The proof equivalence problem for multiplicative linear logic is PSPACE-complete v0.3

Willem Heijltjes and Robin Houston · January 13, 2014

Abstract

MLL proof equivalence is the problem of deciding whether two proofs in multiplicative linear logic are related by a series of inference permutations. Previous work has shown the problem to be equivalent to a rewiring problem on proof nets, which are not canonical for full MLL due to the presence of the two units. Drawing from recent work on reconfiguration problems, in this paper it is shown that MLL proof equivalence is PSPACE-complete, using a reduction from Nondeterministic Constraint Logic.

$$\dfrac{\Gamma}{\Gamma,\bot}\,\bot \qquad \dfrac{}{1}\,1 \qquad \dfrac{\Gamma,A,B}{\Gamma,A\parr B}\,\parr \qquad \dfrac{\Gamma,A \qquad \Delta,B}{\Gamma,\Delta,A\otimes B}\,\otimes$$

Figure 1: Inference rules for unit-only MLL

$$\dfrac{\dfrac{\Gamma}{\Gamma,\bot_a}\,\bot}{\Gamma,\bot_a,\bot_b}\,\bot \sim \dfrac{\dfrac{\Gamma}{\Gamma,\bot_b}\,\bot}{\Gamma,\bot_a,\bot_b}\,\bot \qquad \dfrac{\dfrac{\Gamma,A,B}{\Gamma,A\parr B}\,\parr}{\Gamma,A\parr B,\bot}\,\bot \sim \dfrac{\dfrac{\Gamma,A,B}{\Gamma,A,B,\bot}\,\bot}{\Gamma,A\parr B,\bot}\,\parr$$

$$\dfrac{\dfrac{\Gamma,A \qquad \Delta,B}{\Gamma,\Delta,A\otimes B}\,\otimes}{\Gamma,\Delta,A\otimes B,\bot}\,\bot \sim \dfrac{\dfrac{\Gamma,A}{\Gamma,A,\bot}\,\bot \qquad \Delta,B}{\Gamma,\Delta,A\otimes B,\bot}\,\otimes \sim \dfrac{\Gamma,A \qquad \dfrac{\Delta,B}{\Delta,B,\bot}\,\bot}{\Gamma,\Delta,A\otimes B,\bot}\,\otimes$$

$$\dfrac{\dfrac{\Gamma,A,B,C,D}{\Gamma,A\parr B,C,D}\,\parr}{\Gamma,A\parr B,C\parr D}\,\parr \sim \dfrac{\dfrac{\Gamma,A,B,C,D}{\Gamma,A,B,C\parr D}\,\parr}{\Gamma,A\parr B,C\parr D}\,\parr$$

$$\dfrac{\dfrac{\Gamma,A \qquad \Delta,B,C,D}{\Gamma,\Delta,A\otimes B,C,D}\,\otimes}{\Gamma,\Delta,A\otimes B,C\parr D}\,\parr \sim \dfrac{\Gamma,A \qquad \dfrac{\Delta,B,C,D}{\Delta,B,C\parr D}\,\parr}{\Gamma,\Delta,A\otimes B,C\parr D}\,\otimes$$

$$\dfrac{\Gamma,A \qquad \dfrac{\Delta,B,C \qquad \Lambda,D}{\Delta,\Lambda,B,C\otimes D}\,\otimes}{\Gamma,\Delta,\Lambda,A\otimes B,C\otimes D}\,\otimes \sim \dfrac{\dfrac{\Gamma,A \qquad \Delta,B,C}{\Gamma,\Delta,A\otimes B,C}\,\otimes \qquad \Lambda,D}{\Gamma,\Delta,\Lambda,A\otimes B,C\otimes D}\,\otimes$$

Figure 2: Permutations

1 MLL

The formulae of unit-only multiplicative linear logic are given by the following grammar.

$$A,B,C := \bot \mid 1 \mid A\parr B \mid A\otimes B$$

The connectives $\otimes$ and $\parr$ will be considered up to associativity, and *duality* $A^\star$ is via DeMorgan. A *sequent* Γ,Δ will be a multiset of formulae. Within a sequent, connectives and units will be *named* with distinct elements from an arbitrary set of names N, e.g. $1_a \parr_b 1_c, \bot_d \otimes_e \bot_f$. This allows to 1) avoid using the notion of *occurrence*, and instead refer to subformulae by the name of their root connective, as e.g. A_b, 2) distinguish the two proofs of the above sequent while using standard multiset sequents, and 3) easily extract proof nets, as graphs using the names of connectives as vertices. Names will mostly be left implicit.

Proofs are constructed from the inference rules in Figure 1. The names of connectives are preserved through inferences. Only cut-free proofs are considered, and no cut-rule is added. *Permutations* of inference rules are displayed in Figure 2; the symmetric variants of the last two permutations, *par-tensor* and *tensor-tensor*, have been omitted.

Definition 1. *Equivalence* of proofs ($\sim$) in (cut-free, unit-only) multiplicative linear logic is the congruence generated by the permutations given in Figure 2. *MLL proof equivalence* is the problem of deciding whether two given proofs are equivalent.

The motivation to consider proofs up to equivalence is three-fold. Firstly, there is the strong intuition that the order of permutable inferences does not contribute to the essential content of the proof. Secondly, a technical motivation is that cut-elimination in MLL incorporates permutation steps, and composition via cut-elimination is only associative up to permutations. Thirdly, equivalent proofs are identified in natural models of multiplicative linear logic such as coherence spaces, and in the categorical semantics of MLL, $\star$-autonomous categories.

In one of several possible definitions, a *$\star$-autonomous category* (Barr, 1979) is a symmetric monoidal category $(\mathcal{C},\otimes,1)$ with:

- a *duality*, a contravariant functor $-^\star$ such that $A\cong A^{\star\star}$, and

- *closure*, an adjunction $- \otimes B \dashv (B \otimes -^\star)^\star$ for any object B,

satisfying natural coherence conditions. The category with as objects unit-only MLL-formulae and as morphisms $A \to B$ the equivalence classes of proofs of $A^\star \parr B$, denoted $\text{MLL}(\varnothing)$, is a $\star$-autonomous category. The present formulation of formulae induces two forms of *strictness*, instances where isomorphisms of the definition are identities: DeMorgan duality means $A = A^{\star\star}$, while associativity is an identity by decree. Modulo strictness, $\text{MLL}(\varnothing)$ is the *free* $\star$-autonomous category over the empty category $\varnothing$. This means that *any* $\star$-autonomous category is a model of the logic, and that MLL proof equivalence is the *word problem* for $\star$-autonomous categories, the problem of deciding when two term representations denote the same morphism.

