)$, for any provable sequent $\vdash A \otimes B, C \wp D, \Gamma$ there is a proof where $\wp$ is applied before (closer to the root than) the $\otimes$ rule, but not necessary the proof with $\otimes$ applied before $\wp$. For instance, the proof construction of the sequent $\vdash a \otimes b, a^\perp \wp b^\perp$ terminates successfully only when $\wp$ precedes $\otimes$:

$$\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax \qquad \dfrac{}{\vdash b, b^\perp}\,Ax}{\vdash a \otimes b, a^\perp, b^\perp}\,\otimes}{\vdash a \otimes b, a^\perp \wp b^\perp}\,\wp \qquad \dfrac{\dfrac{\dfrac{?}{\vdash a, a^\perp, b^\perp}}{\vdash a, a^\perp \wp b^\perp}\,\wp \qquad \dfrac{?}{\vdash b}\,\wp}{\vdash a \otimes b, a^\perp \wp b^\perp}\,\otimes \qquad \dfrac{\dfrac{?}{\vdash a}\,\wp \qquad \dfrac{\dfrac{?}{\vdash b, a^\perp, b^\perp}}{\vdash b, a^\perp \wp b^\perp}\,\wp}{\vdash a \otimes b, a^\perp \wp b^\perp}\,\otimes$$

while in a proof construction of the sequent $\vdash a \otimes b, a^\perp \wp ?b^\perp, b^\perp$, the order in which the rules are applied, does not matter:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?b^\perp}\,w? \qquad \dfrac{}{\vdash b, b^\perp}\,Ax}{\vdash a \otimes b, a^\perp, ?b^\perp, b^\perp}\,\otimes}{\vdash a \otimes b, a^\perp \wp ?b^\perp, b^\perp}\,\wp \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a, a^\perp}\,Ax}{\vdash a, a^\perp, ?b^\perp}\,w?}{\vdash a, a^\perp \wp ?b^\perp}\,\wp \qquad \dfrac{}{\vdash b, b^\perp}\,Ax}{\vdash a \otimes b, a^\perp \wp ?b^\perp, b^\perp}\,\otimes$$

Thus, in the proof construction of a sequent $\vdash A \otimes B, C \wp D, \Gamma$ the choice to decompose $C \wp D$ first has a greater chance of being successful (*i.e.* choice that lead to a proof).

The rules $\otimes$ and $\oplus$ have identical behaviour (from permutability point of view) towards each other ($p(\otimes, \oplus)$ and $p(\oplus, \otimes)$) and from that prospective they are equal in the proof construction of a sequent $\vdash A \otimes B, C \oplus D$. On the other hand both rules correspond to "do not know" nondeterminism that is difficult to reduce. Regarding the existence of lazy methods for the treatment of $\otimes$ [47], it seems that there is a slightly stronger reason to apply $\otimes$ before $\oplus$.

7.2 Strategy Analysis

Permutation properties can be used to either prove that a given search strategy will find all possible proofs which use a particular subset of inference rules, or construct an example of a provable sequent which can not be proved with that strategy. This is specific for strategies which deal with the order of inference rules. Strategies which pass this test(s) may be amenable to further analysis, such as minimising the amount of branching in the proof, or delaying certain choices until the optimum amount of information is available.

We explain this on a simplified version of *"normal proof"* strategy whose full version is given in [13]. Let assume that, in the logical fragment we consider, the sequents are one-sided: $\vdash \Gamma$. Γ is a multiset of goal formulas, denoted G and defined by the following grammar:

$$G := A|G \otimes G|G \wp G|G \oplus G|\forall x G|\exists x G|?G$$

The allowed set of inference rules for proving the specified sequents is $\{\otimes, \wp, \oplus_1, \oplus_2, \forall, \exists, w?\}$. The strategy is defined by the following algorithm:

If $\vdash \Gamma$ contains a formula of the form $?F$
then $\vdash \Gamma$ is the conclusion of $w?$ rule
else
if $\vdash \Gamma$ contains a formula whose top-most connective is $\wp$ or $\forall$
then $\vdash \Gamma$ is the conclusion of rule which introduces that particular formula
else $\vdash \Gamma$ contains a formula whose top-most connective is $\otimes$ or $\oplus$ or $\exists$ and
$\vdash \Gamma$ is the conclusion of rule which introduces that particular formula

The completeness properties of the outlined strategy follow directly from the permutability properties of the concerned linear logic fragment:

	t_1					
t_2	$\wp$	$\forall$	$\otimes$	$\oplus$	$\exists$	$w?$
$\wp$	p	p	np	p	p	p
$\forall$	p	p	p	p	np	p
$\otimes$	p	p	p	p	p	p
$\oplus$	p	p	p	p	p	p
$\exists$	p	p	p	p	p	p
$w?$	p	p	p	p	p	p

Any proof, from this logical fragment, can be transformed, by permutations of its inferences, to a proof that obeys the given strategy:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\vdash p(t), p^{\perp}(t)$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\vdash a, a^{\perp}$}
\RightLabel{$w?$}
\UnaryInfC{$\vdash a, a^{\perp}, ?b$}
\RightLabel{$\wp$}
\UnaryInfC{$\vdash a, a^{\perp} \wp ?b$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash p(t) \otimes a, p^{\perp}(t), ?b \wp a^{\perp}$}
\RightLabel{$\exists$}
\UnaryInfC{$\vdash p(t) \otimes a, \exists y p^{\perp}(y), ?b \wp a^{\perp}$}
\RightLabel{$\forall$}
\UnaryInfC{$\vdash \forall x(p(x) \otimes a), \exists y p^{\perp}(y), ?b \wp a^{\perp}$}
\end{prooftree}
\quad \longmapsto \quad
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\vdash a, a^{\perp}$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\vdash p(t), p^{\perp}(t)$}
\RightLabel{$\exists$}
\UnaryInfC{$\vdash p(t), \exists y p^{\perp}(y)$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash p(t) \otimes a, \exists y p^{\perp}(y), a^{\perp}$}
\RightLabel{$w?$}
\UnaryInfC{$\vdash p(t) \otimes a, \exists y p^{\perp}(y), ?b, a^{\perp}$}
\RightLabel{$\wp$}
\UnaryInfC{$\vdash p(t) \otimes a, \exists y p^{\perp}(y), ?b \wp a^{\perp}$}
\RightLabel{$\forall$}
\UnaryInfC{$\vdash \forall x(p(x) \otimes a), \exists y p^{\perp}(y), ?b \wp a^{\perp}$}
\end{prooftree}
$$

7.3 Detection of permutability properties by analysing schema rules

In [13] Galmiche and Perrier have studied the permutability properties of linear logic inference rules, and exploited them for the process of proof construction. They give results about permutability properties (Theorem 3.2 of [13]) which are based on the analysis of the appropriate schema-rules. The existing analysis is concerned with the analysing of permutabilities on paper by a human. Thus, for example, by analysing and comparing of schema rules:

$$
\frac{\vdash F, ?\Gamma}{\vdash !F, ?\Gamma}\ ! \qquad \frac{\vdash A, B, \Gamma}{\vdash A \wp B, \Gamma}\ \wp
$$

it is not hard to notice that the principal formula of rule $\wp$, can not be a part of the *Context* for the rule !. The only possible subproof where the rule $\wp$ is followed by the rule ! must has the form as shown below:

$$
\begin{prooftree}
\AxiomC{$\vdash A, B, ?\Gamma$}
\RightLabel{$\wp$}
\UnaryInfC{$\vdash A \wp B, ?\Gamma$}
\RightLabel{$!$}
\UnaryInfC{$\vdash !(A \wp B), ?\Gamma$}
\end{prooftree}
$$

Hence, rules $\wp$ and ! can never be in the strong (or weak) permutation position. Thus, by analysing of schema-rules $\wp$ and !, independently of concrete proofs, we extract an useful permutability property: the rule $\wp$ is *never permutable* over the rule !.

The studying of schema-rules involves many details that are ripe for automation and details that can be generalised and extended on new, possible situations. So, the naturally-arising question is the automation of the analysis of a given set of schema-rules, independently of concrete inferences and proofs. In the rest of this subsection we identify some categories of schema rules which have a particular behaviour regarding the permutability properties.

Definition 7.1 *(never-permutable, extra-permutable rules)*
Rules I and J are *never-permutable* rules (denoted $n(I, J)$), if there can be no proof in which I is permutable over J.
Rules I and J are *extra-permutable* rules, (denoted $e(I, J)$), if whenever I and J are adjacent in a proof, I is permutable over J.

For instance, in linear logic, rules $\sharp$-R and $w?$-R are extra-permutable rules, iff $\sharp$-R is one of $\oplus$-R, $\wp$-R, $\otimes$-R, $\bot$-R, $w?$-R, !-R, ?-R, $c?$-R, $\forall$-R, and $\exists$-R. A never-permutable pair of rules is, for example, the pair ! and $\wp$, or the pair $\exists$ and !.

Identifying of pairs of never-permutable and extra-permutable rules, by analysing the appropriate schema rules, eliminate checking of any permutation condition (see definition 5.3) in a concrete proof. The following lemmas identify some situations for the recognition of never-permutable pair(s) of rules.

Lemma 7.2 *(never-unifiable multisets)*
Let ${}_{top/constant/atom}\Gamma_{forbidden}$ and ${}_{top/constant/atom}\Delta_{forbidden}$ be given schema multisets. If they verify the conditions:

1. $\textit{ATOM}(\Gamma)\cdot\textit{ATOM}(\Delta) = 0$
2. $\textit{CONSTANT}(\Gamma) \cap \textit{CONSTANT}(\Delta) = \emptyset$
3. $\textit{PERMITTED}(\Gamma) \cap \textit{PERMITTED}(\Delta) \neq \emptyset \;\Rightarrow\; \left[\textit{TOP}(\Delta) \neq \emptyset \;\wedge\; \textit{TOP}(\Gamma) \cap \textit{TOP}(\Delta) = \emptyset\right]$

then there can be no instantiation such that $\Gamma \subseteq \Delta$ (we say Γ and Δ are *never-unifiable* multisets).

Proof:

If the structure of any possible formula in a multiset Γ can not be accepted for a multiset Δ then there is no instantiation such that $\Gamma \subseteq \Delta$. That is the case when any connective that is allowed (if any) in Γ is forbidden in Δ (so any formula which contains that particular connective can never appear in Δ) or there is a connective that can appear in both Γ and Δ but the observed multisets have disjunctive requirements for the top-most connectives. This can be formalised as:

$$\textit{PERMITED}(\Gamma) \cap \textit{PERMITED}(\Delta) \neq \emptyset \;\Rightarrow\; \left[\textit{TOP}(\Delta) \neq \emptyset \;\wedge\; \textit{TOP}(\Gamma) \cap \textit{TOP}(\Delta) = \emptyset\right]$$

Also, atomic formulas can not appear (as immediate formulas) in both Γ and Δ, which is equivalent to the condition: $\textit{ATOM}(\Gamma)\cdot\textit{ATOM}(\Delta) = 0$ and finally, sets of logical constants that can appear (as immediate formulas) in both Γ and Δ are disjunctive: $\textit{CONSTANT}(\Gamma) \cap \textit{CONSTANT}(\Delta) = \emptyset$

□

Lemma 7.3
For given schema-rules I and J:

If $\mathcal{P}_I$ and/or $\mathcal{E}_I$ are *never unifiable* with $Context_J$ then $n(I,J)$ and $n(J,I)$.

Proof: For a I,J permutation object:

$$\dfrac{\underbrace{\dfrac{\vdots}{\vdash \mathcal{P}_I \;;\; \underbrace{\mathcal{QA}_I}_{\mathcal{A}''_J,\mathcal{QA}''_J,q\mathcal{A}_I} \;;\; \underbrace{Context_I}_{\mathcal{A}'_J,\mathcal{QA}'_J,\mathcal{C}_I} \;;\; \mathcal{E}_I}\,I}_{\vdash \underbrace{\mathcal{A}'_J,\mathcal{A}''_J}_{\mathcal{A}^1_J} \;;\; \underbrace{\mathcal{QA}'_J,\mathcal{QA}''_J}_{\mathcal{QA}^1_J} \;;\; \underbrace{\mathcal{C}_I,q\mathcal{A}_I,\mathcal{P}_I,\mathcal{E}_I}_{Context^1_J} \;;} \qquad \dfrac{\vdots}{\vdash \mathcal{A}^2_J \;;\; \mathcal{QA}^2_J \;;\; Context^2_J}\,L}{\vdash \mathcal{P}_J \;;\; \mathcal{QA}_J, \;;\; Context_J \;;\; \mathcal{E}_J}\,J$$

for the $Context^1_J$ holds: $Context^1_J = \mathcal{P}_I,\mathcal{E}_I,(Context_I,\mathcal{QA}_I)\backslash(\mathcal{A}_J,\mathcal{QA}_J)$. Since $\mathcal{P}_I$ and/or $\mathcal{E}_I$ are never unifiable with $Context_J$, the permutation object I,J can not exist. Also, in the J,I permutation object (if it exists) it is not possible to invert the rules (on the basis of Proposition 6.1) .

