

Jump from Parallel to Sequential Proofs: Additives

Paolo Di Giamberardino

Laboratoire d'Informatique de Paris Nord

99, Avenue Jean Baptiste Clement 93430 Villetaneuse

digiambe@lipn.univ-paris13.fr

Abstract

In previous work, [DGF06] we introduced a framework for proof nets of the multiplicative fragment of Linear Logic, where *partially sequentialized* nets are allowed.

In this paper we extend this result to include additives, using a definition of proof net, called *J-proof net*, which corresponds to a typed version of an L-net of Faggian and Maurel.

In J-proof nets, sequentiality constraints are represented using the proof-net notion of *jumps*; by gradual insertion or removal of jumps then it is possible to characterize nets with different degrees of sequentiality. As a byproduct, we obtain then a proof of the sequentialization theorem.

Moreover, we provide a denotational model for J-proof nets which is an extension of the relational one, and we show that the “degree” of sequentiality of a J-proof net can be read-off from its semantics, by proving that the model is injective with respect to J-proof nets.

1 Introduction

Game models of computation interpret a program as a strategy in a game: usually such strategies appear as trees, (for instance, innocent Hyland-Ong strategies, see [HO00]), and composition between strategies yields a linearly ordered set of moves, which somehow describes the “trace” of the computation. Using tree strategies, it has been possible to solve the outstanding open problem of providing a fully abstract model of PCF, a paradigmatic sequential programming language based on λ -calculus ([HO00],[AMJ00]).

Several proposals have been made (see among others [HS02, AM99, Mel04]) in order to extend game semantics to represent more parallel forms of computation; in these approaches, strategies are no more trees, but more generally graphs, so that the order between the actions is not completely specified: the composition between graph strategies yields a partially ordered set of moves.

Recently, in the context of ludics (a game model based on linear logic, introduced by Girard in [Gir01]), a parallel generalization of ludics strategies called *L-nets* was proposed by Faggian and Maurel [FM05]. A design (i.e. a ludics strategy) can be thought of as a merging of two kinds of orders between actions, as pointed out by Faggian in [Fag02]:

- the *causal* (or *spatial*) order, representing the causal dependencies between actions;
- the *sequential* (or *temporal*) order, representing how causally independent actions are scheduled.

L-nets are built from standard designs by relaxing the sequential order between actions, in such a way to have still enough information to compute; a characteristic feature of this approach is to provide an homogeneous space of strategies with different levels of sequentiality, depending on the amount of sequential order they contain. Such a space has as extremes from one side parallel L-nets (with minimal sequentiality), from the other sequential L-nets (which correspond to designs, see [CF05]); by the way, designs are a special case of L-nets, as trees are a special case of graphs.

The definition of L-nets is directly inspired from the one of *proof net* (a graph representation of linear logic proofs introduced by Girard in [Gir87]); actually we could think of L-nets as a sort of abstract, untyped proof nets.

In [CF05] Faggian and Curien analyzed the relation between parallel and sequential L-nets using specific techniques coming from the framework of proof nets (namely sequentialization, desequenzialization, splitting theorems, and so on); in particular, they showed how to turn a graph strategy (L-net) into a tree strategy (design) by introducing a maximal amount of sequential order, and viceversa. Nevertheless, in the conclusions of the paper the following questions were still open :

- What is the notion of proof net underlying the one of L-net?
- Is it possible to turn *gradually* an L-net into a design, by introducing sequential order little by little (that is to move from graph strategies to tree strategies in a continuum) ?

The first motivation of the present paper is to provide answers for the questions above by introducing a new kind of proof nets called *J-proof nets*, which are the typed, logical counterpart of L-nets. By the insertion of special edges called *jumps*, representing sequentiality constraints, we can gradually increase the amount of sequential order on a J-proof net; in this way as in L-nets, we can characterize J-proof nets with different degrees of sequentiality.

A second, more general motivation behind the introduction of J-proof nets lies in the attempt to reconcile two different notions of *canonical representation of proofs* provided by linear logic, namely that of *proof net* and the one of *focusing proof*, as we point out in the following paragraphs.

Proof nets. In 1987, in the seminal paper [Gir87], Girard introduces linear logic (briefly *LL*) from a fine analysis of intuitionistic and classical logic; such a refinement provides a logical status to the structural rules (*weakening* and *contraction*) of sequent calculus (due to the introduction of the exponential connectives, ! and ?) and splits the usual propositional connectives (“and”, “or”) in two classes (the additives $\&$, \oplus , and the multiplicatives \otimes , \wp).

The deep insight on the standard connectives operated by *LL* allows to represent proofs as *graphs* called *proof nets*, whose nodes correspond to linear logic rules and which satisfy some specific geometrical properties (called *correctness criteria*); such a discovery brought to the fore the *geometrical* nature of proofs.

The main characteristic of proof nets is to be *modular*, *parallel* objects: in a proof net, there is no direct reference to the sequential succession of steps which brought to its construction.

As a consequence, a proof net turns out to be a canonical representative of a class of sequent calculus proofs equivalent modulo permutations of rules: *sequentialization*, one of the key results in the theory of proof nets, allows to recover a sequent calculus proof from a proof net, by proving that among the nodes of the proof net one can be chosen as a sequent calculus last rule; by iterated application of this property, it is possible to retrieve a proof in sequent calculus.

We must remark that outside multiplicative linear logic (briefly *MLL*), the beautiful theory of proof nets becomes less elegant and (in some cases) quite complicated: for instance, the search of a proper representation of additives in proof nets has been an open problem for a long time, only recently solved by Hughes and Van Glaabek in [HVG03].

Nevertheless, proof nets allows to eliminate those *naive* aspects of sequentiality which are not *naturally* inherent to the structure of proofs.

However there exists also another side of sequentiality, more intrinsic than the simple ordered succession of rules. Such a meaningful notion of sequentiality has been disclosed by the discovery inside linear logic of *polarities*.

Polarities. *LL*, tampering the structural rules with the exponential connectives, allows for the first time to define an involutive negation independent from structural rules, called *linear negation*.

Such a discovery enlightened the operational meaning of negation as a *change of viewpoint*. In computer science, a program is executed in a given environment; this process can be either analyzed from the point of view of the program or from the one of the environment, and negation is here the switch between these two positions.

It is worth mentioning that the above intuition contributed in a remarkable way to the birth of *game semantics*, which interprets computation as a game between two players, one representing the program, and the other representing the environment (see [AMJ00], [HO00]).

The refinement of usual negation inside *LL* was the starting point of a deep analysis of the logical notion of *duality*, which eventually brought to the discovery of *polarities*.

Multiplicative and additive connectives of *LL* naturally splits into two dual families:

- *Positives (or synchronous)*: \otimes, \oplus ;
- *Negatives (or asynchronous)*: $\wp, \&$.

A formula is *positive* (resp. *negative*) if its outermost connective is positive (resp. negative).

The difference between positive and negative formulas relies in the fact that, while the rules introducing negative formulas are *reversible* (that is, the conclusion of the rule implies the premises), the rules introducing positive formulas are *irreversible*.

Following this distinction, in [And92], Andreoli proved that any proof of linear logic can be transformed modulo permutations of rules into a proof which satisfies (bottom-up) the following discipline:

- i) negative formulas, if any, are decomposed immediately;
- ii) otherwise, one chooses a positive formula, and keeps decomposing it up to its negative subformulas.

Such proofs are called *focusing*.

The alternation of positive and negative steps provides then a canonical way to construct a sequent calculus proof, yielding an intrinsic, *non-trivial* notion of *time* in proofs, as pointed out by Girard in [Gir99].

Looking at focusing proofs through the lens of interaction, a positive cluster of rules appears as the act of posing a question to an Opponent by a Proponent, and a negative cluster of rules as the reception of an answer from the Opponent; if we apply linear negation, we switch the point of view, turning questions into answers and answers into questions.

To provide a canonical representation of focusing proofs in multiplicative-additive linear logic (briefly *MALL*), in [Gir00] Girard introduced a calculus called *hypersequentialized calculus* (briefly *HS*), using *synthetic connectives*, that is considering clusters of connectives of the same polarity as a single connective. The distinctive feature of *HS* is the presence of only two kinds of logical rules (the positive rule and the negative rule), one strictly alternating with the other.

Nevertheless, the hypersequentialized approach too has its limitations: mainly, it forces to leave aside the geometrical representation of proof nets, in spite of the simplicity and elegance of its multiplicative part.

Jumps: sequentializing *à la carte*. Clearly there is a mismatch regarding the nature of proofs between proof nets and hypersequentialized calculus. Firstly, proof nets exhibit an elegant geometrical structure which is absent in *HS*; furthermore, while proof nets are *timeless*, parallel objects, in hypersequentialized proofs there is an explicit marking of time, which makes them *sequential* in a strong, natural sense.

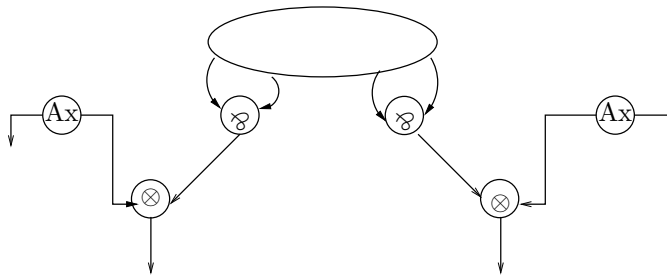
The spirit behind the present work is to try to reconcile this mismatch, by proposing a notion of proof net compatible with *HS* and recovering in this way polarities (and time) in a geometrical, parallel setting: here is where L-nets come into the picture.

It is well known that a design in ludics corresponds to an abstract, untyped hypersequentialized proof. What we would like to achieve is a notion of proof net which corresponds to the one of L-net, as an hypersequentialized proof corresponds to a design. In order to accomplish this task, we must find a counterpart for both the causal and the sequential ordering in the representation of proof nets. Concerning the causal ordering, if we restrict to *MALL*, we have already an answer in the *subformula* relation induced by the structure of the links in a proof net. To properly characterize sequential order in proof nets, we have to resort to the notion of *jump* (as observed in [FM05]).

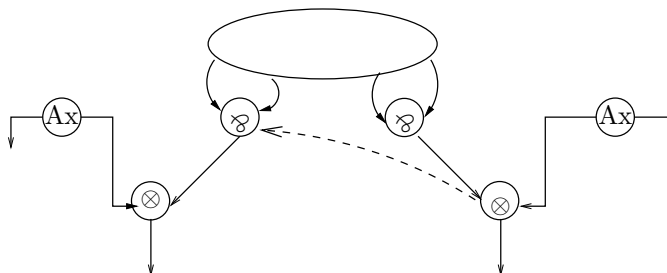
The idea of using edges to represent sequentiality constraints has been widely used in the study of correctness criteria for proof nets: in [Gir91] and [Gir96], Girard, as a part of the correctness criterion for proof nets, introduces *jumps*: if a link n is a \wp , $\&$, \perp or \vee link, a jump is an untyped edge between n and another link m , which represents a sequential ordering between n and m ; n precedes m (bottom-up) in every sequentialization.

Girard, in several occasions, suggested that it could be possible to retrieve a sequent calculus proof from a proof net just by fixing some temporal information on it, using jumps.

Let us try to make this point clearer with an example; consider the sketch of proof net below:

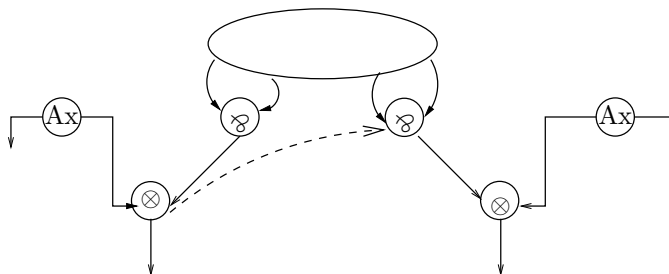


We remark that such a configuration is forbidden in sequent calculus: one must decide which one of the two \otimes is the last rule. To retrieve a proof then we draw a jump between the leftmost (negative) \wp and the rightmost (positive) \otimes , meaning that the corresponding \wp rule must precede (bottom-up) the corresponding \otimes rule in the sequentialization;

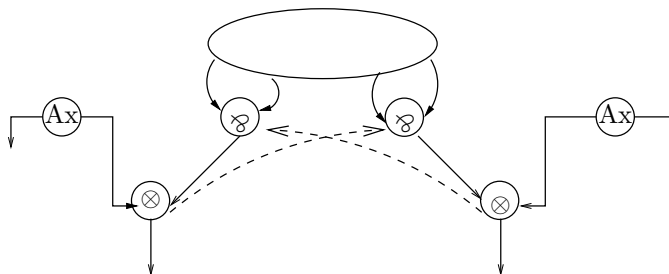


Now, the sequent calculus proof π induced by this proof net will have as last rule the leftmost \otimes , followed respectively by the leftmost \wp , the rightmost \otimes and the rightmost \wp ; we stress that such a proof respects the

focusing discipline. Instead of fixing the order in the way above, we could as well draw a jump between the the rightmost (negative) \wp and the leftmost (positive) \otimes , as below, obtaining a different focusing proof π' :



Furthermore, once fixed an order between links using jumps, some other choices become unavailable: namely one cannot draw both the jumps above at the same time, without creating a *cycle*, which would prevent us to obtain an order on the rules.



Once all possible jumps have been chosen, one directly retrieves in this way a focusing proof.

In previous work [DGF06, DGF08], realizing the proposal put forward by Girard, we provided a simple proof of sequentialization theorem for the multiplicative fragment of Linear Logic; we proved that by gradual insertion of jumps in a proof net, one increases step by step sequentiality information, until one directly retrieves a sequent calculus proof. Our proof relied on a technical result, that we called *arborisation lemma*, which provides a way to insert edges into a directed acyclic graph G , preserving a geometrical property called *switching acyclicity* (which corresponds to the correctness criterion of proof nets) and increasing at the same time the order induced by G .

J-proof nets. Building up on such previous results, in the present work :

- we introduce proof nets with jumps for the hypersequentialized calculus, called *J-proof nets*, by translating L-nets to a typed setting: the basic idea is to use jumps to represent sequential order;
- we extend the sequentialization method used in [DGF06, DGF08] for the multiplicative fragment in order to take into account additives, and we show that we can gradually remove or add sequentiality in a J-proof net, in the form of jumps, to get more parallel or more sequential proofs, in the style of L-nets. At any time, the information provided by jumps always makes possible to retrieve a fully sequentialized J-proof net, which directly corresponds to an hypersequentialized proof.

In this way, as in L-nets, we get an homogeneous space of J-proof nets with different degrees of sequentiality, having as extremes from one side J-proof nets with minimal sequentiality, from the other J-proof nets with maximal sequentiality, that is *HS* proofs; moreover, adding jumps provides a way to move in a continuum from one extreme to the other.

1.1 Outline of the paper

First, in section 2 we provide some background. We present the hypersequentialised calculus, together with an informal introduction to two important notions to understand additive connectives: the one of *slice* (see [Gir87]) and the one of *superposition*. Then we recall some graph theoretical definitions that we will use throughout the paper.

In section 3 we introduce our main objects of study: J-proof nets. As usual in proof nets theory, we first define a general class of graphs, called J-proof structures, and then we isolate among them the ones coming from *HS*, called J-proof nets, using a *correctness criterion*.

In section 4 we provide a formal definition in our setting of the “superposition” effect induced by additives, using the key notion of *sharing equivalence*; to define such equivalence we will import concepts and tools from the setting of ludics and L-nets, establishing in this way a direct connection between the two frameworks.

Section 5 is dedicated to the proof of the sequentialization theorem. We extend the sequentialization method used in [DGF06] to include additives; the relevance of this result is clear, if one consider that the problem of sequentializing in presence of additives is one of the most difficult in the framework of proof nets.

In section 6 we introduce and study the dynamics of cut reduction. In subsection 6.2 we define cut-elimination on J-proof structures, following [FM05] and [LTdF04]). In subsection 6.3 using *pointed sets*, we define a model of cut-reduction in J-proof nets, called *pointed semantics*, which extends usual relational semantics: differently from the standard relational model, pointed sets allow in fact to semantically characterize jumps. In subsection 6.3.2 we show that pointed semantics is a faithful description of J-proof nets, by proving that the model is *injective* with respect to J-proof nets: namely, two J-proof nets with the same interpretation in pointed semantics are syntactically equivalent.

Then in section 6.4, using injectivity of pointed semantics, we will prove that composition of J-proof nets is stable under cut reduction (a similar strategy was used also by Laurent and Tortora de Falco in [LTdF04]). The use of semantics to prove a syntactical property should not be surprising; in the spirit of the program launched in [Gir99] by Girard, syntax and semantics are just two different ways to describe the same object.

Finally, in section 7 we adress some general questions concerning J-proof nets; namely, the problem of canonicity (which was first posed in the setting of L-nets by Faggian and Curien in [CF]), the relation with monomial proof nets (see [Gir96], [HVG03]) and the link with *multifocalization* (see [CMS08]).

Acknowledgements

The author is indebted with Jean-Yves Girard, for having first suggested the idea of sequentializing with jumps. This paper relies on a huge amount of work and discussions with Claudia Faggian; I would like to thank her for all her support. Thanks to Pierre Boudes, Damiano Mazza and Lorenzo Tortora de Falco for many useful comments and insights.

Contents

1	Introduction	1
1.1	Outline of the paper	4
2	Background	6
2.1	Hypersequentialized calculus	6
2.1.1	From <i>MALL</i> to <i>HS</i>	6
2.1.2	<i>HS</i> syntax	7
2.2	Preliminaries on graphs	9
2.3	<i>HS</i> proofs as d.a.g.’s	10
3	J-proof nets	12
3.1	J-proof structures	12
3.1.1	Totality	14
3.1.2	Jumps and the additive structure	15
3.1.3	Axiom links	15
3.2	J-proof structures and sequent calculus	16
3.2.1	Sequentializable J-proof-structures	17
3.3	Correctness criterion	18
4	J-proof nets, L-nets and superposition	19
4.1	Adresses and ludification	19
4.2	Sharing equivalence	21
4.3	Superposition	21
5	Sequentializing with jumps	23
5.1	Arborescence and sequent calculus	24
5.2	Sequentializing with jumps: the multiplicative case	24
5.3	Sequentializing with jumps: the additive case	25

5.4	Sequentializing with jumps: the general case	27
5.5	Sequentialization	30
5.5.1	A saturated J-proof net is switching acyclic.	32
5.5.2	A saturated J-proof net is arborescent.	33
5.5.3	J-proof nets and Mix rule	35
6	Cut elimination and semantics	35
6.1	Cut and sequentialization	35
6.2	Cut elimination	36
6.2.1	Cut elimination on J-proof structures: the multiplicative case	36
6.2.2	Cut elimination on J-proof structures: the general case	38
6.2.3	Preservation of weakly correctness	41
6.3	Semantics	43
6.3.1	Experiments	44
6.3.2	Injectivity	45
6.4	Correctness criterion is stable under reduction	47
7	Are J-proof nets canonical objects?	48
7.1	Parallel J-proof nets	48
7.2	J-proof nets and monomiality	49
7.3	J-proof nets and multifocusing	50

2 Background

In this section we introduce the principal notions and concepts we will use throughout the paper. In subsection 2.1 after an informal introduction of synthetic formulas, we define the hypersequentialized calculus; then we introduce the reader to the notions of slices and superposition, absolutely necessary to understand additives. In subsection 2.2 we recall basic concepts of graph theory. Finally in subsection 2.3 we provide some indications on how to represent hypersequentialized proofs as graphs.

2.1 Hypersequentialized calculus

2.1.1 From *MALL* to *HS*

Let us consider the grammar of *MALL* without atoms, and additive constants:

$$F := 1 \mid \perp \mid F \wp F \mid F \otimes F \mid F \& F \mid F \oplus F$$

Starting from this grammar, we define the notion of *synthetic formula* as a sort of “canonical representative” of the formulas of *MALL* equivalent modulo the usual associative, distributive and neutral element properties of linear logic connectives.

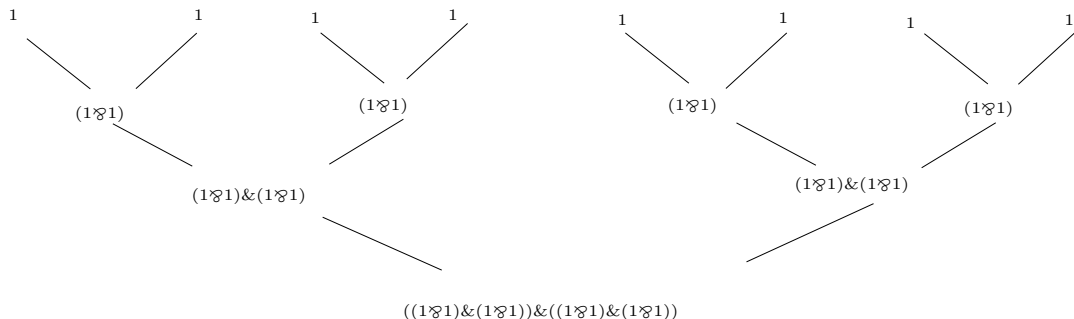
Let us give an informal sketch of how to build a synthetic formula . Let us consider the following formula:

$$A = (1\&1)\wp(1\&1)$$

First of all, we apply to A the distributivity property $(B\&C)\wp D \simeq (B\wp D)\&(C\wp D)$ using it as a rewriting rule oriented from left to right. We obtain as “normal form” the formula A' :

$$A' = ((1\wp 1)\&(1\wp 1))\&((1\wp 1)\&(1\wp 1))$$

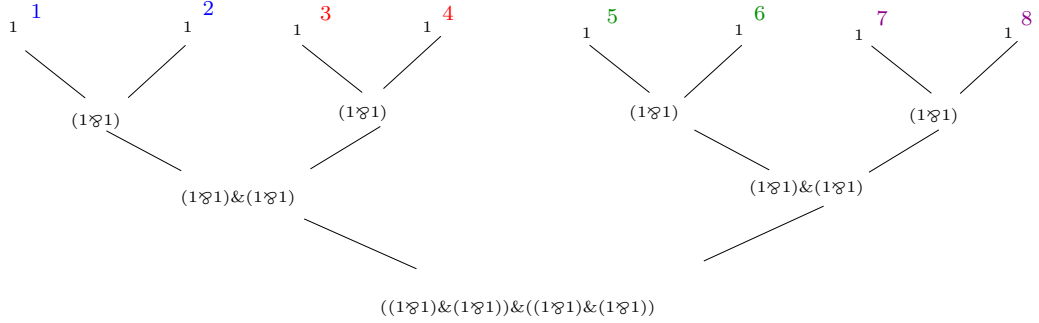
Now let us look at the subformula tree t of A' :



The tree t is composed bottom-up by two layers: a first layer of negative formulas, and then a set of positive formulas, which are the leaves of the tree. The negative layer is itself divided in two layers: first a set of formulas whose outermost connective is $\&$ (its “additive” sub-layer), and then a set of formulas whose outermost connective is \wp (its “multiplicative” sub-layer).

The negative layer of connectives in t can be represented by a single “big step” negative connective, as follows.

Let us assign to every positive leaf an integer index as a place marker, in the following way:



Let us consider the maximal subtrees of t having a \wp formula as root (we can call them \wp -trees); there are four of them in t . We can partition the leaves of t by grouping together the leaves belonging to the same \wp -tree.

We call a set I of indexes of leaves belonging to the same \wp -tree a *ramification* of t ; we call *directory* the set \mathcal{N} of all ramifications I of t . In our case, the directory of t contains the following ramifications: $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, $\{7, 8\}$.

Now, it is easy to verify that if we replace t with any other subformula tree t' equivalent to t modulo associativity and neutral elements, the directories of t and t' will be the same.

The information provided by the directory \mathcal{N} is enough to retrieve all the formulas whose subformula tree is equivalent to t modulo the usual isomorphisms of linear connectives. Now if we want to rewrite the formula retaining only the information provided by the directory \mathcal{N} we would obtain something like this:

$$\&(\wp(1_1, 1_2), \wp(1_3, 1_4), \wp(1_5, 1_6), \wp(1_7, 1_8))$$

We call such an object a *synthetic formula*; we stress that the possibility of defining such a formula depends strictly from the fact that rewriting a formula, using the associative, distributive and neutral element property of LL , preserves the polarity of the formula.

Due to the focalization property (see the introduction of the paper) then it is possible to define a linear logic calculus manipulating only synthetic formulas. We call such a calculus *hypersequentialized calculus*.

2.1.2 HS syntax

The grammar of hypersequentialized formulas is the following:

$$N ::= \&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$$

$$P ::= \oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(N_i))$$

where

- there are two kinds of formulas: N formulas are *negative* while P formulas are *positive*;
- I is a ramification (that is an index set) and \mathcal{N} is a non-empty¹ directory (that is, a set of ramifications).

Duality is defined as follows:

$$(\oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(A_i)))^\perp = \&_{I \in \mathcal{N}}(\wp_{i \in I}(A_i^\perp))$$

$$(\&_{I \in \mathcal{N}}(\wp_{i \in I}(A_i)))^\perp = \oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(A_i^\perp))$$

$$P^{\perp\perp} = P$$

¹For technical reasons we do not consider the additive constants \top and 0

Note We remark the following facts:

- By $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$ we indicate the synthetic formula which represents all possible combinations of the formulas P_i (where i is an element of the ramification I belonging to the directory \mathcal{N}) modulo the associativity and distributivity properties of usual \wp and $\&$ connectives in LL , and neutral element property of \wp connective of LL ; in case the directory \mathcal{N} is a singleton, we shall use the abbreviation $(\wp_{i \in I}(P_i))$; if for some $I \in \mathcal{N}$, I is a singleton or is empty, then we will denote the component $(\wp_{i \in I}(P_i))$ corresponding to I as $\uparrow P$ in the former case, and as \perp in the latter.
- By $\oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(N_i))$ we indicate the synthetic formula which represents all possible combinations of the formulas N_i (where i is an element of the ramification I belonging to the directory \mathcal{N}) modulo the associativity and distributivity of usual \otimes and \oplus connectives in LL , and neutral element property of \otimes connective of LL ; in case the directory \mathcal{N} is a singleton, we shall use the abbreviation $(\otimes_{i \in I}(N_i))$; if for some $I \in \mathcal{N}$, I is a singleton or is empty, then we will denote the component $(\otimes_{i \in I}(N_i))$ corresponding to I as $\downarrow N$ in the former case, and as 1 in the latter.

Note that \uparrow and \downarrow are “linear” versions of the exponentials $?$, $!$, acting as polarity inverter (see [Lau04],[Gir01]). Moreover, for technical reasons we must forbid the presence of more than one empty ramification I in a directory \mathcal{N} ; this will prevent us from representing, for example the formula $1 \oplus 1$.