1.1 Proof nets

A partial solution to the MLL proof equivalence problem is provided by proof nets.

Definition 2. For a sequent Γ,

- a *linking* ℓ is a function from the names of $\bot$-subformulae to the names of 1-subformulae,
- a *switching graph* for ℓ is an undirected graph over the names of Γ, with for every subformula $A_a \otimes_c B_b$ the edges $a - c$ and $b - c$, for every subformula $A_a \parr_c B_b$ either the edge $a - c$ or the edge $b - c$, and for every subformula $\bot_a$ the edge $a - \ell(a)$,
- a *proof net* ℓ or (Γ, ℓ) is a linking ℓ such that every switching graph is acyclic and connected.

An edge $a - \ell(a)$ in a proof net or switching graph is a *link* or *jump*.

Definition 3. A *permutation* between proof nets is the redirection of exactly one link. *Equivalence* $(\sim)$ of proof nets over a sequent Γ is the congruence generated by permutations.

There is no canonical interpretation of a proof as a proof net, since the introduction rule for $\bot$ in proofs joins a $\bot$-formula to a sequent, rather than a formula.

Definition 4. The relation $(\Mapsto)$ interprets a proof Π for a sequent Γ by a linking ℓ as follows: $\Pi \Mapsto \ell$ if for each $\bot_a$ in Γ, if Δ is the context of the inference introducing $\bot_a$, as illustrated below, then $\ell(a)$ is the name of some 1 in Δ.

$$\frac{\Delta}{\Delta, \bot_a}\,\bot$$

Proposition 5 (Danos and Regnier, 1989). *For a proof* Π *with conclusion* Γ*, if* $\Pi \Mapsto \ell$ *then* ℓ *is a proof net for* Γ*. For a net* ℓ *for* Γ*, there is a proof* Π *of* Γ *such that* $\Pi \Mapsto \ell$ *(*sequentialisation*).*

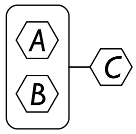
Proof nets are canonical representations of proofs in the absence of units: they factor out the permutations among tensor- and par-inferences, which are the last three permutations in Figure 2. Equivalence of proof nets is generated by the remaining equations, the permutations on $\bot$-introduction.

Proposition 6 (Hughes, 2012). *For proofs* Π*,* Π' *and proof nets* ℓ*,* ℓ' *such that* $\Pi \Mapsto \ell$ *and* $\Pi' \Mapsto \ell'$*,* $\Pi \sim \Pi'$ *if and only if* $\ell \sim \ell'$*.*

MLL proof equivalence is the problem of deciding equivalence of proof nets.

1.2 Notation

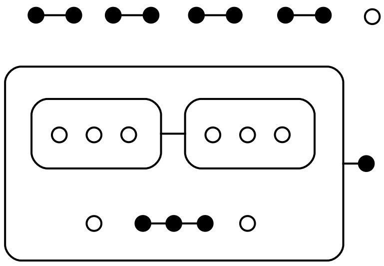
We will use a concise diagrammatic notation for sequents and proof nets. The units 1 and $\bot$ are represented by a circle $\circ$ and a disc $\bullet$ respectively. A tensor is represented by a line connecting both subformulae, and a par by juxtaposition: if A and B are represented by $\langle A \rangle$ and $\langle B \rangle$, then $A \parr B$ is $\langle A \rangle\ \langle B \rangle$ and $A \otimes B$ is $\langle A \rangle\!-\!\langle B \rangle$. A tensor of multiple elements is denoted by stringing them together in a line, so $A \otimes B \otimes C$ is $\langle A \rangle\!-\!\langle B \rangle\!-\!\langle C \rangle$. Boxes play the role of parentheses around par-formulae, so $(A \parr B) \otimes C$ is drawn as



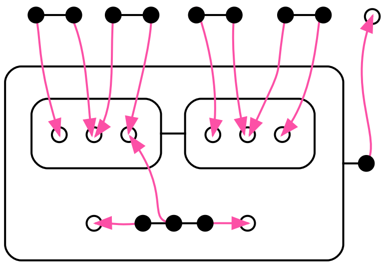
For example, this sequent

$$\vdash \bot \otimes \bot, \bot \otimes \bot, \bot \otimes \bot, \bot \otimes \bot, 1, \{[(1 \parr 1 \parr 1) \otimes (1 \parr 1 \parr 1)] \parr 1 \parr (\bot \otimes \bot \otimes \bot) \parr 1\} \otimes \bot$$

could be drawn like this:



We represent a proof net by drawing an arrow from each $\bullet$ to some $\circ$. For example, one proof net on the above sequent is



2 Equivalence in the absence of ⅋

Let a *1-alternation* sequent be one over formulae of the form 1 or $\bot \otimes \ldots \otimes \bot$, where the number of $\bot$-subformulae is at least 2. Such a sequent is inhabited exactly when the number of formulae in the sequent is one greater than the total number of $\bot$-subformulae it contains. An inhabited 1-alternation sequent with only one tensor-formula, i.e. a sequent of the form $1, \ldots, 1, \bot \otimes \ldots \otimes \bot$ with n $\bot$-subformulae and n 1-subformulae, will admit $n!$ different proof nets, each with n links. Since no link can re-attach, its equivalence classes are singletons.

Proposition 7. *For a 1-alternation sequent with at least two tensor-formulae there are at most two equivalence classes of proof nets.*

Proof. It will be shown by induction on the number of $\bot$-formulae in Γ that every proof net for Γ belongs to one of two equivalence classes. For the base case, the smallest inhabited sequent with two tensor-formulae is the following.

$$1, 1, 1, \bot \otimes \bot, \bot\otimes, \bot$$

It has two equivalence classes, of 12 proof nets each. (Apart from listing these exhaustively, this can also be shown by using the proof of the inductive step, below, to reduce the base case to that of the sequent $1, 1, \bot \otimes \bot$, which has two singleton equivalence classes.)

For the inductive step, let Γ be the following sequent.

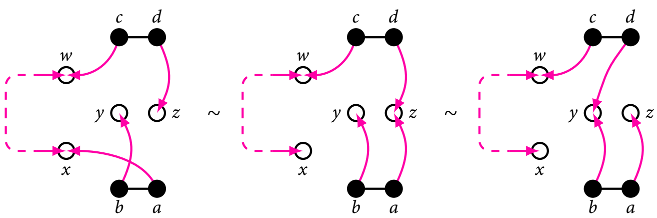
$$\Delta, A \otimes \bot_a, 1_z$$

There are two cases: 1) where A is a tensor-formula, and 2) where A is $\bot$ and where, for the induction hypothesis to apply, Δ contains at least two tensor-formulae. For both cases, it will be shown that any net ℓ for Γ is equivalent to a net ℓ' where $\bot_a$ connects to 1_z, and is the only link to do so. This reduces equivalence on Γ to equivalence on Δ, A in case 1, and on Δ in case 2, so that the induction hypothesis applies.

