Graphical Proof Theory I: Sequent Calculi operating on Undirected Graphs

Matteo Acclavio

Abstract

In this paper we explore the design of sequent calculi operating on graphs. For this purpose, we introduce a set of logical connectives allowing us to extend the correspondence between cographs and classical propositional formulas to any graph. We then provide sequent calculi operating on these formulas, we prove cut-elimination and that formula encoding the same graph are logically equivalent.

We show that these systems provide conservative extensions of multiplicative linear logic (with and without mix) and classical propositional logic. We conclude by showing that one of these systems is equivalent to the graphical logic GS *defined via a system of context-free graph rewiring rules, therefore providing an alternative proof of analyticity for this logic over graphs.*

Contents

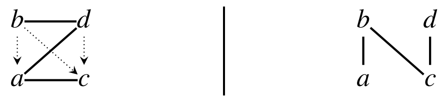


Figure 1: In both graphs, vertices represent processes. In the graph on the left, the direct edges represent the causality relation, while the undirected edges the non-causality between processes. In the graph on the right, the edge relation represent the existence of a condition between processes to access to a same resource.

1 Introduction

In theoretical computer science, formulas are used to describe complex abstract objects by means of elementary operators. In particular, the proof theory of propositional logic typically considers formulas built from a very limited palette of binary and unary operators, respectively called connectives and modalities. In general, such a restriction does not imply a limit in the expressiveness of the language. However, as soon as proof theory is used to define paradigms as "formulas-as-types" or "formulas-as-processes" for concurrent programs this limitation leads to a payout in term of efficiency: as soon as complex interactions need to be represented, ad-hoc encodings are required. This leaves the proof systems to deal with the hard task of handling the syntactic bureaucracy required to handle these encodings. As a consequence, automated tools relying on formula-based proof systems are either sub-optimal, because of the blow-up in computational complexity due to the use of encodings, or sacrifice the quality of information, by reducing their scope on simple configurations [28, 25]. This latter possibility may lead to information loss, potentially causing, among others, security issues or imprecise results in AI for decision systems.

At the same time, graphs are often used in computer science practice from abstract definitions to practical implementation to describe systems with complex interactions: it is often the case that "a picture is worth a thousand words"[1]. By means of example, consider two systems with four processes a, b, c and d represented in Figure 1: in the first, we assume a dependency relation (e.g., causality) in a system with where a depends from b, and c depends from both b and d; in the second, we assume the pairs of processes a and b, b and c, and c and d race to access a shared resource. It is not by accident that both graphs contain four vertices inducing a four-vertices undirected path. In fact, it is well known that graphs containing such a pattern (called P_4) cannot be represented using binary operators on graphs [57, 21]. However, such configurations can be observed in, e.g., *producer-consumer buffers* [26] or control access models with non-transitive conflict relations such as in [45].

The use of graphs as representation of complex interactions is largely used in logic an theoretical computer science because a same abstract object may admit multiple representations, and graphs could be able to provide canonical representative. By means of example, consider semantics of programming languages (see, e.g., label transition systems) or logics (see, e.g., Kripke semantics for modal logics [12]), or proof systems capturing proof equivalence (e.g., proof nets [31] or combinatorial proofs [38, 38]). However, logic, and in particular proof theory, has rarely considered graphs as the main language for expressing its primitive terms: prior to [3, 4, 1] we cannot find proof systems conceived to handle graphs as terms of an inference system defined with proof-theoretical purposes.[2] In these works, the authors move from the well-known correspondence between classical propositional formulas and cographs (graphs containing no induced subgraph isomorphic to a P_4) [41] to generalize proof theoretical methodologies for inference systems on formulas to graphs. In fact, we could say that inference systems operating on formulas can be seen as inference systems operating on cographs, that is, on graphs with "less complex" structure where no induced subgraph isomorphic to P_4 occurs[3]. In these works, the authors consider only *deep inference* [34, 8] formalism to design proof systems operating on graphs. Such unconventional choice with respect to, e.g., sequent calculi or natural deduction, pays off in [1], where a proof system operating on graphs with both symmetric and non-symmetric edges defines a conservative extension of the non-commutative logic BV[4], for which a cut-free sequent calculus cannot exist [55].

[1]To be more precise, we should instead say "a picture is worth an exponential number of words"...

[2]Another line of works [16, 58, 17, 23, 24] explored the extensions of the semantics of boolean logic from cographs enconding formulas to graphs. However, in these works graphical logic is investigated from a semantical viewpoint rather than under the lens of proof systems.

[3]Note that several NP-hard optimization problems on graphs become solvable in polynomial time if restricted to cographs [43].

[4]The logic BV is a NP-time decidable fragment of Pomset logic [53, 52]. This logic is sound and complete with respect to series-

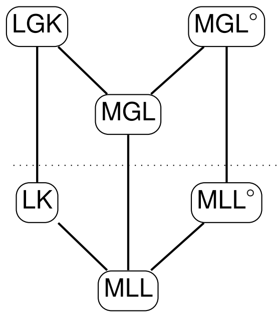
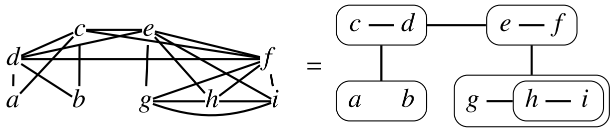


Figure 2: The lattice of the proof systems studied in this paper. The systems below the dotted line contain formulas constructed only using conjunction and disjunction.



$$= \mathsf{P}_4 \left(\!\left| a \parr b, c \otimes d, e \otimes f, g \otimes (h \otimes i) \right|\!\right)$$

Figure 3: A graph, its more compact modular representation, and one of its possible formula-like representation.

1.1 Main contributions

This paper aims at studying sequent calculi operating on graphs, by defining a language whose basic logical operators allow us to have a linear encoding of graphs.

We first recall the notion of *modular decomposition* [29, 41] for undirected graphs[5]. We then use *prime graphs* (graphs admitting only trivial modular decompositions) to define a class of logical operators we call *graphical connectives* allowing us to extending the linear encoding of cographs by classical propositional formulas to any graph.

We then define proof systems operating on these generalized formulas (see Figure 2) and we prove the cut-elimination results and that they are conservative extensions of *multiplicative linear logic*, the *multiplicative linear logic with mix* and the *propositional classical logic*. For this purpose, we use the standard *analyticity* argument, requiring us to reformulate of the standard *subformula property* to accommodate the richer structure of non-binary connectives.

Moreover, we prove that one of these systems proves a class of formulas corresponding to the graphs provable in the system GS from [3, 4], providing a more concise proof of analiticity and transitiveness of implication for the logic GS using more standard techniques[6].

1.2 Outline of the paper

In Section 2 we recall definitions and results in graph theory and the notion of modular decomposition. In Section 3 we use these notions to extend the correspondence between classical propositional formulas and cographs to any graph. We define linear sequent calculi and we prove their properties. In Section 4 we show that one of these calculi is sound and complete with respect to the set of non-empty graphs provable in the deep inference system GS studied in [3, 4]. In Section 5 we define a proof system which is a conservative extension of classical logic. To conclude, we summarize in Section 6 some of the possible the research directions opened by this work.

parallel order refinements: if ϕ and ψ are formulas encoding series-parallel orders, then the order encoded by ϕ is a refinement of the order encoded by ψ iff $\vdash_{\mathsf{BV}} \phi \multimap \psi$.

[5]Note that in this paper we only discuss graphical connectives designed on undirected graphs generalizing the well-known correspondence between classical propositional formulas and cographs. However, the proposed methodology scales to more general graphs such as the mixed graphs used in [1].

[6]The full proof of the admissibility of the rule simulating the cut in deep inference systems in the system GS, as well as the proof that GS is a conservative extension of multiplicative linear logic with mix, are quite convoluted and takes several pages in the Appendix of [4].

2 From Formulas To Graphs

In this section we recall standard results from the literature on graphs such as *modular decomposition* and *cographs*. We then introduce the notion of *graphical connectives* allowing us to extend the correspondence between cographs and classical propositional formulas to general graphs.

2.1 Graphs and Modular Decomposition

In this work are interested in using graphs to represent patterns of interactions by means of the binary relations (edges) between their components (vertices). For this reason we recall the definition of *labeled graph* (the mathematical structure we use to encode these patterns) together with the definition of *isomorphism* (the standard notion of identity on labeled graphs) and the rougher notion of *similarity* (equivalence up-to labels over vertices).

Definition 1. A ***$\mathcal{L}$-labeled graph*** (or simply ***graph***) $G = \langle V_G, \ell_G, \overset{G}{\frown} \rangle$ is given by a finite set of ***vertices*** V_G, a partial ***labeling function*** $\ell_G \colon V_G \to \mathcal{L}$ associating a label $\ell(v)$ from a given set of labels $\mathcal{L}$ to each vertex $v \in V_G$ (we may represent ℓ_G as a set of equations of the form $\ell(v) = \ell_v$ and denote by $\varnothing$ the empty function), and a non-reflexive symmetric edge relation $\overset{G}{\frown} \subset V_G \times V_G$ whose elements, called ***edges***, may be denoted vw instead of (v, w). The ***empty*** graph $\langle \varnothing, \varnothing, \varnothing \rangle$ is denoted $\varnothing$.

A ***similarity*** between two graphs G and G' is a bijection $f \colon V_G \to V_{G'}$ such that $x \overset{G}{\frown} y$ iff $f(x) \overset{G'}{\frown} f(y)$ for any $x, y \in V_G$. An ***isomorphism*** is a similarity f such that $\ell(v) = \ell(f(v))$ for any $x, y \in V_G$. Two graphs G and G' are ***similar*** (denoted $G \sim G'$) if there is an similarity between G and G'. A ***symmetry*** is a similarity of a graph with itself. They are ***isomorphic*** (denoted $G = G'$) if there is a isomorphism between G and G'. From now on, we consider two isomorphic graphs to be ***the same*** graph.

Two vertices v and w in G are ***connected*** if there is a sequence $v = u_0, \ldots, u_n = w$ of vertices in G (called ***path***) such that $u_{i-1} \overset{G}{\frown} u_i$ for all $i \in \{1, \ldots, n\}$. A ***connected component*** of G is a maximal set of connected vertices in G. A graph G is a ***clique*** (resp. a ***stable set***) iff $\overset{G}{\not\frown} = \varnothing$ (resp. $\overset{G}{\frown} = \varnothing$).

Notation 2. When drawing a graph or an unlabeled graph we draw v—w whenever $v \frown w$, we draw no edge at all whenever $v \not\frown w$. We may represent a vertex of a graph by using its label instead of its name. For example, the single-vertex graph $G = \langle \{v\}, \ell_G, \varnothing \rangle$ may be represented either by a the vertex name v or by the vertex label $\ell(v)$ (or $\bullet$ if $\ell(v)$ is not defined). Note that, since we are considering isomorphic graphs to be the same, as soon as there is no ambiguity due to vertices represented by the same symbol, we can assume that the representation of a graph to provide us one of the possible triple (set of vertices, label function, and set of edges) defining it.

Example 3. Consider the following graphs:

$$
\begin{aligned}
F &= \langle \{u_1, u_2, u_3, u_4\}, \{\ell(u_1) = a, \ell(u_2) = b, \ell(u_3) = c, \ell(u_4) = d\}, \{u_1u_2, u_2u_3, u_3u_4\} \rangle \\
G &= \langle \{v_1, v_2, v_3, v_4\}, \{\ell(v_1) = b, \ell(v_2) = a, \ell(v_3) = c, \ell(v_4) = d\}, \{v_1v_2, v_1v_3, v_3v_4\} \rangle \\
H &= \langle \{w_1, w_2, w_3, w_4\}, \{\ell(w_1) = a, \ell(w_2) = b, \ell(w_3) = c, \ell(w_4) = d\}, \{w_1w_2, w_1w_3, w_3w_4\} \rangle
\end{aligned}
$$

They are all symmetric, that is $F \sim G \sim H$, but $F = G \neq H$ as can easily be verified using their representations:

$$F = a\text{—}b\text{—}c\text{—}d = G \qquad \text{and} \qquad H = b\text{—}a\text{—}c\text{—}d$$

Observation. The problem of graph isomorphism is a standard **NP**-problem (to be more precise, its complexity is quasi-polynomial [11]). That is, verify that a given bijection between the sets of vertices of two graphs is an isomorphism can be checked in polynomial time, while there is no known polynomial time algorithm to find such an isomorphism. For this reason, whenever we say that two graphs are the same, either we assume they share the same set of vertices, therefore implicitly assuming the isomorphism f to be defined by the identity function over the set of vertices, or we assume an isomorphism to be given. This allows us to verify whether two graphs are the same in polynomial time.

In order to use proof theoretical methodologies on graphs, we need a suitable notion of subgraphs to be used in the same way sub-formulas are used in proof systems, that is, to state properties of the calculus or to define the behavior of rules. For this purpose, we use for a notion of *module* to identify subgraph allowing us to decompose a graph using abstract syntax trees similar to the ones underlying formulas [29, 41, 37, 46, 49, 27]. Intuitively, a module is a subset of vertices of a graph having the same edge-relation with any vertex outside the subset. This generalize what we observe in formulas, where any propositional atom of a subformula has the same relation (the one given by the least common ancestor node in the formula tree) with a given propositional atom not in the subformula with a propositional atom .

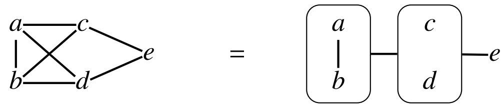
Definition 4. Let $G = \langle V_G, \ell_G, E_G \rangle$ be a graph and $W \subseteq V_G$. The ***graph induced*** by W is the graph $G|_W := \left\langle W, \ell_G|_W, \overset{G}{\frown} \cap (W \times W) \right\rangle$ where $\ell_G|_W(v) := \ell_G(v)$ for all $v \in W$.

A ***module*** of a graph G is a subset M of V_G such that $x \frown z$ iff $y \frown z$ for any $x, y \in M$, $z \in V_G \setminus M$. A module M is ***trivial*** if $M = \emptyset$, $M = V_G$, or $M = \{x\}$ for some $x \in V_G$. From now on, we identify a module M of a graph G with the induced subgraph $G|_M$.

Remark 5. A connected component of a graph G is a module of G.

Using modules we can optimize the way we represent graphs reducing the number of edges drawn without losing information, relying on the fact that all vertices of a module has the same edge-relation with any vertex outside the module.

Notation 6. In representing graphs we may border vertices of a same module by a closed line. An edges connected to such a closed line denotes the existence of an edge to each vertex inside it. By means of example, consider the following graph and its more compact modular representation.



(1)

The notion of module is related to a notion of context, which can be intuitively formulated as a graph with a special vertex playing the role of a hole in which we can plug in a module.

Definition 7. A ***context*** $C[\square]$ is a (non-empty) graph containing a single occurrence of a special vertex $\square$ (such that $\ell(\square)$ is undefined). It is ***trivial*** if $C[\square] = \square$. If $C[\square]$ is a context and G a graph, we define $C[G]$ as the graph obtained by replacing $\square$ by G. Formally,

$$C[G] := \left\langle (V_{C[\square]} \setminus \{\square\}) \uplus V_G \ ,\ \ell_C \cup \ell_G \ ,\ \left\{ vw \mid v, w \in V_{C[\square]} \setminus \{\square\}, v \overset{C[\square]}{\frown} w \right\} \cup \left\{ vw \mid v \in V_{C[\square]} \setminus \{\square\}, w \in V_G, v \overset{C[\square]}{\frown} \square \right\} \right\rangle$$

Remark 8. A set of vertices M is a module of a graph G iff there is a context $C[\square]$ such that $G = C[M]$.

We generalize this idea of replacing a vertex of a graph with a module by defining the operations of *composition-via* a graph, where all vertices of a graph are replaced in a "modular way" by modules.

