Sequent Systems on Undirected Graphs

Matteo Acclavio[1] [0000-0002-0425-2825] ✉

University of Sussex, Brighton, UK, `m.acclavio@sussex.ac.uk`

Abstract. In this paper we explore the design of sequent calculi operating on graphs. For this purpose, we introduce logical connectives allowing us to extend the well-known correspondence between classical propositional formulas and cographs. We define sequent systems operating on formulas containing such connectives, and we prove, using an analyticity argument based on cut-elimination, that our systems provide conservative extensions of multiplicative linear logic (without and with mix) and classical propositional logic. We conclude by showing that one of our systems captures graph isomorphism as logical equivalence and that it is sound and complete for the graphical logic GS.

Keywords: Sequent Calculus · Graph Modular Decomposition · Analyticity.

1 Introduction

In theoretical computer science, *formulas* play a crucial role in describing complex abstract objects. At the syntactical level, the formulas of a logic describe complex structures by means of unary and binary operators, usually thought of as *connectives* and *modalities* respectively. On the other hand, graph-based syntaxes are often favored in formal representation, as they provide an intuitive and canonical description of properties, relations and systems. By means of example, consider the two graphs below:

$$a \longleftarrow b \longrightarrow c \longleftarrow d \qquad \text{or} \qquad a \text{---} b \text{---} c \text{---} d$$

It follows from results in [62,21] that describing any of the above graphs by means of formulas only employing binary connectives would require repeating at least one vertex. As a consequence, formulas describing complex graphs are usually long and convoluted, and specific *encodings* are needed to standardize such formulas.

Since graphs are ubiquitous in theoretical computer science and its applications, a natural question to ask is whether it is possible to define formalisms having graphs, instead of formulas, as first-class terms of the syntax. Such a paradigm shift would allow the design of efficient automated tools, reducing the need to handle the bureaucracy introduced in order to deal with the encoding required to represent graphs. At the same time, a graphical syntax would provide a useful tool for investigations such as the ones in [36] or [27,25], where the authors restrain their framework to sequential-parallel orders, as these can be represented by means of formulas with at most binary connectives.

Two recent lines of work have generalized proof theoretical methodologies to graphs, extending the correspondence between classical propositional formulas and cographs. In these works, systems operating on graphs are defined via local and context-free rewriting rules, similar to the approach in *deep inference* systems [33,34,8]. The first line of research, carried out by Calk, Das, Rice and Waring in various works, explores the use of maximal stable sets/cliques-preserving homomorphisms to define notions of entailment[1], and study the resulting proof theory [17,16,63,23,24]. Here, The use of a deep inference formalism is natural, since the rules of the calculus are local rewritings. The second line of research, investigated by the author, Horne, Mauw and Straßburger in several contributions [4,5,3], studies the (sub-)structural proof theory of arbitrary graphs, with an approach inspired by linear logic [29] and deep inference [33]. The main goal of this line of research, partially achieved with the system $\mathsf{GV}^{\mathsf{sl}}$ operating on mixed graphs [3], is to obtain a generalization of the completeness result of the logic BV with respect to pomset inclusion. The logic BV contains a non-commutative binary connective $\triangleleft$ allowing to represent series-parallel partial order multisets as formulas in the syntax (as in Retoré's Pomset logic [57]), and to capture order inclusion as logical implication. However, as shown in [60], no cut-free sequent system for BV can exist – therefore neither for Pomset logic, which strictly contains it [54,53]. For this reason, the aforementioned line of work focused on deep inference systems, and the question about the existence of a cut-free sequent calculus for GS (the restriction of $\mathsf{GV}^{\mathsf{sl}}$ on undirected graphs originally defined in [4]) was left open.

In this paper, we focus on the definition of sequent calculi for *graphical logics*, and we positively answer the above question by providing, among other results, a cut-free sound and complete sequent calculus for GS. By using standard techniques in sequent calculus, we thus obtain a proof of analyticity for this logic which is simpler and more concise with respect to the one in [5].

To achieve these results, we introduce *graphical connectives*, which are operators that can be naturally interpreted as graphs. We then define the sequent calculi MGL, MGL° and KGL, containing rules to handle these connectives. After showing that cut-elimination holds for these systems, we prove that MGL, MGL° and KGL define conservative extensions of *multiplicative linear logic*, *multiplicative linear logic with mix* and *classical propositional logic* respectively. We then prove that formulas interpreted as the same graph are logically equivalent, thus justifying the fact that we consider these systems as operating on graphs rather than formulas. We conclude by showing that MGL° is sound and complete with respect to the logic GS, thus providing a simple sequent calculus for the logic.

The paper is structured as follows. In Section 2 we show how to use the notion of *modular decomposition* for graphs from [28,41] to define graphical connectives. In this way, we extend to general graphs the well-known correspondence between classical propositional formulas and *cographs* [28,41,21]. Then, in Section 3, we introduce the proof systems MGL, MGL° and KGL, and we prove their cut-elimination and analyticity. This section also discusses the conservativity results. In Section 4 we show that formulas representing isomorphic graphs are logically equivalent in these logics. Finally, in Section 5 we prove that MGL° is sound and complete with respect to the graphical logic GS. We conclude with Section 6, by discussing future research directions and applications. Due to space limitations, details of certain proofs have been omitted from this manuscript However, detailed proofs can be found in [2].

[1] A similar approach was proposed in [56] for studying pomsets.

2 From Graphs to Formulas

In this section we first recall standard results from the literature on graphs, the notion of *modular decomposition* and the one of *cographs*, which are graphs whose modular decomposition only contains two prime graphs which can be naturally interpreted as (binary) conjunction and disjunction. We then introduce the notion of *graphical connectives*, allowing us to extend the correspondence between cographs and propositional formulas to general graphs, allowing us to represent graphs via formulas constructed using graphical connectives.

2.1 Graphs and Modules

In this work are interested in using *(labeled) graphs* to represent patterns of interactions by means of the binary relations (edges) between their components (vertices). We recall the standard notion of identity on labeled graphs (i.e., *isomorphism*) and define the rougher notion of *similarity* (isomorphism up-to vertex labels).

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* $\emptyset$ *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* $\langle \emptyset, \emptyset, \emptyset \rangle$ *is denoted* $\emptyset$ *and we define the edge relation* $\overset{G}{\not\frown} := \left\{ (v, w) \mid v \neq w \text{ and } vw \notin \overset{G}{\frown} \right\}$.

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* $x, y \in V_G$*. A* ***symmetry*** *is a similarity of a graph with itself. An* ***isomorphism*** *is a similarity* f *such that* $\ell(v) = \ell(f(v))$ *for any* $v \in V_G$*. Two graphs* G *and* G' *are* ***similar*** *(denoted* $G \sim G'$*) if there is a similarity between* G *and* G'*. They are* ***isomorphic*** *(denoted* $G = G'$*) if there is an 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* $\overset{G}{\not\frown} = \emptyset$ *(resp.* $\overset{G}{\frown} = \emptyset$*).*

Note 1. 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 by using its label instead of its name. For example, the single-vertex graph $G = \langle \{v\}, \ell_G, \emptyset \rangle$ may be represented either by the vertex (name) v or by the vertex label $\ell_G(v)$ (in this case we may write $\bullet$ if $\ell_G(v)$ is not defined).

Example 1. Consider the following graphs:

$$\begin{array}{l} F = \langle \{u_1, u_2, u_3, u_4\}, \{\ell(u_1) = a, \ell(u_2) = b, \ell(u_3) = c, \ell(u_4) = d\}, \{u_1u_2, u_2u_3, u_3u_4\} \rangle \\ G = \langle \{v_1, v_2, v_3, v_4\}, \{\ell(v_1) = b, \ell(v_2) = a, \ell(v_3) = c, \ell(v_4) = d\}, \{v_1v_2, v_1v_3, v_3v_4\} \rangle \\ H = \langle \{w_1, w_2, w_3, w_4\}, \{\ell(w_1) = a, \ell(w_2) = b, \ell(w_3) = c, \ell(w_4) = d\}, \{w_1w_2, w_1w_3, w_3w_4\} \rangle \end{array} \tag{1}$$

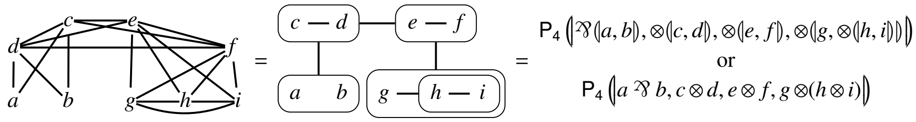


Fig. 1. A graph and one of its modular and the corresponding formula-like representations.