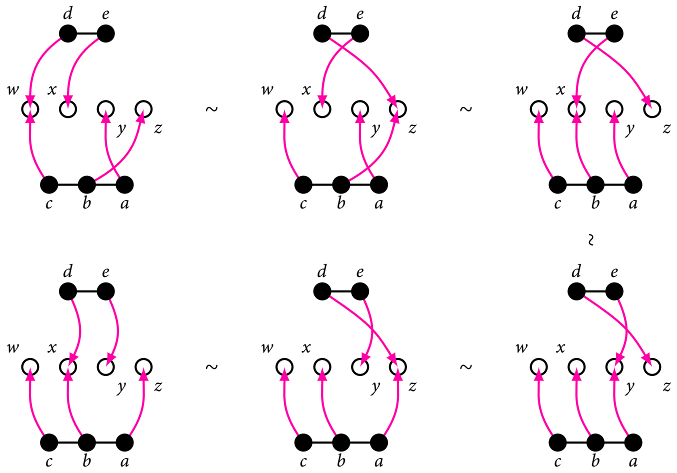
In constructing ℓ', since the proof net must be connected, there is a path from a to z. We will consider only the jumps on this path, not any other edges. There are four cases.

i. The path consists of exactly the jump $a - z$. Let $b - y$ be a jump from a $\bot_b$ in A; then ℓ' is obtained from ℓ by changing: $\ell'(e) = y$ for every $\bot_e$ such that $\ell(e) = z$ and $e \neq a$.

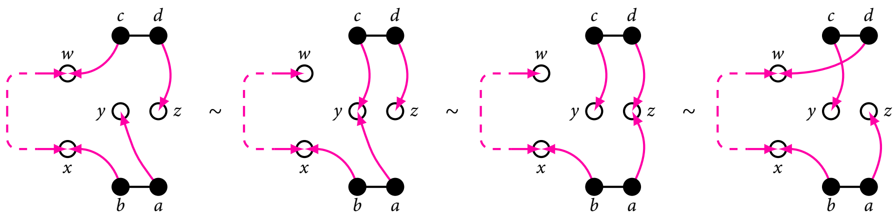
ii. The path starts with the jump $a - x$ (and $x \neq z$). Let $b - y$ be a jump from a $\bot_b$ in A, and let the path end with the jumps $w - c$ and $d - z$, where $\bot_c$ and $\bot_d$ are in the same tensor-formula B. Then ℓ' is obtained from ℓ by changing: $\ell'(a) = z$, and $\ell'(e) = w$ for every $\bot_e$ such that $\ell(e) = z$ and $e \neq a$, including d. These changes are illustrated as permutations below.



iii. The path consists of exactly one jump $b-z$ from a $\bot_b$ in A (and $b \neq a$). Let $\ell(a) = y$. Choose a jump $c - w$ from $\bot_c$ in A such that there is another $d - w$ from $\bot_d$ in a formula B (not excluding the possibilities $c = b$ and $c = a$). Let $e - x$ be a further jump from $\bot_e$ in B. Then ℓ' is obtained from ℓ by changing: $\ell'(a) = z$, $\ell'(d) = x$, $\ell'(e) = y$, and $\ell'(f) = x$ for each f (including b) such that $\ell(f) = z$ and $f \neq a$. These changes are exhibited as a series of permutations below, from top left to bottom left (note that the jump from d moves twice).



iv. The path starts with a jump $b - x$ from a $\bot_b$ in A (and $b \neq a$, $x \neq z$). Let the path end with the jumps $w - c$ and $d - z$, and let $\ell(a) = y$. Then ℓ' is obtained from ℓ by changing: $\ell'(c) = y$, $\ell'(a) = z$, and $\ell'(e) = w$ for every $\bot_e$ such that $e \neq a$ and $\ell(e) = z$, including d. This is illustrated below.



□

Proposition 8. *For a proof net for a 1-alternation sequent containing a link $\bot_a - 1_b$ and a formula 1_c, the following are equivalent.*

- *The edge $a - b$ can be permuted to $a - c$.*
- *There is a path from b to c not passing through a.*
- *The path from a to c starts with the jump $a - b$.*

If a link $a - b$ may be reconnected as $a - c$ it is said that *a may connect to c*. By the above proposition, it is immediate that if a and b may both connect to c, then after actually reconnecting $a - c$, still b may connect to c.

Consider the following naming scheme for the units in a 1-alternation sequent Γ with tensor-formulae $A_1, \ldots, A_n$.

- One 1 in Γ is named $\star$, and the remaining ones with the numbers $n + 1, \ldots, m$.
- A $\bot$-formula in A_i is named by a pair (i, k), where $k = i$ for the first $\bot$-formula in each A_i, and for the remaining $\bot$-formulae in all A_i, each k is a distinct number in $n + 1, \ldots, m$.

The naming scheme suggests a linking for Γ, defined by $\ell(i, i) = \star$ and $\ell(i, k) = k$ otherwise; i.e the first $\bot$ in each tensor-formula connects to $1_\star$, while other $\bot$-subformulae connect uniquely to the remaining 1-subformulae.

A net for Γ is interpreted as a combinatorial permutation (an automorphism on $\{1, \ldots, m\}$) as follows.

Definition 9. To a proof net ℓ for a 1-alternation sequent Γ named as above, associate the *permutation* $p_\ell : \{1, \dots, m\} \to \{1, \dots, m\}$ given by:

$$p_\ell(k) = \begin{cases} i & \text{if } (i,k) \text{ may connect to } *\text{; and} \\ \ell(i,k) & \text{otherwise.} \end{cases}$$

The *parity* of ℓ is the parity of its permutation.

To see that p_ℓ is injective, consider the following.

- The domains of i and $\ell(i,k)$, respectively $1, \dots, n$ and $n+1, \dots, m$, are disjoint.
- Exactly one $\bot$-formula in each A_i may connect to $*$ because of connectedness and acyclicity, since if a $\bot$-formula may connect to $*$ it has a path to $*$ (Proposition 8).
- If two $\bot$-formulae have the same target, which means they are in different tensor-formulae, at least one may connect to $*$ via the other tensor-formula, which must have a path to $*$ by the above.