Definition 9. Let G be a graph with $V_G = \{v_1, \ldots, v_n\}$ and let $H_1, \ldots, H_n$ be graphs. We define the ***composition of $H_1, \ldots, H_n$ via G*** as the graph $G(\!| H_1, \ldots, H_n |\!)$ obtained by replacing each vertex v_i of G with a module H_i for all $i \in \{1, \ldots, n\}$. Formally,

$$G(\!| H_1, \ldots, H_n |\!) \ = \ \left\langle \ \biguplus_{i=1}^{n} V_{H_i} \ ,\ \bigcup_{i=1}^{n} \ell_{H_i} \ ,\ \left(\bigcup_{i=1}^{n} \overset{H_i}{\frown} \right) \cup \left\{ (x, y) \ \middle| \ x \in V_{H_i}, y \in V_{H_j}, v_i \overset{G}{\frown} v_j \right\} \ \right\rangle \tag{2}$$

The subgraphs $H_1, \ldots, H_n$ are called ***factors*** of $G(\!| H_1, \ldots, H_n |\!)$ and, by definition, are (possibly not maximal) modules of $G(\!| H_1, \ldots, H_n |\!)$.

Remark 10. The information about the labels of the graph G used to define the composition-via operation is lost. Moreover, if G is a graph with $V_G = \{v_1, \ldots, v_n\}$ and σ a permutation over the set $\{1, \ldots, n\}$ such that the map $f_\sigma : V_G \to V_G$ mapping v_i in $f_\sigma(v_i) = v_{\sigma(i)}$ for all $i \in \{1, \ldots, n\}$ is an similarity between G and G, then $G(\!| H_1, \ldots, H_n |\!) = G'(\!| H_1, \ldots, H_n |\!)$.

In order to establish a connection between graphs and formulas, from now on we only consider graphs whose set of labels belong to the set $\mathcal{L} = \{a, a^\perp \mid a \in \mathcal{A}\}$ where $\mathcal{A}$ is a fixed set of propositional variables. We then define the *dual* of a graphs.

Definition 11. Let $G = \langle V_G, \ell_G, E_G\rangle$ be a graph. We define the edge relation $\overset{G}{\not\frown} := \left\{(v,w) \mid v \neq w \text{ and } vw \notin \overset{G}{\frown}\right\}$ and we define the ***dual*** graph of G as the graph $G^\perp := \left\langle V_G, \overset{G}{\not\frown}, \ell_{G^\perp}\right\rangle$ with $\ell_{G^\perp}(v) = (\ell_G(v))^\perp$ (assuming $a^{\perp\perp} = a$ for all $a \in \mathcal{A}$).

Remark 12. By definition, each module of a graph corresponds to a module of its dual graph. It follows that a connected component of $G^\perp$ is a module of G.

Notation 13. If $\mathcal{G}$ is the representation of a graph G, then we may represent the graph $G^\perp$ by bordering the representation of G with a closed line with the negation symbol on the upper-right corner, that is, $\left(\mathcal{G}\right)^\perp$.

2.2 Classical Propositional Formulas and Cographs

The set of ***classical (propositional) formulas*** is generated from a set of propositional variable $\mathcal{A}$ using the ***negation*** $(\cdot)^\perp$, the ***disjunction*** $\vee$ and the ***conjunction*** $\wedge$ using the following grammar:

$$\phi, \psi := a \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi^\perp \qquad \text{with } a \in \mathcal{A}. \tag{3}$$

We denote by $\equiv$ the equivalence relation over formulas generated by the following laws:

$$\begin{array}{rll} \textbf{\textit{Equivalence laws}} & \left\{ \begin{array}{l} \phi \vee \psi \equiv \psi \vee \phi \\ \phi \wedge \psi \equiv \psi \wedge \phi \end{array} \right. & \begin{array}{l} \phi \vee (\psi \vee \chi) \equiv (\phi \vee \psi) \vee \chi \\ \phi \wedge (\psi \wedge \chi) \equiv (\phi \wedge \psi) \wedge \chi \end{array} \\ \textbf{\textit{De-Morgan laws}} & \left\{ (\phi^\perp)^\perp \equiv \phi \right. & (\phi \wedge \psi)^\perp \equiv \phi^\perp \vee \psi^\perp \end{array} \tag{4}$$

We define a map from literals to single-vertex graphs, which extends to formulas via the composition-via the unlabeled two-vertices stable set S_2 and two-vertices clique K_2.

Definition 14. Let ϕ be a classical formula, then $[\![\phi]\!]$ is the graph inductively defined as follows:

$$[\![a]\!] = a \qquad \left[\!\left[\phi^\perp\right]\!\right] = [\![\phi]\!]^\perp \qquad [\![\phi \vee \psi]\!] = \mathsf{S}_2 \left(\!\left|[\![\phi]\!], [\![\psi]\!]\right|\!\right) \qquad [\![\phi \wedge \psi]\!] = \mathsf{K}_2 \left(\!\left|[\![\phi]\!], [\![\psi]\!]\right|\!\right)$$

where S_2 and K_2 are respectively a stable set and a clique with 2 vertices, and where we denote by a the single-vertex graph, whose vertex is labeled by a.

We can easily observe that the map $[\![\cdot]\!]$ well-behaves with respect to the equivalence over formulas $\equiv$, that is, equivalent formulas are mapped to the symmetric graphs.

Proposition 15. *Let ϕ and ψ be classical formulas. Then $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$.*

We finally recall the definition of *cographs*, and the theorem establishing the relation between cographs and classical formulas, i.e., providing an alternative definition of cographs as graphs generated by single-vertex graphs using the composition-via a two-vertices no-edge graph and a two-vertices one-edge graph.

Definition 16. A ***cograph*** is a graph G such that there are no four vertices v_1, v_2, v_3, v_4 in G such that the induced subgraph $G|_{\{v_1,v_2,v_3,v_4\}}$ is similar to the graph $\langle\{a,b,c,d\}, \emptyset, \{ab, bc, cd\}\rangle = a — b — c — d$.

Theorem 17 ([29])**.** *A graph G is a cograph iff there is a formula ϕ such that $G \sim [\![\phi]\!]$.*

2.3 Modular Decomposition of Graphs

We recall the notion of *prime graph*, allowing us to provide canonical representatives of graphs via modular decomposition. (see e.g., [29, 41, 37, 46, 49, 27]).

Definition 18. A graph G is ***prime*** if $|V_G| > 1$ and all its modules are trivial.

We recall the following standard result from the literature.

Theorem 19 ([41])**.** *Let G be a graph with at least two vertices. Then there are non-empty modules $M_1, \dots, M_n$ of G and a prime graph P such that $G = P(\!|M_1, \dots, M_n|\!)$.*

This result implies the possibility of describing graphs using single-vertex graphs and the operation of composition-via prime graphs. More precisely, we can define the notion of *modular decomposition* of a graph composition-via prime graphs to provide a more canonical representation.

Definition 20. Let G be a non-empty graph. A ***modular decomposition*** of G is a way to write G using single-vertex graphs and the operation of composition-via prime graphs:

- if G is a graph with a single vertex x labeled by a, then $G = a$ (i.e., $G = \langle \{x\}, \ell(x) = a, \emptyset \rangle$);
- if $H_1, \ldots, H_n$ are maximal modules of G such that $V_G = \biguplus_{i=1}^{n} V_{H_i}$, then there is a unique prime graph P such that $G = P(\!| H_1, \ldots, H_n |\!)$.

Remark 21. There are various reasons why modular decomposition is not unique.

The first is due to the possible presence of cliques and stable sets. By means of example, consider a clique with three vertices u, v and w can be represented as $(u \otimes v) \otimes w$ or $u \otimes (v \otimes w)$.

We already observed the second reason in Remark 10, since graph symmetries allow us to represent the same graph by different decompositions, as shown in top-most modular decomposition below on the left.

$$\begin{array}{l} P(\!| u, v, w, t |\!) = u\text{—}v\text{—}w\text{—}t = P(\!| t, w, v, u |\!) \\ P(\!| u, v, w, t |\!) = u\text{—}v\text{—}w\text{—}t = P'(\!| u, w, v, t |\!) \end{array} \quad \text{where} \quad P = a\text{—}b\text{—}c\text{—}d \quad \text{and} \quad P' = a\text{—}c\text{—}b\text{—}d \,.$$

Finally, two symmetric prime graphs could provide distinct modular decompositions of the same graph, as shown above with symmetric prime graphs P and P'.

The first problem could be addressed by considering in the modular decomposition not only prime graphs, but also cliques and stable sets, that is, including n-ary versions of the operations $\parr$ and $\otimes$. We show later in this paper that this problem is irrelevant due to the associativity of $\parr$ and $\otimes$. The second problem cannot be addressed without enforcing a cumbersome order over graphs taking into account vertex labels and factor positions. However, we can address the latter source of ambiguity by introducing the notion of *base* of *graphical connectives*, allowing us to provide a single canonical prime graph for each class of symmetric prime graphs.

Definition 22. A ***graphical connective*** $C = \langle V_C, \overset{C}{\frown} \rangle$ (with ***arity*** $n = |V_C|$) is given by a finite list of vertices $V_C = \langle v_1, \ldots, v_n \rangle$ and a non-reflexive symmetric edge relation $\overset{C}{\frown}$ over the set of vertices occurring in V_C. We denote by G_C the graph corresponding to C, that is, the graph $G_C = \langle \{v \mid v \text{ in } V_C\}, \emptyset, \overset{C}{\frown} \rangle$. The ***composition-via*** a graphical connective is defined as the composition-via the graph G_C.

A graphical connective is ***prime*** if G_C is a prime graph. A set $\mathcal{P}$ of prime graphical connectives is a ***base*** if for each prime graph P there is a unique connective $C \in \mathcal{P}$ such that $P \sim G_C$.

Given an n-ary connective C, we define the following sets of permutations over the set $\{1, \ldots, n\}$:

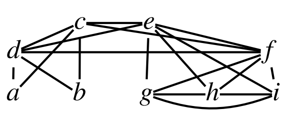
$$\begin{array}{lcl} \text{the } \textit{\textbf{group}}^{7} \textit{\textbf{ of symmetries}} \text{ of } C & : \mathfrak{S}(C) & := \{ \sigma \mid C(\!| a_1, \ldots, a_n |\!) = C(\!| a_{\sigma(1)}, \ldots, a_{\sigma(n)} |\!) \} \\ \text{the } \textit{\textbf{set of dualizing symmetries}} \text{ of } C & : \mathfrak{S}^{\perp}(C) & := \left\{ \sigma \mid (C(\!| a_1, \ldots, a_n |\!))^{\perp} = C(\!| a^{\perp}_{\sigma(1)}, \ldots, a^{\perp}_{\sigma(n)} |\!) \right\} \end{array} \tag{5}$$

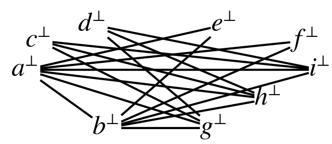
for some single-vertex graphs $a_1, \ldots, a_n$.

Notation 23. We define the following graphical connectives (with $n > 1$):

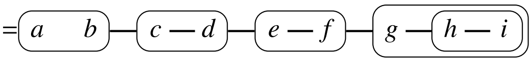
$$\begin{array}{rcl} \parr(\!| v_1, v_2 |\!) := \langle \langle v_1, v_2 \rangle, \emptyset \rangle & = & \boxed{v_1 \quad v_2} \\ \otimes(\!| v_1, v_2 |\!) := \langle \langle v_1, v_2 \rangle, \{v_1 v_2\} \rangle & = & \boxed{v_1 \text{—} v_2} \\ \mathsf{P}_{\mathsf{n}}(\!| v_1, \ldots, v_n |\!) := \langle \langle v_1, \ldots, v_n \rangle, \{v_i v_{i+1} \mid i \in \{1, \ldots, n-1\}\} \rangle & = & \boxed{v_1 \text{—} v_2 \text{—} \cdots \text{—} v_{n-1} \text{—} v_n} \\ \mathsf{Bull}(\!| v_1, \ldots, v_5 |\!) := \langle \langle v_1, \ldots, v_5 \rangle, \{(v_1 v_2, v_2 v_3, v_3 v_4, v_5 v_2, v_5 v_3)\} \rangle & = & \boxed{v_1 \text{—} v_2 \text{—} v_3 \text{—} v_4 \;\; (v_2 \text{—} v_5 \text{—} v_3)} \end{array} \tag{6}$$

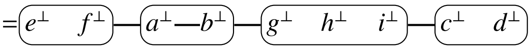
Example 24. Consider the following graph G and its dual $G^{\perp}$:

$G =$ 

and $G^{\perp} =$ 

We can write them as

$$G = \mathsf{P}_4 \llparenthesis a \parr b, c \otimes d, e \otimes f, g \otimes (h \otimes i) \rrparenthesis =$$ 

$$G^\perp = \mathsf{P}_4^\perp \llparenthesis a^\perp \otimes b^\perp, c^\perp \parr d^\perp, e^\perp \parr f^\perp, g^\perp \parr (h^\perp \parr i^\perp) \rrparenthesis =$$
$$= \mathsf{P}_4 \llparenthesis c^\perp \parr d^\perp, a^\perp \otimes b^\perp, g^\perp \parr (h^\perp \parr i^\perp), e^\perp \parr f^\perp \rrparenthesis =$$ 

We can reformulate the standard result on modular decomposition as follows.

Theorem 25. *Let G be a non-empty graph and $\mathcal{P}$ a base. Then then there is a unique way (up to symmetries of graphical connectives and associativity of $\parr$ and $\otimes$) to write G using single-vertex graphs and the graphical connectives in $\mathcal{P}$.*

Corollary 26. *Two graphs are isomorphic iff they admit a same modular decomposition.*

2.4 Graphs as Formulas

In order to represent graphs as formulas, we define new connectives beyond conjunction and disjunction to represent graphical connectives in a base $\mathcal{P}$. From now on, we assume bases $\mathcal{P}$ containing the graphical connectives in Equation (6) to be fixed.

Definition 27. The set of ***formulas*** is generated by the set of propositional atoms $\mathcal{A}$, a ***unit*** $\circ$, using the following syntax:

$$\phi_1, \ldots, \phi_n \coloneqq \circ \mid a \mid a^\perp \mid \kappa_P \llparenthesis \phi_1, \ldots, \phi_{n_P} \rrparenthesis \qquad \text{with } a \in \mathcal{A} \text{ and } P \in \mathcal{P} \tag{7}$$

We simply denote $\parr$ (resp. $\otimes$) the binary connective $\kappa_\parr$ (resp. $\kappa_\otimes$) and we write $\phi \parr \psi$ instead of $\kappa_\parr \llparenthesis \phi, \psi \rrparenthesis$ (resp. $\phi \otimes \psi$ instead of $\kappa_\otimes \llparenthesis \phi, \psi \rrparenthesis$). The ***arity*** of the connective κ_P is the arity n_P of P.

A ***literal*** is a formula of the form a or $a^\perp$ for an atom $a \in \mathcal{A}$. The set of literals is denoted $\mathcal{L}$. A ***κ-formula*** is a formula with ***main connective*** κ, that is, a formula of the form $\kappa \llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis$. A formula is ***unit-free*** if it contains no occurrences of $\circ$ and ***vacuous*** if it contains no atoms. A formula is ***pure*** if non-vacuous and such that its vacuous subformulas are $\circ$. A **MLL-*formula*** is a formula containing only occurrences of $\parr$ and $\otimes$ connectives.

A ***context formula*** (or simply ***context***) $\zeta[\square]$ is a formula containing an ***hole*** $\square$ taking the place of an atom. Given a context $\zeta[\square]$, the formula $\zeta[\phi]$ is defined by simply replacing the atom $\square$ with the formula ϕ. For example, if $\zeta[\square] = \psi \parr (\square \otimes \chi)$, then $\zeta[\phi] = \psi \parr (\phi \otimes \chi)$.