We have $F \sim G \sim H$ and $G = F = a—b—c—d \neq b—a—c—d = H$.

Note 2. Whenever we say that two graphs are the same, we assume they share the same set of vertices and labeling function, therefore implicitly assuming the isomorphism f to be given. This allows us to verify whether two graphs are isomorphic (i.e., the same) in polynomial time on the number of vertices.

We recall the notion of *module* [28,41,35,45,48,26], allowing us to represent a graph using a tree-like syntax. A module is a subset of vertices of a graph having the same edge-relation with any vertex outside the subset, generalizing what can usually be observed in formulas, where, in the formula tree, each literal in a subformula has the same least common ancestor with a given literal not belonging to the subformula itself.

Definition 2. *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 1. A connected component of a graph G is a module of G.

Note 3. We may optimize graph representations by bordering vertices of a same module by a closed line. An edge connected to such a closed line denotes the existence of an edge to each vertex inside it (see Figure 1). By means of example, consider the following graph and its more compact modular representation.

(2)

The notion of module is related to a notion of context, which can be intuitively formulated as a graph with a "hole".

Definition 3. *A* ***context*** *$C[\square]$ is a (non-empty) graph containing a single occurrence of a special vertex $\square$ (with $\ell(\square)$ 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,*

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

Remark 2. The notion of context and the one of module are interdefinable. In fact, a set of vertices M is a module of a graph G iff there is a context $C[\square]$ such that $G = C[M]$.

Note that 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 4. *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,*

$$G(\!|H_1, \ldots, H_n|\!) = \left\langle \biguplus_{i=1}^{n} V_{H_i} \,,\, \bigcup_{i=1}^{n} \ell_{H_i} \,,\, \left(\bigcup_{i=1}^{n} \overset{H_i}{\frown} \right) \cup \left\{ (x, y) \middle| x \in V_{H_i}, y \in V_{H_j}, v_i \overset{G}{\frown} v_j \right\} \right\rangle \tag{3}$$

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 3. The operation of composition-via G forgets the information carried by the labeling function ℓ_G. Moreover, if σ is a similitude between two graphs G and G', then $G(\!|H_1, \ldots, H_n|\!) = G'(\!|H_{\sigma(1)}, \ldots, H_{\sigma(n)}|\!)$.

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 graph.

Definition 5. *Let $G = \langle V_G, \ell_G, E_G \rangle$ be a graph. 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 $a \in \mathcal{A}$).*

2.2 Classical Propositional Formulas as 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:

$$\phi, \psi := a \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi^\perp \qquad \text{with } a \in \mathcal{A}. \tag{4}$$

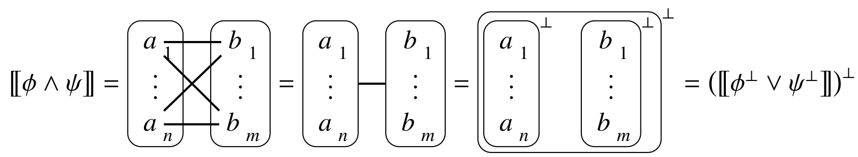
We define a map from literals to single-vertex graphs, which extends to formulas via the composition-via the unlabeled two-vertices stable set and two-vertices clique.

Definition 6. *Let ϕ be a classical formula, and let $\mathsf{S}_2 = \langle \{v_1, v_2\}, \emptyset, \emptyset \rangle$ and $\mathsf{K}_2 = \langle \{v_1, v_2\}, \emptyset, \{v_1 v_2\} \rangle$. We define the graph $[\![\phi]\!]$ 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 we denote by a the single-vertex graph, whose vertex is labeled by a. A ***cograph*** *is a graph G such that there is a classical formula ϕ such that $G = [\![\phi]\!]$.*

Example 2. Let ϕ and ψ classical formulas containing occurrences of atoms $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots b_m\}$ respectively. Then the graph $[\![\phi \wedge \psi]\!]$ can be represented as follows:



Note that an equivalent definition of cographs can be given using only the graph S_2 (or K_2) and duality.

We can easily observe that the map $[\![\cdot]\!]$ well-behaves with respect to the equivalence over formulas generated by the associativity and commutativity of connectives and the de Morgan laws below.

$$\begin{array}{rl} \textbf{\textit{Equivalence laws}} & \begin{cases} \phi \vee \psi \equiv \psi \vee \phi & \qquad \phi \vee (\psi \vee \chi) \equiv (\phi \vee \psi) \vee \chi \\ \phi \wedge \psi \equiv \psi \wedge \phi & \qquad \phi \wedge (\psi \wedge \chi) \equiv (\phi \wedge \psi) \wedge \chi \end{cases} \\ \textbf{\textit{De-Morgan laws}} & \left\{ (\phi^\perp)^\perp \equiv \phi \qquad\qquad (\phi \wedge \psi)^\perp \equiv \phi^\perp \vee \psi^\perp \right. \end{array} \tag{5}$$

Proposition 1. *Let ϕ and ψ be classical formulas. Then $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$.*

We finally recall an alternative definition of cographs as graphs containing no induced subgraph of a specific shape, and we recall the theorem establishing the relation between

Definition 7. *A graph G is* ***$\mathsf{P_4}$-free*** *if there it contains no four vertices v_1, v_2, v_3, v_4 such that the induced subgraph $G|_{\{v_1,v_2,v_3,v_4\}}$ is similar to the graph a—b—c—d .*

Theorem 1 ([28]). *Let G be a graph. Then G is a cograph iff G is P_4-free.*

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., [28,41,35,45,48,26]).

Definition 8. *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 2 ([41]). *Let G be a graph with at least two vertices. Then there are non-empty modules $M_1, \ldots, M_n$ of G and a prime graph P such that $G = P(\!|M_1, \ldots, M_n|\!)$.*

This result allows us to describe graphs using its *modular decomposition*, that is, using single-vertex graphs and operations of composition-via prime graphs only.

Definition 9. *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 $G = a$;*

– *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 $G = P(\!|H_1, \ldots, H_n|\!)$.*

Ambiguity arises in modular decomposition due to the presence of cliques or stable sets with more than three vertices, graph symmetries, and the presence of symmetric but non-isomorphic graphs. The first two ambiguities are akin to the one observed in propositional logic, where conjunction and disjunction are considered associative and commutative. These are addressed similarly in the framework we discuss in this paper. However, to reduce the latter source of ambiguity, we introduce the notion of *basis of graphical connectives*.

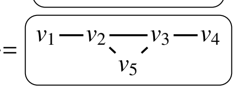
Definition 10. *A **graphical connective** $C = \langle V_C, \overset{C}{\frown} \rangle$ (with **arity** $n = |V_C|$) is given by a finite list of vertices $V_C = \langle v_1, \ldots, v_n \rangle$ and a non-reflexive symmetric edge relation $\overset{C}{\frown}$ over the set of vertices occurring in V_C. We denote by G_C the graph corresponding to C, that is, the graph $G_C = \langle \{v \mid v \text{ in } V_C\}, \varnothing, \overset{C}{\frown} \rangle$. 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 **basis** 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 **group**[2] **of symmetries of C** ($\mathfrak{S}(C)$) and the **set of dualizing symmetries of C** ($\mathfrak{S}^{\perp}(C)$) as the following sets of permutations over the set $\{1, \ldots, n\}$:*

$$\left.\begin{array}{l} \mathfrak{S}(C) := \{\, \sigma \mid C(\!|H_1, \ldots, H_n|\!) = C(\!|H_{\sigma(1)}, \ldots, H_{\sigma(n)}|\!)\} \\ \mathfrak{S}^{\perp}(C) := \left\{\, \sigma \mid (C(\!|H_1, \ldots, H_n|\!))^{\perp} = C(\!|H^{\perp}_{\sigma(1)}, \ldots, H^{\perp}_{\sigma(n)}|\!)\right\} \end{array}\right\} \text{(for any } H_1, \ldots, H_n\text{).} \tag{6}$$

We introduce the following graphical connectives:

$$\begin{array}{l} ⅋(\!|v_1, v_2|\!) := \langle\langle v_1, v_2\rangle, \varnothing\rangle = \boxed{v_1 \quad v_2} \qquad \otimes(\!|v_1, v_2|\!) := \langle\langle v_1, v_2\rangle, \{v_1 v_2\}\rangle = \boxed{v_1 — v_2} \\ \mathsf{P_n}(\!|v_1, \ldots, v_n|\!) := \langle\langle v_1, \ldots, v_n\rangle, \{v_i v_{i+1} \mid i \in \{1, \ldots, n-1\}\}\rangle \quad = \boxed{v_1 — v_2 — \cdots — v_n} \\ \mathsf{Bull}(\!|v_1, \ldots, v_5|\!) := \langle\langle v_1, \ldots, v_5\rangle, \{(v_1 v_2, v_2 v_3, v_3 v_4, v_5 v_2, v_5 v_3)\}\rangle = \end{array} \tag{7}$$



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

Theorem 3. *Let G be a non-empty graph and $\mathcal{P}$ a basis. Then there is a unique way (up to symmetries of graphical connectives and associativity of ⅋ and $\otimes$) to write G using single-vertex graphs and the graphical connectives in $\mathcal{P}$.*

Corollary 1. *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 basis $\mathcal{P}$. From now on, we assume to be fixed a basis $\mathcal{P}$ containing the graphical connectives in Equation (7).