Proposition 10. *A permutation on a net ℓ preserves its parity.*

Proof. Let ℓ be a net for Γ, with Γ named as above, and let the link $(i,k) - x$ in ℓ re-attach as $(i,k) - y$, forming ℓ'. There are two cases, depending on whether (i,k) may connect to $*$. If so, using Proposition 8, the re-wiring preserves which $\bot$-formulae may connect to $*$, since for any path to $*$ via $(i,k) - x$ in ℓ there is a path to $*$ via $(i,k) - y$. Then the permutation of ℓ' is that of ℓ.

If (i,k) may not connect to $*$, let the path from x to y run via the following $\bot$- and 1-vertices.

$$x = x_1, (i_1, j_1), (i_1, k_1), x_2, (i_2, j_2), \dots, (i_n, k_n), x_{n+1} = y$$

Note that the $\bot$-formulae (i_a, j_a) may connect to $*$. On the relevant domain, this gives the following permutation for ℓ.

$$\begin{pmatrix} j_1 & \dots & j_n & k & k_1 & \dots & k_n \\ i_1 & \dots & i_n & x_1 & x_2 & \dots & x_{n+1} \end{pmatrix}$$

In ℓ', since $\ell'(i,k) = y$, the $\bot$-formulae that may connect to $*$ are the (i_a, k_a). The permutation $p_{\ell'}$ is the following.

$$\begin{pmatrix} j_1 & \dots & j_n & k & k_1 & \dots & k_n \\ x_1 & \dots & x_n & x_{n+1} & i_1 & \dots & i_n \end{pmatrix}$$

The parity of both permutations is the same if and only if the relative permutation, below, is even.

$$\begin{pmatrix} i_1 & \dots & i_n & x_1 & x_2 & \dots & x_{n+1} \\ x_1 & \dots & x_n & x_{n+1} & i_1 & \dots & i_n \end{pmatrix}$$

This is the case, as it is obtained by the exchange of x_a and i_a for each $a \leq n$, and subsequently the exchange of x_{n+1} and each i_a in turn. $\square$

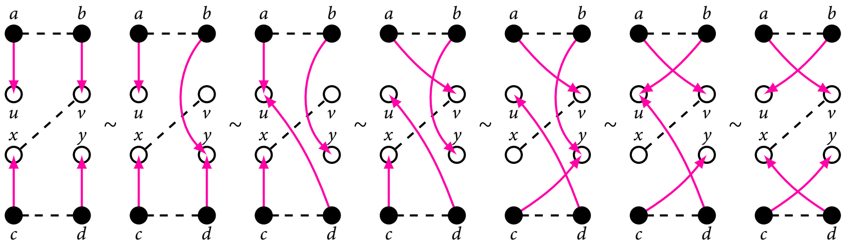
Proposition 11. *Two proof nets for a 1-alternation sequent with at least two tensor-formulae are equivalent if and only if they have the same parity.*

Theorem 12. *MLL proof equivalence in the absence of $\parr$ is linear-time decidable.*

Proof. For a sequent with 1 tensor-formula, the problem is reduced to syntactic equality. For a sequent with 2 or more tensor-formulae, by Propositions 11 the equivalence of two nets is determined by their parity. Following Definition 9 the parity of a net can be read off in a single traversal of the net. This yields a linear-time algorithm. $\square$

Lemma 13. *Let ℓ be a proof net where $\ell(a) = (u)$, $\ell(b) = v$, $\ell(c) = x$, and $\ell(d) = y$, and for any switching the path from u to y passes through the links $b - v$ and $x - c$. Then $\ell \sim \ell'$ where $\ell'(a) = v$, $\ell'(b) = u$, $\ell'(c) = y$, and $\ell'(d) = x$, and $\ell'(z) = \ell(z)$ otherwise.*

Proof.



a b a b a b a b a b a b a b
u v u v u v u v u v u v u v
x y x y x y x y x y x y x y
c d c d c d c d c d c d c d
~ ~ ~ ~ ~ ~

□

3 Par

Definition 14. A *sub-sequent* $\Delta \leq \Gamma$ of a sequent Γ is a sequent consisting of disjoint subformulae of Γ, preserving names.

Definition 15. A *subnet* $(\Gamma', \ell') \leq (\Gamma, \ell)$ of a proof net is a net such that $\Gamma' \leq \Gamma$ and ℓ' is $(\ell|_{\Gamma'})$, the restriction of ℓ to Γ'.

The root vertices of Γ' are the *ports* of the sub-sequent Γ' and of the subnet (Γ', ℓ'). A *scope* of a par $\parr_v$ is a subnet that has v as a port. In a proof net, the *kingdom* and the *empire* of a par are respectively its smallest and largest scope.

The scopes of a par correspond to the possible subproofs of its introduction rule in a sequentialisation of the proof net that it occurs in. In the graph of a proof net, the scope of a par $\parr_v$ may be *contracted* to a single vertex v by removing all vertices except v and re-attaching all arcs connecting to removed vertices to v.

[[ADD ILLUSTRATED EXAMPLE]]

The contraction of scopes may replace the switching condition as a correctness criterion. The following is a variant of the local retraction algorithm by Danos **?**.

Proposition 16. *A linking ℓ for a sequent Γ is a proof net if and only if each $\parr_v$ is a port of a sub-sequent $s(v) \leq \Gamma$ such that:*

1. *sub-sequents are strictly nested: if $\parr_w$ occurs in $s(v)$ then $s(w) < s(v)$; and*
2. *for each $\parr_v$, the graph $\sigma(v) = (s(v), \ell|_{s(v)})$ becomes a tree when all immediate sub-sequents $s(w)$ are contracted.*

Proof. For the 'if' direction, it follows by induction on the nesting of sub-sequents that each graph $\sigma(v)$ satisfies the switching condition. For the 'only if' direction, given a sequentialisation of (Γ, ℓ), a sub-sequent $s(v) \leq \Gamma$ for each $\parr_v$ is found by taking the conclusion $\Delta, A \parr_v B$ of its introduction rule, below.

$$\frac{\Delta, A, B}{\Delta, A \parr_v B}$$

□

[[IDEA: the following could help simplify octopus-arithmetic]]

Definition 17. The *balance* of a sequent is the number of $\bot$s minus the number of $\parr$s and comma's. A sequent is *balanced* if its balance is zero.

An unbalanced sequent is uninhabited: a positive balance guarantees a cycle in any switching graph, for any linking, while a negative balance similarly guarantees disconnectedness.