The *HS* (+ Mix) calculus is the following:

$$\frac{\frac{\vdash \Gamma_1, N_1 \quad \dots \quad \vdash \Gamma_n, N_n}{\vdash \Gamma_1, \dots, \Gamma_n, \oplus_{J \in \mathcal{N}}(\otimes_{j \in J} N_j)} (+, I)}{\vdash \Gamma, \Delta} \text{mix} \quad \frac{\frac{\vdash \Gamma, P_1^1, \dots, P_1^{k_1} \quad \dots \quad \vdash \Gamma, P_n^1, \dots, P_n^{k_n}}{\vdash \Gamma, \&_{J \in \mathcal{N}}(\wp_{j \in J} P_j)} (-, \mathcal{N})}{\vdash \Gamma, P \quad \vdash \Delta, P^\perp} \text{cut}}{\vdash \Gamma, \Delta}$$

where:

- in the $(+, I)$ rule the cardinality of I is n ;
- in the $(-, \mathcal{N})$ rule the cardinality of \mathcal{N} is n , with $n \neq 0$, and the cardinality of J is k_i for $J \in \mathcal{N}$ and $i \in \{1, \dots, n\}$
- Γ, Δ, \dots only contain positive formulas.

Additives and superposition Before presenting J-proof nets, we want to give a first intuition about two fundamental notions which naturally come out whenever a linear logic calculus includes additives: the one of *slice* and the one of *superposition*.

Let us consider the following sequent proof π of $\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), \&(\wp(1_1, 1_2), \wp(1_3, 1_4))$:

$$\frac{\frac{\frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\}) \quad \frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), 1, 1} (+, \{1, 2\}) \quad \frac{\frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\}) \quad \frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), 1, 1} (+, \{3, 4\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), \&(\wp(1_1, 1_2), \wp(1_3, 1_4))} (-, \{\{1, 2\}, \{3, 4\}\})$$

Now, we choose a branch for each $(-, \mathcal{N})$ rule, (in this case just one); by erasing the right branch we get the following derivation s_1 :

$$\frac{\frac{\frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\}) \quad \frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), 1, 1} (+, \{1, 2\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), \&(\wp(1_1, 1_2), \wp(1_3, 1_4))} (-, \{\{1, 2\}, \{3, 4\}\})$$

and by erasing the left branch, we get the following derivation s_2 :

$$\frac{\frac{\frac{\overline{\vdash 1}}{\vdash 1, \perp} (+, \{\}) \quad \frac{\overline{\vdash 1}}{\vdash 1, \perp} (-, \{\})}{\vdash 1, \perp} (-, \{\}) \quad \frac{\overline{\vdash 1}}{\vdash 1, \perp} (+, \{3, 4\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), 1, 1} (+, \{3, 4\})}{\vdash \oplus(\otimes(\perp_1, \perp_2), \otimes(\perp_3, \perp_4)), \&(\wp(1_1, 1_2), \wp(1_3, 1_4))} (-, \{\{1, 2\}, \{3, 4\}\})$$

In both s_1 and s_2 the $(-, \mathcal{N})$ -rule is unary (the “additive” part is erased and only the “multiplicative” part is kept).

Following this observation, a $(-, \mathcal{N})$ -rule appears as a set (superposition) of unary rules having the same active formulas. If we consider a sequent calculus derivation in HS , and for each $(-, \mathcal{N})$ -rule we select one of the premises, we obtain a derivation where all $(-, \mathcal{N})$ -rules contain a single ramification; actually we “forget” the additive structure of the proof, keeping only the multiplicative one. This is called a *slice*. Hence, an HS proof containing some $(-, \mathcal{N})$ -rule can be thought of as a superposition of multiplicative proofs (that is, slices).

Actually, the notion of slice is as old as Linear Logic itself: it appears for the first time in the seminal paper [Gir87]; it has been used by Laurent and Tortora de Falco for studying normalization on polarized proof nets (see [LTdF04]) and is a key notion of ludics and L-nets.

The main point when one deals with additive proof nets is to properly reconstruct the structure of the multiplicative proofs an additive proof is composed of, and to correctly superpose them; such a task is usually fulfilled by boxes (as in [Gir87], [LTdF04], [TdF03b]) or by boolean weights (as in [Gir96], [Lau99], [Mai07]), which provide enough “synchronization points” to glue slices together. As we will see later, in additive J-proof nets it is possible to glue together slices due to the presence of *jumps*, acting as synchronization points.

2.2 Preliminaries on graphs

A *directed graph* G is an ordered pair (V, E) , where V is a finite set whose elements are called *nodes*, and E is a set of ordered pairs of nodes called *edges*.

We will denote nodes by small initial Latin letters a, b, c, \dots and edges by small final Latin letters \dots, x, y, z .

To denote that there is an edge from a node a to a node b , we will write $a \rightarrow b$; we say that an edge x from a to b is *emergent* from a and *incident* on b ; b is called the *target* of x and a is called the *source*.

The *in-degree* (resp. *out-degree*) of a node is the number of its incident (resp. emergent) edges; two edges are *coincident* when they have the same target.

Given a directed graph $G = (V, E)$ and a subset V' of V , the *restriction* of G to V' is the graph $G' = (V', E')$ where E' is the restriction of E to the elements of V' .

Given a directed graph G a *path* (resp. *directed path*) r from a node b to a node c is a sequence $\langle a_1, \dots, a_n \rangle$ of nodes such that $b = a_1$, $c = a_n$, and for each a_i, a_{i+1} , there is an edge x either from a_i to a_{i+1} , either from a_{i+1} to a_i (resp. from a_i to a_{i+1}); in this case, x is said to be *used* by r ; given a path $r = \langle a_1, \dots, a_n \rangle$ we will say that r *leaves* a_1 and *enters* in a_n . Moreover, we require all nodes in a path from a node b to a node c to be distinct, with the possible exception of b and c .

If there is a directed path from a to b , we denote it by $a \xrightarrow{+} b$.

A graph G is *connected* if for any pair of nodes a, b of G there exists a path from a to b .

A *cycle* (resp. *directed cycle*) is a path (resp. directed path) $\langle a_1, \dots, a_n \rangle$ such that $a_1 = a_n$.

A *directed acyclic graph* (d.a.g.) is a directed graph without directed cycles.

When drawing a d.a.g we will represent edges oriented up-down so that we may speak of moving downwardly or upwardly in the graph; in the same spirit we will say that a node is just above (resp. hereditarily above) or below (resp. hereditarily below) another node.

A *d.a.g. with pending edges* is a d.a.g. where some edges have a source but not a target.

We call *typed d.a.g.* a d.a.g. with pending edges whose edges are possibly labelled with formulas ; we call such edges *typed*.

We call *typed d.a.g. with ports* a typed d.a.g. G where for each node b the typed edges incident on b and the typed pending edges of G are partitioned into subsets called *ports*.

We call *module* a d.a.g, where some edges have a target but not a source.

We recall that we can represent a strict partial order as a d.a.g., where we have an edge $b \rightarrow a$ whenever $a <_1 b$ (i.e. $a < b$, and there is no c such that $a < c$ and $c < b$.) Conversely (the transitive closure of) a d.a.g. G induces a strict partial order \prec_G on the nodes of G .

We call *predecessor* of a node c , a node which immediately precedes (bottom-up) c in \prec_G (similarly for the successor).

A *root* of a d.a.g. is a node with no predecessors.

A *forest* (resp. *tree*) is a d.a.g. G such that, given a node a and a root b of G , there exists at most one (resp. exactly one) directed path from a to b .

Similarly, a strict order on a set is *arborescent* when each element has at most one immediate predecessor. An edge $a \rightarrow b$ is *transitive* if there is a node c such that $a \xrightarrow{+} c$ and $c \rightarrow b$.

The *skeleton* of a d.a.g. G (denoted $Sk(G)$) is the directed graph obtained from G by erasing all the edges which are transitive

Since $Sk(G)$ is obtained from G just by erasing transitive edges, the order associated with $Sk(G)$ as a d.a.g. and the order associated with G as a d.a.g. are equal; so if the order associated with G is arborescent, the skeleton of G is a forest.

2.3 HS proofs as d.a.g.'s

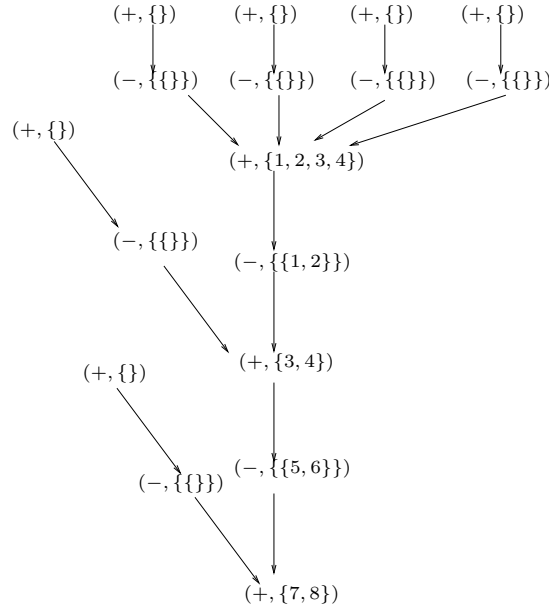
Let us consider the following HS proof π_1 :

$$\frac{\frac{\frac{\frac{\frac{\vdash \perp, 1 \quad \vdash \perp, 1 \quad \vdash \perp, 1 \quad \vdash \perp, 1}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), 1, 1, 1, 1} (+, \{1, 2, 3, 4\})}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), 1, 1, \wp(1_1, 1_2)} (-, \{\{1, 2\}\}) \quad \vdash 1, \perp}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), 1, 1, \otimes(\wp(1_1, 1_2)_3, \perp_4), 1} (+, \{3, 4\})}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), \wp(1_5, 1_6), \otimes(\wp(1_1, 1_2)_3, \perp_4), 1} (-, \{\{5, 6\}\}) \quad \vdash 1, \perp}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), \otimes(\wp(1_5, 1_6)_7, \perp_8), \otimes(\wp(1_1, 1_2)_3, \perp_4), 1, 1} (+, \{7, 8\})$$

where each initial sequent is the conclusion of the following proof:

$$\frac{\overline{\vdash 1} (+, \{\})}{\vdash 1, \perp} (-, \{\})$$

The order of application of rules provides an arborescent partial order, which yields the following d.a.g. G_1 :

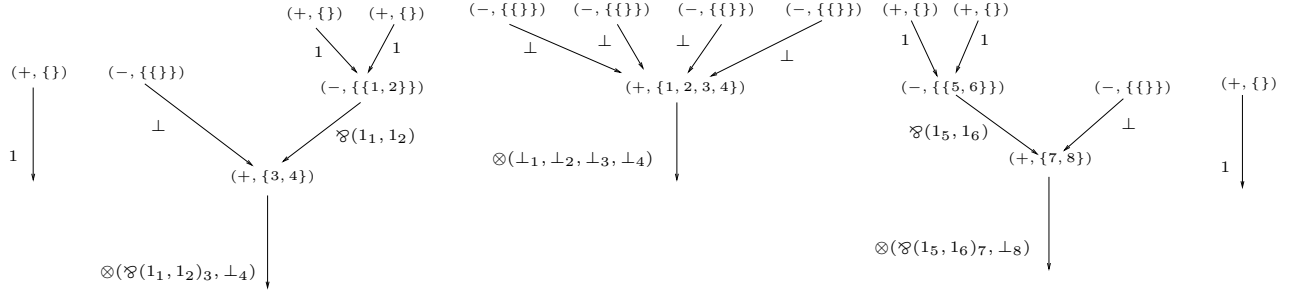


We can refine our analysis, and observe that the order associated with π_1 it is actually the merging of two orders:

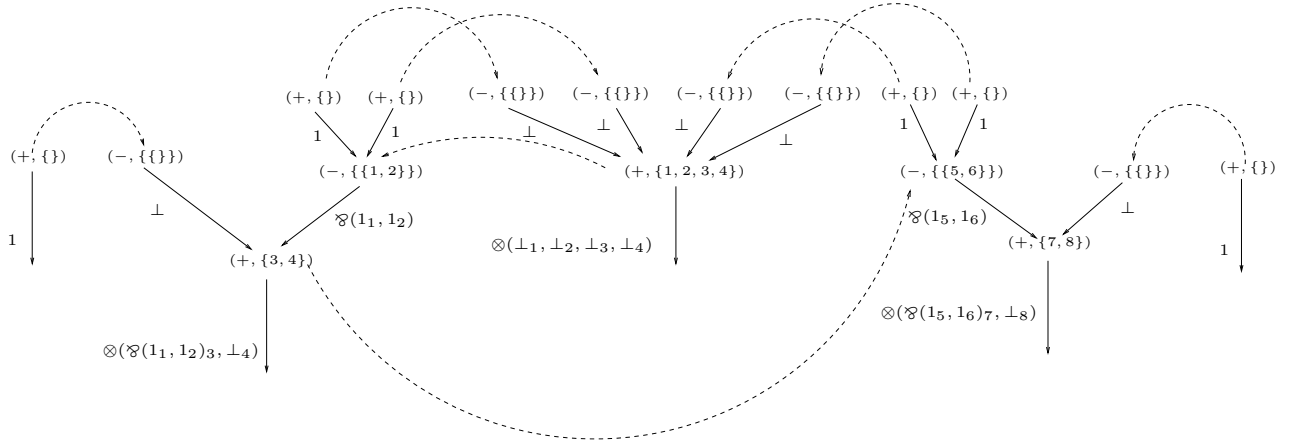
- the *causal* order, which is the transitive closure of the following relation: a rule a is causally dependent from a rule b when the formula introduced by b is an active formula in the premises of a ;
- the *sequential* order: a is sequentially dependent from b if a follows b (top-down) in the proof but it is not causally dependent from it.

For instance, the rule $(+, \{7, 8\})$ is causally dependent from the rule $(-, \{\{5, 6\}\})$, and is sequentially dependent for example from the rule $(+, \{3, 4\})$.

Let us draw the d.a.g. G'_1 associated with the causal order (we label the edges with types in order to stress the subformula relation)



If we add to G'_1 the information provided by the sequential order (in the form of dashed edges) we obtain the following d.a.g G''_1 :



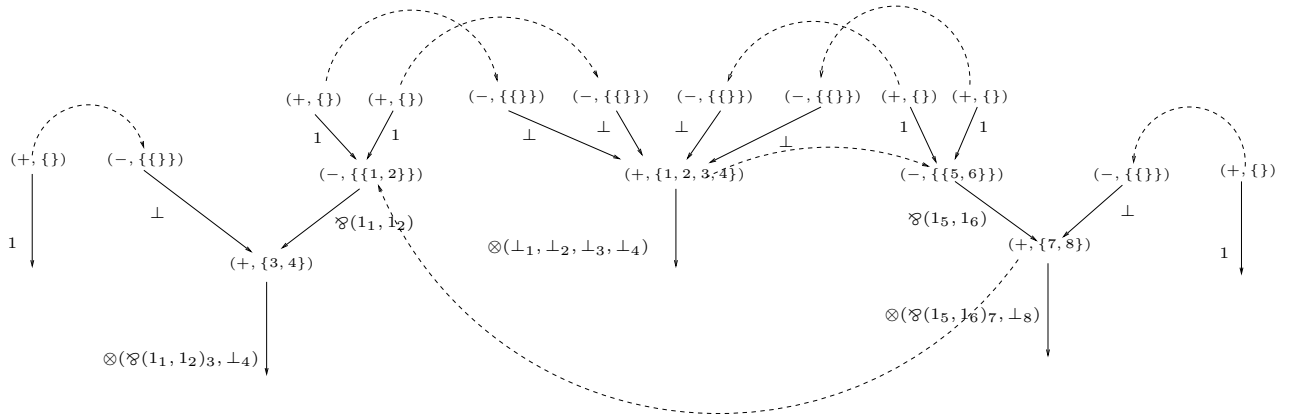
whose transitive closure bring us back to G_1 .

In G''_2 we can observe that there are two kind of edges: the ones corresponding to causal order, (which is the correspondent of the structure of links in proof nets) and the ones corresponding to sequential order. We call the latter *jumps*.

Now consider this other *HS* proof π_2 (differing from the first one only by permutation of rules):

$$\frac{\frac{\frac{\frac{\frac{\frac{\vdash \perp, 1 \quad \vdash \perp, 1 \quad \vdash \perp, 1 \quad \vdash \perp, 1}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4)}, 1, 1, 1, 1}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4)}, 1, 1, \wp(15, 16)}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4)}, 1, 1, \otimes(\wp(15, 16)_7, \perp_8), 1}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), \wp(11, 12), \otimes(\wp(15, 16)_7, \perp_8), 1}{\vdash \otimes(\perp_1, \perp_2, \perp_3, \perp_4), \otimes(\wp(15, 16)_7, \perp_8), \otimes(\wp(11, 12)_3, \perp_4), 1, 1}{\vdash \perp, \perp} (+, \{7, 8\})}{\vdash \perp, \perp} (-, \{5, 6\})}{\vdash \perp, \perp} (+, \{3, 4\})$$

If we perform on π_2 the same analysis we did on π_1 , we get the following graph G''_2 , where the causal order is the same as G''_1 but the sequential order is different.



Starting from this observations, in the following section we will define a notion of proof net with jumps, whose links correspond to HS rules. The question we will try to answer will be the following:

If we “relax” the sequential order in both G''_1, G''_2 can we find another d.a.g. G''_3 such that:

- Both G''_1, G''_2 can be retrieved from G''_3 ;
- G''_3 is “logically” sound (as proof nets are sound with respect to linear sequent calculus)?

We stress that the examples above do not contain any “additive” feature; in order to represent proofs respecting the constraints induced by additives (namely the “contraction” of contexts implicit in the $\&$ -rule of LL) we will make use of d.a.g.’s with ports instead of plain d.a.g.’s.

3 J-proof nets

In this section we present J-proof nets for HS , using a syntax which is directly inspired by L-nets.

Firstly, in subsection 3.1, we give a first definition of *pre-proof structure* as a typed d.a.g. with ports; we introduce this first notion in order to represent the “additive contraction” effect induced by the $(-, \mathcal{N})$ -rule. While in others syntaxes (as [Lau99]) this is done by introducing an explicit “additive contraction” link, here we adopt the convention of incorporating contraction into links, by partitioning edges of the same type incident on the same link into *ports*. Then we refine the definition of pre-proof structure into the one of J-proof structure, which will allow us to define in our setting the notion of *slice*, a fundamental tool to deal with additives. In subsection 3.2 we study the relation between J-proof structures and HS , by defining *sequentializable* J-proof structures; then, in subsection 3.3, we introduce the correctness criterion and we define J-proof nets.

3.1 J-proof structures

Definition 1 (Pre-proof structure) *A pre-proof structure is a typed d.a.g. with ports whose nodes (also called **links**) are labelled by one of the symbols $\text{cut}, +I \in \mathcal{N}, -I \in \mathcal{N}$*

*The label of a link imposes some constraints on the number and the types of its incident and emergent edges: we will call **premises** (resp. **conclusions**) the typed edges incident on (resp. emergent from) a link; an untyped edge is called a **jump**.*

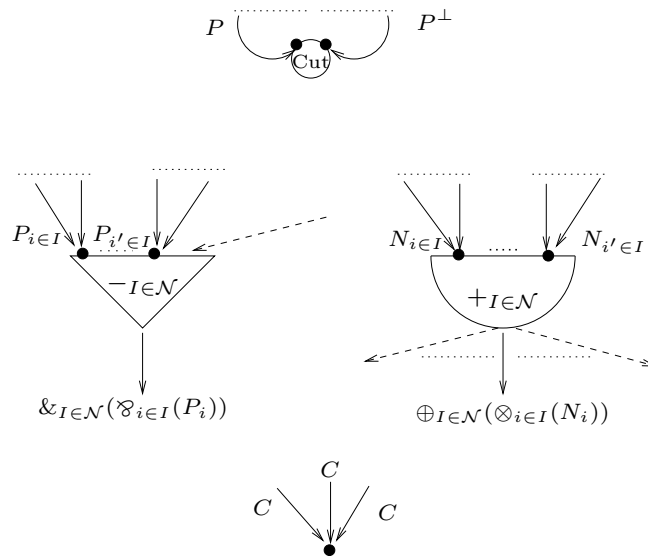


Figure 1: Links of a pre-proof structure

- a *cut-link* has no conclusions, and two ports, one containing $n \geq 1$ premises all typed by a formula A and the other containing $k \geq 1$ premises all typed by A^\perp ;
- a $-I \in \mathcal{N}$ link b (also called **negative link**) has:
 - one port for each $i \in I$; each port contains $n \geq 1$ premises which are typed by the same formula P ;

- one conclusion, typed by a formula N .
- $n \geq 0$ jumps incident on b .

If the premises of the i -th port (for $i \in I$) are typed by a formula P_i , then the conclusion is typed by $\&_{j \in \mathcal{N}}(\wp_{j \in J}(P_j))$, where $I \in \mathcal{N}$.

- a $+_{I \in \mathcal{N}}$ link b (also called **positive link**) has:

- one port for each $i \in I$; each port contains $n \geq 1$ premises which are typed by the same formula N ;
- one conclusion, typed by a formula P .
- $n \geq 0$ jumps emergent from b .

If the premises in the i -th port (for $i \in I$) are typed by a formula N_i , then the conclusion is typed by $\oplus_{j \in \mathcal{N}}(\otimes_{j \in J}(N_j))$, where $I \in \mathcal{N}$.

A link whose conclusion is a pending edge is called a **terminal link**; pending edges are also partitioned into ports called **terminal ports** in such a way that, if two pending edges belong to the same terminal port, they have the same type. Informally, we will call type of a port the type of the edges contained by the port. The **conclusions** of a pre-proof structure are the types of its terminal ports.

The constraints and the links of definition 1 are synthetically represented in fig. 1; we denote ports by black spots, and we distinguish positive and negative links by their shape; following this graphical convention, we will often label a positive (resp. negative) link simply with $I \in \mathcal{N}$ instead of $+_{I \in \mathcal{N}}$ (resp. $-_{I \in \mathcal{N}}$).

J-proof structures and slices To properly take into account the structure of additives, we must now refine our definition of proof structure, as follows:

Definition 2 (Contracted links and $\&$ -rule) Given a pre-proof structure, two links a, b are **contracted** if the conclusion of a and the conclusion of b belong to the same port.

A **$\&$ rule** W is a maximal set $\{w_1, \dots, w_n\}$ of negative contracted links. We will consider $\&$ rules as a sort of “macro“ links, so we adopt for them the same terminology we use for links; we will often refer to the premises (resp. conclusion) of a $\&$ -rule W , meaning the premises (resp. conclusion) of the elements of W .

An **additive pair** is a pair of negative links belonging to the same $\&$ rule.

A $\&$ rule containing terminal links is called a **terminal $\&$ rule**.

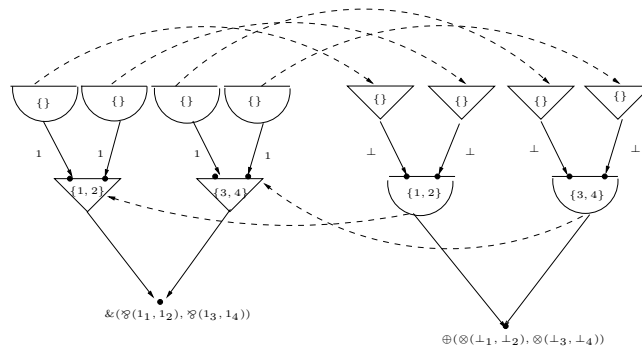


Figure 2: example of J-proof structure

Definition 3 (J-proof structure) A pre-proof structure R is a **J-proof structure** if it satisfies the followings:

Positivity: If a link a has in-degree 0, then it is positive;

Additives: if two links a, b of R belong to the same $\&$ rule, then the label of a is $-_{J \in \mathcal{N}}$ and the label of b is $-_{J' \in \mathcal{N}}$ and $J \neq J'$;

Coherence: given a link a and two negative links b, c , if $a \xrightarrow{+} b$ and $a \xrightarrow{+} c$ then b and c do not belong to the same $\&$ rule;

Contraction: given two positive contracted links a, b , there exists an additive pair w_1, w_2 s.t. $a \xrightarrow{+} w_1$ and $b \xrightarrow{+} w_2$.

In addition to the above conditions, a J-proof structure must have at most one negative conclusion.

Remark 1 Since every J-proof structure R is a d.a.g, we can associate with it a partial order \prec_R . From now on we will always identify two J-proof structures which are equal modulo transitive jumps, where a transitive jump is a transitive edge which is a jump.

Let us give some explanations on the conditions of definition 3:

- Positivity forbids the presence of negative 0-ary links in a J-proof structure. Such a constraint is required to represent the fact that in the calculus HS defined above there can be only positive initial rules.
- Given the synthetic connective $\&_{I \in \mathcal{N}}(\wp_{i \in I}(A_i))$, each of the ramifications $I \in \mathcal{N}$ corresponds to a branch of a (clustered) $\&$. In J-proof structures we represent the sequent calculus rule introducing a $\&$ formula not with a single link, but with a set of links (the $\&$ -rule defined above), each of them corresponding to a branch of a $\&$. The Additives condition then allows to recognize the different branches of which a $\&$ -rule is composed, by assigning to two different components of a $\&$ -rule a different ramification $I \in \mathcal{N}$.
- The Coherence condition is required in order to define a proper notion of *slice* on J-proof structures. It assures that no conflict rises when determining if a link belongs or not to a given slice: namely, if a link is hereditarily above a component of a $\&$ -rule, then it cannot be above any other component of the same $\&$ -rule.
- The Contraction condition allows us to recognize whenever two different slices are superposed; by Contraction, we know that each time two links are contracted, we can determine the $\&$ -rule which justifies the superposition. Combining this condition with the previous one, we can then identify slices simply by looking at the links which are hereditarily above a given element of a $\&$ rule W , for all $\&$ rules.

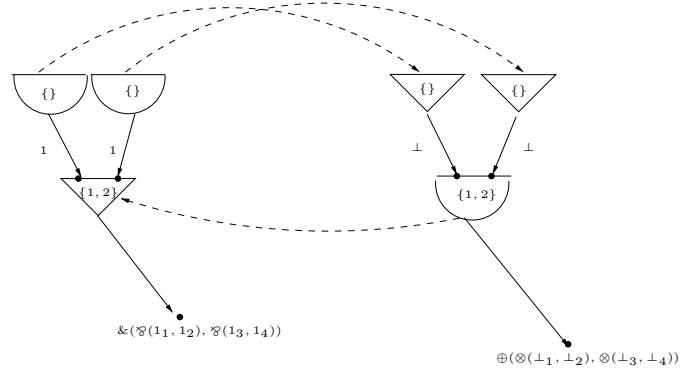


Figure 3: a slice of the J-proof structure in fig. 2

Now we are in the position to formally define the notion of *slice* of a J-proof structure:

Definition 4 A *slice* of a J-proof structure R is a graph S obtained by selecting, for each $\&$ rule $W = \{w_1, \dots, w_n\}$ of R , an element w_i of W , erasing all the $w_j \neq w_i \in W$, and recursively all the links above them, and all the edges emergent from an erased link.

3.1.1 Totality

Once introduced the notion of slice, a natural restriction is to consider only “well defined” J-proof structures: that is, J-proof structures whose slices are themselves J-proof structures. We do that introducing the notion of *totality*.

Definition 5 A *multiplicative J-proof structure* is a J-proof structure without contracted links.

Definition 6 Given two J-proof structures R, R' we say that R, R' **have the same conclusions** whenever there exists a bijection between the terminal ports of R and the terminal ports of R' s.t. if b is terminal port of R and b' is the corresponding terminal port of R' , then b, b' have the same type (and vice versa).