For each ϕ formula (or context), the graph $[\![\phi]\!]$ is defined as follows:

$$[\![\square]\!] = \square \qquad [\![\circ]\!] = \varnothing \qquad [\![a]\!] = a \qquad \left[\!\!\left[\phi^\perp\right]\!\!\right] = [\![\phi]\!]^\perp \qquad [\![\kappa_P \llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis]\!] = P \left\llparenthesis [\![\phi_1]\!], \ldots, [\![\phi_n]\!] \right\rrparenthesis \tag{8}$$

Notation 28. We could consider a formula ϕ over the set of occurrences of literals $\{x_1, \ldots, x_n\}$ as a ***synthetic connective***. That is, we may denote by $\phi \llparenthesis \psi_1, \ldots, \psi_n \rrparenthesis$ the formula obtained by replacing each literal x_i with a corresponding ψ_i for all $i \in \{1, \ldots, n\}$. The set of ***symmetries*** of ϕ (denoted $\mathfrak{S}(\phi)$) is the set of permutations σ over $\{1, \ldots, n\}$ such that $[\![\phi \llparenthesis x_1, \ldots, x_n \rrparenthesis]\!] = [\![\phi \llparenthesis x_{\sigma(1)}, \ldots, x_{\sigma(n)} \rrparenthesis]\!]$.

Definition 29. The equivalence relation $\equiv$ over formulas is generated by the following equations:

$$\textit{Equivalence laws} \begin{cases} \kappa_P \llparenthesis \phi_1, \ldots, \phi_{|P|} \rrparenthesis \equiv \kappa_P \llparenthesis \phi_{\sigma(1)}, \ldots, \phi_{\sigma(|V_P|)} \rrparenthesis & \text{for each } \sigma \in \mathfrak{S}(P) \\ \phi \otimes (\psi \otimes \chi) \equiv (\phi \otimes \psi) \otimes \chi \\ \phi \parr (\psi \parr \chi) \equiv (\phi \parr \psi) \parr \chi \end{cases}$$

$$\textit{De-Morgan laws} \begin{cases} \circ^\perp \equiv \circ \qquad \phi^{\perp\perp} \equiv \phi \\ \text{only if } \mathfrak{S}^\perp(P) = \varnothing: \quad (\kappa_P \llparenthesis \phi_1, \ldots \phi_{n_P} \rrparenthesis)^\perp \equiv \kappa_{P^\perp} \llparenthesis \phi^\perp_{\sigma(1)}, \ldots, \phi^\perp_{\sigma(n_P)} \rrparenthesis \\ \text{only if } \mathfrak{S}^\perp(Q) \neq \varnothing: \quad (\kappa_P \llparenthesis \phi_1, \ldots \phi_{n_P} \rrparenthesis)^\perp \equiv \kappa_P \llparenthesis \phi^\perp_{\rho(1)}, \ldots, \phi^\perp_{\rho(n_P)} \rrparenthesis & \text{for each } \rho \in \mathfrak{S}^\perp(P) \end{cases}$$

for each $P \in \mathcal{P}$ (with arity n_P).

The ***(linear) negation*** over formulas is defined by letting

$$\circ^\perp = \circ \qquad \phi^{\perp\perp} = \phi \qquad (\kappa_P \llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis)^\perp = \kappa_Q \llparenthesis \phi^\perp_{\sigma_P(1)}, \ldots, \phi^\perp_{\sigma_P(n)} \rrparenthesis$$

where Q is the unique graphical connective in $\mathfrak{P}$ such that $\left[\!\left[\kappa_P \left(\!\left| a_1, \ldots, a_n \right|\!\right)\right]\!\right] = Q \left(\!\left| a^{\perp}_{\sigma(1)}, \ldots, a^{\perp}_{\sigma_n} \right|\!\right)$ for any single-vertex graphs $a_1^{\perp}, \ldots, a_n^{\perp}$ (with vertex labeled by $a_1^{\perp}, \ldots, a_n^{\perp}$ respectively) and a permutationσ_P over the set $\{1, \ldots, n\}$. [8]

The ***linear implication*** $\phi \multimap \psi$ is defined as $\phi^{\perp} \parr \psi$, while the ***logical equivalence*** $\phi \multimapboth \psi$ is defined as $(\phi \multimap \psi) \otimes (\psi \multimap \phi)$.

Remark 30. As explained in [4] (Section 9), the graphical connectives we discuss in this paper are *multiplicative connectives* (in the sense of [22, 33]) but they are not the same as the *connectives-as-partitions* discussed in these works. In fact, there is a unique 4-ary graphical connectives P_4 with symmetry group $\{\mathsf{id}, (1,4)(2,3)\}$, while, as shown in [48, 5], there is a unique pair of dual 4-ary multiplicative connectives-as-partitions G_4 and $\mathsf{G}_4^{\perp}$ with a strictly larger symmetry group.

The following result is consequence of Theorem 19.

Proposition 31. *Let ϕ and ψ be formulas. If $\phi \equiv \psi$, then $[\![\phi]\!] = [\![\psi]\!]$. Moreover, if ϕ and ψ are unit-free, then $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$.*

Note that the the stronger result does not hold in presence of units. For an example consider any two distinct vacuous formulas.

3 Sequent calculi over operating on graphs-as-formulas

We assume the reader to be familiar with the definition of sequent calculus derivations as trees of sequents (see, e.g., [56]) but we recall here some definitions.

Definition 32. We define a ***sequent*** is a set of occurrences of formulas. A ***sequent system*** S is a set of ***sequent rules*** as the ones in Figure 4. In a sequent rule ρ, we say that a formula is ***active*** if it occurs in one of its premises (the sequents above the horizontal line) but not in its conclusion (the sequent below the horizontal line), and ***principal*** if it occurs in its conclusion but in none of its premises.

A ***proof*** of a sequent Γ is a derivation with no open premises, denoted $\overset{\pi}{\underset{\Gamma}{\|}}\mathsf{S}$. We denote by $\begin{array}{c}\Gamma' \\ \pi' \| \mathsf{S} \\ \Gamma\end{array}$ an ***(open) derivation*** of Γ from Γ', that is, is a proof tree having exactly one open premise Γ'.

A rule is ***admissible*** in S if there is a derivation of the conclusion of the rule whenever all premises of the rule are derivable. A rule is ***derivable*** in S, if there is a derivation in S from the premises to the conclusion of the rule.

Notation 33. In this paper we use the same notation to denote a sequent system S and the set of formulas admitting a proof in S.

Definition 34. We define the following sequent systems using the rules in Figure 4.

$$\begin{array}{ll} \textbf{\textit{Multiplicative Graphical Logic}}: & \mathsf{MGL} \;= \{\mathsf{ax}, \parr, \otimes, \mathsf{d}\text{-}P \mid P \in \mathfrak{P}\} \\ \textbf{\textit{Multiplicative Graphical Logic with mix}}: & \mathsf{MGL}^{\circ} = \mathsf{MGL} \cup \{\mathsf{mix}, \mathsf{wd}_{\otimes}, \mathsf{unitor}_{\kappa}\} \end{array} \tag{9}$$

Observation (Rules Exegesis)**.** The rules ***axiom*** (ax), ***par*** ($\parr$), ***tensor*** ($\otimes$), ***cut*** (cut), and ***mix*** (mix) are the standard as in multiplicative linear logic with mix. Note that ax is restricted to atomic formulas.

The ***dual connectives*** rule (d-κ) handles a pair of dual connectives at the same time.[9] To get an intuition of this rule, consider the right-conjunction rule ($\wedge_R$) used in two-sided sequent calculi for classical logic shown below on the left. The interpretation of this rule is that if the left premise *and* the right premise are derivable, then the conclusion is. Note that, even if the rule does not introduce a conjunction on the lefthand-side of the $\vdash$, the interpretation of the conclusion sequent is the same of the interpretation of the

[8] Note that the permutation σ_P may be not unique. This is not a problem if we consider formulas up-to the equivalence relation $\equiv$. Otherwise, in order to properly define the linear negation, we should fix a permutation σ_P for each graphical connective $P \in \mathfrak{P}$ in such a way either σ_P is an involution (in case $G_P \sim (G_P)^{\perp}$), or $\sigma_P \sigma_Q$ is the identity (in case $G_P \nsim (G_P)^{\perp} \sim G_Q$ for a $Q \in \mathfrak{P} \setminus \{P\}$).

[9] Rules handling multiple operators at the same time are not a novelty in structural proof theory: in focused proof systems (see, e.g.[9, 51, 50]) rules can handle multiple connectives of a same formula, while in modal logic and linear logic (see, e.g., [32, 12, 14, 44]) is quite standard to have rules handling modalities occurring in different formulas of a same sequent.

$$\text{ax}\,\frac{}{\vdash a, a^{\perp}} \qquad \text{⅋}\,\frac{\vdash \Gamma, \phi, \psi}{\vdash \Gamma, \phi ⅋ \psi} \qquad \otimes\,\frac{\vdash \Gamma, \phi \quad \vdash \psi, \Delta}{\vdash \Gamma, \phi \otimes \psi, \Delta}$$

$$\text{d-}\kappa\,\frac{\vdash \Gamma_1, \phi_{\sigma(1)}, \psi_{\tau(1)} \qquad \cdots \qquad \vdash \Gamma_n, \phi_{\sigma(n)}, \psi_{\tau(n)}}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa(\!|\phi_1, \ldots, \phi_n|\!), \kappa^{\perp}(\!|\psi_1, \ldots \psi_n|\!)} \begin{cases}\sigma \in \mathfrak{S}(\kappa) \\ \tau \in \mathfrak{S}(\kappa^{\perp})\end{cases}$$

$$\text{mix}\,\frac{\vdash \Gamma_1 \quad \vdash \Gamma_2}{\vdash \Gamma_1, \Gamma_2} \qquad \text{wd}_{\otimes}\,\frac{\vdash \Gamma, \phi_k \quad \vdash \Delta, \kappa(\!|\phi_1, \ldots, \phi_{k-1}, \circ, \phi_{k+1}, \ldots, \phi_n|\!)}{\vdash \Gamma, \Delta, \kappa(\!|\phi_1, \ldots, \phi_n|\!)}$$

$$\text{unitor}_{\kappa}\,\frac{\vdash \Gamma, \chi(\!|\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)}|\!)}{\vdash \Gamma, \kappa(\!|\phi_1, \ldots, \phi_k, \circ, \phi_{k+1}, \ldots, \phi_n|\!)} \dagger$$

$$\dagger := \sigma \in \mathfrak{S}(\chi) \quad \text{and} \quad [\![\kappa(\!|\phi_1, \ldots, \phi_k, \circ, \phi_{k+1}, \ldots, \phi_n|\!)]\!] = [\![\chi(\!|\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)}|\!)]\!] \neq \emptyset$$

Figure 4: Linear sequent calculus rules for MGL and MGL$^{\circ}$.

$$\text{AX}\,\frac{}{\vdash \phi, \phi^{\perp}}\ \phi\text{ pure} \qquad \text{cut}\,\frac{\vdash \Gamma_1, \phi \quad \vdash \Gamma_2, \phi^{\perp}}{\vdash \Gamma_1, \Gamma_2} \qquad \text{wd}_{⅋}\,\frac{\vdash \Gamma, \kappa(\!|\phi, \psi_1, \ldots, \psi_n|\!)}{\vdash \Gamma, \kappa(\!|\circ, \psi_1, \ldots, \psi_n|\!), \phi}$$

$$\text{deep}\,\frac{\vdash \Gamma, \phi \quad \vdash \Delta, \psi}{\vdash \Gamma, \Delta, \zeta[\phi]}\ [\![\zeta[\circ]]\!] = [\![\psi]\!] \qquad \text{d-}\chi\,\frac{\vdash \Gamma_1, \phi_{\sigma(1)}, \psi_{\tau(1)} \qquad \cdots \qquad \vdash \Gamma_n, \phi_{\sigma(n)}, \psi_{\tau(n)}}{\vdash \Gamma_1, \ldots, \Gamma_n, \chi(\!|\phi_1, \ldots, \phi_n|\!), \chi^{\perp}(\!|\psi_1, \ldots \psi_n|\!)} \begin{cases}\sigma \in \mathfrak{S}(\chi) \\ \tau \in \mathfrak{S}(\chi^{\perp})\end{cases}$$

Figure 5: Admissible rules in MGL$^{\circ}$.

sequent in which ϕ_1 and ϕ_2 are in conjunction because the standard interpretation of a two-sides sequent $\Gamma \vdash \Delta$ is defined as $\left(\bigwedge_{\phi \in \Gamma} \phi^{\perp}\right) \vee \left(\bigvee_{\psi \in \Delta} \psi\right)$.

$$\wedge_R\,\frac{\boxed{\Gamma_1, \phi_1 \vdash \psi_1, \Delta_1} \quad \text{“and”} \quad \boxed{\Gamma_2, \phi_2 \vdash \psi_2, \Delta_2}}{\Gamma_1, \Gamma_2, \phi_1, \phi_2 \vdash \psi_1 \wedge \psi_2, \Delta_1, \Delta_2} \;\Bigg|\; \frac{\mathsf{P}_4\left(\!\left|\boxed{\Gamma_1, \phi_1 \vdash \psi_1, \Delta_1}, \boxed{\Gamma_2, \phi_2 \vdash \psi_2, \Delta_2}, \boxed{\Gamma_3, \phi_3 \vdash \psi_3, \Delta_3}, \boxed{\Gamma_4, \phi_4 \vdash \psi_4, \Delta_4}\right|\!\right)}{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \kappa_{\mathsf{P}_4}(\!|\phi_1, \phi_2, \phi_3, \phi_4|\!) \vdash \kappa_{\mathsf{P}_4}(\!|\psi_1, \psi_2, \psi_3, \psi_4|\!), \Delta_1, \Delta_2, \Delta_3, \Delta_4}$$

In a two-sided setting the rule d-κ could have been reformulated by introducing the same connective in both sides. Intuitively, such a rule would internalize in the logic a “meta” relation between the premises of the rule, as intuitively shown above on the right for the connective P_4.

The names of the rules ***unitor*** (unitor$_{\kappa}$) and ***weak-distributivity*** (wd$_{\otimes}$) are inspired by the literature of *monoidal categories* [47] and *weakly distributive categories* [54, 20, 19]. The rule unitor$_{\kappa}$ internalize the fact that the unit $\circ$ is the neutral element for all connectives (its side condition prevents the creation of non-pure formulas). Under the assumption of the existence of a $\circ$ which is the unit of both $\otimes$ and ⅋, the rule wd$_{\otimes}$ generalizes the *weak-distribution law* (shown below on the left) of the $\otimes$ over the ⅋ to the weak-distributivity of $\otimes$ over any connective (see below on the top-right)

$$\phi \otimes (\psi ⅋ \chi) \longrightarrow (\phi \otimes \psi) ⅋ \chi \;\Bigg|\; \begin{array}{l} \chi \otimes \kappa(\!|\phi_1, \ldots, \phi_k, \psi, \phi_{k+1}, \ldots, \phi_n|\!) \longrightarrow \kappa(\!|\phi_1, \ldots, \phi_k, \psi \otimes \chi, \phi_{k+1}, \ldots, \phi_n|\!) \\ \kappa(\!|\phi_1, \ldots, \phi_k, \psi ⅋ \chi, \phi_{k+1}, \ldots, \phi_n|\!) \longrightarrow \kappa(\!|\phi_1, \ldots, \phi_k, \psi, \phi_{k+1}, \ldots, \phi_n|\!) ⅋ \chi \end{array} \tag{10}$$

Note that an additional law is required to formalize the weak-distributivity law of all connectives over ⅋ (see above on the bottom-right). This law corresponds to the rule wd$_{⅋}$ in Figure 5.