□

Extra-permutable rules implies pairs of rules which verify the conditions:

1. whenever they are adjacent in a proof, they are in the strong permutation position;
2. either there is no restriction on the structure of $Context_J$, or $\mathcal{A}_I$ can *always* be *unified* with $Context_J$;
3. either there is no restriction on the structure of $Context_I$, or $\mathcal{E}_J$ can *always* be *unified* with $Context_I$;
4. either J is multiplicative rule, or

   J is weak additive rule such that $\mathcal{P}_I,\mathcal{E}_I$ can be unified only with $\mathcal{FREE}(Context_J)$.

Condition 1. insures that I and J are in (strong) permutation position every time when they are adjacent in a proof. Conditions 2. and 3. are consequences of Proposition 6.1 and provide the application of J on the premise of I and then the application of I on the generated end-sequent.

Condition 4. eliminates additive rules J as well as those weak additive rules J where $\mathcal{P}_I, \mathcal{E}_I$ can belong to the part of $Context_J$ which must be the same in both premises and conclusion, because such a situation requires the special-permutation of I over J and, as we have already seen in subsection 6.3, this depends on the concrete proof.

Regarding the proposed specification of forms of inference rules it is possible to notice the following connections between the structure of rules and (strong) permutation position:

Proposition 7.4

1. A rule which verifies: $\mathcal{A} = \mathcal{QA} = \mathcal{P} = \emptyset \ \wedge \ \mathcal{E} \neq \emptyset$ is always in a strong permutation position with a rule that precedes it in a proof.

2. A rule which verifies: $\mathcal{QA} = \mathcal{P} = \mathcal{E} = \emptyset$ is always in a strong permutation position with a rule that follows it in a proof.

3. A rule which verifies: $\mathcal{A} = \mathcal{QA} = \mathcal{P} = \mathcal{E} = \emptyset$ (Mix-rule) is always in a strong permutation position with a rules that are immediate above and below it in a subproof.

4. Rules I and J for which $\mathcal{A}_J, \mathcal{QA}_J$ is not unifiable with $\mathcal{P}_I, \mathcal{E}_I, \mathcal{QA}_I$ are in a strong permutation position, whenever they are adjacent in a proof.

5. A rule which verifies: $\mathcal{QA} = \emptyset \ \wedge \ Context = \emptyset$ is never in a weak ($\Rightarrow$ never in strong) permutation position with a rule that precedes it in a proof.

Proof: straightforward

□

Let us illustrate "groups" of rules specified by the above theorem:

1.

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?F}\ ?w\ (\text{CLL}) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, F}\ w\ (\text{LK}) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\ \bot\ (\text{CLL})$$

from $\mathcal{SC}$ 3.4:

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta, A_? \otimes B}\ \otimes_{lr} \qquad \frac{\vdash \Gamma_l \qquad \vdash \Gamma_r}{\vdash \Gamma_{lr}, A_? \otimes B}\ \&_{lr} \qquad \frac{\vdash ?\Gamma}{\vdash ?\Gamma, !A}\ ! - storage$$

2.

$$\frac{\vdash A, \Gamma \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ \text{Cut}$$

3.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\ \text{mix}$$

4. (from LLL)

$$I \ = \frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes \qquad J \ = \frac{\vdash \Gamma, A_1 \oplus A_2 \oplus \ldots \oplus A_n}{\vdash \Gamma, A_1 \oplus A_2 \oplus \ldots \oplus A_n \oplus B}\ \text{A-weakening}$$

5.

$$\frac{\vdash A, \Delta}{\vdash \Box A, \Diamond\Delta}\ \Box + jump\ \ (\text{KT}^{\mathcal{S}}\text{ - modal logic})$$

(from LLL):

$$\frac{\vdash B_1 \oplus \ldots \oplus Bn, A}{\vdash ?B_1, \ldots ?Bn, !A}\ \text{“of course”} \qquad \frac{\vdash B_1 \oplus \ldots \oplus B_n, A_1, \ldots A_m}{\vdash ?B_1, \ldots ?B_n, \S A_1, \ldots \S A_m}\ \text{neutral}_1 \qquad \frac{\vdash B_1, \ldots B_n, A_1, \ldots A_m}{\vdash ?B_1, \ldots ?B_n, \S A_1, \ldots \S A_m}\ \text{neutral}_2$$

7.4 Analysis of Proofs

The property that both logic programming and theorem provers share is that both kinds of system are generally interested in determining whether or not there is a proof of a given goal. It is well known that a proof contains much more information than the simple fact that its conclusion is true. Unlike a theorem prover, a logic programming system is generally interested in a more detailed analysis of the proof in order to extract some kind of extra information from the proof.
For example, when the user asks for the truth (or provability) of a non-ground query $\exists x P(x)$ the system is expected to return not only *yes* or *no*, but also an *answer i.e.* substitution for the (existentially quantified) variables of the goal: a value of x for which $P(x)$ is true.

For example, consider the (Prolog) program below:

$$\begin{array}{l} p(a). \\ p(X): \ -q(X). \\ q(b). \end{array}$$

Clearly the goal $\exists x p(x)$ succeeds, and there are two answer substitutions, being $x \leftarrow a$ and $x \leftarrow b$. Computation of the given goal from the given program is searching for a proof of the sequent:

$$p(a), \forall y q(y) \Rightarrow p(y), q(b) \ \vdash \ \exists x p(x)$$

The answer substitution amounts to a summary of the important information contained in the proofs of each $p(a)$ and $p(b)$ and proofs look a little different in each case. All possible uniform proofs[8] which provide $x \leftarrow b$ substitution are given below:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{q(b) \vdash q(b)}\,Ax \quad \dfrac{\dfrac{}{p(b) \vdash p(b)}\,Ax}{p(a), p(b) \vdash p(b)}\,w-L}{p(a), q(b) \Rightarrow p(b), q(b) \vdash p(b)}\,\Rightarrow -L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(b)}\,\forall - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q(b) \vdash q(b)}\,Ax \quad \dfrac{}{p(b) \vdash p(b)}\,Ax}{q(b) \Rightarrow p(b), q(b) \vdash p(b)}\,\Rightarrow -L}{\forall y q(y) \Rightarrow p(y), q(b) \vdash p(b)}\,\forall - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(b)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q(b) \vdash q(b)}\,Ax \quad \dfrac{}{p(b) \vdash p(b)}\,Ax}{q(b) \Rightarrow p(b), q(b) \vdash p(b)}\,\Rightarrow -L}{p(a), q(b) \Rightarrow p(b), q(b) \vdash p(b)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(b)}\,\forall - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R$$

It is obvious that above proofs are equivalent modulo inference permutations. All possible uniform proofs which provide $x \leftarrow a$ substitution are:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a), q(b) \vdash p(a)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(a)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a), \forall y q(y) \Rightarrow p(y) \vdash p(a)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(a)}\,w - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{q(b) \vdash q(b)}\,Ax \quad \dfrac{\dfrac{}{p(a) \vdash p(a)}\,Ax}{p(a), p(b) \vdash p(a)}\,w-L}{p(a), q(b) \Rightarrow p(b), q(b) \vdash p(a)}\,\Rightarrow -L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(a)}\,\forall - L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\,\exists - R$$

[8] Uniform proof of a sequent $\mathcal{P} \vdash G$ means a goal-directed proof, in a sense that the goal G is decomposed uniformly, based only on its structure and without reference to the program $\mathcal{P}$, until atomic goals are reached. The program is only consulted when atomic goals are to be proved.

Again, among them the first two proofs belong to the same equivalence class, modulo permutations.

For the "substitution for existentially quantified variables", it is important to investigate *all* possible proofs of a given goal, rather than just determining if a goal is provable. Different proofs are only interesting if they lead to different answers. A logic program for which there are many proofs of the same goal (and hence there are goals which return the same answer substitution many times) is generally considered to be somewhat deficient. Clearly, the question of finding all proofs which lead to different answers have some overlap with the question of finding all equivalent proofs modulo inference permutations. Since proofs from the same equivalence class lead to the same answer, it is sufficient to generate just one of them, which than will be representative for the whole class. This is reduced to the problem of finding all permutations of the proof *i.e.* of finding the appropriate equivalence class, which than will be "excluded from further consideration".

Uniformity of proofs is the characteristic feature for the logic programming proofs. In the above example, in the uniform proofs which provide $x \leftarrow b$ substitution, the reasons for such substitution, which is performed (because of the uniformity constraint) in the first step of proof construction, are vague:

$$\dfrac{\begin{array}{c}\vdots\\ p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash p(b)\end{array}}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\ \exists - R$$

Such substitution is result of analysing $\forall y q(y) \Rightarrow p(y)$, $q(b)$ part of the program. Confronted with the question of substitution for the quantified variables, an implementation might relax the uniformity constraint, recognising the advantage of the application of certain left rules before the right ones. One of possible (non-uniform) proofs that can result is:

$$\dfrac{\dfrac{\dfrac{}{q(b) \vdash q(b)}\ Ax \qquad \dfrac{\dfrac{\dfrac{}{p(b) \vdash p(b)}\ Ax}{p(a), p(b) \vdash p(b)}\ w - L}{p(a), p(b) \vdash \exists x p(x)}\ \exists - R}{p(a), q(b) \Rightarrow p(b), q(b) \vdash \exists x p(x)}\ \Rightarrow -L}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\ \forall - L$$

After the step $\dfrac{p(a), q(b) \Rightarrow p(b) \vdash \exists x p(x)}{p(a), \forall y q(y) \Rightarrow p(y), q(b) \vdash \exists x p(x)}\ \forall - L$, the origin of the idea for the substitution $x \leftarrow b$ is more clear. Again, permutations play important role, in that such (non-uniform but let say clear) proof can be rearranged (by permutations) into a uniform proof.

7.5 Optimisations by Permutations

Many optimisations are possible based solely on the permutations of inference rules of a given proof. Such transformations are easy to understand, and, more importantly, their correctness is immediate. One such optimisation is minimisation of number of rules in a proof, which is important specially for rules with the high level of non-determinism. For instance, given a proof:

$$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \vdash A, P\end{array}}{\vdash A, P \oplus Q}\ \oplus \qquad \dfrac{\begin{array}{c}\Sigma_2\\ \vdash B, P\end{array}}{\vdash B, P \oplus Q}\ \oplus}{\vdash A \& B, P \oplus Q}\ \&$$

it is possible to permute (see special-permutations in subsection 6.3.3) the rules $\oplus$ and $\&$. The proof tree that results is equivalent to the starting one, with one less application of the rule $\oplus$:

$$\dfrac{\dfrac{\begin{matrix}\Sigma_1 \\ \vdash A, P\end{matrix} \qquad \begin{matrix}\Sigma_2 \\ \vdash B, P\end{matrix}}{\vdash A\&B, P}\ \&}{\vdash A\&B, P \oplus Q}\ \oplus$$

The above optimisation can be also interpreted as an attempt for reducing the amount of application of rules with the high level of non-determinism such as the case with the $\oplus$ rule.