Definition 7 (Total J-proof structure) A J-proof structure R is **total** if

- for each negative-link a of R , with conclusion $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$, if \mathcal{N} contains more than one ramification, then a belongs to a $\&$ -rule W and there is a bijective correspondence between the elements of \mathcal{N} and the elements of W .
- for any slice S of R , S is a multiplicative J-proof structure with the same conclusions as R .

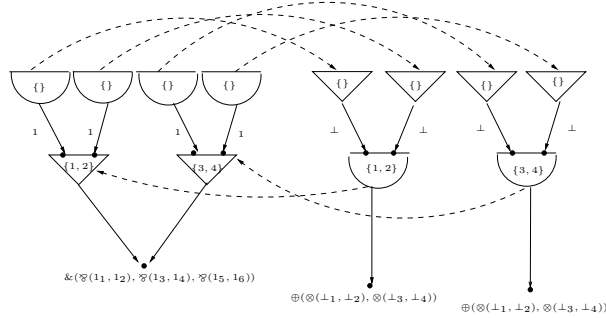


Figure 4: example of J-proof structure which is not total

In figure 4 we show a J-proof structure R which is not total: there is not a bijective correspondence between the elements of the leftmost negative rule W and the elements of the directory $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ of the conclusion of W ; moreover the two slices of R have not the same conclusions of R . The J-proof structure in figure 2 instead is total.

3.1.2 Jumps and the additive structure

One important difference with respect to the definition of J-proof structure already given in [DGF06] for the multiplicative fragment, is the role of jumps; in the additive case they not only graduate sequentiality, but allow also to keep track of the additive structure through the notion of *dependency*, which we define in the following.

Given a total J-proof structure R , a $\&$ rule W and a slice S of R , we denote by S^W any slice of R obtained by choosing the same element as S for all $\&$ rule $Z \neq W$ of R , while making a different choice only for W .

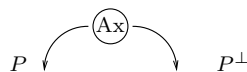
Definition 8 Given a $\&$ -rule W of a total J-proof structure R , we say that a link c depends from W iff for every slice S which contains a , all the slices S^W does not contain a .

Proposition 1 Let R be a total J-proof structure, a be a link of R and W a $\&$ rule. a depends from W iff for some $w \in W$, $a \xrightarrow{+} w$ in R .

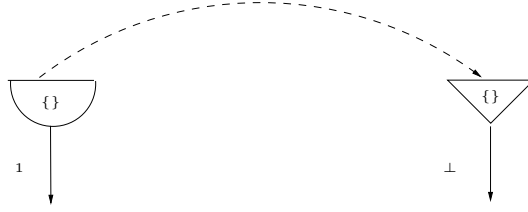
PROOF. The right to left direction is trivial. From left to right, let us suppose by absurd that a depends from W but for no element w of W , $a \xrightarrow{+} w$ in R . But then for every slice S of R , by definition of slice, if a belongs to S then a belongs to any slice S^W too, contradicting the hypothesis that a depends from W . \square

3.1.3 Axiom links

The reader confident with linear logic proof nets has certainly noticed that in the list of links defining a J-proof structure we did not include the axiom link, corresponding to the axiom rule of sequent calculus. An axiom link in proof nets has the following shape, with no premises and two conclusion typed by dual atomic formulas:



This choice was made to simplify our presentation of J-proof structure; nevertheless, we can simulate axiom links in our setting. If a is an axiom link, then we can replace it with a 0-ary positive link and a 0-ary negative link connected together through a jump as follows:



Modulo this replacement then, all the results which follow still hold in presence of axiom links.

3.2 J-proof structures and sequent calculus

Given an *HS* proof π and a J-proof structure R , we say that R can be associated with π , if R can be inductively decomposed in such a way that each step of decomposition of R corresponds to the writing down of a rule of π . If a J-proof structure R can be associated with a proof π of *HS*, we say that π is a *sequentialization* of R .

Decomposing a J-proof structure Given a J-proof structure R and a terminal link b of R by *removing b from R* we mean the following operation:

- erasing b (and all edges emergent from b) from R . We stress that, after erasing b , the edges incident on b become pending edges.
- partitioning the new typed pending edges into new terminal ports, in such a way that two new typed pending edges belong to two different terminal ports if and only if they were contained in two different ports of b . Any non typed pending edge is erased.

Definition 9 (Scope) Let R be a J-proof structure and W a terminal & rule of R : we call **scope** of an element w of W (denoted R_w) the graph obtained from R by erasing all links w' such that $w' \in W$ and $w' \neq w$, and all links and edges hereditarily above some w' .

Proposition 2 Let R be a total J-proof structure with a terminal & rule W and let w be an element of W . The graph R'_w obtained by taking R_w (the scope of w in R) and removing w is a total J-proof structure.

PROOF. First we have to prove that R'_w is a J-proof structure: to prove that is enough to prove that R_w is a J-proof structure: we must check then the four conditions of definition 3. The preservation of the coherence condition is a direct consequence of the fact that R_w is obtained by restricting R , so if a link a is above two different elements of the same & rule in R_w then it is the case also in R , which is impossible by coherence condition on R . For the same reason, additive condition on R implies additive condition on R_w . Concerning contraction condition, we reason by absurd and suppose that there are two positive contracted links a, b of R_w that do not satisfy the contraction condition; since R_w is a restriction of R , a, b must be contracted also in R ; so by contraction condition on R there is an additive pair y_1, y_2 such that $a \xrightarrow{+} y_1$ (resp. $b \xrightarrow{+} y_2$) in R . Since we supposed that a, b do not satisfy the contraction condition on R_w , the path $a \xrightarrow{+} y_1$ or the path $b \xrightarrow{+} y_2$ do not belong to R_w . W.l.o.g., suppose that the path $a \xrightarrow{+} y_1$ does not belong to R_w ; then at least one node c in the path does not belong to R_w . But then we obtain a contradiction, since, by definition of scope, in R $c \xrightarrow{+} w'$ for some element $w' \neq w$ of W , and then $a \xrightarrow{+} w'$; but then a cannot belong to R_w , contradicting our initial hypothesis on a, b . Concerning positivity condition, it is easy to verify that by definition of scope, a link a of R belongs to R_w iff a belongs to a slice S of R obtained by choosing w for the & rule W ; then by totality condition on R , S is a multiplicative J-proof structure, so it satisfies the positivity condition; but then also R_w satisfies it. So R_w is a J-proof structure, and then R'_w is trivially a J-proof structure too. The fact that R'_w is total is a direct consequence of the fact that 1) any slice of R_w is a slice of R so by totality on R is a multiplicative J-proof structure; 2) any slice of R obtained by choosing the element w of W is a slice of R_w , by definition of scope. It is then easy to verify that, once removed w , R'_w is total. \square

Definition 10 (Splitting positive) Let R be a J-proof structure and c a positive terminal link with n ports (resp. a cut link with two ports); We say that c is **splitting** for R if removing c , R splits into R_1, \dots, R_n (resp. R_1, R_2) J-proof structures, one for each port of c .

3.2.1 Sequentializable J-proof-structures

The following definitions are adapted from [Lau99].

Definition 11 We define the relation “ L sequentializes R in ε ”, where R is a J -proof structure, L is a terminal link or a terminal $\&$ -rule of R and ε is a set of J -proof structures, in the following way, depending from L :

- If L is a positive link with no ports, and is the only link of R , then L sequentializes R into \emptyset ;
- if L is a positive link or a cut link with n ports and is splitting, then L sequentializes R into $\{R_1, \dots, R_n\}$ J -proof structures;
- if L is a terminal $\&$ rule $W = \{w_1, \dots, w_n\}$ with conclusion $\&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$ such that there is a bijection between the elements of \mathcal{N} and the elements of W , we consider for each w_j the scope R_{w_j} of w_j . If for all $j \in \{1, \dots, n\}$, R_{w_j} is a J -proof structure with the same conclusions of R , and the graph R'_{w_j} obtained by removing w_j from R_{w_j} is a J -proof structure too, then L sequentializes R into $\{R'_{w_1}, \dots, R'_{w_n}\}$.
- if L is a terminal negative link which does not belong to a $\&$ -rule, and removing it we get a J -proof structure R_0 , then L sequentializes R into R_0 .

Definition 12 (Sequentializable J-proof structure) A J -proof structure R is **sequentializable** if

- R has a terminal $\&$ rule (resp. a terminal negative link), which sequentializes R into a set of sequentializable J -proof structures (resp. a J -proof structure);
- R has no terminal $\&$ rule or terminal negative link and
 - R is composed by a single connected component, and at least one of its links sequentializes R into a set of sequentializable J -proof structures or into the empty set;
 - R is composed by more than one connected component and each component is a sequentializable J -proof structure.

Proposition 3 If a J -proof structure R is sequentializable, there exists at least one proof π of HS, such that π is the sequentialization of R .

PROOF.

By induction on the links of R :

1. $n = 1$: by positivity, the only node in R is a 0-ary positive link, whose sequentialization is the proof $\frac{}{\vdash 1}$.
2. $n > 1$: suppose R contains one terminal $\&$ -rule $W = \{w_1, \dots, w_n\}$ with conclusion $\&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$; then by definition of sequentializable J -proof structure, W sequentializes R into R_1, \dots, R_n J -proof structures with conclusions respectively $\Gamma, P_1^1, \dots, P_1^{k_1}, \dots, \Gamma, P_n^1, \dots, P_n^{k_n}$; each R_j by induction hypothesis has a sequentialization π_j with conclusion $\vdash \Gamma, P_1^j, \dots, P_j^{k_j}$. We obtain the sequentialization π of R by applying a $(-, \mathcal{N})$ rule with conclusion $\vdash \Gamma, \&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$ to all π_1, \dots, π_n .

If R has no terminal $\&$ -rule, but has a terminal negative link n of conclusion $\wp(P_1, \dots, P_n)$, then by definition of sequentializable J -proof structure n sequentializes R into a J -proof structure R_0 with conclusion Γ, P_1, \dots, P_n ; R_0 by induction hypothesis has a sequentialization π_0 with conclusion $\vdash \Gamma, P_1, \dots, P_n$. We obtain the sequentialization π of R by applying a $(-, \{\{J\}\})$ rule with conclusion $\vdash \Gamma, \wp(P_1, \dots, P_n)$ to π_0 .

Otherwise R has no negative conclusion; suppose R is composed by a single connected component; since it is sequentializable there exists at least one link L which sequentializes R . We choose one such link and we reason by cases:

- L is a cut link whose premises are typed by P, P^\perp ; then L sequentializes R into two J -proof structures R_1, R_2 with conclusions respectively Γ, P and Δ, P^\perp ; by induction hypothesis R_1 (resp. R_2) has a sequentialization π_1 with conclusion $\vdash \Gamma, P$ (resp. π_2 with conclusion $\vdash \Delta, P^\perp$). We obtain the sequentialization π of R by applying to π_1, π_2 a cut rule with conclusion $\vdash \Gamma, \Delta$;
- L is a positive link $+_{I \in \mathcal{N}}$ with conclusion $\oplus_{J \in \mathcal{N}}(\otimes_{j \in J}(N_j))$; we recall that each port of L corresponds to an $i \in I$. L sequentializes R into R_1, \dots, R_n J -proof structures with conclusions respectively $\Gamma_1, N_1, \dots, \Gamma_n, N_n$; by induction hypothesis there exist π_1, \dots, π_n proofs with conclusion $\vdash \Gamma_1, N_1, \dots, \vdash \Gamma_n, N_n$, which are sequentializations of R_1, \dots, R_n , respectively. We obtain the sequentialization π of R by applying a $(+, I)$ rule with conclusion $\vdash \Gamma_1, \dots, \Gamma_n, \oplus_{J \in \mathcal{N}}(\otimes_{j \in J}(N_j))$ to π_1, \dots, π_n .

Otherwise, R is composed by more than one connected component, and each of them is a sequentializable J-proof structure; we conclude by applying induction hypothesis on them, followed by a sequence of Mix rules.

□

3.3 Correctness criterion

A correctness criterion must allow to characterize in an intrinsic, purely geometrical way all sequentializable J-proof structures, that is the ones which can be sequentialized into HS proofs; the J-proof structures satisfying the criterion are called *J-proof nets*. The correctness criterion for J-proof structures in the additive case is composed by two conditions:

- a *quantitative* one, the **totality** condition, which assures that in a J-proof net there are enough slices to retrieve a sequent calculus proof. We have already discussed this condition in subsection 3.1.1.
- a *qualitative* one, called **cycles** condition, which is a reformulation in our setting of the homonymous condition introduced by Curien and Faggian on L-nets, see [CF05].

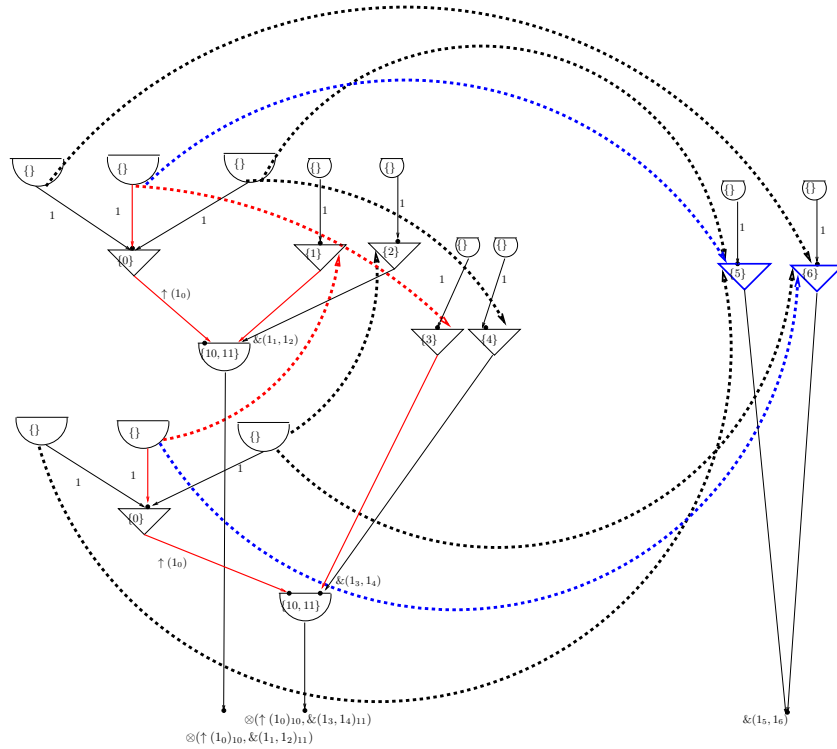


Figure 5: An example of a J-proof structure with a “legal” cycle (marked in red), justified by a negative rule (marked in blue).

In order to define the Cycles condition, we need to introduce the notion of *switching path* in a J-proof structure:

Definition 13 (Switching path and cycle) Given a J-proof structure R , and a $\&$ -rule W (resp. a negative link n) of R , we call **switching edge** of W (resp. n) any edge incident on an element of W (resp. on n). A **switching path** is a path which never uses two different switching edges of the same $\&$ -rule (resp. negative link) of R ; a **switching cycle** is a switching path which is a cycle.

Definition 14 (Cycles-correct J-proof-structure) A J-proof structure R is **cycles-correct** if, given a non empty union \mathcal{C} of switching cycles of R , there is a negative rule $W \in R$ not intersecting \mathcal{C} and a pair $w_1, w_2 \in W$ such that for some links $c_1, c_2 \in \mathcal{C}$, $c_1 \xrightarrow{+} w_1$ and $c_2 \xrightarrow{+} w_2$; in this case we say that the additive pair $w_1, w_2 \in W$ **breaks** \mathcal{C} .

Cycles condition deals with cycles which crosses *different* slices of the same J-proof structure; in fact, in the framework of additive proof nets, as first pointed out by Girard in [Gir96], the correctness of the single slices of a proof structure does not imply the sequentiability of the whole proof net (see for example the representation of the Gustave function as a proof structure given in [HVG03]).

In fig 6 we provide an example (due to Claudia Faggian) of a J-proof structure which is slice by slice switching acyclic, but it is not cycles-correct: there is a cycle crossing all the $\&$ -rules and all the terminal positive links.

As a consequence, we cannot choose any of the terminal positive links as the last rule of a sequent calculus proof, since none of them is splitting. In fig 5 instead we provide an example of a cycles-correct J-proof structure containing a “legal” cycle, which does not violate the Cycles condition. The example is based on an analogous one contained in [HVG03].

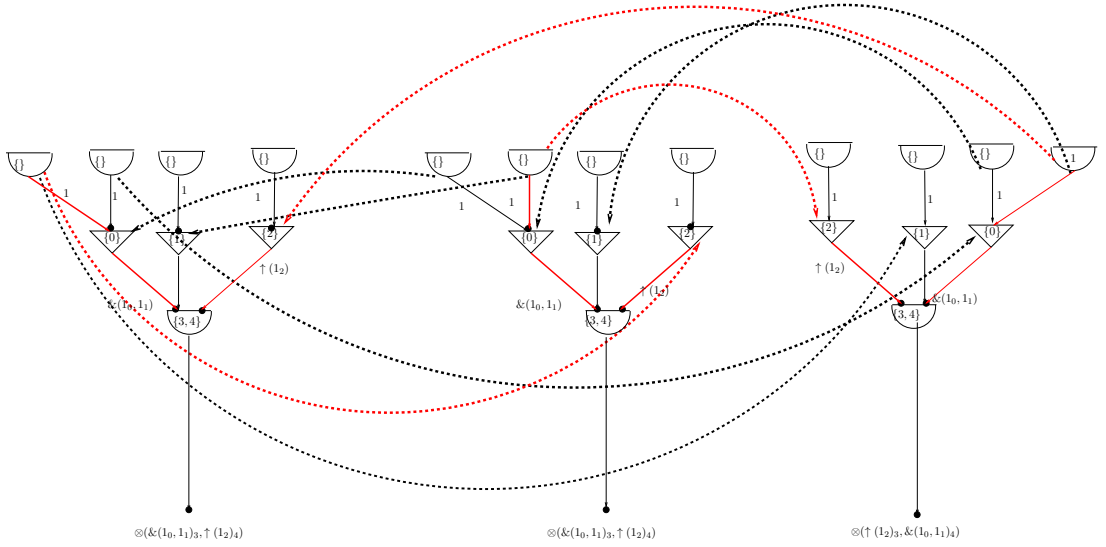


Figure 6: An example of a J-proof structure with an “illegal” cycle (marked in red)

Definition 15 (J-proof net) A *J-proof net* is a J-proof structure which is total and cycles-correct.

Theorem 1 (Sequentialization) Let R be a J-proof structure; R is a J-proof net iff it is sequentializable.

The right to left direction is (as usual) trivial. In section 5 we provide a proof of the left to right one, upgrading the technique already used in [DGF06] for the multiplicative fragment.

4 J-proof nets, L-nets and superposition

We already remarked that the “essence” of an additive proof is to be a *superposition* of proofs; in order to address the problem of sequentialization, we need then to give a precise definition of what it means to “superpose” two proofs (or in our case, two J-proof structures). We define the operation of superposition of J-proof structures, by adapting the notion of union of *chronicles* from ludics and L-nets (see [CF]) to our setting. We begin by recalling the basic ludic notion of *adress* and by defining its relation with J-proof structure.

Note. In this section, we will only consider J-proof structures without cut links.

4.1 Adresses and ludification

An *adress* is a sequence ξ of natural numbers. An adress σ is subadress of an adress ξ if ξ is a prefix of σ .

Let us take a non-empty multiset of formulas of HS . Let us call them A_1, \dots, A_n . A *localization* f of A_1, \dots, A_n is a function which associates with every node F of the subformula tree of A_j an adress, for every $j \in \{1, \dots, n\}$, in the following way:

- For every $j \in \{1, \dots, n\}$ $f(A_j) = \xi$, where ξ is an arbitrary adress;
- $f(A_j) \neq f(A_k)$, for $j \neq k \in \{1, \dots, n\}$;

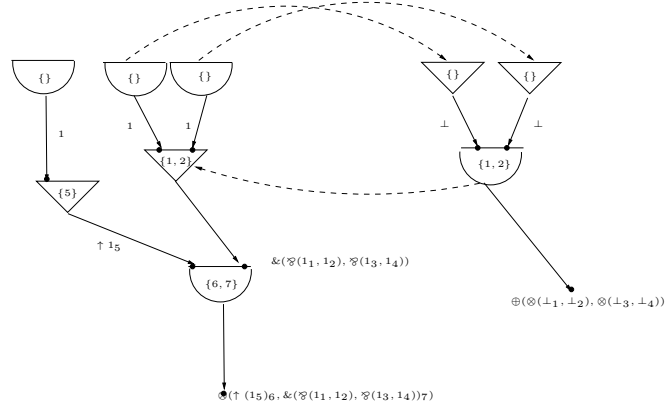


Figure 7: A J-proof structure R_1 .

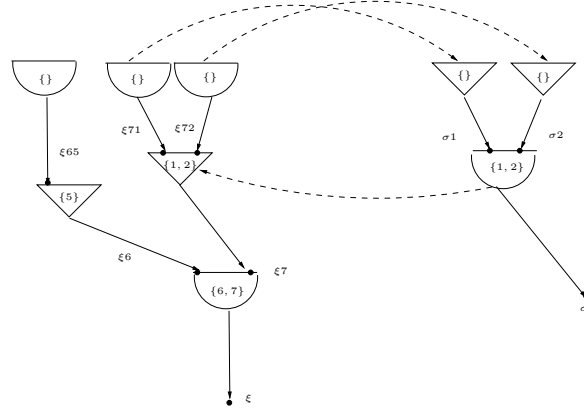


Figure 8: A J-proof structure R_1 labelled with actions

- If $f(F) = \sigma$, and $F = \oplus_{I \in \mathcal{N}} (\otimes_{i \in I} (N_i))$ (resp. $\&_{I \in \mathcal{N}} (\wp_{i \in I} (P_i))$) then $f(N_i)$ (resp. $f(P_i)$) = σ_i , for $i \in I \in \mathcal{N}$.

Informally, localization can be thought of as a way to unambiguously retrieve the position of a formula, given a forest of subformula trees.

Ludification An *action* is a pair $k = (\xi, I)$ where ξ is an address and I is a finite index set (that is, a ramification). An action $k = (\xi, I)$ is said to *use* the address ξ . We say that an action (ξ, I) *generates* the address ξ_i for all $i \in I$, and that an action a is *parent* of an action b if a generates the address used by b .

Given a J-proof structure R with conclusions A_1, \dots, A_n and a localization f of A_1, \dots, A_n , a *ludification* of R is a function l which associates with each link of R an action, in the following way:

- if a is a positive link labeled by $(+, I)$ (resp. a negative link labelled by $(-, I)$), with conclusion P (resp. N) and $f(P) = \xi$ (resp. $f(N) = \xi$) then $l(a) = (\xi, I)$.

With a slight abuse, we will extend the parent relation from actions to links, meaning that if a, b are links, a is parent of b iff $l(a)$ is a parent of $l(b)$; moreover, we will say that a and b are *similar*, if $l(a)$ and $l(b)$ use the same address.

We call *view* of a link a the set of actions $\{l(a)\} \cup \{l(a'); a \xrightarrow{+} a'\}$, equipped with the following order relation : $l(b) < l(c)$ iff $b \prec_R c$.

Remark 2 It is easy to observe that the ludification function l is completely determined by the localization function f on the conclusions of R ; meaning, up to the choice of f , the function l is unique.

In figure 8 , in order to clarify how actions can be associated with links, we show how, starting from the J-proof structure R_1 given in figure 7, formulas can be replaced with addresses.

J-proof structure and L-nets The reader acquainted with L-nets will observe that the object represented in figure 8 is actually an L-net. In fact, to transform a cut-free J-proof structure R with conclusion Γ, N (where N is the unique negative conclusion of R) into an L-net G with interface $\xi \vdash \Lambda$, it is enough to provide a localization of the conclusions of R which associate with N the address ξ , and with the formulas in Γ the addresses in Λ ; The set of views associated with the links of R straightforwardly induces an L-net with interface $\xi \vdash \Lambda$.

4.2 Sharing equivalence

Before explaining how to superpose two J-proof structures, first we must define how to compare them in order to decide the portion of J-proof structures to be superposed. We introduce then an equivalence relation on J-proof structures, that we call *sharing equivalence*.

Definition 16 Let R_1, \dots, R_n be J-proof structures with the same conclusions. Let f be a localization for the conclusions of R_1 ; since R_1, \dots, R_n have the same conclusions, f is a localization also for the conclusions of R_i for all i in $\{1, \dots, n\}$. Let l be a function from the links of R_1, \dots, R_n to actions, such that l restricted to R_i is the ludification of R_i induced by f , for all i in $\{1, \dots, n\}$. Given two links a, a' of R_1, \dots, R_n , we say that a is **sharing equivalent** to a' modulo l (denoted by $a \equiv_l a'$) iff the view of a induced by l is equal to the view of a' induced by l .

Proposition 4 Let R_1, \dots, R_n be J-proof structures with the same conclusions and a, a' be two links of R_1, \dots, R_n . Let f, f' be two localizations of the conclusions of R_1 (and hence of the conclusions of all R_i for $i \in \{1, \dots, n\}$), and l, l' the ludifications of R_1, \dots, R_n corresponding respectively to f, f' . Then $a \equiv_l a'$ if and only if $a \equiv_{l'} a'$.

PROOF.

Consider a single conclusion A of R_1, \dots, R_n and the two addresses ξ, σ associated with A respectively by f, f' . Now let us consider all the links a of R_1, \dots, R_n , such that $l(a) = (\lambda, I)$ and λ has ξ as prefix (so that the conclusion of a is subformula of A); if we substitute in λ ξ with σ , we get an address λ' such that $(\lambda', I) = l'(a)$. Now, if we generalize the procedure to all the conclusions of R_1, \dots, R_n , substituting in l the addresses associated with the conclusions of R_1, \dots, R_n by f with the ones associated by f' , we get that $l(a) = l'(a)$ for all links a ; and moreover, for all links a , the view of a induced by l is equal to the view of a induced by l' . We conclude that, given f and two links a, b of R_1, \dots, R_n such that a is sharing equivalent with b modulo l , if we replace f with f' in l' , then a is still sharing equivalent with b modulo l' . \square

Since by proposition 4, the relation of sharing equivalence induced by a ludification l does not depend from the specific ludification l chosen, we will speak more generally of sharing equivalence and denote it by \equiv (instead of \equiv_l).

Proposition 5 Given R_1, \dots, R_n J-proof structures with the same conclusions, the relation \equiv is an equivalence relation on the links of R_1, \dots, R_n ; moreover, there exists a unique sharing equivalence on R_1, \dots, R_n .

PROOF. Sharing equivalence is trivially an equivalence relation; concerning unicity is a consequence of the definition of ludification of a J-proof structure, of remark 2 and proposition 4. \square

Proposition 6 Let R_1, \dots, R_n be J-proof structures with the same conclusions. and a, a' be two links in R_1, \dots, R_n . If a is sharing equivalent to a' then the type of the conclusion of a and the type of the conclusion of a' are the same.

PROOF. Trivial from the fact that to be sharing equivalent for two links a, a' implies that $l(a) = l(a')$ for some ludification l ; that is $l(a)$ and $l(a')$ are two actions using the same addresses, meaning that the conclusions of a, a' are typed by the same formula, by definition of ludification of a J-proof structure. \square

In figure 10 we provide a representation of the J-proof structure R_2 in figure 9 where we have replaced formulas with addresses: comparing R_2 with the J-proof structure R_1 with the same conclusions in fig.7 and looking at their views, we can tell for example that the left terminal positive links of R_1 and R_2 are sharing equivalent, but the right terminal positive links of R_1, R_2 are not.