An early conjecture of Girard, which turned out to be false, was that a sequent is inhabited if and only if it is balanced. [[FIND CITATION (probably TCS87)]]

4 Proof nets and constraint graphs

[[NOTE: we should decide on notation for steps versus paths in permutation relations]]

Definition 18. A *non-deterministic constraint graph* (NCG) $G = (V, E, c, v, w)$ consists of a set V of vertices with *minimum inflow constraint* $c\colon V \to \mathbb{N}$, and a set E of at most one undirected edge e per vertex-pair $v(e) = \{v_1, v_2\}$, with *weight* $w\colon E \to N$.

A (partial) *configuration* of an NCG is a (partial) function $\gamma\colon E \to V$ such that

- for every edge e, $\gamma(e) \in v(e)$ or (in the partial case) $\gamma(e)$ is undefined, and
- for every vertex v, the sum of its inflow weights is at least its inflow constraint, $\sum\{w(e) \mid \gamma(e) = v\} \geq c(v)$.

A *reconfiguration step* $\gamma \sim \delta$ connects two (partial) configurations for an NCG G that differ in the assignment of exactly one edge.

The *(partial) NCG-reconfiguration problem* is the problem of deciding when two (partial) configurations are connected by a path in the graph of (partial) configurations and reconfiguration steps.

Theorem 19 (?). *NCG-reconfiguration is PSPACE-complete.*

Proposition 20. *Partial NCG-reconfiguration is PSPACE-complete.*

Proof. There is a path between total configurations γ and δ in partial NCG-reconfiguration if and only if there is one in NCG-reconfiguration, for the following two reasons. Firstly, if $\gamma \sim \delta$ are partial configurations, they may be completed to total configurations $\gamma' \sim \delta'$ or $\gamma' = \delta'$. Secondly, if γ' and γ'' are total configurations that both agree with a partial configuration γ where it is defined, then γ' and γ'' are connected in NCG-reconfiguration by re-assigning the values where they disagree. □

For a graph $G = (V, E, c, v, w)$ let $|V|$ and $|E|$ denote the number of vertices and edges, respectively, and let $|c|$ and $|w|$ denote the sum of all inflow constraints, $\sum_{v \in V} c(v)$, and the sum of all edge weights, $\sum_{e \in E} w(e)$.

Let A^n denote the sequent of n copies of a formula A, and for a sequent $\Gamma = A_1, \ldots, A_n$ let $\bigotimes \Gamma = A_1 \otimes \ldots \otimes A_n$ and $\bigparr \Gamma = A_1 \parr \ldots \parr A_n$.

Definition 21. The *interpretation* $[\![G]\!]$ of an NCG $G = (V, E, c, v, w)$ is a sequent constructed as follows. Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ where $|V| = n$ and $|E| = m$.

The interpretation of a vertex v_k is the formula

$$[\![v_k]\!] = \bigparr \left(C^{m \times c(v_k)}\right) \parr 1$$

where each *constraint element* C is the formula

$$C = \bigparr \left(1^{3k+2}\right) \otimes \bigparr \left(1^{3(n-k)+3}\right)$$

The interpretation of an edge e connecting vertices v_i and v_j with $i < j$ is the formula

$$[\![e]\!] = \bigotimes \left(W^{m \times w(e)}\right) \otimes \bot$$

where each *weight element* W is the formula

$$W = \bigotimes \left(\bot^{3i+2}\right) \parr \bigotimes \left(\bot^{3(j-i)+1}\right) \parr \bigotimes \left(\bot^{3(n-j)+3}\right)$$

The interpretation of the graph G is the sequent

$$[\![G]\!] = [\![v_1]\!] \otimes \ldots \otimes [\![v_n]\!], [\![e_1]\!], \ldots, [\![e_m]\!], 1^p$$

where $p = m \times (|w| - |c|) \times (3n + 4)$; the p instances of 1 are called *edge-absorbers*.

In an NCG G, a vertex v and an edge e will be called *appropriate* (for each other) if $v \in v(e)$, and *inappropriate* otherwise. This notion is extended to vertex-gadgets $\llbracket v \rrbracket$ and edge-gadgets $\llbracket e \rrbracket$ in $\llbracket G \rrbracket$.

For a weight element W of an edge connecting v_i and v_j, let the $\bot$-occurrences be named as follows,

$$W = \left(\bot_{\dagger} \otimes \bot_1 \otimes \ldots \otimes \bot_{3i+1}\right) \parr \left(\bot_{\ddagger} \otimes \bot_{3i+2} \otimes \ldots \otimes \bot_{3j+1}\right) \parr \left(\bot_{\text{⸸}} \otimes \bot_{3j+2} \otimes \ldots \otimes \bot_{3n+3}\right)$$

and the 1-occurrences of a constraint element C in $\llbracket v_k \rrbracket$,

$$C = \left(1_{\underline{\dagger}} \parr 1_{\underline{1}} \parr \ldots \parr 1_{\underline{3k+1}}\right) \otimes \left(1_{\underline{\text{⸸}}} \parr 1_{\underline{3k+2}} \parr \ldots \parr 1_{\underline{3n+3}}\right) .$$

There is a *natural linking* for the sequent W, C if e is appropriate for v_k, i.e. if $k = i$ or $k = j$, as follows:

$$\ell(x) = \underline{x} \qquad \text{for } x \in \{\dagger, \text{⸸}\} \cup \mathbb{N}$$

$$\ell(\ddagger) = \begin{cases} \underline{\dagger} & \text{if } k = i \\ \underline{\text{⸸}} & \text{if } k = j . \end{cases}$$

There is also a natural linking for the sequent $W, 1_{\star}, 1_{\underline{1}}, \ldots, 1_{\underline{3n+3}}$ consisting of the weight element W and $3n + 4$ edge-absorbers:

$$\ell(x) = \begin{cases} \star & \text{if } x \in \{\dagger, \ddagger, \text{⸸}\} \\ \underline{x} & \text{otherwise} . \end{cases}$$

Proposition 22. 1. *Given a vertex v and an appropriate edge e, for a constraint element C in $\llbracket v \rrbracket$ and a weight element W in $\llbracket e \rrbracket$ the natural linking for the sequent W, C is a proof net.*

2. *Given a weight element W for a graph with $|V| = n$, the natural linking for the sequent $W, 1^{3n+4}$ is a proof net.*

The rightmost 1-occurrence of each vertex-gadget $\llbracket v \rrbracket$, named $\underline{v}$ below, and the rightmost $\bot$-occurrence of each edge-gadget $\llbracket e \rrbracket$, named $\underline{e}$, will be called the *indicator vertices* of $\llbracket v \rrbracket$ and $\llbracket e \rrbracket$.

$$\llbracket v \rrbracket = C \parr \ldots \parr C \parr 1_{\underline{v}} \qquad \llbracket e \rrbracket = W \otimes \ldots \otimes W \otimes \bot_{\underline{e}} .$$

Definition 23. The *interpretation* $\llbracket \gamma \rrbracket$ of a total configuration γ for a graph G is a linking ℓ constructed incrementally, for each successive edge e, and for each successive weight element W within e, as follows. Let $\gamma(e) = v$; firstly, the the indicator vertex of $\llbracket e \rrbracket$ links to the indicator of $\llbracket v \rrbracket$. Then successively for each weight element W in e, if $\llbracket v \rrbracket$ has a first free constraint element C, extend $\llbracket \gamma \rrbracket$ to include the natural linking on W, C; otherwise, extend $\llbracket \gamma \rrbracket$ by the natural linking on the sequent consisting of W plus the first $3n + 4$ free edge absorbers.

