Linear Realisability over Nets: Multiplicatives

Adrien Ragot
LIPN – UMR 7030 CNRS, Université Sorbonne Paris Nord, France
Dipartimento di Matematica e Fisica, Università Degli Studi Roma Tre, Italy

Thomas Seiller
CNRS, LIPN – UMR 7030, Université Sorbonne Paris Nord, France

Lorenzo Tortora de Falco
Dipartimento di Matematica e Fisica, Università Degli Studi Roma Tre, Italy
GNSAGA, Istituto Nazionale di Alta Matematica, Roma, Italy

Abstract

We provide a new realisability model based on orthogonality for the multiplicative fragment of linear logic, both in presence of generalised axioms ($\mathsf{MLL}^{✠}$) and in the standard case (MLL). The novelty is the definition of cut elimination for generalised axioms. We prove that our model is adequate and complete both for $\mathsf{MLL}^{✠}$ and MLL.

2012 ACM Subject Classification Theory of computation → Linear logic; Theory of computation → Program semantics

Keywords and phrases Linear Logic, Proof Nets, Realisability, Orthogonality, Hypergraphs, Rewriting, Correctness

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.43

Funding *Adrien Ragot*: Supported by a VINCI PhD fellowship from the Franco-Italian Université.
Thomas Seiller: Partially supported by the ANR-22-CE48-0003-01 project DySCo.

Introduction

Since the inception of Linear Logic (LL), proofs are represented as graphs that naturally live in a wider space of agents called proof structures (*nets* in this paper) that can freely interact. These nets were introduced by J.Y. Girard in [7], together with the *desequentialisation*: a simple process transforming proof trees from the sequent calculus of LL into nets. However, not every net is the desequentialisation of a proof: it is *impossible* to extract a proof tree from a net that "contains" cycles or disconnections [5]. Nets can therefore present forms of (what we call) *geometrical incorrectness*, and geometrically correct nets are (representants of) proof trees of LL. More recently, J.Y. Girard proposed *Ludics*, an interpretation of LL given in terms of "desseins": proof trees of the LL sequent calculus with the addition of the daimon (✠) rule, a generalised axiom allowing to prove any sequent. Ludics introduces a new kind of incorrectness that we call *provability incorrectness*: dessein are geometrically correct (they are proof trees) but can be provably incorrect. In the standard theory of proof nets geometrical and provability correctness coincide; it is the presence of daimons that allows to distinguish between provability correctness and geometrical correctness.

Understanding the relationship between correctness and computational behavior is (one of) the goal(s) of *realisability*, which, restricted to LL, will be our focus in this paper. We briefly sum up the existing works on linear realisability[1] by positioning them with respect to Table 1. We also recall if these models enjoy completeness or not. Two lines of research on realisability of LL can be identified.

[1] We use the expression linear realisability in the sense of [20] i.e. realisability models for LL.

33rd EACSL Annual Conference on Computer Science Logic (CSL 2025).
Editors: Jörg Endrullis and Sylvain Schmitz; Article No. 43; pp. 43:1–43:21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Presence of incorrectness, restricted to multiplicative linear logic, for realisability models.

	MLL	MLL$^{\maltese}$
Proof Nets	no incorrectness	provability incorrectness
Nets	geometrical incorrectness	geometrical and provability incorrectness

One was initiated by V. De Paiva *Dialectica Interpretation* [6] and led to P. Oliva's adequate and complete realisability model of first order LL [14] where realisers are proof trees (with standard axioms) from a decorated sequent calculus of LL. As a consequence realisers are typed and are "by construction" *geometrically and provably correct* (placing this model in the top left corner of Table 1).

The other originates in the work of J.Y. Girard: *Ludics* [10], whose "desseins" are geometrically correct but can be provably incorrect (top right corner of Table 1), which enjoys *completeness*. E. Beffara proposed adequate models in a concurrent π-calculus [2] and conjunctive structure [3]. T. Seiller's *interaction graphs* (inspired by Girard's Geometry of Interaction [8]) model various LL fragments adequately [15–19]. Beffara's and Seiller's approaches exhibit both geometrical and provability incorrectness (bottom-right corner of Table 1), but contain no completeness result.

We give the first complete realisability model of the multiplicative fragment of linear logic in terms of nets, essentially the well–known untyped proof–structures of LL [9] with *daimons*, as in the work of P.L. Curien [4]: this places us in the bottom–right corner of Table 1. The main tool we use in our approach to realisability is LL cut elimination: we interpret formulas as types, sets of nets closed under bi–orthogonality, where the notion of orthogonality is defined via the rewriting rules of nets induced by cut elimination. We prove completeness for MLL$^{\maltese}$, multiplicative LL with generalised axioms, meaning our model can capture *geometrical correctness*. As a byproduct we obtain completeness for the standard multiplicative fragment of linear logic (MLL), thus capturing *provability correctness*.

Although not expressed in the terms of realisability, a completeness result for MLL$^{\maltese}$ (in the atomic case) using a notion of orthogonality is already apparent in the work of P.L. Curien [4], where the partitions involved in the Danos Regnier criterion [5] are encoded using daimons. More precisely, one can test the geometrical correctness of a net by confronting it against carefully chosen *opponents* (which as in the work of Béchet [1] are geometrically correct nets). However the method in [4] does not allow to derive a completeness result for MLL. By contrast, we use *geometrically incorrect* opponents to prove completeness for MLL (Remark 87).

The novelty is the *cut elimination* of non-homogenous cuts (a generalised axiom against a connective – say a tensor): unlike in Ludics [2] [10] our daimon is the "perfect" opponent/evaluation context; it never stops responding during computation and never prevents proof search to go on (Figure 4 and Remark 24). These new cut elimination steps are key to interactively identify provability correctness and so to obtain our completeness result for MLL (Remark 86). The computational behavior of the daimon also differs from Krivine's continuations involved in *classical realisability* [12]: they restore a previously stored context while our daimon rather behaves like an adaptive evaluation context.
The general aim is to understand the computational content of proofs and of (incorrect) nets, following a "purely interactive approach to logic" (to quote [10]). We follow the approach initiated with Ludics, we present a framework in which proofs and refutations are objects

[2] In Ludics, the daimon means the end of the game, or the end of the proof search.

of the same nature that can freely interact: a proof–object proves a formula A whenever it "defeats" all the refutations of A. The correctness of an object is evaluated using a dynamic criterion (we make an object interact with each of its refutations) rather than a static one (such as a typing discipline).

Outline. In Section 1, we give a detailed introduction of nets that we define as ordered hypergraphs. In Section 2, we recall the elementary notions of multiplicative linear logic, we introduce the ✠-links and we formulate the criterion of Danos Regnier [5] in our setting. In Section 3, we define orthogonality between two nets as "successful interaction" through cut elimination (Definition 42); this leads to the notion of type: a set of nets closed under bi–orthogonality. We then show how to perform the usual multiplicative constructions in the framework of types. In Section 4, we define our realisability model interpreting formulas as types and we prove its adequacy: a net representing a proof of A is a realiser of A (Theorem 64). In Section 5, we relate correctness criteria with orthogonality. The Danos-Regnier criterion applied to a cut-free net with conclusion A yields a set of nets called tests (Definition 74). We prove that the tests of A are proofs of $A^{\perp}$ (Theorem 77) and that the interaction between a net π with conclusion A and its tests allows to determine whether or not π is indeed a proof: we thus extend to our framework a result of Béchet [1]. In Section 6, we prove the completeness of our realisability model: if a net S realises A (in every basis), then S is a proof of A in $\mathsf{MLL}^{✠}$ (Theorem 85). Finally we show that completeness of $\mathsf{MLL}^{✠}$ implies that of MLL (Theorem 88).

1 Untyped nets

We introduce the framework of *nets* in which our construction takes place. Nets are a special kind of *directed hypergraphs* together with an order of *some* of their vertices which will come in play later on to define the notion of orthogonality. These hypergraphs enjoy a natural notion of sum (Definition 6). In subsection 1.2, we define our "realisers" that we call nets and their computational rules, the cut elimination procedure as known for multiplicative proof structures [9] but with a novelty: the generalised axiom or daimon–link (✠) which behave like an adaptative evaluation context.

1.1 Directed hypergraphs

Given a set X we will let $\mathcal{P}_{\leq}(X)$ denote the set of totally ordered finite subsets of X. An element of $\mathcal{P}_{\leq}(X)$ is equivalently a finite sequence of elements of X but, *without repetitions*.

▶ **Definition 1.** *Suppose given a set L of* labels*. A* directed (L-labelled) hypergraph *is a tuple $(V, E, \mathrm{s}, \mathrm{t}, \ell)$ where V is a finite set of* positions *and E is a finite set of* links*, $\mathrm{s} : E \to \mathcal{P}_{\leq}(V)$ is the* source map*, $\mathrm{t} : E \to \mathcal{P}_{\leq}(V)$ is the* target map *and $\ell : E \to L$ is the* labelling *map.*

Given a link $e \in E$, since the finite sets $\mathrm{t}(e)$ and $\mathrm{s}(e)$ are totally ordered, to support readability we will represent them as sequences: they are respectively called the *target* and the *source* sets of e. A *source* (resp. target) of e is an element of its source (resp. target) set $\mathrm{s}(e)$ (resp.$\mathrm{t}(e)$). The set of targets and sources of e is the *domain* of the link e. We will use superscripts to denote sequences of positions ($\overline{p}, \overline{q}, \overline{u}, \ldots$). A link is a *loop* when its target set and source set are not disjoint.

Convention. Along this work we assume all the hypergraphs to be loop–free i.e. containing only links which are not loops.

Given an hypergraph $\mathcal{H}$ with E as its set of links, we denote $\mathrm{s}(\mathcal{H})$ (resp. $\mathrm{t}(\mathcal{H})$) the set of all positions which are source (resp. target) of at least one link:

$$\mathrm{s}(\mathcal{H}) = \bigcup_{e \in E} \mathrm{s}(e), \qquad \mathrm{t}(\mathcal{H}) = \bigcup_{e \in E} \mathrm{t}(e).$$

A *conclusion/output* (resp. a *premise/input*) of a directed hypergraph $\mathcal{H}$ is a position which is the source (resp. target) of no link in $\mathcal{H}$, i.e. an element of $V \setminus \mathrm{s}(\mathcal{H})$ (resp. of $V \setminus \mathrm{t}(\mathcal{H})$). The set of conclusions (resp. premises) of an hypergraph $\mathcal{H}$ is denoted $\mathsf{out}(\mathcal{H})$ (resp. $\mathsf{in}(\mathcal{H})$). A position p is *isolated* in an hypergraph $\mathcal{H}$ if p is both an output and an input of $\mathcal{H}$, i.e. $p \notin \mathrm{s}(\mathcal{H}) \cup \mathrm{t}(\mathcal{H})$. The *size* of a directed hypergraph is the number of its links. There is a unique empty hypergraph $\mathcal{H} = (V, E, \mathrm{s}, \mathrm{t}, \ell)$ with $V = E = \mathrm{s} = \mathrm{t} = \emptyset$.

An *isomorphism* of hypergraphs $f : (V_1, E_1, \mathrm{s}_1, \mathrm{t}_1, \ell_1) \to (V_2, E_2, \mathrm{s}_2, \mathrm{t}_2, \ell_2)$ is a pair of functions (f_V, f_E) such that $f_V : V_1 \to V_2$ and $f_E : E_1 \to E_2$ are bijections, f_E preserve labels i.e. $\ell(f_E(e)) = \ell(e)$, and f_E preserves the target and source of a link, i.e. $\mathrm{s}_2(f_E(e)) = f_V^*(\mathrm{s}_1(e))$ and $\mathrm{t}_2(f_E(e)) = f_V^*(\mathrm{t}_1(e))$, where f_V^* is the natural extension of f_V to sequences of positions. Along this work we work with hypergraphs up to isomorphism.

▶ Notation 2. We denote $\langle \overline{u} \rhd_l \overline{v} \rangle$ the hypergraph $(V, E, \mathrm{s}, \mathrm{t}, \ell)$ such that $E = \{e\}$, $V = \mathrm{s}(e) \cup \mathrm{t}(e)$, $\mathrm{s}(e) = \overline{u}$, $\mathrm{t}(e) = \overline{v}$ and $\ell(e) = l$ (an example of such a single–link hypergraph is found in Figure 1a). In the sequel $\langle \overline{u} \rhd_l \overline{v} \rangle$ will denote both the described hypergraph and its unique link.