Another optimisation by using permutations can be interpreted as decreasing the amount of branching in the proof. For instance, the proof:

$$\dfrac{\dfrac{\begin{matrix}\Xi_1 \\ \vdash P, A\end{matrix} \qquad \begin{matrix}\Sigma_1 \\ \vdash B\end{matrix}}{\vdash P, A \otimes B}\ \otimes \qquad \dfrac{\begin{matrix}\Xi_2 \\ \vdash Q, A\end{matrix} \qquad \begin{matrix}\Sigma_2 \\ \vdash B\end{matrix}}{\vdash Q, A \otimes B}\ \otimes}{\vdash P\&Q, A \otimes B}\ \&$$

can be transformed into the proof:

$$\dfrac{\dfrac{\begin{matrix}\Xi_1 \\ \vdash P, A\end{matrix} \qquad \begin{matrix}\Xi_2 \\ \vdash Q, A\end{matrix}}{\vdash P\&Q, A}\ \& \qquad \begin{matrix}\Sigma \\ \vdash B\end{matrix}}{\vdash P\&Q, A \otimes B}\ \otimes$$

where Σ is one of two proofs of $\vdash B$ (which are not necessary different but could be). If the proof of $\vdash B$ is large or involves certain non-determinism then this optimisation may be appropriate.

8 Conclusions and Future Work

We have outlined the first steps in the investigation automation of the design of LP languages. The aim is to develop a prototype of a languages compiler generator, *i.e.* a logic-independent system, which will contain various implementation and proof-theoretic techniques together with the general strategic planner, capable of determining the most appropriate way to assemble the various pieces into a LP system, but in a formalism and framework which is designed and intended for humans.

The first steps towards such an automated analyser are dedicated to specifications of appropriate machine formalisms for the encapsulation, manipulation and automation of proofs and search strategies. We have proposed a more detailed specification of inference rules, for a better understanding of their structure and their generic properties. On the basis of introduced specifications we have generalised some definitions and meta-rules for the permutation analyses. Also, it seems that it is possible to exploit some similarities in the structure of $\mathcal{SC}$ inference rules from different proof systems for the extension and generalisation of some methods and techniques originally designed and dedicated to particular sets of rules. For example, proof-nets, Girard's "parallel syntax for proof theory" [16], are among the most fascinating constructions for a concise graph-theoretic representation of deductions by "decontextualization of inferences". They have arisen and are based on some fragments of linear logic, principally first order $\otimes, \wp, cut$ and $\otimes, \wp, cut, mix$ fragments. The question that arises is: are the same principles for the construction and analysis of proof-nets in, say, $\wp, cut, mix$ fragment of linear logic applicable in $\vee, cut, mix$ fragment of LK, in regard to the same structure (same number of active, quasi-active, principal and extra formulas, the same type of restrictions on the structure of involved formulas and multisets, etc) of rules:

$$\text{CLL:}\quad \dfrac{\vdash A, B\ ;\ ;\ \Gamma\ ;}{\vdash A \wp B\ ;\ ;\ \Gamma\ ;}\ \wp \qquad\qquad \text{LK:}\quad \dfrac{\vdash A, B\ ;\ ;\ \Gamma\ ;}{\vdash A \vee B\ ;\ ;\ \Gamma\ ;}\ \vee$$

The permutations of some rules require a further differentiation within the active, quasi-active and principal formulas to *active, quasi-active, principal* and *context subformulas.* As an example, consider the $\exists$ rule from $\mathcal{SC}$ 2.3., and an example of its permutation:

$$\frac{\Gamma \vdash A[t/x] \vee ((\exists x A) \vee (\Delta_S))}{\Gamma \vdash (\exists x A) \vee (\Delta_S)}\ \exists(\vee) \qquad \begin{array}{c}\Delta_S = A_{i(1)} \vee A_{i(2)} \vee \ldots \vee A_{i(n)}, where \\ i : \{1, 2, \ldots n\} \mapsto \{1, 2, \ldots n\} \text{denotes an arbitrary permutation}\end{array}$$

$$\frac{\dfrac{\begin{array}{c}\vdots\\ A(a) \vdash A(a) \vee \exists x A(x)\end{array}}{\vdash \neg A(a) \vee A(a) \vee \exists x A(x)}\ \neg}{\vdash \neg A(a) \vee \exists x A(x)}\ \exists \quad \Longleftrightarrow \quad \frac{\dfrac{\begin{array}{c}\vdots\\ A(a) \vdash A(a) \vee \exists x A(x)\end{array}}{A(a) \vdash \exists A(x)}\ \exists}{\vdash \neg A(a) \vee \exists x A(x)}\ \neg$$

The similar treatment is required for the permutation of the following rules from light linear logic:

$$\frac{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i}{\vdash \Gamma, \bigoplus_{i=1}^{m} A_i \oplus B}\ \text{A-weakening} \qquad \frac{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i \oplus B \oplus B}{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i \oplus B}\ \text{A-contraction}$$

Specification and detection of such rules is question for investigation related further work.

Much work remains to be done, of course, including the extension of the permutation analysis from the manipulations of different permutations of the same proof towards the using these properties to perform an analysis of the performance of particular search strategies. Various properties of search strategies have direct influence on the design of LP languages. Also, the investigation of properties of goal-directed proofs for new systems (such as sequent calculi for intuitionistic logic which are multiple-conclusioned, light linear logic, contraction-free sequent calculi for various logics, etc.) remains to be done. Making use of the generic properties of inference rules rather than particular instances of rules will allow analyses on various logics and various classes of formulae within each logic.

References

[1] Andreoli J.M. *Logic Programming with focusing proofs in linear logic*, Journal of Logic and Computation, 2(3), 1992, pp.297-347

[2] J.M. Andreoli, R. Pareschi *Logic programming with sequent systems: A linear Logic Approach.* In P.Schröder-Heister ed., Proceedings of Workshop to Extensions of Logic Programming, Tübingen, 1989, LNAI 475, pp.1-30

[3] J.M. Andreoli, R. Pareschi *Linear Objects: Logical processes with builth-in inheritance*, New Gen. Comp. 9, 1991, pp.445-473.

[4] Bellin G. *Mechanising Proof Theory: Resource-Aware Logics and Proof-Transformations to Extract Implicit Information*, PhD Dissertation, Department of philosophy, Stanford University, 1990.

[5] Constable R.E. et. al. *Implementing Mathematics with NuPRL proof development system*, Prentice Hall, 1986.

[6] Davoren J. *A Lazy Logician's Guide to Linear Logic*, Technical Report 76b, Department of Mathematics, Monash University, 1992.

[7] Dyckhoff R. *Contraction-free Sequent Calculi for Intuitionistic Logic*, The Journal of Symbolic Logic, Vol. 57, No. 3, Sept. 1992, pp.795-807

[8] Felty A. *A Logic Program for Transforming Sequent Proofs to Natural Deduction Proofs* In P.Schröder-Heister ed. Proceedings of Workshop to Extensions of Logic Programming, Tübingen, 1989, LNAI 475, pp.157-178.

[9] Gallier J. *Logic for Computer Science. Foundations of Automatic Theorem Proving* John Wiley & Sons, 1987.

[10] Gallier J. *Constructive Logics. Part I: A Tutorial on Proof Systems and Typed λ -Calculi,* , manuscript, 1992.

[11] Gallier J. *Constructive Logics. Part II: Linear Logic and Proof Nets*, manuscript, 1992.

[12] Galmiche D. *Canonical Proofs for Linear Logic Programming Frameworks*, in A. Momigliano and M. Ornaghi, ed., Proof-Theoretical Extensions of Logic Programming, Post-Conference Workshop for ICLP'94, Santa Margherita Ligure, Italy, June 1994, pp.2-10.

[13] Galmiche D. and Perrier G. *On proof normalisation in Linear Logic*, Theoretical Computer Science 135, 1994, pp. 67-110.

[14] Girard J.-Y. *Linear Logic: its Syntax and Semantics*, in Advances in Linear Logic, J-Y.Girard, Y.Lafont and L.Regnier (eds.), Cambridge University Press, 1995., pp. 1-42.

[15] Girard J.-Y. *Light Linear Logic*, manuscript sent via linear mailing list, February, 1995.

[16] Girard J.-Y. *Proof-nets: the Parallel syntax for Proof-theory*, Logic and Algebra, Marcel Dekker, New York, 1995.

[17] Girard J.-Y., Lafont Y., Taylor P. *Proofs and Types*, Cambridge University Press, 1989.

[18] Gordeev L. *Cut Free Formalisation of Logic with Finitely Many Variables. Part I*, In Proceedings of the 16^{th} German Conference on Artificial Intelligence, Bonn, Germany, LNAI 671, H.J.Ohlbach (ed.), 1992, pp.136-150

[19] Keronen S. *Natural Deduction Proof Theory for Logic Programming*, in Proceedings of the Third International Workshop Extensions of Logic Programming, Italy, E.Lamma, P.Mello Eds., LNAI 660, 1992, pp. 265-281.

[20] Kleene S.C. *Introduction to Metamathematics*, North-Holland, 1952.

[21] Kleene S.C. *Permutability of Inferences in Gentzen's Calculi LK and LJ*, Memoirs of the American Mathematical Society 10, 1952, pp. 1-26.

[22] Harland J. *On Hereditary Harrop Formulae as a Basis for Logic Programming* PhD Thesis, Department of Computer Science, University of Edinburgh, July, 1991.

[23] Harland J. *A proof-theoretic Analysis of Goal-directed Provability*, Journal of Logic and Computation, Vol.4 No 1, 1994, pp.69-88.

[24] Harland J. *Lecture notes for Mathematical Logic and Logic Programming*, Department of Computer Science, RMIT University, Melbourne, 1996.

[25] Harper R., Honsell F., Plotkin G.: *A framework for defining logics* Journal of the Association for Computing Machinery, 40(1), pp. 143-184, January 1993. A preliminary version appeared in the Proceedings of the Symposium on Logic in Computer Science, pp. 194-204, June 1987.

[26] Heuerding A., Seyfried M., Zimmermann H. *Efficient Loop-Check for Backward Proof Search in Some Non-classical Propositional Logics*, In Proceedings of the 5th International Workshop, TABLEAUX '96, P.Miglioli, U.Moscato, D.Mundici and M.Ornaghi Eds., Italy 1996., pp.210-225.

[27] Hodas J.S. *Logic Programming in Intuitionistic Linear Logic: Theory, Design and Implementation*, PhD thesis, University of Pennsylvania, 1994.

[28] Hodas J.S. and Miller D. *Logic programming in a Fragment of Intuitionistic Linear Logic*, Journal of Information and Computation 110:2, 1994, pp.327-365.

[29] Lincoln P.D. and N. Shankar *Proof Search in First-order Linear Logic and Other Cut-free Sequent Calculi*, In S. Abramsky ed., Ninth Annual Symposium on Logic in Computer Science, Paris, France, 1994., IEEE Computer Society Press, pp. 282 - 291.

[30] Lloyd J.W. *Foundations of Logic Programming*, Springer-Verlag, Berlin, 1984.

[31] Miller D. *Expansion Tree Proofs and their Conversion to Natural Deduction Proofs*, in R.E. Shostak ed. The 7th Conference on Automated Deduction, Napa, 1984., LNAI 170, 1984, pp.375-393.

[32] Miller D. *A logical analysis of modules in logic programming*, Journal of Logic Programming 6, 1989, pp.79-108

[33] Miller D., Nadathur G., Pfenning F., Ščedrov A. *Uniform proofs as a Foundation for Logic Programming*, Annals of Pure and Applied Logic 51, 1991, pp.125-157

[34] Miller D. *Forum: A multiple-conclusion specification-logic*, Theoretical Computer Science 165(1): 201-232, 1996.

[35] Nadathur G. and Miller D. *An overview of λProlog*, In Kenneth A. Bowen and Robert A. Kowalski, editors, Fifth International Logic Programming Conference, pp. 810-827, Seattle, Washington, August 1988, MIT Press.

[36] Nederpelt R.P., Geuvers J.H. and de Vrijer R.C., editors *Selected Papers on Automath*, volume 133 of Studies in Logic and the Foundations of Mathematics, North-Holland, 1994.

[37] Paulson L. *Isabelle - A Generic Theorem Prover*, in G. Goos and J. Hartmanis ed. Lecture Notes in Computer Science no. 828, Springer-Verlag Berlin Heidelberg, 1994.

[38] Pym J.D., Harland A.J. *A Uniform Proof-theoretic Investigation of Linear Logic Programming*, Journal of Logic and Computation, Vol. 4 No 2, 1994, pp. 175-207

[39] Pfenning F. *Analytic and Non-analytic Proofs*, in M.E. Stickel ed., 10th International Conference on Automated Deduction, Kaiserslautern, FRG, 1990, LNAI 449, pp.394-413

[40] Pfenning F. *Elf: A meta-language for deductive systems*, In A. Bundy editor, Proceedings of the 12th International Conference on Automated Deduction, pp. 811-815, Nancy, France, June 1994. Springer-Verlag LNAI 814, System abstract.

