An Analytic Propositional Proof System On Graphs

Matteo Acclavio, Ross Horne, Lutz Strassburger

AN ANALYTIC PROPOSITIONAL PROOF SYSTEM ON GRAPHS

MATTEO ACCLAVIO, ROSS HORNE, AND LUTZ STRASSBURGER

Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg
URL: matteoacclavio.com/Math.html

Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg
e-mail address: ross.horne@uni.lu

Inria, Equipe Partout, Ecole Polytechnique, LIX UMR 7161, France
URL: www.lix.polytechnique.fr/Labo/Lutz.Strassburger/

Abstract. In this paper we present a proof system that operates on graphs instead of formulas. Starting from the well-known relationship between formulas and cographs, we drop the cograph-conditions and look at arbitrary (undirected) graphs. This means that we lose the tree structure of the formulas corresponding to the cographs, and we can no longer use standard proof theoretical methods that depend on that tree structure. In order to overcome this difficulty, we use a modular decomposition of graphs and some techniques from deep inference where inference rules do not rely on the main connective of a formula. For our proof system we show the admissibility of cut and a generalization of the splitting property. Finally, we show that our system is a conservative extension of multiplicative linear logic with mix, and we argue that our graphs form a notion of generalized connective.

Contents

Key words and phrases: Proof theory, prime graphs, cut elimination, deep inference, splitting, analyticity.
* This is an extended version of a paper published at LICS 2020 [AHS20].

Preprint submitted to
Logical Methods in Computer Science

1. Introduction

The notion of formula is central to all applications of logic and proof theory in computer science, ranging from the formal verification of software, where a formula describes a property that the program should satisfy, to logic programming, where a formula represents a program [MNPS91, KY93], and functional programming, where a formula represents a type [How80]. Proof theoretical methods are also employed in concurrency theory, where a formula can represent a process whose behaviours may be extracted from a proof of the formula [Mil93, Bru02, HT19, Hor19, Hor20]. This *formulas-as-processes* paradigm is not as well-investigated as the *formulas-as-properties*, *formulas-as-programs* and *formulas-as-types* paradigms mentioned before. In our opinion, a reason for this is that the notion of formula reaches its limitations when it comes to describing processes as they are studied in concurrency theory.

For example, BV [Gug07] and *pomset logic* [Ret97] are proof systems which extend linear logic with a notion of sequential composition and can model series-parallel orders. However, series-parallel orders cannot express some ubiquitous patterns of *causal dependencies* such as producer-consumer queues [LW00], which are within the scope of pomsets [Pra86], event structures [NPW81], and Petri nets [Pet77]. The essence of this problem is already visible when we consider *symmetric dependencies*, such as separation, which happens to be the dual concept to concurrency in the *formulas-as-processes* paradigm.

Let us use some simple examples to explain the problem. Suppose we are in a situation where two processes A and B can communicate with each other, written as $A \parr B$, or can be separated from each other, written as $A \otimes B$, such that no communication is possible. Now assume we have four atomic processes a, b, c, and d, from which we form the two processes $P = (a \otimes b) \parr (c \otimes d)$ and $Q = (a \parr c) \otimes (b \parr d)$. Both are perfectly fine formulas of multiplicative linear logic (MLL) [Gir87a]. In P, we have that a is separated from b but can communicate with c and d. Similarly, d can communicate with a and b but is separated from c, and so on. On the other hand, in Q, a can only communicate with c and is separated from the other two, and d can only communicate with b, and is separated from the other two. We can visualize this situation via graphs where a, b, c, and d are the vertices, and we draw an edge between two vertices if they are separated, and no edge if they can communicate. Then P and Q correspond to the two graphs shown below.

$$P = (a \otimes b) \parr (c \otimes d) \qquad Q = (a \parr c) \otimes (b \parr d) \tag{1.1}$$

It should also be possible to describe a situation where a is separated from b, and b is separated from c, and c is separated from d, but a can communicate with c and d, and b can communicate with d, as indicated by the graph below.

(1.2)

However, this graph cannot be described by a formula in such a way that was possible for the two graphs in (1.1). Consequently, the tools of proof theory, that have been developed over the course of the last century, and that were very successful for the *formulas-as-properties*,

formulas-as-programs, and *formulas-as-types* paradigms, cannot be used for the *formulas-as-processes* paradigm unless situations as in (1.2) above are forbidden. This seems to be a very strong and unnatural restriction, and the purpose of this paper is to propose a way to change this unsatisfactory situation.

We will present a proof system, called GS (for *graphical proof system*), whose objects of reason are not formulas but graphs, giving the example in (1.2) the same status as the examples in (1.1). In a less informal way, one could say that standard proof systems work on cographs (which are the class of graphs that correspond to formulas as in (1.1)), and our proof systems works on arbitrary graphs. In order for this to make sense, our proof system should obey the following basic properties:

(1) *Consistency*: There are graphs that are not provable.
(2) *Transitivity*: The proof system should come with an implication that is transitive, i.e., if we can prove that A implies B and that B implies C, then we should also be able to prove that A implies C.
(3) *Analyticity*: As we no longer have formulas, we cannot ask that every formula that occurs in a proof is a subformula of its conclusion. But we can ask that in a proof search situation, there is always only a finite number of ways to apply an inference rule.
(4) *Minimality*: We want to make as few assumptions as possible, so that the theory we develop is as general as possible.

Properties 1-3 are standard for any proof system, and they are usually proved using cut elimination. In that respect our paper is no different. We introduce a notion of cut and show its admissibility for GS. Then Properties 1-3 are immediate consequences.

Property 4 is of a more subjective nature. In our case, we only make the following two basic assumptions:

(1) For any graph A, we should be able to prove that A implies A. This assumption is almost impossible to argue against, so can be expected for any logic.
(2) If a graph A is provable, then the graph $G = C[A]$ is also provable, provided that $C[\cdot]$ is a provable context.[1] This can be compared to the *necessitation rule* of modal logic, which says that if A is provable then so is $\Box A$, except that in our case the $\Box$ is replaced by the provable graph context $C[\cdot]$.

All other properties of the system GS follow from the need to obtain admissibility of cut. This means that this paper does not present some random system, but follows the underlying principles of proof theory.

We also target the desirable property of *conservativity*. That is, it is desirable for there to be a well-known logic L based on formulas such that when we restrict our proof system to graphs corresponding to formulas, then we prove exactly the theorems of L. This cannot be an assumption used to design a logical system, since it would create circularity (to specify a logic we need a logic); conservativity is more so a cultural sanity condition to check that we have not invented an esoteric logic. Conservativity follows from cut admissibility, where in our case the logic L is multiplicative linear logic with mix (MLL°) [Gir87a, Bel97, FR94].

Logics are not really designed, they are discovered, since a logic will typically follow logical principles where design parameters are limited. For example, we will see that we do

[1]Formally, the notation $G = C[A]$ means that A is a module of G, and $C[\cdot]$ is the graph obtained from G by removing all vertices belonging to A. We give the formal definition in Section 3.

not really get to chose whether or not the following implications hold:

(1.3)

There is no pre-existing semantics or proof system we can refer to at this point. Nonetheless, from the previously discussed principles we can argue that, in a logic on graphs, that, in (1.3), the former implication cannot hold while the latter must hold. Over the course of this paper, we explore the design of proof systems on graphs based on logical principles, which enables us to confidently state such facts.

Let us now summarize how this paper is organized. In Section 2, we give preliminaries on cographs, which form the class of graphs that correspond to formulas as in (1.1). Then, in Section 3 we give some preliminaries on modules and prime graphs, which are needed for our move away from cographs, so that in Section 4, we can present our proof system, which uses the notation of open deduction [GGP10] and follows the principles of deep inference [GS01, BT01, Gug07]. To our knowledge, this is the first proof system that is not tied to formulas/cographs but handles arbitrary (undirected) graphs instead. In Section 5 we show some properties of our system, and Sections 6 and 7 are dedicated to cut elimination. In these sections we explain the technology we must develop in order to be able to prove cut elimination for our proof system. The interesting point is that, not only do we go beyond methods developed for the sequent calculus, but we also go beyond methods developed for deep inference on formulas. In particular, we require entirely new statements of the tools called *splitting* and *context reduction*, and furthermore their proofs are inter-dependent, whereas normally context reduction follows from splitting.

Then, in Section 8, we show that our system is a conservative extension of MLL°. Finally, in Section 9, we show how our work is related to the work on generalized connectives. We end this paper with a discussion of related work in Section 10, a discussion of what it means for a proof system to be analytic in Section 11, and a conclusion in Section 12.

Compared to the conference version [AHS20] of this paper, there are the following two major additions:

- We give detailed proofs of the *Splitting Lemma* and the *Context Reduction Lemma* (in Section 6), which are crucial for the cut elimination proof. In fact, we also completely reorganized the proofs with respect to the technical appendix of [AHS20][2]. For proving these lemmas, we could not rely on the general method that has been proposed by Aler Tubella in her PhD [AT17] and that works for many deep inference systems. The reason is due to the fact that we use the notion of modular decomposition for graphs from [Gal67] to recover a tree-like structure for graphs. This have a more complex branching structure with respect to formula trees; hence contexts are treated differently in the statements of our versions of these lemmas.
- We show that general graphs with n vertices can be seen as generalized n-ary connectives (in Section 9), and we compare this notion with the existing notion of generalized (multiplicative) connective [Gir87b, DR89, Mai19, AM20].

[2]This appendix is available at `https://hal.inria.fr/hal-02560105`.

2. From Formulas to Graphs

In this preliminary section we recall the basic textbook definitions for graphs and formulas, and show how they are related via cographs.

Definition 2.1. A ***(simple, undirected) graph*** G is a pair $\langle V_G, E_G\rangle$ where V_G is a set of vertices and E_G is a set of two-element subsets of V_G. We omit the index G when it is clear from the context. For $v, w \in V_G$ we write vw as an abbreviation for $\{v, w\}$. A graph G is ***finite*** if its vertex set V_G is finite. Let L be a set and G be a graph. We say that G is ***L-labelled*** (or just ***labelled*** if L is clear from context) if every vertex in V_G is associated with an element of L, called its ***label***. We write $\ell_G(v)$ to denote the label of the vertex v in G. A graph G' is a ***subgraph*** of a graph G, denoted as $G' \subseteq G$ iff $V_{G'} \subseteq V_G$ and $E_{G'} \subseteq E_G$. We say that G' is an ***induced subgraph*** of G if G' is a subgraph of G and for all $v, w \in V_{G'}$, if $vw \in E_G$ then $vw \in E_{G'}$. For a graph G we write $|V_G|$ for its number of vertices and $|E_G|$ for its number of edges.

In the following, we will just say *graph* to mean a finite, undirected, labelled graph, where the labels come from the set $\mathcal{A}$ of atoms which is the (disjoint) union of a countable set of propositional variables $\mathcal{V} = \{a, b, c, \ldots\}$ and their duals $\mathcal{V}^\perp = \{a^\perp, b^\perp, c^\perp, \ldots\}$.

Since we are mainly interested in how vertices are labelled, but not so much in the identity of the underlying vertex, we heavily rely on the notion of graph isomorphism.

Definition 2.2. Two graphs G and G' are ***isomorphic*** if there exists a bijection $f\colon V_G \to V_{G'}$ such that for all $v, u \in V_G$ we have $vu \in E_G$ iff $f(v)f(u) \in E_{G'}$ and $\ell_G(v) = \ell_{G'}(f(v))$. We denote this as $G \simeq_f G'$, or simply as $G \simeq G'$ if f is clear from context or not relevant.

In the following, we will, in diagrams, forget the identity of the underlying vertices, showing only the label, as in the examples in the introduction.

In the rest of this section we recall the characterization of those graphs that correspond to formulas. For simplicity, we restrict ourselves to only two connectives, and for reasons that will become clear later, we use the $\parr$ (***par***) and $\otimes$ (***tensor***) of linear logic [Gir87a]. More precisely, ***formulas*** are generated by the grammar

$$\phi, \psi ::= \circ \mid a \mid a^\perp \mid \phi \parr \psi \mid \phi \otimes \psi \tag{2.1}$$

where $\circ$ is the ***unit***, and a can stand for any propositional variable in $\mathcal{V}$. As usual, we can define the negation of formulas inductively by letting $a^{\perp\perp} = a$ for all $a \in \mathcal{V}$, and by using the De Morgan duality between $\parr$ and $\otimes$: $(\phi \parr \psi)^\perp = \phi^\perp \otimes \psi^\perp$ and $(\phi \otimes \psi)^\perp = \phi^\perp \parr \psi^\perp$; the unit is self-dual: $\circ^\perp = \circ$.

On formulas we define the following structural equivalence relation:

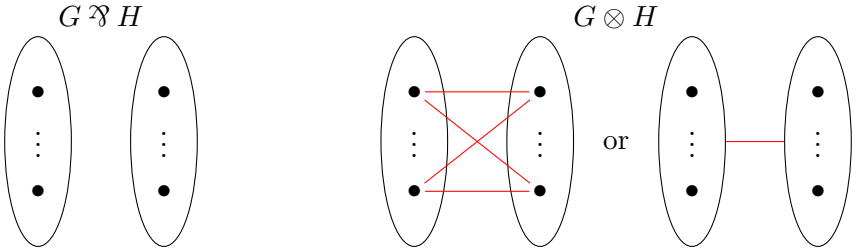
$$\begin{array}{cc} \phi \parr (\psi \parr \xi) \equiv (\phi \parr \psi) \parr \xi & \phi \otimes (\psi \otimes \xi) \equiv (\phi \otimes \psi) \otimes \xi \\ \phi \parr \psi \equiv \psi \parr \phi & \phi \otimes \psi \equiv \psi \otimes \phi \\ \phi \parr \circ \equiv \phi & \phi \otimes \circ \equiv \phi \end{array}$$

In order to translate formulas to graphs, we define the following two operations on graphs:

Definition 2.3. Let $G = \langle V_G, E_G\rangle$ and $H = \langle V_H, E_H\rangle$ be graphs with $V_G \cap V_H = \varnothing$. We define the ***par*** and ***tensor*** operations between them as follows:

$$\begin{aligned} G \parr H &= \langle V_G \cup V_H, E_G \cup E_H\rangle \\ G \otimes H &= \langle V_G \cup V_H, E_G \cup E_H \cup \{vw \mid v \in V_G, w \in V_H\}\rangle \end{aligned}$$

These operations can be visualized as follows:



(2.2)

For a formula ϕ, we can now define its associated graph $[\![\phi]\!]$ inductively as follows: $[\![\circ]\!] = \varnothing$ the empty graph; $[\![a]\!] = a$ a single-vertex graph whose vertex is labelled by a (by a slight abuse of notation, we denote that graph also by a); similarly $[\![a^\perp]\!] = a^\perp$; finally we define $[\![\phi \parr \psi]\!] = [\![\phi]\!] \parr [\![\psi]\!]$ and $[\![\phi \otimes \psi]\!] = [\![\phi]\!] \otimes [\![\psi]\!]$.

Theorem 2.4. *For any two formulas, $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$.*

Proof. By a straightforward induction. □

Definition 2.5. A graph is **P_4-*free*** (or ***N*-free** or ***Z*-free**) iff it does not have an induced subgraph of the shape

(2.3)

Theorem 2.6. *Let G be a graph. Then there is a formula ϕ with $[\![\phi]\!] = G$ iff G is P_4-free.*

A proof of this can be found, e.g., in [Möh89] or [Gug07].