▶ Notation 3. We write $u \cdot v$ the concatenation of sequences. Given $u = (u_1, \dots, u_n)$ a sequence of elements of a set X and an integer $i \in \{1, \dots, n\}$, we denote by $u_{<i}$ (resp. $u_{>i}$) the sequence $(u_1, \dots, u_{i-1})$ (resp. $(u_{i+1}, \dots, u_n)$). Moreover, given two – potentially empty – sequences u and v we denote by $u[i \leftarrow v]$ the sequence $u_{<i} \cdot v \cdot u_{>i}$.

A link is *initial* (resp. *final*) when it has no input (resp. no output). A position is *initial* (resp. *final*) when it is an output (resp. input) of an initial (resp. final) link. In an hypergraph $\mathcal{H}$, a link e is *terminal* when every target of e is a conclusion of $\mathcal{H}$ – thus a final link is a terminal link.

▶ **Example 4.** For instance a link $\langle \rhd_\ell\, a, b, c \rangle$ is an initial link and the positions a, b and c are initial, on the other hand a link $\langle a, b \rhd_\ell c \rangle$ is not initial and neither are the positions a, b or c.

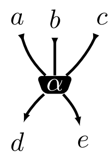
Hypergraphs enjoy a natural notion of sum based on the disjoint union of the set of links.

▶ Notation 5. Given two sets X_0 and X_1 we denote $X_0 \uplus X_1$ the set $X_0 \cup X_1$ whenever X_0 and X_1 are disjoint. Given two functions $f : X_0 \to E$ and $g : X_1 \to E$ with disjoint domains we denote $f \uplus g$ the function which takes an element x of $X_0 \uplus X_1$, and returns $f(x)$ if $x \in X_0$ and $g(x)$ if $x \in X_1$.

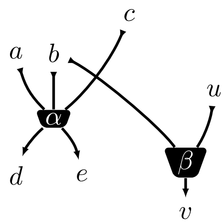
▶ **Definition 6.** *Given two hypergraphs $\mathcal{H}_1 = (V_1, E_1, \mathrm{t}_1, \mathrm{s}_1, \ell_1)$ and $\mathcal{H}_2 = (V_2, E_2, \mathrm{t}_2, \mathrm{s}_2, \ell_2)$ such that $E_1 \cap E_2 = \emptyset$. The sum of $\mathcal{H}_1$ and $\mathcal{H}_2$ is defined as:*

$$\mathcal{H}_1 + \mathcal{H}_2 = (V_1 \cup V_2, E_1 \uplus E_2, \mathrm{t}_1 \uplus \mathrm{t}_2, \mathrm{s}_1 \uplus \mathrm{s}_2, \ell_1 \uplus \ell_2).$$

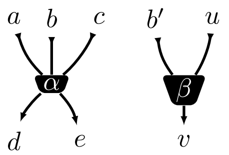
▶ Remark 7. Whenever $\mathcal{H}_1 = (V_1, E_1, \mathrm{t}_1, \mathrm{s}_1, \ell_1)$ and $\mathcal{H}_2 = (V_2, E_2, \mathrm{t}_2, \mathrm{s}_2, \ell_2)$ are such that $E_1 \cap E_2 \neq \emptyset$, we will abusively write their sum as $\mathcal{H}_1 + \mathcal{H}_2 = (V_1 \cup V_2, E_1 \uplus E_2, \mathrm{t}_1 \uplus \mathrm{t}_2, \mathrm{s}_1 \uplus \mathrm{s}_2, \ell_1 \uplus \ell_2)$, since up to renaming the sets of links of two hypergraphs can always be considered disjoint.



(a) The representation of the single–link hypergraph $\langle a, b, c \rhd_\alpha d, e\rangle$ (Notation 2).



(b) The sum of two single link hypergraphs $\langle a, b, c \rhd_\alpha d, e\rangle$ and $\langle b, u \rhd_\beta v\rangle$ (Definition 6).



(c) The parallel sum of the two single link hypergraphs $\langle a, b, c \rhd_\alpha d, e\rangle$ and $\langle b, u \rhd_\beta v\rangle$ (Definition 11).

Figure 1 Hypergraphs can naturally be represented in a graphical way, we illustrate the notation of a hypergraph containing a single link, the sum of hypergraphs and the parallel sum of hypergraphs. In Figure 1c The position b is present in both hypergraphs therefore we rename it in one of the two hypergraphs: thus $\langle a, b, c \rhd_\alpha d, e\rangle \parallel \langle b, u \rhd_\beta v\rangle$ equals $\langle a, b, c \rhd_\alpha d, e\rangle \parallel \langle b', u \rhd_\beta v\rangle$ (that is, upto isomorphism).

▶ Remark 8. Vertices may overlap in a sum (as we take the union of vertex sets rather than the disjoint union). As a consequence, a position may be input (or output) of several distinct links (Figure 1b). We can describe hypergraphs as sums of simple hypergraphs; namely those that contain only one link. Indeed using Notation 2, an hypergraph consisting of two links $\langle \overline{a} \rhd_\ell \overline{b}\rangle$ and $\langle \overline{c} \rhd_{\ell'} \overline{d}\rangle$ is in fact equal to the sum of the single-link hypergraphs $\langle \overline{a} \rhd_\ell \overline{b}\rangle$ and $\langle \overline{c} \rhd_{\ell'} \overline{d}\rangle$. By induction on the number of links, this shows that any hypergraph $\mathcal{H}$ without isolated positions can be written as $\mathcal{H} = \sum_{e\in E}\langle \mathrm{s}(e) \rhd_{\ell(e)} \mathrm{t}(e)\rangle$.

▶ **Example 9.** In the hypergraph $\langle \rhd_{\ell 1} a, b, c\rangle + \langle a \rhd_{\ell 2} d\rangle + \langle \rhd_{\ell 3} e\rangle + \langle e \rhd_{\ell 4}\rangle$ the set of initial positions is $\{a, b, c, e\}$, while e is the only final position of the hypergraph, and it belongs to the domain of the unique final link $\langle e \rhd_{\ell 4}\rangle$.

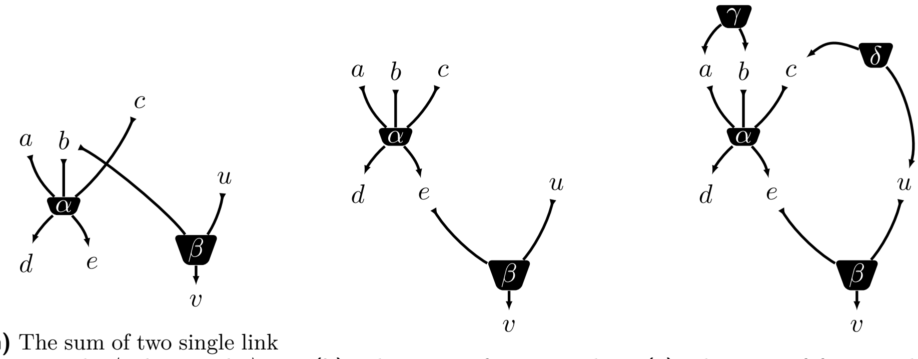
▶ Remark 10. The sum of hypergraphs enjoys the properties of an abelian monoid; associativity, commutativity, and a neutral element which is the empty hypergraph.

We will also use extensively the notion of *parallel composition* or *parallel sum* of hypergraphs, an analogue of the *union–graph* of two simple graphs.

▶ **Definition 11.** *Given* $\mathcal{H}_1 = (V_1, E_1, \mathrm{t}_1, \mathrm{s}_1, \ell_1)$ *and* $\mathcal{H}_2 = (V_2, E_2, \mathrm{t}_2, \mathrm{s}_2, \ell_2)$ *two hypergraphs such that* $V_1 \cap V_2 = E_1 \cap E_2 = \emptyset$*, we define their* parallel sum *as:* $\mathcal{H}_1 \parallel \mathcal{H}_2 = (V_1 \uplus V_2, E_1 \uplus E_2, \mathrm{t}_1 \uplus \mathrm{t}_2, \mathrm{s}_1 \uplus \mathrm{s}_2, \ell_1 \uplus \ell_2)$.

▶ Remark 12. The parallel sum of two hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ corresponds to a regular sum whenever the sets of vertices are disjoint. Just like the sum, parallel composition can always be performed between two hypergraphs (up to a renaming, see Figure 1c).

A hypergraph $\mathcal{H} = (V, E, \mathrm{t}, \mathrm{s}, \ell)$ is: (1) *target–surjective* whenever $\mathrm{t}(\mathcal{H}) = V$, (2) *source–disjoint* if the sets $\mathrm{s}(e)$ for $e \in E$ are pairwise disjoint, (3) *target–disjoint* if the sets $\mathrm{t}(e)$ for $e \in E$ are pairwise disjoint (Figure 2). A *module* is an hypergraph which is target–disjoint and source–disjoint, which means that for each position p there exists *at most* one link e such that $\mathrm{s}(e)$ (resp. $\mathrm{t}(e)$) contains p. Any single–link hypergraph is a module. Uncarefully summing two modules does not necessarily result in a module; for instance the single link hypergraphs $e = \langle \rhd_\ell a\rangle$ and $e' = \langle \rhd_{\ell'} a\rangle$ are both modules but their sum isn't as a is the target of the two links e and e'.



(a) The sum of two single link hypergraphs $\langle a, b, c \rhd_\alpha d, e\rangle + \langle b, u \rhd_\beta v\rangle$. The hypergraph is target–disjoint, but because b belongs to the source of both links it is not source–disjoint, it is also not target surjective.

(b) The sum of two single link hypergraphs $\langle a, b, c \rhd_\alpha d, e\rangle + \langle e, u \rhd_\beta v\rangle$. The hypergraph is target–disjoint and source–disjoint, however it is not target surjective.

(c) The sum of four single link hypergraphs $\langle a, b, c \rhd_\alpha d, e\rangle + \langle e, u \rhd_\beta v\rangle + \langle \rhd_\gamma a, b\rangle + \langle \rhd_\delta c, u\rangle$. The hypergraph is target–disjoint, source–disjoint, and target surjective.

Figure 2 Properties of hypergraphs: source–disjoint, target–disjoint and target–surjective hypergraphs.

An *arrangement* of a directed hypergraph $\mathcal{H}$ is a total order $<_{\mathbf{a}}$ on its set of conclusions; equivalently the order may be identified as a bijection $\mathbf{a} : \{1, \dots, card(\mathsf{out}(\mathcal{H}))\} \to \mathsf{out}(\mathcal{H})$. An *ordered hypergraph* is a pair $(\mathcal{H}, \mathbf{a})$ of an hypergraph $\mathcal{H}$ together with an arrangement $\mathbf{a}$ of $\mathcal{H}$. Given an ordered hypergraph $(\mathcal{H}, \mathbf{a})$ with n conclusions for an integer $1 \leq i \leq n$, we denote $\mathbf{a}(i)$ by $\mathcal{H}(i)$ whenever there is no ambiguity. The arrangement $\mathbf{a}$ is denoted $\mathbf{a}(\mathcal{H})$, and we might refer to $\mathcal{H}$ as the *unordered hypergraph underlying* $(\mathcal{H}, \mathbf{a})$.

For $n, m \in \mathbb{N}$ we denote by $[n; m]$ the set of integers i such that $n \leq i \leq m$. Given two functions $f : [1; n] \to E$ and $g : [1; m] \to E$ we denote $f \uplus g : [1; m+n] \to E$ the function such that $f \uplus g(i) = f(i)$ when $1 \leq i \leq n$ and $f \uplus g(i) = g(i-n)$ when $n+1 \leq i \leq n+m$. This operation is not commutative. The parallel sum of two ordered hypergraph $(\mathcal{H}_1, \mathbf{a}_1)$ and $(\mathcal{H}_2, \mathbf{a}_2)$ naturally yields an ordered hypergraph as $(\mathcal{H}_1 \parallel \mathcal{H}_2, \mathbf{a}_1 \uplus \mathbf{a}_2)$ (note that however this is not a commutative operation).

1.2 Multiplicative nets

Up to this point we have allowed any kind of link to occur in a hypergraph. We now consider untyped multiplicative nets in which only some specific kinds of links occur. We fix the set of labels as the set made of the *daimon* (✠) the tensor ($\otimes$) the par ($⅋$) and the cut (cut) symbol. Furthermore we fix a family of links, namely ✠-labelled links that have no inputs (they are initial links), cut-labelled links that have exactly two inputs and no outputs (they are final links), $\otimes$- and $⅋$-labelled links that have exactly two inputs and one output. As a consequence, the hypergraphs considered will closely resemble to multiplicative linear logic proof structures, with two important points of divergence: the absence of typing and the presence of generalised axioms, a standard MLL axiom link can be seen as daimon link with two conclusions [3].