[2] It can be easily shown that $\mathfrak{S}_n$ contains the identity permutation (denoted **id**) and is a subgroup of the group of permutations over the set $\{1, \ldots, n\}$.

Definition 11. *The set of **formulas** is generated by the set of propositional atoms $\mathcal{A}$, a **unit** $\circ$, and a basis of graphical connective $\mathcal{P}$ using the following syntax:*

$$\phi_1, \ldots, \phi_n \coloneqq \circ \mid a \mid a^\perp \mid \kappa_P(\!|\phi_1, \ldots, \phi_{n_P}|\!) \qquad \text{with } a \in \mathcal{A} \text{ and } P \in \mathcal{P} \tag{8}$$

*We simply denote $⅋$ (resp. $\otimes$) the binary connective $\kappa_{⅋}$ (resp. $\kappa_\otimes$) and we write $\phi ⅋ \psi$ instead of $\kappa_{⅋}(\!|\phi, \psi|\!)$ (resp. $\phi \otimes \psi$ instead of $\kappa_\otimes(\!|\phi, \psi|\!)$). 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 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 connectives $⅋$ and $\otimes$. 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 $\zeta[\square] = \psi ⅋ (\square \otimes \chi)$, then $\zeta[\phi] = \psi ⅋ (\phi \otimes \chi)$.*

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

$$[\![\square]\!] = \square \quad [\![\circ]\!] = \emptyset \quad [\![a]\!] = a \quad [\![a^\perp]\!] = a^\perp \quad [\![\kappa_P(\!|\phi_1, \ldots, \phi_n|\!)]\!] = P\left(\!\left|[\![\phi_1]\!], \ldots, [\![\phi_n]\!]\right|\!\right) \tag{9}$$

Note 4. We may consider a formula ϕ over the set of occurrences of literals $\{x_1, \ldots, x_n\}$ as a ***synthetic connective*** ϕ with arity n. That is, we may denote by $\phi(\!|\psi_1, \ldots, \psi_n|\!)$ the formula obtained by replacing each literal x_i (with $i \in \{1, \ldots, n\}$) with a formula ψ_i. The set of ***symmetries*** of ϕ (denoted $\mathfrak{S}(\phi)$) is the set of permutations σ over $\{1, \ldots, n\}$ such that $[\![\phi(\!|x_1, \ldots, x_n|\!)]\!] = [\![\phi(\!|x_{\sigma(1)}, \ldots, x_{\sigma(n)}|\!)]\!]$.

Definition 12. *The equivalence relation $\equiv$ over formulas is generated by the following:*

$$\textbf{\textit{Equivalence laws}} \begin{cases} \kappa_P(\!|\phi_1, \ldots, \phi_{n_P}|\!) \equiv \kappa_P(\!|\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n_P)}|\!) \\ \phi \otimes (\psi \otimes \chi) \equiv (\phi \otimes \psi) \otimes \chi \\ \phi ⅋ (\psi ⅋ \chi) \equiv (\phi ⅋ \psi) ⅋ \chi \end{cases}$$

$$\textbf{\textit{De-Morgan laws}} \begin{cases} \circ^\perp \equiv \circ \qquad \phi^{\perp\perp} \equiv \phi \\ \text{only if } \mathfrak{S}^\perp(P) = \emptyset : \quad (\kappa_P(\!|\phi_1, \ldots \phi_{n_P}|\!))^\perp \equiv \kappa_{P^\perp}(\!|\phi^\perp_{\sigma(1)}, \ldots, \phi^\perp_{\sigma(n_P)}|\!) \\ \text{only if } \mathfrak{S}^\perp(P) \neq \emptyset : \quad (\kappa_P(\!|\phi_1, \ldots \phi_{n_P}|\!))^\perp \equiv \kappa_P(\!|\phi^\perp_{\rho(1)}, \ldots, \phi^\perp_{\rho(n_P)}|\!) \end{cases}$$

for each $P \in \mathcal{P}$ (with arity $n_P = |V_P|$), and for each $\sigma \in \mathfrak{S}(P)$ and $\rho \in \mathfrak{S}^\perp(P)$.

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

$$\circ^\perp = \circ \qquad \text{and} \qquad \phi^{\perp\perp} = \phi \qquad \text{and} \qquad (\kappa_P(\!|\phi_1, \ldots, \phi_{n_P}|\!))^\perp = \kappa_Q(\!|\phi^\perp_{\sigma(1)}, \ldots, \phi^\perp_{\sigma(n_P)}|\!)$$

where Q is the (unique) prime connective in $\mathcal{P}$ such that we have $[\![\kappa_P(\!|a_1, \ldots, a_n|\!)]\!] = Q(\!|a^\perp_{\sigma(1)}, \ldots, a^\perp_{\sigma(n)}|\!)$ for a permutation σ over the set $\{1, \ldots, n\}$. [3]

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

[3] Note that the permutation σ may be not unique. If we consider formulas up-to the equivalence relation $\equiv$, this is irrelevant. Otherwise, in the definition of the linear negation we should also provide a specific permutation σ_P for each prime connective $P \in \mathcal{P}$.

Remark 4. As explained in [5] (Section 9), the graphical connectives we discuss in this paper are *multiplicative connectives* (in the sense of [22,32,47,6]) but they are not the same as the *connectives-as-partitions* discussed in these works. In fact, there is a unique 4-ary graphical connective P_4, which has the symmetry group $\{\mathsf{id}, (1,4)(2,3)\}$, while, as shown in [47,6], there is a unique pair of dual *non-decomposable* (i.e., which cannot be described using smaller connectives) 4-ary multiplicative connectives-as-partitions G_4 and $\mathsf{G}_4^\perp$, and $\mathfrak{S}(\mathsf{P}_4) \subsetneq \mathfrak{S}(\mathsf{G}_4) = \mathfrak{S}(\mathsf{G}_4^\perp)$.

The following result is a consequence of Theorem 2.

Proposition 2. *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 $[\![\phi]\!] = [\![\psi]\!]$.*

For an example of why the equivalence result does not hold in the presence of units, consider the (non-equivalent) formulas $\circ \otimes \circ$ and $\circ \parr \circ$.

3 Sequent calculi over graphs-as-formulas

We assume the reader to be familiar with the definition of sequent calculus derivations as trees of sequents (see, e.g., [61]) but we recall here some definitions.

Definition 13. *A **sequent** is a set of occurrences of formulas. A **sequent system** S is a set of **sequent rules** as the ones in Figure 2. A **derivation** (resp. **open derivation**) over S is a tree of sequents such that each node (resp. each node except some leaves, called **open premises**) is the conclusion of a rule with premises its children. In a sequent rule r, we say that a formula is **active** (resp. **principal**) if it occurs in one of its premises (resp. in its conclusion) but not in its conclusion (resp. but in none of its premises) A **proof** of a sequent Γ is a derivation with root Γ denoted $\overset{\pi \| \mathsf{S}}{\Gamma}$. We denote by $\overset{\Gamma'}{\underset{\Gamma}{\pi' \| \mathsf{S}}}$ an **open derivation** with conclusion Γ and a single 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.*

Definition 14. *We define the following sequent systems using the rules **axiom** (ax), **par** ($\parr$), **tensor** ($\otimes$), **weakening** (w), **contraction** (c), **mix** (mix), **dual connectives** (d-κ) **unitor** (u_κ), and **weak-distributivity** ($\mathsf{wd}_\otimes$) in Figure 2.*

$$\begin{array}{lll} \textbf{\textit{Multiplicative Graphical Logic}}: & \mathsf{MGL} & = \{\mathsf{ax}, \parr, \otimes, \mathsf{d}\text{-}P \mid P \in \mathcal{P}\} \\ \textbf{\textit{Multiplicative Graphical Logic with mix}}: & \mathsf{MGL}^\circ & = \mathsf{MGL} \cup \{\mathsf{mix}, \mathsf{wd}_\otimes, \mathsf{u}_\kappa\} \\ \textbf{\textit{Classical Graphical Logic}} & \mathsf{KGL} & = \mathsf{MGL} \cup \{\mathsf{w}, \mathsf{c}\} \end{array} \quad (10)$$