The graphs characterized by Theorem 2.6 are called ***cographs***, because they are the smallest class of graphs containing all single-vertex graphs and being closed under complement and disjoint union.

Because of Theorem 2.6, one can think of standard proof system as *cograph proof systems*. Since in this paper we want to move from cographs to general graphs, we need to investigate, how much of the tree structure of formulas (which makes cographs so interesting for proof theory [Ret03, Hug06, Str17]) can be recovered for general graphs.

3. Modules and Prime Graphs

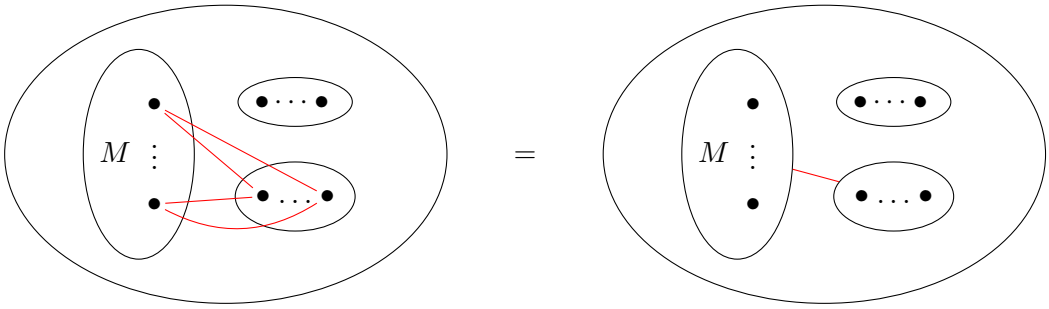
In this second preliminary section we look more closely at some of the concepts that make working with formulas so convenient and lift them from cographs to general graphs.

Definition 3.1. Let G be a graph. A ***module*** of G is an induced subgraph $M = \langle V_M, E_M \rangle$ of G such that for all $v \in V_G \setminus V_M$ and all $x, y \in M$ we have $vx \in E_G$ iff $vy \in E_G$.

Modules are used in this paper since they are for graphs what subformulas are for formulas.

Notation 3.2. Let G be a graph and M be a module of G. Let $V_C = V_G \setminus V_M$ and let C be the graph obtained from G by removing all vertices in M (including incident edges). Let $R \subseteq V_C$ be the set of vertices that are connected to a vertex in V_M (and hence to all vertices in M). We denote this situation as $G = C[M]_R$ and call $C[\cdot]_R$ (or just C) the ***context*** of M in G. Alternatively, $C[M]_R$ can be defined as follows. If we write $C[x]_R$ for a graph in which x is a distinct vertex and R is the set of neighbours x, then $C[M]_R$ is the graph obtained from $C[x]_R$ by substituting M for x.

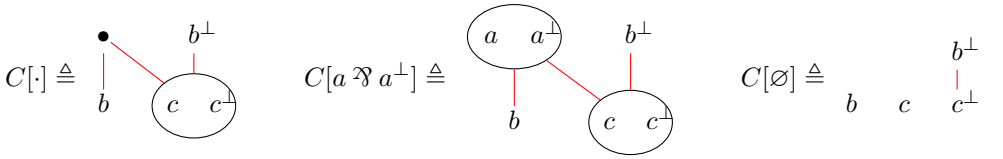
Notation 3.3. As a module M of a graph G is a subset of vertices M, where each vertex outside M has the same relation with all vertices inside M, we can use the following notation when drawing diagramms



(3.1)

which allows us to reduce the number of drawn edges to increase readability. We used this already in (2.2).

Example 3.4. Consider the graph context on the left below with the hole denoted by a bullet. If we substitute the two-vertex graph $a \parr a^\perp$ for the hole, then we obtain the graph in the middle below:



(3.2)

Finally, if we replace the hole with the empty graph we obtain the graph on the right above.

Lemma 3.5. *Let G be a graph and M, N be modules of G. Then*

(1) *$M \cap N$ is a module of G;*
(2) *if $M \cap N \neq \varnothing$, then $M \cup N$ is a module of G; and*
(3) *if $N \not\subseteq M$ then $M \setminus N$ is a module of G.*

Proof. The first statement follows immediately from the definition. For the second one, let $L = M \cap N \neq \varnothing$, and let $v \in G \setminus (V_M \cup V_N)$ and $x, y \in V_M \cup V_N$. If x, y are both in M or both in N, then we have immediately $vx \in E_G$ iff $vy \in E_G$. So, let $x \in V_M$ and $y \in V_N$, and let $z \in L$. We have $vx \in E_G$ iff $vz \in E_G$ iff $vy \in E_G$. Finally, for the last statement, let $x, y \in V_M \setminus V_N$ and let $v \in V_G \setminus (V_M \setminus V_N)$. If $v \notin V_M$, we immediately have $vx \in E_G$ iff $vy \in E_G$. So, let $v \in V_M$, and therefore $v \in V_M \cap V_N$. Let $z \in V_N \setminus V_M$. Then $vx \in E_G$ iff $zx \in E_G$ iff $zy \in E_G$ iff $vy \in E_G$. □

Definition 3.6. Let G be a graph. A module M in G is ***maximal*** if for all modules M' of G such that $M \neq G$ we have that $M \subseteq M'$ implies $M = M'$.

Definition 3.7. A module M of a graph G is ***trivial*** iff either $V_M = \varnothing$ or V_M is a singleton or $V_M = V_G$. A graph G is ***prime*** iff $|V_G| \geq 2$ and all modules of G are trivial.

Definition 3.8. Let G be a graph with n vertices $V_G = \{v_1, \ldots, v_n\}$ and let $H_1, \ldots, H_n$ be n graphs. We define the ***composition of $H_1, \ldots, H_n$ via G***, denoted as $G(\!|H_1, \ldots, H_n|\!)$, by replacing each vertex v_i of G by the graph H_i; and there is an edge between two vertices x and y if either x and y are in the same H_i and $xy \in E_{H_i}$ or $x \in V_{H_i}$ and $y \in V_{H_j}$ for $i \neq j$

and $v_iv_j \in E_G$. Formally, $G(\!|H_1, \ldots, H_n|\!) = \langle V^*, E^* \rangle$ with

$$V^* = \bigcup_{1 \leq i \leq n} V_{H_i} \qquad \text{and} \qquad E^* = \left(\bigcup_{1 \leq i \leq n} E_{H_i} \right) \cup \left\{ xy \mid x \in V_{H_i}, y \in V_{H_j}, v_iv_j \in E_G \right\}$$

This concept allows us to decompose graphs into prime graphs (via Lemma 3.9 below) and recover a tree structure for an arbitrary graph.[3] The two operations $\parr$ and $\otimes$, defined in Definition 2.3 are then represented by the following two prime graphs:

$$\parr\colon \quad \bullet \qquad \bullet \qquad \text{and} \qquad \otimes\colon \quad \bullet\!\!-\!\!-\!\!\bullet \tag{3.3}$$

We have $\parr(\!|G, H|\!) = G \parr H$ and $\otimes(\!|G, H|\!) = G \otimes H$.

Lemma 3.9. *For every non-empty graph G we have exactly one of the following four cases:*

(i) *G is a singleton graph.*
(ii) *$G = A \parr B$ for some A, B with $A \neq \varnothing \neq B$.*
(iii) *$G = A \otimes B$ for some A, B with $A \neq \varnothing \neq B$.*
(iv) *$G = P(\!|A_1, \ldots, A_n|\!)$ for some prime graph P with $n = |V_P| \geq 4$ and $A_i \neq \varnothing$ for every $i \in \{1, \ldots, n\}$.*

Proof. Let G be given. If $|G| = 1$, we are in case (i). Now assume $|G| > 1$, and let $M_1, \ldots, M_n$ be the maximal modules of G. Now we have two cases:

- For all $i, j \in \{1, \ldots, n\}$ with $i \neq j$ we have $M_i \cap M_j = \varnothing$. Since every vertex of G forms a module, every vertex must be part of a maximal module. Hence $V_G = V_{M_1} \cup \cdots \cup V_{M_n}$. Therefore there is a graph P such that $G = P(\!|M_1, \ldots, M_n|\!)$. Since all M_i are maximal in G, we can conclude that P is prime. If $|V_P| \geq 4$ we are in case (iv). If $|V_P| < 4$ we are either in case (ii) or (iii), as the two graphs in (3.3) are only two prime graphs with $|V_P| = 2$, and there are no prime graphs with $|V_P| = 3$.
- We have some $i \neq j$ with $M_i \cap M_j \neq \varnothing$. Let $L = M_i \cap M_j$ and $N = M_i \setminus M_j$ and $K = M_j \setminus M_i$. By Lemma 3.5, L, N, K, and $M_i \cup M_j$ are all modules of G. Since M_i and M_j are maximal, it follows that $G = M_i \cup M_j$, and therefore $G = N \otimes L \otimes K$ or $G = N \parr L \parr K$. □

This lemma allows us to use the modular decomposition of graphs [Gal67] in a similar way as formula trees.

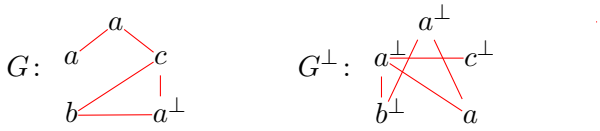
4. The Proof System

To define a proof system, we need a notion of implication. To do so, we first introduce a notion of negation.

Definition 4.1. For a graph $G = \langle V_G, E_G \rangle$, we define its ***dual*** $G^\perp = \langle V_G, E_{G^\perp} \rangle$ to have the same set of vertices, and an edge $vw \in E_{G^\perp}$ iff $vw \notin E_G$ (and $v \neq w$). The label of a vertex v in $G^\perp$ is the dual of the label of that vertex in G, i.e., $\ell_{G^\perp}(v) = \ell_G(v)^\perp$. For any two graphs G and H, the ***implication*** $G \multimap H$ is defined to be the graph $G^\perp \parr H$.

[3] This allows us to consider prime graphs as generalized non-decomposable n-ary connectives. We will come back to this point in Section 9.

Example 4.2. To give an example, consider the graph G on the left below



(4.1)

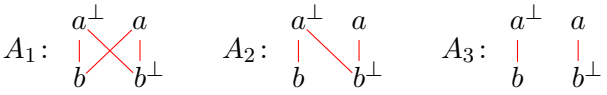
Its negation $G^\perp$ is shown on the right above.

Observation 4.3. The dual graph construction defines the standard De Morgan dualities relating conjunction and disjunction, i.e., for every formula ϕ, we have $[\![\phi^\perp]\!] = [\![\phi]\!]^\perp$. De Morgan dualities extended to prime graphs as $P(\!|M_1, \ldots, M_n|\!)^\perp = P^\perp(\!|M_1^\perp, \ldots, M_n^\perp|\!)$, where $P^\perp$ is the dual graph of P. Furthermore, $P^\perp$ is prime if and only if P is prime. Thus each pair of prime graphs P and $P^\perp$ defines a pair of connectives that are De Morgan duals to each other.

We will now develop our proof system based on the above notion of negation as graph duality. From the requirements mentioned in the introduction it follows that:

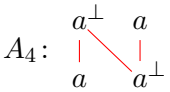
(i) for any isomorphic graphs G and H, the graph $G \multimap H$ should be provable;
(ii) if $G \neq \varnothing$ then G and $G^\perp$ should not be both provable;
(iii) the implication $\multimap$ should be transitive, i.e., if $G \multimap H$, and $H \multimap K$ are provable then so should be $G \multimap K$;
(iv) the implication $\multimap$ should be closed under context, i.e., if $G \multimap H$ is provable and $C[\cdot]_R$ is an arbitrary context, then $C[G]_R \multimap C[H]_R$ should be provable;
(v) if A and C are provable graphs, and $R \subseteq V_C$, then the graph $C[A]_R$ should also be provable.

Example 4.4. As an example, consider the following three graphs:



(4.2)

The graph A_1 on the left should clearly be provable, as it corresponds to the formula $(a^\perp \parr a) \otimes (b \parr b^\perp)$, which is provable in multiplicative linear logic (MLL).[4] The graph A_3 on the right should not be provable, as it corresponds to the formula $(a^\perp \otimes b) \parr (a \otimes b^\perp)$, which is not provable in MLL. But what about the graph A_2 in the middle? It does not correspond to a formula, and therefore we cannot resort to MLL. Nonetheless, we can make the following observations. If A_2 were provable, then so would be the graph A_4 shown below:



(4.3)

as it is obtained from A_2 by a simple substitution. However, $A_4^\perp = A_4$, and therefore $A_4^\perp$ and A_4 would both be provable, which would be a contradiction and should be ruled out. Hence, A_2 should not be provable. It also follows that $A_1 \multimap A_2$ cannot hold, as otherwise we would be able to use A_1 and modus ponens to establish that A_2 is provable, which cannot hold as we just observed. By applying a dual argument, we conclude that $A_2 \multimap A_3$ cannot hold either.

[4]We come back to MLL in Section 8.

This example also shows that that implication is not simply subset inclusion of edges. However, in our minimal logic on graphs that we present here, the converse does hold: we will see later that whenever we have $G \multimap H$ and $V_G = V_H$ then $E_H \subseteq E_G$.[5]

For presenting the inference system we use a *deep inference* formalism [GS01, Gug07], which allows rewriting inside an arbitrary context and admits a rather flexible composition of derivations. In our presentation we will follow the notation of *open deduction*, introduced in [GGP10].

Let us start with the following two inference rules

$$\mathsf{i}{\downarrow}\,\frac{\varnothing}{A^{\perp} \parr A} \qquad\qquad \mathsf{ss}{\uparrow}\,\frac{B \otimes A}{B[A]_S}\ {\scriptstyle S\subseteq V_B,\ S\neq V_B} \tag{4.4}$$

which are induced by the two Points (i) and (v) above, and which are called ***identity down*** and ***super switch up***, respectively. The $\mathsf{i}{\downarrow}$ says that for arbitrary graphs C and A and any $R \subseteq V_C$, if C is provable, then so is the graph $C[A \parr A^{\perp}]_R$. Similarly, the rule $\mathsf{ss}{\uparrow}$ says that whenever $C[B \otimes A]_R$ is provable, then so is $C[B[A]_S]_R$ for any three graphs A, B, C and any $R \subseteq V_C$ and $S \subseteq V_B$. The condition $S \neq V_B$ is there to avoid a trivial rule instance, as $B[A]_S = B \otimes A$ if $S = V_B$.