4.3 Superposition

Definition 17 Let R_1, \dots, R_n be J-proof structures with the same conclusions. Let $a \in R_j$ and $a' \in R_k$ be two similar links. Then a, a' are said to be *siblings*, whenever a is not sharing equivalent to a' , but either the parent b of a in R_j and the parent b' of a' in R_k are sharing equivalent or a, a' are both terminal links.

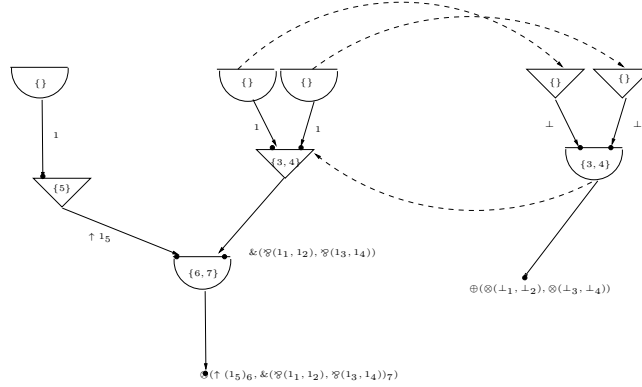


Figure 9: A J-proof structure R_2 .

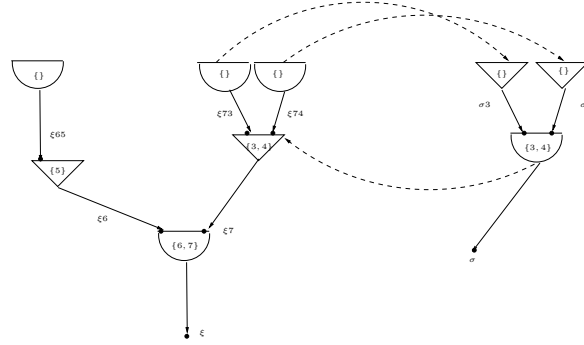


Figure 10: A J-proof structure labelled by actions R_2 .

Definition 18 (Compatibility) Let R_1, \dots, R_n be J-proof structures with the same conclusions. We say that R_1, \dots, R_n satisfy the **compatibility** condition whenever for any two sibling positive links $b \in R_j$ and $b' \in R_k$, there are two negative sibling links $a \in R_j$, $a' \in R_k$, such that $b \xrightarrow{+} a$ in R_j and $b' \xrightarrow{+} a'$ in R_k .

Definition 19 (Superposition) Let R_1, \dots, R_n be J-proof structures with the same conclusions. The **superposition** of R_1, \dots, R_n , is the J-proof structure R having the same conclusions of R_1, \dots, R_n , such that:

1. for all R_1, \dots, R_n and for every link a' of R_i , there exists a link a of R s.t. $a \equiv a'$;
2. $R = R'_1 \cup \dots \cup R'_n$, where, given an R_i for $i \in \{1, \dots, n\}$, R'_i is the restriction of R to the set $\{a \in R : \exists a' \in R_i \text{ s.t. } a \equiv a'\}$.

Given R_1, \dots, R_n J-proof structures with the same conclusions, we will denote their superposition by $\checkmark(R_1, \dots, R_n)$.

Proposition 7 Let R_1, \dots, R_n be J-proof structures with the same conclusions; there exists a unique J-proof structure $R = \checkmark(R_1, \dots, R_n)$ iff R_1, \dots, R_n satisfies the compatibility condition.

PROOF. First of all, it is easy to see that if a, b are links of R_1, \dots, R_n , the links a', b' of R s.t. $a \equiv a'$ and $b \equiv b'$ are contracted iff a, b are sibling; as a consequence, it is easy to see that the compatibility condition is nothing but a reformulation of the contraction condition in terms of sharing equivalence.

Once stressed this, the left to right direction is trivial. Concerning the right to left direction, we have to prove that R is a J-proof structure; that is we have to prove that the constraints given in the definition of J-proof structure are respected. Concerning positivity condition, let us consider a negative link a of R ; by point 2) of definition 19, there exists a link a' in some R_i s.t. $a \equiv a'$; by positivity condition on R_i there exists a positive link b' in R_i such that $b' \xrightarrow{+} a'$ in R_i , but then by point 1) of definition 19 and by definition of sharing equivalence there exists a positive link $b \in R$ such that $b \equiv b'$ and $b \xrightarrow{+} a$, so R respects the positivity condition.

Concerning coherence condition, let us consider a link a of R ; by point 2) of definition 19, there exist a link a' in some R_i s.t. $a \equiv a'$. Now it is easy to verify that, by definition 19, for all links b such that $a \xrightarrow{+} b$ in R ,

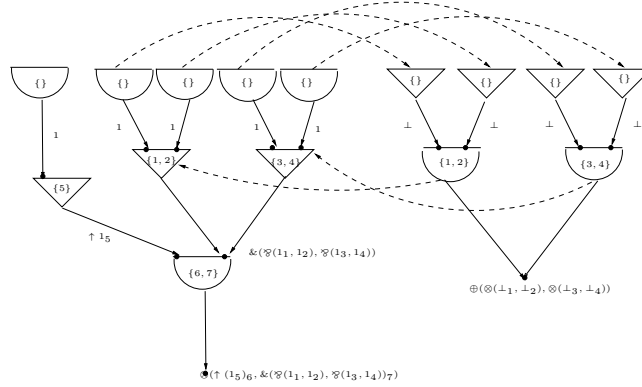


Figure 11: The superposition of R_1, R_2 .

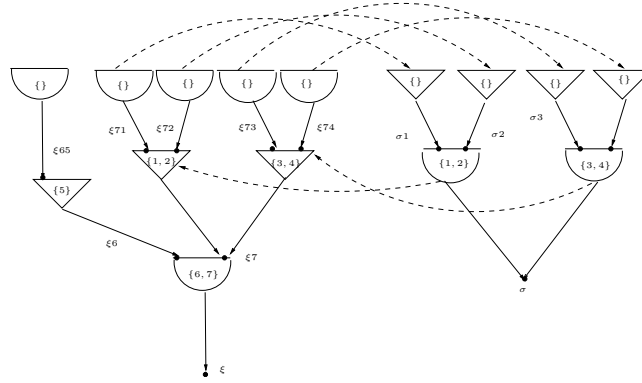


Figure 12: The superposition of R_1, R_2 , labelled by actions

there exists a link b' in R_i such that $a' \xrightarrow{+} b'$ in R_i and $b \equiv b'$ (otherwise a, a' would not be sharing equivalent); but then if a is above two different elements of the same $\&$ rule in R , so does a' in R_i , contradicting the hypothesis that R_i is a J-proof structure.

Last, we show that additives condition holds for R .

Suppose that w, w' are two negative sibling links of R_j, R_k respectively. We have two cases: either w, w' are both terminal links, or they aren't. In the first case, by definition of sharing equivalence, in order to be not sharing equivalent w, w' must have two different labels $J, K \in \mathcal{N}$; then w, w' satisfy the Additives condition on R . In the second case, since w, w' are siblings and not terminal, there exists in R_j a link x (resp. in R_k a link x') such that x is the parent of w in R_j (resp. x' is the parent of w' in R_k) and $x \equiv x'$. Since w, w' are negative links, and the views of their parents are the same, in order to be not sharing equivalent w, w' must have two different labels $J, K \in \mathcal{N}$; then w, w' satisfy the additives condition on R . Unicity of R is a consequence of the unicity of the sharing equivalence on R_1, \dots, R_n .

□

Proposition 8 *If S_1, \dots, S_n are the slices of a total J-proof structure R , then $\check{\cap} (S_1, \dots, S_n) = R$*

PROOF. Straightforward from the definitions.

□

In fig. 11 we provide the superposition R_3 of the J-proof structures R_1, R_2 in fig. 7, 9. The set of views associated with R_3 can be retrieved looking at fig 12. It can be easily verified that R_3 contains exactly the views of R_1, R_2 .

5 Sequentializing with jumps

In this section we will provide a proof of the sequentialization theorem by showing how to retrieve a sequentialization of a J-proof net by adding jumps on it. First in subsection 5.1 we will isolate a particular class of J-proof structures, called *arborescent*, which we prove to be in a one-to one correspondence with *HS* proofs.

In the rest of the section we will show how we can turn any J-proof net R into an arborescent J-proof net R^J by adding jumps. In subsection 5.2 we will first show a simple, purely multiplicative example recalling the technique already used in [DGF06]. Then in subsection 5.3, we will give an example of how to generalize the technique to include the additive case. In subsection 5.4 we formally define the operation of adding jumps on a J-proof structure and in subsection 5.5 we will finally prove the sequentialization theorem for J-proof nets.

Note. In this section, we will only consider J-proof nets without cut-links; we will speak about the question of sequentialization with cut-links in section 6.

5.1 Arborescence and sequent calculus

Definition 20 (Arborescent J-proof structure) A J-proof structure R is **arborescent** when the order \prec_R associated with R as a d.a.g. is arborescent.

Proposition 9 There is a one to one correspondance between cut-free arborescent total J-proof structures and cut-free proofs of HS, such that if a proof π is associated with an arborescent J-proof structure R , then π is the unique sequentialization of R .

PROOF.

The proof is by induction on the height n of an HS proof π :

$n = 1$: in this case, π is composed by just a 0-ary positive rule $\frac{}{\vdash 1}$ to which we associate a 0-ary positive link, whose unique sequentialization is trivially π .

$n = k + 1$: We have different cases depending on the last rule of π :

- Let the last rule of π be a $(-, \mathcal{N})$ rule L with conclusion $\vdash \Gamma, \&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$ and premises $\Gamma, P_1^1, \dots, P_1^{k_1}, \dots, \Gamma, P_n^1, \dots, P_n^{k_n}$. Let R_I be the arborescent J-proof structure associated by induction hypothesis with the proof π_I of the premise $\Gamma, P_I^1, \dots, P_I^{k_I}$ of L , corresponding to the ramification $I \in \mathcal{N}$; by induction hypothesis, π_I is the unique sequentialization of R_I . The arborescent J-proof structure R associated with π is the J-proof structure obtained in the following way:
 1. add a $-_{I \in \mathcal{N}}$ link b_I with conclusion $\&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$ to each R_I for $I \in \mathcal{N}$, in order to get an arborescent J-proof structure R'_I with conclusions $\Gamma, \&_{J \in \mathcal{N}}(\wp_{j \in J}(P_j))$ for each $I \in \mathcal{N}$;
 2. for every R'_I and for every positive terminal link a of R'_I , add a jump in R'_I from a to b_I , obtaining an arborescent J-proof-structure R''_I for each $I \in \mathcal{N}$;
 3. $R = \bigcup_{J \in \mathcal{N}} R''_J$.

It is easy to verify that R is a total arborescent J-proof structure whose unique sequentialization is π .

- Let the last rule of π be a $(+, I)$ rule with conclusion $\vdash \Gamma_1, \dots, \Gamma_n, \oplus_{J \in \mathcal{N}}(\otimes_{j \in J}(N_j))$, and premises respectively $\vdash \Gamma_1, N_1, \dots, \vdash \Gamma_n, N_n$. Let R_i be the arborescent J-proof structure associated with the proof π_i of the premise $\vdash \Gamma_i, N_i$ of L for $i \in \{1, \dots, n\}$; by induction hypothesis, π_i is the unique sequentialization of R_i for $i \in \{1, \dots, n\}$. Then the arborescent J-proof structure associated with π is the J-proof structure R obtained by connecting all R_i together with a $+_{I \in \mathcal{N}}$ -link with conclusion $\oplus_{J \in \mathcal{N}}(\otimes_{j \in J}(N_j))$ for $i \in \{1, \dots, n\}$.
- Let the last rule of π be a Mix rule with conclusions $\vdash \Gamma, \Delta$ and premises respectively $\vdash \Gamma, \vdash \Delta$; let R_1, R_2 be the arborescent J-proof structures associated by induction hypothesis respectively with the proof π_1, π_2 of the premises of L ; always by induction hypothesis, π_1 (resp. π_2) is the unique sequentialization of R_1 (resp. R_2). Then the arborescent J-proof structure associated with π is the union of R_1, R_2 .

□

5.2 Sequentializing with jumps: the multiplicative case

Let us consider the multiplicative J-proof net R given in fig. 13.

Is easy to verify that R is sequentializable and that the proofs π_1, π_2 of section 2.3, are two sequentializations of R .

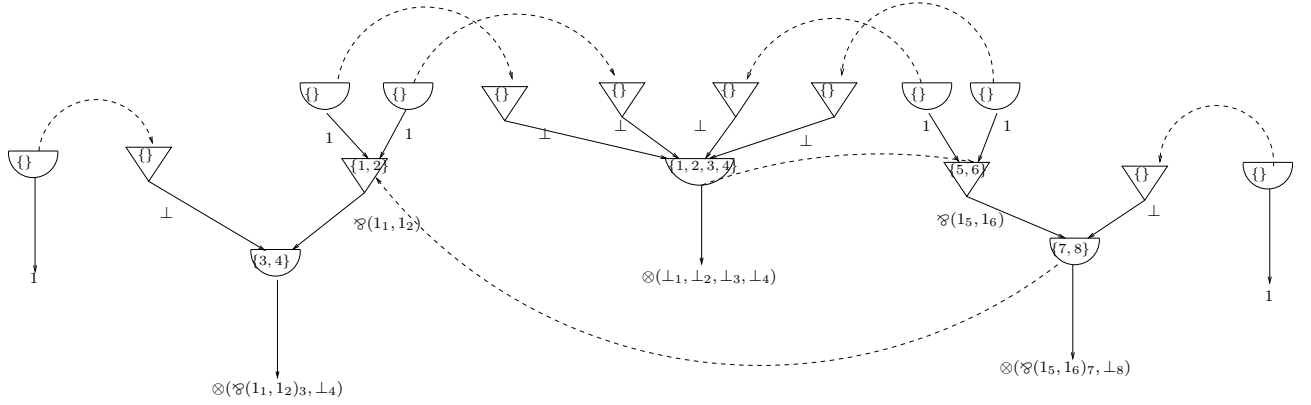


Figure 15:

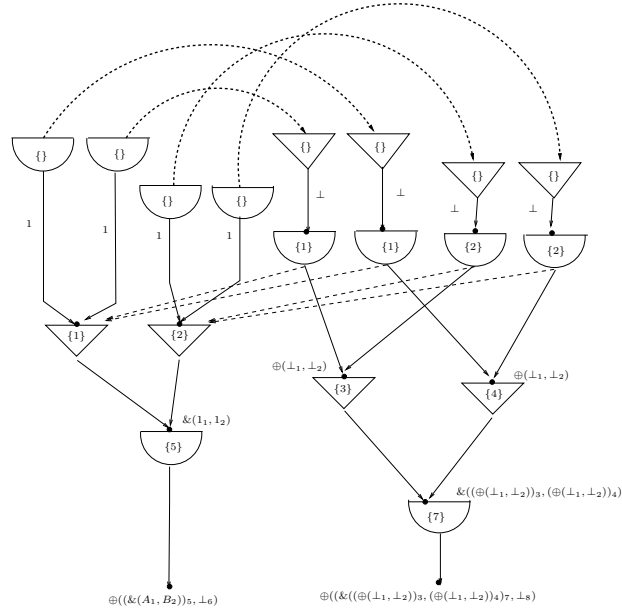


Figure 16: The J-proof net R' .

$$\frac{\frac{\frac{\frac{\overline{\vdash 1}}{\vdash 1, \perp} (+, \{1\})}{\vdash 1, \oplus(\perp_1, \perp_2)} (+, \{1\})}{\vdash \&((\oplus(\perp_1, \perp_2))_3, (\oplus(\perp_1, \perp_2))_4), 1} (-, \{\{3\}, \{4\}\})}{\vdash \oplus((\&((\oplus(\perp_1, \perp_2))_3, (\oplus(\perp_1, \perp_2))_4))_7, \perp_8), 1} (+, \{7\})}{\vdash \&(1_1, 1_2), \oplus((\&((\oplus(\perp_1, \perp_2))_3, (\oplus(\perp_1, \perp_2))_4))_7, \perp_8)} (+, \{5\})}$$

The arborescent J-proof structures R_1, R_2 , which directly correspond respectively to π_1, π_2 , are depicted in fig. 17, 18. The fact that R_1, R_2 directly correspond to π_1, π_2 is evident if we look at their skeletons in fig. 19, 20 ; the order between links matches perfectly the order between rules.

Now, we must define a procedure to transform R in, for example, R_1 using jumps. A first, naive approach would be to add some jumps in R from the leftmost terminal positive link a to the elements of the rightmost $\&$ -rule W . Unfortunately, this method is wrong: it can be easily verified that adding such jumps on R will violate the coherence condition in the definition of J-proof structure. Moreover we have to deal with the fact that some links of R must be duplicated in order to get a J-proof structure corresponding to the proof π_1 : the naive approach does not take that into account.

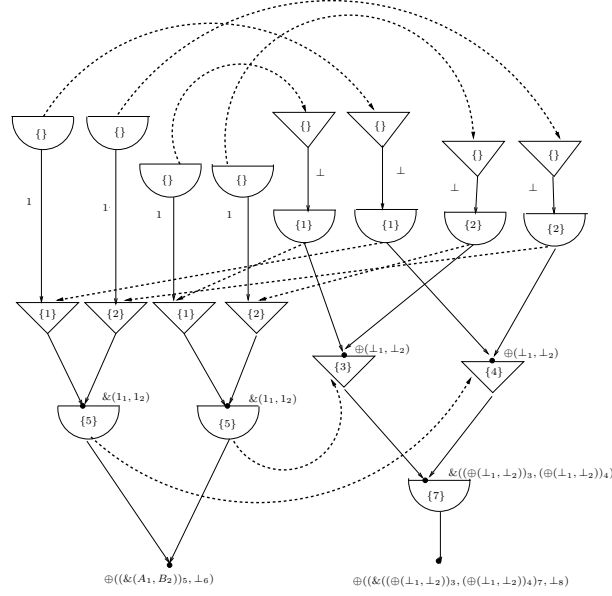


Figure 17: The J-proof net R_1

So we refine our procedure in the following way: instead of adding jumps directly on R , we add the jumps on the *slices* of R . Consider separately each of the four slices S_1, S_2, S_3, S_4 of R and add in each slice a jump from a to the element of W which appears in the slice as in fig. 21 (we mark in red the jump added in each slice). In this way we get four multiplicative J-proof structures S'_1, S'_2, S'_3, S'_4 ; now if we superpose them we retrieve exactly the J-proof structure R_1 in fig. 17. Observe that superposing S'_1, S'_2, S'_3, S'_4 duplicates the leftmost positive terminal link in R_1 . We could as well add jumps in each of the slices S_1, S_2, S_3, S_4 from the rightmost terminal positive link a' to elements of the leftmost $\&$ -rule W' and then superpose, obtaining the J-proof structure R_2 in fig 18. Observe that this time is the rightmost terminal positive link which is duplicated in R_2 .

5.4 Sequentializing with jumps: the general case

In the example above, we have considered only the case where the target of the jump we want to add belongs to a terminal $\&$ -rule. To deal with the general case, we must take into account the fact that adding a jump from a link a to a link b implies (by transitivity) implicitly adding a jump from a to every negative link b' s.t. $b \xrightarrow{+} b'$.

Definition 21 (Adding a jump) Let R be a total J-proof structure, and a (resp. b) a positive (resp. negative) link of R . To add a jump from a to b we perform the following operations:

1. take the set of all the slices S_1, \dots, S_n of R ;
2. for each slice S_i of R , modify S_i into S'_i in the following way:
 - if a does not belong to S_i , then $S'_i = S_i$;
 - if both a, b belongs to S_i , S'_i is obtained by adding a jump from a to b in S_i ;
 - if a belongs to S_i and b does not, then there exists at least one $\&$ rule W such that b depends from W in R (otherwise b would belong to S_i). To get S'_i we add a jump from a to w , for all negative links w in S_i such that for some $\&$ -rule W in R , $w \in W$ and b depends from W .

The result of adding a jump from a to b is the J-proof structure R' which is the superposition of S'_1, \dots, S'_n .

Remark 3 It is straightforward that adding jumps on a multiplicative J-proof structure, as in the example of subsection 5.2, is a special case of the procedure defined in Definition 21. In this special case the target of the jump to be added does not depend from any $\&$ -rule.

Proposition 10 Let R be a total J-proof structure, a a positive link and b a negative link; then there exists a unique total J-proof structure R' obtained by adding a jump from a to b .

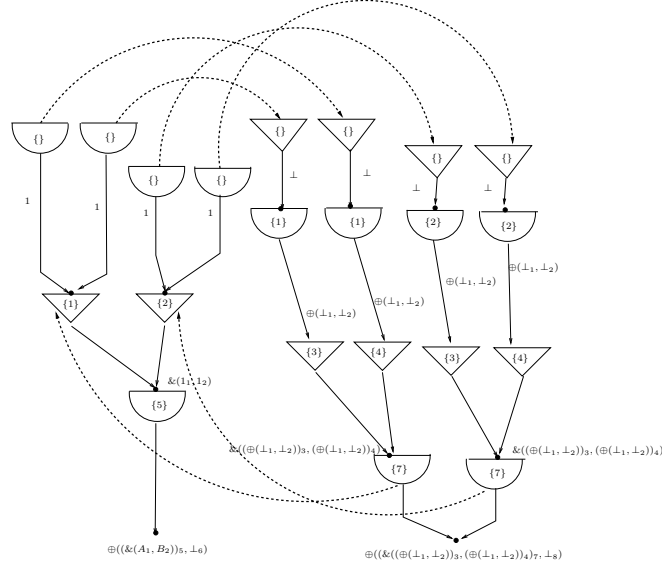


Figure 18: The J-proof net R_2

PROOF.

Let S_1, \dots, S_n be the slices of R and S'_1, \dots, S'_n be the multiplicative J-proof structures obtained by adding jumps following definition 21. To check that the superposition of S'_1, \dots, S'_n is a total J-proof structure, by proposition 7 we have just to check that compatibility condition on S'_1, \dots, S'_n is respected.

Let us consider two positive links a', a'' respectively belonging to S'_j, S'_k such that a', a'' are siblings.

Let us consider the occurrences² of a', a'' in the slices S_j, S_k of R from which S'_j, S'_k are build. We have to prove that there exist two negative links w' in S'_j w'' in S'_k , such that $a' \xrightarrow{+} w'$ in S'_j , $a'' \xrightarrow{+} w''$ in S'_k and w', w'' are siblings in S'_j, S'_k . We have two cases:

1. Suppose a' in S_j and a'' in S_k are not siblings; first we show that $a' \equiv a''$ in S_j, S_k . In fact, let a' in S_j and a'' in S_k be not siblings and suppose by absurd that a' is not sharing equivalent to a'' in S_j, S_k . Since a', a'' are siblings in S'_j, S'_k , then the link b' which is the parent of a' and the link b'' which is the parent of a'' are sharing equivalent in S'_j, S'_k . From the fact that a', a'' are neither siblings nor sharing equivalent in S_j, S_k we must conclude that b', b'' are not sharing equivalent in S_j, S_k (otherwise a', a'' would be siblings in S_j, S_k); but this would contradict the assumption that b', b'' are sharing equivalent in S'_j, S'_k , because two nodes which are not sharing equivalent in S_j, S_k cannot be sharing equivalent in S'_j, S'_k (since the set of predecessors of a link can only increase by adding jumps). We must conclude then that $a' \equiv a''$ in S_j, S_k ; then it is straightforward that a', a'' are two different occurrences in S_j, S_k of the same link c of R . Now the set of predecessors of c are different in S'_j and S'_k (since a', a'' are siblings in S'_j, S'_k) so it must be that $c \xrightarrow{+} a$, where a is the source of the jumps to be added in R . Now let us consider all the $\&$ -rules b depends from in R . There must exists at least one $\&$ -rule Z such that b depends from Z in R , and for two different elements z, z' of $Z \in S_j, z' \in S_k$; otherwise by definition 21, we would add the same jumps from a in S'_j and in S'_k , and then a', a'' would be sharing equivalent in S'_j, S'_k since $a' \equiv a''$ in S_j, S_k and $a' \xrightarrow{+} a$ in S'_j (resp. $a'' \xrightarrow{+} a$ in S'_k). By the existence of z, z' and by definition of adding jumps, $a \xrightarrow{+} z'$ in S'_j and $a \xrightarrow{+} z''$ in S'_k ; since z', z'' belong to the same $\&$ -rule in R , they are siblings in S_j, S_k ; the set of predecessors of z' (resp. of z'') is the same in S_j and in S'_j (resp. in S_k and in S'_k), so they are sibling also in S'_j, S'_k , and $a' \xrightarrow{+} z'$ in S'_j , $a'' \xrightarrow{+} z''$ in S'_k . We conclude that S'_j, S'_k respect the compatibility condition.
2. Suppose a' in S_j and a'' in S_k are siblings: then by contraction condition on R (and the equivalence between compatibility and contraction condition) there exists respectively a negative link w' in S_j s.t. $a' \xrightarrow{+} w'$ and a negative link w'' in S_k s.t. $a'' \xrightarrow{+} w''$ and w', w'' are siblings (so they belong to the same $\&$ -rule W of R). Now we have two sub-cases:
 - w', w'' are siblings also in S'_j, S'_k . Then compatibility condition is respected.

²Since slices are subgraphs of a given graph, it makes sense to talk of different occurrences of the same node in different slices.

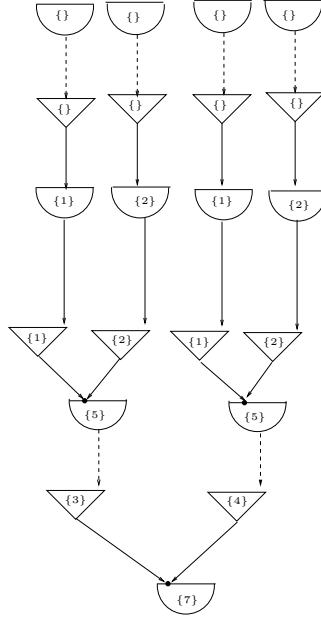


Figure 19: The skeleton of R_1

- w', w'' are not siblings in S'_j, S'_k . If this is the case, then $w' \xrightarrow{+} a$ in S_j , $w'' \xrightarrow{+} a$ in S_k , and adding the jump from a to b makes that in S'_j, S'_k , w', w'' are no more siblings. Now let us consider all the $\&$ -rules b depends from in R ; using the same argument used in point 1), we can prove that there must exist a pair of negative links z' in S_j, z'' in S_k s.t. z', z'' are two different elements of the same $\&$ -rule Z of R , b depends from Z in R , $a \xrightarrow{+} z'$ in S'_j (but not in S_j) and $a \xrightarrow{+} z''$ in S'_k (but not in S_k). Since z', z'' belong to the same $\&$ -rule in R , they are siblings in S_j, S_k ; by definition of adding jumps, the set of predecessors of z' (resp. of z'') is the same in S_j and in S'_j (resp. in S_k and in S'_k), so they are sibling also in S'_j, S'_k , and $a' \xrightarrow{+} z'$ in $S'_j, a'' \xrightarrow{+} z''$ in S'_k , and we are done.