Notation 35. Unless strictly needed for sake of clarity, we omit to the permutations over the indices of the subformulas in rules.

3.1 Properties of the systems MGL and MGL$^{\circ}$

We start by observing that these systems are *initial coherent* [10, 50], that is, we can derive the implication $\phi \multimap \phi$ for any formula ϕ only using atomic axioms. To prove this result we observe that the generalized version of d-κ (that is, the rule d-χ) is derivable by induction on the structure of χ using the rule d-κ.

$$\mathsf{cut}\dfrac{\mathsf{ax}\dfrac{}{\vdash a, a^{\perp}} \quad \vdash a, \Gamma}{\vdash a, \Gamma} \rightsquigarrow \vdash a^{\perp}, \Gamma \qquad \mathsf{cut}\dfrac{\otimes\dfrac{\vdash \Gamma, \phi \quad \vdash \Delta, \psi}{\vdash \Gamma, \Delta, \phi \otimes \psi} \quad \invamp\dfrac{\vdash \Sigma, \phi^{\perp} \invamp \psi^{\perp}}{\vdash \Sigma, \phi^{\perp} \invamp \psi^{\perp}}}{\vdash \Gamma, \Delta, \Sigma} \rightsquigarrow \mathsf{cut}\dfrac{\vdash \Gamma, \phi \quad \mathsf{cut}\dfrac{\vdash \Delta, \psi \quad \vdash \Sigma, \phi^{\perp}, \psi^{\perp}}{\vdash \Delta, \Sigma, \phi^{\perp}}}{\vdash \Gamma, \Delta, \Sigma}$$

$$\mathsf{cut}\dfrac{\mathsf{d}\text{-}\kappa\dfrac{\vdash \Gamma_1, \phi_1, \psi_1 \quad \cdots \quad \vdash \Gamma_n, \phi_n, \psi_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa(\!|\phi_1, \ldots, \phi_n|\!), \kappa^{\perp}(\!|\psi_1, \ldots, \psi_n|\!)} \quad \mathsf{d}\text{-}\kappa\dfrac{\vdash \Delta_1, \psi_1^{\perp}, \chi_1 \quad \cdots \quad \vdash \Delta_n, \psi_n^{\perp}, \chi_n}{\vdash \Delta_1, \ldots, \Delta_n, \kappa(\!|\psi_1^{\perp}, \ldots, \psi_n^{\perp}|\!), \kappa^{\perp}(\!|\chi_1, \ldots, \chi_n|\!)}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n, \kappa(\!|\phi_1, \ldots, \phi_n|\!), \kappa^{\perp}(\!|\chi_1, \ldots, \chi_n|\!)}$$

$$\wr\!\wr$$

$$\mathsf{d}\text{-}\kappa\dfrac{\mathsf{cut}\dfrac{\vdash \Gamma_1, \phi_1, \psi_1 \quad \vdash \Delta_1, \psi_1^{\perp}, \chi_1}{\vdash \Gamma_1, \Delta_1, \phi_1, \chi_1} \quad \cdots \quad \mathsf{cut}\dfrac{\vdash \Gamma_n, \phi_n, \psi_n \quad \vdash \Delta_n, \psi_n^{\perp}, \chi_n}{\vdash \Gamma_n, \phi_n, \chi_n}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n, \kappa^{\perp}(\!|\psi_1, \ldots, \psi_n|\!), \kappa(\!|\chi_1, \ldots, \chi_n|\!)}$$

Figure 6: Cut-elimination steps for MGL.

Therefore, we can prove that the generalized non-atomic axiom rule (AX) is derivable, and that both MGL and MGL$^{\circ}$ are initial coherent

Lemma 36. *Let χ be a pure formula. Then rule* d-χ *is derivable.*

Proof. By induction on the structure of χ:

- if $\phi = a$ is a literal, then AX is an instance of ax;
- if $\phi = \kappa\psi_1, \ldots, \psi_k, \circ, \psi_{k+1}, \ldots, \psi_n$, then apply twice unitor_{κ} to the sequent $\vdash \phi, \phi^{\perp}$ to obtain the sequent of pure formulas $\vdash \kappa_{\chi}\psi_1, \ldots, \psi_n, \kappa_{\chi^{\perp}}\psi_1^{\perp}, \ldots, \psi_n^{\perp}$. We conclude by inductive hypothesis;
- if $\phi = \kappa(\!|\psi_1, \ldots, \psi_n|\!)$ and $\psi_i \neq \circ$ for all $i \in \{1, \ldots, n\}$, then apply the rule d-κ to obtain sequents of pure formulas the form $\psi_i, \psi_i^{\perp}$ for all $i \in \{1, \ldots, |\kappa|\}$. We conclude by inductive hypothesis.

□

Corollary 37. *The rule* AX *is derivable in* MGL *and in* MGL$^{\circ}$.

Theorem 38. *The systems* MGL *and* MGL$^{\circ}$ *are initial coherent (with respect to pure formulas).*

We then prove the admissibility of cut via *cut-elimination* by providing a cut-elimination procedure.

Theorem 39 (Cut-elimination)**.** *Let* X $\in$ {MGL, MGL$^{\circ}$}. *The rule* cut *is admissible in* X.

Proof. We define the *size* of a formula as sum of the number of $\circ$, connectives and twice the number of literals in it. The *size* of a derivation is the sum of the sizes of the active formulas in all cut-rules. The result follows by the fact that each *cut-elimination step* from Figures 6 and 7 reduces the size of a derivation.

Note that in order to ensure that both active formulas of a cut are principal with respect to the rule immediately above it we also need to consider the *commutative* cut-elimination steps from Figure 8. The treatment of these rule, as well as the definition of a size taking into account them, is not covered in the detail here because it is standard in the literature (see, e.g., [56]). □

Corollary 40. *Let* X $\in$ {MGL, MGL$^{\circ}$}. *If* $\vdash_{\mathsf{X}} \phi \multimap \psi$ *and* $\vdash_{\mathsf{X}} \psi \multimap \chi$, *then* $\vdash_{\mathsf{X}} \phi \multimap \chi$.

The admissibility of cut implies analyticity of MGL via the standard *sub-formula property*, that is, all (occurrences of) formulas occurring in the premises of a rule are subformulas of the ones in the conclusion.

Corollary 41 (Analyticity of MGL)**.** *Let* Γ *be a sequent. If* $\vdash_{\mathsf{MGL}} \Gamma$ *then there is a proof of* Γ *in* MGL *only containing occurrences of sub-formulas of formulas* Γ.

$$
\text{cut}\,\dfrac{\text{unitor}_\kappa\,\dfrac{\vdash \Gamma, \chi(\!|\phi_2,\ldots,\phi_n|\!)}{\vdash \Gamma, \kappa_P(\!|\circ,\phi_2,\ldots,\phi_n|\!)} \qquad \text{unitor}_\kappa\,\dfrac{\vdash \Delta, \chi^\perp(\!|\phi_2^\perp,\ldots,\phi_n^\perp|\!)}{\vdash \Delta, \kappa_{P^\perp}(\!|\circ,\phi_2^\perp,\ldots,\phi_n^\perp|\!)}}{\vdash \Gamma, \Delta} \rightsquigarrow \text{cut}\,\dfrac{\vdash \Gamma, \chi(\!|\phi_2,\ldots,\phi_n|\!) \qquad \vdash \Delta, \chi^\perp(\!|\phi_2^\perp,\ldots,\phi_n^\perp|\!)}{\vdash \Gamma, \Delta}
$$

$$
\text{cut}\,\dfrac{\text{wd}_\otimes\,\dfrac{\vdash \Gamma_1,\phi_1 \quad \vdash \Gamma_2, \kappa_P(\!|\circ,\phi_2,\ldots,\phi_n|\!)}{\vdash \Gamma_1,\Gamma_2,\kappa_P(\!|\phi_1,\ldots,\phi_n|\!)} \qquad \text{wd}_\otimes\,\dfrac{\vdash \Delta_1,\phi_1^\perp \quad \vdash \Delta_2, \kappa_{P^\perp}(\!|\circ,\phi_2^\perp,\ldots,\phi_n^\perp|\!)}{\vdash \Delta, \kappa_{P^\perp}(\!|\phi_1^\perp,\ldots,\phi_n^\perp|\!)}}{\vdash \Gamma_1,\Gamma_2,\Delta_1,\Delta_2}
$$

$$
\rightsquigarrow
$$

$$
\text{mix}\,\dfrac{\text{cut}\,\dfrac{\vdash \Gamma_1,\phi_1 \quad \vdash \Delta_1,\phi_1^\perp}{\vdash \Gamma_1,\Delta_1} \qquad \text{cut}\,\dfrac{\vdash, \Gamma_2, \kappa_P(\!|\circ,\phi_2,\ldots\phi_n|\!) \quad \vdash \Delta_2, \kappa_{P^\perp}(\!|\circ,\phi_2^\perp,\ldots\phi_n^\perp|\!)}{\vdash \Gamma_2,\Delta_2}}{\vdash \Gamma_1,\Gamma_2,\Delta_1,\Delta_2}
$$

$$
\text{cut}\,\dfrac{\text{d-}\kappa\,\dfrac{\vdash \Gamma_1,\phi_1,\psi_1 \quad \cdots \quad \vdash \Gamma_n,\phi_n,\psi_n}{\vdash \Gamma_1,\ldots,\Gamma_n,\kappa_P(\!|\phi_1,\ldots,\phi_n|\!),\kappa_{P^\perp}(\!|\psi_1,\ldots,\psi_n|\!)} \qquad \text{wd}_\otimes\,\dfrac{\vdash \Delta,\psi_1^\perp \quad \vdash \Sigma, \kappa_P(\!|\circ,\psi_2^\perp,\ldots,\psi_n^\perp|\!)}{\vdash \Delta,\Sigma,\kappa_P(\!|\psi_1^\perp,\ldots,\psi_n^\perp|\!)}}{\vdash \Gamma_1,\ldots,\Gamma_n,\Delta,\Sigma,\kappa_P(\!|\phi_1,\ldots,\phi_n|\!)}
$$

$$
\rightsquigarrow
$$

$$
\text{wd}_\otimes\,\dfrac{\text{cut}\,\dfrac{\vdash \Gamma_1,\phi_1,\psi_1 \quad \vdash \Delta,\psi_1^\perp}{\vdash \Gamma_1,\Delta,\phi_1} \qquad \text{cut}\,\dfrac{2\times\text{unitor}_\kappa\,\dfrac{\text{d-}\chi\,\dfrac{\vdash \Gamma_2,\phi_2,\psi_2 \quad \cdots \quad \vdash \Gamma_n,\phi_n,\psi_n}{\vdash \Gamma_2,\ldots,\Gamma_n,\kappa_\chi(\!|\phi_1,\ldots,\phi_n|\!),\kappa_\chi^\perp(\!|\psi_1,\ldots,\psi_n|\!)}}{\vdash \Gamma_2,\ldots,\Gamma_n,\kappa_P(\!|\circ,\phi_1,\ldots,\phi_n|\!),\kappa_{P^\perp}(\!|\circ,\psi_1,\ldots,\psi_n|\!)} \quad \vdash \Sigma,\kappa_P(\!|\circ,\psi_2^\perp,\ldots,\psi_n^\perp|\!)}{\vdash \Gamma_2,\ldots,\Gamma_n,\Sigma,\kappa_P(\!|\circ,\phi_2,\ldots,\phi_n|\!)}}{\vdash \Gamma_1,\ldots,\Gamma_n,\Delta,\Sigma,\kappa_P(\!|\phi_1,\ldots,\phi_n|\!)}
$$

Figure 7: Additional cut-elimination steps in GML°.

However, the same result does not hold for MGL° because of the rule unitor_κ. In fact, the presence of more-than-binary connectives and their units (in this case, a unique unit $\circ$) implies, as observed in the previous works on graphical logic [3, 4, 1], the possibility of having *sub-connectives*, that is, connectives with smaller arity behaving as if certain entries of the connective are fixed to be units.

Definition 42. Let P and P' be prime graphs. If $P(\!|\circ,\ldots,\circ,v_{i_1},\circ,\ldots,\circ,v_{i_k},\circ,\ldots,\circ|\!) \sim P'(\!|v_1,\ldots,v_n|\!)$ for single-vertex graphs $v_1,\ldots,v_n$ and for some $i_1,\ldots,i_k \in \{1,\ldots,n\}$ such that $i_i < \cdots < i_k$, then we may write $\kappa_{P|_{i_1,\ldots,i_k}} = \kappa_{Q'}$ and we say that the connective $\kappa_{P'}$ is a ***sub-connective*** of if κ_P.

A ***quasi-subformula*** of a formula $\phi = \kappa_P(\!|\psi_1,\ldots,\psi_n|\!)$ is a formula of the form $\kappa_{P'|_{i_1,\ldots,i_k}}(\!|\psi'_{i_1},\ldots,\psi'_{i_k}|\!)$ with ψ'_{i_j} a quasi-subformula of ψ_{i_j} for all $i_j \in \{i_1,\ldots,i_k\}$.

Corollary 43 (Analyticity of MGL°)**.** *Let Γ be a sequent. If $\vdash_{\mathsf{MGL}^\circ} \Gamma$ then there is a proof of Γ in MGL° only containing occurrences of quasi-subformula of formulas in Γ.*

Corollary 44 (Conservativity)**.** *The logic* MGL *is a conservative extension of* MLL. *The logic* MGL° *is a conservative extension of* MLL°.

Proof. For MGL it is consequence of the subformula property. For MGL° it suffices to remark that $\invamp$ and $\otimes$ have no sub-connectives, therefore quasi-subformula are simply sub-formulas. □

For both MGL and MGL° we have the following result which takes the name of *splitting* in the deep inference literature (see, e.g, [7, 35, 36]). This result states that is always possible, during proof search, to apply a rule removing a connective after having applied certain rules in the context.[10]

[10] Note that in the linear logic literature, "splitting" is only used to refer to the special case in which a $\otimes$ is removed, without requiring the application of rules to the context.

$$
\rho\,\dfrac{\mathsf{cut}\,\dfrac{\dfrac{\vdash \Gamma_1, \Delta', \phi}{\vdash \Gamma_1, \Delta, \phi} \quad \vdash \phi^\perp, \Gamma_2}{\vdash \Gamma_1, \Gamma_2, \Delta}}{} \quad \rightsquigarrow \quad \rho\,\dfrac{\mathsf{cut}\,\dfrac{\vdash \Gamma_1, \Delta', \phi \quad \vdash \phi^\perp, \Gamma_2}{\vdash \Gamma_1, \Gamma_2, \Delta}}{\vdash \Gamma_1, \Gamma_2, \Delta'}
$$

$$
\mathsf{cut}\,\dfrac{\rho\,\dfrac{\vdash \Gamma_1, \Delta'_1 \quad \cdots \quad \vdash \Gamma_n, \Delta'_n, \phi}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta, \phi} \quad \vdash \phi^\perp, \Gamma_{n+1}}{\vdash \Gamma_1, \ldots, \Gamma_{n+1}, \Delta} \quad \rightsquigarrow \quad \rho\,\dfrac{\vdash \Gamma_1, \Delta'_1 \quad \cdots \quad \vdash \Gamma_{n-1}, \Delta'_{n-1} \quad \mathsf{cut}\,\dfrac{\vdash \Gamma_n, \Delta'_n, \phi \quad \vdash \Gamma_{n+1}, \phi^\perp}{\vdash \Gamma_n, \Gamma_{n+1}, \Delta'_n}}{\vdash \Gamma_1, \ldots, \Gamma_{n+1}, \Delta}
$$

$$
\mathsf{unitor}_\kappa\,\dfrac{\mathsf{unitor}_\kappa\,\dfrac{\vdash \Gamma, \chi(\!|\phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_{j-1}, \phi_{j+1}, \ldots, \phi_n|\!)}{\vdash \Gamma, \kappa_P(\!|\phi_1, \ldots, \phi_{i-1}, \circ, \phi_{i+1}, \ldots, \phi_{j-1}, \phi_{j+1}, \ldots, \phi_n|\!)}}{\vdash \Gamma\kappa_P(\!|\phi_1, \ldots, \phi_{i-1}, \circ, \phi_{i+1}, \ldots, \phi_{j-1}, \circ, \phi_{j+1}, \ldots, \phi_n|\!)} \quad \rightsquigarrow \quad \mathsf{unitor}_\kappa\,\dfrac{\mathsf{unitor}_\kappa\,\dfrac{\vdash \Gamma, \chi(\!|\phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_{j-1}, \phi_{j+1}, \ldots, \phi_n|\!)}{\vdash \Gamma, \kappa_{P'}(\!|\phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_{j-1}, \circ, \phi_{j+1}, \ldots, \phi_n|\!)}}{\vdash \Gamma, \kappa_P(\!|\phi_1, \ldots, \phi_{i-1}, \circ, \phi_{i+1}, \ldots, \phi_{j-1}, \circ, \phi_{j+1}, \ldots, \phi_n|\!)}
$$

Figure 8: Commutative cut-elimination steps.