Definition 4.5. An ***inference system*** S is a set of inference rules. We define the set of ***derivations*** in S inductively below, and we denote a derivation $\mathcal{D}$ in S with premise G and conclusion H, as follows:

$$\begin{array}{c} G \\ \mathcal{D} \,\|\, \mathsf{S} \\ H \end{array}$$

(1) Every graph G is a derivation (also denoted by G) with premise G and conclusion G.
(2) 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, then $\mathcal{D}_1 \parr \mathcal{D}_2$ is a derivation with premise $G_1 \parr G_2$ and conclusion $H_1 \parr H_2$, and similarly, $\mathcal{D}_1 \otimes \mathcal{D}_2$ is a derivation with premise $G_1 \otimes G_2$ and conclusion $H_1 \otimes H_2$, denoted as

$$\begin{array}{c} G_1 \\ \mathcal{D}_1 \,\|\, \mathsf{S} \\ H_1 \end{array} \parr \begin{array}{c} G_2 \\ \mathcal{D}_2 \,\|\, \mathsf{S} \\ H_2 \end{array} \qquad \text{and} \qquad \begin{array}{c} G_1 \\ \mathcal{D}_1 \,\|\, \mathsf{S} \\ H_1 \end{array} \otimes \begin{array}{c} G_2 \\ \mathcal{D}_2 \,\|\, \mathsf{S} \\ H_2 \end{array}$$

respectively.
(3) 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

$$\mathsf{r}\,\frac{H_1}{G_2}$$

[5]But this observation is not true in general for all logics that might be designed on graphs. For example in the extension of Boolean logic, defined in [CDW20], is is not necessarily true that implication preserves edges.

is an instance of an inference rule r, then $\mathcal{D}_2 \circ_{\mathsf{r}} \mathcal{D}_1$ is a derivation with premise G_2 and conclusion H_2, denoted as

$$\mathsf{r}\,\frac{\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \end{array}}{\begin{array}{c} G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array}} \qquad \text{or} \qquad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \\ \mathsf{r}\,\dfrac{}{G_2} \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array}$$

If $H_1 \simeq_f G_2$ we can compose $\mathcal{D}_1$ and $\mathcal{D}_2$ directly to $\mathcal{D}_2 \circ \mathcal{D}_1$, denoted as

$$f\,\frac{\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \end{array}}{\begin{array}{c} G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array}} \quad \text{or} \quad \simeq\,\frac{\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \end{array}}{\begin{array}{c} G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array}} \quad \text{or} \quad \frac{\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \end{array}}{\begin{array}{c} G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array}} \tag{4.5}$$

If f is the identity, i.e., $H_1 = G_2$, we can write $\mathcal{D}_2 \circ \mathcal{D}_1$ as

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

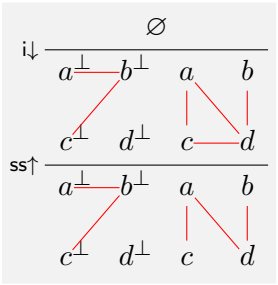
A ***proof*** in S is a derivation in S whose premise is $\varnothing$. A graph G is ***provable*** in S iff there is a proof in S with conclusion G. We denote this as $\vdash_{\mathsf{S}} G$ (or simply as $\vdash G$ if S is clear from context). The ***length*** of a derivation $\mathcal{D}$, denoted by $|\mathcal{D}|$, is the number of inference rule instances in $\mathcal{D}$.

Remark 4.6. If we have a derivation $\mathcal{D}$ from A to B, and a context $G[\cdot]_R$, then we also have a derivation from $G[A]_R$ to $G[B]_R$. We can write this derivation as

$$\begin{array}{c} G[A]_R \\ G[\mathcal{D}]_R \Vert \\ G[B]_R \end{array} \qquad \text{or} \qquad G[\,\begin{array}{c} A \\ \mathcal{D} \Vert \\ B \end{array}\,]_R \qquad \text{or} \qquad G\left[\begin{array}{c} A \\ \mathcal{D} \Vert \\ B \end{array}\right]_R$$

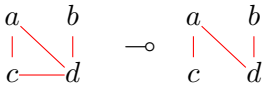
Example 4.7. Let us emphasize that the conclusion of a proof in our system is not a formula but a graph. The following derivation is an example of a proof of length 2, using

only i↓ and ss↑:



(4.6)

where the ss↑ instance moves the module d in the context consisting of vertices labelled a, b, c. The derivation in (4.6) establishes that the following implication is provable:



(4.7)

which is a fact beyond the scope of formulas.

As in other deep inference systems, we can give for the rules in (4.4) their *duals*, or *corules*. In general, if

$$\mathsf{r}\,\frac{G}{H}$$

is an instance of a rule, then

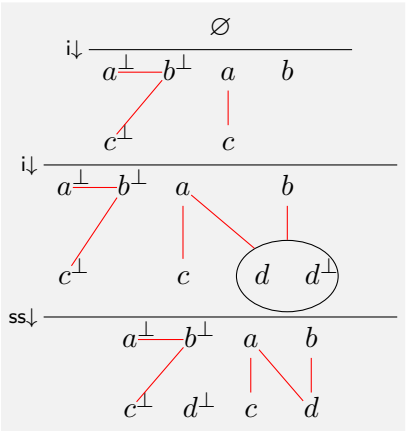
$$\mathsf{r}^\perp\,\frac{H^\perp}{G^\perp}$$

is an instance of the dual rule. The corules of the two rules in (4.4) are the following:

$$\mathsf{i}\uparrow\,\frac{A \otimes A^\perp}{\varnothing} \qquad\qquad \mathsf{ss}\downarrow\,\frac{B[A]_S}{B \parr A}\ {\scriptstyle S\subseteq V_B,\ S\neq\varnothing} \tag{4.8}$$

called ***identity up*** (or ***cut***) and ***super switch down***, respectively. We have the side condition $S \neq \varnothing$ to avoid a triviality, as $B[A]_S = B \parr A$ if $S = \varnothing$.

Example 4.8. The implication in (4.7) can also be proven using only only ss↓ and i↓ instead of ss↑ and i↓, as the following proof of length 3 shows:



(4.9)

Definition 4.9. Let S be an inference system. We say that an inference rule r is ***derivable*** in S iff

$$\text{for every instance } \mathsf{r}\frac{G}{H} \quad \text{there is a derivation } \begin{array}{c} G \\ \mathcal{D} \Vert \mathsf{S} \\ H \end{array} .$$

We say that r is ***admissible*** in S iff

$$\text{for every instance } \mathsf{r}\frac{G}{H} \quad \text{we have that } \vdash_{\mathsf{S}} G \text{ implies } \vdash_{\mathsf{S}} H \ .$$

If $\mathsf{r} \in \mathsf{S}$ then r is trivially derivable and admissible in S.

Most deep inference systems in the literature (e.g. [GS01, BT01, Gug07, Str02, GS02, HTAC19]) contain the ***switch*** rule:

$$\mathsf{s}\frac{(A \parr B) \otimes C}{A \parr (B \otimes C)} \tag{4.10}$$

On can immediately see that it is its own dual and is a special case of both $\mathsf{ss}{\downarrow}$ and $\mathsf{ss}{\uparrow}$. We therefore have the following:

Lemma 4.10. *If in an inference system* S *one of the rules* $\mathsf{ss}{\downarrow}$ *and* $\mathsf{ss}{\uparrow}$ *is derivable, then so is* s*.*

Remark 4.11. In a standard deep inference system for formulas we also have the converse of Lemma 4.10, i.e., if s is derivable, then so are $\mathsf{ss}{\uparrow}$ and $\mathsf{ss}{\downarrow}$ (see, e.g., [Str03]). However, in the case of arbitrary graphs this is no longer true, and the rules $\mathsf{ss}{\uparrow}$ and $\mathsf{ss}{\downarrow}$ are strictly more powerful than s.

Lemma 4.12. *Let* S *be an inference system. If the rules* $\mathsf{i}{\downarrow}$ *and* $\mathsf{i}{\uparrow}$ *and* s *are derivable in* S*, then for every rule* r *that is derivable in* S*, also its corule* $\mathsf{r}^{\perp}$ *is derivable in* S*.*

Proof. Suppose we have two graphs G and H, and a derivation from G to H in S. Then it suffices to show that we can construct a derivation from $H^{\perp}$ to $G^{\perp}$ in S:

$$\mathsf{s}\frac{\mathsf{i}{\downarrow}\dfrac{\varnothing}{G^{\perp} \parr G} \otimes H^{\perp}}{G^{\perp} \parr \mathsf{i}{\uparrow}\dfrac{\begin{array}{c} G \\ \Vert \mathsf{S} \\ H \end{array} \otimes H^{\perp}}{\varnothing}}$$

Note that $\varnothing \otimes H^{\perp} = H^{\perp}$ and $G^{\perp} \parr \varnothing = G^{\perp}$. □

Lemma 4.13. *If the rules* $\mathsf{i}{\uparrow}$ *and* s *are admissible for an inference system* S*, then* $\multimap$ *is transitive, i.e., if* $\vdash_{\mathsf{S}} G \multimap H$ *and* $\vdash_{\mathsf{S}} H \multimap K$ *then* $\vdash_{\mathsf{S}} G \multimap K$*.*

Proof. We can construct the following derivation

$$\mathsf{s}\,\frac{\dfrac{\varnothing}{\|\,\mathsf{s}}\;\otimes\;\dfrac{\varnothing}{\|\,\mathsf{s}}}{G^\perp \parr H \qquad\qquad H^\perp \parr K}\qquad\qquad \mathsf{s}\,\frac{H \otimes (H^\perp \parr K)}{G^\perp \parr \quad \mathsf{i}{\uparrow}\,\dfrac{H \otimes H^\perp}{\varnothing}\;\parr K}$$

from $\varnothing$ to $G^\perp \parr K$ in S. □

Lemma 4.13 is the reason why $\mathsf{i}{\uparrow}$ is also called *cut*. In a well-designed deep inference system for formulas, the two rules $\mathsf{i}{\downarrow}$ and $\mathsf{i}{\uparrow}$ can be restricted in a way that they are only applicable to atoms, i.e., replaced by the following two rules that we call ***atomic identity down*** and ***atomic identity up***, respectively:

$$\mathsf{ai}{\downarrow}\,\frac{\varnothing}{a^\perp \parr a} \qquad \text{and} \qquad \mathsf{ai}{\uparrow}\,\frac{a \otimes a^\perp}{\varnothing} \tag{4.11}$$

We would like to achieve something similar for our proof system on graphs. For this it is necessary to be able to decompose prime graphs into atoms, but the two rules $\mathsf{ss}{\downarrow}$ and $\mathsf{ss}{\uparrow}$ cannot do this, as they are only able to move around modules in a graph. For this reason, we add the following two rules to our system:

$$\mathsf{p}{\downarrow}\,\frac{(M_1 \parr N_1) \otimes \cdots \otimes (M_n \parr N_n)}{P(\!|M_1, \ldots, M_n|\!) \parr P^\perp(\!|N_1, \ldots, N_n|\!)}\; P \text{ prime, } |V_P|{\geq}4,\ M_1{\neq}\varnothing,\ldots,M_n{\neq}\varnothing \tag{4.12}$$

called ***prime down***, and

$$\mathsf{p}{\uparrow}\,\frac{P(\!|M_1, \ldots, M_n|\!) \otimes P^\perp(\!|N_1, \ldots, N_n|\!)}{(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)}\; P \text{ prime, } |V_P|{\geq}4\ M_1{\neq}\varnothing,\ldots,M_n{\neq}\varnothing \tag{4.13}$$

called ***prime up***. In both cases, the side condition is that P needs to be a prime graph and has at least 4 vertices. We also require that for all $i \in \{1, \ldots, n\}$ we have that M_i is non-empty in an application of $\mathsf{p}{\downarrow}$ and $\mathsf{p}{\uparrow}$.[6] The reason for the side conditions on the $\mathsf{p}{\downarrow}$ is not that the rules would become unsound otherwise. In fact, these rules without side conditions are admissible in the general case. The details will be discussed in the proofs of Lemmas 5.1 and at the end of Section 7.

[6] In the conference version [AHS20] of this paper, we had the weaker condition that for all $i \in \{1, \ldots, n\}$ at least one of M_i and N_i is non-empty. The advantage of the stronger condition that we use here is that we obtain a more controlled proof system as the prime graphs involved in a $\mathsf{p}{\downarrow}$ can only be those that appear in the modular decomposition of the graph.

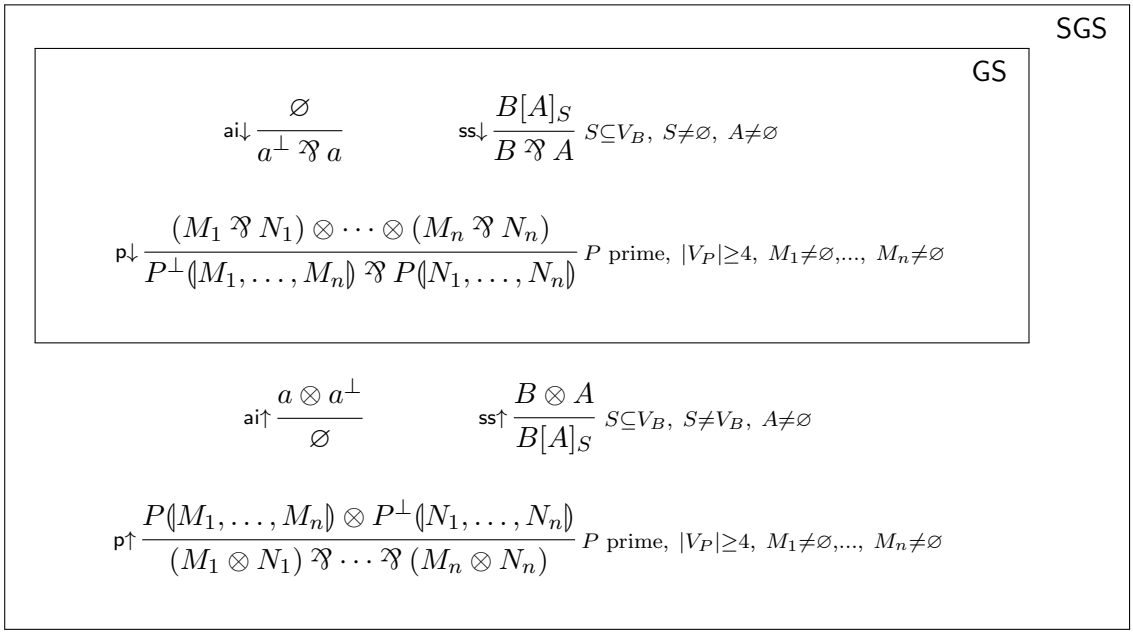
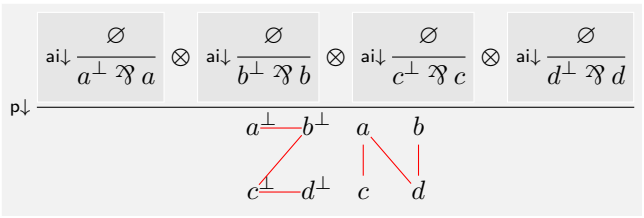


Figure 1: The inference rules for systems SGS and GS.

Example 4.14. Below is a derivation of length 5 using the p↓-rule, and proving that a prime graph implies itself.



This completes the presentation of our system, which is shown in Figure 1, and formally defined below.

Definition 4.15. We define ***system* SGS** to be the set $\{\mathsf{ai}\downarrow, \mathsf{ss}\downarrow, \mathsf{p}\downarrow, \mathsf{p}\uparrow, \mathsf{ss}\uparrow, \mathsf{ai}\uparrow\}$ of inference rules shown in Figure 1. The ***down-fragment*** (resp. ***up-fragment***) of SGS consists of the rules $\{\mathsf{ai}\downarrow, \mathsf{ss}\downarrow, \mathsf{p}\downarrow\}$ (resp. $\{\mathsf{ai}\uparrow, \mathsf{ss}\uparrow, \mathsf{p}\uparrow\}$) and is denoted by SGS↓ (resp. SGS↑). The down-fragment SGS↓ is also called ***system* GS**.