Concerning totality of R' , is trivially preserved from totality of R . The result is a consequence of the following observation: take a slice S_i of R containing an element w of a $\&$ -rule W of R . Now if we choose a different element w' of W there exists (by totality of R) exactly one slice S_j of R containing w' and such that for any other $\&$ -rule Z of R , S_i, S_j contain the same element z of Z . If we consider the multiplicative J-proof structures S'_i, S'_j corresponding respectively to S_i, S_j , induced by the adding of jumps, it is easy to see that the occurrence of w in S'_i and the occurrence of w' in S'_j are siblings, so they belong to the same $\&$ -rule of R' .

□

Proposition 11 *Let R, R' be two total J-proof structure s.t. R' is obtained from R by adding jumps and $\prec_{R'}$ is arborescent. Then R is sequentializable.*

PROOF. By proposition 9, R' is sequentializable into a unique proof π . We show that also R is sequentializable into π , by proving that if a link sequentializes R' , then a corresponding link of R sequentializes R too.

We reason by induction on the number of links in R :

$n = 1$: in this case, R and R' are composed by just a positive link without premises, trivially sequentializable;

$n = k + 1$: Suppose R, R' have a negative conclusion. If it is the conclusion of a $\&$ -rule consider the corresponding terminal $\&$ -rule $W = \{w_1 \dots, w_n\}$ in R' ; since R' is sequentializable, W sequentializes R' into $\{R'_1, \dots, R'_n\}$ arborescent, sequentializable J-proof structures. Now we take the corresponding terminal negative rule $W' = \{w'_1 \dots, w'_n\}$ of R ; for each element w'_i of W' by proposition 2 the graph $R'_{w'_i}$ obtained by removing w'_i from the scope $R_{w'_i}$ is a total J-proof structure. Since R' is obtained by adding jumps on R , each slice S' of R' can be obtained from a unique slice S of R by adding jumps on it. It is not difficult to prove then that each R'_i can be obtained by adding jumps on $R_{w'_i}$, so $R_{w'_i}$ is sequentializable by induction hypothesis, and W' sequentializes R . But then R is sequentializable.

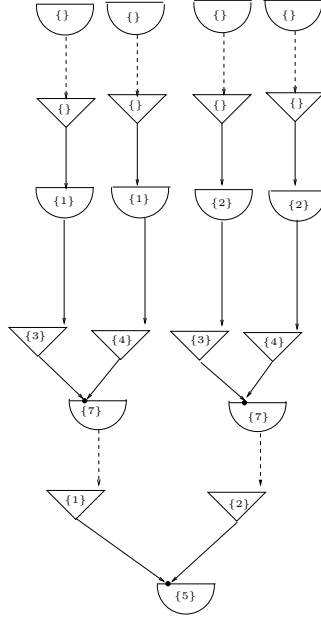


Figure 20: The skeleton of R_2

If the negative conclusion of R, R' is not the conclusion of a $\&$ -rule we consider the corresponding terminal negative link c of R' ; since R' is sequentializable, c sequentializes R' into an arborescent, sequentializable J-proof structure R'_0 ; it is easy to check that the J-proof structure R_0 obtained by removing the terminal negative link c' corresponding to c from R is a total J-proof structure and that R'_0 can be obtained by adding jumps on R_0 , so by induction hypothesis R_0 is sequentializable and c' sequentializes R .

Otherwise, we reason by cases, depending if R' is composed by one or more than one connected component:

- if R' is composed by a single connected component, we consider the terminal positive link c of R' which is minimal in $\prec_{R'}$; c sequentializes R' into $\{R'_1, \dots, R'_n\}$ arborescent, sequentializable J-proof structures, so c is splitting for R . The corresponding terminal positive link c' of R must be splitting too (because adding jumps preserve connectedness) so it splits R into $\{R_1, \dots, R_n\}$ total J-proof structures such that each R'_i can be obtained by adding jumps on R_i . But then $\{R_1, \dots, R_n\}$ are sequentializable by induction hypothesis, and c' sequentializes R , so R is sequentializable.
- if R' is composed by $\{R'_1, \dots, R'_n\}$ connected components, each component R'_i is an arborescent, sequentializable J-proof structure; is easy to verify in this case R can be divided into $\{R_1, \dots, R_n\}$ disjoint, total J-proof structure and that each R'_i can be obtained by adding jumps on R_i , so each R_i by induction hypothesis is sequentializable. Then R is sequentializable.

□

5.5 Sequentialization

Now all is set to proceed with the main result of this section, namely the proof of the sequentialization theorem. To prove that a J-proof net is sequentializable, we use the following road map:

- given a J-proof net R , we iteratively add jumps on R until we get a J-proof net R' which contains a “maximal” amount of jumps; that is, any jump added on R' either breaks the correctness criterion (obtaining something which is not a J-proof net anymore) or is redundant (it does not increase the order associated to R' as a d.a.g.). J-proof nets as R' are called *saturated* (we formally define them below).
- **The order associated with a saturated J-proof net is arborescent** (Lemma 3).
- If the order associated with R' is arborescent, there exists a unique proof π which is the sequentialization of R' (Prop. 9).
- if R' sequentializes into π , R too is sequentializable into π (prop.11); so R is sequentializable.

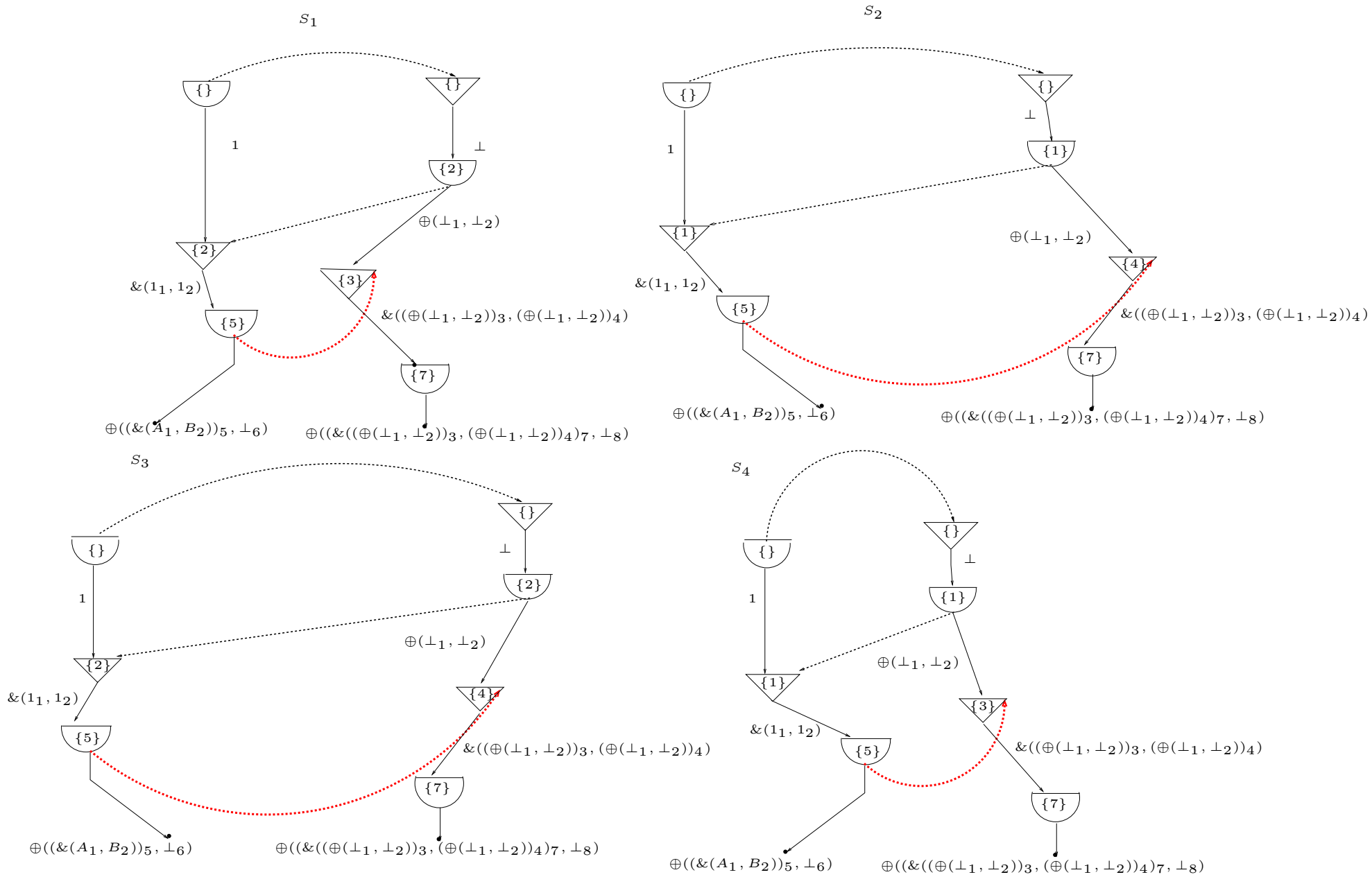


Figure 21: adding a jump on R

The crucial point is the proof of Lemma 3; a restricted version of Lemma 3 for the multiplicative fragment was the main technical novelty introduced in [DGF06].

A major obstacle to the extension of Lemma 3 to additives is the possible presence of “legal” switching cycles in additive J-proof nets, allowed by Cycles condition of the correctness criterion, but forbidden in the multiplicative case.

The first step towards the proof of sequentialization will be then to prove that Lemma 3 still holds in presence of additives; the auxiliary result allowing this will be Lemma 2, which states that *any saturated J-proof net does not contain switching cycles*.

5.5.1 A saturated J-proof net is switching acyclic.

Definition 22 (Saturated J-proof net) *A J-proof net R is saturated if for every J-proof structure R' obtained from R by adding a jump, either R' is not a J-proof net or $\prec_R = \prec_{R'}$.*

Given a J-proof net R , a saturation R' of R is a saturated J-proof net obtained from R by adding jumps.

Definition 23 (Strong switching path) *Given a negative link b of of a J-proof structure R , a **strong switching path** $\langle b, \dots, a \rangle$ from a node b to a node a is a switching path which does not use any switching edge of b (or of any other node b' belonging to the same $\&$ -rule as b in R).*

Proposition 12 *Let a be a positive link, b a negative link and W any $\&$ -rule in a J-proof net R , s.t. b depends from W . If there isn't any strong switching path from b to a in R , then there isn't any strong switching path from w to a in R for any element w of W .*

PROOF. The proof is a direct consequence of the fact that, if there was a strong switching path from w to a , then there would be also a strong switching path from b to a , since b depends from W , so $b \xrightarrow{+} w$ for some element w of W by Prop. 1; this would contradict the hypothesis, q.e.d. \square

Proposition 13 *Let R be a J-proof net, a be a positive link and b a negative link of R ; if there isn't any strong switching path from b to a in R , then the J-proof structure R' obtained by adding a jump from a to b in R , is a J-proof net.*

PROOF. By proposition 10, we must only prove that Cycles condition holds. The property follows from the fact that adding a jump from a to b cannot create new switching cycles: any new switching cycle in R' should use a jump from a which was not in R , and the conclusion of the link which is the target of the jump. A cycle like that cannot exist in R' , by the hypothesis that there is not any strong switching path from b to a in R^3 , and Prop. 12. \square

Lemma 1 *Let R be a J-proof net, \mathcal{C} a union of switching cycles of R ; then there exists an additive pair $w_1, w_2 \in W$ of R which breaks \mathcal{C} and a positive link $c \in \mathcal{C}$ (we call such a positive link **proper**) s.t.*

1. $\neg(c \xrightarrow{+} w_1)$ and $\neg(c \xrightarrow{+} w_2)$;
2. c belongs to a cycle $C' \in \mathcal{C}$ s.t. there exists a node $d \in C'$ which is hereditarily above w_1 or w_2 .

PROOF.

The proof is by induction on the number n of cycles in \mathcal{C} :

$n = 1$: Since $n = 1$, \mathcal{C} is composed by a single switching cycle C ; by the correctness criterion, there exists an additive pair $w_1, w_2 \in W$ which breaks C in R . Let's suppose by absurd that every link of C is above either w_1 or w_2 ; then we can partition the nodes of C in two sets, $A = \{a : a \xrightarrow{+} w_1\}$ and $B = \{b : b \xrightarrow{+} w_2\}$, disjoint by condition coherence of the definition of J-proof structure. Given any two elements $a \in A$ and $b \in B$, there exists a path $r : \langle a \dots b \rangle$ connecting them. We consider the first edge of r starting from a which connects a node d of A with a node d' of B ; either it is an incident edge $d \rightarrow d'$, and then $d \xrightarrow{+} w_1$ and $d \xrightarrow{+} w_2$, or it is an emergent edge $d \leftarrow d'$, and then $d' \xrightarrow{+} w_1$ and $d' \xrightarrow{+} w_2$; in any case we contradict the condition coherence of definition 3, so there exists some link c s.t. $\neg(c \xrightarrow{+} w_i)$ for $i = 1, 2$. Furthermore there has to be at least one positive link which enjoys the property, otherwise C would not be switching.

³It is not possible that adding a jump having as target b in R increases the number of strong switching paths starting from b in R' since such paths cannot use the switching edges of b (and a jump is a switching edge).

$n > 1$: We proceed *ad absurdum* and we suppose that for all additive pairs in R breaking \mathcal{C} , \mathcal{C} does not contain a positive proper link.

Let us consider an additive pair $w_1, w_2 \in W$ breaking \mathcal{C} in R , which exists by the correctness criterion on R .

Then by our hypothesis, by definition of proper link and by coherence condition on J-proof structures, we can partition the cycles of \mathcal{C} in three groups: \mathcal{C}_1 (the cycles with all elements above w_1), \mathcal{C}_2 (the cycles with all elements above w_2) and \mathcal{C}_0 (the cycles whose elements are neither above w_1 or w_2). Now by induction hypothesis on $\mathcal{C}_1 \cup \mathcal{C}_0$, there exists an additive pair w'_1, w'_2 which breaks $\mathcal{C}_1 \cup \mathcal{C}_0$ and a positive link c' belonging to some switching cycle $C' \in \mathcal{C}_1 \cup \mathcal{C}_0$ such that c' is proper. Obviously neither w'_1 or w'_2 can belong to \mathcal{C}_2 ; otherwise they would be above w_2 , and then either there would be some cycle in \mathcal{C}_1 which is above w_1 and w_2 , contradicting our hypothesis on \mathcal{C}_1 , or there would be some cycle in \mathcal{C}_0 which is above w_2 , contradicting our hypothesis on \mathcal{C}_2 . But then the additive pair w'_1, w'_2 breaks \mathcal{C} too, and c' is a proper link of \mathcal{C} contradicting our hypothesis that, for all additive pairs breaking \mathcal{C} , \mathcal{C} does not contain any proper positive link. □

Lemma 2 *Let R be a J-proof net; if R contains a switching cycle, then R is not saturated.*

PROOF.

We consider the union \mathcal{C} of all the switching cycles of R (there is at least one by hypothesis). There exists, by lemma 1, an additive pair w_1, w_2 which breaks \mathcal{C} and a positive proper link c belonging to a cycle $C' \in \mathcal{C}$. By the fact that c is proper there exists a node d belonging to the same cycle $C' \in \mathcal{C}$ as c s.t. d is hereditarily above w_1 or w_2 ; w.l.o.g., let us assume that d is hereditarily above w_1 . By the existence of d there exists a switching path r' from c to w_1 , composed by a switching path from c to d (containing only nodes of C') and nodes in a directed path from $d \in C'$ to w_1 .

Let's suppose that w_1 belong to a terminal &-rule W of R : in this case by proposition 13 we can add a jump from c to W , this doesn't create cycles and increases the order.

Otherwise W is not terminal; if this is the case, we show that there cannot be any strong switching path from a link w' in W to c . Let us suppose that there is a strong switching path $r : \langle w' \dots c \rangle$ in R ; now if r and r' are disjoint by composing them we get a switching cycle intersecting W ; by correctness criterion, since w_1, w_2 break \mathcal{C} and w_1, w_2 belong to W , W doesn't intersect any switching cycle of \mathcal{C} , and since \mathcal{C} contains all the switching cycles of R , we contradict the hypothesis that R is correct.

If r and r' do intersect, let's take the first point e starting from w' and going down on r where r meets C' (if r doesn't meet C' , this means that r and r' intersect on the directed path from d in C' to w_1 ; then we have a switching cycle intersecting W , contradicting correctness of R as above). The only interesting case is if e is negative: by the fact that e is in a switching cycle where at least d is above w_1 , there exists a strong switching path r'' from e to w_1 , so we compose the subpath of r from w' to e with r'' and we get a switching cycle intersecting W , contradicting correctness of R as above.

So there isn't any strong switching path from w' to c , then by proposition 13 we can add a jump from c to W in R , and retrieve a J-proof net R' . □

5.5.2 A saturated J-proof net is arborescent.

Lemma 3 (Arborisation) *Let R be a J-proof net. If R is saturated then \prec_R is arborescent.*

PROOF.

We prove that if \prec_R is not arborescent, then there exists a negative and a positive link s.t. we can add a jump between them which does not create cycles and makes the order increase, the proof being just an adaption to J-proof nets of the arborisation lemma of [DGF06].

If R contains some switching cycles, then we apply lemma 2 and we have done; so we can restrict ourselves to the case where R doesn't contain any switching cycle.

If \prec_R is not arborescent, then in \prec_R there exists a link a with two immediate predecessors b and c (they are incomparable). Observe that b and c are immediately below a in $Sk(R)$ and also in R , that a must be positive and that b and c are two negative links which cannot depend from the same & rule, by condition coherence on a .

We have two possibilities:

1. either b or c is terminal in R . Let assume that b is terminal; then c cannot be terminal (by definition of J-proof structure), and there is a positive link c' which immediately precedes c . Then we can add a jump between b and c' (see fig 22).

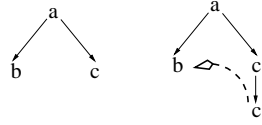


Figure 22: Case 1 in the proof of the arborisation lemma

2. Neither b or c are terminal in R . Each of them has an immediate positive predecessor, respectively b' and c' .

Now we want to prove that either we can add a jump from b' to c , or we can add a jump from c' to b .

Let's suppose that we cannot add any jump in R from b' to c ; then by proposition 13 there is in R a strong switching path $r = \langle c, c' \dots b \rangle$. If we cannot add a jump from c' to b too, then there is a strong switching path $r' = \langle b, b' \dots c \rangle$ in R .

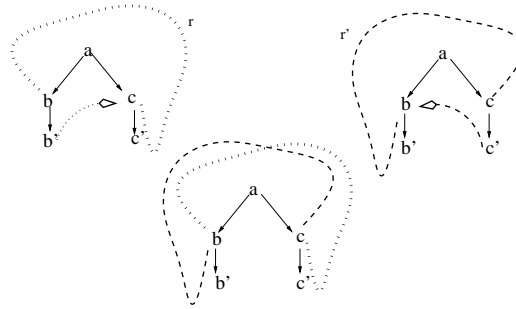


Figure 23: Case 2 in the proof of the arborisation lemma, where the paths r, r' are disjoint.

Assume that r and r' are disjoint: we exhibit a switching cycle in $R \langle c, c' \dots b, b' \dots c \rangle$ by concatenation of r and r' , contradicting the hypothesis that R has no switching cycles (see fig 23).

Assume that r and r' are not disjoint. Let d be the first node (starting from b) where r and r' meets. Observe that d must be negative (otherwise there would be a cycle). Each path uses one of the premises, and the conclusion of d (hence the paths meet also in the node below d). From the fact that d is the first point starting from b where r and r' meet it follows that: (i) r' enters in d using one of the switching edges of d , and exits from its conclusion; (ii) each of r and r' must use a different switching edge of d . Then we distinguish two cases (see fig 24):

- r enters d from one of its switching edges; we build a switching cycle taking the sub path $\langle b, \dots, d \rangle$ of r' and the sub path $\langle d, \dots, b \rangle$ of r .
- r enters d from the conclusion; then we build a switching cycle composing the sub path of $r \langle c, \dots, d \rangle$, the reversed sub path of $r' \langle d, \dots, b \rangle$ and the path $\langle b, a, c \rangle$.

□

Proof of Theorem 1 PROOF. The left to right direction is trivial as usual; concerning the right to left direction, let us consider a saturation R' of R . By the arborisation lemma, $\prec_{R'}$ is arborescent, so by proposition 9 R' is sequentializable into a unique proof π . But then, by proposition 11, R too is sequentializable into π . □

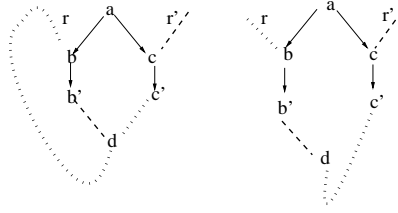


Figure 24: Case 2 in the proof of the arborisation lemma, where the paths r, r' are not disjoint.

5.5.3 J-proof nets and Mix rule

The proof of sequentialization provided above, could be easily adjusted in order to characterize exactly the J-proof nets whose sequentializations do not contain any occurrence of the Mix rule. As usual in proof nets syntax, this is done by properly introducing the notion of *correction graph*; we skip the proof, which does not present any relevant difference w.r.t. the standard one (see [Gir87]).

Definition 24 (Correction graph) *Given a J-proof net R , a **switching** s is the choice of a switching edge for every $\&$ rule of R (and for every negative link which does not belong to a $\&$ -rule); a **correction graph** $s(R)$ is the graph obtained by erasing the switching edges of R not chosen by s .*

Definition 25 (s-connected) *A J-proof net R is s-connected if all its correction graphs are connected.*

Theorem 2 *Let R be a J-proof net which is s-connected; then R is sequentializable, and any proof π associated with R does not make use of the Mix rule.*

6 Cut elimination and semantics

In this section we study the dynamics of cut-reduction in J-proof structures.

First in subsection 6.1 we discuss about sequentialization in presence of cut-links.

Then in subsection 6.2 we define cut-elimination on J-proof structures. As in L-nets, reduction is performed in parallel on each slice: so to reduce a J-proof structure R , we will decompose R in slices, perform reduction separately on each of them, and then superpose all the cut free slices obtained in this way.

In subsection 6.3 we provide a denotational semantics of J-proof nets, using a variation of standard relational semantics based on the notion of *pointed set* (i.e. a set A equipped with a special element called *point* of A). The aim is to refine the relational model, in order to be able to semantically characterize sequential ordering, which in our setting is represented by *jumps*; as a matter of fact, jumps usually are not captured by relational semantics.

Actually, our approach is inspired by [Bou04], where a step is made in the direction of developing a unified framework for both *static* (sets, coherence spaces, etc) and *dynamic* (games) denotational semantics.

Finally, in 6.4 we solve the question, left opened in 6.1, of the stability of correctness under cut-reduction.

6.1 Cut and sequentialization

Unfortunately, we cannot straightforwardly extend our proof of sequentialization in presence of *cut*-links. The problem relies in the definition of superposition of J-proof structures: as a matter of fact, superposing J-proof structures in presence of cut links is quite difficult. This is not a novelty: actually, in the framework of sliced polarized proof nets of [LTdF04], a similar problem is present, which makes hard to conciliate the presence of cuts *inside* proof nets and sequentialization.

The way out is to consider only cut-free J-proof nets (for which we can prove sequentialization), compose them using *cut*-links, and then reducing the J-proof net obtained until we reach the normal form (which is cut-free, so that we can deal with it again).

The absence of “internal” cut links is not so restricting as it seems: as pointed out in [LTdF04], the “syntactical” difference between considering or not “internal” cut links is the same as the “semantical” distinction between “small steps” and “big steps” operational semantics in computer science.

The central point of this argument is the preservation of the property of being sequentializable under cut reduction; we prove this result in section 6.4 by using the injectivity property of pointed semantics with respect to J-proof nets, that we state in subsection 6.3.2; actually, this strategy is the same used by Laurent and Tortora de Falco for sliced polarized proof nets, using relational semantics.

6.2 Cut elimination

Now we define cut elimination on J-proof structures; we first begin by defining cut reduction for multiplicative J-proof structures in subsection 6.2.1. In subsection 6.2.2 then we extend cut elimination to the general case. In subsection 6.2.3 we prove that total J-proof structures are closed under cut-reduction, assuming a weaker form of correctness (namely that all slices composing a J-proof structure are switching acyclic).

6.2.1 Cut elimination on J-proof structures: the multiplicative case

There are two kinds of cut-elimination steps (we denote by $R \rightsquigarrow R'$ the relation “ R reduces to R' ”):

1. the $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ (where $I = I$) reduction step, which replaces the module β in R (depicted on top of Fig. 25) with the module γ (depicted on bottom of Fig. 25).
2. the $+_{J \in \mathcal{N}} / -_{K \in \mathcal{N}}$ (where $J \neq K$) reduction step, which replaces R with the empty J-proof structure (see Fig. 26).

Let $R \rightsquigarrow R'$; when a node a (resp. an edge l) of R' comes from a (unique) node (resp. edge) \overleftarrow{a} (resp. \overleftarrow{l}) of R we say that \overleftarrow{a} (resp. \overleftarrow{l}) is the *ancestor* of a (resp. l) in R and that a (resp. l) is a *residue* of \overleftarrow{a} (resp. \overleftarrow{l}) in R' . We denote sometimes the residues of a node b (resp. an edge r) by \overrightarrow{b} (resp. \overrightarrow{r}).

Definition 26 (Correct multiplicative J-proof structure) *A multiplicative J-proof structure is **correct** iff it is switching acyclic.*

Proposition 14 *Let c be a cut link between a positive link a and a negative link b in R , such that for some links a', b' of R , $b' \xrightarrow{+} b$ and $a \xrightarrow{+} a'$; then if $R \rightsquigarrow R'$ by reducing c with a $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step, in R' $b' \xrightarrow{+} a'$.*

PROOF. Straightforward from the definition of $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step. □

With respect to the rewriting rules $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$, $+_{K \in \mathcal{N}} / -_{J \in \mathcal{N}}$, reduction enjoys the following properties:

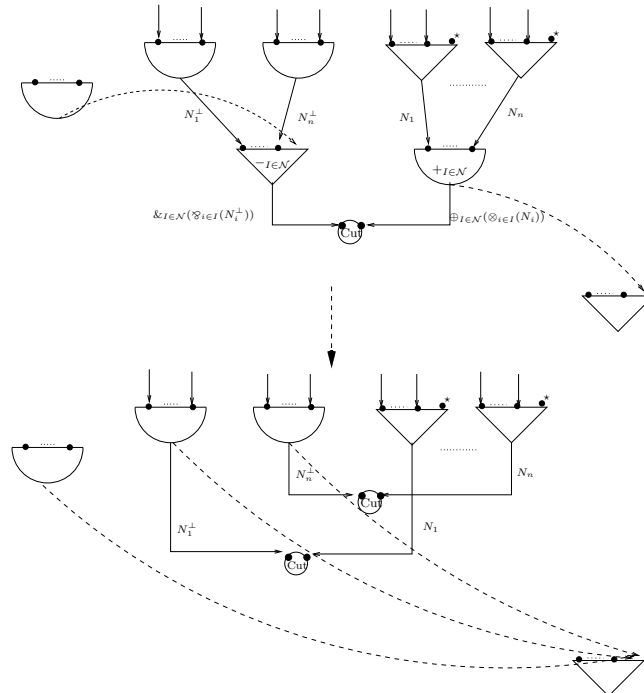


Figure 25: $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ cut reduction.