Proposition 24. *If γ is a total configuration for G then $\llbracket \gamma \rrbracket$ is a proof net for $\llbracket G \rrbracket$.*

Proof. Using Proposition 16, it is sufficient to give a suitable scope for each $\parr$. The scope of each weight element W is the sequent W, C or $W, 1^{3n+4}$ of its natural linking, which forms a proof net by Proposition 22. The scope of each vertex-gadget $\llbracket v \rrbracket$ contains the edge-gadgets $\llbracket e \rrbracket$ such that $\gamma(e) = v$, plus all the edge-absorbers within scopes of weight elements inside $\llbracket e \rrbracket$. Since the weights of the connected edges e sum to more than the inflow constraint of v, there are no unused constraint elements remaining in $\llbracket v \rrbracket$. After contracting the scope of each W, each edge-gadget in the scope of $\llbracket v \rrbracket$ becomes a single string of connected vertices, connected to other edge-gadgets only via the special 1_v of $\llbracket v \rrbracket$, thus forming a tree. $\square$

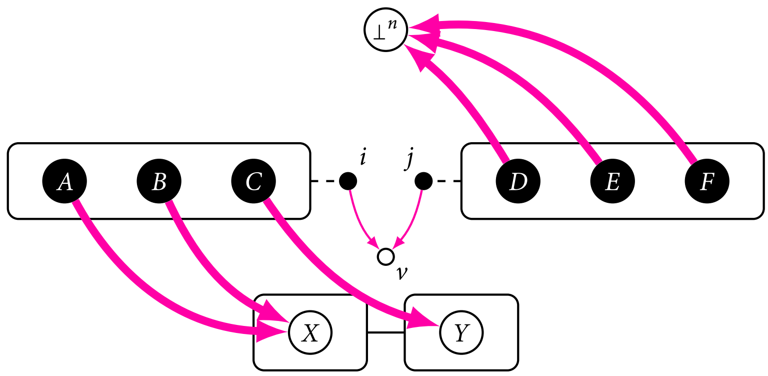
4.1 Completeness

In a proof net for $[\![G]\!]$, an edge-gadget $[\![e]\!]$ that is in the empire of an appropriate vertex-gadget $[\![v]\!]$ is *naturally linked* if the indicator of $[\![e]\!]$ connects to the indicator of $[\![v]\!]$, and each weight element W of $[\![e]\!]$ is either naturally linked to a constraint element C in $[\![v]\!]$ or linked only to edge-absorbers.

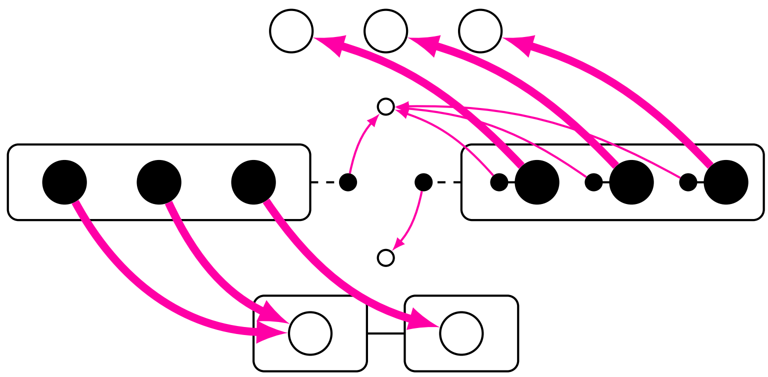
Lemma 25. *In a proof net ℓ for a sequent $\Gamma = [\![v]\!], [\![e_1]\!], \dots, [\![e_m]\!], \bot^p$ where each edge-gadget is naturally linked, if a weight element W_i in $[\![e_i]\!]$ is linked to C in $[\![v]\!]$ and W_j in $[\![e_j]\!]$ is linked to edge-absorbers $\bot^n$, then there is a net $\ell' \sim \ell$ in which W_j is naturally linked to C, W_i is linked to $\bot^n$, and ℓ' agrees with ℓ otherwise.*

Proof. Let $W_i = A \parr B \parr C$, $W_j = D \parr E \parr F$, and $C = X \otimes Y$. We will illustrate the path of permutations for the case where A, B, X and D, X, and thus also C, Y and E, F, Y, are balanced sequents; other cases are similar.

1. The initial configuration is illustrated below; other weight and constraint elements are omitted, and v, i, and j are the indicator vertices of $[\![v]\!]$, $[\![e_i]\!]$, and $[\![e_j]\!]$ respectively.

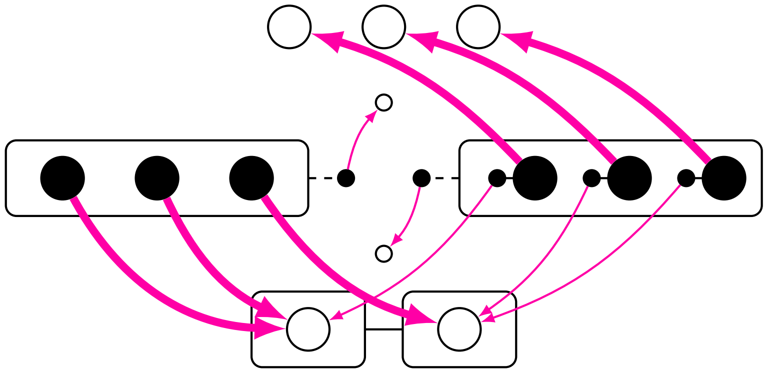


2. The link $i - v$ is re-attached to connect to an edge-absorber together with only the links from the first $\bot$ of each of D, E, and F.