[41] Pollack R. *The theory of LEGO - a proof checker for the extended calculus of constructions*, PhD thesis, University of Edinburgh, 1994.

[42] Schmitt S, Kreitz C. *On transforming Intuitionistic Matrix Proofs into Standard-Sequent Proofs*, in P. Baumgartner, R. Hähnle, J. Posegga eds., Proceedings of the 4th International Workshop TABLEAUX '95, Schloβ Rheinfels, St. Goar, Germany, 1995, LNAI 918, pp. 106-121

[43] Shankar N. *Proof Search in the Intuitionistic Sequent Calculi*, In 11th Conference on automated Deduction, LNAI 607, Saratoga Springs, June 1992, pp. 522-536

[44] Sterling L., Shapiro E. *The Art of Prolog*, MIT Press, 1986.

[45] Tammet T. *Proof Search Strategies in Linear Logic*, Journal of Automated Reasoning no. 12, 1994, pp.273-304.

[46] Wallen L.A. *Automated Proof Search in Non-classical Logic*, MIT Press, 1990.

[47] Winikoff M., Harland J. *Implementing the Linear Logic Programming Language Lygon*, Proceedings of the International Logic Programming Symposium, Portland, Oregon, December, 1995, pp.66-80

[48] Winikoff M., Harland J. *Deriving Logic Programming Languages*, Technical Report 95/26, Department of Computer Science, The University of Melbourne, 1996.

[49] Winikoff M. *Lygon home page* http://www.cs.mu.oz.au/∼winikoff/lygon/lygon.html

A Multisets, operations, relations

Definition A.1 *Multisets, operations, relations:*

- A *multiset* is a collection of object that are not necessarily distinct.
- The *multiplicity* of an element in a multiset is defined to be the number of times the element appears in the multiset.

Multiset operations:

- The *union* of multisets Γ and Δ, denoted $\Gamma \cup \Delta$, is a multiset such that the multiplicity of an element in $\Gamma \cup \Delta$ is equal to the maximum of the multiplicities of the element in Γ and in Δ.
- The *sum* of multisets Γ and Δ, denoted $\Gamma\mathbf{,}\ \Delta$, is a multiset such that the multiplicity of an element in $\Gamma\mathbf{,}\ \Delta$ is equal to the sum of the multiplicities of the element in Γ and in Δ.
- The *intersection* of multisets Γ and Δ, denoted $\Gamma \cap \Delta$, is a multiset such that the multiplicity of an element in $\Gamma \cap \Delta$ is equal to the minimum of the multiplicities of the element in Γ and in Δ.
- The *difference* of multisets Γ and Δ, denoted $\Gamma \backslash \Delta$, is a multiset such that the multiplicity of an element in $\Gamma \backslash \Delta$ is equal to the multiplicity of the element in Γ minus the multiplicity of the element in Δ if the difference is positive, and is equal to 0 if the difference is 0 or is negative.

Multiset relations:

- A multiset Γ is *proper multi-subset* of Δ, denoted by $\Gamma \subset \Delta$, if every element that appears in Γ appears also in Δ with the same or greater multiplicity.
- A multiset Γ is *multi-subset* of Δ, denoted by $\Gamma \subseteq \Delta$, if Γ is equal or proper multi-subset of Δ.

B Sequent Calculi for various logics

B.1 Classical logic

B.1.1 LK: Two-Sided Gentzen's Sequent Calculus

$$\frac{}{\phi \vdash \phi}\ axiom \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma, \phi \vdash \Delta}{\Gamma \vdash \Delta}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash \Delta}{\Gamma, \psi, \phi, \Gamma' \vdash \Delta}\ ex\text{-L} \qquad \frac{\Gamma \vdash \Delta, \phi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \phi, \Delta'}\ ex\text{-R}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ W\text{-L} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ W\text{-R}$$

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ C\text{-L} \qquad \frac{\Gamma \vdash F, F, \Delta}{\Gamma \vdash F, \Delta}\ C\text{-R}$$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta}\ \wedge\text{-R}$$

$$\frac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}\ \vee\text{-L} \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee\text{-R}$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \to \psi, \Gamma \vdash \Delta}\ \to\text{-L} \qquad \frac{\phi, \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \to \psi, \Delta}\ \to\text{-R}$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta}\ \neg\text{-R}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot\text{-L} \qquad \frac{}{\Gamma \vdash \top, \Delta}\ \top\text{-R}$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x], \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x], \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ

B.1.2 LK1: A One-Sided Formulation

$$\frac{}{\vdash \phi, \neg\phi}\ axiom \qquad \frac{\vdash \phi, \Delta \quad \vdash \neg\phi, \Delta}{\vdash \Delta}\ cut$$

$$\frac{\Gamma \vdash \Delta, \phi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \phi, \Delta'}\ ex\text{-R} \qquad \frac{\vdash \Delta}{\vdash F, \Delta}\ W\text{-R}$$

$$\frac{\vdash F, F, \Delta}{\vdash F, \Delta}\ C\text{-R} \qquad \frac{\vdash \phi, \Delta \quad \vdash \psi, \Delta}{\vdash \phi \wedge \psi, \Delta}\ \wedge\text{-R}$$

$$\frac{}{\vdash \top, \Delta}\ \top\text{-R}$$

$$\frac{\vdash \phi, \psi, \Delta}{\vdash \phi \vee \psi, \Delta}\ \vee\text{-R} \qquad \frac{\vdash \neg\phi, \psi, \Delta}{\vdash \phi \to \psi, \Delta}\ \to\text{-R}$$

$$\frac{\vdash \phi[y/x], \Delta}{\vdash \forall x.\phi, \Delta}\ \forall\text{-R}^* \qquad \frac{\vdash \phi[t/x], \Delta}{\vdash \exists x.\phi, \Delta}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ

B.1.3 C$_3$: A $\mathcal{SC}$ without Structural Rules

$$\frac{}{\Gamma, \phi \vdash \phi, \Delta}\ axiom \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma, \phi \vdash \Delta}{\Gamma \vdash \Delta}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash \Delta}{\Gamma, \psi, \phi, \Gamma' \vdash \Delta}\ ex\text{-L} \qquad \frac{\Gamma \vdash \Delta, \phi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \phi, \Delta'}\ ex\text{-R}$$

$$\frac{\phi, \phi \wedge \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge\text{-L}_1 \qquad \frac{\psi, \phi \wedge \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge\text{-L}_2 \qquad \frac{\Gamma \vdash \phi, \phi \wedge \psi, \Delta \quad \Gamma \vdash \psi, \phi \wedge \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta}\ \wedge\text{-R}$$

$$\frac{\phi, \phi \vee \psi, \Gamma \vdash \Delta \quad \psi, \phi \vee \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}\ \vee\text{-L} \qquad \frac{\Gamma \vdash \phi, \phi \vee \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee\text{-R} \qquad \frac{\Gamma \vdash \psi, \phi \vee \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee\text{-R}$$

$$\frac{\Gamma, \phi \to \psi \vdash \phi, \Delta \quad \psi, \phi \to \psi, \Gamma \vdash \Delta}{\phi \to \psi, \Gamma \vdash \Delta}\ \to\text{-L} \qquad \frac{\phi, \Gamma \vdash \psi, \phi \to \psi, \Delta}{\Gamma \vdash \phi \to \psi, \Delta}\ \to\text{-R}$$

$$\frac{\Gamma, \neg\phi \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash \neg\phi, \Delta}{\Gamma \vdash \neg\phi, \Delta}\ \neg\text{-R}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot\text{-L} \qquad \frac{}{\Gamma \vdash \top, \Delta}\ \top\text{-R}$$

$$\frac{\Gamma, \phi[t/x], \forall x.\phi \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x], \forall x.\phi, \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x], \exists x.\phi \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x], \exists x.\phi, \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ

B.2 Intuitionistic Logic

B.2.1 LJ: Gentzen's Sequent Calculus

$$\frac{}{\phi \vdash \phi}\ axiom \qquad \frac{\Gamma_1 \vdash \phi \quad \Gamma_2, \phi \vdash F}{\Gamma_1, \Gamma_2 \vdash F}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash F}{\Gamma, \psi, \phi, \Gamma' \vdash F}\ ex\text{-L}$$

$$\frac{\Gamma \vdash F}{\Gamma, \phi \vdash F}\ W\text{-L} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \phi}\ W\text{-R}$$

$$\frac{\Gamma, \phi, \phi \vdash F}{\Gamma, \phi \vdash F}\ C\text{-L}$$

$$\frac{}{\Gamma, \bot \vdash F}\ \bot\text{-L} \qquad \frac{}{\Gamma \vdash \top}\ \top\text{-R}$$

$$\frac{\phi, \psi, \Gamma \vdash F}{\phi \wedge \psi, \Gamma \vdash F}\ \wedge\text{-L} \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-R}$$

$$\frac{\phi, \Gamma \vdash F \quad \psi, \Gamma \vdash F}{\phi \vee \psi, \Gamma \vdash F}\ \vee\text{-L} \qquad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi, \Delta} \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee\text{-R}$$

$$\frac{\Gamma_1 \vdash \phi \quad \psi, \Gamma_2 \vdash F}{\phi \to \psi, \Gamma_1, \Gamma_2 \vdash F} \to\text{-L} \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \to \psi} \to\text{-R}$$

$$\frac{\Gamma \vdash \phi}{\Gamma, \neg\phi \vdash}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash}{\Gamma \vdash \neg\phi}\ \neg\text{-R}$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x]}{\Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

where y is not free in Γ

B.2.2 G_3: A Sequent Calculus without Structural Rules

$$\frac{}{\Gamma, \phi \vdash \phi}\ axiom \qquad \frac{\Gamma \vdash \phi \quad \Gamma, \phi \vdash F}{\Gamma \vdash F}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash F}{\Gamma, \psi, \phi, \Gamma' \vdash F}\ ex\text{-L}$$

$$\frac{\phi, \phi \wedge \psi, \Gamma \vdash F}{\phi \wedge \psi, \Gamma \vdash F}\ \wedge\text{-L}_1 \qquad \frac{\psi, \phi \wedge \psi, \Gamma \vdash F}{\phi \wedge \psi, \Gamma \vdash F}\ \wedge\text{-L}_2 \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-R}$$

$$\frac{\phi, \phi \vee \psi, \Gamma \vdash F \quad \psi, \phi \vee \psi, \Gamma \vdash F}{\phi \vee \psi, \Gamma \vdash F}\ \vee\text{-L} \qquad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-R} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-R}$$

$$\frac{\Gamma, \phi \to \psi \vdash \phi \quad \psi, \phi \to \psi, \Gamma \vdash F}{\phi \to \psi, \Gamma \vdash F} \to\text{-L} \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \to \psi} \to\text{-R}$$

$$\frac{\Gamma, \neg\phi \vdash \phi}{\Gamma, \neg\phi \vdash F}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash p}{\Gamma \vdash \neg\phi}\ \neg\text{-R}^{**}$$

$$\frac{}{\Gamma, \bot \vdash F}\ \bot\text{-L} \qquad \frac{}{\Gamma \vdash \top}\ \top\text{-R}$$

$$\frac{\Gamma, \phi[t/x], \forall x.\phi \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x], \exists x.\phi \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x]}{\Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ
** p is new atom *i.e.* atom that not occur in Γ, ϕ