5. Properties of the Proof System

The first observation about SGS is that the general forms of the identity rules i↓ and i↑ are derivable, as we show in Lemma 5.2 below. Before, we show an auxiliary lemma stating a similar result for the prime rules, namely that their general form is also derivable, i.e., they can be applied to any graph instead of only prime graphs.

Lemma 5.1. *Let G and $M_1, \dots, M_n, N_1, \dots, N_n$ be graphs with $|V_G| = n$, such that for every $i \in \{1, \dots, n\}$, we have either $M_i \neq \varnothing$ or $M_i = \varnothing = N_i$. Then there are derivations*

$$\begin{array}{c} (M_1 \parr N_1) \otimes \cdots \otimes (M_n \parr N_n) \\ \| \, \mathsf{SGS}\downarrow \\ G(\!|M_1, \dots, M_n|\!) \parr G^\perp(\!|N_1, \dots, N_n|\!) \end{array} \tag{5.1}$$

and dually

$$\begin{array}{c} G(\!|M_1, \dots, M_n|\!) \otimes G^\perp(\!|N_1, \dots, N_n|\!) \\ \| \, \mathsf{SGS}\uparrow \\ (M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n) \end{array} \tag{5.2}$$

Proof. We only show (5.3), and proceed by induction on the size of $|V_G|$. First consider the case that there is an $i \in \{1, \dots, n\}$ such that $N_i = \varnothing = M_i$. Without loss of generality, we can assume that $i = 1$. Then there is a graph H with $|V_H| = n - 1$ and $G(\!|M_1, \dots, M_n|\!) = H(\!|M_2, \dots, M_n|\!)$ and $G^\perp(\!|N_1, \dots, N_n|\!) = H^\perp(\!|N_2, \dots, N_n|\!)$, and therefore we have the derivation

$$= \dfrac{\dfrac{(\varnothing \parr \varnothing) \otimes (M_2 \parr N_2) \otimes \cdots \otimes (M_n \parr N_n)}{(M_2 \parr N_2) \otimes \cdots \otimes (M_n \parr N_n)} \, \mathcal{D} \| \, \mathsf{SGS}\downarrow}{= \dfrac{H(\!|M_2, \dots, M_n|\!) \parr H^\perp(\!|N_2, \dots, N_n|\!)}{G(\!|\varnothing, M_2, \dots, M_n|\!) \parr G^\perp(\!|\varnothing, N_2, \dots, N_n|\!)}}$$

where $\mathcal{D}$ exists by induction hypothesis. We can therefore now assume that $M_i \neq \varnothing$ for all $i \in \{1, \dots, n\}$. We now make a case analysis on G using Lemma 3.9:

(i) If G is a singleton graph, the statement holds trivially.

(ii) If $G = A \otimes B$ then $G(\!|M_1, \dots, M_n|\!) = A(\!|M_1, \dots, M_k|\!) \otimes B(\!|M_{k+1}, \dots, M_n|\!)$ and $G^\perp(\!|N_1, \dots, N_n|\!) = A^\perp(\!|N_1, \dots, N_k|\!) \parr B^\perp(\!|N_{k+1}, \dots, N_n|\!)$ for some $1 \leq k \leq n$. We therefore have

$$\mathsf{ss}\downarrow \dfrac{\begin{array}{c}(M_1 \parr N_1) \otimes \cdots \otimes (M_k \parr N_k) \\ \mathcal{D}_1 \| \, \mathsf{SGS}\downarrow \\ A(\!|M_1, \dots, M_k|\!) \parr A^\perp(\!|N_1, \dots, N_k|\!)\end{array} \otimes \begin{array}{c}(M_{k+1} \parr N_{k+1}) \otimes \cdots \otimes (M_n \parr N_n) \\ \mathcal{D}_2 \| \, \mathsf{SGS}\downarrow \\ B(\!|M_{k+1}, \dots, M_n|\!) \parr B^\perp(\!|N_{k+1}, \dots, N_n|\!)\end{array}}{\mathsf{ss}\downarrow \dfrac{(A(\!|M_1, \dots, M_k|\!) \parr A^\perp(\!|N_1, \dots, N_k|\!)) \otimes B(\!|M_{k+1}, \dots, M_n|\!)}{(A(\!|M_1, \dots, M_k|\!) \otimes B(\!|M_{k+1}, \dots, M_n|\!)) \parr A^\perp(\!|N_1, \dots, N_k|\!)}} \parr B^\perp(\!|N_{k+1}, \dots, N_n|\!)$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ exist by induction hypothesis. Note that it is possible that $A^\perp(\!|N_1, \dots, N_k|\!) = \varnothing$ or $B^\perp(\!|N_{k+1}, \dots, N_n|\!) = \varnothing$ (or both). In that case one (or both) of the two instances of $\mathsf{ss}\downarrow$ is vacuous.

(iii) If $G = A \parr B$, we proceed similarly.

(iv) If $G = P(\!|A_1, \dots, A_k|\!)$ for P prime and $k = |V_P| \geq 4$ and $A_l \neq \varnothing$ for each $l \in \{1, \dots, k\}$, then we have that $G(\!|M_1, \dots, M_n|\!) = P(\!|A_1(\!|M_{11}, \dots, M_{1h_1}|\!), \dots, A_k(\!|M_{k1}, \dots, M_{kh_k}|\!)|\!)$ where $\{M_1, \dots, M_n\} = \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_k$ and $\mathcal{M}_l = \{M_{l1}, \dots, M_{lh_l}\}$ and $\mathcal{M}_l \cap \mathcal{M}_j = \varnothing$ for $l \neq j$. Similarly, we have $G^\perp(\!|N_1, \dots, N_n|\!) = P^\perp(\!|A_1^\perp(\!|N_{11}, \dots, N_{1h_1}|\!), \dots, A_k^\perp(\!|N_{k1}, \dots, N_{kh_k}|\!)|\!)$ where $\{N_1, \dots, N_n\} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_k$ and $\mathcal{N}_l = \{N_{l1}, \dots, N_{lh_l}\}$ and $\mathcal{N}_l \cap \mathcal{N}_j = \varnothing$ for $l \neq j$. Since $M_i \neq \varnothing$ for all $i \in \{1, \dots, n\}$,

we also have that for each $l \in \{1, \ldots, k\}$ that $A_l(\!|M_{l1}, \ldots, M_{lh_l}|\!) \neq \varnothing$. Therefore, we have the following derivation

$$\mathsf{p}\downarrow \frac{\begin{array}{c}(M_{11} \parr N_{11}) \otimes \cdots \otimes (M_{1h_1} \parr N_{1h_1}) \\ \mathcal{D}_1 \,\|\, \mathsf{SGS}\downarrow \\ A_1^\perp(\!|M_{11}, \ldots, M_{1h_1}|\!) \parr A_1(\!|N_{11}, \ldots, N_{1h_1}|\!)\end{array} \otimes \cdots \otimes \begin{array}{c}(M_{k1} \parr N_{k1}) \otimes \cdots \otimes (M_{kh_k} \parr N_{kh_k}) \\ \mathcal{D}_k \,\|\, \mathsf{SGS}\downarrow \\ A_k^\perp(\!|M_{k1}, \ldots, M_{kh_k}|\!) \parr A_k(\!|N_{k1}, \ldots, N_{kh_k}|\!)\end{array}}{P(\!|A_1(\!|M_{11}, \ldots, M_{1h_1}|\!), \ldots, A_k(\!|M_{k1}, \ldots, M_{kh_k}|\!)\!|) \parr P^\perp(\!|A_1^\perp(\!|N_{11}, \ldots, N_{1h_1}|\!), \ldots, A_k^\perp(\!|N_{k1}, \ldots, N_{kh_k}|\!)\!|)}$$

where $\mathcal{D}_1, \ldots, \mathcal{D}_k$ exist by induction hypothesis.

The derivation in (5.2) can be constructed dually. □

Lemma 5.2. *The rule* $\mathsf{i}\downarrow$ *is derivable in* $\mathsf{SGS}\downarrow$*, and dually, the rule* $\mathsf{i}\uparrow$ *is derivable in* $\mathsf{SGS}\uparrow$*.*

Proof. Each graph G, can be written as $G = G(\!|a_1, \ldots, a_n|\!)$ where $V_G = \{a_1, \ldots, a_n\}$. Since $a_1, \ldots, a_n$ are non-empty modules of G we can apply Lemma 5.1 to obtain the following derivation of $G \parr G^\perp$ from $\varnothing$ in $\mathsf{SGS}\downarrow$:

$$\begin{array}{c}\mathsf{ai}\downarrow \dfrac{\varnothing}{a_1 \parr a_1^\perp} \otimes \cdots \otimes \mathsf{ai}\downarrow \dfrac{\varnothing}{a_n \parr a_n^\perp} \\ \| \,\mathsf{GS} \text{ by Lemma 5.1} \\ G(\!|a_1, \ldots, a_n|\!) \parr G^\perp(\!|a_1^\perp, \ldots, a_n^\perp|\!)\end{array} \tag{5.3}$$

Dually, we can show that for any G there is a derivation from $G \otimes G^\perp$ to $\varnothing$ in $\mathsf{SGS}\uparrow$. □

Corollary 5.3. *If* $G, M_1, \ldots, M_n$ *are non-empty graphs, then there is a derivation*

$$\begin{array}{c}M_1 \otimes \cdots \otimes M_n \\ \| \,\mathsf{GS} \\ G(\!|M_1, \ldots, M_n|\!)\end{array}$$

Proof. This follows from the previous lemma, using $G^\perp(\!|\varnothing, \ldots, \varnothing|\!)$. □

Next, observe that Lemmas 4.12 and 4.13 hold for system SGS. In particular, we have that if $\vdash_{\mathsf{SGS}} A \multimap B$ and $\vdash_{\mathsf{SGS}} B \multimap C$ then $\vdash_{\mathsf{SGS}} A \multimap C$ because $\mathsf{i}\uparrow \in \mathsf{SGS}$. The main result of this paper is that Lemma 4.13 also holds for GS. More precisely, we have the following theorem:

Theorem 5.4 (Cut Admissibility)**.** *The rule* $\mathsf{i}\uparrow$ *is admissible for* GS*.*

To prove this theorem, we will show that the whole up-fragment of SGS is admissible for GS.

Theorem 5.5. *The rules* $\mathsf{ai}\uparrow$*,* $\mathsf{ss}\uparrow$*,* $\mathsf{p}\uparrow$ *are admissible for* GS*.*

Then Theorem 5.4 follows immediately from Theorem 5.5 and the second statement in Lemma 5.2.

The following three sections are devoted to the proof of Theorem 5.5. But before, let us finish this section by exhibiting some immediate consequences of Theorem 5.4.

Corollary 5.6. *For every graph* G*, we have* $\vdash_{\mathsf{SGS}} A$ *iff* $\vdash_{\mathsf{GS}} A$*.*

Corollary 5.7. *For all graphs G and H, we have*

$$\vdash_{\mathsf{GS}} G \multimap H \iff \begin{array}{c} \varnothing \\ \| \,\mathsf{GS} \\ G^{\perp} \parr H \end{array} \iff \begin{array}{c} \varnothing \\ \| \,\mathsf{SGS} \\ G^{\perp} \parr H \end{array} \iff \begin{array}{c} G \\ \| \,\mathsf{SGS} \\ H \end{array}$$

Proof. The first equivalence is just the definition of $\vdash$. The second equivalence follows from Theorem 5.5, and the last equivalence follows from the two derivations

$$\mathsf{i}{\downarrow}\dfrac{\varnothing}{G^{\perp} \parr \begin{array}{c} G \\ \| \\ H \end{array}} \quad \text{and} \quad \mathsf{ss}{\downarrow}\dfrac{G \otimes \begin{array}{c} \varnothing \\ \| \\ G^{\perp} \parr H \end{array}}{\mathsf{i}{\uparrow}\dfrac{G \otimes G^{\perp}}{\varnothing} \parr H}$$

together with Lemma 5.2. □

Corollary 5.8. *For all graphs G and H and all contexts $C[\cdot]_R$, we have that*

$$\vdash_{\mathsf{GS}} G \multimap H \quad \iff \quad \vdash_{\mathsf{GS}} C[G]_R \multimap C[H]_R \quad .$$

Proof. This is a consequence of Corollary 5.7 and Remark 4.6. But it can also be proved directly using Lemma 5.1. □

Corollary 5.9. *We have $\vdash A \otimes B$ iff $\vdash A$ and $\vdash B$.*

Proof. This follows immediately by inspecting the inference rules of GS. □

Corollary 5.10. *We have $\vdash P(\!|M_1, \ldots, M_n|\!)$ with P prime and $n \geq 4$ and $M_i \neq \varnothing$ for all $i = \{1, \ldots, n\}$, if and only if there is at least one $i = \{1, \ldots, n\}$ such that $\vdash M_i$ and $\vdash P(\!|M_1, \ldots, M_{i-1,}, \varnothing, M_{i+1}, \ldots, M_n|\!)$.*

Proof. As before, this follows immediately by inspecting the inference rules of GS. In fact, this can be seen as a generalization of the previous corollary. □

Corollary 5.11 (consistency)**.** *If $G \neq \varnothing$ and $\vdash_{\mathsf{GS}} G$, then $\nvdash_{\mathsf{GS}} G^{\perp}$.*

Proof. Since $G \neq \varnothing$ and $\vdash_{\mathsf{GS}} G$, for some a we have $\begin{array}{c} \mathsf{ai}{\downarrow}\dfrac{\varnothing}{a^{\perp} \parr a} \\ \| \,\mathsf{GS} \\ G \end{array}$. Hence, by Corollary 5.7 we have $\vdash_{\mathsf{GS}} G^{\perp} \multimap a \otimes a^{\perp}$. Thus, if we assume for contradiction that $\vdash_{\mathsf{GS}} G^{\perp}$ holds then by Theorem 5.4 we have $\vdash_{\mathsf{GS}} a \otimes a^{\perp}$, which is impossible. □

Observation 5.12. The system GS forms a proof system in the sense of Cook and Reckhow [CR79], as the time complexity of checking the correct application of inference rules is polynomial: the modular decomposition of graphs can be obtained in linear time [MS94], and whenever a graph isomorphism is used to compose derivations, as in (4.5), we assume that the isomorphism f is explicitly given.

Let us now define the notion of *size* of a graph that will play a central role in the normalization proof as an induction measure.

Definition 5.13. The *size* of a graph G, denoted by $\|G\|$, is the lexicographic pair $\langle |V_G|, |E_{G^\perp}| \rangle$. That is, if G and H are graphs, then $\|G\| < \|H\|$ if $|V_G| < |V_H|$ or if $|V_G| = |V_H|$ and $|E_{G^\perp}| < |E_{H^\perp}|$.

Observation 5.14. The first observation about the size of a graph is that every inference rule discussed so far, except for $\mathsf{ai}{\uparrow}$ and $\mathsf{i}{\uparrow}$ do strictly decrease the size of a graph when going bottom-up in a derivation. That is, whenever we have an instance $\mathsf{r}\dfrac{H}{G}$ of an inference rule $\mathsf{r} \in \{\mathsf{ai}{\downarrow}, \mathsf{i}{\downarrow}, \mathsf{ss}{\downarrow}, \mathsf{ss}{\uparrow}, \mathsf{p}{\downarrow}, \mathsf{p}{\uparrow}, \mathsf{g}{\downarrow}, \mathsf{g}{\uparrow}\}$, then $\|H\| < \|G\|$. This means that the length of a derivation in GS and conclusion G is bound by n^3 where $n = |V_G|$ is the number of vertices in G.[7]

This observation is used to prove the following theorem.