Remark 5. 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 rule d-κ handles a pair of dual connectives at the same time, as it may be done by rules in focused proof systems (see, e.g.[9,51,50]) or rules for modalities

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

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

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

$$\text{u}_\kappa\frac{\vdash \Gamma, \chi(\!|\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)}|\!)}{\vdash \Gamma, \kappa(\!|\phi_1, \ldots, \phi_k, \circ, \phi_{k+1}, \ldots, \phi_n|\!)} \begin{cases}\sigma \in \mathfrak{S}(\chi) \\ [\![\kappa(\!|\phi_1, \ldots, \phi_k, \circ, \phi_{k+1}, \ldots, \phi_n|\!)]\!] = [\![\chi(\!|\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)}|\!)]\!] \neq \emptyset\end{cases}$$

Fig. 2. Sequent rules.

in modal logic and linear logic (see, e.g., [31,12,14,44]). Intuitively, while in standard two-sided sequent calculi the right-conjunction rule ($\wedge_R$ below) internalizes a meta-conjunction between the premises of the rule, that is,

$$\wedge_R\frac{\boxed{\Gamma_1, \phi_1 \vdash \psi_1, \Delta_1} \quad \text{“and”} \quad \boxed{\Gamma_2, \phi_2 \vdash \psi_2, \Delta_2}}{\Gamma_1, \Gamma_2, \phi_1, \phi_2 \vdash \psi_1 \wedge \psi_2, \Delta_1, \Delta_2} \tag{11}$$

the rule d-κ internalizes a meta-κ-connective between the premises by introducing the same connective on both sides of the sequent, as shown below in the case $\kappa = \mathsf{P}_4$.

$$\frac{\mathsf{P}_4\left(\!\left|\boxed{\Gamma_1, \phi_1 \vdash \psi_1, \Delta_1}, \boxed{\Gamma_2, \phi_2 \vdash \psi_2, \Delta_2}, \boxed{\Gamma_3, \phi_3 \vdash \psi_3, \Delta_3}, \boxed{\Gamma_4, \phi_4 \vdash \psi_4, \Delta_4}\right|\!\right)}{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \kappa_{\mathsf{P}_4}(\!|\phi_1, \phi_2, \phi_3, \phi_4|\!) \vdash \kappa_{\mathsf{P}_4}(\!|\psi_1, \psi_2, \psi_3, \psi_4|\!), \Delta_1, \Delta_2, \Delta_3, \Delta_4} \tag{12}$$

Note that in the rule $\wedge_R$ in Equation (11) only a single occurrence of the connective $\wedge$ occurs in the conclusion, on the right-hand side of $\vdash$. This because the absence of the conjunction $\wedge$ on the left-hand side is irrelevant since a two-sided sequent $\Gamma \vdash \Delta$ is interpreted as the formula $\left(\bigwedge_{\phi \in \Gamma} \phi^\perp\right) \vee \left(\bigvee_{\psi \in \Delta} \psi\right)$.

The names of the rules *unitor* (u_κ) and *weak-distributivity* ($\mathsf{wd}_\otimes$) are inspired by the literature of *monoidal categories* [46] and *weakly distributive categories* [59,20,19]. The rule u_κ internalizes 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 $\mathsf{wd}_\otimes$ generalizes the *weak-distributive law* of the $\otimes$ over the ⅋, that is,

$$\phi \otimes (\psi ⅋ \chi) \longrightarrow (\phi \otimes \psi) ⅋ \chi \tag{13}$$

to the weak-distributive law of $\otimes$ over any connective (see below on the top)

$$\begin{array}{rcl} \chi \otimes \kappa(\!|\phi_1, \ldots, \phi_k, \psi, \phi_{k+1}, \ldots, \phi_n|\!) & \longrightarrow & \kappa(\!|\phi_1, \ldots, \phi_k, \psi \otimes \chi, \phi_{k+1}, \ldots, \phi_n|\!) \\ \kappa(\!|\phi_1, \ldots, \phi_k, \psi ⅋ \chi, \phi_{k+1}, \ldots, \phi_n|\!) & \longrightarrow & \kappa(\!|\phi_1, \ldots, \phi_k, \psi, \phi_{k+1}, \ldots, \phi_n|\!) ⅋ \chi \end{array} \tag{14}$$

Note that an additional law is required to formalize the weak-distributive law of all connectives over ⅋ (see the bottom of Equation (14)). This law corresponds to the rule $\mathsf{wd}_⅋$ in Figure 3.

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

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

Fig. 3. Admissible rules in MGL°.

3.1 Properties of the sequent systems

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

Lemma 1. *Let χ be a pure formula. Then rule $\mathsf{d}\text{-}\chi$ is derivable.*

Corollary 2. *The rule* AX *is derivable in* MGL *and in* MGL°.

Theorem 4. MGL, MGL°*, and* KGL *are initial coherent w.r.t. pure formulas.*

The admissibility of cut is proven via *cut-elimination*.

Theorem 5. *Let* $\mathsf{X} \in \{\mathsf{MGL}, \mathsf{MGL}^{\circ}, \mathsf{KGL}\}$. *The rule* cut *is admissible in* X.

Proof. We define the *size* of a formula as the 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. In Figure 4 we only provide the less standard cut-elimination steps: the ones for ax, w, c, and $\otimes$-*vs*-$\parr$ are the standard ones, while $\mathsf{d}\text{-}\kappa$-*vs*-$\mathsf{d}\text{-}\kappa$ and u_{κ}-*vs*-u_{κ} (where both u_{κ} rules introduce a $\circ$ in the same "position") are as expected, that is, by cutting each of the corresponding premises of the rules. The result for MGL and MGL° follows by the fact that each *cut-elimination step* applied to any cut-rule reduces the size of a derivation, while for KGL we have to consider also weak-normalization result via a cut-elimination strategy prioritizing the elimination of top-most cut-rules.

Note that to ensure that both active formulas of a cut-rule are principal with respect to the rule immediately above it, we also need to consider among the standard *commutative* cut-elimination steps (independent rule permutations) and the special step in Figure 5. The treatment of these steps, as well as the definition of a size taking into account them, is not covered in detail here because it is standard in the literature.

Corollary 3. *Let* $\mathsf{X} \in \{\mathsf{MGL}, \mathsf{MGL}^{\circ}, \mathsf{KGL}\}$. *If* $\vdash_{\mathsf{X}} \phi \multimap \psi$ *and* $\vdash_{\mathsf{X}} \psi \multimap \chi$*, then* $\vdash_{\mathsf{X}} \phi \multimap \chi$.

The admissibility of the cut-rule implies analyticity of MGL and KGL via the standard *sub-formula property*, that is, all formulas occurring in a premise of a rule are subformulas of the ones in the conclusion. However, as already observed in [4,5,3], the same result does not hold for MGL° because the rule u_{κ} and more-than-binary connectives introduce 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.

$$
\dfrac{\mathsf{wd}_\otimes\dfrac{\vdash \Gamma_1,\phi_1 \quad \vdash \Gamma_2,\kappa_P\llparenthesis\circ,\phi_2,\ldots,\phi_n\rrparenthesis}{\vdash \Gamma_1,\Gamma_2,\kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis} \quad \mathsf{wd}_\otimes\dfrac{\vdash \Delta_1,\phi_1^\perp \quad \vdash \Delta_2,\kappa_{P^\perp}\llparenthesis\circ,\phi_2^\perp,\ldots,\phi_n^\perp\rrparenthesis}{\vdash \Delta,\kappa_{P^\perp}\llparenthesis\phi_1^\perp,\ldots,\phi_n^\perp\rrparenthesis}}{\vdash \Gamma_1,\Gamma_2,\Delta_1,\Delta_2}\,\mathsf{cut}
$$

$$\rightsquigarrow$$

$$
\mathsf{mix}\dfrac{\mathsf{cut}\dfrac{\vdash \Gamma_1,\phi_1 \quad \vdash \Delta_1,\phi_1^\perp}{\vdash \Gamma_1,\Delta_1} \quad \mathsf{cut}\dfrac{\vdash,\Gamma_2,\kappa_P\llparenthesis\circ,\phi_2,\ldots\phi_n\rrparenthesis \quad \vdash \Delta_2,\kappa_{P^\perp}\llparenthesis\circ,\phi_2^\perp,\ldots\phi_n^\perp\rrparenthesis}{\vdash \Gamma_2,\Delta_2}}{\vdash \Gamma_1,\Gamma_2,\Delta_1,\Delta_2}
$$