B.2.3 Non-standard Sequent Calculus

$$\frac{}{\Gamma, \phi \vdash \phi}\ axiom \qquad \frac{\Gamma \vdash \phi \quad \Gamma, \phi \vdash F}{\Gamma \vdash F}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash F}{\Gamma, \psi, \phi, \Gamma' \vdash F}\ ex\text{-L}$$

$$\frac{\phi, \psi, \Gamma \vdash F}{\phi \wedge \psi, \Gamma \vdash F}\ \wedge\text{-L} \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-R}$$

$$\frac{\phi, \Gamma \vdash F \quad \psi, \Gamma \vdash F}{\phi \vee \psi, \Gamma \vdash F}\ \vee\text{-L} \qquad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-R} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-R}$$

$$\frac{\Gamma, \phi \to \psi \vdash \phi \quad \psi, \Gamma \vdash F}{\phi \to \psi, \Gamma \vdash F}\ \to\text{-L} \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \to \psi}\ \to\text{-R}$$

$$\frac{\Gamma, \neg\phi \vdash \phi}{\Gamma, \neg\phi \vdash F}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash}{\Gamma \vdash \neg\phi}\ \neg\text{-R}$$

$$\frac{\Gamma, \phi[t/x], \forall x.\phi \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x], \exists x.\phi \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x]}{\Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

$$\frac{\Gamma \vdash A_i \vee \Delta_S^i}{\Gamma \vdash \Delta_S}\ \vee - changeA_i$$

$$\frac{\Gamma, A \to B \vdash A \vee (\Delta_S) \quad \Gamma, B \vdash \Delta_S}{\Gamma, A \to B \vdash \Delta_S}\ \to(\vee)L \qquad \frac{\Gamma, \neg A \vdash A \vee (\Delta_S)}{\Gamma, \neg A \vdash \Delta_S}\ \neg(\vee)R$$

$$\frac{\Gamma \vdash A \vee (B \vee (\Delta_S))}{\Gamma \vdash (A \vee B) \vee (\Delta_S)}\ \vee(\vee)L \qquad \frac{\Gamma \vdash A \vee (\Delta_S) \quad \Gamma \vdash B \vee (\Delta_S)}{\Gamma \vdash (A \wedge B) \vee (\Delta_S)}\ \vee(\vee)R$$

$$\frac{\Gamma \vdash A[t/x] \vee ((\exists x A) \vee (\Delta_S))}{\Gamma \vdash (\exists x A) \vee (\Delta_S)}\ \exists(\vee)R \qquad \frac{\Gamma \vdash \phi[t/x] \vee \exists x.\phi}{\Gamma \vdash \exists x.\phi}\ \exists_1\text{-R}$$

where $\Delta = A_1, A_2, \ldots A_n$

$\Delta_S = A_{i(1)} \vee A_{i(2)} \vee \ldots \vee A_{i(n)}$ where $i : \{1, 2, \ldots n\} \mapsto \{1, 2, \ldots n\}$ denotes an arbitrary permutation.

$\Delta_S^i = A_1 \vee A_2 \vee \ldots A_{i-1} \vee A_{i+1} \vee \ldots \vee A_{i(n)}$

* y is not free in Γ, Δ, ϕ

B.2.4 LJT: Contraction Free Sequent Calculus

$$\frac{}{a, \Gamma \vdash a}\ axiom \qquad \frac{}{\bot, \Gamma \vdash G}\ \bot$$

$$\frac{\phi, \psi, \Gamma \vdash G}{\phi \wedge \psi, \Gamma \vdash G}\ \wedge l \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge r$$

$$\frac{\phi, \Gamma \vdash G \quad \psi, \Gamma \vdash G}{\phi \vee \psi, \Gamma \vdash G}\ \vee l \qquad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee r$$

$$\frac{B, a, \Gamma \vdash G}{a \supset B, a, \Gamma \vdash G} \supset_1 l \quad \text{a being atomic}$$

$$\frac{C \supset (D \supset B), \Gamma \vdash G}{(C \wedge D) \supset B, \Gamma \vdash G} \supset_2 l$$

$$\frac{C \supset B, D \supset B, \Gamma \vdash G}{(C \vee D) \supset B, \Gamma \vdash G} \supset_3 l$$

$$\frac{D \supset B, \Gamma \vdash C \supset D \quad B, \Gamma \vdash G}{(C \supset D) \supset B, \Gamma \vdash G} \supset_4 l \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \supset \psi} \supset r$$

B.3 Classical Linear Logic

B.3.1 CLL: A Two-Sided Sequent Calculus

$$\frac{}{\phi \vdash \phi}\ axiom \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma', \phi \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ cut$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash \Delta}{\Gamma, \psi, \phi, \Gamma' \vdash \Delta}\ ex\text{-L} \qquad \frac{\Gamma \vdash \Delta, \phi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \phi, \Delta'}\ ex\text{-R}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}\ \mathbf{1}\text{-L} \qquad \frac{}{\vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\frac{}{\bot \vdash}\ \bot\text{L} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}\ \bot\text{R}$$

$$\frac{}{\Gamma \vdash \top, \Delta}\ \top\text{-R}$$

$$\frac{}{\Gamma, \mathbf{0} \vdash \Delta}\ \mathbf{0}\text{-L}$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\ \neg\text{-L} \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta}\ \neg\text{-R}$$

$$\frac{\Gamma, \phi, \psi \vdash \Delta}{\Gamma, \phi \otimes \psi \vdash \Delta}\ \otimes\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma' \vdash \psi, \Delta}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, \phi \& \psi \vdash \Delta} \quad \frac{\Gamma, \psi \vdash \Delta}{\Gamma, \phi \& \psi \vdash \Delta}\ \&\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \phi, \Delta}{\Gamma \vdash \phi \& \psi, \Delta}\ \&\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \phi \oplus \psi \vdash \Delta}\ \oplus\text{-L} \qquad \frac{\Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \oplus \psi, \Delta} \quad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash \phi \oplus \psi, \Delta}\ \oplus\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta \quad \Gamma', \psi \vdash \Delta'}{\Gamma, \Gamma', \phi \wp \psi \vdash \Delta, \Delta'}\ \wp\text{-L} \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \wp \psi, \Delta}\ \wp\text{-R}$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma', \psi \vdash \Delta'}{\Gamma, \Gamma', \phi \multimap \psi \vdash \Delta, \Delta'}\ \multimap\text{-L} \qquad \frac{\Gamma, \phi \vdash \psi, \Delta}{\Gamma \vdash \phi \multimap \psi, \Delta}\ \multimap\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ !\text{-L} \qquad \frac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\ !\text{-R}$$

$$\frac{!\Gamma, \phi \vdash ?\Delta}{!\Gamma, ?\phi \vdash ?\Delta}\ ?\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash ?\phi, \Delta}\ ?\text{-R}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ W!\text{-L} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?\phi, \Delta}\ W?\text{-R}$$

$$\frac{\Gamma, !\phi, !\phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ C\text{-L} \qquad \frac{\Gamma \vdash ?\phi, ?\phi, \Delta}{\Gamma \vdash ?\phi, \Delta}\ C\text{-R}$$

$$\frac{\Gamma, \phi[t/x] \vdash \Delta}{\Gamma, \forall x.\phi \vdash \Delta}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x], \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x] \vdash \Delta}{\Gamma, \exists x.\phi \vdash \Delta}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x], \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists\text{-R}$$

* y is not free in Γ, Δ

B.3.2 CLV: A Two-Sided Sequent Calculus with Four Zones

A sequent has the form $\Psi; \Gamma \vdash \Delta; \Theta$ which may be interpreted as $!\Psi, \Gamma \vdash \Delta, \Theta$. Note that formulas of the form $!A$ and $?A$ may occur freely in Γ and Δ.

$$\frac{}{\Psi; \phi \vdash \phi; \Theta}\ \mathit{axiom} \qquad \frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta \quad \Psi; \Gamma', \phi \vdash \Delta'; \Theta}{\Psi; \Gamma, \Gamma' \vdash \Delta, \Delta'; \Theta}\ \mathit{Cut}$$

$$\frac{\Psi; \vdash \psi; \Theta \quad (\Psi, \psi); \Gamma \vdash \Delta; \Theta}{\Psi; \Gamma \vdash \Delta; \Theta}\ \mathit{Cut!} \qquad \frac{\Psi; \Gamma \vdash \Delta; (\psi, \Theta) \quad \Psi; \psi \vdash ; \Theta}{\Psi; \Gamma \vdash \Delta; \Theta}\ \mathit{Cut?}$$

$$\frac{\Psi; \Gamma \vdash \Delta; \Theta}{\Psi; \Gamma, \mathbf{1} \vdash \Delta; \Theta}\ \mathbf{1}\text{-L} \qquad \frac{}{\Psi; \vdash \mathbf{1}; \Theta}\ \mathbf{1}\text{-R}$$

$$\frac{}{\Psi; \bot \vdash ; \Theta}\ \bot\text{L} \qquad \frac{\Psi; \Gamma \vdash \Delta; \Theta}{\Psi; \Gamma \vdash \bot, \Delta; \Theta}\ \bot\text{R}$$

$$\frac{}{\Psi; \Gamma \vdash \top, \Delta; \Theta}\ \top\text{-R}$$

$$\frac{}{\Psi; \Gamma, \mathbf{0} \vdash \Delta; \Theta}\ \mathbf{0}\text{-L}$$

$$\frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta}{\Psi; \Gamma, \neg\phi \vdash \Delta; \Theta}\ \neg\text{-L} \qquad \frac{\Psi; \Gamma, \phi \vdash \Delta; \Theta}{\Psi; \Gamma \vdash \neg\phi, \Delta; \Theta}\ \neg\text{-R}$$

$$\frac{\Psi; \Gamma, \phi, \psi \vdash \Delta; \Theta}{\Psi; \Gamma, \phi \otimes \psi \vdash \Delta; \Theta}\ \otimes\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta \quad \Psi; \Gamma' \vdash \psi, \Delta; \Theta}{\Psi; \Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'; \Theta}\ \otimes\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \Delta; \Theta}{\Psi; \Gamma, \phi \& \psi \vdash \Delta; \Theta} \quad \frac{\Gamma, \psi \vdash \Delta; \Theta}{\Psi; \Gamma, \phi \& \psi \vdash \Delta; \Theta}\ \&\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta \quad \Psi; \Gamma \vdash \phi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \& \psi, \Delta; \Theta}\ \&\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \Delta; \Theta \quad \Psi; \Gamma, \psi \vdash \Delta; \Theta}{\Psi; \Gamma, \phi \oplus \psi \vdash \Delta; \Theta}\ \oplus\text{-L} \qquad \frac{\Psi; \Gamma \vdash \psi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \oplus \psi, \Delta; \Theta} \quad \frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \oplus \psi, \Delta; \Theta}\ \oplus\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \Delta; \Theta \quad \Psi; \Gamma', \psi \vdash \Delta'; \Theta}{\Psi; \Gamma, \Gamma', \phi \wp \psi \vdash \Delta, \Delta'; \Theta}\ \wp\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi, \psi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \wp \psi, \Delta; \Theta}\ \wp\text{-R}$$

$$\frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta \quad \Psi; \Gamma', \psi \vdash \Delta'; \Theta}{\Psi; \Gamma, \Gamma', \phi \multimap \psi \vdash \Delta, \Delta'; \Theta}\ \multimap\text{-L} \qquad \frac{\Psi; \Gamma, \phi \vdash \psi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \multimap \psi, \Delta; \Theta}\ \multimap\text{-R}$$

$$\frac{\Psi; \Gamma \vdash \phi, \Delta; \Theta \quad \Psi; \Gamma, \psi \vdash \Delta; \Theta}{\Psi; \Gamma, \phi \leadsto \psi \vdash \Delta; \Theta}\ \leadsto\text{-L} \qquad \frac{\Psi; \Gamma, \phi \vdash \Delta; \Theta}{\Psi; \Gamma \vdash \phi \leadsto \psi, \Delta; \Theta}\ \leadsto\text{-R} \quad \frac{\Psi; \Gamma \vdash \psi, \Delta; \Theta}{\Psi; \Gamma \vdash \phi \leadsto \psi, \Delta; \Theta}$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ !\text{-L} \qquad \frac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\ !\text{-R}$$

$$\frac{!\Gamma, \phi \vdash ?\Delta}{!\Gamma, ?\phi \vdash ?\Delta}\ ?\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash ?\phi, \Delta}\ ?\text{-R}$$

$$\frac{(\Psi, \phi); \Gamma \vdash \Delta; \Theta}{\Psi; (\Gamma, !\phi) \vdash \Delta; \Theta}\ !\text{-L} \qquad \frac{\Psi; \vdash \phi; \Theta}{\Psi; \vdash !\phi; \Theta}\ !\text{-R}$$

$$\frac{\Psi; \phi \vdash ; \Theta}{\Psi; ?\phi \vdash ; \Theta}\ ?\text{-L} \qquad \frac{\Psi; \Gamma \vdash \Delta; (\phi, \Theta)}{\Psi; \Gamma \vdash (?\phi, \Delta); \Theta}\ ?\text{-R}$$

$$\frac{\Psi; \Gamma \vdash (\phi, \Delta); (\phi, \Theta)}{\Psi; \Gamma \vdash \Delta; (\phi, \Theta)}\ ?D \qquad \frac{(\Psi, \phi); (\Gamma, \phi) \vdash \Delta; \Theta}{(\Psi, \phi); \Gamma \vdash \Delta; \Theta}\ !D$$

$$\frac{\Psi; \Gamma, \phi[t/x] \vdash \Delta; \Theta}{\Psi; \Gamma, \forall x.\phi \vdash \Delta; \Theta}\ \forall\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi[y/x], \Delta; \Theta}{\Psi; \Gamma \vdash \forall x.\phi, \Delta; \Theta}\ \forall\text{-R}^*$$

$$\frac{\Psi; \Gamma, \phi[y/x] \vdash \Delta; \Theta}{\Psi; \Gamma, \exists x.\phi \vdash \Delta; \Theta}\ \exists\text{-L}^* \qquad \frac{\Psi; \Gamma \vdash \phi[t/x], \Delta; \Theta}{\Psi; \Gamma \vdash \exists x.\phi, \Delta; \Theta}\ \exists\text{-R}$$