Formally we fix a countable set Pos of positions and a family of links $\mathcal{L}$ defined as:

$$\mathcal{L} \triangleq \{\langle p_1, p_2 \rhd_\otimes p\rangle,\ \langle p_1, p_2 \rhd_{⅋} p\rangle,\ \langle p_1, p_2 \rhd_{\mathsf{cut}}\rangle \mid p_1, p_2, p \in \mathsf{Pos}\} \quad \cup \quad \{\langle \rhd_{✠} p_1, \dots, p_n\rangle \mid n \in \mathbb{N}, p_1, \dots, p_n \in \mathsf{Pos}\}.$$

[3] To be precise one should say that an *atomic* standard MLL axiom link *is* a daimon link with two conclusions (Remark 29).

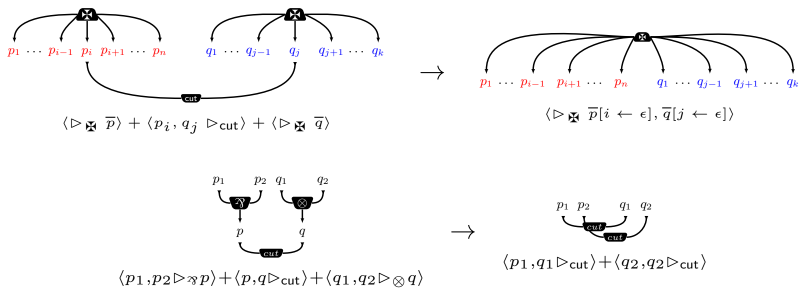


Figure 3 Rewriting defining the homogeneous cut elimination. We provide a representation of each hypergraph involved above its expression. In the step of the glueing cut we assume the two daimons to be distinct i.e. the cut is acyclic. In this figure $\overline{p} = p_1, \ldots, p_n$ while $\overline{q} = q_1, \ldots, q_k$.

▶ **Definition 13.** *A multiplicative module is an ordered hypergraph $M = (|M|, \mathbf{a}(M))$ where $|M|$ is a sum of links of $\mathcal{L}$ which is a module.*

A multiplicative net is a multiplicative module $S = (|S|, \mathbf{a}(S))$ where $|S|$ is target–surjective.

From now on we will omit the word *multiplicative* but a module (resp. net) will always be a multiplicative module (resp. net). For a module M (resp. a net S) we refer to $|M|$ (resp. $|S|$) as the unordered hypergraph underlying M (resp. S). An *unordered* module (resp. net) is the unordered hypergraph underlying a module (resp. net).

▶ Remark 14. For two nets $S_1 = (V_1, E_1, \mathrm{s}_1, \mathrm{t}_1, \ell_1)$ and $S_2 = (V_2, E_2, \mathrm{s}_2, \mathrm{t}_2, \ell_2)$, if $S_1 + S_2$ remains a net then $S_1 + S_2 = S_1 \parallel S_2$. Indeed, by Definition 6, $E_1 \cap E_2 = \emptyset$. Then, by target–disjointness $\mathrm{t}(S_1) \cap \mathrm{t}(S_2) = \emptyset$; and finally because S_1 and S_2 are target surjective we have $V_1 \cap V_2 = \mathrm{t}(S_1) \cap \mathrm{t}(S_2) = \emptyset$, so that Definition 11 applies.

▶ Notation 15. Given an integer n we denote by $\maltese_n$ any multiplicative net consisting of a single daimon link with n outputs, i.e. isomorphic to $\langle \rhd_{\maltese}\ p_1, \ldots, p_n\rangle$.

▶ **Definition 16.** *Given a multiplicative net S the* type *of a cut link $c = \langle p, q \rhd_{\mathsf{cut}}\rangle$ occurring in S is the multiset of the two labels of the links of output p and q; for readability we write these multisets as ordered pairs. Thus there are six* types *of cuts (up to symmetry). More precisely, we distinguish:* multiplicative *cuts, of type* $(\otimes/⅋)$*;* clash *cuts, of type* $(\otimes/\otimes)$ *or* $(⅋/⅋)$*;* glueing *cuts, of type* $(\maltese/\maltese)$*;* non–homogeneous *cuts, of type* $(\otimes/\maltese)$ *or* $(⅋/\maltese)$*, which are respectively called* reversible *and* irreversible *cuts. In a net S, a cut $\langle p, q \rhd_{\mathsf{cut}}\rangle$ is* cyclic *whenever p and q are targets of the same link.*

▶ Remark 17. Each cut link occurring in a net S has a type since a net is target–surjective. However in a module this isn't true: for instance in the module $\langle p, q \rhd_{\mathsf{cut}}\rangle$ consisting of a single cut link, the type of the cut link is not defined.

▶ Remark 18. The inputs of a cut link $\langle p, q \rhd_{\mathsf{cut}}\rangle$ are ordered, making the two links $\langle p, q \rhd_{\mathsf{cut}}\rangle$ and $\langle q, p \rhd_{\mathsf{cut}}\rangle$ distinct. However (up to isomorphism) this plays no role during cut elimination.

Multicative nets comes with their notion of computation called *cut elimination*: it is a rewriting on nets and more precisely it rewrites a redex (that is a sub–net made of a single cut link and two non–cut links) into redexes or daimons (in the very specific case of glueing cuts). Up to isomorphism, how a redex is rewritten depends solely on its type (Definition 16).

▶ **Definition 19.** *The relation of* homogeneous cut elimination *on unordered nets is denoted by $\to_h$ and it is the rewriting relation defined as the contextual closure (with respect to the sum) of the relation defined in Figure 3.*

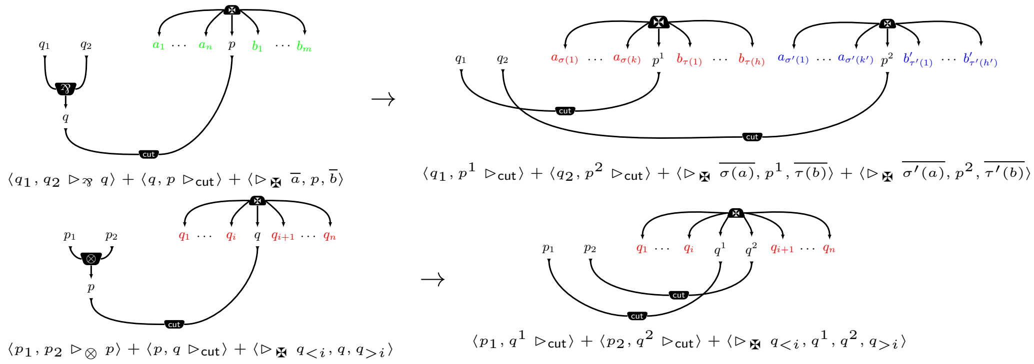


Figure 4 Rules defining the non–homogeneous cut elimination. In the elimination of the $(\parr/\maltese)$ cut - first row - $\overline{a} = (a_1, \ldots, a_n)$ and $\overline{b} = (b_1, \ldots, b_m)$ while $\sigma(\overline{a}) = (a_{\sigma(1)}, \ldots, a_{\sigma(k)})$, $\sigma'(\overline{a}) = (a_{\sigma'(1)}, \ldots, a_{\sigma'(k')})$, $\tau(\overline{b}) = (b_{\tau(1)}, \ldots, b_{\tau(h)})$, $\tau'(\overline{b}) = (b_{\tau'(1)}, \ldots, b_{\tau'(h')})$ (with $n = k + k'$ and $m = h + h'$) are sequences that define a partition of $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ more precisely $\{a_1, \ldots, a_n\} = \{a_{\sigma(1)}, \ldots, a_{\sigma(k)}, a_{\sigma'(1)}, \ldots, a_{\sigma'(k')}\}$ and $\{b_1, \ldots, b_n\} = \{b_{\tau(1)}, \ldots, b_{\tau(h)}, b_{\tau'(1)}, \ldots, b_{\tau'(h')}\}$, and $\sigma, \sigma', \tau, \tau'$ are permutations. Furthermore p^1, p^2, q^1, q^2 are fresh positions. The figure is slightly misleading: q_1 and q_2 may be elements of $\overline{a}$ or $\overline{b}$ (in the first row) while p_1 and p_2 may be elements of $q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_n$ (in the second row), these cases are illustrated in Figure 13. This has an important consequence: a cut can belong to a cycle and still be reducible (Remark 28).

▶ Remark 20. The (homogeneous) cut elimination procedure on unordered nets leave the conclusions unchanged. As a consequence the homogeneous cut elimination can be lifted from unordered nets to nets: whenever two unordered nets are such that $S \to S'$, for any arrangement $\mathbf{a}$ of S we have $(S, \mathbf{a}) \to (S', \mathbf{a})$.

The following result is easily established, in particular since the number of links strictly decreases during homogeneous cut elimination.

▶ **Proposition 21.** *Homogeneous cut elimination is confluent and strongly normalizing.*

▶ **Definition 22.** *The non homogeneous reduction is denoted $\to_{nh}$ and it is defined on unordered nets as the contextual closure of the relation given in Figure 4.*

▶ Remark 23. The non–homogeneous reduction preserves the conclusion of the nets, hence it can be lifted to ordered nets – as in remark 20.

▶ Remark 24. In the framework of Multiplicative Linear Logic (Section 2, Figure 6c), non homogeneous cut elimination simulates proof search in the sequent calculus:

$$\dfrac{\dfrac{}{\Gamma, A \parr B}\maltese \quad \dfrac{\dfrac{\dfrac{}{A^\perp, A}\maltese \quad \dfrac{}{B^\perp, B}\maltese}{A^\perp \otimes B^\perp, A, B}\otimes}{A^\perp \otimes B^\perp, A \parr B}\parr}{\Gamma, A \parr B}\mathsf{cut} \quad \to^* \quad \dfrac{\dfrac{}{\Gamma, A, B}\maltese}{\Gamma, A \parr B}\parr \qquad \dfrac{\dfrac{}{\Gamma, A \otimes B}\maltese \quad \dfrac{\dfrac{\dfrac{}{A^\perp, A}\maltese \quad \dfrac{}{B^\perp, B}\maltese}{A^\perp, B^\perp, A \otimes B}\otimes}{A^\perp \parr B^\perp, A \otimes B}\parr}{\Gamma, A \otimes B}\mathsf{cut} \quad \to^* \quad \dfrac{\dfrac{}{\Gamma_1, A}\maltese \quad \dfrac{}{\Gamma_2, B}\maltese}{\Gamma, A \otimes B}\otimes$$

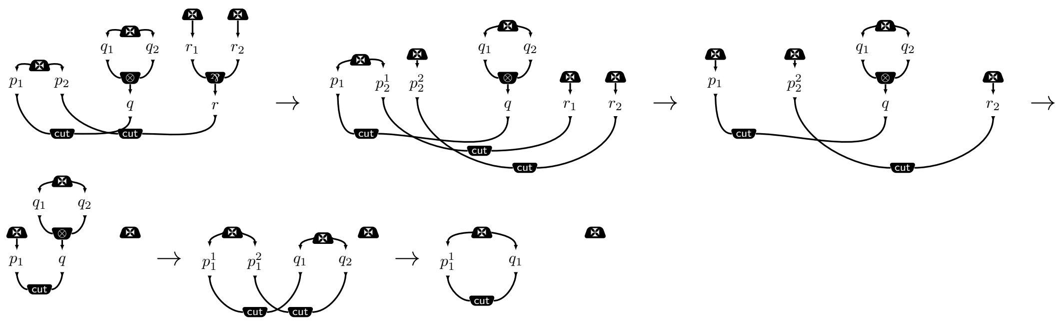
This also illustrates the non determinism of the $(\maltese/\parr)$ reduction step which corresponds to proof search on a formula of the form $A \otimes B$: going from bottom to top the $\otimes$–introduction rule splits the context Γ, which is a non deterministic process. A consequence of non determinism is the loss of confluence for cut elimination (but not of strong normalisation, Proposition 30); since splitting the context is irreversible, a net can have different normal forms, like the second net of figure 5b (from left to right) which coincides with the second net of figure 5c: this same net reduces, following the two figures, to two different normal forms.