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

$$\rightsquigarrow$$

$$
\mathsf{wd}_\otimes\dfrac{\mathsf{cut}\dfrac{\vdash \Gamma_1,\phi_1,\psi_1 \quad \vdash \Delta,\psi_1^\perp}{\vdash \Gamma_1,\Delta,\phi_1} \quad \mathsf{cut}\dfrac{2{\times}\mathsf{u}_\kappa\dfrac{\mathsf{d}\text{-}\chi\dfrac{\vdash \Gamma_2,\phi_2,\psi_2 \quad \cdots \quad \vdash \Gamma_n,\phi_n,\psi_n}{\vdash \Gamma_2,\ldots,\Gamma_n,\kappa_\chi\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis,\kappa_\chi^\perp\llparenthesis\psi_1,\ldots,\psi_n\rrparenthesis}}{\vdash \Gamma_2,\ldots,\Gamma_n,\kappa_P\llparenthesis\circ,\phi_1,\ldots,\phi_n\rrparenthesis,\kappa_{P^\perp}\llparenthesis\circ,\psi_1,\ldots,\psi_n\rrparenthesis} \quad \vdash \Sigma,\kappa_P\llparenthesis\circ,\psi_2^\perp,\ldots,\psi_n^\perp\rrparenthesis}{\vdash \Gamma_2,\ldots,\Gamma_n,\Sigma,\kappa_P\llparenthesis\circ,\phi_2,\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma_1,\ldots,\Gamma_n,\Delta,\Sigma,\kappa_P\llparenthesis\phi_1,\ldots,\phi_n\rrparenthesis}
$$

Fig. 4. The cut-elimination steps for the structural rules.

$$
\mathsf{u}_\kappa\dfrac{\mathsf{u}_\kappa\dfrac{\vdash \Gamma,\chi\llparenthesis\phi_1,\ldots,\phi_{i-1},\phi_{i+1},\ldots,\phi_{j-1},\phi_{j+1},\ldots,\phi_n\rrparenthesis}{\vdash \Gamma,\kappa_P\llparenthesis\phi_1,\ldots,\phi_{i-1},\circ,\phi_{i+1},\ldots,\phi_{j-1},\phi_{j+1},\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma\kappa_P\llparenthesis\phi_1,\ldots,\phi_{i-1},\circ,\phi_{i+1},\ldots,\phi_{j-1},\circ,\phi_{j+1},\ldots,\phi_n\rrparenthesis}
\rightsquigarrow
\mathsf{u}_\kappa\dfrac{\mathsf{u}_\kappa\dfrac{\vdash \Gamma,\chi\llparenthesis\phi_1,\ldots,\phi_{i-1},\phi_{i+1},\ldots,\phi_{j-1},\phi_{j+1},\ldots,\phi_n\rrparenthesis}{\vdash \Gamma,\kappa_{P'}\llparenthesis\phi_1,\ldots,\phi_{i-1},\phi_{i+1},\ldots,\phi_{j-1},\circ,\phi_{j+1},\ldots,\phi_n\rrparenthesis}}{\vdash \Gamma,\kappa_P\llparenthesis\phi_1,\ldots,\phi_{i-1},\circ,\phi_{i+1},\ldots,\phi_{j-1},\circ,\phi_{j+1},\ldots,\phi_n\rrparenthesis}
$$

Fig. 5. Special commutative cut-elimination step for u_κ.

Definition 15. *Let P and Q be prime graphs and let $i_1 < \ldots < i_k$ be integers in $\{1,\ldots,|P|\}$. If $P\llparenthesis\circ,\ldots,\circ,v_{i_1},\circ,\ldots,\circ,v_{i_k},\circ,\ldots,\circ\rrparenthesis \sim Q\llparenthesis v_1,\ldots,v_n\rrparenthesis$ for (any) single-vertex graphs $v_1,\ldots,v_n$, then we say that the connective κ_Q is a* ***sub-connective*** *of κ_P and we may write $\kappa_{P|_{i_1,\ldots,i_k}} = \kappa_Q$. A* ***quasi-subformula*** *of a formula $\phi = \kappa_P\llparenthesis\psi_1,\ldots,\psi_n\rrparenthesis$ is a formula of the form $\kappa_{P'|_{i_1,\ldots,i_k}}\llparenthesis\psi'_{i_1},\ldots,\psi'_{i_k}\rrparenthesis$ with ψ'_{i_j} a quasi-subformula of ψ_{i_j} for all $i_j \in \{i_1,\ldots,i_k\}$.*

Corollary 4 (Conservativity). MGL *is a conservative extension of* MLL $= \{\mathsf{ax},\parr,\otimes\}$. MGL$^\circ$ *is a conservative extension of* MLL$^\circ$ $= \{\mathsf{ax},\parr,\otimes,\mathsf{mix}\}$. KGL *is a conservative extension of* LK $=$ MLL $\cup \{\mathsf{w},\mathsf{c}\}$.

Proof. The results for MGL and KGL follow from the fact that these systems satisfy the standard sub-formula property for cut-free derivations, therefore no connective other than $\parr$ and $\otimes$ can be introduced during proof search. The result for MGL$^\circ$ follows from the fact that it satisfies the *quasi-subformula property* (i.e., every formula in the premise of a rule is a quasi-subformula a formula in its conclusion), and that $\parr$ and $\otimes$ have no sub-connectives.

For both MGL and MGL$^\circ$ we have the following *splitting* result, ensuring that it is always possible, during proof search, to apply a rule removing a connective after having applied certain rules in the context. Note that, in the literature of linear logic, the

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

$$\mathsf{wd}_{\parr}\dfrac{\mathsf{u}_{\kappa}\dfrac{\vdash \Gamma, \chi\llparenthesis \phi, \psi_2, \ldots, \psi_{n-1} \rrparenthesis}{\vdash \Gamma, \kappa\llparenthesis \phi, \psi_2, \ldots, \psi_{n-1}, \circ \rrparenthesis}}{\vdash \Gamma, \kappa\llparenthesis \circ, \psi_2, \ldots, \psi_{n-1}, \circ \rrparenthesis, \phi} \rightsquigarrow \mathsf{u}_{\kappa}\dfrac{\mathsf{wd}_{\parr}\dfrac{\vdash \Gamma, \chi\llparenthesis \phi, \psi_2, \ldots, \psi_{n-1} \rrparenthesis}{\vdash \Gamma, \chi\llparenthesis \circ, \psi_2, \ldots, \psi_{n-1} \rrparenthesis, \phi}}{\vdash \Gamma, \kappa\llparenthesis \circ, \psi_2, \ldots, \psi_{n-1}, \circ \rrparenthesis, \phi}$$

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

$$\rightsquigarrow$$

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

$$\mathsf{wd}_{\parr}\dfrac{\mathsf{d}\text{-}\kappa\dfrac{\vdash \Gamma_1, \phi, \psi_1 \quad \vdash \Gamma_2, \phi_2, \psi_2 \quad \cdots \quad \vdash \Gamma_n, \phi_n, \psi_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa^{\perp}\llparenthesis \psi_1, \ldots, \psi_n \rrparenthesis, \kappa\llparenthesis \phi, \phi_2, \ldots, \phi_n \rrparenthesis}}{\vdash \Gamma_1, \ldots, \Gamma_n, \kappa^{\perp}\llparenthesis \psi_1, \ldots, \psi_n \rrparenthesis, \kappa\llparenthesis \circ, \phi_2, \ldots, \phi_n \rrparenthesis, \phi}$$

$$\rightsquigarrow$$

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

Fig. 6. Steps to eliminate $\mathsf{wd}_{\parr}$ rules.

splitting lemma is usually formulated as a special case of the next lemma, ensuring that an occurrence of the connective $\otimes$ can be removed (by applying a $\otimes$-rule), but without requiring the possibility of the need of applying rules to the context.

Lemma 2 (Splitting). *Let $\Gamma, \kappa\llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis$ be a sequent and let $\mathsf{X} \in \{\mathsf{MGL}, \mathsf{MGL}^{\circ}\}$. If $\vdash_{\mathsf{X}} \Gamma, \kappa\llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis$, then there is a derivation of the following shape*

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

Proof. By case analysis of the last rule occurring in a proof π of $\Gamma, \kappa\llparenthesis \phi_1, \ldots, \phi_n \rrparenthesis$.

We conclude this section by proving the admissibility of rules $\mathsf{wd}_{\parr}$ and deep.