Theorem 5.15. *Provability in* GS *is decidable and in* **NP**.

Proof. Decidability follows from the fact that to each graph only finitely many inference rules can be applied, and by Observation 5.14 the length of a derivation in GS is $O(n^3)$ where n is the number of vertices in the conclusion. This also entails membership in **NP**. □

6. Splitting and Context Reduction

The standard syntactic method for proving cut elimination in the sequent calculus is to permute the cut rule upwards in the proof, while decomposing the cut formula along its main connective, and so inductively reduce the cut rank. However, in systems formulated using deep inference this method cannot be applied, as derivations can be constructed in a more flexible way than in the sequent calculus. For this reason, the *splitting* technique has been developed in the literature on deep inference [Gug07, Str03, GS11, HTAC19, AT17].

From the sequent calculus to splitting for graphs. Consider for example the typical rule for $\otimes$ in the sequent calculus.

$$\otimes \frac{\Gamma, \phi \quad \Delta, \psi}{\Gamma, \phi \otimes \psi, \Delta}$$

The effect of the above rule can be simulated by applying the following splitting lemma for the prime graph $\otimes$.

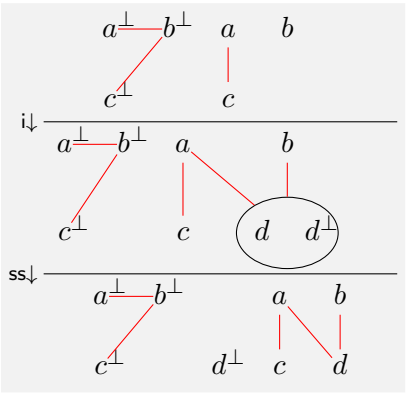
Lemma 6.1 (Splitting Tensor)**.** *Let G, A and B be graphs. If $\vdash_{\mathsf{GS}} G \parr (A \otimes B)$ then there is a context C and there are graphs K_A and K_B such that there are derivations*

$$\begin{array}{c} C[K_A \parr K_B]_R \\ \mathcal{D}_G \,\|\, \mathsf{GS} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_A \,\|\, \mathsf{GS} \\ K_A \parr A \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_B \,\|\, \mathsf{GS} \\ K_B \parr B \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array} .$$

[7] For the case that the derivation contains only cographs, it can in fact be shown that the length is bound by $\frac{1}{2}n^2$ (see [BS17]), but as the details are not needed for this paper, we leave them to the reader.

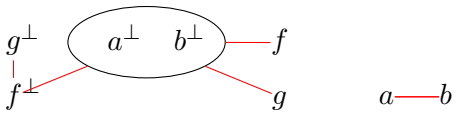
The graphs A and B in Lemma 6.1 play the role of formulas ϕ and ψ in the sequent calculus rule $\otimes$; while K_A and K_B play the role of sequents Γ and Δ. Thus when we say that $\vdash_{\mathsf{GS}} K_A \parr A$ and $\vdash_{\mathsf{GS}} K_B \parr B$ hold in Lemma 6.1 this corresponds to the provability of the premises Γ, ϕ and Δ, ψ in the $\otimes$ sequent calculus rule. This means that Lemma 6.1 allows us to rewrite a *sequent-like graph* $G \parr (A \otimes B)$ into a form where the $\otimes$ rule of the effect of the sequent calculus can be simulated.

We will use the proof in (4.9) as an example to illustrate this pattern. Observe that in that proof the conclusion is of the form $G \parr ((a^\perp \parr c^\perp) \otimes b^\perp)$ for some graph G, and that after applying $\mathsf{ss}\downarrow$ and $\mathsf{ai}\downarrow$ we have split the graph G into two graph $K_A = a \otimes c$ and $K_B = b$, such that $K_A \parr a^\perp \parr c^\perp$ and $K_B \parr b^\perp$ are provable, as recalled below for convenience.



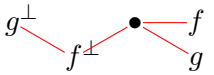
Thus applying Lemma 6.1 to the sub-graph $(a^\perp \parr c^\perp) \otimes b^\perp$ above leads us to proof (4.9).

There is a fundamental difference between the statement of Lemma 6.1 and other splitting lemmas in the in the literature on deep inference [Gug07, Str03, GS11, HTAC19, AT17], namely the need of the context C. To see that this context is necessary, consider the following example graph:



(6.1)

which is of the form $G \parr (a \otimes b)$ for some G. Furthermore, it is provable in GS. It is impossible however to apply a derivation to G to obtain two disjoint graphs K_A and K_B, such that $K_A \parr a$ and $K_B \parr b$ are provable, unless we define a suitable context for $K_A \parr K_B$. Indeed we do have the following context formed from the the provable graph $C = g^\perp \parr f^\perp \parr (g \otimes f)$.



Following the naming convention in the statement of Lemma 6.1, we have $K_A = a^\perp$ and $K_A = b^\perp$ such that $\vdash_{\mathsf{GS}} K_A \parr a$ and $\vdash_{\mathsf{GS}} K_B \parr b$ hold, and furthermore G is formed from plugging $K_A \parr K_B$ insider the above mentioned context formed from C.

From the above ingredients satisfying the condition of the splitting lemma we can always construct a proof. We can apply $\mathsf{ss}\downarrow$ twice to bring K_A and a together inside a context formed from C, and similarly for K_B and b. After having completed the proofs that consume a and b (by a simple application of $\mathsf{ai}\downarrow$) we are left with the provable graph C. The important observation here is that it is impossible to apply a derivation that changes the shape of C until we have completed the derrivation that consumes $a^\perp$ and $b^\perp$.

The problem illustrated above with the context that cannot be removed until the end of the proof is a fundamental problem that we have with graphs that do not happen for formulae. Essentially it is due to the presence of P_4 induced sub-graphs that create indirect dependencies between atoms, such as the indirect dependency between $g^\perp$ and g in the above example.

Generalising splitting to prime graphs. The general idea of a splitting lemma is that, in a provable "sequent-like graph", consisting of a number of disjoint connected components, we can select any of these components as the *principal graph* and apply a derivation to the other components, such that eventually a rule breaking down the principal component can be applied. This allows us to approximate the effect of applying rules in the sequent calculus, even when no sequent calculus exists, as is the case when the principal graph is formed using a prime graph that is not a tensor.

Lemma 6.2 (Splitting Prime)**.** *Let G be a graph and $P \neq \parr$ be a prime graph with $|V_P| = n$, and $M_1, \ldots, M_n$ be non-empty graphs. If $\vdash_{\mathsf{GS}} G \parr P(\!|M_1, \ldots, M_n|\!)$, then one of the following holds:*

(A) *either there is a context $C[\cdot]_R$ and graphs K_1, …, K_n, such that there are derivations*

$$\begin{array}{c} C[P^\perp(\!|K_1, \ldots, K_n|\!)]_R \\ \mathcal{D}_G \,\|\, \mathsf{GS} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array}$$

for all $i \in \{1, \ldots, n\}$,

(B) *or there is a context $C[\cdot]_R$ and graphs K_X and K_Y, such that there are derivations*

$$\begin{array}{c} C[K_X \parr K_Y]_R \\ \mathcal{D}_G \,\|\, \mathsf{GS} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_i \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|M_1, \ldots, M_{i-1}, \varnothing, M_{i+1}, \ldots, M_n|\!) \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array}$$

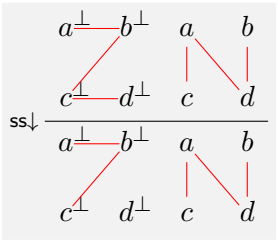
for some $i \in \{1, \ldots, n\}$.

The immediate question to address at this point why we need two cases, where splitting lemmas in the literature require only one case. Notice that that Lemma 6.1 is a special case of Lemma (6.2), since $\otimes$ is a prime graph, as indicated in (3.3) in Section 3. In the case of the tensor, the two sub-cases of Lemma (6.2) collapse; hence for $\otimes$ the above lemma for prime graphs collapses to the form of Lemma 6.1.

When the principal graph is formed using a prime graph with at least four vertices, the two cases of Lemma 6.1 are properly distinct. We illustrate first the case (A) of Lemma 6.1, by renormalising the proof (4.9) as an example to explain this idea.

Observe that in the conclusion of the proof (4.9) we have the disjoint union of three connected graphs. We can select the N-shape component on the right as the principal component, and apply Case (A) of Lemma 6.2 to reorganise the proof (4.9) such that an instance of $\mathsf{p}{\downarrow}$ involving the N-shape can be applied, as in Example 4.14. The bottommost

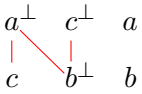
step of such a reorganised proof is shown below:



(6.2)

Clearly now we have that Case (A) applies, since all of four graphs $a^\perp \parr a$, $b^\perp \parr b$, $c^\perp \parr c$ and $d^\perp \parr d$ are provable. This idea is that, having reorganised the proof such that a prime graph and its complement can appear next to each other, we are in a position to apply the $\mathsf{p}\downarrow$. This case is clearly in the spirit of splitting, since we were able to select the N-shaped prime graph as the principal graph and then apply a derivation to reach a form where an instance of the $\mathsf{p}\downarrow$ rule completely consumes that principal graph, reducing it to sub proofs.

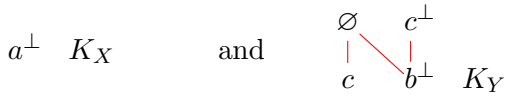
Case (B) of Lemma 6.2 effectively destroys a prime graph by killing one of its modules, thereby breaking it down into a smaller graph. This case is required to apply splitting to graphs such as the example below.



(6.3)

Firstly, observe that Case (A) cannot be applied to (6.3), since if we select the N-shape as the principal component and try to apply $\mathsf{p}\downarrow$, then c and $c^\perp$ can no longer communicate. Therefore, we must first move b or c into the structure and apply an $\mathsf{ai}\downarrow$, in order to destroy the prime graph such that c and $c^\perp$ have no indirect dependencies. For example, by using a to cancel out $a^\perp$, we obtain a provable graph of the form $c \parr (b^\perp \otimes c^\perp) \parr b$.

To be explicit, using the naming convention in Lemma 6.2 we have graphs $K_X = a$ and $K_Y = b$ such that the following two graphs are provable, as required for splitting.



(6.4)

Observe that we can reassemble the proofs of the above two graphs, using $\mathsf{ss}\downarrow$ to obtain a proof of the graph in (6.3) above.

In summary, it is always the case that either Case (A) or Case (B) hold for graphs formed using prime graphs. The two cases collapse only in the case where the prime graph is the tensor, hence both cases can bee seen as generalisations of splitting for tensor. The existence of these two separate cases is a surprising novelty in the move from proof system on formulas to our graphical system GS.

Splitting for Atoms. Finally, we also need a splliting lemma where the principal graphs is just a singleton.

Lemma 6.3 (Atomic Splitting)**.** *Let G be a graph. If $\vdash_{\mathsf{GS}} G \parr a$ for an atom a, then there is a context $C[\cdot]_R$ such that there are derivations*

$$\begin{array}{c} C[a^\perp]_R \\ \mathcal{D}_G \,\|\, \mathsf{GS} \\ G \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array} \quad .$$

Atomic splitting simply states that for an atom a in the proof, there is a matching $a^\perp$ somewhere else in the proof. Note there may be more than one $a^\perp$, in which case the lemma will pick out exactly the instance that interacts with a via an the $\mathsf{ai}{\downarrow}$ rule.

As with splitting for prime graphs the novelty compared to the literature is the presence of context, although this in fact simplifies matters since it allows the derivation in $\mathcal{D}_G$ to be always empty. However, its proof will be more in tradition with respect to the literature, where $\mathcal{D}_G$ is not necessarily empty.

Context reduction. Observe that the splitting lemma is not applied to a $\parr$-graph, as the $\parr$ plays a similar role as the multiset divider in the sequent calculus. Splitting lemmas such Lemma 6.2 apply in a sequent-like context, sometimes called a *shallow context*. Splitting cannot be applied to proofs where the principal graph (that is not a $\parr$) appears at any depth in the conclusion of the proof. In order, to use splitting to eliminate cuts in any context, we need to extend splitting to contexts of arbitrary depth. For this, we need the context reduction lemma, stated below.

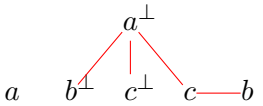
Lemma 6.4 (Context reduction)**.** *Let A be a graph and $G[\cdot]_S$ be a context, such that $\vdash_{\mathsf{GS}} G[A]_S$. Then there are a context $C[\cdot]_R$ and a graph K, such that there are derivations*

$$\begin{array}{c} C[K \parr X]_R \\ \mathcal{D}_G \,\|\, \mathsf{GS} \\ G[X]_S \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_A \,\|\, \mathsf{GS} \\ K \parr A \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array}$$

for any graph X.

The idea of context reduction is that, if we select any module, called A in Lemma 6.4 above, we can always rewrite its context, say $G[\cdot]_S$, such that it is of the form $K \parr [\cdot]_\varnothing$ inside some provable context. The use of X reflects the fact that, in doing so, you never need to change A; hence A could be replaced by another graph X and the same transformation can be applied to the context.

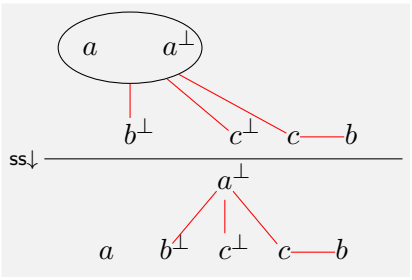
Similarly to splitting, the novelty compared to context reduction in the literature is that we must keep track of a provable context. This is required to cope with graphs such as the following:



(6.5)

Suppose that we are interested in the singleton graph $a^\perp$ in the above formula and we wish to apply context reduction to it. When we apply Lemma 6.4, clearly there is only one choice of graph K such that $K \parr a^\perp$ is provable, namely $K = a$. Observe, furthermore, that in

order to rewrite the rest of the above graph, so it is of the form $C[K \parr a^\perp]_R$ such that C is provable, the only choice is to set $C = b^\perp \parr c^\perp \parr (b \otimes c)$, as done in the following derivation:



(6.6)

which can be completed to a proof, by applying ai↓ to the module $a \parr a^\perp$ and then i↓ to prove the resulting graph C. Examples such as (6.1), (6.3) and (6.6) where each conclusion can only be proven if rules can be applied deep inside a graph instead of to a graph at the top level of the modular decomposition of the conclusion, show that deep inference is necessary for developing a proof theory on graphs allowing P_4 as an induced sub-graph.