▶ Remark 25. A cyclic cut is a glueing cut. Indeed, given a cyclic cut link $\langle p, q \rhd_{\mathsf{cut}} \rangle$ in a net, because p and q belong to the target of a same link e and the only links which may have several targets are daimon links it follows that e is a daimon link.

▶ Remark 26. The side condition of Figure 3 entails that a cyclic cut is not reducible: for example the net $\langle \rhd_{\maltese} p, q \rangle + \langle p, q \rhd_{\mathsf{cut}} \rangle$ is a net in normal form.

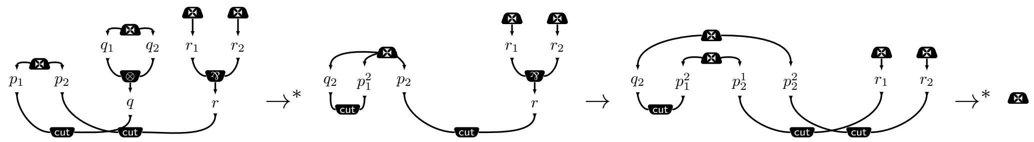
▶ Remark 27. A cut link which is not reducible is either a clashing cut or a cyclic glueing cut. Notice, however, that while clashing cuts never disappear during cut elimination, cyclic cuts may disappear (see Figure 10b).

▶ Remark 28. In the standard framework of MLL proof structures the cut elimination of an axiom against a cut is defined as the identification of the two extreme positions, therefore eliminating such a cut may create *loops* (Section 1). To avoid loops from occurring during cut elimination an ad hoc condition is usually added (see for example [13]). In our framework, this condition is the rather natural side condition of Figure 3.

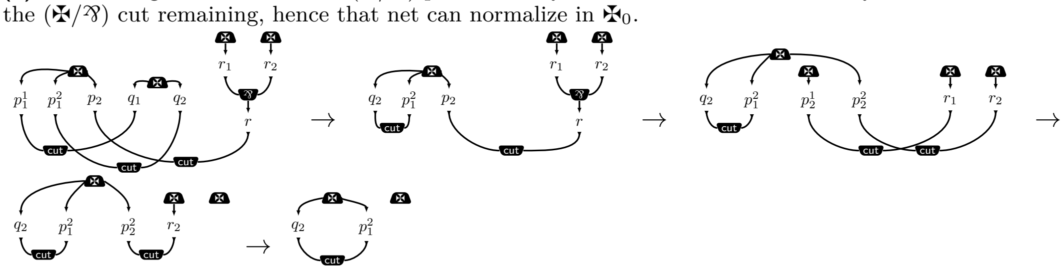


(a) Eliminating first the *irreversible cut* ($\maltese/\parr$) produces a net[a] which cannot normalize in $\maltese_0$.

[a] The ($\maltese/\parr$) reduction step is not deterministic but in this very special case any choice yields the same net.



(b) Eliminating the reversible cut ($\maltese/\otimes$) produces a cycle which can be eliminated by the elimination of the ($\maltese/\parr$) cut remaining, hence that net can normalize in $\maltese_0$.



(c) Non determinism also comes from the choice of how we reduce ($\parr/\maltese$) cuts, different choices leading to different normal forms: the "wrong" choice results in a net which cannot normalize to $\maltese_0$.

Figure 5 Non homogeneous cut eliminations contains two sources of non–determinism.

▶ Remark 29. Notice that whenever daimons are binary and typed by dual atomic formulas the cut elimination procedure for $\mathsf{MLL}^{\maltese}$ defined in Definition 19 is exactly the standard cut elimination procedure for MLL [7], [13].

The rewriting rule, denoted $\to$, associated with cut elimination is the union of the homogeneous and non–homogeneous cut elimination i.e. $\to_h \cup \to_{nh}$. We write $S \xrightarrow{c} S'$, when S' is obtained from S by eliminating the cut c. We write by $S \to_{\mathsf{mult}} S'$ (resp. $S \to_{\neg\mathsf{mult}} S'$) whenever $S \xrightarrow{c} S'$ and c is multiplicative (resp. not multiplicative). Given two binary relations R_1 and R_2 on a set X we denote by $R_1 \cdot R_2$ their composition, i.e. for two $x, y \in X$ $xR_1 \cdot R_2 y$ if and only if there exists z such that $xR_1 z$ and $zR_2 y$.

▶ **Proposition 30.** *Cut elimination is strongly normalising, furthermore:*
1. $\to^*$ *can be factorised as* $\to^*_{\mathsf{mult}} \cdot \to^*_{\neg\mathsf{mult}}$.
2. *If c is a* $(\parr/\maltese)$ *cut in S; if* $S \xrightarrow{c} \cdot \to^* S'$ *then* $S \to^* \cdot \xrightarrow{c} S'$.
3. *If c is not a* $(\parr/\maltese)$ *cut in S; if* $S \to^* \cdot \xrightarrow{c} S'$ *then* $S \xrightarrow{c} \cdot \to^* S'$.

$$\begin{aligned} A, B &\triangleq X \in \mathsf{Var} \\ &\mid A \parr B \mid A \otimes B \\ \mathcal{H}_1, \mathcal{H}_2 &\triangleq A \in \mathsf{Form} \\ &\mid \mathcal{H}_1, \mathcal{H}_2 \mid \mathcal{H}_1 \parallel \mathcal{H}_2 \end{aligned}$$

(a) Grammar defining Form (first two rows), and grammar defining Hseq (last two rows).

$$(A \parr B)^\perp = A^\perp \otimes B^\perp \qquad (A \otimes B)^\perp = A^\perp \parr B^\perp$$

(b) De Morgan laws lifting the involution $(\cdot)^\perp$ from Var to Form.

$$\frac{}{\Gamma}\,\maltese \qquad \frac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr \qquad \frac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\Gamma, A \quad \Delta, A^\perp}{\Gamma, \Delta}\,\mathsf{cut} \qquad \frac{\Gamma, A, B, \Delta}{\Gamma, B, A, \Delta}\,\mathsf{ex} \qquad \frac{}{A, A^\perp}\,\mathsf{ax}$$

(c) Rules used for constructing the proof trees. The rules $(\maltese, \parr, \otimes, \mathsf{cut}, \mathsf{ex})$ define the $\mathsf{MLL}^{\maltese}$ fragment. Substituting the $(\maltese)$–daimon rule with the (ax)–axiom rule results in the fragment MLL, that is $(\mathsf{ax}, \parr, \otimes, \mathsf{cut}, \mathsf{ex})$.

Figure 6 Grammar of formulas and (hyper)sequent, de Morgan laws and inference rules.

$$\langle \bar{c} \rhd_\ell \bar{a}, p_1, \bar{b} \rangle + \langle p_1, p_2 \rhd_\parr p \rangle \to^l_\parr \langle \bar{c} \rhd_\ell \bar{a}, p, \bar{b} \rangle \qquad \langle \bar{c} \rhd_\ell \bar{a}, p_2, \bar{b} \rangle + \langle p_1, p_2 \rhd_\parr p \rangle \to^r_\parr \langle \bar{c} \rhd_\ell \bar{a}, p, \bar{b} \rangle$$

Figure 7 The two cases (left and right) defining the switching rewriting. The left reduction $\to^l_\parr$ destroys p_1 and makes p_2 a conclusion; while the right reduction $\to^r_\parr$ destroys p_2 and makes p_1 a conclusion.

$$\frac{}{\Gamma}\,\maltese \qquad \frac{\begin{matrix}\pi_1 \\ A, \Gamma\end{matrix} \quad \begin{matrix}\pi_2 \\ A^\perp, \Delta\end{matrix}}{\Gamma, \Delta}\,\mathsf{cut} \qquad \frac{\begin{matrix}\pi_1 \\ A, \Gamma\end{matrix} \quad \begin{matrix}\pi_2 \\ B, \Delta\end{matrix}}{\Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\begin{matrix}\pi_0 \\ A, B, \Gamma\end{matrix}}{A \parr B, \Gamma}\,\parr \qquad \frac{\begin{matrix}\pi_0 \\ \Gamma, B, A, \Delta\end{matrix}}{\Gamma, A, B, \Delta}\,\mathsf{ex}$$

$$\langle \rhd_\maltese p_1, \ldots, p_n \rangle \qquad \begin{matrix} S_1 + S_2 + \\ \langle S_1(1), S_2(1) \rhd_{\mathsf{cut}} \rangle \end{matrix} \qquad \begin{matrix} S_1 + S_2 + \\ \langle S_1(1), S_2(1) \rhd_\otimes p \rangle \end{matrix} \qquad \begin{matrix} S_0 + \\ \langle S_0(1), S_0(2) \rhd_\parr p \rangle \end{matrix} \qquad (S_0, a)$$

Figure 8 Induction defining the relation $\equiv_\mathcal{R}$. The proof in the first row is represented by a net below it in the second row. The position p is always supposed fresh. In each case and for each $0 \leq i \leq 2$, S_i is a net which represent π_i i.e. $S_i \equiv_\mathcal{R} \pi_i$. In the case of the exchange rule we explicitly mention the arrangement i.e. the order of the conclusion and assume $(S_0, a') \equiv_\mathcal{R} \pi_0$ and $a(i) = a'(i)$ whenever $i \leq |\Gamma|$ or $|\Gamma| + 2 < i$. On the other hand, $a'(|\Gamma| + 1) = a(|\Gamma| + 2)$ and $a'(|\Gamma| + 2) = a(|\Gamma| + 1)$.

2 Multiplicative Linear Logic and proof nets

We define the well–known notion of proof net [7] in our setting: in the presence of the generalised axiom $(\maltese)$, proof nets are similar to the *paraproof nets* of Curien [4] (which come from Girard Ludics [10]). We then formulate the Danos–Regnier criterion [5]: testing the acyclicity and connectedness of (several) graphs allows to determine whether a net is a (para)proof net or not [4].

We fix a countable set Var of *propositional variables*. The set Var comes with an (explicit) involution $(\cdot)^\perp$; for each atomic variable X there exists its *dual* atomic variable $X^\perp$ in Var. The set Form of *formulas* of multiplicative linear logic is defined by the grammar in Figure 6a. The involution $(\cdot)^\perp$ is lifted from Var to Form as in Figure 6b. The set Hseq of *hypersequents* is defined by the grammar in Figure 6a, a *sequent* is an hypersequent without the *parallel* "$\parallel$" constructor. The introduction of hypersequents is naturally suggested by the constructions on types (Section 3): indeed as the interpretation of the $⅋$–connective is based on the interpretation of the ","–connective, the interpretation of the $\otimes$–connective relies on that of the "$\parallel$"–connective (Definition 49 and Definition 58). Technically hypersequents are necessary in *our* proof of the completeness theorem (Theorem 88).

A *proof* of MLL (resp. $\mathsf{MLL}^{✠}$) is a tree constructed using the rules $(\mathsf{ax}, ⅋, \otimes, \mathsf{cut}, \mathsf{ex})$ (resp. $(✠, ⅋, \otimes, \mathsf{cut}, \mathsf{ex})$) of Figure 6c.

▶ **Definition 31.** *A net S* represents[4] *a proof π of $\mathsf{MLL}^{✠}$, denoted $\pi \equiv_{\mathcal{R}} S$ or $S \equiv_{\mathcal{R}} \pi$, whenever the relation defined in Figure 8 holds. A net represents a proof of* MLL *whenever it represents a proof of $\mathsf{MLL}^{✠}$ where every sequent conclusion of a $(✠)$–rule has shape $A, A^\perp$ for $A \in$ Form. A representation of a proof π is a net S which represents π. A* proof net *of $\mathsf{MLL}^{✠}$ (resp.* MLL*) is a net which represents a proof of $\mathsf{MLL}^{✠}$ (resp.* MLL*): we say that S is* correct*. A net S is* correctly typeable[5] *by a sequent Γ whenever it represents a proof of Γ in $\mathsf{MLL}^{✠}$.*

▶ Notation 32. Let $\mathfrak{P}$ denote MLL or $\mathsf{MLL}^{✠}$ and let S be a net. We write $S \vdash_{\mathfrak{P}} \Gamma$ whenever there exists a proof π in $\mathfrak{P}$ such that S is the representation of π. Furthermore we denote $\{\!\![\Gamma : \mathfrak{P}]\!\!\}$ the set of all the nets S such that $S \vdash_{\mathfrak{P}} \Gamma$.