Theorem 3 (Preservation of correctness) *Given a multiplicative J-proof structure R , if R is correct and $R \rightsquigarrow R'$, then R' is correct.*

PROOF.

If the cut t to be reduced in R is reduced through a $+_{K \in \mathcal{N}} / -_{J \in \mathcal{N}}$ step, the result is obvious. Otherwise, if t is reduced through a $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step, we proceed ad absurdum: we assume that there is a switching cycle in R' , and we show that this contradicts the hypothesis that R is correct.

Now, if the cycle does not cross the module γ in R' , then the switching cycle is also in R and we are done.

Otherwise, it does cross γ : let us call c_1, \dots, c_n the cut links of R' created by reducing t in R , a_i being the negative premise and b_i being the positive premise of c_i for $i = 1, \dots, n$; let us call f any negative link and g any positive link in γ in R' such that g jumps on f but \overleftarrow{g} does not jump on \overleftarrow{f} in R .

The switching cycle may cross the module in different ways, using n new cuts and m new jumps created by reducing t in R ; we detail some relevant cases, showing that each of them brings to a contradiction.

1. Suppose the cycle crosses exactly one cut link c_i of R' created by reducing t in R and no new jumps (so $n = 1$ and $m = 0$); in this case, the cycle connects a_i with b_i with a path going outside the module γ in R' . Then there is a switching cycle in R , obtained by choosing $\overleftarrow{b_i}$ as a switching edge in R .
2. Suppose that the cycle crosses exactly two new cut links c_i, c_j of R' created by reducing t in R and no new jumps (so $n = 2$ and $m = 0$). Then we have two subcases:
 - Suppose the cycle connects a_j with b_i and b_j with a_i (for $j \neq i$) with a path going outside the module γ in R' ; then we reason as in case 1, opportunely choosing a switching edge.
 - Suppose the cycle connects a_i with a_j and b_i with b_j (for $j \neq i$) with a path going outside the module γ in R' . Then it is easy to see that there is a switching cycle in R too, since $\overleftarrow{a_i}$ and $\overleftarrow{a_j}$ are connected inside the module β in R .

If the cycle crosses more than two cut links of R' created by reducing t in R and no new jumps (so $n \geq 2$ and $m = 0$) the argument is completely analogous.

3. suppose the cycle does not cross any of the new cut links of R' created by reducing t in R , but it does cross one new created jump (so $n = 0$ and $m = 1$). In this case the cycle connects f with g outside the module γ in R' . Then there is a switching path connecting \overleftarrow{g} with \overleftarrow{f} outside the module β in R too, and is easy to see that there is a switching cycle in R too (since \overleftarrow{g} and \overleftarrow{f} are connected in the module β in R). If the cycle crosses more than two new jumps created by reducing t in R and no new cut links (so $n = 0$ and $m \geq 1$) the argument is completely analogous.
4. suppose the cycle crosses at least one cut link c_i of R' created by reducing t in R , and one new jump (so $n \geq 1$ and $m = 1$). In this case the cycle connects a_i with f outside the module γ in R' . Then there is a switching path connecting \overleftarrow{f} with $\overleftarrow{a_i}$ outside the module β in R too, and is easy to see that there is a switching cycle in R too (since \overleftarrow{f} and $\overleftarrow{a_i}$ are connected in the module β in R). If the cycle crosses more than two new jumps created by reducing t in R (so $n \geq 1$ and $m \geq 1$) the argument is completely analogous.

In all cases, supposing that R' contains a switching cycle implies that R contains a switching cycle, contradicting the hypothesis that R is correct. So R' is correct. □

Theorem 4 (Strong normalization) *For every correct multiplicative J-proof structure S , there is no infinite sequences of reductions $S \rightsquigarrow S_1 \rightsquigarrow S_2 \dots \rightsquigarrow S_n \dots$*

PROOF.

Trivial by the fact that at each step the number of links decreases. □

Theorem 5 (Confluence) *For every correct multiplicative J-proof structure S_1, S_2 and S_3 , such that $S_1 \rightsquigarrow S_2$ and $S_1 \rightsquigarrow S_3$, there is a slice S_4 , s.t. $S_2 \rightsquigarrow S_4$ and $S_3 \rightsquigarrow S_4$.*

PROOF.

It easily follows from confluence of standard multiplicative proof nets (see [Gir87]). □

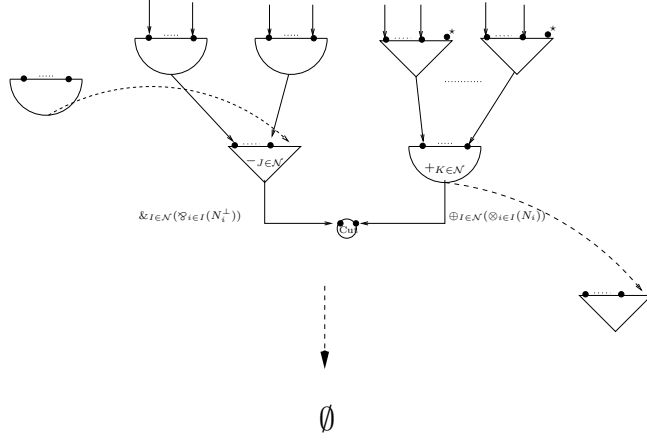


Figure 26: $+_{K \in \mathcal{N}} / -_{J \in \mathcal{N}}$ cut reduction (with $J \neq K$).

6.2.2 Cut elimination on J-proof structures: the general case

Once defined reduction for multiplicative J-proof structures, reduction of a total J-proof structure R in the general case follows these steps:

1. take the set of all the slices S_1, \dots, S_n of R ;
2. for each slice S_i of R , reduce S_i to reach its normal form S'_i ;
3. take the J-proof structure R' which is the superposition of S'_1, \dots, S'_n .

One condition is necessary in order to well define reduction: the normal forms of the slices composing a J-proof structure must be cut-free, otherwise it would be impossible to superpose them. For this purpose we require that J-proof structures for being reduced satisfy a constraint called *weakly correctness*:

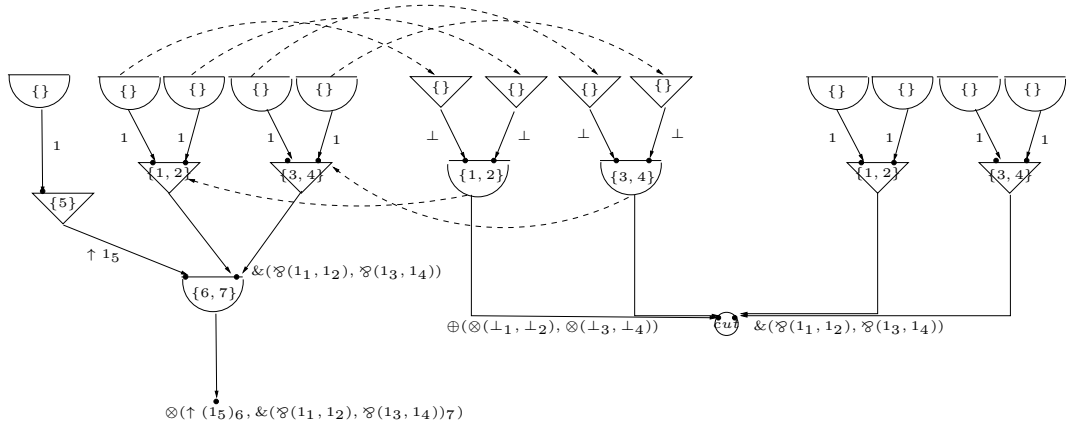
Definition 27 A J-proof structure is **weakly correct** when is total and all its slices are correct (that is switching acyclic).

The fact that the normal form of a slice of a weakly correct J-proof structure is cut-free is a simple consequence of Theorem 3.

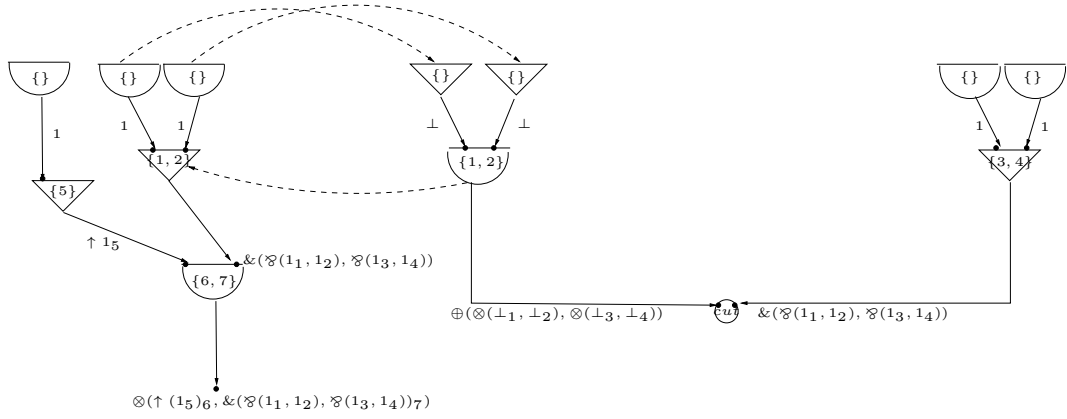
Now we can formally define the *normal form* of a J-proof structure:

Definition 28 Let R be a weakly correct J-proof structure and $\{S_1, \dots, S_n\}$ be the set of the slices of R . If S'_1, \dots, S'_n are the cut-free slices obtained by reducing $\{S_1, \dots, S_n\}$, we call **normal form of R** (denoted by $[R]$) the J-proof structure R' which is the superposition of S'_1, \dots, S'_n .

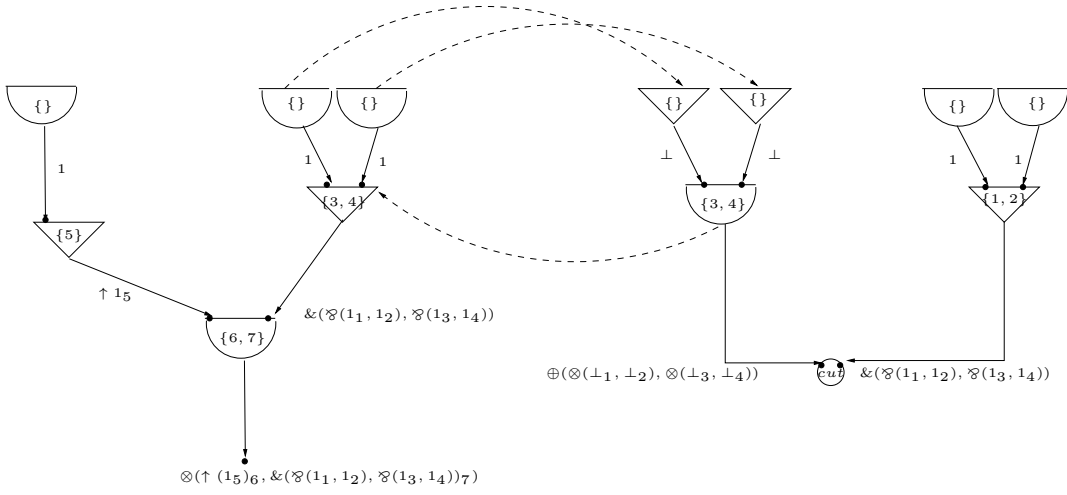
Example of cut elimination Let us consider the following J-proof structure R :



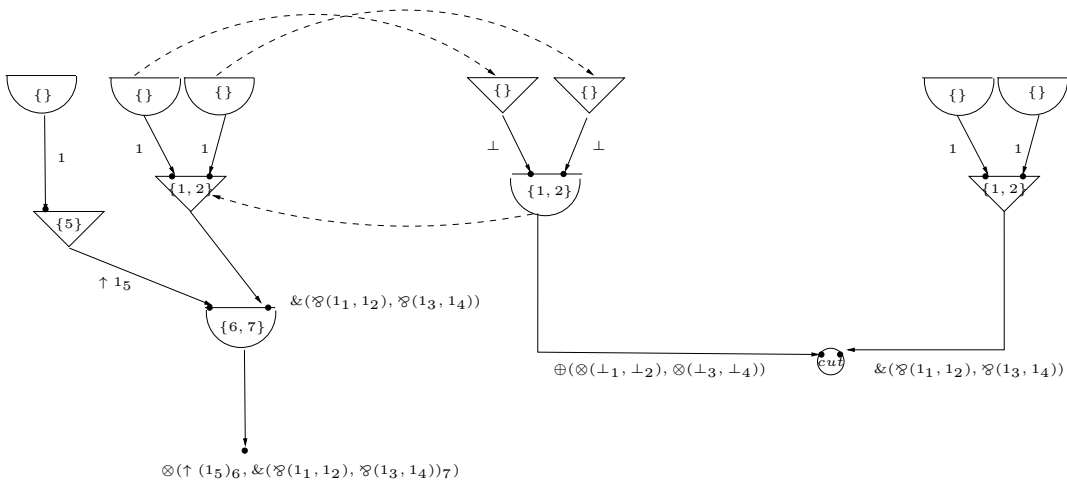
To reduce it to normal form we consider each of the four slices of R .
The slice S_1 :



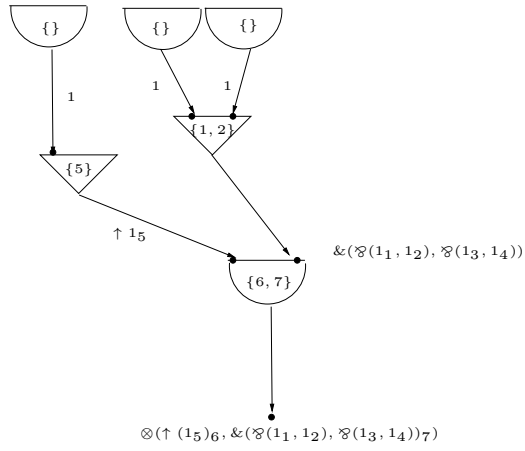
and the slice S_2 :



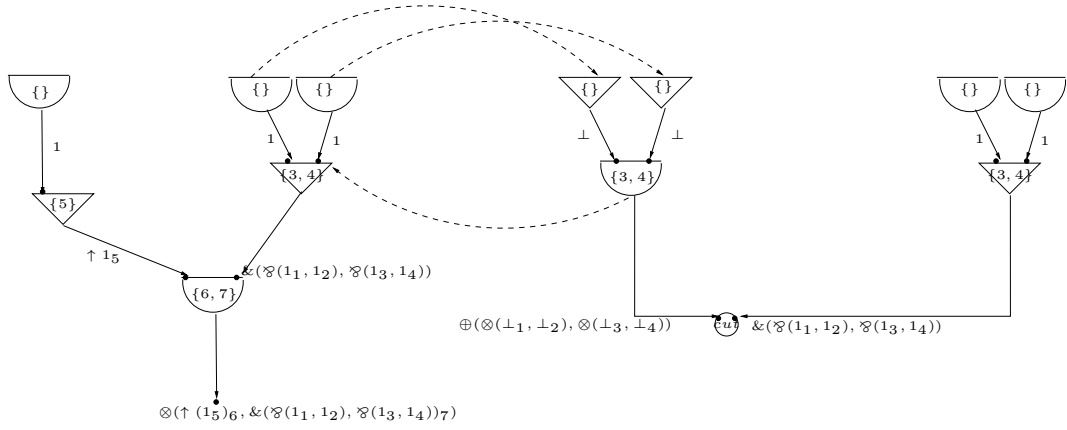
both reduce to the empty slice through a $+K \in \mathcal{N} / -J \in \mathcal{N}$ step.
The slice S_3 :



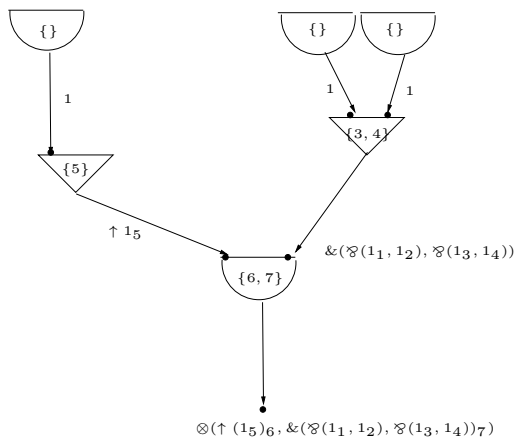
reduces to the slice S'_3 :



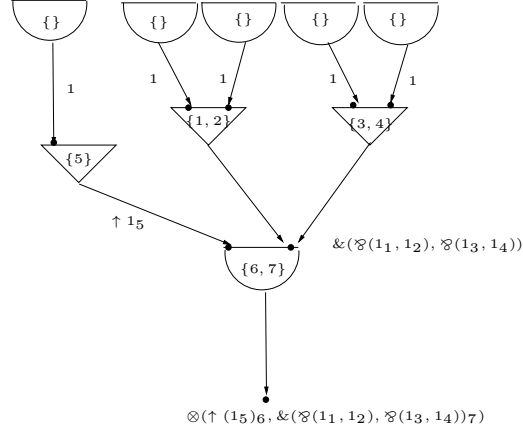
the slice S_4 :



reduces to the slice S'_4 :



Now if we consider the superposition of the normal forms of S_1, S_2, S_3, S_4 we get the J-proof structure R' below, which is the normal form of R :



6.2.3 Preservation of weakly correctness

Now we want to prove that the normal form $[R]$ of a weakly correct J-proof structure R is still a weakly correct J-proof structure; the delicate part is proving that $[R]$ is still a J-proof structure.

Given a link a of a J-proof structure R , we say that a belongs to the *right (resp. left) branch* of a cut-link c iff there exists a directed path from a to c , which does not use jumps, entering c by its right (resp. left) port.

Given a J-proof structure R and a link a of R , we say that a is *hidden* by a cut link c (or simply hidden) if a belongs to a branch of a cut link c of R , we call it *visible* otherwise.

A slice is *persistent* if it does not reduce itself to the empty slice.

If S is a persistent slice, and a is a link hidden by some cut-link c of S , the *opposite link* of a is the link b in the opposite branch of c w.r.t. a , such that there exists a slice S' obtained by a sequence of reduction from S where the conclusion of a and the conclusion of b are premises of the same cut link c' of S' .

Remark 4 Consider an hidden negative link a of a persistent slice S and its opposite link b ; a and b belongs to two different branches of the same cut-link c . Now if there are two links a', b' in S such that $a' \xrightarrow{+} a$ and $b \xrightarrow{+} b'$ and a', b' are not hidden by c , then by definition of $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step, by proposition 14 and by definition of opposite link, there is a slice S' obtained by reducing S such that in S' $a' \xrightarrow{+} b'$.

Lemma 4 Let S_1, S_2 be two persistent slices of a weakly correct J-proof structure R . Let a and b be a pair of positive links of R , such that $a \xrightarrow{+} w$ in S_1 and $b \xrightarrow{+} y$ in S_2 , where w, y are two negative links of R which form an additive pair. Then there exists a finite sequence $s = \langle (w_1, y_1), \dots, (w_n, y_n) \rangle$ of additive pairs of R , s.t.:

1. $w_1 = w, y_1 = y$;
2. for each $i \neq n$, w_i, y_i are hidden, their respectively opposite links a', b' are contracted in R and $a' \xrightarrow{+} w_{i+1}$ in S_1 , $b' \xrightarrow{+} y_{i+1}$ in S_2 ;
3. w_n, y_n are visible.

PROOF. We proceed by induction on the length n of the sequence s . If $n = 1$, trivially w, y are visible. Otherwise, w_i, y_i are hidden; then by persistence of S_1, S_2 the opposite link a' of w_i in S_1 and the opposite link b' of y_i in S_2 must be two different contracted links of R (otherwise one of S_1, S_2 should reduce to the empty slice). But then by contraction condition on R , there exists a negative link w' in S'_1 (resp. a negative link y' in S'_2) such that $a \xrightarrow{+} w'$ and $b' \xrightarrow{+} y'$ and w', y' form an additive pair in R ; we take w', y' as w_{i+1}, y_{i+1} in s . Finiteness of s is a consequence of finiteness and correctness of S_1, S_2 . \square

Lemma 5 Let S_1, S_2 be two persistent slices of a weakly correct J-proof structure R . Let a and b be a pair of positive links of S_1, S_2 respectively, such that a, b are two different occurrences of the same visible link c of R and $c \xrightarrow{+} d$ for some hidden link d of R (so $a \xrightarrow{+} d$ in S_1 , $b \xrightarrow{+} d$ in S_2). Then there exists a maximal finite sequence of links $s : \langle c_1, d_1 \rangle, \dots, \langle c_n, d_n \rangle$ where c_i is a positive link of R , d_i is an hidden negative link of R , such that:

- $c_1 = c$ (resp $d_1 = d$);
- $c_i \xrightarrow{+} d_i$ in R ;

- for each i , the opposite positive link of d_i in S_1 (resp. the opposite positive link of d_i in S_2) are both occurrences of c_{i+1} in S_1, S_2 , and d_i, d_{i+1} are not hidden by the same cut link c of R .

PROOF. The proof is a simple induction on the length n of s . Finiteness of s is a consequence of weak correctness and finiteness of R , and of the observation which follows. Given the final element $\langle c_n, d_n \rangle$ of s , by definition of s and by maximality of s it must be that d_n is hidden by a cut link m of R , and the opposite positive link a' of d_n in S_1 (resp. the opposite positive link b' of d_n in S_2) are either occurrences of two different contracted links of R , or they are occurrences of the same link c' of R . In the latter case, there is not any negative link d' hidden by a cut link $m' \neq m$ such that $c' \xrightarrow{+} d'$ in R in R .

□

Proposition 15 *Given a weakly correct J-proof structure R and a normal form $[R]$ of R , $[R]$ is a weakly correct J-proof structure.*

PROOF.

We first have to check that $[R]$ is a J-proof structure; by proposition 7, it is enough to check that the compatibility condition of definition 18 is respected.

Consider the multiplicative, cut-free J-proof structures S'_j, S'_k obtained by reducing two different persistent slices S_j, S_k of R . Now let us suppose that there is a positive link $a \in S'_j$ and a positive link $b \in S'_k$ such that a and b are siblings; a, b are clearly visible in S_j, S_k . We have to prove that there exist two negative links w' in S'_j , w'' in S'_k , such that $a \xrightarrow{+} w'$ in S'_j , $b \xrightarrow{+} w''$ in S'_k and w', w'' are siblings in S'_j, S'_k . Now we have the following cases:

1. Suppose that $a \xrightarrow{+} w_j$ in S_j and $b \xrightarrow{+} w_k$ in S_k , where w_j, w_k are two negative links which forms an additive pair in R ; then by lemma 4 there exist a finite sequence $s = \langle (w_{j_1}, w_{k_1}), \dots, (w_{j_n}, w_{k_n}) \rangle$ of additive pairs of R . We remark that every two different hidden elements $(w_{j_i}, w_{k_i}), (w_{j_l}, w_{k_l})$ of s are hidden by two different cut links of R (by weakly correctness of R); the final pair w_{j_n}, w_{k_n} of s is visible with w_{j_n} belonging to S_j (resp w_{k_n} belonging to S_k). By repeated applications of remark 4 and by theorem 5 it follows that in S'_j $a \xrightarrow{+} w_{j_n}$; similarly we can find that $b \xrightarrow{+} w_{k_n}$ in S'_k . We have two subcases:
 - (a) if w_{j_n}, w_{k_n} are not above any cut link of R , then w_{j_n}, w_{k_n} must be sibling in S'_j, S'_k , and they form an additive pair in $[R]$ too. But then S'_j, S'_k satisfy the compatibility condition.
 - (b) otherwise, there exists a negative hidden link d of R s.t. $c \xrightarrow{+} d$ for the parent c of w_{j_n}, w_{k_n} in R . Since both S_j, S_k contains c , we can apply lemma 5 to obtain a maximal sequence $\langle c_1, d_1 \rangle, \dots, \langle c_n, d_n \rangle$ of links of R , as described in the statement of lemma 5. Let us consider the last element $\langle c_n, d_n \rangle$ of the sequence. There are two possibilities:
 - i. If the opposite positive link d' of d_n in S_j and the opposite positive link d'' of d_n in S_k are two occurrences of the same link of R , then it is straightforward that after reduction the view of c in S'_j and the view of c in S'_k will stay the same, so w_{j_n}, w_{k_n} must be sibling in S'_j, S'_k , and they form an additive pair in $[R]$, q.e.d.
 - ii. Otherwise, the opposite positive link d' of d_n in S_j and the opposite positive link d'' of d_n in S_k are two different contracted links of R . In this case after reduction the view of c in S'_j and the view of c in S'_k may differ, so w_{j_n}, w_{k_n} could be not sibling in S'_j, S'_k . Then we search for two new negative sibling links: since d' and d'' are two different contracted links of R , then by contraction condition on R , there exists a negative link w''_j in S_j and a negative link w''_k in S_k such that w''_j, w''_k form an additive pair in R and $d' \xrightarrow{+} w''_j$ in S_j (resp. $d'' \xrightarrow{+} w''_k$ in S_k). By repeated applications of remark 4 and by theorem 5 it follows that in S'_j $a \xrightarrow{+} w''_j$ and that $b \xrightarrow{+} w''_k$ in S'_k . If w''_j, w''_k are not above any cut link of R , they must be sibling in S'_j, S'_k ; but then S'_j, S'_k satisfy the compatibility condition, q.e.d. Otherwise, we replicate on w''_j, w''_k the same procedure from the beginning of point 1. (replacing w_j, w_k with w''_j, w''_k and a, b with d', d''), iterating the applications of lemma 4 and lemma 5 until by finiteness of R we eventually found a proper visible additive pair of R .
2. Suppose that there is no additive pair w_1, w_2 of R such that $a \xrightarrow{+} w_1$ and $b \xrightarrow{+} w_2$, then a, b cannot be contracted in R (by contraction condition); so a, b must be two different occurrences of the same link c of R in S_j, S_k . Then we can apply lemma 5 to obtain a sequence $\langle c_1, d_1 \rangle, \dots, \langle c_n, d_n \rangle$ of link of R ,

as described in the statement of lemma 5. Such a sequence must terminate with an element $\langle c_n, d_n \rangle$ such that the opposite positive link d' of d_n in S_j and the opposite positive link d'' of d_n in S_k are two different contracted links of R (otherwise, a would be sharing equivalent to b in S'_j, S'_k , contradicting the assumption that a, b are sibling in S'_j, S'_k). But then by contraction condition on R there exists a negative link w'_j in S_j and a negative link w''_k in S_k such that w'_j, w''_k form an additive pair in R and $d' \xrightarrow{+} w'_j$ in S_j (resp. $d'' \xrightarrow{+} w''_k$ in S_k). By remark 4 and theorem 5 it follows that in S'_j $a \xrightarrow{+} w'_j$ and that $b \xrightarrow{+} w''_k$ in S'_k . Now we replicate on w'_j, w''_k the same procedure from the beginning of point 1. (replacing w_j, w_k with w'_j, w''_k and a, b with d', d''), iterating the applications of lemma 4 and lemma 5 until by finiteness of R we eventually found a proper visible additive pair of R .