Lemma 3. *The rule $\mathsf{wd}_{\parr}$ is admissible in MGL°.*

$$\mathsf{w}\!\downarrow\frac{\vdash \Gamma, \zeta[\psi]}{\vdash \Gamma, \zeta[\psi \parr \phi]} \quad \mathsf{c}\!\downarrow\frac{\vdash \Gamma, \zeta[\phi \parr \phi]}{\vdash \Gamma, \zeta[\phi]} \quad \Bigg| \quad \mathsf{ac}\!\downarrow\frac{\vdash \Gamma, \zeta[a \parr a]}{\vdash \Gamma, \zeta[a]} \quad \mathsf{m}\frac{\vdash \Gamma, \zeta[P(\!|\phi_1, \dots, \phi_n|\!) \parr P(\!|\psi_1, \dots, \psi_n|\!)]}{\vdash \Gamma, \zeta[P(\!|\phi_1 \parr \psi_1, \dots, \phi_n \parr \psi_n|\!)]} \parr \neq P \text{ prime}$$

Fig. 7. Deep inference structural rules, the atomic contraction and the generalized medial rule.

Proof. In Figure 6 we provide a procedure to remove (top-down) all occurrences of $\mathsf{wd}_{\parr}$. Similar to cut-elimination, this procedure requires the use the commutative steps to ensure that the active formula of a $\mathsf{wd}_{\parr}$ we aim at removing is principal with respect to the rule immediately above it.

Lemma 4. *The rule* deep *is admissible in* MGL°.

Proof. By induction on the structure of $\zeta[\square]$. The case with $\zeta[\square] = \square$ is an application of $\mathsf{wd}_\otimes$, otherwise we conclude using Lemma 2.

3.2 A decomposition result for KGL

We can extend the decomposition result for deep inference systems in the context of classical logic [13,15] to KGL using the deep inference (structural) rules from Figure 7, including the ***generalized medial*** rule proposed in [17].

Theorem 6 (Decomposition). *Let Γ be a sequent. If $\vdash_{\mathsf{KGL}} \Gamma$, 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 replacing structural rules with deep ones, and applying rule permutations. Item 2 is a consequence of the previous point after showing (by induction) that each instance of c↓-rule can be replaced by a derivation containing m and ac↓ only, and conclude by applying rule permutations to push ac-rules below m-rules, and w↓ to the bottom of a derivation. For a reference, see [7].

4 Graph Isomorphism as Logical Equivalence

In this section we show that two pure formulas ϕ and ψ are interpreted by the same graph (i.e., $[\![\phi]\!] = [\![\psi]\!]$) iff they are logically equivalent (i.e., $\phi \multimapboth \psi$).

Theorem 7. *Let ϕ and ψ be formulas.*

1. *If ϕ and ψ are unit-free, then $[\![\phi]\!] = [\![\psi]\!]$ iff $\vdash_{\mathsf{MGL}} \phi \multimapboth \psi$.*
2. *If ϕ and ψ are pure, then $[\![\phi]\!] = [\![\psi]\!]$ iff $\vdash_{\mathsf{MGL}^\circ} \phi \multimapboth \psi$.*

Proof. After Proposition 2, to prove Item 1 it suffices to show that each De Morgan law $\phi \equiv \psi$ in Definition 12 (with ϕ and ψ unit-free) corresponds to a logical equivalence $\phi \multimapboth \psi$ which is derivable in MGL. We then conclude by Corollary 3. To prove Item 2, we first show that we can find unit-free formulas ϕ' and ψ' such that $\phi \multimapboth \phi'$ and $\psi \multimapboth \psi'$ are derivable in MGL° (using AX, d-κ, and u_κ only), and we then conclude using the previous point.

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

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

Fig. 8. Inference rules in GS, with P any prime graph and $M_i \neq \varnothing \neq M'_i$ for all $i \in \{1,\ldots,n\}$.

5 Soundness and Completeness of MGL° with respect to GS

In this section, we show that the graphical logic GS from [4,5], defined by a deep inference system operating on graphs, is the set of graphs corresponding to formulas that are provable in MGL°. Note that we here consider the system $\mathsf{GS} = \{\mathsf{ai}\!\downarrow, \mathsf{s}_\parr, \mathsf{s}_\otimes, \mathsf{p}\!\downarrow\}$ defined by the rules in Figure 8, which have a slightly different formulation with respect to [4] and [5]: we consider p-rules with a stronger side condition which is balanced by the presence of $\mathsf{s}_\otimes$ in the system.[4]

To prove the main result of this section, we use the admissibility of $\mathsf{wd}_\parr$ and deep (Lemmas 3 and 4) to prove that if H and G are graphs such that there is an application of a rule $\mathsf{s}_\parr$, $\mathsf{s}_\otimes$, or $\mathsf{p}\!\downarrow$ (even deep in a context) with premise H and conclusion G, then there are formulas ϕ and ψ, with $[\![\phi]\!] = H$ and $[\![\psi]\!] = G$, such that $\psi \multimap \phi$.

Lemma 5. *Let* $\mathsf{r} \in \{\mathsf{s}_\parr, \mathsf{s}_\otimes, \mathsf{p}\!\downarrow\}$. *If* $\mathsf{r}\frac{H}{G}$, *then there are formulas ϕ and ψ with* $[\![\phi]\!] = G$ *and* $[\![\psi]\!] = H$ *such that* $\vdash_{\mathsf{MGL}^\circ} \psi^\perp, \phi$.

Proof. If $C[\square] = \square$, then the following implications trivially hold in MGL°:

$$\begin{aligned}
\kappa(\!|\mu_1,\ldots,\mu_{i-1},\mu_i \parr \nu,\mu_{i+1},\ldots\mu_n|\!) &\multimap \mu_i \parr \kappa(\!|\mu_1,\ldots,\mu_{i-1},\circ \parr \nu,\mu_{i+1},\ldots\mu_n|\!)\\
\mu_i\otimes\kappa(\!|\mu_1,\ldots,\mu_{i-1},\circ\otimes\nu,\mu_{i+1},\ldots\mu_n|\!) &\multimap \kappa(\!|\mu_1,\ldots,\mu_{i-1},\mu_i\otimes\nu,\mu_{i+1},\ldots\mu_n|\!)\\
(\mu_1 \parr \nu_1)\otimes\cdots\otimes(\mu_n \parr \nu_n) &\multimap \kappa_{P^\perp}(\!|\mu_1,\ldots,\mu_n|\!) \parr \kappa_P(\!|\nu_1,\ldots,\nu_n|\!)
\end{aligned}$$

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 $C[\square]$, there is a derivation as follows:

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

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 the image via $[\![\cdot]\!]$ of formulas provable in MGL°.

Theorem 8. *Let* ϕ *a pure formula and let* $G = [\![\phi]\!] \neq \varnothing$. *Then* $\vdash_{\mathsf{GS}} G$ *iff* $\vdash_{\mathsf{MGL}^\circ} \phi$.

[4] The proof that the formulation we consider in this paper, where all factors M_i and N_i are required to be non-empty is equivalent to the ones in the literature, where is either asked that only all factors M_i (as in [5]) or $M_i \parr N_i$ (as in [4]) are non-empty, is provided in [2].

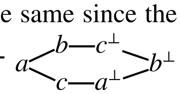
Proof. If there is a derivation π of Γ in MGL°, then we define a derivation $[\![\pi]\!]$ of $[\![\Gamma]\!]$ in GS by induction by induction on the last rule r in π. The translation translates a ax into an instance of $\mathsf{ai}{\downarrow}$, a $\parr$, mix and u_κ into no rule (using properties of the open deduction formalism, and the fact premise and conclusion sequents correspond to the same graph), $\otimes$ and $\mathsf{d}\text{-}\kappa$ into an instance of $\mathsf{p}{\downarrow}$, and $\mathsf{wd}_\otimes$ into an instance of $\mathsf{p}{\downarrow}$.

Conversely, if $\mathcal{D}$ is a proof of $G \neq \emptyset$ in GS, then we define a proof $\pi_{\mathcal{D}}$ of ϕ by induction on the number n of rules in $\mathcal{D}$, where $n \neq 0$ because 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 the derivation $\mathcal{D}$ is of the form $\mathcal{D} = \mathsf{r}\dfrac{\mathcal{D}' \,\| \; H}{G}$ and 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 5 we have a derivation with cut as the one below on the left of a formula ϕ such that $[\![\phi]\!] = G$. Thus we conclude by Theorem 5.

$$\mathsf{cut}\dfrac{\overset{\|\,\mathsf{IH}}{\psi} \qquad \overset{\|\,\text{Lemma } 5}{\vdash \psi^\perp, \phi}}{\vdash \phi} \quad \overset{\textit{Theorem } 5}{\rightsquigarrow^*} \quad \overset{\|\,\mathsf{MGL}^\circ}{\phi} \qquad\Bigg|\qquad \mathsf{deep}\dfrac{\parr\dfrac{\mathsf{ax}\dfrac{}{\vdash a, a^\perp}}{\vdash a \parr a^\perp} \qquad \overset{\pi_{\mathcal{D}'}\,\|\,\mathsf{IH}}{\psi}}{\dfrac{\vdash \zeta[a \parr a^\perp]}{\vdash \phi}\,=}$$

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 4 to the derivation above on the right.