A *substitution* is a map $\theta : \mathsf{Var} \to \mathsf{Form}$ such that $\theta(X^\perp) = \theta(X)^\perp$ for each $X \in$ Var. A substitution can be lifted to formulas and hypersequents by induction: $\theta(A \otimes B) = \theta(A) \otimes \theta(B)$; $\theta(A ⅋ B) = \theta(A) ⅋ \theta(B)$; $\theta(A \parallel B) = \theta(A) \parallel \theta(B)$; $\theta(A, B) = \theta(A), \theta(B)$. Given two hypersequents, we denote $\Delta \leq \Gamma$ whenever there exists a substitution θ such that $\theta\Delta = \Gamma$.

▶ **Proposition 33.** *Let Γ and Δ be two sequents and suppose $\Delta \leq \Gamma$. For any net S: (1) if $S \vdash_{\mathsf{MLL}^{✠}} \Delta$ then $S \vdash_{\mathsf{MLL}^{✠}} \Gamma$ and (2) if $S \vdash_{\mathsf{MLL}} \Delta$ then $S \vdash_{\mathsf{MLL}} \Gamma$.*

▶ **Definition 34.** *The* switching *rewriting is defined on unordered nets as the contextual closure of the rules in Figure 7. A* switching *of a net S is a normal form of S for the switching rewriting: we often denote it σS.*

▶ Remark 35. The switching rewriting strongly normalizes since every step reduces the number of links of the net. The rewriting is also non-deterministic and non-confluent, every normal form is a par–free net. The switching rewriting can be lifted to (ordered) nets; with the notations of Figure 7 whenever an unordered net $|S|$ with n conclusions is such that $|S| \to^{l}_{⅋} |S'|$ we define $(|S|, \mathbf{a}) \to^{l}_{⅋} (|S'|, \mathbf{a}')$ where $\mathbf{a}'(i) = \mathbf{a}(i)$ for each $1 \leq i \leq n$ and $\mathbf{a}'(n+1) = p_2$ i.e. the new conclusion is made last conclusion (similarly we can define it for the case $\to^{r}_{⅋}$).

[4] In the standard Linear Logic terminology π is a sequentialisation of the proof net S.
[5] Notice that with the expressions "correctly typeable" we mean here that the net is both correct (it represents a proof) and that we can label its conclusions with the formulas of Γ.

▶ **Definition 36.** *The* undirected multigraph[6] *induced by two partitions P and Q of a set X is (V, E, brd) denoted $\mathsf{G}(P, Q)$ where: (1) $V = \{1\} \times P \cup \{2\} \times Q$ the vertices are the classes of P and Q (as a disjoint union); (2) $E = X$; (3) For any edge x in X; $\mathsf{brd}(x) = \{(1, P_x), (2, Q_x)\}$ where $P_x \in P$ is such that $x \in P_x$ and $Q_x \in Q$ is such $x \in Q_x$.*

Two partitions P and Q of a set X are orthogonal *if the multigraph $\mathsf{G}(P, Q)$ is acyclic and connected.*

▶ **Definition 37.** *In a net S denote $p \geq_S q$ the relation which holds whenever there exists a link e such that $p \in \mathrm{s}(e)$ and $q \in \mathrm{t}(e)$. Denote $\geq_S^*$ its reflexive and transitive closure; a position p is* above *a position q whenever $p \geq_S^* q$. Given a position q we denote $q \uparrow^i S$ the set of* initial *positions which are above q in S.*

▶ Remark 38. Given a *cut–free* net S with conclusions $p_1, \ldots, p_n$ the sets $p_1 \uparrow^i S, \ldots, p_n \uparrow^i S$ form a partition of the initial positions of S. We denote this partition $\uparrow^i S$.

▶ Notation 39. Let S be a net and let $\{d_1, \ldots, d_n\}$ be the set of daimon links of S. The partition $\{\mathrm{t}(d_1), \ldots, \mathrm{t}(d_n)\}$ on the set of initial positions of S is denoted by $\mathsf{P}_{✠}(S)$.

Reformulated in the context of hypergraphs we get the following theorem from [5].

▶ **Theorem 40** ([4, 5]). *Given a cut–free net S, the following assertions are equivalent:*
1. *S is a proof net of $\mathsf{MLL}^{✠}$;*
2. *For every switching σS of S, the partitions $\mathsf{P}_{✠}(S)$ and $\uparrow^i \sigma S$ of the set of initial positions of S are orthogonal;*
3. *Every switching σS of S is acyclic and connected*[7]*.*

3 Interaction of nets, orthogonality, and types

We define how nets can *interact* and if the interaction of two nets leads to the ✠-link with no outputs ($✠_0$) we say they are *orthogonal*. This recalls classical realisability proposed by J.-L. Krivine [12], where (the closure by antireduction of) the set $\{✠_0\}$ will play the role of the *pole*. Notice, however, that our setting is fully symmetrical: both the elements of truth values and falsity values are nets.

The notion of ordered hypergraph and *arrangement* introduced in Section 1 will now explicitly come into play as it is necessary for defining the interactions of nets (see Figure 9a). We will denote by $\#S$ the number of outputs of a net S. Given a partial function $f : \mathbb{N} \to E$ with a finite domain of cardinality n and ordered as $i_1 < i_2 < \cdots < i_n$, the *collapse* of f, denoted $f\!\downarrow$, is the total function with domain $[1; n]$ such that $f\!\downarrow\!(m) = f(i_m)$ for any integer $1 \leq m \leq n$.

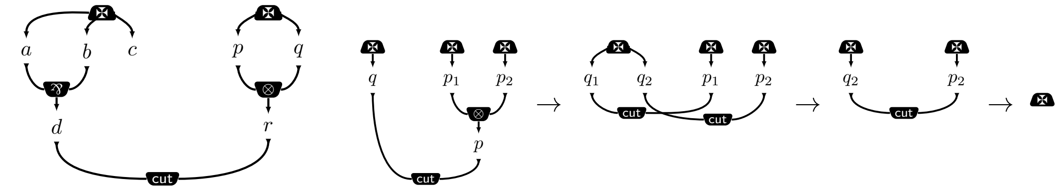
▶ **Definition 41.** *Let $S = (|S|, \mathbf{a}(S))$ and $T = (|T|, \mathbf{a}(T))$ be two nets and $k = min(\#S, \#T)$, we define their* interaction *$S :: T = (|S :: T|, \mathbf{a}(S :: T))$ as:*

$$|S :: T| \triangleq |S| + |T| + \textstyle\sum_{1 \leq i \leq min(\#S,\#T)} \langle S(i), T(i) \rhd_{\mathsf{cut}} \rangle \qquad \mathbf{a}(S :: T) \triangleq \begin{cases} \emptyset & \text{when } \#S = \#T \\ \mathbf{a}(S) \restriction_{[k+1;\#S]} \downarrow & \text{when } \#S > \#T \\ \mathbf{a}(T) \restriction_{[k+1;\#T]} \downarrow & \text{when } \#S < \#T \end{cases}$$

---

[6] Recall that a multigraph is a graph where two vertices may be connected by several edges (not to be confused with the notion of hypergraph of Definition 1). The function brd maps each edge to its endpoints.

[7] We refer to the graph naturally induced by the net σS.



(a) Representation of the interaction $S :: T$ of two nets $S = \langle \rhd_{\maltese}\ a, b, c\rangle + \langle a, b \rhd_{\parr} d\rangle$ and $T = \langle \rhd_{\maltese}\ p, q\rangle + \langle p, q \rhd_{\otimes} r\rangle$.

(b) The cut elimination procedure applied to $S' :: T'$ leads to $\maltese_0$, showing that $S' \perp T'$. In this figure $S' = \langle \rhd_{\maltese}\ q\rangle$ and $T' = \langle \rhd_{\maltese}\ p_1\rangle + \langle \rhd_{\maltese}\ p_2\rangle + \langle p_1, p_2 \rhd_{\otimes} p\rangle$.

Figure 9 The interaction of two nets (Definition 41) and two orthogonal nets (Definition 42).

▶ **Definition 42.** *Two nets S_1 and S_2 are* orthogonal *if $S_1 :: S_2 \to^* \maltese_0$[8]: when this holds we write $S_1 \perp S_2$. For a net S and a set of nets Λ, if for every $\lambda \in \Lambda$ we have $S \perp \lambda$ we write $S \perp \Lambda$.*

▶ Remark 43. Since cut links are asymmetric, namely $\langle p, q \rhd_{\mathsf{cut}}\rangle$ and $\langle q, p \rhd_{\mathsf{cut}}\rangle$ are distinct nets, the interactions $S :: T$ and $T :: S$ are not the same net. However, this has no consequence on cut elimination because the reduction steps do not depend on the order of the inputs of a cut link. Thus $S :: T$ reduces to $\maltese_0$ if and only if $T :: S$ does, and as expected the relation of orthogonality is symmetric.

▶ **Definition 44.** *Given a set A of multiplicative nets, we define the* orthogonal *of A as $A^\perp = \{P \mid \forall R \in A, P \perp R\}$. A* type $\mathbf{A}$ *is a set of multiplicative nets such that $\mathbf{A}^{\perp\perp} = \mathbf{A}$.*[9]

▶ Remark 45. Since cut elimination preserves the conclusions of a net and $\maltese_0$ has no output, two orthogonal nets have the same number of conclusions. Thus, for every type $\mathbf{A}$, for every $R \in \mathbf{A}$ and for every $S \in \mathbf{A}^\perp$, the nets R and S have the same number of conclusions: we denote by $\#\mathbf{A}$ the number of conclusions of the nets in $\mathbf{A}$. Obviously $\#\mathbf{A} = \#\mathbf{A}^\perp$.

▶ Remark 46. Clash cuts are preserved during cut elimination, thus a net containing such a cut cannot reduce to $\maltese_0$. Hence, there cannot be two nets S and S' respectively in $\mathbf{A}$ and $\mathbf{A}^\perp$ such that their ith conclusions $S(i)$ and $S'(i)$ are both outputs of a $\parr$–link (or $\otimes$–link): their interaction $S :: S'$ contains a clash cut and thus the nets cannot be orthogonal.

▶ Remark 47. A net S which is orthogonal to the daimon link with a single output (i.e. $\maltese_1$) has a single conclusion which can be the output of a daimon link, a tensor link or a par link. For instance the three cut–free nets $\langle \rhd_{\maltese}\ p\rangle$, $\langle \rhd_{\maltese}\ p_1\rangle + \langle \rhd_{\maltese}\ p_2\rangle + \langle p_1, p_2 \rhd_{\otimes} p\rangle$ and $\langle \rhd_{\maltese}\ p_1, p_2\rangle + \langle p_1, p_2 \rhd_{\parr} p\rangle$ are all orthogonal to $\maltese_1$ (one case is proved in Figure 9b).

The following proposition is a key step for proving propositions 51 and 54.

▶ **Proposition 48.** *Given three net S and T and R such that $\#S \geq \#T + \#R$: the interaction $S :: (T \parallel R)$ is equal to $(S :: T) :: R$.*

In the following definition 49 the side condition $\#S \geq \#\mathbf{A}$ ensures that whenever a net S in $A \succ B$ interacts with a net of $T \in \mathbf{A}^\perp$ the remaining conclusions of $S :: T$ are conclusions of S, this will allow to activate Proposition 48.

[8] Note that we require the *existence* of such a reduction, not all reductions need to behave this way.
[9] Equivalently, a type is a set $\mathbf{A}$ such that $\mathbf{A} = B^\perp$ for some set B, see, for instance, [11].

▶ **Definition 49.** *Given two sets of nets* $\mathbf{A}$ *and* $\mathbf{B}$ *their* functional composition *denoted* $\mathbf{A} \succ \mathbf{B}$, *and their* parallel composition *denoted* $\mathbf{A} \parallel \mathbf{B}$ *are defined as follows:*

$$\mathbf{A} \succ \mathbf{B} \triangleq \{S \mid \textit{for any } T \in \mathbf{A}^{\perp}, S :: T \in \mathbf{B} \textit{ and } \#S \geq \#\mathbf{A}\} \qquad \mathbf{A} \parallel \mathbf{B} \triangleq \{S \parallel T \mid S \in \mathbf{A}, T \in \mathbf{B}\}^{\perp\perp}$$

▶ Remark 50 (Density of the parallel composition). For any two types $\mathbf{A}$ and $\mathbf{B}$ we have $(\mathbf{A} \parallel^{-} \mathbf{B})^{\perp} = (\mathbf{A} \parallel \mathbf{B})^{\perp}$, where $\mathbf{A} \parallel^{-} \mathbf{B} = \{S \parallel T \mid S \in \mathbf{A}, T \in \mathbf{B}\}$.