Once proved that $[R]$ is a J-proof structure, it remains to check that it is weakly correct: switching acyclicity slice by slice is trivially preserved, due to theorem 3. Preservation of totality is a simple consequence of the following observation: take a non-persistent slice S of a J-proof structure R . Cut-reduction of S must end with a $+_{K \in \mathcal{N}} / -_{J \in \mathcal{N}}$, with $K \neq J$. Then by totality of R there must exist a slice S' of R such that:

- For all visible $\&$ -rules W of R , S' contains the same element w as S ;
- For all hidden $\&$ -rules W' of R , if an element w' of W' chosen by S is the “active” link of a $+_{K \in \mathcal{N}} / -_{J \in \mathcal{N}}$ step reducing S to the empty slice, we choose in S' another element w'' of W' which gives rise instead to a $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step during the reduction of S' (such an element exists by totality of R).

Such a slice S' is clearly persistent, so for every non-persistent slice S of R there exists a persistent slice S' of R which makes the same choices of elements as S on the visible $\&$ -rules of R . This is enough to conclude the totality of $[R]$. □

Theorem 6 (Existence of a normal form) *Given a weakly correct J-proof structure R , there exists a weakly correct J-proof structure R' such that $R' = [R]$.*

PROOF.

The proof is an easy consequence of theorem 4. □

Theorem 7 (Confluence) *If R, R', R'' are weakly correct J-proof structures, such that R', R'' are normal forms of R , then $R' = R''$.*

PROOF. Trivial, from theorem 5. □

6.3 Semantics

We focus now on semantics, following these steps: in subsection 6.3.1 we define the interpretation of a J-proof structure in pointed sets using the notion of *experiment* and we prove that semantics is stable under cut-reduction. Then in subsection 6.3.2, we prove that pointed sets semantics is *injective* with respect to total J-proof structures, that is for any two total J-proof structures R, R' , if R and R' have the same interpretation, then $R = R'$.

Experiments have been introduced by Girard in [Gir87], and extensively studied in [TdF00] by Tortora de Falco, as a way to compute the interpretation of a proof net without resorting to sequentialization. Semantic injectivity has been studied in the setting of linear logic mainly by Tortora (see [TdF00] and Pagani in [Pag06]; however, it is a traditional question in the denotational semantics of λ -calculus, and it is deeply related with the one of *syntactical separability* stated in the *Böhm theorem* for pure λ -calculus ([Boh68]); furthermore, syntactical separability is also one of the main properties in ludics (both of designs, see [Gir01], and more generally of L-nets, see [FM05]).

By now we will denote sets by A, B, C, \dots and elements of a set by a, b, c, \dots

A *pointed set* A^* is given by a set $A \cup \{0_{A^*}\}$ where 0_{A^*} is a distinguished object which does not belong to A ; this object is called the *point* of A^* .

The *pointed product* $A_1 \otimes \dots \otimes A_n$ of n sets A_1, \dots, A_n is the pointed set $A_1 \times \dots \times A_n \cup \{0_{A_1 \otimes \dots \otimes A_n}\}$ whose elements are the tuples of the cartesian product $A_1 \times \dots \times A_n$ together with a distinguished *fresh* object $0_{A_1 \otimes \dots \otimes A_n}$ which does not belong to $A_1 \times \dots \times A_n$.

By $A_1 \uplus \dots \uplus A_n$ we denote the disjoint union of A_1, \dots, A_n .

In general, a formula is interpreted by a disjoint union (indexed by ramifications) of pointed products of pointed sets.

More formally, the interpretation F of a formula F of HS is defined in the following way:

- a positive formula $\oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(N_i))$ is interpreted by the set $\uplus_{I \in \mathcal{N}}(\otimes_{i \in I}(\mathbf{N}^*_i))$;
- a negative formula $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$ is interpreted by the set $\uplus_{I \in \mathcal{N}}(\otimes_{i \in I}(\mathbf{P}^*_i))$.

6.3.1 Experiments

Given a J-proof structure R with conclusions C_1, \dots, C_n , we define the interpretation $\llbracket R \rrbracket$ of R as a subset of $\mathbf{C}^*_1 \otimes \dots \otimes \mathbf{C}^*_n$, which we define using the notion of **experiment**.

In the sequel, for simplicity's sake and when it is clear from the context, given a pointed set $\mathbf{A}^* = \uplus_{I \in \mathcal{N}}(\otimes_{i \in I}(\mathbf{A}^*_i))$ which interprets a formula $\&_{I \in \mathcal{N}}(\wp_{i \in I}(A_i))$ (resp. a formula $\oplus_{I \in \mathcal{N}}(\otimes_{i \in I}(A_i))$), we will denote the elements $\langle I, 0_{\otimes_{i \in I}(\mathbf{A}^*_i)} \rangle$ (for $I \in \mathcal{N}$) of \mathbf{A}^* , simply by $0_{\mathbf{A}^*}$ or by 0 .

Definition 29 (Experiments) Let S be a multiplicative J-proof structure and e an application associating with every edge a of type A of S an element of \mathbf{A}^* ; e is an **experiment** of S when the following conditions hold:

- if x, y are premises of a cut link with premises x and y , then $e(x) = e(y)$.
- Suppose x is the conclusion of a negative link $-_{I \in \mathcal{N}}$ with premises x_1 of type P_1, \dots, x_n of type P_n . If for all $i \in \{1, \dots, n\}$ $e(x_i) = 0_{\mathbf{P}^*_i}$, then $e(x) = \langle I, \langle 0_{\mathbf{P}^*_1}, \dots, 0_{\mathbf{P}^*_n} \rangle \rangle$ or $e(x) = \langle I, 0_{\mathbf{P}^*_1 \otimes \dots \otimes \mathbf{P}^*_n} \rangle$. Otherwise $e(x) = \langle I, \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle \rangle$, where $e(x_1) = \mathbf{a}_1, \dots, e(x_n) = \mathbf{a}_n$.
- Suppose x is the conclusion of a positive link $+_{I \in \mathcal{N}}$ with premises x_1 of type N_1, \dots, x_n of type N_n . If for all $i \in \{1, \dots, n\}$ $e(x_i) = 0_{\mathbf{N}^*_i}$, then $e(x) = \langle I, \langle 0_{\mathbf{N}^*_1}, \dots, 0_{\mathbf{N}^*_n} \rangle \rangle$ or $e(x) = \langle I, 0_{\mathbf{N}^*_1 \otimes \dots \otimes \mathbf{N}^*_n} \rangle$. Otherwise $e(x) = \langle I, \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle \rangle$, where $e(x_1) = \mathbf{a}_1, \dots, e(x_n) = \mathbf{a}_n$.
- If a is a positive link with conclusion x of type A and b is a negative link with conclusion y of type B , and there is a jump between b and a , then if $e(x) \neq 0_{\mathbf{A}^*}$, $e(y) \neq 0_{\mathbf{B}^*}$.

If x_1, \dots, x_n of type respectively A_1, \dots, A_n are the conclusions of S and e is an experiment of S such that $e(x_i) = \mathbf{a}_i$ then we shall say that $\langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle$ is the *conclusion* or the *result* of the experiment e of S , and we will denote it by $|e|$. The set of the results of all experiments on S is the interpretation $\llbracket S \rrbracket$ of S ; in case S is empty, then its interpretation is the empty set.

Let R be a total J-proof structure and $\{S_1, \dots, S_n\}$ the set of slices of R ; the interpretation $\llbracket R \rrbracket$ of R is the union of $\llbracket S_1 \rrbracket, \dots, \llbracket S_n \rrbracket$.

In figure 27, 28 we provide two examples of experiments on the slices of the J-proof structure in fig. 11.

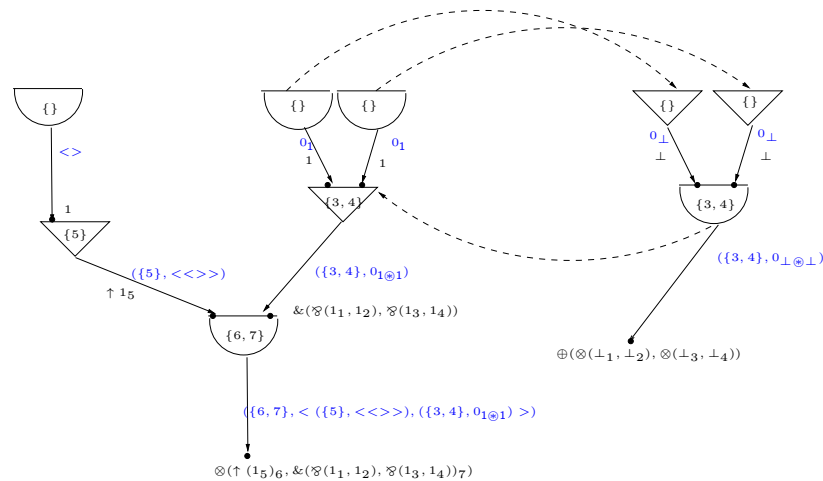


Figure 27: example of experiment

Remark 5 We can consider experiments as a way to travel bottom-up on a J-proof structure R looking only at its semantics; starting from an element of $\llbracket R \rrbracket$, then it is possible to rebuild bottom-up R itself. Observe that the “0” points introduced in pointed sets semantics, can be used in experiments as a sort of semantical “traffic light”, forbidding bottom-up the access to a whole portion of R ; in this way it becomes possible to recover information about the sequential order contained in R .

Proposition 16 If S is a correct multiplicative J-proof structure, and $S \rightsquigarrow S'$, then $\llbracket S \rrbracket = \llbracket S' \rrbracket$.

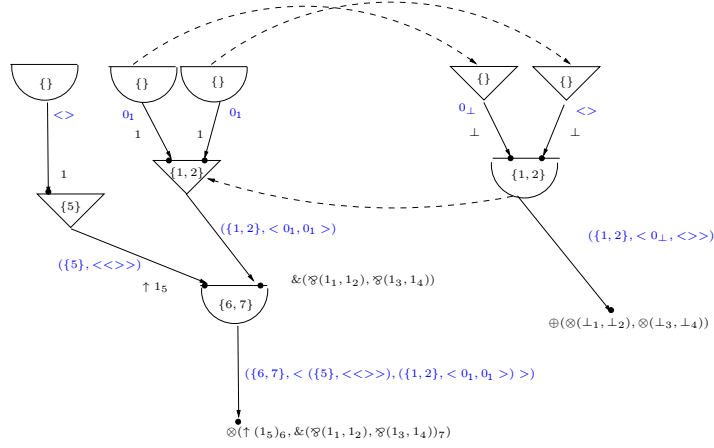


Figure 28: example of experiment

PROOF. We must show that for every experiment e on S' , there is an experiment e' of S with the same result.

Having as reference fig. 25 let us suppose that $S \rightsquigarrow S'$ with a $+_{I \in \mathcal{N}} / -_{I \in \mathcal{N}}$ step reducing a cut in S between a positive link a with conclusion x and premises x_1 of type N_1, \dots, x_n of type N_n , and a negative link b with conclusion y and premises y_1 of type P_1, \dots, y_n of type P_n ; we denote the edges of type P_1, \dots, P_n (resp. N_1, \dots, N_n) by y_1, \dots, y_n (resp. x_1, \dots, x_n). Suppose that a positive link c of typed conclusion z jumps on b , and that a jumps on a negative link d with conclusion w .

The delicate part is the one dealing with jumps, the rest of the proof trivially following from the preservation of relational semantics under cut reduction (see for instance [Gir87]).

Let us consider an experiment e of S' . The relevant cases to check are the following:

- if in S' $e(w) = 0$, then $e(z) = 0$, $e(y_1) = 0, \dots, e(y_n) = 0$ by definition of experiment. We can then build an experiment e' of S with the same values on the corresponding edges of S by assigning $e'(x) = 0$ and $e'(y) = 0$;
- suppose in S' $e(w) \neq 0$, $e(z) \neq 0$ and $e(y_1) = 0, \dots, e(y_n) = 0$. We can then build an experiment e' of S with the same values on the corresponding edges of S by assigning $e'(y) = \langle 0_{P_1}, \dots, 0_{P_n} \rangle$ and $e'(x) = \langle 0_{N^*_1}, \dots, 0_{N^*_n} \rangle$.

□

Proposition 17 *If R, R' are weakly correct J-proof structures, such that $R' = [R]$, then $\llbracket R \rrbracket = \llbracket R' \rrbracket$.*

PROOF. The proof is a consequence of proposition 16.

□

6.3.2 Injectivity

We first prove injectivity with respect to multiplicative J-proof structures, then we extend the result to the general case.

Multiplicative J-proof structures and injectivity.

Definition 30 (Relational result) *Let S be a multiplicative J-proof structure and $|e|$ the result of an experiment on S ; $|e|$ is **relational** if it does not contain any occurrence of the element 0.*

The set of relational results of experiments on a multiplicative J-proof structure S is called the **relational part** of $\llbracket S \rrbracket$; we will denote it by $\llbracket S \rrbracket^{Rel}$.

Remark 6 *Given two multiplicative J-proof structures S, S' , if $\llbracket S \rrbracket = \llbracket S' \rrbracket$ then $\llbracket S \rrbracket^{Rel} = \llbracket S' \rrbracket^{Rel}$.*

Remark 7 *Let S be a multiplicative J-proof structure and e, e' be two experiments of S . If $|e| = |e'|$, then $e = e'$; in other words an experiment is completely determined by its result.*

Remark 8 Given a multiplicative J-proof structure S , there exists a unique relational experiment e on S . In fact the relational part $\llbracket S \rrbracket^{Rel}$ of $\llbracket S \rrbracket$ is a singleton, since the procedure of building an experiment is completely deterministic, provided it never assign the label 0 to any edges.

Given a slice S , we denote by S^- the graph obtained by erasing all the jumps of S .

Lemma 6 Let S, S' be two multiplicative cut-free J-proof structures with the same conclusions; $\llbracket S \rrbracket^{Rel} = \llbracket S' \rrbracket^{Rel}$ iff $S^- = S'^-$.

PROOF.

The proof can be obtained by a straightforward adaptation of the proof of injectivity of relational semantics for *MLL* proof nets given by Tortora de Falco in [TdF03a]. Such a property implies that the unique relational experiment of S (resp. S') allows to completely reconstruct S (resp. S'), up to jumps. □

Lemma 7 Given a multiplicative J-proof structure S , a positive link a with typed conclusion x and a negative link b with typed conclusion y , $b \prec_S a$ (that is, b precedes a in the order associated with S as a d.a.g.) iff for all experiments e of R , $e(x) \neq 0 \Rightarrow e(y) \neq 0$.

PROOF. The proof is an easy consequence of definition of experiment. □

Theorem 8 (Injectivity) Let S and S' be two cut-free multiplicative J-proof structures with the same conclusions. $\llbracket S \rrbracket = \llbracket S' \rrbracket$ iff $S = S'$.

PROOF.

The right to left direction is trivial.

Concerning the left to right direction, $\llbracket S \rrbracket = \llbracket S' \rrbracket$, so $\llbracket S \rrbracket^{Rel} = \llbracket S' \rrbracket^{Rel}$.

Since $\llbracket S \rrbracket^{Rel} = \llbracket S' \rrbracket^{Rel}$, by lemma 6, $S^- = S'^-$.

Now, by remark 7, given an element γ of $\llbracket S \rrbracket$ (resp. $\llbracket S' \rrbracket$) there exists a unique experiment e of S (resp. of S') such that $|e| = \gamma$.

Now, we build from S^- (resp. S'^-) a multiplicative J-proof structure S^J (resp. S'^J) in the following way; for any positive link a of typed conclusion x and for any negative link b of typed conclusion y , we check that for every element γ of $\llbracket S \rrbracket$, given the unique experiment e of S (resp. S') induced by γ , $e(x) \neq 0 \Rightarrow e(y) \neq 0$; if it is the case and a is not above b in S^- (resp. in S'^-) we make a jump on b . Using lemma 7, we can easily conclude that $S^J = S$, and $S'^J = S'$ (up to transitive jumps); since $\llbracket S \rrbracket = \llbracket S' \rrbracket$ and $S^- = S'^-$, $S = S'$. □

Proposition 18 Let S be a multiplicative cut free J-proof structure and S' a multiplicative cut-free J-proof structure with the same conclusions as S . S' can be obtained by adding jumps on S iff $\llbracket S \rrbracket^{Rel} = \llbracket S' \rrbracket^{Rel}$ and $\llbracket S' \rrbracket \subseteq \llbracket S \rrbracket$.

PROOF.

The left to right direction is trivial.

Concerning the right to left direction, by lemma 6, $S^- = S'^-$. Since $\llbracket S' \rrbracket \subseteq \llbracket S \rrbracket$, using lemma 7 we can say that all the jumps of S are jumps of S' . In order to obtain S , we just add to S all the jumps of S' which are not jumps of S ; in this way we obtain a J-proof structure $S^J = S'$. □

Injectivity, general case. Given a J-proof structure R with conclusions Γ , an *index assignment* of R is a function ϕ associating with any formula of type $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$ occurring in Γ a $J \in \mathcal{N}$.

It is easy to check that if R is total, to any index assignment ϕ there is a unique slice S^ϕ of R , and to each slice of R there is (at least one) index assignment ϕ .

Let us consider an element $\langle J, \delta \rangle$ of $\uplus_{I \in \mathcal{N}}(\otimes_{i \in I}(P^*_i))$, interpretation of a formula $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$, and an element γ of the interpretation $\llbracket R \rrbracket$ of a total J-proof structure R with conclusions Γ ; we say that γ is *compatible* with an index assignment ϕ on Γ iff for any occurrence of $\langle J, \delta \rangle$ in γ , on the corresponding occurrence of $\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))$ in Γ , $\phi(\&_{I \in \mathcal{N}}(\wp_{i \in I}(P_i))) = J$.

Proposition 19 *Given a total J-proof structure R , an element γ of $\llbracket R \rrbracket$ is compatible with an index assignment ϕ , iff γ is the result of an experiment on S^ϕ .*

PROOF. Suppose γ is not the result of an experiment on S^ϕ ; then it is the result of an experiment on another slice S' of R , which differs from S^ϕ for at least one component of a $\&$ rule. But then it is easy to observe that γ cannot be compatible with S' . The other direction is trivial. \square

Proposition 20 *Given a total J-proof structure R and an index assignment ϕ , $\llbracket S^\phi \rrbracket = \{\gamma \in \llbracket R \rrbracket \mid \gamma \text{ is compatible with } \phi\}$.*

PROOF. Easy consequence of proposition 19. \square

Theorem 9 (Injectivity) *Let R and R' be two cut-free total J-proof structures with the same conclusions Γ . $\llbracket R \rrbracket = \llbracket R' \rrbracket$ iff $R = R'$.*

PROOF.

The right to left direction is trivial.

Concerning the left to right direction, let us take the slice S^ϕ of R corresponding to the index assignment ϕ of Γ , and suppose S^ϕ is not a slice of R' . By proposition 20 $\llbracket S^\phi \rrbracket = \{\gamma \in \llbracket R \rrbracket \mid \gamma \text{ is compatible with } \phi\}$. Since $\llbracket R \rrbracket = \llbracket R' \rrbracket$, $\{\gamma \in \llbracket R \rrbracket \mid \gamma \text{ is compatible with } \phi\} = \{\gamma \in \llbracket R' \rrbracket \mid \gamma \text{ is compatible with } \phi\}$; then for the unique slice S'^ϕ of R' which corresponds to ϕ by proposition 20 $\llbracket S'^\phi \rrbracket = \{\gamma \in \llbracket R' \rrbracket \mid \gamma \text{ is compatible with } \phi\}$, and $\llbracket S'^\phi \rrbracket = \llbracket S^\phi \rrbracket$; but then by theorem 8 $S'^\phi = S^\phi$ so S^ϕ is a slice of R' , contradicting our hypothesis that S^ϕ was not a slice of R' . \square

Proposition 21 *Let R be a cut-free total J-proof structure and R' a cut-free total J-proof structure with the same conclusions as R . R' is obtained by adding jumps on R iff $\llbracket R \rrbracket^{Rel} = \llbracket R' \rrbracket^{Rel}$ and $\llbracket R' \rrbracket \subseteq \llbracket R \rrbracket$.*

PROOF.

The left to right direction is trivial.

Concerning the right to left direction, we prove that for every slice S of R there exists a slice S' of R' such that S' is obtained from S by adding jumps.

Let us take the slice S^ϕ of R corresponding to the index assignment ϕ of Γ , and suppose that for no slices S' of R' , S' is obtained by adding jumps on S^ϕ . By proposition 20 $\llbracket S^\phi \rrbracket = \{\gamma \in \llbracket R \rrbracket \mid \gamma \text{ is compatible with } \phi\}$, and $\llbracket S^\phi \rrbracket^{Rel} = \{\gamma \in \llbracket R \rrbracket^{Rel} \mid \gamma \text{ is compatible with } \phi\}$.

Since $\llbracket R \rrbracket^{Rel} = \llbracket R' \rrbracket^{Rel}$, then $\{\gamma \in \llbracket R \rrbracket^{Rel} \mid \gamma \text{ is compatible with } \phi\} = \{\gamma \in \llbracket R' \rrbracket^{Rel} \mid \gamma \text{ is compatible with } \phi\}$ and since $\llbracket R \rrbracket \supseteq \llbracket R' \rrbracket$, $\{\gamma \in \llbracket R \rrbracket \mid \gamma \text{ is compatible with } \phi\} \supseteq \{\gamma \in \llbracket R' \rrbracket \mid \gamma \text{ is compatible with } \phi\}$; then for the unique slice S'^ϕ of R' which corresponds to ϕ by proposition 20 $\llbracket S'^\phi \rrbracket = \{\gamma \in \llbracket R' \rrbracket \mid \gamma \text{ is compatible with } \phi\}$, and $\llbracket S'^\phi \rrbracket^{Rel} = \{\gamma \in \llbracket R' \rrbracket^{Rel} \mid \gamma \text{ is compatible with } \phi\}$; so $\llbracket S'^\phi \rrbracket^{Rel} = \llbracket S^\phi \rrbracket^{Rel}$, and $\llbracket S'^\phi \rrbracket \subseteq \llbracket S^\phi \rrbracket$; but then by proposition 18 S'^ϕ is obtained by adding jumps on S^ϕ , contradicting our hypothesis that for no slices S' of R' , S' is obtained by adding jumps on S^ϕ .

Similarly, we can prove that for every slice S' of R' there exists a slice S of R such that S' is obtained by adding jumps on S ; but then it is immediate that R' can be obtained by adding jumps on R . \square

6.4 Correctness criterion is stable under reduction

Definition 31 *Given two J-proof structures R_1, R_2 with conclusions respectively Γ, P and Δ, P^\perp , the **composition** of R_1, R_2 is the J-proof structure with conclusion Γ, Δ obtained by connecting R_1, R_2 through a cut-link with premises P, P^\perp .*

Remark 9 *If R_1, R_2 are sequentializable then their composition R is sequentializable.*

Proposition 22 *Let R be the composition of two arborescent, weakly correct J-proof structures R_1, R_2 ; then the normal form of R is arborescent.*

PROOF. The proof is a simple consequence of Proposition 14 and of the fact that the operation of superposition of J-proof structures preserves arborescence (as it can be easily verified). \square

Proposition 23 *Let R_1, R_2 be two sequentializable cut-free J-proof structures, and R be their composition. The normal form $[R]$ of R is sequentializable.*

PROOF.

Since R_1, R_2 are sequentializable they are J-proof nets by Theorem 1. Then we consider two saturations R'_1, R'_2 of resp. R_1, R_2 ; they are arborescent by the arborisation lemma. We take their composition R' . Now we reduce R to its normal form R_0 and R' to its normal form R'_0 . We show that the following diagram commutes (where vertical edges represents the addition of jumps, and horizontal edges stands for normalization).

$$\begin{array}{ccc} R & \longrightarrow & R_0 \\ \downarrow & & \downarrow \\ R' & \longrightarrow & R'_0 \end{array}$$

Now by proposition 21, $\llbracket R' \rrbracket \subseteq \llbracket R \rrbracket$; by Proposition 17 $\llbracket R \rrbracket = \llbracket R_0 \rrbracket$ and $\llbracket R' \rrbracket = \llbracket R'_0 \rrbracket$, so $\llbracket R'_0 \rrbracket \subseteq \llbracket R_0 \rrbracket$. By Proposition 21 R'_0 is obtained by adding jumps on R_0 and by Proposition 22 R'_0 is arborescent; So R'_0 is sequentializable by Proposition 9 and R_0 is sequentializable by Proposition 11.

□

Theorem 10 *Let R_1, R_2 be two cut-free J-proof nets, and R be their composition. The normal form $[R]$ of R is a J-proof net.*

PROOF. A simple consequence of Proposition 23 and Theorem 1.

□

7 Are J-proof nets canonical objects?

So far, we have defined a syntax placed somehow in-between focusing, hypersequentialized proofs and proof nets, and we proved that the objects of such syntax have a good computational behaviour. We also proved that we can characterize hypersequentialized proofs in our setting as *saturated* J-proof nets, that is objects containing a maximal amount of sequential information. A natural question is then the following: if J-proof nets with “maximal” sequentiality correspond to *HS* proofs, what is the correspondent of a J-proof net with “minimal” sequentiality?

In this section we will try to answer to this question. First, in subsection 7.1 we will precisely define the notion of *parallel* J-proof net and we will consider if such an object can be considered *canonical*; then in subsection 7.2 we will refine our analysis of parallel J-proof nets through the notion of *monomiality*, introduced by Girard in [Gir96]. Finally, in subsection 7.3 we will discuss the relation between J-proof nets and *multifocalization* (see [CMS08]). For a thorough comparison between J-proof nets and polarized proof nets (see [Lau02]) we refer instead to [Gia].

7.1 Parallel J-proof nets

Let us consider the following order relation on J-proof nets: given two J-proof nets R, R' we say that R is *more sequential* than R' (denoted by $R' \ll R$) if R can be obtained from R' by adding jumps.

It is easy to see that the relation \ll is a partial order on the set of J-proof nets, whose maximal elements are saturated J-proof nets; we call the minimal elements of the order *parallel* J-proof nets, defined as follows:

Definition 32 *A parallel J-proof net is a J-proof net R such that, given any other J-proof net R' , cannot be obtained by adding jumps on R' .*

Given two J-proof nets R_1, R_2 , we say that R_1 is *jump-permutable* to R_2 (denoted $R_1 \approx R_2$) iff, for some parallel J-proof net R , $R \ll R_1$ and $R \ll R_2$; informally $R_1 \approx R_2$ if R_2 can be obtained from R_1 by “permutation of jumps”, namely erasing jumps from R_1 to get R and then adding jumps to R to get R_2 .

Given a parallel J-proof net R let us consider the set of J-proof nets $E = \{R' \mid R \ll R'\}$, that is the set of all J-proof nets that can be obtained from R by adding jumps.