* y is not free in $\Gamma, \Delta, \Psi, \Theta$

B.3.3 A One-Sided Formulation

$$\frac{}{\vdash \phi, \phi^{\perp}}\ axiom \qquad \frac{\vdash \phi, \Delta \qquad \vdash \phi^{\perp}, \Delta'}{\vdash \Delta, \Delta'}\ cut$$

$$\frac{\vdash \Delta, \phi, \psi, \Delta'}{\vdash \Delta, \psi, \phi, \Delta'}\ ex\text{-R}$$

$$\frac{}{\vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\frac{\vdash \Delta}{\vdash \perp, \Delta}\ \perp\text{R}$$

$$\frac{}{\vdash \top, \Delta}\ \top\text{-R}$$

$$\frac{\vdash \phi, \Delta \qquad \vdash \psi, \Delta}{\vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

$$\frac{\vdash \phi, \Delta \qquad \vdash \phi, \Delta}{\vdash \phi \& \psi, \Delta}\ \&\text{-R}$$

$$\frac{\vdash \psi, \Delta}{\vdash \phi \oplus \psi, \Delta} \qquad \frac{\vdash \phi, \Delta}{\vdash \phi \oplus \psi, \Delta}\ \oplus\text{-R}$$

$$\frac{\vdash \phi, \psi, \Delta}{\vdash \phi \wp \psi, \Delta}\ \wp\text{-R}$$

$$\frac{\vdash \phi^{\perp}, \psi, \Delta}{\vdash \phi \multimap \psi, \Delta}\ \multimap\text{-R}$$

$$\frac{\vdash \phi, ?\Delta}{\vdash !\phi, ?\Delta}\ !\text{-R}$$

$$\frac{\vdash \phi, \Delta}{\vdash ?\phi, \Delta}\ ?\text{-R}$$

$$\frac{\vdash \Delta}{\vdash ?\phi, \Delta}\ W?\text{-R}$$

$$\frac{\vdash ?\phi, ?\phi, \Delta}{\vdash ?\phi, \Delta}\ C\text{-R}$$

$$\frac{\vdash \phi[y/x], \Delta}{\vdash \forall x.\phi, \Delta}\ \forall\text{-R}^*$$

$$\frac{\vdash \phi[t/x], \Delta}{\vdash \exists x.\phi, \Delta}\ \exists\text{-R}$$

* y is not free in Γ, Δ

B.3.4 One-sided Implicit-weakening fragment

$$\frac{\vdash \Gamma \qquad \vdash B, \Delta}{\vdash A_? \otimes B, \Gamma, \Delta}\ \otimes_{?l} \qquad\qquad \frac{\vdash A, \Gamma \qquad \vdash \Delta}{\vdash A \otimes B_?, \Gamma, \Delta}\ \otimes_{?r}$$

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash A_? \otimes B_?, \Gamma, \Delta}\ \otimes_{?lr}$$

$$\frac{\vdash \Gamma_l \qquad \vdash B, \Gamma_r}{\vdash A_? \otimes B, \Gamma_{lr}}\ \&_{?l} \qquad\qquad \frac{\vdash A, \Gamma_l \qquad \vdash \Gamma_r}{\vdash A \otimes B_?, \Gamma_{lr}}\ \&_{?r}$$

$$\frac{\vdash \Gamma_l \qquad \vdash \Gamma_r}{\vdash A_? \otimes B_?, \Gamma_{lr}}\ \&_{?lr}$$

$$\frac{\vdash \Gamma}{\vdash A_? \oplus B, \Gamma}\ 1\oplus_? \qquad\qquad \frac{\vdash \Gamma}{\vdash A \oplus B_?, \Gamma}\ 2\oplus_?$$

$$\frac{\vdash B, \Gamma}{\vdash A_? \wp B, \Gamma}\ \wp_{?l} \qquad\qquad \frac{\vdash A, \Gamma}{\vdash A \wp B_?, \Gamma}\ \wp_{?r}$$

$$\frac{\vdash \Gamma}{\vdash A_? \wp B_?, \Gamma}\ \wp_{?lr} \qquad\qquad \frac{\vdash ?\Gamma}{\vdash !A, ?\Gamma}\ !\text{-storage}_?$$

where:
- $?\Gamma$ denotes a sequent containing only ?-labels
- $A_?$ and $B_?$ denote ?-labels and a constant $\perp$
- $\Gamma_l = \Delta, ?\Sigma, ?\Theta_l$ and $\Gamma_r = \Delta, ?\Sigma, ?\Theta_r$ where $?\Theta_l \cap ?\Theta_r = \emptyset$ and $\Gamma_{lr} = \Delta, ?\Sigma, ?\Theta_l, ?\Theta_r$
- A and B denote arbitrary formulas (including ?-labels and $\perp$).

B.3.5 One-sided Formulation Without Structural Rules

axiom $\qquad \dfrac{}{\vdash ?\Delta, \phi, \phi^\perp}$

Cut $\qquad \dfrac{\vdash \Gamma, \phi \qquad \vdash \Delta, \phi^\perp}{\vdash \Gamma, \Delta}$

1 $\qquad \dfrac{}{\vdash ?\Delta, \mathbf{1}}$

$\perp$	$\dfrac{\vdash \Gamma}{\vdash \Gamma, \perp}$
$\top$	$\dfrac{}{\vdash \Gamma, \top}$
$\otimes$	$\dfrac{\vdash ?\Delta, \Sigma, \phi, \quad \vdash ?\Delta, \psi, \Gamma}{\vdash ?\Delta, \Sigma, \phi \otimes \psi, \Gamma}$
$\&$	$\dfrac{\vdash \Sigma, \phi \quad \vdash \Sigma, \psi}{\vdash \Sigma, \phi \& \psi}$
$\oplus$	$\dfrac{\vdash \Gamma, \psi}{\vdash \Gamma, \phi \oplus \psi} \qquad \dfrac{\vdash \Gamma, \psi}{\vdash \Gamma, \phi \oplus \psi}$
$\wp$	$\dfrac{\vdash \Gamma, \phi, \psi}{\vdash \Gamma, \phi \wp \psi}$
$!S$	$\dfrac{\vdash ?\Gamma, \phi}{\vdash ?\Gamma, !\phi}$
$?D$	$\dfrac{\vdash \Gamma, ?\phi, \phi}{\vdash \Gamma, ?\phi}$
$\forall$	$\dfrac{\vdash \Gamma, \phi[y/x]}{\vdash \Gamma, \forall x.\phi}$
$\exists$	$\dfrac{\vdash \Gamma, \phi[t/x]}{\vdash \Gamma, \exists x.\phi}$

* y is not free in Γ, Δ

B.3.6 CLL1: A One-sided Formulation

axiom	$\dfrac{}{\vdash ?\Delta, \phi, \phi^{\perp}}$
1	$\dfrac{}{\vdash ?\Delta, \mathbf{1}}$
$\perp$	$\dfrac{\vdash ?\Delta, \Gamma}{\vdash ?\Delta, \Gamma, \perp}$
$\top$	$\dfrac{}{\vdash ?\Delta, \Gamma, \top}$
$\otimes$	$\dfrac{\vdash ?\Delta, \Sigma, \phi, \quad \vdash ?\Delta, \psi, \Gamma}{\vdash ?\Delta, \Sigma, \phi \otimes \psi, \Gamma}$
$\&$	$\dfrac{\vdash ?\Delta, \Sigma, \phi \quad \vdash ?\Delta, \Sigma, \psi}{\vdash ?\Delta, \Sigma, \phi \& \psi}$
$\oplus$	$\dfrac{\vdash ?\Delta, \Gamma, \psi}{\vdash ?\Delta, \Gamma, \phi \oplus \psi} \qquad \dfrac{\vdash ?\Delta, \Gamma, \psi}{\vdash ?\Delta, \Gamma, \phi \oplus \psi}$
$\wp$	$\dfrac{\vdash ?\Delta, \Gamma, \phi, \psi}{\vdash ?\Delta, \Gamma, \phi \wp \psi}$
$!S$	$\dfrac{\vdash ?\Gamma, \phi}{\vdash ?\Gamma, !\phi}$

$?D \qquad \dfrac{\vdash ?\Delta, \Gamma, ?\phi, \phi}{\vdash ?\Delta, \Gamma, ?\phi}$

$\forall \qquad \dfrac{\vdash ?\Delta, \Gamma, \phi[y/x]}{\vdash ?\Delta, \Gamma, \forall x.\phi}$

$\exists \qquad \dfrac{\vdash ?\Delta, \Gamma, \phi[t/x]}{\vdash ?\Delta, \Gamma, \exists x.\phi}$

* y is not free in $\Gamma, ?\Delta$

B.4 Intuitionistic Linear Logic

B.4.1 ILL: A sequent calculus

$$\dfrac{}{\phi \vdash \phi}\ axiom \qquad \dfrac{\Gamma \vdash \phi \quad \Gamma', \phi \vdash F}{\Gamma, \Gamma' \vdash F}\ cut$$

$$\dfrac{\Gamma, \phi, \psi, \Gamma' \vdash F}{\Gamma, \psi, \phi, \Gamma' \vdash F}\ ex\text{-L}$$

$$\dfrac{\Gamma \vdash F}{\Gamma, \mathbf{1} \vdash F}\ \mathbf{1}\text{-L} \qquad \dfrac{}{\vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\dfrac{}{\bot \vdash}\ \text{L}^{\bot} \qquad \dfrac{\Gamma \vdash}{\Gamma \vdash \bot}\ \text{R}^{\bot}$$

$$\dfrac{}{\Gamma \vdash \top}\ \top\text{-R}$$

$$\dfrac{}{\Gamma, \mathbf{0} \vdash F}\ \mathbf{0}\text{-L}$$

$$\dfrac{\Gamma \vdash \phi}{\Gamma, \phi^{\bot} \vdash}\ \bot\text{-L} \qquad \dfrac{\Gamma, \phi \vdash}{\Gamma \vdash \phi^{\bot}}\ \bot\text{-R}$$

$$\dfrac{\Gamma, \phi, \psi \vdash F}{\Gamma, \phi \otimes \psi \vdash F}\ \otimes\text{-L} \qquad \dfrac{\Gamma \vdash \phi \quad \Gamma' \vdash \psi}{\Gamma, \Gamma' \vdash \phi \otimes \psi}\ \otimes\text{-R}$$

$$\dfrac{\Gamma, \phi \vdash F}{\Gamma, \phi \& \psi \vdash F} \qquad \dfrac{\Gamma, \psi \vdash F}{\Gamma, \phi \& \psi \vdash F}\ \&\text{-L} \qquad \dfrac{\Gamma \vdash \phi \quad \Gamma \vdash \phi}{\Gamma \vdash \phi \& \psi}\ \&\text{-R}$$

$$\dfrac{\Gamma, \phi \vdash \Delta \quad \Gamma, \psi \vdash F}{\Gamma, \phi \oplus \psi \vdash F}\ \oplus\text{-L} \qquad \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \phi \oplus \psi} \qquad \dfrac{\Gamma \vdash \phi}{\Gamma \vdash \phi \oplus \psi}\ \oplus\text{-R}$$

$$\dfrac{\Gamma, \phi \vdash F \quad \Gamma, \psi \vdash}{\Gamma, \phi \wp \psi \vdash F} \qquad \dfrac{\Gamma, \phi \vdash \quad \Gamma, \psi \vdash F}{\Gamma, \phi \wp \psi \vdash F}\ \wp\text{-L}$$

$$\dfrac{\Gamma \vdash \phi \quad \Gamma', \psi \vdash F}{\Gamma, \Gamma', \phi \multimap \psi \vdash F}\ \multimap l \qquad \dfrac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \multimap \psi}\ \multimap\text{-R}$$

$$\dfrac{\Gamma, \phi \vdash F}{\Gamma, !\phi \vdash F}\ !\text{-L} \qquad \dfrac{!\Gamma \vdash \phi}{!\Gamma \vdash !\phi}\ !\text{-R}$$

$$\dfrac{!\Gamma, \phi \vdash ?F}{!\Gamma, ?\phi \vdash ?F}\ ?\text{-L} \qquad \dfrac{\Gamma \vdash \phi}{\Gamma \vdash ?\phi}\ ?\text{-R}$$

$$\dfrac{\Gamma \vdash F}{\Gamma, !\phi \vdash F}\ W!\text{-L} \qquad \dfrac{\Gamma \vdash}{\Gamma \vdash ?\phi}\ W?\text{-R}$$

$$\frac{\Gamma, !\phi, !\phi \vdash F}{\Gamma, !\phi \vdash F}\ C\text{-L}$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^* \qquad \frac{\Gamma \vdash \phi[t/x]}{\Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ

B.4.2 JLV: A SC with Four Zones

A sequent always has the form: $\Psi; \Gamma \vdash F;$ or $\Psi; \Gamma \vdash\ ; F$, which are interpreted as $!\Psi, \Gamma \vdash F$ and $!\Psi, \Gamma \vdash ?F$ respectively. The symbol δ denotes either $F; \cdot$ or $\cdot; F$.

$$\frac{}{\Psi; \phi \vdash \phi;}\ axiom \qquad \frac{\Psi; \Gamma \vdash \phi; \qquad \Psi; \Gamma', \phi \vdash \delta}{\Psi; \Gamma, \Gamma' \vdash \delta}\ Cut$$

$$\frac{\Psi; \vdash \phi; \qquad (\Psi, \phi); \Gamma \vdash \delta}{\Psi; \Gamma \vdash \delta}\ Cut! \qquad \frac{\Psi; \Gamma \vdash\ ; \phi \qquad \Psi; \phi \vdash\ ; F}{\Psi; \Gamma \vdash\ ; F}\ Cut?$$

$$\frac{\Psi; \Gamma \vdash \delta}{\Psi; \Gamma, \mathbf{1} \vdash \delta}\ \mathbf{1}\text{-L} \qquad \frac{}{\Psi; \vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\frac{}{\Psi; \bot \vdash\ ; F}\ \text{L}^{\bot} \qquad \frac{\Gamma \vdash\ ; p}{\Gamma \vdash \bot}\ \text{R}^{\bot}$$

$$\frac{}{\Psi; \Gamma \vdash \top;}\ \top\text{-R}$$

$$\frac{}{\Psi; \Gamma, \mathbf{0} \vdash \delta}\ \mathbf{0}\text{-L}$$

$$\frac{\Psi; \Gamma \vdash \phi;}{\Psi; \Gamma, \phi^{\bot} \vdash\ ; F}\ \bot\text{-L} \qquad \frac{\Psi; \Gamma, \phi \vdash\ ; p}{\Psi; \Gamma \vdash \phi^{\bot};}\ \bot\text{-R}$$

$$\frac{\Psi; \Gamma, \phi, \psi \vdash \delta}{\Psi; \Gamma, \phi \otimes \psi \vdash \delta}\ \otimes\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi; \qquad \Psi; \Gamma' \vdash \psi;}{\Psi; \Gamma, \Gamma' \vdash \phi \otimes \psi;}\ \otimes\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \delta}{\Psi; \Gamma, \phi \& \psi \vdash \delta} \qquad \frac{\Psi; \Gamma, \psi \vdash \delta}{\Psi; \Gamma, \phi \& \psi \vdash \delta}\ \&\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi; \qquad \Psi; \Gamma \vdash \phi;}{\Psi; \Gamma \vdash \phi \& \psi;}\ \&\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \delta \qquad \Psi; \Gamma, \psi \vdash \delta}{\Psi; \Gamma, \phi \oplus \psi \vdash \delta}\ \oplus\text{-L} \qquad \frac{\Psi; \Gamma \vdash \psi;}{\Psi; \Gamma \vdash \phi \oplus \psi;} \qquad \frac{\Psi; \Gamma \vdash \phi;}{\Psi; \Gamma \vdash \phi \oplus \psi;}\ \oplus\text{-R}$$

$$\frac{\Psi; \Gamma \vdash \phi; \qquad \Psi; \Gamma', \psi \vdash \delta}{\Psi; \Gamma, \Gamma', \phi \multimap \psi \vdash \delta}\ \multimap l \qquad \frac{\Psi; \Gamma, \phi \vdash \psi}{\Psi; \Gamma \vdash \phi \multimap \psi}\ \multimap\text{-R}$$

$$\frac{(\Psi, \phi); \Gamma \vdash \delta}{\Psi; \Gamma, !\phi \vdash \delta}\ !\text{-L} \qquad \frac{\Psi; \vdash \phi;}{\Psi; \vdash !\phi;}\ !\text{-R}$$

$$\frac{\Psi; \phi \vdash\ ; F}{\Psi; ?\phi \vdash\ ; F}\ ?\text{-L} \qquad \frac{\Psi; \Gamma \vdash\ ; \phi}{\Psi; \Gamma \vdash ?\phi;}\ ?\text{-R}$$

$$\frac{(\Psi, F); \Gamma, F \vdash \delta}{(\Psi, F); \Gamma \vdash \delta}\ DL \qquad \frac{\Psi; \Gamma \vdash F;}{\Psi; \Gamma \vdash\ ; F}\ DR$$

$$\frac{\Psi;\Gamma,\phi[t/x]\vdash\delta}{\Psi;\Gamma,\forall x.\phi\vdash\delta}\ \forall\text{-L}\qquad\qquad \frac{\Psi;\Gamma\vdash\phi[y/x];}{\Psi;\Gamma\vdash\forall x.\phi;}\ \forall\text{-R}^*$$

$$\frac{\Psi;\Gamma,\phi[y/x]\vdash\delta}{\Psi;\Gamma,\exists x.\phi\vdash\delta}\ \exists\text{-L}^*\qquad\qquad \frac{\Psi;\Gamma\vdash\phi[t/x];}{\Psi;\Gamma\vdash\exists x.\phi;}\ \exists\text{-R}$$

* y is not free in Γ,Δ,ϕ

B.4.3 An specialized Fragment

$$\frac{}{\phi,!\Gamma\vdash\phi}\ axiom$$

$$\frac{\Gamma_1,!\Gamma\vdash\phi\quad \Gamma_2,!\Gamma\vdash\psi}{\Gamma_1,\Gamma_2,!\Gamma\vdash\phi\otimes\psi}\ \otimes r$$

$$\frac{}{\Gamma\multimap A,\Gamma_1,\Gamma_2\vdash\top}\ \top r$$

$$\frac{\Gamma\vdash\phi}{\phi\multimap A,\Gamma\vdash A}\ \multimap l\qquad\qquad \frac{\Gamma,\phi\vdash\psi}{\Gamma\vdash\phi\multimap\psi}\ \multimap r$$

$$\frac{\Gamma,\phi\vdash F}{\Gamma,\phi\&\psi\vdash F}\qquad \frac{\Gamma,\psi\vdash F}{\Gamma,\phi\&\psi\vdash F}\ \&l\qquad\qquad \frac{\Gamma\vdash\phi\quad \Gamma\vdash\phi}{\Gamma\vdash\phi\&\psi}\ \&r$$

$$\frac{\Gamma\vdash\psi}{\Gamma\vdash\phi\oplus\psi}\qquad \frac{\Gamma\vdash\phi}{\Gamma\vdash\phi\oplus\psi}\ \oplus\text{r}$$

$$\frac{\phi,!\phi,\Gamma\vdash G}{!\phi,\Gamma\vdash G}\ !l\qquad\qquad \frac{!\Gamma\vdash\phi}{!\Gamma\vdash!\phi}\ !r$$

$$\frac{\Gamma,\phi[t/x]\vdash F}{\Gamma,\forall x.\phi\vdash F}\ \forall\text{l}\qquad\qquad \frac{\Gamma\vdash\phi[y/x]}{\Gamma\vdash\forall x.\phi}\ \forall\text{r}^*$$

$$\frac{\Gamma\vdash\phi[t/x]}{\Gamma\vdash\exists x.\phi}\ \exists\text{r}$$

* y is not free in Γ,Δ,ϕ

B.5 Multiple-conclusioned Intuitionistic Logic

B.5.1 A Standard Intuitionistic Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\ axiom$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \wedge r$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee l \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee r$$

$$\frac{\Gamma, \neg A \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ \neg l \qquad \frac{\Gamma, A \vdash}{\Gamma \vdash \neg A, \Delta}\ \neg r$$

$$\frac{\Gamma, A \to B \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\ \to l \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B, \Delta}\ \to r$$

$$\frac{\Gamma, \forall x A, A[t/x] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta}\ \forall l \qquad \frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall x A, \Delta}\ \forall^* r$$

$$\frac{\Gamma, A[y/x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta}\ \exists^* l \qquad \frac{\Gamma \vdash A[t/x], \exists x A, \Delta}{\Gamma \vdash \exists x A, \Delta}\ \exists r$$

* y is not free in conclusion of $\forall r$ and $\exists l$ rules.

B.5.2 MLJT: Multiple-conclusioned LJT

$$\frac{}{A, \Gamma \vdash A, \Delta}\ axiom \qquad \frac{}{\bot, \Gamma \vdash \Delta}\ \bot$$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta}\ \wedge r$$

$$\frac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}\ \vee l \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}$$

$$\frac{B, a, \Gamma \vdash \Delta}{a \supset B, a, \Gamma \vdash \Delta}\ \supset_1 l \quad \text{a being atomic}$$

$$\frac{C \supset (D \supset B), \Gamma \vdash \Delta}{(C \wedge D) \supset B, \Gamma \vdash \Delta}\ \supset_2 l$$

$$\frac{C \supset B, D \supset B, \Gamma \vdash \Delta}{(C \vee D) \supset B, \Gamma \vdash \Delta}\ \supset_3 l$$

$$\frac{D \supset B, \Gamma \vdash C \supset D \quad B, \Gamma \vdash \Delta}{(C \supset D) \supset B, \Gamma \vdash \Delta}\ \supset_4 l \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \supset \psi, \Delta}\ \supset r$$

B.5.3 A Variant of Multiple-conclusioned Intuitionistic Logic

$$\frac{}{\Gamma, \phi \vdash \phi, \Delta}\ axiom \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma, \phi \vdash \Delta}{\Gamma \vdash \Delta}\ cut$$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta}\ \wedge r$$

$$\frac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}\ \vee l \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee r$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \to \psi, \Gamma \vdash \Delta}\ \to l \qquad \frac{\phi, \Gamma \vdash \psi}{\Gamma \vdash \phi \to \psi, \Delta}\ \to r$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\ \neg l \qquad \frac{\Gamma, \phi \vdash}{\Gamma \vdash \neg\phi, \Delta}\ \neg r$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall l \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall r^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists l^* \qquad \frac{\Gamma \vdash \phi[t/x], \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists r$$

* y is not free in conclusion of $\forall r$ and $\exists l$ rules.