▶ **Proposition 51** (Duality). *Given two types* $\mathbf{A}$ *and* $\mathbf{B}$*:* $(\mathbf{A} \parallel \mathbf{B})^{\perp} = \mathbf{A}^{\perp} \succ \mathbf{B}^{\perp}$ *and* $(\mathbf{A} \succ \mathbf{B})^{\perp} = \mathbf{A}^{\perp} \parallel \mathbf{B}^{\perp}$.

▶ Remark 52. The duality of the constructions (Proposition 51) ensures that the set of types is closed under the $\parallel$ and $\succ$ operations. Moreover, the intersection of two types is still a type. This is not the case for the union which needs to be closed under bi–orthogonal.

▶ Remark 53. For two types $\mathbf{A}$ and $\mathbf{B}$ the unordered nets of $\mathbf{A} \parallel \mathbf{B}$ and of $\mathbf{B} \parallel \mathbf{A}$ are the same, so as the unordered nets of $\mathbf{A} \succ \mathbf{B}$ and $\mathbf{B} \succ \mathbf{A}$.

▶ **Proposition 54.** *Given* $\mathbf{A}, \mathbf{B}$ *and* $\mathbf{C}$ *three types;* $(\mathbf{A} \succ \mathbf{B}) \succ \mathbf{C} = \mathbf{A} \succ (\mathbf{B} \succ \mathbf{C})$ *and* $(\mathbf{A} \parallel \mathbf{B}) \parallel \mathbf{C} = \mathbf{A} \parallel (\mathbf{B} \parallel \mathbf{C})$.

▶ **Definition 55.** *Given* $\mathbf{A}$ *and* $\mathbf{B}$ *two types with one conclusion, we define their* tensor product *(denoted* $\otimes$*) and their* compositional product *(denoted* $⅋$*):*

$$\mathbf{A} \otimes \mathbf{B} \triangleq \{S + \langle S(1), S(2) \rhd_{\otimes} p\rangle \mid S \in \mathbf{A} \parallel \mathbf{B}\}^{\perp\perp} \qquad \mathbf{A} ⅋ \mathbf{B} \triangleq \{S + \langle S(1), S(2) \rhd_{⅋} p\rangle \mid S \in \mathbf{A} \succ \mathbf{B}\}^{\perp\perp}$$

where p *denotes a fresh position.*

▶ **Proposition 56** (Duality). *Given* $\mathbf{A}$ *and* $\mathbf{B}$ *two types with one conclusion,* $(\mathbf{A} \otimes \mathbf{B})^{\perp} = \mathbf{A}^{\perp} ⅋ \mathbf{B}^{\perp}$ *and* $(\mathbf{A} ⅋ \mathbf{B})^{\perp} = \mathbf{A}^{\perp} \otimes \mathbf{B}^{\perp}$.

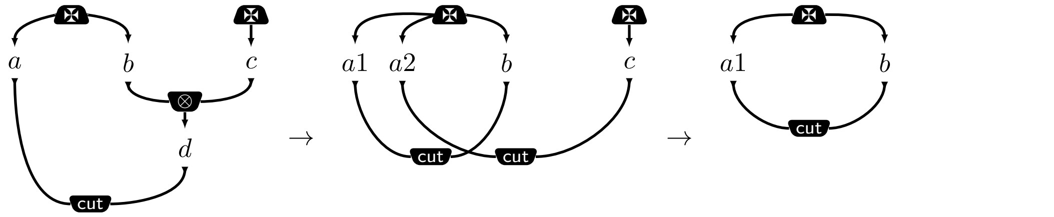
4 Realisability Model: Adequacy

We introduce our realisability model on untyped nets and prove it is adequate. We identify a sufficient property of interpretation bases to prove adequacy (Theorem 64): for any basis $\mathcal{B}$ satisfying the property, a net S representing an MLL✠ proof of a sequent Γ is a realiser of Γ i.e. it belongs to $[\![\Gamma]\!]_{\mathcal{B}}$. This adequacy result immediately applies to MLL, since a net representing a proof of MLL represents, in particular, a proof of MLL✠.

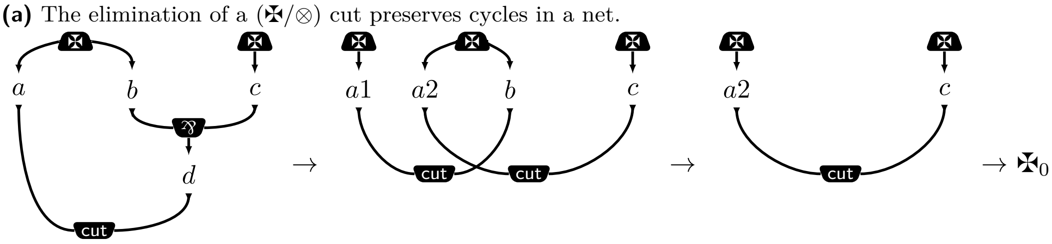
We start by giving an interpretation of formulas and hypersequents of multiplicative linear logic. We provide an interpretation of hypersequents instead of sequents as it turns out that handling hypersequents is more convenient and proving a result on hypersequents proves it on sequents too. However, do keep in mind that the proof trees we defined using Figure 6c are constructed with sequents.

▶ **Definition 57.** *An* interpretation basis $\mathcal{B}$ *is a function that associates with each atomic proposition* X *a type* $[\![X]\!]_{\mathcal{B}}$*, the* interpretation *of* X*, such that:*

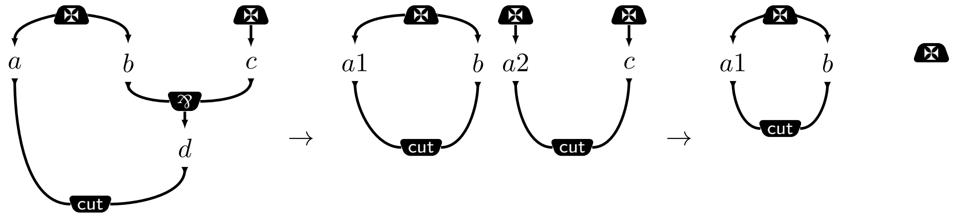
- *Each net in* $[\![X]\!]_{\mathcal{B}}$ *has one conclusion.*
- *For any atomic proposition* X*, we have* $[\![X^{\perp}]\!]_{\mathcal{B}} \subseteq [\![X]\!]_{\mathcal{B}}^{\perp}$.



(a) The elimination of a ($\maltese/\otimes$) cut preserves cycles in a net.



(b) The elimination of a ($\maltese/\parr$) cut *may* break a cycle in a net.



(c) The elimination of a ($\maltese/\parr$) cut can preserve cycles.

Figure 10 The evolution of (switching) cycles and (switching) disconnections during non homogeneous cut elimination.

▶ **Definition 58.** *Given an interpretation basis $\mathcal{B}$, the* interpretation *of* MLL *formulas and of hypersequents of* MLL *is defined by induction:*

$$[\![A \otimes B]\!]_{\mathcal{B}} \triangleq [\![A]\!]_{\mathcal{B}} \otimes [\![B]\!]_{\mathcal{B}}. \qquad [\![\mathcal{H}_1, \mathcal{H}_2]\!]_{\mathcal{B}} \triangleq [\![\mathcal{H}_1]\!]_{\mathcal{B}} \succ [\![\mathcal{H}_2]\!]_{\mathcal{B}}.$$
$$[\![A \parr B]\!]_{\mathcal{B}} \triangleq [\![A]\!]_{\mathcal{B}} \parr [\![B]\!]_{\mathcal{B}}. \qquad [\![\mathcal{H}_1 \parallel \mathcal{H}_2]\!]_{\mathcal{B}} \triangleq [\![\mathcal{H}_1]\!]_{\mathcal{B}} \parallel [\![\mathcal{H}_2]\!]_{\mathcal{B}}.$$

▶ Remark 59. Using duality of types (Proposition 56) and the properties of orthogonality one proves that for an interpretation basis $\mathcal{B}$ and an MLL formula A we have $[\![A^{\perp}]\!]_{\mathcal{B}} \subseteq [\![A]\!]_{\mathcal{B}}^{\perp}$.

▶ **Definition 60.** *A multiplicative net* realises *– with respect to an interpretation basis $\mathcal{B}$ – an hypersequent $\mathcal{H}$ of* MLL *formulas whenever it belongs to $[\![\mathcal{H}]\!]_{\mathcal{B}}$.*

▶ Notation 61. For a hypersequent $\mathcal{H}$, we will often write $S \Vdash_{\mathcal{B}} \mathcal{H}$ instead of $S \in [\![\mathcal{H}]\!]_{\mathcal{B}}$, and sometimes $S \Vdash \mathcal{H}$ or $S \in [\![\mathcal{H}]\!]$ when there is no ambiguity on the basis $\mathcal{B}$.

From the point of view of cut elimination, a daimon link with n outputs may be thought as the approximation of a proof net with n outputs. More precisely, by iterating the process we have seen in Remark 24, every cut–free proof π of a formula C can be obtained by applying the cut elimination procedure to the daimon link $\maltese_1$ (of conclusion C) cut against the appropriate identities of $C, C^{\perp}$ (this generalises to a sequent Γ and $\maltese_n$). Furthermore daimon links and proof nets (with the same number of conclusions) are interchangeable with respect to geometrical correctness (Table 1): in a correct (resp. incorrect) net S, substituting a daimon link with n outputs by a proof net with n outputs produces a correct (resp. incorrect) net. However, proof nets and daimons (with the same number of conclusions) differ on realisability: for instance a proof net ending with a tensor link can never realise a formula of the form $A \parr B$ whereas a daimon link can (Theorem 64). We will thus say that a daimon link "approximates" a sequent: this suggests Definition 62.

▶ **Definition 62.** *A type* $\mathbf{A}$ *is approximable if and only if* $✠_1 \in \mathbf{A}$*. A basis* $\mathcal{B}$ *is* approximable *if for each* $X \in \mathsf{Var}$*, the type* $[\![X]\!]_{\mathcal{B}}$ *is approximable.*

▶ Remark 63. Because inclusion is preserved by bi–orthogonal closure, a type $\mathbf{A}$ is approximable if and only if $\{✠_1\}^{\perp\perp} \subseteq \mathbf{A}$ which is equivalent to the inclusion $\mathbf{A}^{\perp} \subseteq \{✠_1\}^{\perp}$.

▶ **Theorem 64** (Adequacy)**.** *Let* $\mathcal{B}$ *be an approximable basis. For any net* S *and sequent* Γ $S \vdash_{\mathsf{MLL}^{✠}} \Gamma \Rightarrow S \Vdash_{\mathcal{B}} \Gamma$*.*

Proof. The technique is standard in the works on realisability (see [12] or [14]): one proceeds by induction on the size of a proof π represented by S. For the base case one must show that $✠_n$ realises any sequent Γ with n formulas. To do so one first checks that, for any formula A, $[\![A]\!]_{\mathcal{B}}$ is approximable ($✠_1 \in [\![A]\!]_{\mathcal{B}}$). ◀

▶ Remark 65. An approximable basis yields adequacy, in particular, for MLL. Notice, however, that there exist bases yielding an interpretation that is adequate for MLL but not for $\mathsf{MLL}^{✠}$.

5 Testability and tests

The partitions involved in the Danos Regnier criterion (Theorem 76) and their orthogonality with the daimons of a net can be translated as *tests*; so that for a formula A, a net S testable by A (definition 66 below) and orthogonal to $\mathsf{tests}(A)$ is a correct net (Theorem 76). We will show that these tests are proofs of $\mathsf{MLL}^{✠}$ (Theorem 77). This means that for realisers in an approximable basis, testability (Definition 66) and correct typeability (Definition 31) coincide: this is Proposition 82.