Lemma 45 (Splitting)**.** *Let $\Gamma, \kappa(\!|\phi_1, \ldots, \phi_n|\!)$ be a sequent and let $\mathsf{X} \in \{\mathsf{MGL}, \mathsf{MGL}^\circ\}$. If $\vdash_{\mathsf{X}} \Gamma, \kappa(\!|\phi_1, \ldots, \phi_n|\!)$, then there is a derivation of the following shape*

$$
\begin{array}{c} \pi_1 \,\|\!\| \\ \mathsf{unitor}_\kappa\,\dfrac{\vdash \Gamma', \chi(\!|\phi_1, \ldots, \phi_{k-1}, \phi_{k+1}, \phi_n|\!)}{\vdash \Gamma', \kappa(\!|\phi_1, \ldots, \phi_{k-1}, \circ, \phi_{k+1}, \phi_n|\!)} \\ \pi_0 \,\| \\ \vdash \Gamma, \kappa(\!|\phi_1, \ldots, \phi_{k-1}, \circ, \phi_{k+1}, \phi_n|\!) \end{array} \quad \textit{or} \quad \begin{array}{c} \pi_1 \,\|\!\| \qquad\qquad \pi_n \,\|\!\| \\ \rho\,\dfrac{\vdash \Delta_1, \phi_1 \quad \cdots \quad \vdash \Delta_n, \phi_n}{\vdash \Gamma', \kappa(\!|\phi_1, \ldots, \phi_n|\!)} \\ \pi_0 \,\| \\ \vdash \Gamma, \kappa(\!|\phi_1, \ldots, \phi_n|\!) \end{array} \ \textit{with } \rho \in \{\parr, \otimes, \mathsf{d}\text{-}\kappa\}
$$

Proof. By case analysis of the last rule occurring in a proof π of $\Gamma, \kappa(\!|\phi_1, \ldots, \phi_n|\!)$:

- the last rule cannot be a ax since $\kappa(\!|\phi_1, \ldots, \phi_n|\!)$ contains at least one connective;
- if the last rule is a $\parr$ or a unitor_κ, then either this is the desired rule, or we conclude by inductive hypothesis on its premise;
- if the last rule is a mix, then we conclude by inductive hypothesis on the premise containing the formula $\kappa(\!|\phi_1, \ldots, \phi_n|\!)$;
- if the last rule is in $\{\otimes, \mathsf{d}\text{-}\kappa, \mathsf{wd}_\otimes, \mathsf{unitor}_\kappa\}$ then either this is the desired rule or one of the (provable) premises of this rule is of the shape $\Gamma', \kappa(\!|\phi_1, \ldots, \phi_n|\!)$, allowing us to conclude by inductive hypothesis. □

We conclude this section proving the admissibility of the rule $\mathsf{wd}_\parr$ which we use to simplify proofs in the next section.

Lemma 46. *The rule $\mathsf{wd}_\parr$ is admissible in MGL°.*

Proof. In Figure 9 we providde a procedure to remove (top-down) all occurrences of $\mathsf{wd}_\parr$. Similar to cut-elimination, we use the commutative steps from Figure 8 to ensure that the active formula of the $\mathsf{wd}_\parr$ we want to remove is principal with respect to the rule immediately above it. □

Lemma 47. *The rule deep is admissible in MGL°.*

Proof. Since $\zeta[\circ] \neq \circ$, then w.l.o.g., $\zeta[\Box] = \kappa(\!|\zeta'[\Box], \psi'_1, \ldots, \psi'_n|\!)$. If $\zeta'[\Box] = \Box$, then w.l.o.g., $\psi = \chi(\!|\psi'_1, \ldots, \psi'_n|\!)$ and we conclude since we have

$$
\mathsf{wd}_\parr\,\dfrac{\vdash \Gamma, \phi \qquad \mathsf{unitor}_\kappa\,\dfrac{\vdash \Delta, \chi(\!|\psi'_1, \ldots, \psi'_n|\!)}{\vdash \Delta, \kappa(\!|\circ, \psi'_1, \ldots, \psi'_n|\!)}}{\vdash \Gamma, \Delta, \kappa(\!|\phi, \psi'_1, \ldots, \psi'_n|\!)}
$$

Otherwise we conclude by inductive hypothesis on the size of $\zeta[\Box]$ since by Lemma 45 we can define

$$\mathsf{wd}_{\parr}\,\dfrac{\parr\,\dfrac{\overset{\pi_1}{\vdash \Gamma,\phi,\psi}}{\vdash \Gamma,\phi\parr\psi}}{\vdash \Gamma,\phi,\circ\parr\psi} \quad\rightsquigarrow\quad \mathsf{unitor}_\kappa\,\dfrac{\overset{\pi_1}{\vdash \Gamma,\phi,\psi}}{\vdash \Gamma,\phi,\circ\parr\psi} \qquad\qquad \mathsf{wd}_{\otimes}\,\dfrac{\otimes\,\dfrac{\overset{\pi_1}{\vdash \Gamma,\phi} \quad \overset{\pi_2}{\vdash \Delta,\psi}}{\vdash \Gamma,\Delta,\phi\otimes\psi}}{\vdash \Gamma,\Delta,\phi,\circ\otimes\psi} \quad\rightsquigarrow\quad \mathsf{unitor}_\kappa\,\dfrac{\mathsf{mix}\,\dfrac{\overset{\pi_1}{\vdash \Gamma,\phi} \quad \overset{\pi_2}{\vdash \Delta,\psi}}{\vdash \Gamma,\Delta,\phi,\psi}}{\vdash \Gamma,\Delta,\phi,\circ\otimes\psi}$$

$$\mathsf{wd}_{\parr}\,\dfrac{\mathsf{d}\text{-}\kappa\,\dfrac{\overset{\pi_1}{\vdash \Gamma_1,\phi,\psi_1} \quad \overset{\pi_2}{\vdash \Gamma_2,\phi_2,\psi_2} \quad \cdots \quad \overset{\pi_n}{\vdash \Gamma_n,\phi_n,\psi_n}}{\vdash \Gamma_1,\ldots,\Gamma_n,\kappa^{\perp}\llparenthesis\psi_1,\ldots,\psi_n\rrparenthesis,\kappa\llparenthesis\phi,\phi_2,\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma_1,\ldots,\Gamma_n,\kappa^{\perp}\llparenthesis\psi_1,\ldots,\psi_n\rrparenthesis,\kappa\llparenthesis\circ,\phi_2,\ldots,\phi_n\rrparenthesis,\phi} \quad\rightsquigarrow\quad \mathsf{wd}_{\otimes}\,\dfrac{\overset{\pi_1}{\vdash \Gamma_1,\phi,\psi_1} \quad 2\times\mathsf{unitor}_\kappa\,\dfrac{\mathsf{d}\text{-}\chi\,\dfrac{\overset{\pi_2}{\vdash \Gamma_2,\psi_2,\chi_2} \quad \cdots \quad \overset{\pi_n}{\vdash \Gamma_n,\psi_n,\chi_n}}{\vdash \Gamma_2,\ldots,\Gamma_n,\chi^{\perp}\llparenthesis\psi_2,\ldots,\psi_n\rrparenthesis,\chi\llparenthesis\phi_2,\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma_2,\ldots,\Gamma_n,\kappa^{\perp}\llparenthesis\circ,\psi_1,\ldots,\psi_n\rrparenthesis,\kappa\llparenthesis\circ,\phi_2,\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma_1,\ldots,\Gamma_n,\kappa^{\perp}\llparenthesis\psi_1,\ldots,\psi_n\rrparenthesis,\kappa\llparenthesis\circ,\phi_2,\ldots,\phi_n\rrparenthesis,\phi}$$

$$\mathsf{wd}_{\parr}\,\dfrac{\mathsf{wd}_{\otimes}\,\dfrac{\overset{\pi_1}{\vdash \Gamma_1,\psi_k} \quad \overset{\pi_2}{\vdash \Gamma_2,\kappa\llparenthesis\phi,\psi_2,\ldots,\psi_{k-1},\circ,\psi_{k+1},\ldots,\psi_n\rrparenthesis}}{\vdash \Gamma_1,\Gamma_2,\kappa\llparenthesis\phi,\psi_2,\ldots,\psi_n\rrparenthesis}}{\vdash \Gamma_1,\Gamma_2,\kappa\llparenthesis\circ,\psi_2,\ldots,\psi_n\rrparenthesis,\phi} \quad\rightsquigarrow\quad \mathsf{wd}_{\otimes}\,\dfrac{\overset{\pi_1}{\vdash \Gamma_1,\psi'} \quad \mathsf{wd}_{\parr}\,\dfrac{\overset{\pi_2}{\vdash \Gamma_2,\kappa\llparenthesis\phi,\psi_2,\ldots,\psi_{k-1},\circ,\psi_{k+1},\ldots,\psi_n\rrparenthesis}}{\vdash \Gamma_2,\kappa\llparenthesis\circ,\psi_2,\ldots,\psi_{k-1},\circ,\psi_{k+1},\ldots,\psi_n\rrparenthesis,\phi}}{\vdash \Gamma,\kappa\llparenthesis\circ,\psi_2,\ldots,\psi_k,\psi',\psi_{k+1},\ldots,\psi_n\rrparenthesis,\phi}$$

$$\mathsf{wd}_{\parr}\,\dfrac{\mathsf{unitor}_\kappa\,\dfrac{\overset{\pi_1}{\vdash \Gamma,\chi\llparenthesis\phi,\psi_2,\ldots,\psi_{n-1}\rrparenthesis}}{\vdash \Gamma,\kappa\llparenthesis\phi,\psi_2,\ldots,\psi_{n-1},\circ\rrparenthesis}}{\vdash \Gamma,\kappa\llparenthesis\circ,\psi_2,\ldots,\psi_{n-1},\circ\rrparenthesis,\phi} \quad\rightsquigarrow\quad \mathsf{unitor}_\kappa\,\dfrac{\overset{\pi'_1\ \mathsf{IH}}{\vdash \Gamma,\chi\llparenthesis\circ,\psi_2,\ldots,\psi_{n-1}\rrparenthesis,\phi}}{\vdash \Gamma,\kappa\llparenthesis\circ,\psi_2,\ldots,\psi_{n-1},\circ\rrparenthesis,\phi}$$

Figure 9: Steps to eliminate $\mathsf{wd}_{\parr}$ rules.

a derivation of the form

$$\begin{array}{c}\mathsf{unitor}_\kappa\,\dfrac{\overset{\mathsf{IH}}{\vdash \Gamma',\Delta',\chi\llparenthesis\zeta[\phi],\psi_1,\ldots,\psi_n\rrparenthesis}}{\vdash \Gamma',\Delta',\kappa\llparenthesis\zeta[\phi],\psi_1,\ldots,\psi_{k-1},\circ,\psi_{k+1}\ldots\psi_n\rrparenthesis} \\ \pi_0 \,\| \\ \vdash \Gamma,\Delta,\kappa\llparenthesis\zeta[\phi],\psi_1,\ldots,\psi_{k-1},\circ,\psi_{k+1}\ldots\psi_n\rrparenthesis\end{array} \quad\text{or}\quad \begin{array}{c}\rho\,\dfrac{\overset{\mathsf{IH}}{\vdash \Gamma',\Delta_0,\zeta[\phi]} \quad \overset{\pi_1}{\vdash \Delta_1,\psi'_1} \quad \cdots \quad \overset{\pi_n}{\vdash \Delta_n,\psi'_n}}{\vdash \Gamma',\Delta',\kappa\llparenthesis\zeta[\phi],\psi'_1,\ldots,\psi'_n\rrparenthesis} \\ \pi_0 \,\| \\ \vdash \Gamma,\Delta,\kappa\llparenthesis\zeta[\phi],\psi'_1,\ldots,\psi'_n\rrparenthesis\end{array}$$

with $\rho \in \{\parr,\otimes,\mathsf{d}\text{-}\kappa\}$. □

3.2 Soundness of Logical Equivalence in MGL°

In this sub-section we prove that if two formulas ϕ and ψ interpreted by a same graph (i.e., $[\![\phi]\!] = [\![\phi]\!]$) iff they are logically equivalent (i.e., $\phi \multimapboth \psi$). For this purpose, we show that all equivalence and De Morgan laws from Definition 29 can be reformulated as logical equivalence which are derivable in MGL.

Lemma 48. *The following implications are provable in* MGL *whenever they are unit-free:*

$$\begin{array}{rll} \textit{for all } \sigma \in \mathfrak{S}(P): & \kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis \multimap \kappa_P\llparenthesis\phi_{\sigma(1)},\ldots,\phi_{\sigma(n)}\rrparenthesis & \kappa_P\llparenthesis\phi_{\sigma(1)},\ldots,\phi_{\sigma(n)}\rrparenthesis \multimap \kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis \\ \textit{for all } \tau \in \mathfrak{S}^{\perp}(P): & \kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis \multimap \kappa_{P^\perp}\llparenthesis\phi_{\rho(1)},\ldots,\phi_{\rho(n)}\rrparenthesis & \kappa_{P^\perp}\llparenthesis\phi_{\rho(1)},\ldots,\phi_{\rho(n)}\rrparenthesis \multimap \kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis \\ & ((\phi\parr\psi)\parr\chi) \multimap (\phi\parr(\psi\parr\chi)) & (\phi\parr(\psi\parr\chi)) \multimap ((\phi\parr\psi)\parr\chi) \\ & ((\phi\otimes\psi)\otimes\chi) \multimap (\phi\otimes(\psi\otimes\chi)) & (\phi\otimes(\psi\otimes\chi)) \multimap ((\phi\otimes\psi)\otimes\chi) \end{array}$$

Proof. The implications in the first two lines are derivable using an instance of $\parr$ followed by a $\mathsf{d}\text{-}\kappa$ and some AX-rules. The remaining implications are derivable applying four $\parr$-rules followed by three $\otimes$-rules and AX-rules. □

We conclude that MGL is sound and complete with respect to graph isomorphism for non-empty graphs.

Theorem 49. *Let ϕ and ψ unit-free formulas. Then $[\![\phi]\!] = [\![\psi]\!]$ iff $\vdash_{\mathsf{MGL}} \phi \multimapboth \psi$.*

Proof. By Corollary 40, Proposition 31 and Lemma 48. □

4 Soundness and Completeness of MGL° with respect to GS

In this section we prove that set of graphs which are derivable in the graphical logic GS from [3, 4] is the same set of graph corresponding to formulas which are provable in MGL°.

In Figure 10 we recall the definition of the rules of the deep inference system[11] $\mathsf{GS} = \{\mathsf{ai}\!\downarrow, \mathsf{s}_{\parr}, \mathsf{s}_{\otimes}, \mathsf{p}\!\downarrow\}$.