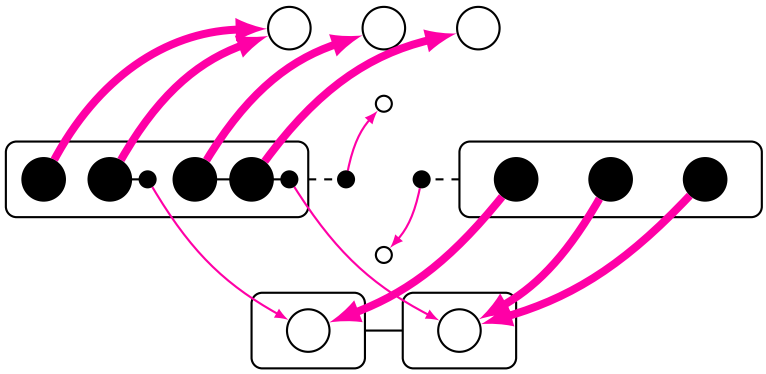


3. Secondly, the link from the first $\bot$ of D is moved to X, and those of E and F are

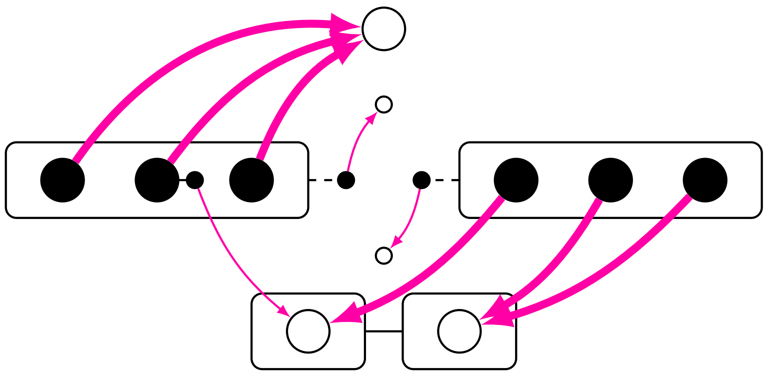
moved to Y.



4. There are subnets over the sub-sequents $A, B, X, D, \bot^m$ and $C, Y, E, F, \bot^k$. These subnets may be rewired so that $D \parr E \parr F$ is naturally linked to $X \otimes Y$: by Proposition 11, the resulting subnets are equivalent as long as their parity is preserved. Two links from C to X should remain exchanged, compared to the natural linking, for step 6 below. The links of A, B, C connect to the edge-absorbers $\bot^m$ and $\bot^k$, with one remaining link from B to X and one from C to Y.

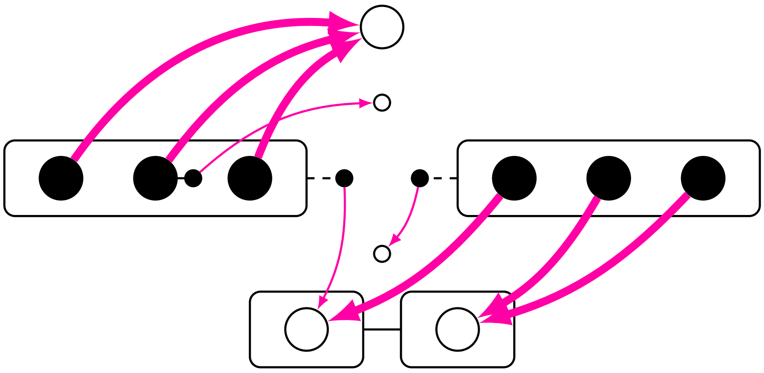


5. The link from C to Y is moved towards an edge-absorber connected to A, B.

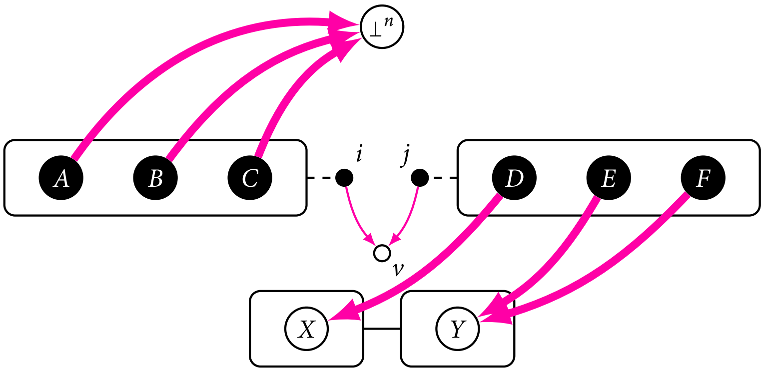


6. The link from B to X is the one remaining connection between the edge-gadgets $[\![e_i]\!]$ and $[\![e_j]\!]$. Lemma 13 allows to swap the targets of the link from B to X and the link from i, and simultaneously undo the exchange in the links from C to X added

in step 4 above.



7. The link from i may be re-attached to v to yield the final configuration.



□

An edge-gadget $[\![e]\!]$ is *free* if each of its weight elements is linked only to edge-absorbers.

Lemma 26. *If ℓ is a naturally linked proof net for $[\![G]\!]$ with a free edge-gadget $[\![e]\!]$, and ℓ' agrees with ℓ up to an even permutation of edge-absorbers, then $\ell \sim \ell'$.*

Proof. Let e_0 be the indicator vertex of $[\![e]\!]$; since $[\![e]\!]$ is free, e_0 may re-attach anywhere within the proof net. Let e_1 and e_2 be arbitrary other $\perp$-occurrences in $[\![e]\!]$.

1. To permute two edge-absorbers v and w linked to by other edge-gadgets than $[\![e]\!]$, attach e_0 to v, and apply Lemma 13 to exchange v and w, also exchanging the targets of e_0 and e_1.

2. To reinstate e_0 as the connection between $[\![e]\!]$ and the remainder of the proof net, either perform another permutation as above, or permute v and w twice again, once exchanging the targets of e_1 and e_2, and once exchanging the targets of e_2 and e_0.

3. To exchange an edge-absorber w linked from a $\perp_d$ outside $[\![e]\!]$ for the target v of e_1, and attach e_0 to w, attach d to v. At this point, e_1 forms the only connect between $[\![e]\!]$ and the remainder of the proof net; to reinstate e_0 in this role, do as above, producing a net effect of cycling the targets of e_1, e_2, and d.

4. Finally, to permute edge-absorbers v and w linked by $[\![e]\!]$, attach e_0 to the indicator of a vertex-gadget where also another edge-gadget $[\![d]\!]$ is connected, with indicator vertex d_0 and an arbitrary other vertex d_1. Connect d_0 to v, and apply Lemma 13 to permute v and w, as well as exchanging the targets of d_0 and d_1. A second such exchange is needed to re-attach d_0 and d_1 to their original targets.