▶ **Definition 66** ((Atomic) testable cut–free nets)**.** *A* formula labelling *of a cut–free net* S *is a function* $\tau : V_S \to \mathsf{Form}$ *such that:*

- ***(Par)*** *When* $\langle p_1, p_2 \rhd_{⅋} p\rangle$ *occurs in* S*: if* $\tau(p_1) = A$ *and* $\tau(p_2) = B$ *then* $\tau(p) = A ⅋ B$*.*
- ***(Tens)*** *When* $\langle p_1, p_2 \rhd_{\otimes} p\rangle$ *occurs in* S*: if* $\tau(p_1) = A$ *and* $\tau(p_2) = B$ *then* $\tau(p) = A \otimes B$*.*

A formula labelling of a cut–free net S *is* atomic *when for each daimon link* $\langle \rhd_{✠} p_1, \dots, p_n\rangle$ *in* S *the formula* $\tau(p_i)$ *is a propositional variable.*

A cut–free net S *with* n *conclusions is* testable *(resp.* atomic testable*) by a sequent* $\Gamma = A_1, \dots, A_n$*, which we denote* $S \models \Gamma$ *(resp.* $S \overset{at}{\models} \Gamma$*), if there exists a formula (resp. an atomic formula) labelling* τ *of* S *such that* $\tau(S(i)) = A_i$ *for each* $1 \leq i \leq n$*.*

▶ Remark 67. $S \models \Gamma$ iff $S \overset{at}{\models} \Delta$ and $\Gamma = \theta\Delta$ for some substitution θ and sequent Δ.

▶ Remark 68. $S \overset{at}{\models} \Gamma$ iff S without its ✠–links is the syntactic forest of (the formulas of) Γ.

▶ Remark 69. A cut–free proof net $S \vdash^{✠}_{\mathsf{MLL}} \Gamma$ is in particular testable by that sequent i.e. $S \models \Gamma$. However, a net $S \models \Gamma$ which is testable by Γ may not be a proof net because it could contain cycles or disconnections: the testability condition only provides information on the multiplicative links constituting the net S. When is S atomic testable by A, orthogonality with the tests of A coincides with correctness (Proposition 75).

▶ Remark 70. Let $S \overset{at}{\models} A_1, \dots, A_n$ be a cut–free net. For any nets $T_1, \dots, T_n$ cut–free and atomically testable respectively by ${A_1}^{\perp}, \dots, {A_n}^{\perp}$ denoting S_0 the normal form of $S :: T_1 \parallel \cdots \parallel T_n$, S_0 is obtained by homogeneous cut–elimination, and we have (1) S_0 equals $✠_0$ (2) S_0 is equal to the sum of $k \geq 2$ daimon without conclusions ($S_0 = \sum_{1 \leq i \leq k} ✠_0$) or (3) S_0 contains a cyclic cut ($S_0 = R + \langle \rhd_{✠} \vec{q}, a, \vec{r}, b, \vec{p}\rangle + \langle a, b \rhd_{\mathsf{cut}}\rangle$).

▶ Remark 71. Given a net $S = (|S|, \mathbf{a}(S))$ we denote $S^{\maltese} = (|S^{\maltese}|, \mathbf{a}(S^{\maltese}))$ the net such that $|S^{\maltese}|$ is the hypergraph consisting of the daimon links occurring in S. The arrangement $\mathbf{a}(S^{\maltese})$ is induced by $\mathbf{a}(S)$ because above every conclusion of S there is binary tree: each initial position p can be associated with a sequence $\xi = \mathsf{adr}(p)$ of $\{\ell, r\}^*$ and an integer $i = \mathsf{root}(p)$ so that going up from $S(i)$ following the left/right instruction of ξ one reaches the initial position p. The initial positions of S are then ordered by the lexicographical order of $(\mathsf{root}(p), \mathsf{adr}(p))$ fixing $\ell \leq r$.

▶ Notation 72. Given a net S with n initial positions, and $P = \{C_1, \ldots, C_k\}$ a partition of the initial positions of S we denote by $\mathbf{Nat}_S(P)$ the partition $\{\mathbf{a}(S^{\maltese})^{-1}(C_1), \ldots, \mathbf{a}(S^{\maltese})^{-1}(C_k)\}$ of $\{1, \ldots, n\}$. We might abusively write $\mathbf{Nat}(P)$ for $\mathbf{Nat}_S(P)$.

▶ **Proposition 73.** *Let A be a formula, given two cut free nets $S \overset{at}{\vDash} A$ and $T \overset{at}{\vDash} A^{\perp}$ the assertions are equivalent:*
1. *The nets S and T are orthogonal.*
2. *The nets $S^{\maltese}$ and $T^{\maltese}$ are orthogonal.*
3. *The partition $\mathbf{Nat}_S(\mathsf{P}_{\maltese}(S))$ and $\mathbf{Nat}_T(\mathsf{P}_{\maltese}(T))$ are orthogonal.*

▶ **Definition 74.** *A cut-free net T is a* test *of a formula A if $T \overset{at}{\vDash} A^{\perp}$ and there exists a net $S \overset{at}{\vDash} A$ and a switching σS such that $\mathbf{Nat}_T(\mathsf{P}_{\maltese}(T)) = \mathbf{Nat}_S(\uparrow^i \sigma S)$. We denote by* $\mathsf{tests}(A)$ *the set $\{S \mid S$ is a test of $A\}$.*

▶ **Proposition 75.** *For S cut–free, $S \overset{at}{\vDash} A$, we have: $S \vdash_{\mathsf{MLL}^{\maltese}} A \Leftrightarrow S \perp \mathsf{tests}(A)$.*

A net S with n conclusion can always be transformed in a net with 1 conclusion by putting a bunch of par–links below its conclusions; this allows to generalise the previous proposition.

▶ **Theorem 76** (Danos–Regnier Tests). *Given a cut–free net $S \overset{at}{\vDash} A_1, \ldots, A_n$; $S \vdash_{\mathsf{MLL}^{\maltese}} A_1, \ldots, A_n$ if and only if S is orthogonal to $\mathsf{tests}(A_1) \parallel \cdots \parallel \mathsf{tests}(A_n)$.*

▶ **Theorem 77.** *Any test T of a formula A is correctly typeable by $A^{\perp}$, $T \vdash_{\mathsf{MLL}^{\maltese}} A^{\perp}$.*

Proof. Consider a test T of A then by Theorem 76 any net $S \vdash_{\mathsf{MLL}^{\maltese}} A$ is orthogonal to T. By the counter–proof criterion [4] a net $N \overset{at}{\vDash} A^{\perp}$ orthogonal to each proof of A is a proof; therefore it follows that T is a proof of $A^{\perp}$. ◀

▶ Remark 78. Theorem 76 is a refinement of the counter–proof criterion of P.L. Curien [4]: if $S \overset{at}{\vDash} A$ and $S \perp \mathsf{tests}(A)$ then $S \vdash_{\mathsf{MLL}^{\maltese}} A$ – and every element of $\mathsf{tests}(A)$ are proofs of $A^{\perp}$ (Theorem 77), but the converse does not hold.

From Theorem 76 and Theorem 77 one obtains an "interactive" criterion for the nets of multiplicative linear logic (MLL). One takes a net of S of MLL (i.e. a net with binary daimons) and confronts it with the tests of the according formulas (Definition 74). A straightforward consequence of the Theorem 76 is the reformulation of Béchet's theorem in our framework.

▶ **Corollary 79.** *Let $S \overset{at}{\vDash} A_1, \ldots, A_n$ be a cut–free net. If S is not correct then there exists nets $T_1 \in \mathsf{tests}(A_1), \ldots, T_n \in \mathsf{tests}(A_n)$ such that the normal form of $S :: T_1 \parallel \cdots \parallel T_n$ is not correct: we are in case (2) or (3) of Remark 70.*

▶ Remark 80. The Corollary 79 obviously applies to MLL nets, the main difference with Béchet's original result is that his opponents are MLL proof nets (in our framework they are $\mathsf{MLL}^{\maltese}$ proof nets). However it is not difficult to adapt our techniques to obtain Béchet's result.

▶ Remark 81. Consider an approximable basis $\mathcal{B}$ and a sequent $\Gamma = A_1, \ldots, A_n$ we have $[\![\Gamma]\!]_\mathcal{B} = ([\![A_1]\!]_\mathcal{B}^\perp \parallel \cdots \parallel [\![A_n]\!]_\mathcal{B}^\perp)^\perp$. By Theorem 64, for any $A_i^\perp$ we have $\{\!\!\{A_i^\perp : \mathsf{MLL}^{\maltese}\}\!\!\} \subseteq [\![A_i^\perp]\!]_\mathcal{B}$ while $\mathsf{tests}(A_i) \subseteq \{\!\!\{A_i^\perp : \mathsf{MLL}^{\maltese}\}\!\!\}$ (Theorem 77) thus $\mathsf{tests}(A_i) \subseteq [\![A_i^\perp]\!]_\mathcal{B} \subseteq [\![A_i]\!]_\mathcal{B}^\perp$ (Remark 59). Because the $\parallel$–construction preserves inclusions and orthogonality inverts inclusions we derive that $[\![\Gamma]\!]_\mathcal{B} \subseteq (\mathsf{tests}(A_1) \parallel \cdots \parallel \mathsf{tests}(A_n))^\perp$.

Remark 81 combined with the previous theorem (Theorem 76) means that for realisers in an approximable basis, testability and (correct) typeability collapse.

▶ **Proposition 82.** *Given $\mathcal{B}$ an approximable basis*[10] *and a sequent Γ for any cut–free net $S \in [\![\Gamma]\!]_\mathcal{B}$ the assertions are equivalent:*

1. $S \models \Gamma$ *i.e.* $S \models^{at} \Delta$ *for some sequent* $\Delta \leq \Gamma$.
2. $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma$.

6 Completeness

Using Proposition 82 we provide a completeness result; we exhibit an approximable basis for which a net S realising a sequent Γ is testable, and so equivalently $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma$. This basis, denoted $\mathbf{1}$, maps each atomic formula to $\{\maltese_1\}^{\perp\perp}$.

▶ **Proposition 83.** *For any sequent Γ and any cut–free net S; if $S \in [\![\Gamma]\!]_\mathbf{1}$ then $S \models \Gamma$.*

▶ Remark 84. By the Proposition 83 and the Theorem 64 we have that $S \in [\![\Gamma]\!]_\mathbf{1}$ iff $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma$.

Since the base $\mathbf{1}$ is approximable, Proposition 82 allows to prove:

▶ **Theorem 85** ($\mathsf{MLL}^{\maltese}$ completeness)**.** *Given a cut–free net S and a sequent Γ;*

- *If for all basis $\mathcal{B}$ we have $S \in [\![\Gamma]\!]_\mathcal{B}$, then $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma$.*
- *$S \in [\![\Gamma]\!]_\mathcal{B}$ for any approximable basis $\mathcal{B}$ iff $S \vdash_{\mathsf{MLL}^{\maltese}} \Gamma$.*

▶ Remark 86. The non homogeneous cut elimination allows to distinguish the types $[\![X, X^\perp]\!]_\mathcal{B}$ and $[\![X, Y]\!]_\mathcal{B}$ for a well chosen basis: for instance for the basis, that we will denote $\mathcal{B}\langle\parr\rangle$, which maps positive propositional variables to $\{\maltese_\parr\}^\perp$ and negative propositional variables to $\{\maltese_\parr\}^{\perp\perp}$, where $\maltese_\parr$ denotes the geometrically incorrect net $\langle \rhd_\maltese a\rangle + \langle \rhd_\maltese b\rangle + \langle a, b \rhd_\parr c\rangle$.

In that case, (1) because $\maltese_2$ is not orthogonal to $\maltese_\parr \parallel \maltese_\parr$ (Figure 11) it follows that $\maltese_2 \notin [\![X, X]\!]_{\mathcal{B}\langle\parr\rangle}$ and more generally $\maltese_2 \notin [\![X, Y]\!]_{\mathcal{B}\langle\parr\rangle}$; (2) by the property expressed in Remark 90 (and illustrated in Figure 12), $\maltese_2 \in [\![X, X^\perp]\!]_{\mathcal{B}\langle\parr\rangle}$; (3) point (1) above is not in contradiction with the theorem of adequacy (Theorem 64) because, even though $\maltese_2 \vdash_{\mathsf{MLL}^{\maltese}} X, Y$, the basis $\mathcal{B}\langle\parr\rangle$ is not approximable.

▶ Remark 87. The ability to distinguish realisers of the sequents $X, X^\perp$ and X, Y (Remark 86) allows us to derive the completeness result for MLL (Theorem 88) from the completeness result for $\mathsf{MLL}^{\maltese}$ (Theorem 85). In Remark 86, to show that $\maltese_2 \notin [\![X, Y]\!]_{\mathcal{B}\langle\parr\rangle}$ we have used incorrect nets (specifically $\maltese_\parr$), which explains that the completeness theorem for MLL (Theorem 88) refers to *any* basis $\mathcal{B}$ (and not only to approximable basis). In the terms of Table 1, we retrieve provability correctness by using interactions with geometrically incorrect nets.