[11] The definition of deep inference systems operating on graphs can be found in [4] or in Appendix A.

$$\text{ai}\downarrow\frac{\varnothing}{a^\perp \parr a} \qquad \text{p}\downarrow\frac{(M_1 \parr N_1)\otimes\cdots\otimes(M_n \parr M'_n)}{P^\perp(\!|M_1,\ldots,M_n|\!) \parr P(\!|M'_1,\ldots,M'_n|\!)}$$

$$\text{s}_\parr\frac{P(\!|M_1,\ldots,M_{i-1},M_i \parr N,M_{i+1},\ldots M_n|\!)}{M_i \parr P(\!|M_1,\ldots,M_{i-1},N,M_{i+1},\ldots,M_n|\!)} \qquad \text{s}_\otimes\frac{M_i\otimes P(\!|M_1,\ldots,M_{i-1},N,M_{i+1},\ldots,M_n|\!)}{P(\!|M_1,\ldots,M_{i-1},M_i\otimes N,M_{i+1},\ldots,M_n|\!)}$$

Figure 10: Inference rules for the system GS, where P is a prime graph and $M_i \neq \varnothing \neq M'_i$ for all $i \in \{1,\ldots,n\}$.

Remark 50. At the syntactical level, the system GS operates on graphs by manipulating their modular decompositions trees. Therefore, for any graph occurring in a derivation in GS we assume a unique formula $[\![G]\!]^{-1}$ to be given. Note that in GS the authors allow themselves to consider modular decomposition trees in which leaves may be empty graphs, corrisponding to formulas with unit.

Remark 51. The set of rules we consider here is a slightly different formulation of with respect to [3] and [4]: we consider a p-rules with a stronger side condition (all factors to be non-empty) which is balanced by the presence of $\text{s}_\otimes$ in the system. The proof that the formulation we consider in this paper is equivalent to the ones in the literature is provided in Appendix A.1.

We can easily prove that each sequent provable in MGL° is interpreted by $[\![\cdot]\!]$ as a graph which is admitting a proof in GS.

Lemma 52. *Let* Γ *be a sequent. If* $\vdash_{\text{MGL}^\circ} \Gamma$*, then* $\vdash_{\text{GS}} [\![\Gamma]\!]$.

Proof. We define a derivation $[\![\pi]\!]$ of $[\![\Gamma]\!]$ in GS by induction by induction on the last rule r in a derivation π of Γ in MGL° according to Figure 11. □

To prove the converse, we use the admissibility of $\text{wd}_\parr$ to prove that every time there is a rule in GS with premise H and conclusion G, then there are formulas ϕ and ψ such that $[\![\phi]\!]$ and $[\![\psi]\!]$, and such that $\psi \multimap \phi$.

Lemma 53. *Let* $\text{r} \in \{\text{s}_\parr, \text{s}_\otimes, \text{p}\downarrow\}$*. If* $\text{r}\frac{H}{G}$*, then there are formulas* ϕ *and* ψ *with* $[\![\phi]\!] = G$ *and* $[\![\psi]\!] = H$ *such that* $\vdash_{\text{MGL}^\circ} \psi^\perp, \phi$.

Proof. We first discuss the case if $C[\square] = \square$:

- if $\text{r} = \text{s}_\parr$, then $\phi = \mu_i \parr \kappa(\!|\mu_1,\ldots,\mu_{i-1},\circ \parr \nu,\mu_{i+1},\ldots\mu_n|\!)$ and $\psi = \kappa(\!|\mu_1,\ldots,\mu_{i-1},\mu_i \parr \nu,\mu_{i+1},\ldots\mu_n|\!)$ for some formulas $\mu_1,\ldots,\mu_n,\nu$ such that $[\![\mu_i]\!] = M_i$ for all $i \in \{1,\ldots,n\}$ and $[\![\nu]\!] = N$. We conclude by Corollary 37 and lemma 46 since we have the following derivation

$$\parr\frac{\text{wd}_\parr\dfrac{\text{AX}\dfrac{}{\vdash \psi^\perp,\kappa(\!|\mu_1,\ldots,\mu_{i-1},\mu_i \parr \nu,\mu_{i+1},\ldots,\mu_n|\!)}}{\vdash \psi^\perp,\mu_i,\kappa(\!|\mu_1,\ldots,\mu_{i-1},\circ \parr \nu,\mu_{i+1},\ldots,\mu_n|\!)}}{\vdash \psi^\perp,\phi}$$

- if $\text{r} = \text{s}_\otimes$ then $\phi = \kappa(\!|\mu_1,\ldots,\mu_{i-1},\mu_i\otimes\nu,\mu_{i+1},\ldots\mu_n|\!)$ and $\psi = \mu_i\otimes\kappa(\!|\mu_1,\ldots,\mu_{i-1},\circ\otimes\nu,\mu_{i+1},\ldots\mu_n|\!)$ for some formulas $\mu_1,\ldots,\mu_n,\nu$ such that $[\![\mu_i]\!] = M_i$ for all $i \in \{1,\ldots,n\}$ and $[\![\nu]\!] = N$. We conclude by Corollary 37 and lemma 46 since we have the following derivation

$$\parr\frac{\text{cxt-}\parr\dfrac{\text{AX}\dfrac{}{\vdash \kappa^\perp(\!|\mu_1^\perp,\ldots,\mu_{i-1}^\perp,\mu_i^\perp \parr \nu^\perp,\mu_{i+1}^\perp,\ldots\mu_n^\perp|\!),\phi}}{\vdash \mu_i^\perp,\kappa^\perp(\!|\mu_1^\perp,\ldots,\mu_{i-1}^\perp,\circ \parr \nu^\perp,\mu_{i+1}^\perp,\ldots\mu_n^\perp|\!),\phi}}{\vdash \psi^\perp,\phi}$$

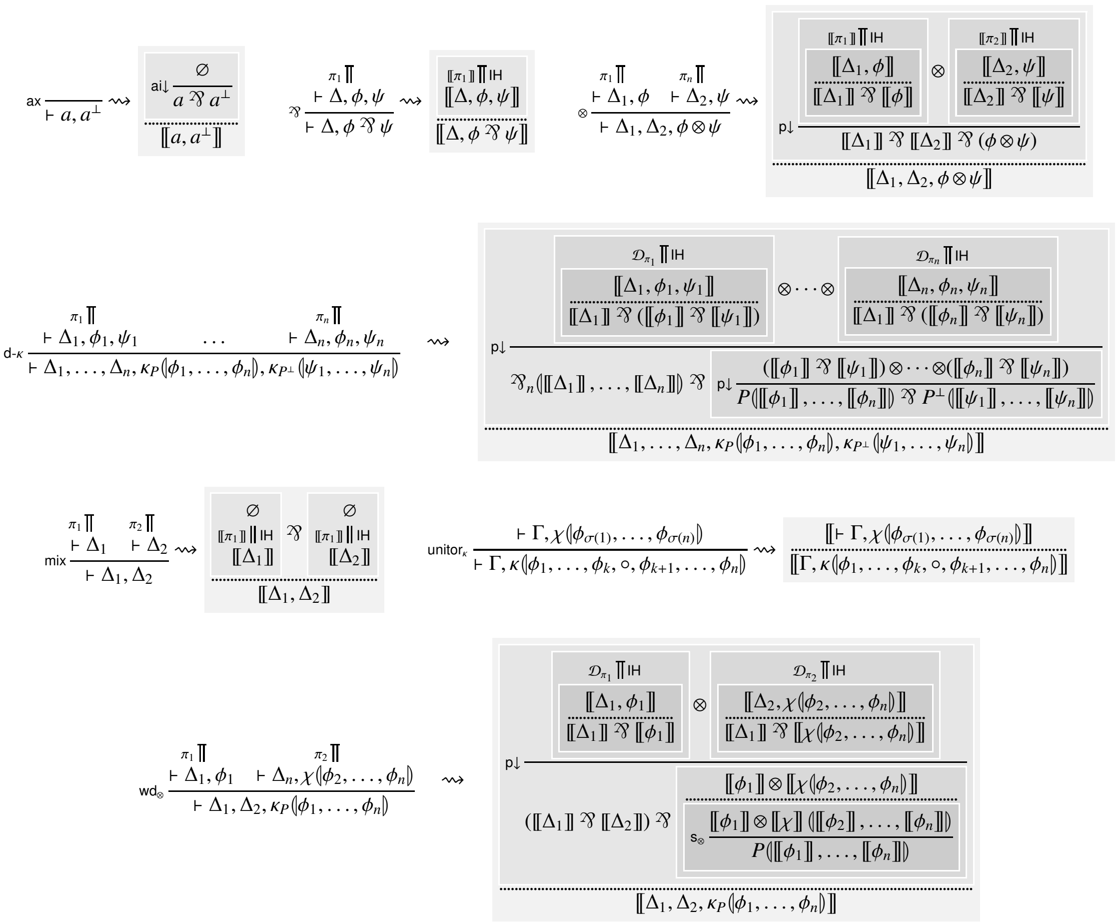


Figure 11: Rules to translate derivations in MGL° into derivations in GS.

- if $\mathsf{r} = \mathsf{p}{\downarrow}$ then $\phi = \kappa_{P^\perp}(\mu_1, \ldots, \mu_n) \parr \kappa_P(\nu_1, \ldots, \nu_n)$ and $\psi^\perp = (\mu_1^\perp \otimes \nu_1^\perp) \parr \cdots \parr (\mu_n^\perp \otimes \nu_n^\perp)$ for some formulas $\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n$ such that $[\![\mu_i]\!] = M_i \neq \emptyset$ and $[\![\nu_i]\!] = N_i \neq \emptyset$ for all $i \in \{1, \ldots, n\}$. We conclude since we have the following derivation

$$\parr\dfrac{\mathsf{d}\text{-}\kappa\dfrac{\otimes\dfrac{\mathsf{AX}\dfrac{}{\vdash \mu_1, \mu_1^\perp} \quad \mathsf{AX}\dfrac{}{\vdash \nu_1, \nu_1^\perp}}{\vdash \mu_1^\perp \otimes \nu_1^\perp, \mu_1, \nu_1} \quad \cdots \quad \otimes\dfrac{\mathsf{AX}\dfrac{}{\vdash \mu_n, \mu_n^\perp} \quad \mathsf{AX}\dfrac{}{\vdash \nu_n, \nu_n^\perp}}{\vdash \mu_n^\perp \otimes \nu_n^\perp, \mu_n, \nu_n}}{\vdash (\mu_1^\perp \otimes \nu_1^\perp), \ldots, (\mu_n^\perp \otimes \nu_n^\perp), \phi}}{(\mu_1^\perp \otimes \nu_1^\perp) \parr \cdots \parr (\mu_n^\perp \otimes \nu_n^\perp), \phi}$$

If $C[\square] = \kappa_P(C'[\square], M_1, \ldots, M_n) \neq \square$, then we assume w.l.o.g., there is a context formula $\zeta[\square] = \kappa_P(\zeta'[\square], \mu_1, \ldots, \mu_n)$ such that $[\![\zeta[\square]]\!] = C[\square]$ and $[\![\zeta'[\square]]\!] = C'[\square]$. We conclude since, by inductive hypothesis on the structure of $C[\square]$, there is a derivation of the following form:

$$\mathsf{d}\text{-}\kappa\dfrac{\dfrac{\Vert\,\mathsf{IH}}{\vdash (\zeta'[\psi'])^\perp, \zeta'[\phi']} \quad \mathsf{AX}\dfrac{}{\vdash \mu_1^\perp, \mu_1} \quad \cdots \quad \mathsf{AX}\dfrac{}{\vdash \mu_n^\perp, \mu_n}}{\vdash \kappa_{P^\perp}\left(\!\left|(\zeta'[\psi'])^\perp, \mu_1^\perp, \ldots, \mu_n^\perp\right|\!\right), \kappa_P\left(\!\left|\zeta'[\phi'], \mu_1, \ldots, \mu_n\right|\!\right)} \quad .$$

$\square$

$$\mathsf{w}\,\frac{\vdash \Gamma}{\vdash \Gamma, \phi} \qquad \mathsf{c}\,\frac{\vdash \Gamma, \phi, \phi}{\vdash \Gamma, \phi} \qquad \Big| \qquad \mathsf{w}{\downarrow}\,\frac{\psi}{\psi \parr \phi} \qquad \mathsf{c}{\downarrow}\,\frac{\phi \parr \phi}{\phi} \qquad \Big| \qquad \mathsf{ac}{\downarrow}\,\frac{a \parr a}{a} \qquad \mathsf{m}\,\frac{P(\!|\phi_1, \ldots, \phi_n|\!) \parr P(\!|\psi_1, \ldots, \psi_n|\!)}{P(\!|\phi_1 \parr \psi_1, \ldots, \phi_n \parr \psi_n|\!)}\ \parr \neq P \text{ prime}$$

Figure 12: Structural rules for sequent calculi, and the corresponding rules in deep inference together with the atomic contraction and the generalized medial rule.

We are now able to prove the main result of this section, that is, establishing a correspondence between graphs provable in GS and graphs which are interpretation via $[\![\cdot]\!]$ of formulas provable in MGL°.

Theorem 54. *Let $G \neq \emptyset$ be a graph and ϕ a pure formula such that $[\![\phi]\!] = G$. Then $\vdash_{\mathsf{GS}} G$ iff $\vdash_{\mathsf{MGL}^\circ} \phi$.*

Proof. By Lemma 52, if $\vdash_{\mathsf{MGL}^\circ} \phi$, then by there is a proof of $[\![\phi]\!]$ in MGL°.

To prove the converse, let $\mathcal{D}$ be a proof of $G \neq \emptyset$ in GS. We define a proof $\pi_{\mathcal{D}}$ of ϕ by induction on the number n of rules in $\mathcal{D}$.

- We cannot have $n = 0$ since we are assuming $G \neq \emptyset$.
- If $n = 1$, then $G = a \parr a^\perp$ and $\pi_{\mathcal{D}} = \parr\,\dfrac{\mathsf{ax}\,\dfrac{}{\vdash a, a^\perp}}{\vdash a \parr a^\perp}$.
- If $n > 1$, then $\mathcal{D} = \mathsf{r}\,\dfrac{\begin{array}{c}\mathcal{D}' \| \\ H\end{array}}{G}$ then by inductive hypothesis we have a proof $\pi_{\mathcal{D}'}$ of a formula ψ such that $[\![\psi]\!] = H$. If $\mathsf{r} \in \{\mathsf{s}_\parr, \mathsf{s}_\otimes, \mathsf{p}{\downarrow}\}$, then by Lemma 53 we have a derivation with cut as the one below on the left of a formula ϕ such that $[\![\phi]\!] = G$. We then conclude by Theorem 39.

$$\mathsf{cut}\,\dfrac{\begin{array}{c}\|\mathsf{IH} \\ \psi\end{array} \qquad \begin{array}{c}\|\text{Lemma 53} \\ \vdash \psi^\perp, \phi\end{array}}{\vdash \phi} \quad \overset{\textit{Theorem } 39}{\rightsquigarrow^*} \quad \begin{array}{c}\|\mathsf{MGL}^\circ \\ \phi\end{array} \qquad \Bigg| \qquad \begin{array}{c} \mathsf{deep}\,\dfrac{\parr\,\dfrac{\mathsf{ax}\,\dfrac{}{\vdash a, a^\perp}}{\vdash a \parr a^\perp} \qquad \begin{array}{c}\pi_{\mathcal{D}'} \|\mathsf{IH} \\ \psi\end{array}}{\vdash \zeta[a \parr a^\perp]} \\ = \cdots\cdots\cdots \\ \vdash \phi \end{array}$$

  Otherwise $\mathsf{r} = \mathsf{ai}{\downarrow}$, then it must have been applied deep inside a context $C[\square] = [\![\zeta[\square]]\!] \neq \square$ such that $C[\emptyset] = H = [\![\psi]\!]$. Therefore $\phi = \zeta[a \parr a^\perp]$. We conclude by applying Lemma 47 to the derivation above on the right. □

5 Classical Logic Beyond Cographs

We conclude this paper by providing an extension of MGL with standard *contraction* and *weakening* structural rules, showing that it provides a conservative extension of propositional classical logic. We then show decomposition results allowing us to factorize any proof into a linear proof (i.e., a proof in *PML*) and a *resource management* proof (i.e., a derivation only using weakening and contraction rules).