In each case, if one of the edge-absorbers exchanged is linked to by multiple $\bot$-occurrences within the same weight element, these may be termporarily attached elsewhere.

□

Let $[\![\gamma]\!]'$ be $[\![\gamma]\!]$ where the first two edge-absorbers are exchanged.

Lemma 27. *If $\gamma \sim \delta$ for total configurations γ and δ, then either $[\![\gamma]\!] \sim [\![\delta]\!]$ or $[\![\gamma]\!] \sim [\![\delta]\!]'$.*

Proof. We will prove the case where $\gamma \sim \delta$ is a single reconfiguration step; the general case follows because the proof goes through also when $[\![\gamma]\!]'$ replaces $[\![\gamma]\!]$ in the statement of the lemma. Let γ and δ agree on every edge except e, where $\gamma(e) = v$ and $\delta(e) = w$. Firstly, using Lemma 25, for the edges d other than e such that $\gamma(d) = v$, the weight elements of the edge-gadgets $[\![d]\!]$ may be linked to the constraint elements of $[\![v]\!]$, in accordance with the target configuration $[\![\delta]\!]$. Since e is mobile in γ, the weights of the edges d suffice to fill the inflow constraint of v, and correspondingly the weight elements of edge-gadgets $[\![d]\!]$ suffice to fill the constraint elements of $[\![v]\!]$, so that $[\![e]\!]$ is free. Next, the indicator vertex of $[\![e]\!]$, which links to the indicator of $[\![v]\!]$, is re-attached to the indicator of $[\![w]\!]$. Again using Lemma 25, the weight elements of edge-gadgets connected to $[\![w]\!]$, including $[\![e]\!]$, may be linked in accordance with $[\![\delta]\!]$. The resulting proof net is $[\![\delta]\!]$ modulo a permutation of edge-absorbers; then it is equivalent to either $[\![\delta]\!]$ or $[\![\delta]\!]'$ by Lemma **??**. □

4.2 Soundness

Lemma 28. *In a proof net for $[\![G]\!]$, an edge-gadget $[\![e]\!]$ belongs to the empire of at most one vertex-gadget $[\![v]\!]$.*

Proof. Since vertex-gadgets are joined by a tensor, the lemma is immediate from (**?**, Proposition 1). □

Lemma 29. *In a proof net for $[\![G]\!]$, for each vertex v, the weights of the appropriate edge-gadgets in the empire of $[\![v]\!]$ are equal to or greater than the constraint of v.*

Proof. Let $|V| = n$ and $|E| = m$, and consider the vertex v_i and an edge e connecting vertices v_a and v_b where $i \neq a, b$. Each constraint element in $[\![v_i]\!]$ is an instance of the formula

$$C = \bigparr\left(1^{3i+2}\right) \otimes \bigparr\left(1^{3(n-i)+3}\right) .$$

The two $\parr$-subformulae have a balance of $-3i-1$ and $-3(n-i)-2$ respectively. Each weight element in $[\![e]\!]$ is an instance of

$$W = \bigotimes\left(\bot^{3a+2}\right) \parr \bigotimes\left(\bot^{3(b-a)+1}\right) \parr \bigotimes\left(\bot^{3(n-b)+3}\right) .$$

The three $\otimes$-subformulae have a net balance of $3a+1$, $3(b-a)$, and $3(n-b)+2$ respectively. In pairs, they have a net balance of $3b+1$ (1st and 2nd subformula), $3(a+n-b)+3$ (1st and 3rd), and $3(n-a)+2$ (2nd and 3rd). Since $i \neq a, b$, and since no $\otimes$-subformula of W can connect to more than one $\parr$-subformula of C, it follows that W can balance the scope of at most one of both subformulae of C.

The vertex-gadget $[\![v_i]\!]$ is the formula

$$[\![v_i]\!] = \bigparr\left(C^{m\times c(v_i)}\right) \parr 1 .$$

It will be shown that m inappropriate edge-gadgets may balance at most $m-1$ constraint elements C.

Let the root nodes of the two $\parr$-subformulae of each instance of C be labelled x_j and y_j, for $1 \leq j \leq m \times c(v_i)$. If the scopes $s(x_j)$ and $s(y_m)$ of any x_j and y_m are balanced by weight elements W and W' of the same edge-gadget $[\![e]\!]$, then since W and W' are connected by a tensor, there are switchings of W, W', $s(x_j)$, and $s(y_m)$ such that x_j and y_m are connected in the proof net for $[\![G]\!]$. Then the first constraint element of $[\![v_i]\!]$ requires 2 edge-gadgets to balance, and each successive element requires one additional edge-gadget. □

Using the above, a proof net for $\llbracket G \rrbracket$ may be interpreted as a configuration for G.

Definition 30. For a proof net ℓ for the interpretation of a graph $\llbracket G \rrbracket$, let $(\!|\ell|\!)$ be the partial configuration for G where $(\!|\ell|\!)(e)$ is v if 1) e is appropriate for v and 2) $\llbracket e \rrbracket$ belongs to the empire of $\llbracket v \rrbracket$, and undefined otherwise.

Lemma 31. *If $\ell \sim \ell'$ are proof nets for $\llbracket G \rrbracket$ then $(\!|\ell|\!) \sim (\!|\ell'|\!)$.*

Proof. A single permutation $\ell \sim \ell'$ on proof nets may move a number of edge-gadgets $\llbracket e_1 \rrbracket, \dots, \llbracket e_n \rrbracket$ in three ways: 1) out of the empire of a vertex-gadget $\int v$, 2) into the empire of a vertex-gadget $\int w$, or 3) both. Then in both $(\!|\ell|\!)$ and $(\!|\ell'|\!)$ the edges e_1 through e_n are mobile, since by Lemma 29 the empires of the vertex-gadgets $\llbracket v \rrbracket$ (in cases 1 and 3) and $\llbracket w \rrbracket$ (in cases 2 and 3) contain appropriate edge-gadgets other than $\llbracket e_1 \rrbracket$ through $\llbracket e_n \rrbracket$ of sufficient combined weight. It follows that in the graph G the edges e_1 through e_n can be moved away from v and/or onto w one at a time.

□

References

Michael Barr. **-Autonomous categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, Heidelberg, New York, 1979.

Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

Dominic J.D. Hughes. Simple multiplicative proof nets with units. *Annals of Pure and Applied Logic*, 2012. arXiv:math.LO/0507003.