▶ **Theorem 88** (MLL completeness)**.** *Let S be a cut–free net such that each of its daimon link has exactly two outputs, Γ be a sequent such that $S \models^{at} \Gamma$; if $S \in [\![\Gamma]\!]_\mathcal{B}$ for any basis $\mathcal{B}$ then, $S \vdash_{\mathsf{MLL}} \Gamma$.*

[10] The Proposition 82 actually holds for any “adequate” basis $\mathcal{B}$.

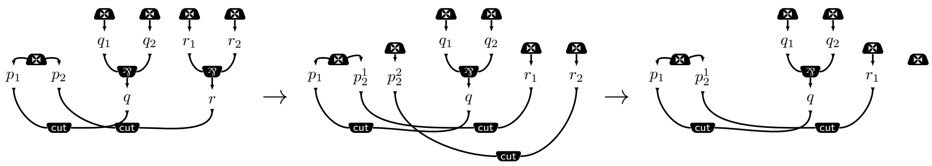


Figure 11 The daimon link $\maltese_2$ is not orthogonal to $\maltese_{\parr} \parallel \maltese_{\parr}$: a disconnected net never reduces to a connected one (and $\maltese_0$ is connected).

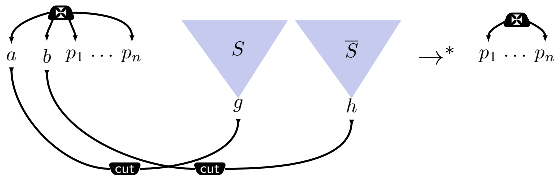


Figure 12 The interaction of two orthogonal nets S and $\overline{S}$ with a daimon reduces to a daimon (with two less outputs).

▶ Remark 89. A result of adequacy for MLL can also be stated: given an interpretation basis $\mathcal{B}$ (not necessarily approximable) such that for each propositional variable X we have $[\![X^\perp]\!]_\mathcal{B} = [\![X]\!]_\mathcal{B}^\perp$, for any net S, if $S \vdash_{\mathsf{MLL}} \Gamma$ then $S \in [\![\Gamma]\!]_\mathcal{B}$.

▶ Remark 90. The completeness result for MLL (Theorem 88) only identifies cut–free and *atomic* proofs (i.e. where axioms introduce sequents of the form $X, X^\perp$). This is because for any atomic formulas X and Y, and for any basis $\mathcal{B}$ such that $[\![X^\perp]\!]_\mathcal{B} = [\![X]\!]_\mathcal{B}^\perp$, $\maltese_2 \in [\![X \parr X^\perp, Y \parr Y^\perp]\!]_\mathcal{B}$ while $X \parr X^\perp$ and $Y \parr Y^\perp$ are not dual formulas: contrary to the atomic case we cannot use $\maltese_2$ to distinguish $[\![X \parr X^\perp, Y \parr Y^\perp]\!]_{\mathcal{B}\langle\parr\rangle}$ from $[\![X \parr X^\perp, X^\perp \otimes X]\!]_{\mathcal{B}\langle\parr\rangle}$.

The fact that $\maltese_2 \in [\![X \parr X^\perp, Y \parr Y^\perp]\!]_{\mathcal{B}\langle\parr\rangle}$ (and more generally for any basis $\mathcal{B}$ such that $[\![X^\perp]\!]_\mathcal{B} = [\![X]\!]_\mathcal{B}^\perp$) is derived from the fact that, for any integer k and for any two orthogonal nets S_1 and S_2 with one conclusion, the interaction $\maltese_{k+2} :: (S_1 \parallel S_2)$ has *at least one* reduction to $\maltese_k$ by cut elimination (Figure 12). We use this property for $k = 2$ and $k = 4$ to show that $\maltese_2 \in [\![X \parr X^\perp, Y \parr Y^\perp]\!]_\mathcal{B}$. More precisely, we prove that, $\maltese_2 \perp [\![X \parr X^\perp]\!]_\mathcal{B}^\perp \parallel [\![Y \parr Y^\perp]\!]_\mathcal{B}^\perp$: given $S, \overline{S}$ and $R, \overline{R}$ two pairs of orthogonal nets (with one conclusion), when all nets $S, \overline{S}, R, \overline{R}$ have disjoint sets of vertices, we can derive the following:

$$\begin{array}{rl} & \langle \rhd_{\maltese}\ a, b\rangle :: S + \overline{S} + \langle S(1), \overline{S}(1) \rhd_\otimes q\rangle + R + \overline{R} + \langle R(1), \overline{R}(1) \rhd_\otimes r\rangle \\ \to \cdot \to & \langle \rhd_{\maltese}\ a_1, a_2, b_1, b_2\rangle :: S + \overline{S} + R + \overline{R} \\ \to^* & \langle \rhd_{\maltese}\ b_1, b_2\rangle :: R + \overline{R} \\ \to^* & \maltese_0 \end{array}$$

References

1 Denis Bechet. Minimality of the correctness criterion for multiplicative proof nets. *Mathematical Structures in Computer Science*, 8(6):543–558, 1998. URL: `http://journals.cambridge.org/action/displayAbstract?aid=44779`, `doi:10.1017/S096012959800262X`.

2 Emmanuel Beffara. A concurrent model for linear logic. *Electronic Notes in Theoretical Computer Science*, 155:147–168, 2006. Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI). `doi:10.1016/j.entcs.2005.11.055`.

3 Emmanuel Beffara, Félix Castro, Mauricio Guillermo, and Étienne Miquey. Concurrent realizability on conjunctive structures. In Marco Gaboardi and Femke van Raamsdonk, editors, *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023, July 3-6, 2023, Rome, Italy*, volume 260 of *LIPIcs*, pages 28:1–28:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSCD.2023.28.

4 Pierre-Louis Curien. Introduction to linear logic and ludics, part II. *CoRR*, abs/cs/0501039, 2005. arXiv:cs/0501039.

5 Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28(3):181–203, October 1989. doi:10.1007/BF01622878.

6 Valeria C. V. de Paiva. A dialectica-like model of linear logic. In David H. Pitt, David E. Rydeheard, Peter Dybjer, Andrew M. Pitts, and Axel Poigné, editors, *Category Theory and Computer Science*, pages 341–356, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. doi:10.1007/BFB0018360.

7 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:10.1016/0304-3975(87)90045-4.

8 Jean-Yves Girard. Multiplicatives. In G. Lolli, editor, *Logic and Computer Science: New Trends and Applications*, pages 11–34. Rosenberg & Sellier, 1987.

9 Jean-Yves Girard. Proof-nets: The parallel syntax for proof-theory. In *Logic and Algebra*, pages 97–124. Marcel Dekker, 1996.

10 Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. In Laurent Fribourg, editor, *Computer Science Logic*, pages 38–38, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

11 Jean-Baptiste Joinet and Thomas Seiller. From abstraction and indiscernibility to classification and types: revisiting hermann weyl's theory of ideal elements. *Kagaku tetsugaku*, 53(2):65–93, 2021. doi:10.4216/jpssj.53.2_65.

12 Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2005. URL: https://hal.science/hal-00154500.

13 International Research Network (IRN) Linear Logic. *Handbook of Linear Logic*. International Research Network (IRN) Linear Logic, 2023. URL: https://ll-handbook.frama.io/ll-handbook/ll-handbook-public.pdf.

14 Paulo Oliva. Modified realizability interpretation of classical linear logic. In *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 431–442, 2007. doi:10.1109/LICS.2007.32.

15 Thomas Seiller. Interaction graphs: Multiplicatives. *Annals of Pure and Applied Logic*, 163(12):1808–1837, 2012. doi:10.1016/j.apal.2012.04.005.

16 Thomas Seiller. Interaction graphs: Exponentials. *Log. Methods Comput. Sci.*, 15, 2013.

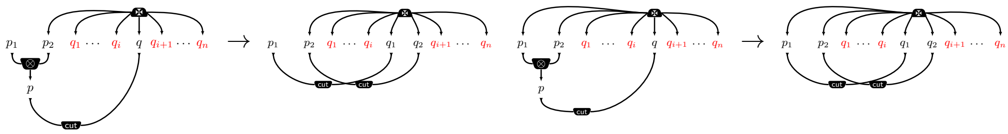
17 Thomas Seiller. Interaction graphs: Full linear logic. *CoRR*, abs/1504.04152, 2015. arXiv:1504.04152.

18 Thomas Seiller. Interaction graphs: Additives. *Annals of Pure and Applied Logic*, 167(2):95–154, 2016. doi:10.1016/j.apal.2015.10.001.

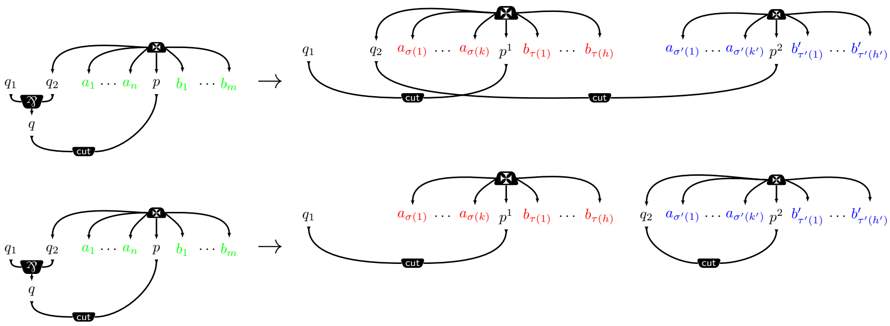
19 Thomas Seiller. Interaction graphs: Graphings. *Annals of Pure and Applied Logic*, 168(2):278–320, 2017. doi:10.1016/j.apal.2016.10.007.

20 Thomas Seiller. Mathematical informatics, 2024. Habilitation thesis. URL: https://theses.hal.science/tel-04616661.

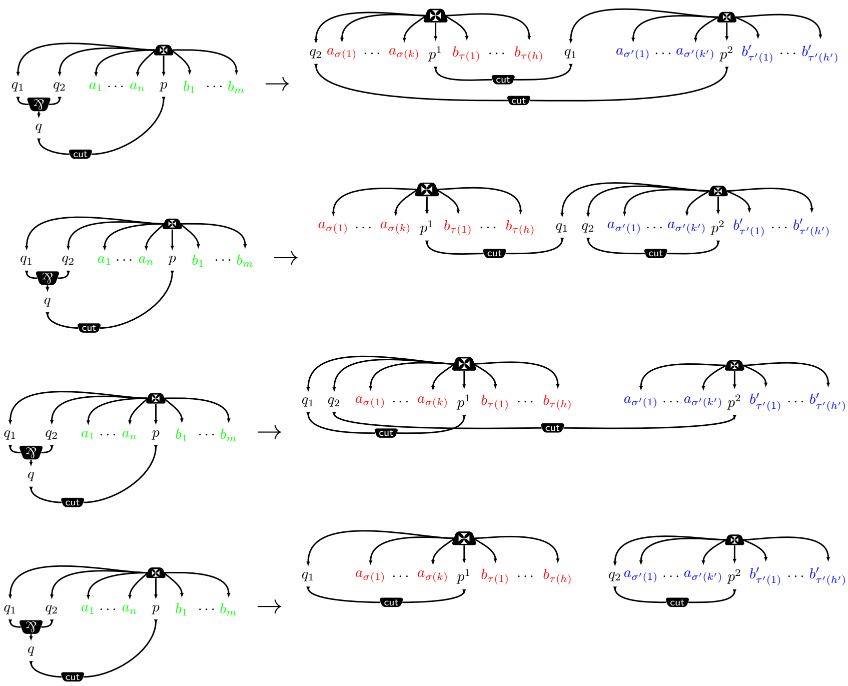
A Additional Figures



(a) Extra cases for the elimination of (✠/⊗) cuts, on the left the elimination step when one of the inputs belongs to the daimon above the cut, on the right the elimination step when both inputs belong to the daimon above the cut.



(b) Extra cases for the elimination of (✠/⅋) cuts: when one of the inputs belongs to the daimon above the cut.



(c) Extra cases for the elimination of (✠/⅋) cuts: when both inputs belong to the daimon above the cut.

Figure 13 Complements to Figure 4 for defining non homogeneous cut elimination (Definition 22).