Definition 55. We define the following sequent system:

$$\textbf{\textit{Classical Graphical Logic}}: \ \mathsf{LGK} = \mathsf{MGL} \cup \{\mathsf{w}, \mathsf{c}\} \tag{11}$$

For LGK we can prove the admissibility of the cut-rule via cut-elimination.

Theorem 56 (Cut-elimination)**.** *The rule* cut *is admissible in* LGK.

Proof. Consider the *cut-elimination steps* from Figure 6 and Figure 13 and the definition of weight from the proof of Theorem 39. A proof of weak normalization of the cut-elimination procedure can be given using the same measure used in the proof of Theorem 39 [a] by restraining the application of the cut-elimination steps only to top-most cut-rules in the derivation. □

[a]For the sake of determining if a cut-formula is principal, in a contraction rule (c) we assume both occurrences of ϕ in the premise to be active and the occurrence of ϕ in the conclusion to be principal.

$$
\text{cut}\,\dfrac{\text{w}\,\dfrac{\vdash \Gamma}{\vdash \Gamma, \phi} \quad \vdash \phi^{\perp}, \Delta}{\vdash \Gamma, \Delta} \rightsquigarrow \text{w}\,\dfrac{\vdash \Gamma}{\vdash \Gamma, \Delta}
\qquad
\text{cut}\,\dfrac{\text{c}\,\dfrac{\vdash \Gamma, \phi, \phi}{\vdash \Gamma, \phi} \quad \vdash \phi^{\perp}, \Delta}{\vdash \Gamma, \Delta} \rightsquigarrow \text{c}\,\dfrac{\text{cut}\,\dfrac{\text{cut}\,\dfrac{\vdash \Gamma, \phi, \phi \quad \vdash \phi^{\perp}, \Delta}{\vdash \Gamma, \Delta, \phi} \quad \vdash \phi^{\perp}, \Delta}{\vdash \Gamma, \Delta, \Delta}}{\vdash \Gamma, \Delta}
$$

Figure 13: The cut-elimination steps for the structural rules.

We consider the deep inference rules in Figure 12, that is, rules which can be applied on subformulas in a sequent. Using the deep inference version of the structural rules (weakening and contraction) and the ***generalized medial*** rule proposed in [17] we can define a inference system where structural rules can pushed down in a derivation obtaining a decomposition result extending the one in [13, 15] for classical logic.

Lemma 57. *The contraction rule* $\mathsf{c}\downarrow$ *is derivable using atomic contraction (*$\mathsf{ac}\downarrow$*) and medial rule (*m*).*

Proof. By induction on the contracted formula ϕ. If $\phi = a$ is an atom, then $\mathsf{c}\downarrow$ is an instance of $\mathsf{ac}\downarrow$. Otherwise, $\phi = \kappa(\!|\psi_1, \ldots, \psi_n|\!)$ and we conclude since we can apply inductive hypothesis to replace each application of $\mathsf{c}\downarrow$ with a derivation of the following form

$$
\mathsf{c}\downarrow\,\dfrac{\kappa(\!|\psi_1, \ldots, \psi_n|\!) \parr \kappa(\!|\psi_1, \ldots, \psi_n|\!)}{\kappa(\!|\psi_1, \ldots, \psi_n|\!)}
\quad \rightsquigarrow \quad
\mathsf{m}\,\dfrac{\kappa(\!|\psi_1, \ldots, \psi_n|\!) \parr \kappa(\!|\psi_1, \ldots, \psi_n|\!)}{\kappa\left(\!\!\left|\; \begin{array}{c} \psi_1 \parr \psi_1 \\ \mathsf{IH} \Big\| \{\mathsf{m},\mathsf{ac}\downarrow\} \\ \psi_1 \end{array}, \ldots, \begin{array}{c} \psi_n \parr \psi_n \\ \mathsf{IH} \Big\| \{\mathsf{m},\mathsf{ac}\downarrow\} \\ \psi_n \end{array} \;\right|\!\!\right)} \; .
$$

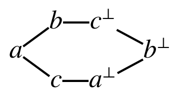
□

Theorem 58 (Decomposition)**.** *Let* Γ *be a sequent. If* $\vdash_{\mathsf{LGK}} \Gamma$*, then:*

1. *there is a sequent* Γ' *such that* $\vdash_{\mathsf{MGL}} \Gamma' \vdash_{\{\mathsf{w}\downarrow,\mathsf{c}\downarrow\}} \Gamma$
2. *there are sequent* Γ'*,* Δ'*, and* Δ *such that* $\vdash_{\mathsf{MGL}} \Gamma' \vdash_{\{\mathsf{m}\}} \Delta' \vdash_{\{\mathsf{ac}\downarrow\}} \Delta \vdash_{\{\mathsf{w}\downarrow\}} \Gamma$

Proof. The proof of Item 1 is immediate by applying rule permutations. For a reference, see [6]. Item 2 is consequence of the previous point since by Lemma 57 we can replace all instances of $\mathsf{c}\downarrow$-rules with derivations containing only m and $\mathsf{ac}\downarrow$, and conclude by applying rule permutations to move all ac-rules below the instances of m-rules, and $\mathsf{w}\downarrow$ to the bottom of a derivation. □

To conclude this section, we recall that classical graphical logic, is not the same logic of the ***boolean graphical logic*** (denoted GBL) defined in [17] (an inference systems on graphs by extending the semantics of read-once boolean relations from cographs to general graphs). In fact, even if both are conservative extensions of classical logic, the following graph from [4] which is expected to be provable in GBL, but is not provable in GS (and there is no formula ϕ provable in LGK such that $[\![\phi]\!]$ is the given graph).



6 Conclusion and Future Works

providing the logical foundations to define

In this paper we have provided foundations for the design of proof systems operating on graph by defining *graphical connectives*, a class of logical operators generalizing the classical conjunction and disjunction, and whose semantics is solely defined by their interpretation as prime graphs.

We studied two sub-structural sequent calculi operating on formulas defined via graphical connectives (MGL and MGL°), proving that cut-elimination holds in these systems and that they are conservative extensions of the multiplicative linear logic and the multiplicative linear logic with mix respectively. For these

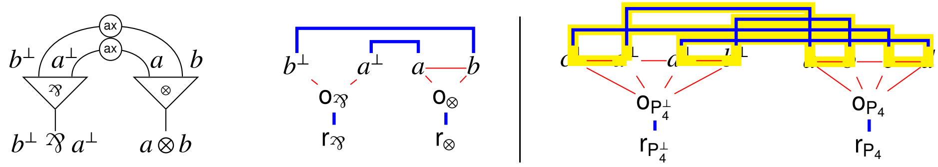


Figure 14: On the left: the same proof net in the original Girard's syntax and Retoré's one. On the right: an RB-proof net of $\kappa_{P_4}(\!|a,b,c,d|\!) \multimap \kappa_{P_4}(\!|a,b,c,d|\!)$ containing the chorded æ-cycle $a \cdot b \cdot b^\perp \cdot d^\perp \cdot d \cdot c \cdot c^\perp \cdot a^\perp$.

calculi, we proved that they capture graph isomorphisms as provable logical equivalences[12]. We were able to prove that the class of graphs representing provable formulas in MGL° coincides with the class of non-empty graphs provable in the proof system GS from [3, 2]. As a consequence, the proofs of cut-elimination in MGL serves as simplified version of the proof of transitivity of implication in GS.

We concluded by providing a conservative extension of both classical propositional logic and MGL, and proving the existence of a decomposition result allowing us to have canonical forms for proofs in which all structural rules can be relegated at the bottom of a derivation.

6.1 Future Works

Categorical Semantics. The systems MGL and MGL° define conservative extensions of MLL and MLL° respectively. We expect to be able to define categorical models by extending (unit-free) *star-autonomous* and *IsoMix* [19, 20] categories respectively with additional (n-ary) monoidal products whose symmetries would be dictated by the symmetry group of the corresponding prime graph.

Digraphs, Games and Event Structures. In this work we started our investigation from the correspondence between classical propositional formulas and cographs. A similar approach could be developped for directed graphs by extending the encoding of intuitionistic propositional formulas used in defining Hyland-Ong *arenas* [40]. We foresee interesting connections with concurrent games and event structures [59]. In particular, graphs generalizing the connectives from *additive linear logic* [18] could allow us to express non-transitive conflict relations.

Proof nets and proof equivalence. We plan designing proof nets [30, 22, 31] for MGL and MGL°. For this purpose, we consider extending the Retoré's *handsome proof net* syntax emplying two-colored graphs (see the left of Figure 14) where the graph induced over the vertices corresponding to the inputs of the $\parr$- and $\otimes$-gates of those proof nets are isomorphic to the corresponding prime graphs. Therefore, we could generalize them to represent P gates for any prime connective (see the right of Figure 14 where a P_4- and a $\mathsf{P}_4^\perp$-gate occur). For these proof structure the standard *acyclicity* condition, usually employed to guarantee correctness, is doomed to fail as shown in the right-hand side of Figure 14, where a correct proof-net of the sequent $\mathsf{P}_4(\!|a,b,c,d|\!) \multimap \mathsf{P}_4(\!|a,b,c,d|\!)$ contains a cylce. We foresee the possibility of using results on the *primeval* decomposition of graphs [42] to isolate those cycles witnessing unsoundness.

Such a result would open to the possibility of defining *combinatorial proofs* [39, 38] for LGK relying on the decomposition result (Theorem 58).

[12]Note that the sequent calculus is only capable of checking if two graphs sharing the same set of vertices are isomorphic (problem with polynomial complexity), but not to find an correspondence between vertices of two graphs which is an isomorphism (a well-known **NP** problem)

References

[1] Matteo Acclavio, Ross Horne, Sjouke Mauw, and Lutz Straßburger. A Graphical Proof Theory of Logical Time. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:25, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[2] Matteo Acclavio, Ross Horne, and Lutz Straßburger. An Analytic Propositional Proof System On Graphs. This is an extended version of a paper published at LICS 2020 [AHS20]., December 2020.

[3] Matteo Acclavio, Ross Horne, and Lutz Straßburger. Logic beyond formulas: A proof system on graphs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 38–52, New York, NY, USA, 2020. Association for Computing Machinery.

[4] Matteo Acclavio, Ross Horne, and Lutz Straßburger. An Analytic Propositional Proof System on Graphs. *Logical Methods in Computer Science*, Volume 18, Issue 4, October 2022.

[5] Matteo Acclavio and Roberto Maieli. Generalized connectives for multiplicative linear logic. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *LIPIcs*, pages 6:1–6:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] Matteo Acclavio and Lutz Straßburger. From syntactic proofs to combinatorial proofs. In Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, editors, *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, volume 10900, pages 481–497. Springer, 2018.

[7] Andrea Aler Tubella and Alessio Guglielmi. Subatomic proof systems: Splittable systems. *ACM Trans. Comput. Logic*, 19(1), January 2018.

[8] Andrea Aler Tubella and Lutz Straßburger. Introduction to Deep Inference. Lecture, August 2019.

[9] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[10] Arnon Avron and Iddo Lev. Canonical propositional Gentzen-type systems. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Automated Reasoning*, pages 529–544, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[11] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 684–697, New York, NY, USA, 2016. Association for Computing Machinery.

[12] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic: graph. Darst*, volume 53. Cambridge University Press, 2001.

[13] Kai Brünnler. Locality for classical logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006.

[14] Kai Brünnler and Lutz Straßburger. Modular sequent systems for modal logic. In Martin Giese and Arild Waaler, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'09*, volume 5607 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2009.

[15] Paola Bruscoli and Lutz Straßburger. On the length of medial-switch-mix derivations. In Juliette Kennedy and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 24th International Workshop, WoLLIC 2017, London, UK, July 18-21, 2017, Proceedings*, volume 10388 of *Lecture Notes in Computer Science*, pages 68–79. Springer, 2017.

[16] Cameron Calk. A graph theoretical extension of boolean logic. Bachelor's thesis, 2016.

[17] Cameron Calk, Anupam Das, and Tim Waring. Beyond formulas-as-cographs: an extension of boolean logic to arbitrary graphs, 2020.

[18] J.R.B. Cockett and C.A. Pastro. A language for multiplicative-additive linear logic. *Electronic Notes in Theoretical Computer Science*, 122:23–65, 2005. Proceedings of the 10th Conference on Category Theory in Computer Science (CTCS 2004).

[19] J.R.B. Cockett and R.A.G. Seely. Proof theory for full intuitionistic linear logic, bilinear logic, and mix categories. *Theory and Applications of Categories*, 3(5):85–131, 1997.

[20] J.R.B. Cockett and R.A.G. Seely. Weakly distributive categories. *J. of Pure and Applied Algebra*, 114:133–173, 1997.

[21] D.G. Corneil, H. Lerchs, and L.Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.

[22] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical logic*, 28(3):181–203, 1989.

[23] Anupam Das. Complexity of evaluation and entailment in boolean graph logic. preprint, 2019.

[24] Anupam Das and Alex A. Rice. New minimal linear inferences in boolean logic independent of switch and medial. In Naoki Kobayashi, editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, pages 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[25] Pierre-Malo Deniélou and Nobuko Yoshida. Buffered communication analysis in distributed multiparty sessions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory*, pages 343–357, Berlin, Heidelberg, 2010. Springer.

[26] Edsger W. Dijkstra. Information streams sharing a finite buffer. *Information Processing Letters*, 1(5):179–180, 1972.

[27] A. Ehrenfeucht, T. Harju, and G Rozenberg. *The Theory of 2-Structures A Framework for Decomposition and Transformation of Graphs*. World Scientific, 1999.

[28] Xiang Fu, Tevfik Bultan, and Jianwen Su. Analysis of interacting BPEL web services. In *Proceedings of the 13th international conference on World Wide Web*, pages 621–630. ACM, 2004.

[29] Tibor Gallai. Transitiv orientierbare Graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1–2):25–66, 1967.

[30] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[31] Jean-Yves Girard. Proof-nets : the parallel syntax for proof-theory. In Aldo Ursini and Paolo Agliano, editors, *Logic and Algebra*. Marcel Dekker, New York, 1996.

[32] Jean-Yves Girard. Light linear logic. *Information and Computation*, 143:175–204, 1998.

[33] Jean-Yves Girard. On the meaning of logical rules II: multiplicatives and additives. *NATO ASI Series F: Computer and Systems Sciences*, 175:183–212, 2000.

[34] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *LIPIcs*, pages 135–150, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[35] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic*, pages 54–68, Berlin, Heidelberg, 2001. Springer.

[36] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 231–246, Berlin, Heidelberg, 2002. Springer.

[37] Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.

[38] Dominic Hughes. Proofs Without Syntax. *Annals of Mathematics*, 164(3):1065–1076, 2006.

[39] Dominic Hughes. Towards Hilbert's 24^{th} problem: Combinatorial proof invariants: (preliminary version). *Electr. Notes Theor. Comput. Sci.*, 165:37–63, 2006.

[40] J. Martin E. Hyland and Chih-Hao Luke Ong. On full abstraction for PCF: I. Models, observables and the full abstraction problem, II. Dialogue games and innocent strategies, III. A fully abstract and universal game model. *Information and Computation*, 163:285–408, 2000.