Remark 6. In a different line of work [17] the authors define the ***boolean graphical logic*** (or GBL), as a graphical logic conservatively extending LK defined by maximal-clique-preserving graph morphisms. As a consequence of Corollary 4 and theorem 8, we conclude that KGL and GBL are not the same since the following counterexample from [5] (for GS) is in GBL but not in KGL  .

6 Conclusion and Future Works

In this paper we have provided foundations for the design of proof systems operating on graphs 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 introduced cut-free sequent calculi operating on formulas containing graphical connectives, where graph isomorphism can be captured by logical equivalence. We also discussed the relationship of these systems with graphical logics studied in the literature [4,5,17].

We illustrate below a number of future research directions originating from this work different from the suggestions of the respective authors of using the graphical logic GS to extend the works in [11,49,18], where the authors suggest the possibility

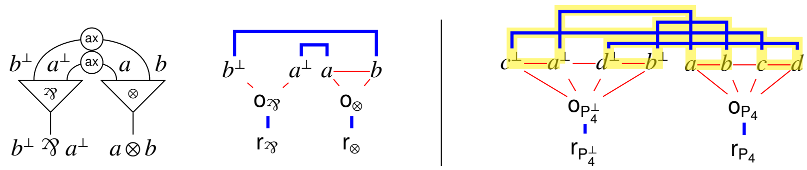


Fig. 9. On the left: the same proof net in the original Girard's syntax and Retoré's one. On the right: an RB-proof net of $\kappa_{\mathsf{P}_4}(\!|a,b,c,d|\!) \multimap \kappa_{\mathsf{P}_4}(\!|a,b,c,d|\!)$ containing the chorded æ-cycle $a \cdot b \cdot b^\perp \cdot d^\perp \cdot d \cdot c \cdot c^\perp \cdot a^\perp$.

of extending their current results by generalizing their methods based on "classical" formulas to graphs.

Categorical Semantics. Unit-free *star-autonomous* and *IsoMix* categories [19,20] provide categorical models of MLL and MLL° respectively. We conjecture that categorical models for MGL and MGL° can be defined by enriching such structures with additional n-ary monoidal products and natural transformations, reflecting the symmetries observed in the symmetry groups of prime graphs.

Digraphs, Games and Event Structures. In this work we have extended the correspondence between classical propositional and cographs from [21] to the case of general (undirected) graphs using graphical connectives, and the same idea can be found in [3] where mixed graphs generalize *relation webs* used to encode BV-formulas [33]. Similarly, we foresee the definition of proof systems operating on directed graphs as conservative extensions of intuitionistic propositional logic beyond *arenas* – directed graphs used in Hyland-Ong *game semantics* [40] to encode propositional intuitionistic formulas, which are characterized by the absence of induced subgraphs of a specific shape. This would provide new insights on the proof theory connected to concurrent games [1,58,64], and could be used to define automated tools operating on event structures [55].

Proof nets and automated proof search. We plan to design proof nets [29,22,30] for MGL and MGL°, as well as combinatorial proofs [39,38] for KGL. For this purpose, we envisage extending Retoré's *handsome proof net* syntax, where proof nets are represented by two-colored graphs (see the left of Figure 9). In Retoré's syntax, the graph induced by the vertices corresponding to the inputs of a $⅋$-gate (or a $\otimes$-gate) is similar to the corresponding prime graph $⅋$ (resp. $\otimes$). Thus, gates for graphical connectives could be easily defined by extending this correspondence (see the proof net on the right of Figure 9). The standard correctness condition defined via *acyclicity* fails for these proof nets, as shown in the right-hand side of Figure 9: the (correct) proof-net of the sequent $\mathsf{P}_4(\!|a,b,c,d|\!) \multimap \mathsf{P}_4(\!|a,b,c,d|\!)$ contains a cycle. We foresee the possibility of using results on the *primeval* decomposition of graphs [42,37] to isolate those cycles witnessing unsoundness, as proposed in [52]. This may provide a methodology to develop machine-learning guided automated theorem provers using the methods in [43].

Acknowledgments

The author thanks the anonymous reviewers for the feedback which helped improve the final version of this manuscript.

References

1. Abramsky, S., Mellies, P.A.: Concurrent games and full completeness. In: Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158). pp. 431–442. IEEE (1999)
2. Acclavio, M.: Graphical proof theory I: Sequent systems on undirected graphs (2023)
3. Acclavio, M., Horne, R., Mauw, S., Straßburger, L.: A Graphical Proof Theory of Logical Time. In: Felty, A.P. (ed.) 7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 228, pp. 22:1–22:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.FSCD.2022.22, https://drops.dagstuhl.de/opus/volltexte/2022/16303
4. Acclavio, M., Horne, R., Straßburger, L.: Logic beyond formulas: A proof system on graphs. In: Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science. p. 38–52. LICS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3373718.3394763, https://doi.org/10.1145/3373718.3394763
5. Acclavio, M., Horne, R., Straßburger, L.: An Analytic Propositional Proof System on Graphs. Logical Methods in Computer Science **Volume 18, Issue 4** (Oct 2022). https://doi.org/10.46298/lmcs-18(4:1)2022, https://lmcs.episciences.org/10186
6. Acclavio, M., Maieli, R.: Generalized connectives for multiplicative linear logic. In: Fernández, M., Muscholl, A. (eds.) 28th EACSL Annual Conference on Computer Science Logic (CSL 2020). LIPIcs, vol. 152, pp. 6:1–6:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2020). https://doi.org/10.4230/LIPIcs.CSL.2020.6, https://drops.dagstuhl.de/opus/volltexte/2020/11649
7. Acclavio, M., Straßburger, L.: From syntactic proofs to combinatorial proofs. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) 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. vol. 10900, pp. 481–497. Springer (2018)
8. Aler Tubella, A., Straßburger, L.: Introduction to Deep Inference (Aug 2019), https://hal.inria.fr/hal-02390267, lecture
9. Andreoli, J.M.: Logic programming with focusing proofs in linear logic. Journal of Logic and Computation **2**(3), 297–347 (1992)
10. Avron, A., Lev, I.: Canonical propositional Gentzen-type systems. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) Automated Reasoning. pp. 529–544. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
11. Bellandi, V., Frati, F., Siccardi, S., Zuccotti, F.: Management of uncertain data in event graphs. In: Ciucci, D., Couso, I., Medina, J., Ślęzak, D., Petturiti, D., Bouchon-Meunier, B., Yager, R.R. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems. pp. 568–580. Springer International Publishing, Cham (2022)
12. Blackburn, P., De Rijke, M., Venema, Y.: Modal logic: graph. Darst, vol. 53. Cambridge University Press (2001)
13. Brünnler, K.: Locality for classical logic. Notre Dame Journal of Formal Logic **47**(4), 557–580 (2006), http://www.iam.unibe.ch/~kai/Papers/LocalityClassical.pdf
14. Brünnler, K., Straßburger, L.: Modular sequent systems for modal logic. In: Giese, M., Waaler, A. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'09. Lecture Notes in Computer Science, vol. 5607, pp. 152–166. Springer (2009)
15. Bruscoli, P., Straßburger, L.: On the length of medial-switch-mix derivations. In: Kennedy, J., de Queiroz, R.J.G.B. (eds.) Logic, Language, Information, and Computation - 24th International Workshop, WoLLIC 2017, London, UK, July 18-21, 2017, Proceedings. Lecture Notes