We want to know if :

1. the relation \approx is an equivalence relation;
2. E can be considered an “equivalence class” for \approx ;

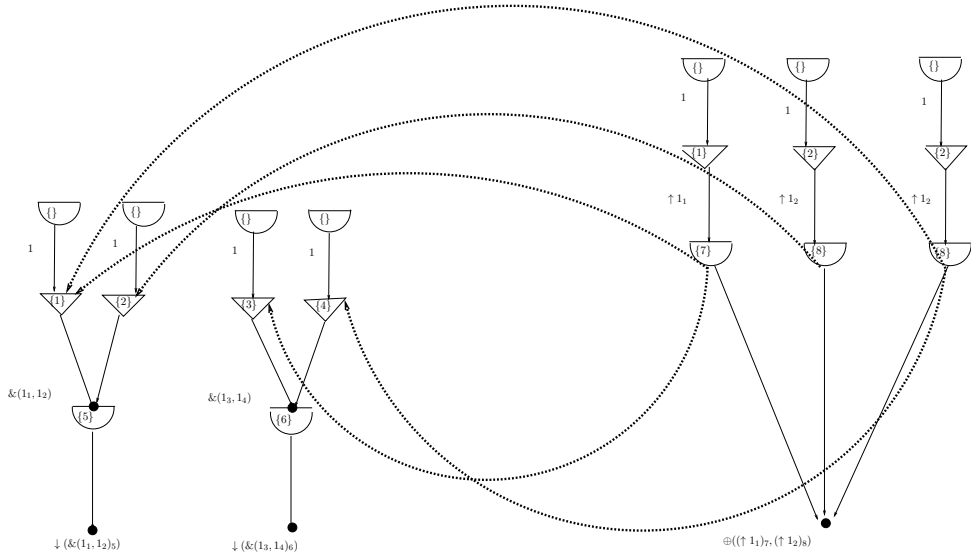


Figure 29: a parallel J-proof net R_1

3. R can be considered a “canonical” representative of the J-proof nets in E .

A minimal requirement to satisfy these conditions would be the following: given any two parallel J-proof nets R_1, R_2 , there must not be a J-proof net R_3 such that $R_1 \ll R_3$ and $R_2 \ll R_3$. Unfortunately, this is not the case, as we can see in the following example.

Let us consider the parallel J-proof nets R_1, R_2 resp. in fig. 29, 30.

Both R_1, R_2 are parallel: we cannot remove any jump from them without violating the contraction condition.

The saturated J-proof net R_3 in fig. 31 can be obtained by adding jumps both on R_1, R_2 , so $R_1 \ll R_3$ and $R_2 \ll R_3$, but R_1 and R_2 are not comparable with respect to \ll . Moreover, since R_3 is saturated, it corresponds to an *HS* proof; so we can say that parallel J-proof nets fail to provide canonical representatives w.r.t *HS* proofs.

7.2 J-proof nets and monomiality

Let us try to clarify the reason behind the presence of two different J-proof nets representing the same *HS* proof with an example. In this example we will resort to boolean weights, a tool first introduced by Girard in [Gir96] to study the dependency relation on the links of a proof net induced by the presence of additives.

Let us consider the two J-proof nets R_1, R_2 above. We can characterize the dependencies between the links and the $\&$ -rules in R_1, R_2 in the following way: first we assign a boolean variable, p (resp q) to the leftmost (resp. rightmost) $\&$ -rule.

Then we label the links with boolean weights, in such a way that if a link a depends from the left component of the leftmost (resp. rightmost) $\&$ -rule then its weight contains the boolean variable p (resp. q); if instead a depends from the right component of the leftmost (resp. rightmost) $\&$ -rule then its weight contains the negation \bar{p} (resp. \bar{q}) of the boolean variable p (resp. q). Assigning a value among 1,0 to each of the boolean variable p, q and erasing the links whose weight is equal to 0 yields a slice of R_1 (resp. R_2). In fig 32, 33 we represent the J-proof nets R_1, R_2 labelled with boolean weights: we omit jumps, since we can retrieve them by looking at dependencies induced by the weights of links.

We can observe that all weights labelling R_1, R_2 are *monomials*⁴. Now we know from [HVG03] that expressing additives dependencies through monomials it is not enough to provide a unique canonical representative for sequent calculus proofs equivalent modulo permutation of rules; this explains why there cannot be a J-proof net “quotienting” R_1, R_2 . Such a J-proof net should satisfy the non-monomial labelling provided in fig 34; but if we try to add the jumps in a way to satisfy the dependency relations induced by the non-monomial weights, we get the structure R represented in fig. 29 which does not satisfy the contraction condition.

It is worth noticing that each evaluation of the boolean weights in R , produce nevertheless a slice which is actually a multiplicative J-proof structure. An interesting point to develop (which we postpone to future work)

⁴Given a proper weight algebra (which takes into account the possible presence of n -ary $\&$ -rules), this observation is easily generalized to all J-proof structures; we left the details to the reader.

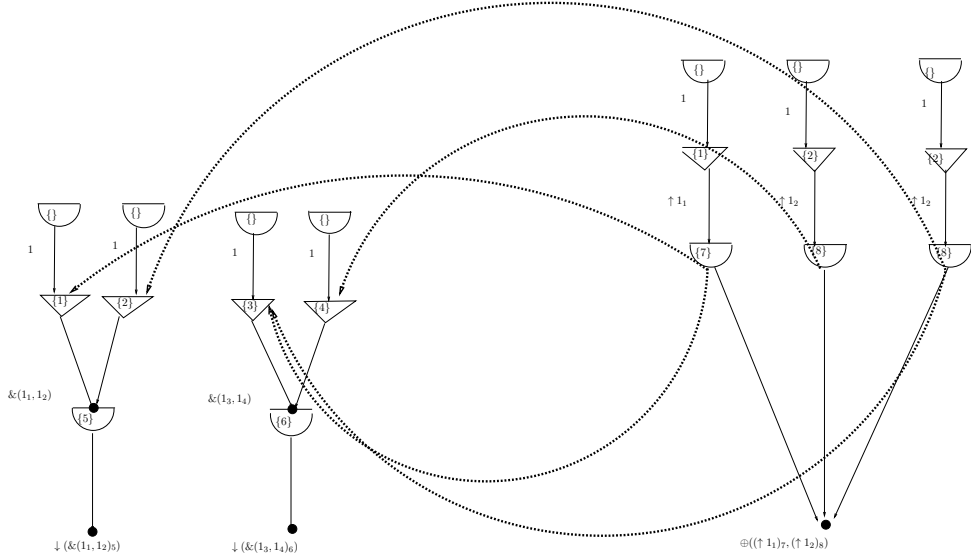


Figure 30: a parallel J-proof net R_2

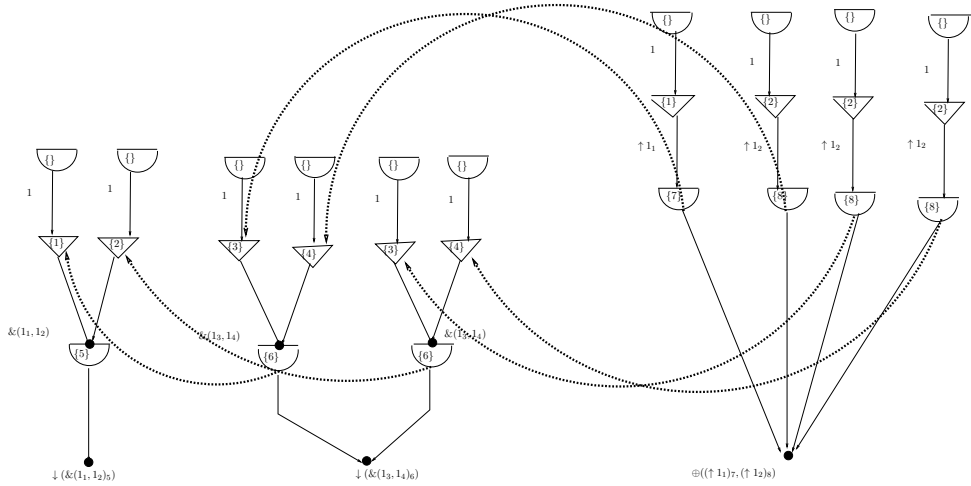


Figure 31: a saturated J-proof net R_3 , which is obtained by adding jumps on R_1, R_2

should be to reformulate the contraction condition, in order to include also non-monomial structures (in the spirit of Hughes and Van Glabbeek work, see [HVG03].)

7.3 J-proof nets and multifocusing

In [CMS08], Miller, Saurin and Chaudhuri have introduced a generalization of $MALL^{foc}$, the multiplicative/additive fragment of Andreoli's focusing calculus ; in such a calculus it is possible to focus on more than one formula at a time, due to a rule called *multifocalization*. Such a rule has the following shape (we refer to [CMS08] for a detailed description of the calculus):

$$\frac{\vdash \Gamma \Downarrow \Delta}{\vdash \Gamma, \Delta \Uparrow} [MF]$$

where Δ contains only positive formulas.

In the multifocusing calculus, as in $MALL^{foc}$, the sequent has a *stoup*, (the part of the sequent after \Downarrow) which contains the formulas to be decomposed (bottom up) during focalization; but while in $MALL^{foc}$ the stoup contains a single formula, in the multifocusing calculus we may focus on more than one formula at a time. All the formulas under focus will then be decomposed until only negative formulas remain, which will be decomposed in the next negative phase until we reach their positive components; a new set of positive formulas to focus on then will be chosen and so on.

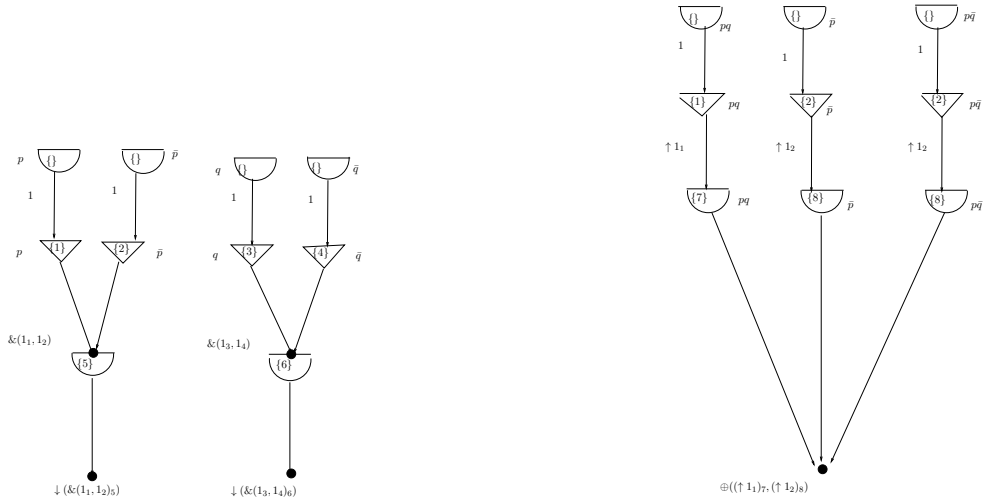


Figure 32: R_1 labelled with boolean weights

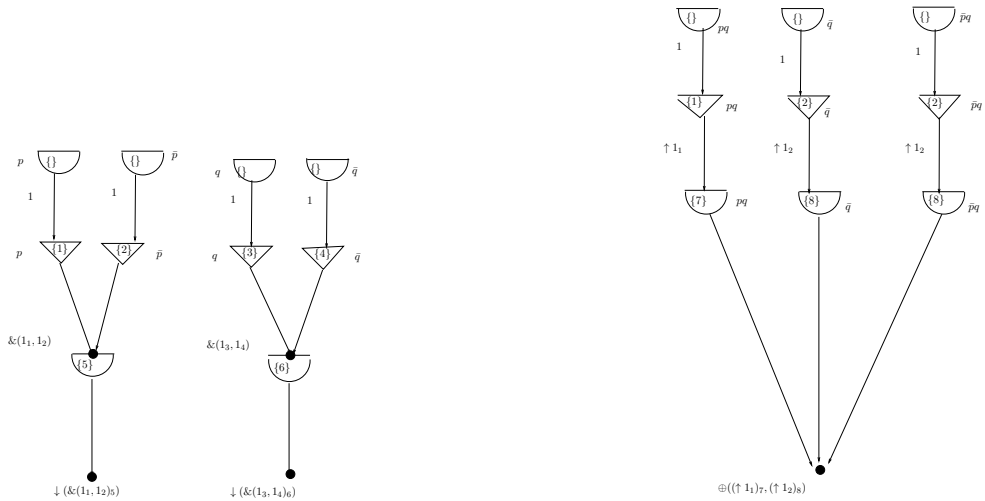


Figure 33: R_2 labelled with boolean weights

An interesting feature of the multifocusing calculus is that it allows to graduate the “level” of focalization of a proof, depending on how many formulas are included in the stoup on an application of the MF rule. In the following we will show an example of how J-proof nets allow to capture such a “multifocalization” property.

For simplicity’s sake we will restrict our examples to the multiplicative case; nevertheless, our argument is perfectly sound also in presence of additives.

Let us consider the following two focusing proofs, denoted respectively π_1, π_2 (to stress the order of application of rules, we mark some of them with an integer), where the focus of the MF rule is a single formula:

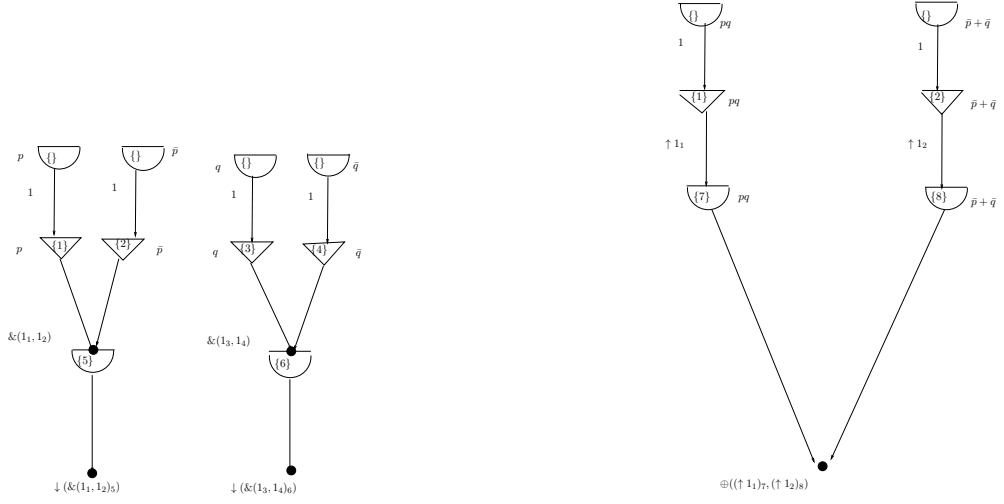


Figure 34: Non-monomial labelling

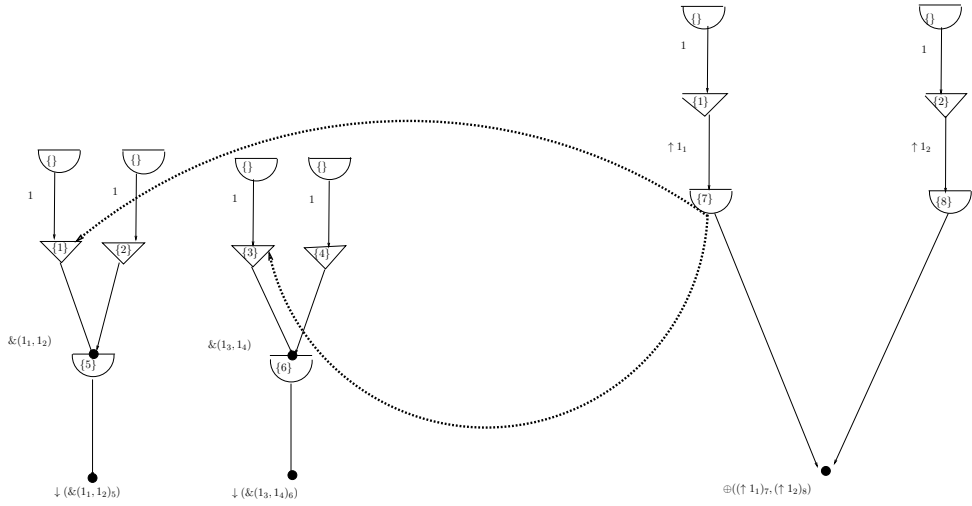
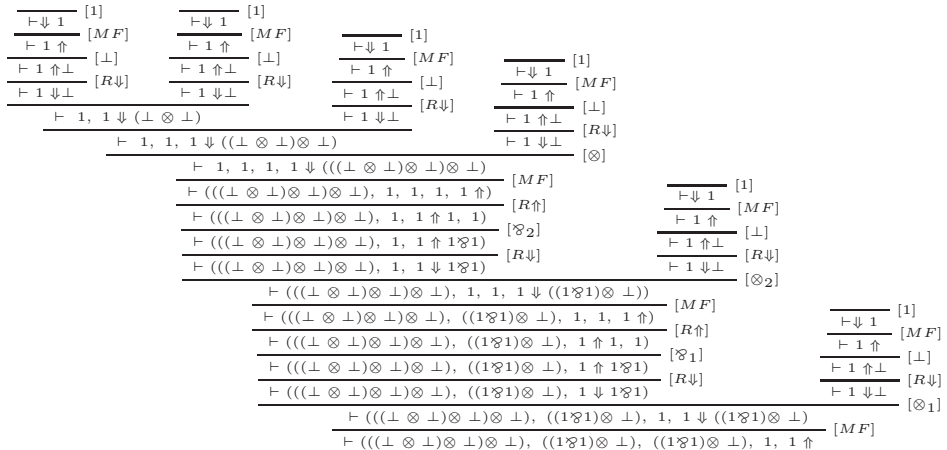


Figure 35: jumps induced by the non monomial labelling



$$\begin{array}{c}
\frac{}{\vdash \downarrow 1} [1] \quad \frac{}{\vdash \downarrow 1} [1] \quad \frac{}{\vdash \downarrow 1} [1] \quad \frac{}{\vdash \downarrow 1} [1] \\
\frac{}{\vdash \uparrow 1} [MF] \quad \frac{}{\vdash \uparrow 1} [MF] \quad \frac{}{\vdash \uparrow 1} [MF] \quad \frac{}{\vdash \uparrow 1} [MF] \\
\frac{}{\vdash \uparrow \perp} [\perp] \quad \frac{}{\vdash \uparrow \perp} [\perp] \quad \frac{}{\vdash \uparrow \perp} [\perp] \quad \frac{}{\vdash \uparrow \perp} [\perp] \\
\frac{}{\vdash \uparrow \downarrow \perp} [R\downarrow] \quad \frac{}{\vdash \uparrow \downarrow \perp} [R\downarrow] \quad \frac{}{\vdash \uparrow \downarrow \perp} [R\downarrow] \quad \frac{}{\vdash \uparrow \downarrow \perp} [R\downarrow] \\
\hline
\vdash 1, 1 \downarrow (\perp \otimes \perp) \\
\hline
\vdash 1, 1, 1 \downarrow ((\perp \otimes \perp) \otimes \perp) \\
\hline
\vdash 1, 1, 1, 1 \downarrow (((\perp \otimes \perp) \otimes \perp) \otimes \perp) [MF] \\
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp) \uparrow 1, 1, 1, 1 [R\uparrow] \\
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp) \uparrow (1\wp 1), 1, 1 [\wp 1] \\
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp) \uparrow (1\wp 1), (1\wp 1) [\wp 2] \\
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp) \downarrow (1\wp 1), (1\wp 1) [R\downarrow] \\
\hline
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp), 1 \downarrow ((1\wp 1) \otimes \perp), 1\wp 1 \\
\hline
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp), 1, 1 \downarrow ((1\wp 1) \otimes \perp), ((1\wp 1) \otimes \perp) \\
\hline
\vdash (((\perp \otimes \perp) \otimes \perp) \otimes \perp), ((1\wp 1) \otimes \perp), ((1\wp 1) \otimes \perp), 1, 1 \uparrow [MF]
\end{array}$$

Such a multifocalized proof π_3 directly corresponds to the J-proof net R_3 in fig. 38. In R_3 too, the minimal bipoles coincide exactly with the splitting ones, and in this case they are more than one: we can then sequentialize R_3 by removing at the same time both bipoles, decomposing them simultaneously in the stoup. The jumps originating from the middle \otimes link tell that it must wait for both (\otimes_1, \wp_1) , (\otimes_2, \wp_2) to be decomposed in order to enter the stoup.

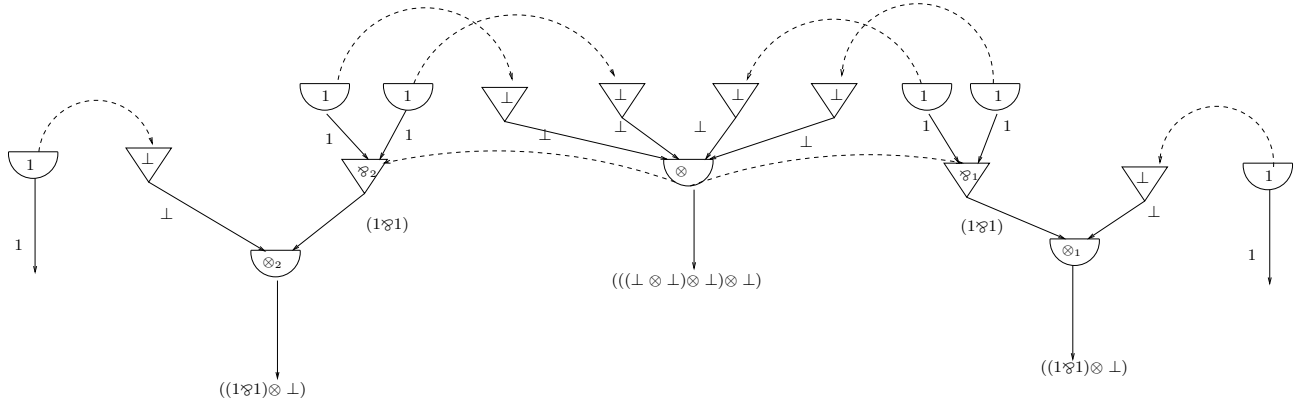


Figure 38:

As a future research direction, we plan to deepen the study of the relation between multifocalization and J-proof nets, especially the connection between parallel J-proof nets and maximal multifocusing proofs.

References

- [AM99] Samson Abramski and Paul-Andre' Mellies. Concurrent games and full completeness. In *LICS'99*, pages 431–442. IEEE, 1999.
- [AMJ00] Samson Abramsky, Pasquale Malacaria, and Radha Jagadeesan. Full abstraction for PCF. *I&C*, 163(2):409–470, 2000.
- [And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. Logic Comput.*, 2(3):297–347, 1992.
- [Boh68] C. Bohm. Alcune proprietà delle forme beta-eta-normali del lambda-kappa-calcolo. Pubblicazioni dell'I.A.C. 696, I.A.C., Roma, 1968.
- [Bou04] Pierre Boudes. Unifying static and dynamic denotational semantics. Preprint, Institut de Mathématiques de Luminy, 2004.
- [CF] P.-L. Curien and C. Faggian. An approach to innocent strategies as graphs. submitted to journal.
- [CF05] P.-L. Curien and C. Faggian. L-nets, strategies and proof-nets. In *CSL 05 (Computer Science Logic)*, LNCS. Springer, 2005.
- [CMS08] Kaustuv Chaudhuri, Dale Miller, and Alexis Saurin. Canonical sequent proofs via multi-focusing. In *IFIP TCS*, pages 383–396, 2008.

- [DGF06] Paolo Di Giamberardino and Claudia Faggian. Jump from parallel to sequential proof: multiplicatives. volume 4207 of *Lecture Notes in Computer Science*, pages 319–333. Springer Berlin / Heidelberg, 2006.
- [DGF08] Paolo Di Giamberardino and Claudia Faggian. Proof nets sequentialisation in multiplicative linear logic. *Annals of Pure and Applied Logic*, pages 173–182, 2008.
- [Fag02] C. Faggian. Travelling on designs: ludics dynamics. In *CSL'02 (Computer Science Logic)*, volume 2471 of *LNCS*. Springer Verlag, 2002.
- [FM05] C. Faggian and F. Maurel. Ludics nets, a game model of concurrent interaction. In *LICS'05*. IEEE, 2005.
- [Gia] Paolo Di Giamberardino. Jump from parallel to sequential proofs: exponentials. To appear in *Mathematical Structures In Computer Science*.
- [Gir87] Jean-Yves Girard. Linear logic. *TCS*, 50:1–102, 1987.
- [Gir91] J.-Y. Girard. Quantifiers in linear logic II. *Nuovi problemi della Logica e della Filosofia della scienza*, 1991.
- [Gir96] J.-Y. Girard. Proof-nets : the parallel syntax for proof-theory. In *Logic and Algebra*, volume 180 of *Lecture Notes in Pure and Appl. Math.*, pages 97–124. 1996.
- [Gir99] Jean-Yves Girard. On the meaning of logical rules. I: Syntax versus semantics. In *Computational logic (Marktoberdorf, 1997)*, volume 165 of *NATO Adv. Sci. Inst. Ser. F Comput. Systems Sci.*, pages 215–272. Springer, Berlin, 1999.
- [Gir00] Jean-Yves Girard. On the meaning of logical rules II: multiplicative/additive case. In *Foundation of Secure Computation*, NATO series F 175, pages 183–212. IOS Press, 2000.
- [Gir01] Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *MSCS*, 11:301–506, 2001.
- [HO00] Martin Hyland and Luke C.-H. Ong. On full abstraction for PCF: I, II, and III. *I&C*, 163(2):285–408, 2000.
- [HS02] Martin Hyland and Andrea Schalk. Games on graphs and sequentially realizable functionals. In *LICS'02*, pages 257–264. IEEE, 2002.
- [HVG03] D. Hughes and R. Van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. In *LICS'03*. IEEE, 2003.
- [Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In *TLCA'99*, volume 1581 of *LNCS*, pages 213–227. Springer, 1999.
- [Lau02] Olivier Laurent. *Étude de la polarisation en logique*. Thèse de doctorat, Université Aix-Marseille II, March 2002.
- [Lau04] Olivier Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1–3):79–123, December 2004.
- [LTdF04] O. Laurent and L. Tortora de Falco. Slicing polarized additive normalization. In *Linear Logic in Computer Science*, volume 316 of *London Math. Soc. Lecture Note Series*, pages 247–282. CUP, 2004.
- [Mai07] Roberto Maieli. Retractable proof nets of the purely multiplicative and additive fragment of linear logic. volume Volume 4790 of *Lecture Notes in Artificial Intelligence*, pages 363–377. Springer Berlin / Heidelberg, 2007.
- [Mel04] P.-A. Mellès. Asynchronous games 2 : The true concurrency of innocence. In *CONCUR 04*, volume 3170 of *LNCS*. Springer Verlag, 2004.
- [Pag06] M. Pagani. *Proof nets and cliques: towards the understanding of analytical proofs*. PhD thesis, Univ. Roma Tre and Université Aix-Marseille II, 2006.
- [TdF00] Lorenzo Tortora de Falco. *Réseaux, cohérence et expériences obsessionnelles*. Thèse de doctorat, Université Paris VII, January 2000. Available at: <http://www.logique.jussieu.fr/www.tortora/index.html>.

- [TdF03a] L. Tortora de Falco. Obsessional experiments for linear logic proof-nets. *MSCS*, 13(6):799–855, 2003.
- [TdF03b] Lorenzo Tortora de Falco. The additive multiboxes. *Annals of Pure and Applied Logic*, 120(1–3):65–102, January 2003.