[41] Lee O James, Ralph G Stanton, and Donald D Cowan. Graph decomposition for undirected graphs. In *Proceedings of the Third Southeastern Conference on Combinatorics, Graph Theory, and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1972)*, pages 281–290, 1972.

[42] Beverly Jamison and Stephan Olariu. P-components and the homogeneous decomposition of graphs. *SIAM Journal on Discrete Mathematics*, 8(3):448–463, 1995.

[43] David S Johnson. The np-completeness column: an ongoing guide. *Journal of Algorithms*, 6(3):434–451, 1985.

[44] Björn Lellmann and Elaine Pimentel. Modularisation of sequent calculi for normal and non-normal modalities. *ACM Trans. Comput. Logic*, 20(2), feb 2019.

[45] T.Y. Lin. Chinese wall security policy-an aggressive model. In *[1989 Proceedings] Fifth Annual Computer Security Applications Conference*, pages 282–289, 1989.

[46] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[47] Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 1971.

[48] Roberto Maieli. Non decomposable connectives of linear logic. *Annals of Pure and Applied Logic*, 170(11):102709, 2019.

[49] Ross M. McConnell and Jeremy P. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 536–545, USA, 1994. Society for Industrial and Applied Mathematics.

[50] Dale Miller and Elaine Pimentel. A formal framework for specifying sequent calculus proof systems. *Theoretical Computer Science*, 474:98–116, 2013.

[51] Dale Miller and Alexis Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of *LNCS*, pages 405–419. Springer-Verlag, 2007.

[52] Lê Thành Dũng Nguyên and Lutz Straßburger. A System of Interaction and Structure III: The Complexity of BV and Pomset Logic. working paper or preprint, 2022.

[53] Lê Thành Dũng Nguyên and Lutz Straßburger. BV and Pomset Logic are not the same. In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[54] R.A.G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. *Contemporary Mathematics*, 92, 1989.

[55] Alwen Fernanto Tiu. A system of interaction and structure II: The need for deep inference. *Logical Methods in Computer Science*, 2(2):1–24, 2006.

[56] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, second edition, 2000.

[57] Jacobo Valdes, Robert E Tarjan, and Eugene L Lawler. The recognition of series parallel digraphs. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 1–12. ACM, 1979.

[58] Timothy Waring. A graph theoretic extension of boolean logic. Master's thesis, 2019.

[59] Glynn Winskel, Silvain Rideau, Pierre Clairambault, and Simon Castellan. Games and strategies as event structures. *Logical Methods in Computer Science*, 13, 2017.

A Deep Inference and the Open Deduction Formalism

Open deduction [34] is a proof formalism based on deep inference [8]. It has originally been defined for formulas, but it is abstract enough such that it can equally well be used for graphs, as already done in [2].

Definition 59. An ***inference system*** S is a set of inference rules (as for example shown in Figure 4). A ***derivation*** $\mathcal{D}$ in S with premise G and conclusion H is denoted $\begin{array}{c} G \\ \mathcal{D}\Vert\mathsf{S} \\ H \end{array}$ and is defined inductively as follows:

- Every graph G is a (***trivial***) derivation with premise G and conclusion G (also denoted G).
- An instance of a rule $\mathsf{r}\frac{G}{H}$ in S is a derivation with premise G and conclusion H.
- If $\mathcal{D}_1$ is a derivation with premise G_1 and conclusion H_1, and $\mathcal{D}_2$ is a derivation with premise G_2 and conclusion H_2, and $H_1 = G_2$, then the composition of $\mathcal{D}_1$ and $\mathcal{D}_2$ is a derivation $\mathcal{D}_2\ ;\ \mathcal{D}_1$ denoted as below.

$$\begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ H_1 \\ \cdots\cdots \\ G_2 \\ \mathcal{D}_2\Vert\mathsf{S} \\ H_2 \end{array} \quad\text{or}\quad \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ H_1 \\ \cdots\cdots \\ G_2 \\ \mathcal{D}_2\Vert\mathsf{S} \\ H_2 \end{array} \quad\text{or}\quad \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ H_1 \\ \cdots\cdots \\ G_2 \\ \mathcal{D}_2\Vert\mathsf{S} \\ H_2 \end{array} \quad\text{or}\quad \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ G_2 \\ \mathcal{D}_2\Vert\mathsf{S} \\ H_2 \end{array} \quad\text{or}\quad \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ H_1 \\ \mathcal{D}_2\Vert\mathsf{S} \\ H_2 \end{array}$$

  Note that even if the symmetry between G_2 and H_1 is not written, we always assume it is part of the derivation and explicitly given.
- If G is a graph with n vertices and $\mathcal{D}_1, \dots, \mathcal{D}_n$ are derivations with premise G_i and conclusion H_i for each $i \in \{1, \dots, n\}$, then $G(\!|\mathcal{D}_1, \dots, \mathcal{D}_n|\!)$ is a derivation with premise $G(\!|G_1, \dots, G_n|\!)$ and conclusion $G(\!|H_1, \dots, H_n|\!)$ denoted as below on the left.

$$G\left(\!\!\left|\ \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert\mathsf{S} \\ H_1 \end{array}, \dots, \begin{array}{c} G_n \\ \mathcal{D}_n\Vert\mathsf{S} \\ H_n \end{array}\ \right|\!\!\right) \qquad\Bigg|\qquad \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert \\ H_1 \end{array} \star \begin{array}{c} G_1 \\ \mathcal{D}_1\Vert \\ H_1 \end{array}$$

  If $G = \star \in \{⅋, \otimes\}$ we may write the derivations as above on the right.

Therefore, $C\left[\begin{array}{c} G \\ \mathcal{D}\Vert\mathsf{S} \\ H \end{array}\right] := \begin{array}{c} C[G] \\ C[\mathcal{D}]\Vert\mathsf{S} \\ C[H] \end{array}$ is well-defined for any context $C[\square]$ and any derivation $\begin{array}{c} G \\ \mathcal{D}\Vert\mathsf{S} \\ H \end{array}$.

A ***proof*** in S is a derivation in S whose premise is $\varnothing$.

A graph G is ***provable*** in S (denoted $\vdash_{\mathsf{S}} G$) iff there is a proof in S with conclusion G.

A.1 Equivalent Definitions of GS

We here show that the formulation of the system GS provided in this paper is equivalent to one provided in [3, 4]. In particular, in the previous these papers the rule $\mathsf{s}_\otimes$ was not included in the system. However, as shown in [4] this rule plays a crucial role in the proof that GS is a conservative extension of MLL° and in [1] it is shown that this rule cannot be admissible in the proof systems operating on mixed graphs. Moreover, we here give a weaker side condition on the p-rule with respect to the rules below:

$\mathsf{p}{\downarrow}$ in [4]	$\mathsf{p}{\downarrow}$ in [3]
$\mathsf{p}_1{\downarrow}\dfrac{(M_1 ⅋ N_1)\otimes\cdots\otimes(M_n ⅋ N_n)}{P^\perp(\!\|M_1,\dots,M_n\|\!) ⅋ P(\!\|N_1,\dots,N_n\|\!)}\star$	$\mathsf{p}_2{\downarrow}\dfrac{(M_1 ⅋ N_1)\otimes\cdots\otimes(M_n ⅋ N_n)}{P^\perp(\!\|M_1,\dots,M_n\|\!) ⅋ P(\!\|N_1,\dots,N_n\|\!)}\dagger$
$\star :=\ P \notin \{⅋,\otimes\}$ prime $M_i \neq \varnothing$ for all $i \in \{1,\dots,n\}$	$\dagger :=\ P \notin \{⅋,\otimes\}$ prime $M_i ⅋ N_i \neq \varnothing$ for all $i \in \{1,\dots,n\}$

(12)

In order to prove the equivalence between our system and the ones in [3, 4] we recall the following lemma allowing us to prove that in GS we can derive any graph of the shape $G \multimap G$.

Lemma 60. *Let $M_1, \ldots, M_n, N_1, \ldots, N_n$ and G be graphs such that $|V_G| = n$. Then there is a derivation*

$$\begin{array}{c} (M_1 \parr N_1) \otimes \cdots \otimes (M_n \parr N_n) \\ \| \{\mathsf{s}_\otimes, \mathsf{p}\downarrow\} \\ G^\perp(\!| M_1, \ldots, M_n |\!) \parr G(\!| N_1, \ldots, N_n |\!) \end{array}$$

Proof. By induction on the modular decomposition of G. □

Thanks to this lemma, we can therefore prove the admissibility of the weaker

Proposition 61. *The following version of* $\mathsf{p}\downarrow$ *with weaker side conditions is admissible in* GS

$$\mathsf{p}_1\downarrow \frac{(M_1 \parr N_1) \otimes \cdots \otimes (M_n \parr N_n)}{P^\perp(\!| M_1, \ldots, M_n |\!) \parr P(\!| N_1, \ldots, N_n |\!)} \ P \text{ prime, } M_i \neq \emptyset \text{ for all } i \in \{1, \ldots, n\}$$

Proof. Note that we may have $N_i = \emptyset$ for some $i \in \{1, \ldots, n\}$. Thus, if $N_i \neq \emptyset$ for all $i \in \{1, \ldots, n\}$, then $\mathsf{p}_1\downarrow$ is an occurrence of $\mathsf{p}\downarrow$. Otherwise, w.l.o.g., $N_1 = \emptyset$, thus we have a derivation

$$\begin{array}{c} M_1 \otimes \left[\begin{array}{c} (M_2 \parr N_2) \otimes \cdots \otimes (M_n \parr N_n) \\ \| \text{Lemma 60} \\ H^\perp(\!| M_2, \ldots, M_n |\!) \parr H(\!| N_2, \ldots, N_n |\!) \end{array} \right] \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathsf{s}_\otimes \dfrac{M_1 \otimes P^\perp(\!| \emptyset, M_2, \ldots, M_n |\!)}{P^\perp(\!| M_1, M_2, \ldots, M_n |\!)} \parr P(\!| \emptyset, N_2, \ldots, N_n |\!) \end{array}$$

□

Theorem 62. *Let G be a graph. Then*

$$\vdash_{\mathsf{GS}} G \Leftrightarrow \vdash_{\{\mathsf{ai}\downarrow, \mathsf{s}_\parr, \mathsf{s}_\otimes, \mathsf{p}_1\downarrow\}} G \Leftrightarrow \vdash_{\{\mathsf{ai}\downarrow, \mathsf{s}_\parr, \mathsf{p}_1\downarrow\}} G \Leftrightarrow \vdash_{\{\mathsf{ai}\downarrow, \mathsf{s}_\parr, \mathsf{p}_2\downarrow\}} G$$

Proof. The first equivalence follows from Proposition 61. The other has been proved in [4]. □

B On Rules Introducing a Connective at a Time

A rule introducing only one connective (different from $\parr$ and $\otimes$) at a time inevitably leads to the same problem observed in the literature of *generalized multiplicative connectives* [22, 33, 48, 5], where *initial coherence* (i.e. the possibility of having only atomic axioms in a cut-free system, [10]) is ruled out because of the so-called *packaging problem*.

However, in this appendix we discuss the results about the system extending multiplicative linear logic with the rule $\mathsf{s}\text{-}\kappa$, that is, the system.

$$\mathsf{MLL}^{\mathsf{s}\text{-}\kappa} := \{\mathsf{ax}, \parr, \otimes, \mathsf{mix}, \mathsf{s}\text{-}\kappa_P \mid P \in \mathcal{P}\} \quad \text{where} \quad \mathsf{s}\text{-}\kappa_P \frac{\vdash \Gamma_1, \phi_1 \qquad \cdots \qquad \vdash \Gamma_n, \phi_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa_P(\!| \phi_1, \ldots, \phi_n |\!)}$$

We first observe that in the system does not satisfy anymore initial coherence; e.g., the formula $\kappa_{\mathsf{P}_4}(\!| a, b, c, d |\!) \multimap \kappa_{\mathsf{P}_4}(\!| a, b, c, d |\!)$ is not provable anymore. However, the system still satisfies cut-elimination. The proof cut-elimination is straightforward by considering the following additional cut-elimination steps.

$$\mathsf{cut} \dfrac{\mathsf{s}\text{-}\kappa \dfrac{\vdash \Gamma_1, \phi_1 \quad \cdots \quad \vdash \Gamma_n, \phi_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa_P(\!| \phi_1, \ldots, \phi_n |\!)} \quad \mathsf{s}\text{-}\kappa \dfrac{\vdash \Delta_1, \phi_1^\perp \quad \cdots \quad \vdash \Delta_n, \phi_n^\perp}{\vdash \Delta_1, \ldots, \Delta_n, \kappa_{P^\perp}(\!| \phi_1^\perp, \ldots, \phi_n^\perp |\!)}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n} \rightsquigarrow \mathsf{mix} \dfrac{\mathsf{cut} \dfrac{\vdash \Gamma_1, \phi_1 \quad \vdash \Delta_1, \phi_1^\perp}{\vdash \Gamma_1, \Delta_1} \quad \cdots \quad \mathsf{cut} \dfrac{\vdash \Gamma_n, \phi_n^\perp \quad \vdash \Delta_n, \phi_n^\perp}{\vdash \Gamma_n, \Delta_n}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n}$$

$$\mathsf{cut} \dfrac{\mathsf{d}\text{-}\kappa \dfrac{\vdash \Gamma_1, \phi_1, \psi_1 \quad \cdots \quad \vdash \Gamma_n, \phi_n, \psi_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa_{P^\perp}(\!| \psi_1, \ldots, \psi_n |\!), \kappa_P(\!| \phi_1, \ldots, \phi_n |\!)} \quad \mathsf{s}\text{-}\kappa \dfrac{\vdash \Delta_1, \phi_1^\perp \quad \cdots \quad \vdash \Delta_n, \phi_n^\perp}{\vdash \Delta_1, \ldots, \Delta_n, \kappa_{P^\perp}(\!| \phi_1^\perp, \ldots, \phi_n^\perp |\!)}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n, \kappa_{P^\perp}(\!| \psi_1, \ldots, \psi_n |\!)} \rightsquigarrow \mathsf{s}\text{-}\kappa \dfrac{\mathsf{cut} \dfrac{\vdash \Gamma_1, \phi_1, \psi_1 \quad \vdash \Delta_1, \phi_1^\perp}{\vdash \Gamma_1, \Delta_1, \psi_1} \quad \cdots \quad \mathsf{cut} \dfrac{\vdash \Gamma_n, \phi_n^\perp, \psi_n \quad \vdash \Delta_n, \phi_n^\perp}{\vdash \Gamma_n, \Delta_n, \psi_n}}{\vdash \Gamma_1, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n, \kappa^\perp(\!| \psi_1, \ldots, \psi_n |\!)}$$

Note that $\mathsf{s}\text{-}\kappa$ is derivable in MGL°.

Lemma 63. *The rule $\mathsf{s}\text{-}\kappa$ is derivable in* MGL°.

Proof. If $\kappa = \parr$, then $\mathsf{s}\text{-}\kappa$ is derivable using $\parr$ and mix. If $\kappa = \otimes$, then $\mathsf{s}\text{-}\kappa = \otimes$. Otherwise, we conclude by induction on the arity of κ since we have a derivation

$$\dfrac{\vdash \Gamma_1, \phi_1 \qquad \dfrac{\dfrac{\mathcal{D}' \;\Vert\; \mathsf{IH}}{\vdash \Gamma_2, \ldots, \Gamma_n, \zeta(\!|\phi_1, \ldots, \phi_n|\!)}}{\vdash \Gamma_2, \ldots, \Gamma_n, \kappa(\!|\circ, \phi_2, \ldots, \phi_n|\!)}\,\mathsf{unitor}_\kappa}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa(\!|\phi_1, \ldots, \phi_n|\!)}\,\mathsf{wd}_\otimes$$

where $\mathcal{D}'$ contains instances of $\mathsf{s}\text{-}\kappa$ introducing connectives whose arities are strictly smaller then the arity of κ. □