in Computer Science, vol. 10388, pp. 68–79. Springer (2017). https://doi.org/10.1007/978-3-662-55386-2_5, https://doi.org/10.1007/978-3-662-55386-2_5
16. Calk, C.: A graph theoretical extension of boolean logic (2016), http://www.anupamdas.com/graph-bool.pdf, bachelor's thesis
17. Calk, C., Das, A., Waring, T.: Beyond formulas-as-cographs: an extension of boolean logic to arbitrary graphs (2020)
18. Chaudhuri, K., Donato, P., Massacci, L., Werner, B.: Certifying Proof-By-Linking (Sep 2022), https://inria.hal.science/hal-04317972, working paper or preprint
19. Cockett, J., Seely, R.: Proof theory for full intuitionistic linear logic, bilinear logic, and mix categories. Theory and Applications of Categories **3**(5), 85–131 (1997)
20. Cockett, J., Seely, R.: Weakly distributive categories. J. of Pure and Applied Algebra **114**, 133–173 (1997)
21. Corneil, D., Lerchs, H., Burlingham, L.: Complement reducible graphs. Discrete Applied Mathematics **3**(3), 163–174 (1981). https://doi.org/https://doi.org/10.1016/0166-218X(81)90013-5, https://www.sciencedirect.com/science/article/pii/0166218X81900135
22. Danos, V., Regnier, L.: The structure of multiplicatives. Archive for Mathematical logic **28**(3), 181–203 (1989). https://doi.org/10.1007/BF01622878
23. Das, A.: Complexity of evaluation and entailment in boolean graph logic (2019), http://www.anupamdas.com/complexity-graph-bool-note.pdf, preprint
24. Das, A., Rice, A.A.: New minimal linear inferences in boolean logic independent of switch and medial. In: Kobayashi, N. (ed.) 6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference). LIPIcs, vol. 195, pp. 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.FSCD.2021.14, https://doi.org/10.4230/LIPIcs.FSCD.2021.14
25. Deniélou, P.M., Yoshida, N.: Buffered communication analysis in distributed multiparty sessions. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010 - Concurrency Theory. pp. 343–357. Springer, Berlin, Heidelberg (2010)
26. Ehrenfeucht, A., Harju, T., Rozenberg, G.: The Theory of 2-Structures A Framework for Decomposition and Transformation of Graphs. World Scientific (1999). https://doi.org/10.1142/4197
27. Fu, X., Bultan, T., Su, J.: Analysis of interacting BPEL web services. In: Proceedings of the 13th international conference on World Wide Web. pp. 621–630. ACM (2004)
28. Gallai, T.: Transitiv orientierbare Graphen. Acta Mathematica Academiae Scientiarum Hungarica **18**(1–2), 25–66 (1967)
29. Girard, J.Y.: Linear logic. Theoretical Computer Science **50**, 1–102 (1987). https://doi.org/10.1016/0304-3975(87)90045-4
30. Girard, J.Y.: Proof-nets : the parallel syntax for proof-theory. In: Ursini, A., Agliano, P. (eds.) Logic and Algebra. Marcel Dekker, New York (1996)
31. Girard, J.Y.: Light linear logic. Information and Computation **143**, 175–204 (1998)
32. Girard, J.Y.: On the meaning of logical rules II: multiplicatives and additives. NATO ASI Series F: Computer and Systems Sciences **175**, 183–212 (2000)
33. Guglielmi, A.: A system of interaction and structure. ACM Transactions on Computational Logic **8**(1), 1–64 (2007). https://doi.org/10.1145/1182613.1182614
34. Guglielmi, A., Gundersen, T., Parigot, M.: A proof calculus which reduces syntactic bureaucracy. In: Lynch, C. (ed.) Proceedings of the 21st International Conference on Rewriting Techniques and Applications. LIPIcs, vol. 6, pp. 135–150. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2010). https://doi.org/10.4230/LIPIcs.RTA.2010.135, http://drops.dagstuhl.de/opus/volltexte/2010/2649

35. Habib, M., Paul, C.: A survey of the algorithmic aspects of modular decomposition. Computer Science Review **4**(1), 41–59 (2010). https://doi.org/https://doi.org/10.1016/j.cosrev.2010.01.001, https://www.sciencedirect.com/science/article/pii/S157401371000002X
36. van Heerdt, G., Kappé, T., Rot, J., Silva, A.: Learning pomset automata. In: Kiefer, S., Tasson, C. (eds.) Foundations of Software Science and Computation Structures. pp. 510–530. Springer International Publishing, Cham (2021)
37. Hougardy, S.: On the P4-structure of perfect graphs. Citeseer (1996)
38. Hughes, D.: Proofs Without Syntax. Annals of Mathematics **164**(3), 1065–1076 (2006). https://doi.org/10.4007/annals.2006.164.1065
39. Hughes, D.: Towards Hilbert's 24^{th} problem: Combinatorial proof invariants: (preliminary version). Electr. Notes Theor. Comput. Sci. **165**, 37–63 (2006)
40. Hyland, J.M.E., Ong, C.H.L.: 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. James, L.O., Stanton, R.G., Cowan, D.D.: 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). pp. 281–290 (1972)
42. Jamison, B., Olariu, S.: P-components and the homogeneous decomposition of graphs. SIAM Journal on Discrete Mathematics **8**(3), 448–463 (1995)
43. Kogkalidis, K., Moortgat, M., Moot, R.: Neural proof nets. In: Fernández, R., Linzen, T. (eds.) Proceedings of the 24th Conference on Computational Natural Language Learning. pp. 26–40. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.conll-1.3, https://aclanthology.org/2020.conll-1.3
44. Lellmann, B., Pimentel, E.: Modularisation of sequent calculi for normal and non-normal modalities. ACM Trans. Comput. Logic **20**(2) (feb 2019). https://doi.org/10.1145/3288757, https://doi.org/10.1145/3288757
45. Lovász, L., Plummer, M.D.: Matching theory, vol. 367. American Mathematical Soc. (2009)
46. Mac Lane, S.: Categories for the Working Mathematician. No. 5 in Graduate Texts in Mathematics, Springer (1971)
47. Maieli, R.: Non decomposable connectives of linear logic. Annals of Pure and Applied Logic **170**(11), 102709 (2019). https://doi.org/https://doi.org/10.1016/j.apal.2019.05.006, http://www.sciencedirect.com/science/article/pii/S0168007219300600
48. McConnell, R.M., Spinrad, J.P.: Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In: Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 536–545. SODA '94, Society for Industrial and Applied Mathematics, USA (1994)
49. Mell, S., Bastani, O., Zdancewic, S.: Ideograph: A language for expressing and manipulating structured data. In: Grabmayer, C. (ed.) Proceedings Twelfth International Workshop on Computing with Terms and Graphs, TERMGRAPH@FSCD 2022, Technion, Haifa, Israel, 1st August 2022. EPTCS, vol. 377, pp. 65–84 (2022). https://doi.org/10.4204/EPTCS.377.4, https://doi.org/10.4204/EPTCS.377.4
50. Miller, D., Pimentel, E.: A formal framework for specifying sequent calculus proof systems. Theoretical Computer Science **474**, 98–116 (2013)
51. Miller, D., Saurin, A.: From proofs to focused proofs: a modular proof of focalization in linear logic. In: Duparc, J., Henzinger, T.A. (eds.) CSL 2007: Computer Science Logic. LNCS, vol. 4646, pp. 405–419. Springer-Verlag (2007)
52. Nguyên, L.T.D., Seiller, T.: Coherent interaction graphs: A non-deterministic geometry of interaction for mll (2019)

53. Nguyên, L.T.D., Straßburger, L.: A System of Interaction and Structure III: The Complexity of BV and Pomset Logic (2022), https://hal.inria.fr/hal-03909547, working paper or preprint
54. Nguyên, L.T.D., Straßburger, L.: BV and Pomset Logic are not the same. In: Manea, F., Simpson, A. (eds.) 30th EACSL Annual Conference on Computer Science Logic (CSL 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 216, pp. 3:1–3:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.CSL.2022.3, https://drops.dagstuhl.de/opus/volltexte/2022/15723
55. Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains, part i. Theoretical Computer Science **13**(1), 85–108 (1981)
56. Pratt, V.: Modeling concurrency with partial orders. International journal of parallel programming **15**, 33–71 (1986)
57. Retoré, C.: Pomset logic: The other approach to noncommutativity in logic. Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics pp. 299–345 (2021)
58. Rideau, S., Winskel, G.: Concurrent strategies. In: 2011 IEEE 26th Annual Symposium on Logic in Computer Science. pp. 409–418. IEEE (2011)
59. Seely, R.: Linear logic, *-autonomous categories and cofree coalgebras. Contemporary Mathematics **92** (1989)
60. Tiu, A.F.: A system of interaction and structure II: The need for deep inference. Logical Methods in Computer Science **2**(2), 1–24 (2006). https://doi.org/10.2168/LMCS-2(2:4)2006
61. Troelstra, A.S., Schwichtenberg, H.: Basic Proof Theory. Cambridge University Press, second edn. (2000)
62. Valdes, J., Tarjan, R.E., Lawler, E.L.: The recognition of series parallel digraphs. In: Proceedings of the eleventh annual ACM symposium on Theory of computing. pp. 1–12. ACM (1979)
63. Waring, T.: A graph theoretic extension of boolean logic (2019), http://anupamdas.com/thesis_tim-waring.pdf, master's thesis
64. Winskel, G., Rideau, S., Clairambault, P., Castellan, S.: Games and strategies as event structures. Logical Methods in Computer Science **13** (2017)