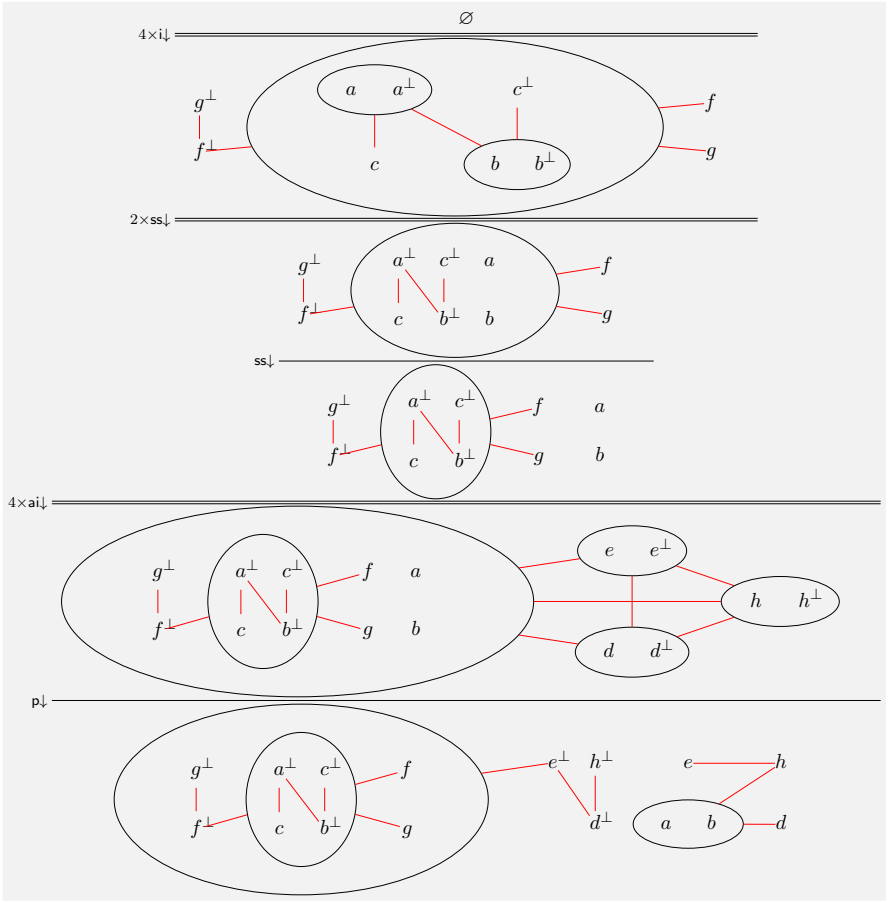


Figure 2: A proof used to illustrate context reduction and splitting. It also shows a non-trivial use of the p↓ rule that cannot be achieved using the i↓ rule.

For a more substantial example consider the derivation in Figure 2. Consider, in particular, when we aim to apply splitting to the N-shaped induced sub-graph in the conclusion consisting of vertices labelled c, $a^\perp$, $b^\perp$ and $c^\perp$. Clearly, the splitting lemma cannot be directly applied to that induced sub-graph, so firstly we reduce the context. Three bottommost inferences shown in Figure 2 correspond to the steps taken by context reduction that do not touch the N-shaped sub-graph we are interested in but does bring the vertices labelled a and b next to our subgraph. Indeed, at that point Case (B) of splitting for prime graphs can be applied to our N-shaped graph, taking into account that, together with the vertices labelled a and b, they form a sequent-like module of the graph at that point in the derivation. At this point, we splitting provides us with the information needed to complete the proof, much like in example 6.3.

An additional reason why Figure 2 is interesting is that it is an example of a proof where the effect of the $\mathsf{p}\downarrow$ rule cannot be simulated by using the $\mathsf{i}\downarrow$ rule. Therefore, this example emphasises the need for the $\mathsf{p}\downarrow$ or at least a rule to the effect of the $\mathsf{p}\downarrow$ in an analytic proof system on graphs.

Proving splitting and context reduction. We now proceed with proving the lemmas discussed in this section so far. As standard, we proceed by exhausting all possible permutations of rules where the main points of interest are where the rules change the principal connective in some way. The novelty compared to results in the literature on deep inference is that, due to the important role of contexts we prove splitting and context reduction together by mutual induction, using the size $\|G\|$ of a graph (see Definition 5.13) as induction measure. The mutual dependence of the proofs of our lemmas is depicted in Figure 3.

During the proof of splitting and context reduction we must handle cases involving the $\mathsf{p}\downarrow$ rule. Observe that when the $\mathsf{p}\downarrow$ rule is applied we obtain at least four graphs connected by tensors. Since we will need to handle frequently such cases, we find it convenient to state splitting for such multi-tensors as a lemma. In what follows we generalise the Lemma 6.1 to to n-ary tensor for any $n \geq 2$ (or *multi-tensor*), which is clearly a consequence of splitting for tensor (which, in turn, we have discussed is a special case of splitting for prime graphs).

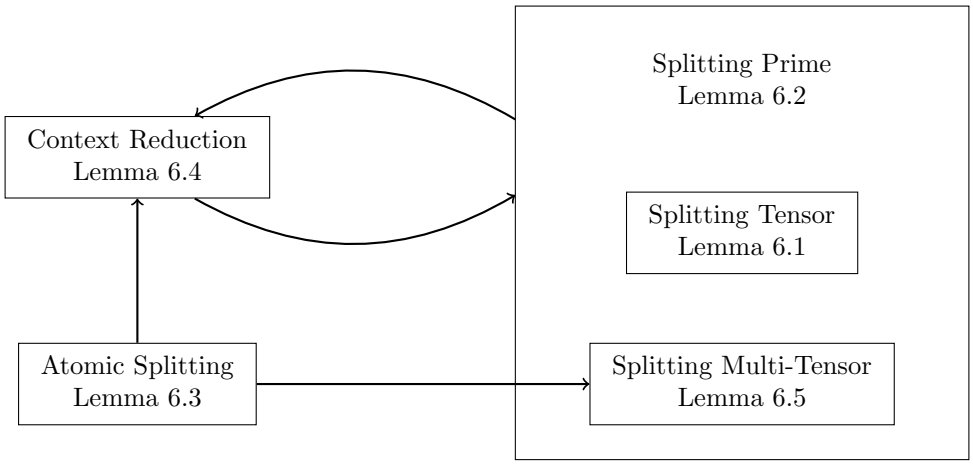


Figure 3: Dependencies between splitting lemmas and context reduction.

Lemma 6.5 (Splitting Multi-tensor)**.** *Let G be a graph, and $M_1, \ldots, M_n$ be non-empty graphs. If $\vdash_{\mathsf{GS}} G \parr (M_1 \otimes \cdots \otimes M_n)$, then there is a context $C[\cdot]_R$ and graphs K_1, …, K_n, such that there are derivations*

$$\dfrac{C[K_1 \parr \cdots \parr K_n]_R}{\mathcal{D}_G \,\|\, \mathsf{GS}\\ G}\ , \quad \dfrac{\varnothing}{\mathcal{D}_i \,\|\, \mathsf{GS}\\ K_i \parr M_i} \quad \textit{and} \quad \dfrac{\varnothing}{\mathcal{D}_C \,\|\, \mathsf{GS}\\ C}$$

for all $i \in \{1, \ldots, n\}$,

Proof. By induction on n. If $n = 2$, then we conclude by Lemma 6.1. If $n \geq 3$, then $M_1 \otimes \cdots \otimes M_n = M_1 \otimes (M_2 \otimes M_3 \otimes \cdots \otimes M_n)$. Then, by Lemma 6.1, there are a context $C'[\cdot]_S$, and graphs K_1 and K_Y such that there are derivations

$$\dfrac{C'[K_1 \parr K_Y]_S}{\mathcal{D}'_G \,\|\, \mathsf{GS}\\ G}\ , \quad \dfrac{\varnothing}{\mathcal{D}_1 \,\|\, \mathsf{GS}\\ K_1 \parr M_1}\ , \quad \dfrac{\varnothing}{\mathcal{D}_Y \,\|\, \mathsf{GS}\\ K_Y \parr (M_2 \otimes M_3 \otimes \cdots \otimes M_n)} \quad \text{and} \quad \dfrac{\varnothing}{\mathcal{D}'_C \,\|\, \mathsf{GS}\\ C'}\ .$$

By applying the same lemma to $K_Y \parr (M_2 \otimes M_3 \otimes \cdots \otimes M_n)$ and by inductive hypothesis on $\mathcal{D}_Y$ there is a context $C_Y[\cdot]_{S'}$ and there are graphs $K_2, \ldots, K_n$ such that there are derivations

$$\dfrac{C_Y[K_2 \parr \cdots \parr K_n]_{S'}}{\mathcal{D}'_Y \,\|\, \mathsf{GS}\\ K_Y}\ , \quad \dfrac{\varnothing}{\mathcal{D}_i \,\|\, \mathsf{GS}\\ K_i \parr M_i} \quad \text{and} \quad \dfrac{\varnothing}{\mathcal{D}_{C_Y} \,\|\, \mathsf{GS}\\ C_Y}$$

for all $i \in \{2, \ldots, n\}$. We conclude by letting $C[\cdot]_R = C'[C_Y[\cdot]_{S'}]_S$ and $\mathcal{D}_G$ be the derivation defined as

$$C'\left[\dfrac{\mathsf{ss}\downarrow\dfrac{C_Y[K_1 \parr \cdots \parr K_n]_{S'}}{K_1 \parr \dfrac{C_Y[K_2 \parr \cdots \parr K_n]_{S'}}{\mathcal{D}'_Y \,\|\\ K_Y}}}{\mathcal{D}'_G \,\|\\ G}\right]_S$$

□

Before tackling the main proof, there are a few relatively simple observations that we appeal to in various places in the splitting proof, to make it easier to read. In the cases that follow proofs are composed and the induction hypothesis is applied inductively.

Lemma 6.6. *Let $C_1[\cdot]_{R_1}, \ldots, C_n[\cdot]_{R_n}$ be contexts. If $\vdash_{\mathsf{GS}} C_i$ for all $i \in \{1, \ldots, n\}$, then $\vdash_{\mathsf{GS}} C_1[C_2[\ldots C_n[\cdot]_{R_n}]_{R_2}]_{R_1}$.*

Proof. We proceed by induction on n. The base case for $n = 1$ is trivial, and the inductive case for $n > 1$ is this derivation:

$$C_1\left[\begin{array}{c}\varnothing\\ \mathcal{D}_1\,\|\\ C_2[\dots C_n[\cdot]_{R_n}]_{R_2}\end{array}\right]_{R_1}\quad\text{,}$$

where $\mathcal{D}_n$ is the derivation for C_n and $\mathcal{D}'$ exists by induction hypothesis. □

Since the multi-tensor Lemma 6.5 is used as lemma inside splitting, we in fact must prove a stronger lemma. Namely that if Lemma 6.2 holds for graphs of a bounded size, then Lemma 6.1 and hence Lemma 6.5 holds for graphs of the same bounded size. Since this is a direct proof, we make this assumption implicitly, that is, in the following proof, where it is enough to remark that whenever we appeal to Lemma 6.2 we do so on a graph H such that $\|H\| \leq \|G \parr (M_1 \otimes \cdots \otimes M_n)\|$.

There is a mutual induction between context reduction and splitting for prime graphs. Thus the induction hypothesis in this proof assumes that both Lemma 6.4 and Lemma 6.2 hold for graphs H such that $\|H\| < \|G[A]_S\|$.

Proof of Context Reduction (Lemma 6.4). The case when $A = \varnothing$ is trivial, since we can take $C = G$ and $R = S$ and $K = \varnothing$.

Otherwise, without loss of generality $G[A]_S = G'' \parr G'[A]_S$ for a graph $G'[A]_S$ which is neither a par nor empty. The base case is where $G' = \varnothing$ and $S = \varnothing$, in which case we can set $K = G''$ and $C = \varnothing$, and the derivations $\mathcal{D}_C$ and $\mathcal{D}_G$ to be trivial. The derivation $\mathcal{D}_A$ is given by the proof of $G[A]_S$, since under these assumptions $K \parr A = G'' \parr A = G[A]_S$.

If G' is non-empty, we proceed by induction on the size of $G'[A]_S$ as follows. $G'[A]_S$ must be composed via a prime graph P with $P \neq \parr$. Then, without loss of generality we can assume that $G[A]_S = G'' \parr P(\!|M_1[A]_{S'}, M_2, \dots, M_n|\!)$. Applying Lemma 6.2 gives us one of the following three cases:

(A) We have $C'[\cdot]_{R'}$ and $K_1, \dots, K_n$, such that

$$\begin{array}{c}C'[P^\perp(\!|K_1,\dots,K_n|\!)]_{R'}\\ \mathcal{D}''_G\,\|\,\mathsf{GS}\\ G''\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}_1\,\|\,\mathsf{GS}\\ K_1 \parr M_1[A]_{S'}\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}_i\,\|\,\mathsf{GS}\\ K_i \parr M_i\end{array}\quad\text{and}\quad\begin{array}{c}\varnothing\\ \mathcal{D}'_C\,\|\,\mathsf{GS}\\ C'\end{array}$$

for $2 \leq i \leq n$. We can apply the induction hypothesis to $K_1 \parr M_1[A]_{S'}$ and obtain K and $C''[\cdot]_{R''}$, such that

$$\begin{array}{c}\varnothing\\ \mathcal{D}''_C\,\|\,\mathsf{GS}\\ C''\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}_A\,\|\,\mathsf{GS}\\ K \parr A\end{array}\quad\text{and}\quad\begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}'_G\,\|\,\mathsf{GS}\\ K_1 \parr M_1[X]_{S'}\end{array}$$

for any X. We let $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$, the derivation $\mathcal{D}_C$ is defined by Lemma 6.6, and the derivation $\mathcal{D}_G$ is defined as

$$C'\left[\dfrac{\begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}'_G \Vert\\ K_1 \parr M_1[X]_{S'}\end{array} \otimes \begin{array}{c}\varnothing\\ \mathcal{D}_2 \Vert\\ K_2 \parr M_2\end{array} \otimes \cdots \otimes \begin{array}{c}\varnothing\\ \mathcal{D}_n \Vert\\ K_n \parr M_m\end{array}}{\mathsf{p}\downarrow\ P^\perp(\!|K_1,\ldots,K_n|\!) \parr K_1 \parr P(\!|M_1[X]_{S'},M_2,\ldots,M_n|\!)}\right]_{R'}$$
$$\mathsf{ss}\downarrow\ \dfrac{}{\begin{array}{c}C'[P^\perp(\!|K_1,\ldots,K_n|\!)]_{R'}\\ \mathcal{D}''_G \Vert\\ G''\end{array} \parr P(\!|M_1[X]_{S'},M_2,\ldots,M_n|\!)}\ ,$$

where $G[X]_S = G'' \parr P(\!|M_1[X]_{S'},M_2,\ldots,M_n|\!)$.