B.6 Minimal Linear Logic

B.6.1 ILL Sequent Calculus

$$\frac{}{\phi \vdash \phi}\ axiom \qquad \frac{\Gamma \vdash \phi \quad \Gamma', \phi \vdash F}{\Gamma, \Gamma' \vdash F}\ Cut$$

$$\frac{\Gamma \vdash F}{\Gamma, \mathbf{1} \vdash F}\ \mathbf{1}\text{-L} \qquad \frac{}{\vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\frac{}{\Gamma \vdash \top}\ \top\text{-R}$$

$$\frac{}{\Gamma, \mathbf{0} \vdash F}\ \mathbf{0}\text{-L}$$

$$\frac{\Gamma, \phi, \psi \vdash F}{\Gamma, \phi \otimes \psi \vdash F}\ \otimes\text{-L} \qquad \frac{\Gamma \vdash \phi \quad \Gamma' \vdash \psi}{\Gamma, \Gamma' \vdash \phi \otimes \psi}\ \otimes\text{-R}$$

$$\frac{\Gamma, \phi \vdash F}{\Gamma, \phi \& \psi \vdash F} \quad \frac{\Gamma, \psi \vdash F}{\Gamma, \phi \& \psi \vdash F}\ \&\text{-L} \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \phi}{\Gamma \vdash \phi \& \psi}\ \&\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta \quad \Gamma, \psi \vdash F}{\Gamma, \phi \oplus \psi \vdash F}\ \oplus\text{-L} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \oplus \psi} \quad \frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \oplus \psi}\ \oplus\text{-R}$$

$$\frac{\Gamma \vdash \phi \quad \Gamma', \psi \vdash F}{\Gamma, \Gamma', \phi \multimap \psi \vdash F}\ \multimap\text{-L} \qquad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \multimap \psi}\ \multimap\text{-R}$$

$$\frac{\Gamma, \phi \vdash F}{\Gamma, !\phi \vdash F}\ !\text{-L} \qquad \frac{!\Gamma \vdash \phi}{!\Gamma \vdash !\phi}\ !\text{-R}$$

$$\frac{\Gamma \vdash F}{\Gamma, !\phi \vdash F}\ W!\text{-L}$$

$$\frac{\Gamma, !\phi, !\phi \vdash F}{\Gamma, !\phi \vdash F}\ C\text{-L}$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x]}{\Gamma \vdash \forall x.\phi}\ \forall\text{-R}^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^{*} \qquad \frac{\Gamma \vdash \phi[t/x]}{\Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

* y is not free in Γ, Δ, ϕ

B.6.2 ILV: A SC with Three Zones

$$\frac{}{\Psi; \phi \vdash \phi}\ \mathit{axiom}$$

$$\frac{\Psi; \Gamma \vdash \phi \quad \Psi\ \Gamma', \phi \vdash F}{\Psi; \Gamma, \Gamma' \vdash F}\ \mathit{Cut} \qquad \frac{\Psi; \vdash \phi \quad (\Psi, \phi); \Gamma \vdash F}{\Psi; \Gamma \vdash F}\ \mathit{Cut!}$$

$$\frac{\Psi; \Gamma \vdash F}{\Psi; \Gamma, \mathbf{1} \vdash F}\ \mathbf{1}\text{-L} \qquad \frac{}{\Psi; \vdash \mathbf{1}}\ \mathbf{1}\text{-R}$$

$$\frac{}{\Psi; \Gamma \vdash \top}\ \top\text{-R}$$

$$\frac{}{\Psi; \Gamma, \mathbf{0} \vdash F}\ \mathbf{0}\text{-L}$$

$$\frac{\Psi; \Gamma, \phi, \psi \vdash F}{\Psi; \Gamma, \phi \otimes \psi \vdash F}\ \otimes\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi \quad \Psi; \Gamma' \vdash \psi}{\Psi; \Gamma, \Gamma' \vdash \phi \otimes \psi}\ \otimes\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash F}{\Psi; \Gamma, \phi \& \psi \vdash F} \qquad \frac{\Psi; \Gamma, \psi \vdash F}{\Psi; \Gamma, \phi \& \psi \vdash F}\ \&\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi \quad \Psi; \Gamma \vdash \phi}{\Psi; \Gamma \vdash \phi \& \psi}\ \&\text{-R}$$

$$\frac{\Psi; \Gamma, \phi \vdash \Delta \quad \Psi; \Gamma, \psi \vdash F}{\Psi; \Gamma, \phi \oplus \psi \vdash F}\ \oplus\text{-L} \qquad \frac{\Psi; \Gamma \vdash \psi}{\Psi; \Gamma \vdash \phi \oplus \psi} \qquad \frac{\Psi; \Gamma \vdash \phi}{\Psi; \Gamma \vdash \phi \oplus \psi}\ \oplus\text{-R}$$

$$\frac{\Psi; \Gamma \vdash \phi \quad \Psi; \Gamma', \psi \vdash F}{\Psi; \Gamma, \Gamma', \phi \multimap \psi \vdash F}\ \multimap\text{-L} \qquad \frac{\Psi; \Gamma, \phi \vdash \psi}{\Psi; \Gamma \vdash \phi \multimap \psi}\ \multimap\text{-R}$$

$$\frac{(\Psi, \phi); \Gamma \vdash F}{\Psi; \Gamma, !\phi \vdash F}\ !\text{-L} \qquad \frac{\Psi; \vdash \phi}{\Psi; \vdash !\phi}\ !\text{-R}$$

$$\frac{(\Psi, \phi); \Gamma, \phi \vdash F}{(\Psi, \phi); \Gamma \vdash F}\ \text{structural rule}$$

$$\frac{\Psi; \Gamma, \phi[t/x] \vdash F}{\Psi; \Gamma, \forall x.\phi \vdash F}\ \forall\text{-L} \qquad \frac{\Psi; \Gamma \vdash \phi[y/x]}{\Psi; \Gamma \vdash \forall x.\phi}\ \forall\text{-R}^{*}$$

$$\frac{\Psi; \Gamma, \phi[y/x] \vdash F}{\Psi; \Gamma, \exists x.\phi \vdash F}\ \exists\text{-L}^{*} \qquad \frac{\Psi; \Gamma \vdash \phi[t/x]}{\Psi; \Gamma \vdash \exists x.\phi}\ \exists\text{-R}$$

* y is not free in $\Gamma, \Delta, \phi, \Psi$

B.7 Modal Logic

B.7.1 S4: A *SC* for Modal Logic

$$\frac{}{A, \Gamma \vdash A, \Delta}\ axiom$$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta}\ \wedge r$$

$$\frac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}\ \vee l \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta}\ \vee r$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \rightarrow \psi, \Gamma \vdash \Delta} \rightarrow l \qquad \frac{\phi, \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \rightarrow \psi, \Delta} \rightarrow r$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\ \neg l \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta}\ \neg r$$

$$\frac{\Gamma, \phi[t/x] \vdash F}{\Gamma, \forall x.\phi \vdash F}\ \forall l \qquad \frac{\Gamma \vdash \phi[y/x], \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall r^*$$

$$\frac{\Gamma, \phi[y/x] \vdash F}{\Gamma, \exists x.\phi \vdash F}\ \exists l^* \qquad \frac{\Gamma \vdash \phi[t/x], \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists r$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, \Box\phi \vdash \Delta}\ \Box l \qquad \frac{\Gamma* \vdash \phi, \Delta*}{\Gamma \vdash \Box\phi, \Delta}\ \Box r$$

$$\frac{\Gamma*, \phi \vdash \Delta*}{\Gamma, \Diamond\phi \vdash \Delta}\ \Diamond l \qquad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash \Diamond\phi, \Delta}\ \Diamond r$$

* y is not free in the conclusion of r∀ and l∃

$\Gamma* \stackrel{\text{def}}{=} \{\Box B | \Box B \in \Gamma\}$
$\Delta* \stackrel{\text{def}}{=} \{\Diamond B | \Diamond B \in \Delta\}$

B.7.2 S4^S: A *SC* for Modal Logic

$$\frac{}{\vdash A, \neg A, \Gamma}\ axiom$$

$$\frac{\vdash A, B, \Gamma}{\vdash A \vee B, \Gamma}\ \vee \qquad \frac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A \wedge B, \Gamma}\ \wedge$$

$$\frac{\vdash A, \Diamond\Delta}{\vdash \Box A, \Diamond\Delta, \Gamma}\ \Box, \text{JUMP} \qquad \frac{\vdash A, \Diamond A, \Gamma}{\vdash \Diamond A, \Gamma}\ \Diamond$$

B.7.3 KT^S: A *SC* for Modal Logic

$$\frac{}{\vdash A, \neg A, \Gamma}\ axiom$$

$$\frac{\vdash A, B, \Gamma}{\vdash A \vee B, \Gamma}\ \vee \qquad \frac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A \wedge B, \Gamma}\ \wedge$$

$$\frac{\vdash A, \Delta}{\vdash \Box A, \Diamond\Delta, \Gamma}\ \Box, \text{JUMP} \qquad \frac{\vdash A, \Diamond A, \Gamma}{\vdash \Diamond A, \Gamma}\ \Diamond$$

B.8 Light Linear Logic

B.8.1 LLL: A One-sided Formulation

$$\frac{}{\vdash A, A^{\perp}}\ \text{axiom}$$

$$\frac{\vdash A, \Gamma \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

$$\frac{\vdash \Gamma, \bigoplus_{i=1}^{m} A_i, \bigoplus_{i=1}^{n} B_i}{\vdash \Gamma, \bigoplus_{i=1}^{m} B_i, \bigoplus_{i=1}^{n} A_i}\ \text{M-exchange}$$

$$\frac{\overset{1 \leq i < n}{\vdash \Gamma, A_1 \oplus ... \oplus A_i \oplus A_{i+1} \oplus ... \oplus A_n}}{\vdash \Gamma, A_1 \oplus \ldots \oplus A_{i+1} \oplus A_i \oplus ... \oplus A_n}\ \text{A-exchange}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ \text{M-weakening}$$

$$\frac{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i}{\vdash \Gamma, \bigoplus_{i=1}^{m} A_i \oplus B}\ \text{A-weakening}$$

$$\frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \text{M-contraction}$$

$$\frac{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i \oplus B \oplus B}{\vdash \Gamma, \bigoplus_{i=1}^{n} A_i \oplus B}\ \text{A-contraction}$$

$$\frac{}{\vdash \Gamma, \perp}\ \perp$$

$$\frac{}{\vdash \Gamma, \top}\ \top$$

$$\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B}\ \wp$$

$$\frac{}{\vdash \mathbf{1}}\ \mathbf{1}$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \& B}\ \&$$

$$\frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ \oplus$$

$$\frac{\overset{n \geq 1}{\vdash B_1 \oplus ... \oplus Bn, A}}{\vdash ?B_1, \ldots ?Bn, !A}\ \text{"of course"}$$

$$\frac{\vdash B_1 \oplus ... \oplus B_n, A_1, ...A_m}{\vdash ?B_1, \ldots ?B_n, \S A_1, ...\S s A_m}\ \text{neutral}_1$$

$$\frac{\vdash B_1, ...B_n, A_1, ...A_m}{\vdash ?B_1, \ldots ?B_n, \S A_1, ...\S s A_m}\ \text{neutral}_2$$

$$\frac{\vdash \Gamma, A[y/x]}{\vdash \Gamma, \forall x A}\ \forall^{*}$$

$$\frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x A}\ \exists$$

* y is not free in Γ

B.9 A SC for Logic with Finitely Many Variables

$$\frac{}{\vdash \Gamma, A, \Delta, \neg A, \Xi}\ \textit{axiom}$$

$$\frac{}{\vdash \Gamma, \top, \Delta}\ \textit{axiom}$$

$$\frac{\vdash \Gamma, \phi \qquad \vdash \Gamma, \neg\phi}{\vdash \Gamma}\ \textit{cut}$$

$$\frac{\vdash \Gamma, \phi, \Delta \qquad \Gamma, \psi, \Delta}{\vdash \Gamma, \phi \wedge \psi, \Delta}\ \wedge$$

$$\frac{\vdash \Gamma, \phi, \psi, \Delta}{\vdash \Gamma, \phi \vee \psi, \Delta}\ \vee$$

$$\frac{\vdash \Gamma, A[t/x], \exists x.A, \Delta}{\vdash \Gamma, \exists x.A, \Delta}\ \exists$$

$$\frac{\vdash \Gamma, A[y/x]}{\vdash \Gamma, \forall x.A, \Delta}\ \forall$$

y is not free in the conclusion of rule $\forall$.