(B) We have $C'[\cdot]_{R'}$ and K_X and K_Y, such that

$$\begin{array}{c}C'[K_X \parr K_Y]_{R'}\\ \mathcal{D}''_G \Vert \mathsf{GS}\\ G''\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_X \Vert \mathsf{GS}\\ K_X \parr M_1[A]_{S'}\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_Y \Vert \mathsf{GS}\\ K_Y \parr P(\!|\varnothing,M_2,\ldots,M_n|\!)\end{array}\quad\text{and}\quad \begin{array}{c}\varnothing\\ \mathcal{D}'_C \Vert\\ C'\end{array}\ .$$

We apply the induction hypothesis to $K_X \parr M_1[A]_{S'}$ and get K and $C''[\cdot]_{R''}$, such that

$$\begin{array}{c}\varnothing\\ \mathcal{D}''_C \Vert \mathsf{GS}\\ C''\end{array}\ ,\quad \begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}'_G \Vert \mathsf{GS}\\ K_X \parr M_1[X]_{S'}\end{array}\quad\text{and}\quad \begin{array}{c}\varnothing\\ \mathcal{D}_A \Vert \mathsf{GS}\\ K \parr A\end{array}$$

for any X. We let $C[\cdot]_R = C'[K_Y \parr P(\!|C''[\cdot]_{R''},M_2,\ldots,M_n|\!)]_{R'}$ and obtain $\mathcal{D}_C$ from Lemma 6.6, and $\mathcal{D}_G$ is as follows:

$$C'\left[K_Y \parr P(\!|\begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}'_G \Vert\\ K_X \parr M_1[X]_{S'}\end{array},M_2,\ldots,M_n|\!)\right]_{R'}$$
$$\mathsf{ss}\downarrow\ \dfrac{}{\begin{array}{c}C'[K_X \parr K_Y]_{R'}\\ \mathcal{D}''_G \Vert\\ G''\end{array} \parr P(\!|M_1[X]_{S'},M_2,\ldots,M_n|\!)}\ .$$

(C) We have $C'[\cdot]_{R'}$ and K_X and K_Y, such that

$$\begin{array}{c}C'[K_X \parr K_Y]_{R'}\\ \mathcal{D}''_G \Vert \mathsf{GS}\\ G''\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_X \Vert \mathsf{GS}\\ K_X \parr M_2\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_Y \Vert \mathsf{GS}\\ K_Y \parr P(\!|M_1[A]_{S'},\varnothing,M_3,\ldots,M_n|\!)\end{array}\quad\text{and}\quad \begin{array}{c}\varnothing\\ \mathcal{D}'_C \Vert \mathsf{GS}\\ C'\end{array}\ .$$

We apply the induction hypothesis to $K_Y \parr P(\!|M_1[A]_{S'},\varnothing,M_3,\ldots,M_n|\!)$ and get K and $C''[\cdot]_{R''}$, such that

$$\begin{array}{c}\varnothing\\ \mathcal{D}''_C \Vert \mathsf{GS}\\ C''\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_A \Vert \mathsf{GS}\\ K \parr A\end{array}\quad\text{and}\quad \begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}''_G \Vert \mathsf{GS}\\ K_Y \parr P(\!|M_1[X]_{S'},\varnothing,M_3,\ldots,M_n|\!)\end{array}$$

for any X. We let $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$, the derivation $\mathcal{D}_C$ be defined by Lemma 6.6 and the derivation $\mathcal{D}_G$ defined as follows

$$\mathsf{ss}\downarrow\dfrac{C'\left[\mathsf{ss}\downarrow\dfrac{K_Y \parr P(\!|M_1[X]_{S'}, \begin{array}{c}C''[K \parr X]_{R''}\\ \mathcal{D}'_G \Big\| \\ \varnothing \\ \mathcal{D}_X \Big\| \\ K_X \parr M_2\end{array}, M_3, \ldots, M_n|\!)}{K_X \parr K_Y \parr P(\!|M_1[X]_{S'}, M_2, M_3, \ldots, M_n|\!)}\right]_{R'}}{\begin{array}{c}C'[K_X \parr K_Y]_{R'}\\ \mathcal{D}''_G \Big\| \\ G''\end{array} \quad \parr P(\!|M_1[X]_{S'}, M_2, M_3, \ldots, M_n|\!)}\,.$$

□

We draw attention again to the mutual induction between context reduction and splitting for prime graphs. Thus the induction hypothesis in the proof below assumes that both Lemma 6.4 and Lemma 6.2 hold for graphs H such that $\|H\| < \|G[A]_S\|$. This is appealed to implicitly throughout this proof. Also, when we appeal to Lemma 6.5 we mean a formulation of that lemma where the size of the graph is bounded, as discussed next to the proof of Lemma 6.5 above. Thus the proofs in this section are not stand alone proofs, but are proven simultaneously by induction. The dependencies between the cases of each of the proofs are illustrated in Figure 4.

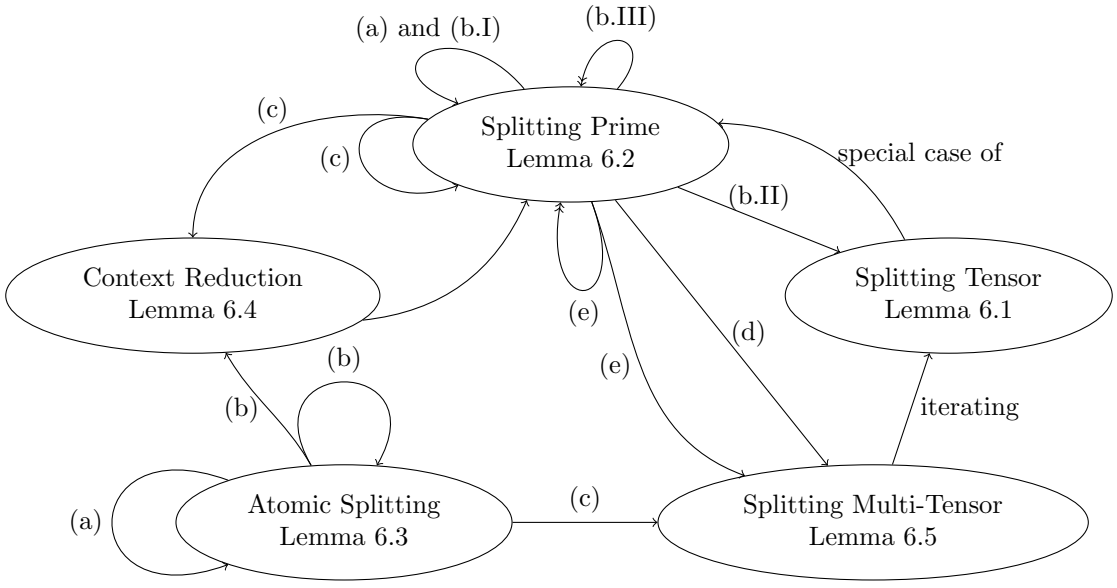


Figure 4: Complete roadmap of splitting and context reduction lemmas proofs. Double head arrows make use of Lemma 5.1.

In the proof of Lemma 6.2 there are five cases to consider which we first enumerate below before proceeding with the proof.

(a) It is possible that the last rule in a proof acts inside G or any M_i with $i \in \{1, \dots, n\}$. This case is relatively simple, since the structure of the conclusion does not change.

(b) It is possible that the last rule in a proof is a $\mathsf{ss}\downarrow$ such that another graph, external to the principal prime graph, moves inside the principal prime graph. $G = G' \parr G''$ with $G' \neq \varnothing$ and $\mathcal{D}$ is of shape

$$\mathsf{ss}\downarrow \frac{\begin{array}{c}\varnothing \\ \mathcal{D}'' \parallel \mathsf{GS} \\ G'' \parr P(\!|M_1, \dots, M_n|\!)[G']_{R_P}\end{array}}{G'' \parr G' \parr P(\!|M_1, \dots, M_n|\!)}$$

This can result in several scenarios, that are not all immediately obvious.

(c) It is possible that the last rule in a proof is a $\mathsf{ss}\downarrow$ that moves the whole prime graph inside the rest of the graph. In this case $\mathcal{D}$ is of shape

$$\mathsf{ss}\downarrow \frac{\begin{array}{c}\varnothing \\ \mathcal{D}' \parallel \mathsf{GS} \\ G[P(\!|M_1, \dots, M_n|\!)]_S\end{array}}{G \parr P(\!|M_1, \dots, M_n|\!)}$$

We will explain that this case appeals to context reduction.

(d) It is possible the last rule is a $\mathsf{p}\downarrow$ is applied directly to the principal connective, so $G = G' \parr P^{\perp}(\!|N_1, \dots, N_n|\!)$ and $\mathcal{D}$ is of shape

$$\mathsf{p}\downarrow \frac{\begin{array}{c}\varnothing \\ \mathcal{D}' \parallel \mathsf{GS} \\ G'' \parr ((N_1 \parr M_1) \otimes \dots \otimes (N_n \parr M_n))\end{array}}{G'' \parr P^{\perp}(\!|N_1, \dots, N_n|\!) \parr P(\!|M_1, \dots, M_n|\!)} .$$

(e) It is possible the last rule is a $\mathsf{p}\downarrow$ such that not all components of the principal connective are non-empty, and we have $G = G'' \parr Q(\!|N_1, \dots, N_k|\!)$ where $N_1, \dots, N_k$ are non-empty graphs, Q is a prime graph with $|V_Q| > |V_P|$ and such that w.l.o.g. $P(\!|M_1, \dots, M_n|\!) = Q^{\perp}(\!|\varnothing, L_2, \dots L_k|\!)$ for some (possibly empty) graphs $L_2, \dots L_k$, and $\mathcal{D}$ is of shape

$$\mathsf{p}\downarrow \frac{\begin{array}{c}\varnothing \\ \mathcal{D}' \parallel \mathsf{GS} \\ G'' \parr (N_1 \otimes (N_2 \parr L_2) \otimes \dots \otimes (N_k \parr L_k))\end{array}}{G'' \parr Q(\!|N_1, \dots, N_k|\!) \parr P(\!|M_1, \dots, M_n|\!)}$$

In this case it is important to ensure the side-conditions on $\mathsf{p}\downarrow$ are respected.

Proof of Splitting Lemma for Prime Graphs (Lemma 6.2). We prove the two cases simultaneously, by induction on the size of the graph $G \parr P(\!|M_1, \dots, M_n|\!)$ defined in Definition 5.13.

We aim to construct $C[\cdot]_R$, $\mathcal{D}_G$, $\mathcal{D}_C$, and either K_i, $\mathcal{D}_i$ for all $i \in \{1, \ldots, n\}$ or K_X, K_Y, $\mathcal{D}_X$, $\mathcal{D}_Y$, satisfying the condition of the statement. We make a case analysis on the bottommost rule instance in the derivation $\mathcal{D}$.

We now proceed with the proof, systematically exhausting the cases (a) to (e) described above. To avoid redundancy in the proof, we assume that whenever we define the context of the shape $C = C_n[\ldots C_2[C_1]_{R_2}]_{R_n}$ using some derivable contexts $C_1[\cdot]_{R_1}, \ldots, C_n[\cdot]_{R_n}$, then we have a derivation of C defined by Lemma 6.6.

(a) If rule r acts inside G or any M_i with $i \in \{1, \ldots, n\}$, then the derivation $\mathcal{D}$ is of shape

$$\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\end{array}}{\mathsf{r}\dfrac{G'}{G} \parr P(\!|M_1, \ldots, M_n|\!)} \quad \text{or} \quad \dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\end{array}}{G \parr P(\!|M_1, \ldots, M_{i-1}, \mathsf{r}\dfrac{M_i'}{M_i}, M_{i+1}, \ldots, M_n|\!)}$$

for some $1 \leq i \leq n$, then as the size of the conclusion of $\mathcal{D}'$ is smaller then the one of $\mathcal{D}$. We apply the induction hypothesis and conclude immediately by adding the corresponding application of r to $\mathcal{D}_G$ or $\mathcal{D}_i$ respectively.

A special case is there $\mathsf{r} = \mathsf{ai}{\downarrow}$ and $M_i' = \varnothing$, we have

$$\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\end{array}}{G \parr P(\!|M_1, \ldots, M_{i-1}, \mathsf{ai}{\downarrow}\dfrac{\varnothing}{a^\perp \parr a}, M_{i+1}, \ldots, M_n|\!)}$$

This is a base case of the induction, hence we can conclude immediately since of the form of the case (B) by letting $C = K_X = \varnothing$, and $K_Y = G$.

(b) If $G = G' \parr G''$ with $G' \neq \varnothing$ and $\mathcal{D}$ is of shape

$$\mathsf{ss}{\downarrow}\dfrac{\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\end{array}}{G'' \parr P(\!|M_1, \ldots, M_n|\!)[G']_{R_P}}}{G'' \parr G' \parr P(\!|M_1, \ldots, M_n|\!)}$$

The graph $P(\!|M_1, \ldots, M_n|\!)[G']_{R_P}$ cannot be an atom since G' and M_i are non-empty, or a par fo two graphs, due to conditions on $\mathsf{ss}{\downarrow}$. Then by Lemma 3.9, $P(\!|M_1, \ldots, M_n|\!)[G']_{R_P}$ is composed via a prime graph. We apply inductive hypothesis. We have the three following possibilities:

(b.I) In this case G moves inside some M_i. In this case, w.l.o.g., $\mathcal{D}$ is of the shape

$$\mathsf{ss}{\downarrow}\dfrac{\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\end{array}}{G'' \parr P(\!|M_1[G']_S, M_2, \ldots, M_n|\!)}}{G'' \parr G' \parr P(\!|M_1, \ldots, M_n|\!)}$$

In this case we apply the induction hypothesis to $\mathcal{D}'$ and get one of the following three sub-cases:

(b.I.A) there is a context $C'[\cdot]_{R'}$ and there are graphs $L_1, K_2, \dots, K_n$ such that there are the following derivations

$$\begin{array}{c} C'[P^{\perp}(\!|L_1, K_2, \dots, K_n|\!)]_{R'} \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ G'' \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_1 \,\|\, \mathsf{GS} \\ L_1 \parr M_1[G']_S \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}$$

for all $i \in \{2, \dots, n\}$. We let $K_1 = L_1 \parr G'$ and $C[\cdot]_R = C'[\cdot]_{R'}$. Then we conclude since $\mathcal{D}_G$ and $\mathcal{D}_1$ are the following derivations

$$\mathsf{ss}{\downarrow} \dfrac{C[P^{\perp}(\!|L_1 \parr G', K_2, \dots, K_n|\!)]_R}{\begin{array}{c} C[P^{\perp}(\!|L_1, \dots, K_2, \dots, K_n|\!)]_R \\ \mathcal{D}''_G \,\| \\ G'' \end{array} \parr G'} \quad \text{and} \quad \mathsf{ss}{\downarrow} \dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}_1 \,\| \\ L_1 \parr M_1[G']_S \end{array}}{L_1 \parr G' \parr M_1}$$

(b.I.B) there is a context $C[\cdot]_R$ and there are two graphs L_X and K_Y such that there are the following derivations

$$\begin{array}{c} C[L_X \parr K_Y]_R \\ \mathcal{D}'_G \,\|\, \mathsf{GS} \\ G'' \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \mathsf{GS} \\ L_X \parr M_1[G']_S \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|M_2, \dots, M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array}$$

We let $K_X = L_X \parr G'$. Then we conclude since $\mathcal{D}_G$ and $\mathcal{D}_X$ are the following derivations

$$\mathsf{ss}{\downarrow} \dfrac{C[L_X \parr G' \parr K_Y]_R}{\begin{array}{c} C[L_X \parr K_Y]_R \\ \mathcal{D}'_G \,\| \\ G'' \end{array} \parr G'} \quad \text{and} \quad \mathsf{ss}{\downarrow} \dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\| \\ L_X \parr M_1[G']_S \end{array}}{L_X \parr G' \parr M_1}$$

(b.I.C) there is a context $C[\cdot]_R$ and there are graphs K_X and L_Y such that

$$\begin{array}{c} C[K_X \parr L_Y]_R \\ \mathcal{D}'_G \,\|\, \mathsf{GS} \\ G'' \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_i \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\|\, \mathsf{GS} \\ L_Y \parr P(\!|M_1[G'], M_1, \dots, M_{i-1}, \varnothing, M_{i+1}, \dots, M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\|\, \mathsf{GS} \\ C \end{array}$$

for some $i \in \{2, \dots, n\}$. We let $K_Y = G' \parr L_Y$. Then we conclude since $\mathcal{D}_G$ and $\mathcal{D}_Y$ are the following derivations

$$\mathsf{ss}{\downarrow} \dfrac{C[L_X \parr G' \parr K_Y]_R}{\begin{array}{c} C[L_X \parr K_Y]_R \\ \mathcal{D}'_G \,\| \\ G'' \end{array} \parr G'} \quad \text{and} \quad \mathsf{ss}{\downarrow} \dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\| \\ L_Y \parr P(\!|M_1, \dots, M_{i-1}, \varnothing, M_{i+1}, \dots, M_j[G'], \dots, M_n|\!) \end{array}}{G' \parr L_Y \parr P(\!|M_1, \dots, M_{i-1}, \varnothing, M_{i+1}, \dots, M_n|\!)} \; .$$

(b.II) There is a special case when the prime graph P is $\otimes$ induced by graph isomorphism, which has the effect of making times associative. Assume $P = \otimes$

and $\mathcal{D}$ is of the shape, where $M_1 = A$ and $M_2 = B$.

$$\mathsf{ss}{\downarrow}\,\frac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\\ G'' \parr (A \otimes B)[G']_S\end{array}}{G'' \parr G' \parr A \otimes B}$$

with $(A \otimes B)[G']_S = A'' \otimes (A' \otimes B)[G']_{S'}$ for some $S' \subseteq S$, where $A = A'' \otimes A'$ and $A'' \neq \varnothing \neq A'$.

Notice that this is neither the case that G' moves entirely inside A or B (hence Case (b.I) cannot be applied), nor is it the case that G' is entirely a module outside the prime graph $\otimes$ connecting A'' and $(A' \otimes B)[G']_{S'}$, (in which we could move forwards to Case (b.III)). Observe furthermore that such a situation can never occur when P is not $\otimes$.

Since $\|G'' \parr (A \otimes B)[G']_S\| < \|G'' \parr G' \parr (A \otimes B)\|$ we can apply Lemma 6.1. Then there is a context C' and there are graphs K''_A and K'_Y such that there are derivations

$$\begin{array}{c}C'[K''_A \parr K_Y]_{R_1}\\ \mathcal{D}''_G \,\|\, \mathsf{GS}\\ G''\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}''_A \,\|\, \mathsf{GS}\\ K''_A \parr A''\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_Y \,\|\, \mathsf{GS}\\ K_Y \parr (A' \otimes B)[G']_{S'}\end{array}\quad \text{and}\quad \begin{array}{c}\varnothing\\ \mathcal{D}'_C \,\|\, \mathsf{GS}\\ C'\end{array}\ .$$

From $\mathcal{D}_Y$ we get that $\vdash_{\mathsf{GS}} K_Y \parr G' \parr (A' \otimes B)$ (via the rule $\mathsf{ss}{\downarrow}$).

It is important to observe at this point that we have the following inequality.

$$\|K_Y \parr G' \parr (A' \otimes B)\| < \|G \parr (A \otimes B)\|$$

This is because, since A'' is non-empty and hence $\|A'\| < \|A\|$ since A' has strictly less vertices, and also $\|K_Y \parr G'\| \leq \|G\|$, by Observation 5.14. Therefore, to the proof of $\vdash_{\mathsf{GS}} K_Y \parr G' \parr (A' \otimes B)$ we can apply the induction hypothesis to get a context $C''[\cdot]_{R''}$ and K'_A and K_B such that there are derivations as follows.

$$\begin{array}{c}C''[K_A \parr K_B]_{R''}\\ \mathcal{D}'_G \,\|\, \mathsf{GS}\\ K_Y \parr G'\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}'_A \,\|\, \mathsf{GS}\\ K'_A \parr A'\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_B \,\|\, \mathsf{GS}\\ K_B \parr B\end{array}\quad \text{and}\quad \begin{array}{c}\varnothing\\ \mathcal{D}''_C \,\|\, \mathsf{GS}\\ C''\end{array}\ .$$

We conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$, $K_A = K''_A \parr K'_A$, and $\mathcal{D}_G$ and $\mathcal{D}_A$ be the derivations defined as

$$\mathsf{ss}{\downarrow}\,\frac{C'\left[\begin{array}{c}\mathsf{ss}{\downarrow}\,\dfrac{C''[K''_A \parr K'_A \parr K_B]_{R''}}{K''_A \parr \begin{array}{c}C''[K'_A \parr K_B]_{R''}\\ \mathcal{D}'_G \,\|\\ K_Y \parr G'\end{array}}\end{array}\right]_{R'}}{\begin{array}{c}C'[K''_A \parr K_Y]_{R'}\\ \mathcal{D}''_G \,\|\\ G''\end{array} \parr G'}\quad \text{and}\quad \mathsf{ss}{\downarrow}\,\frac{\begin{array}{c}\varnothing\\ \mathcal{D}''_A \,\|\\ K''_A \parr (A'' \otimes \begin{array}{c}\varnothing\\ \mathcal{D}_{A'} \,\|\\ K_{A'} \parr A'\end{array})\end{array}}{K''_A \parr K'_A \parr (A'' \otimes A')}\ .$$

(b.III) This is the most involved sub-cases of the proof, induced when the movement of G' via the $\mathsf{ss}{\downarrow}$ rule results in a larger prime graph, where the G' does not overlap with any existing module and possibly adds edges that break up some existing modules of P into smaller modules in the resulting prime graph. Assume $\mathcal{D}$ is of shape

$$\mathsf{ss}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GS}\\ G'' \parr Q(\!|G', N_2, \dots, N_k|\!)\end{array}}{G'' \parr G' \parr P(\!|M_1, \dots, M_n|\!)}$$

for a unique prime graph Q (up to permutation of modules) such that $k = |V_Q| > |V_P|$ (hence $k \geq 4$) such that

$$Q(\!|\varnothing, N_2, \dots, N_k|\!) = P(\!|M_1, \dots, M_n|\!) \tag{6.7}$$

and each N_i for $i \in \{2, \dots k\}$ is such that $N_i \neq \varnothing$ and, furthermore, N_i is a module of some M_j where $j \in \{1, \dots n\}$. We apply the induction hypothesis to $G'' \parr Q(\!|G', N_2, \dots, N_k|\!)$ and get one of the following three sub-cases:
(b.III.A) there is a context $C'[\cdot]_{R'}$ and $K'_1, \dots, K'_k$ such that there are derivations

$$\begin{array}{c}C'[Q^\perp(\!|K'_1, \dots, K'_k|\!)]_{R'}\\ \mathcal{D}''_G \parallel \mathsf{GS}\\ G''\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}'_1 \parallel \mathsf{GS}\\ K'_1 \parr G'\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}'_i \parallel \mathsf{GS}\\ K'_i \parr N_i\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}'_C \parallel \mathsf{GS}\\ C'\end{array}$$

for all $i \in \{2, \dots, k\}$. Since the graphs $N_2, \dots, N_n$ are non-empty by hypothesis, we can apply Lemma 5.1 and obtain a derivation

$$\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}'_2 \parallel\\ K'_2 \parr N_2\end{array} \otimes \cdots \otimes \begin{array}{c}\varnothing\\ \mathcal{D}'_k \parallel\\ K'_k \parr N_k\end{array}}{\begin{array}{c}\parallel \text{Lemma 5.1}\\ Q^\perp(\!|\varnothing, K'_2, \dots, K'_k|\!) \parr Q(\!|\varnothing, N_2, \dots, N_k|\!)\end{array}}$$

By (6.7), such a derivation is a derivation of $Q^\perp(\!|\varnothing, K_2, \dots, K_k|\!) \parr P(\!|M_1, \dots, M_n|\!)$. on which we can now apply the inductive hypothesis. This give us the two following cases

- either there is a context $C''[\cdot]_{R''}$ and graphs $K_1, \dots, K_k$ such that there are derivations

$$\begin{array}{c}C''[P^\perp(\!|K_1, \dots, K_k|\!)]_{R''}\\ \mathcal{D}''_G \parallel \mathsf{GS}\\ Q^\perp(\!|\varnothing, K'_2, \dots, K'_k|\!)\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_i \parallel \mathsf{GS}\\ K_i \parr M_i\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}''_C \parallel \mathsf{GS}\\ C''\end{array}$$

for all $i \in \{1, \dots, n\}$. Then we conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and $\mathcal{D}_G$ be the derivation

$$\mathsf{ss}\downarrow \dfrac{C'\left[\begin{array}{c} C''[P^\perp(\!|K_1, \dots, K_k|\!)]_{R''} \\ \mathcal{D}''_G \,\|\, \\ Q^\perp(\!|\begin{array}{c}\varnothing \\ \mathcal{D}'_1 \,\|\, \\ K'_1 \parr G'\end{array}, K'_2, \dots, K'_k|\!) \end{array}\right]_{R'}}{\begin{array}{c} C'[Q^\perp(\!|K'_1, \dots, K'_k|\!)]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

- or there is a context $C''[\cdot]_{R''}$ and graphs K_X and K_Y such that, w.l.o.g. there are derivations

$$\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ Q^\perp(\!|\varnothing, K'_2, \dots, K'_k|\!) \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_1 \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|\varnothing, M_2, \dots, M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

for an $i \in \{1, \dots, n\}$. Then we conclude by letting $C[\cdot]_R = C'[\cdot]_{R'}$ and $\mathcal{D}_G$ be the derivation

$$\mathsf{ss}\downarrow \dfrac{C'\left[\begin{array}{c} C''\left[K_X \parr K_Y\right]_{R''} \\ \mathcal{D}''_G \,\|\, R' \\ Q^\perp(\!|\begin{array}{c}\varnothing \\ \mathcal{D}'_1 \,\|\, \\ K'_1 \parr G'\end{array}, K'_2, \dots, K'_k|\!) \end{array}\right]}{\begin{array}{c} C''[K' \parr K_2 \parr \cdots \parr K_n]_{R''} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

(b.III.B) there is a context $C'[\cdot]_{R'}$ and K'_X and K'_Y such that (after 6.7) there are derivations

$$\begin{array}{c} C'[K'_X \parr K'_Y]_R \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ G'' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \mathsf{GS} \\ K'_X \parr G' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\|\, \mathsf{GS} \\ K'_Y \parr P(\!|M_1, \dots, M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}$$

We can now apply inductive hypothesis on $K'_Y \parr P(\!|M_1, \dots, M_n|\!)$ and we have the two following cases:

- there is a context $C''[\cdot]_{R''}$ and graphs $K_1, \dots, K_n$ such that there are derivations

$$\begin{array}{c} C''[P^\perp(\!|K_1, \dots, K_n|\!)]_{R''} \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ K'_Y \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

In this case we can conclude similarly to the first case of (b.III.A), that is, by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and $\mathcal{D}_G$ be the derivation

$$\mathsf{ss}\downarrow \dfrac{C'\left[\begin{array}{c} C''[P^{\perp}(\!|K_1,\ldots,K_n|\!)]_{R''} \\ \mathcal{D}''_G \,\|\, \\ K'_Y \end{array} \parr \begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \\ K'_X \parr G' \end{array}\right]_{R'}}{\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

- or there is a context $C''[\cdot]_{R''}$ and graphs K_X and K_Y such that, w.l.o.g. there are derivations

$$\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ Q^{\perp}(\!|\varnothing, K'_2,\ldots,K'_k|\!) \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_1 \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|\varnothing, M_2,\ldots,M_n|\!) \end{array} \text{ and } \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

we conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and $\mathcal{D}_G$ be the derivation

$$\mathsf{ss}\downarrow \dfrac{C'\left[\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}''_G \,\|\, \\ K'_Y \end{array} \parr \begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \\ K'_X \parr G' \end{array}\right]_{R'}}{\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

(b.III.C) There is a context $C'[\cdot]_{R'}$ and there are graphs K'_X and K'_Y such that for have the following derivations for some $\ell \in \{2,\ldots k\}$.

$$\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ G'' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \mathsf{GS} \\ K'_X \parr N_\ell \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\|\, \mathsf{GS} \\ K'_Y \parr Q(\!|G', N_2,\ldots,N_{\ell-1},\varnothing,N_{\ell+1},\ldots N_k|\!) \end{array} \text{ and } \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}$$

There are two cases to consider: either $N_\ell = M_m$ for some $m \in \{1,\ldots,k\}$ or we have $M_m = M'[N_\ell]_{R_m}$ for some non-empty M'.
We consider first the former case where $N_\ell = M_m$ for some $m \in \{1,\ldots,k\}$, the derivation $\mathcal{D}_Y$ is defined, recalling that by (6.7) we have $Q(\!|\varnothing, N_2,\ldots,N_k|\!) = P(\!|M_1,\ldots,M_n|\!)$ and hence $Q(\!|\varnothing, N_2,\ldots,N_{\ell-1},\varnothing,N_{\ell+1},\ldots N_k|\!) = P(\!|M_1,\ldots,M_{m-1},\varnothing,M_{m+1},\ldots M_n|\!)$,

$$\mathsf{ss}\downarrow \dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\|\, \\ K'_Y \parr Q(\!|G', N_2,\ldots,N_{\ell-1},\varnothing,N_{\ell+1},\ldots N_k|\!) \end{array}}{K'_Y \parr G' \parr P(\!|M_1,\ldots,M_{m-1},\varnothing,M_{m+1},\ldots M_n|\!)}$$

We can conclude almost immediately by letting $C[\cdot]_R = C'[\cdot]_{R'}$, $K_Y = K'_Y \parr G'$, $K_X = K'_X$, $\mathcal{D}_X = \mathcal{D}'_X$, $\mathcal{D}_Y = \mathcal{D}'_Y$ and $\mathcal{D}_G$ be the following derivation.

$$\mathsf{ss}{\downarrow}\dfrac{C'[K_Y \parr K_X]_{R'}}{\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

.

Otherwise, we pursue the case where $M_m = M'[N_\ell]_{R_m}$ for some non-empty M', which is more involved than the case above. By (6.7) we have

$$Q(\!|\varnothing, N_2, \ldots, N_{\ell-1}, \varnothing, N_{\ell+1}, \ldots N_k|\!) = P(\!|M_1, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)$$

Hence, we have a proof of $K'_Y \parr G' \parr P(\!|M_1, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)$ obtained by applying the rule $\mathsf{ss}{\downarrow}$ to the conclusion of $\mathcal{D}'_Y$ above, as follows.

$$\mathsf{ss}{\downarrow}\dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}'_Y \,\|\, \\ K'_Y \parr Q(\!|G', N_2, \ldots, N_{\ell-1}, \varnothing, N_{\ell+1}, \ldots N_k|\!) \end{array}}{K'_Y \parr G' \parr P(\!|M_1, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)}$$

By Observation 5.14 $\|K'_Y \parr G'\| \leq \|G\|$ and also $N_\ell \neq \varnothing$, hence we have the following inequality.

$$\|K'_Y \parr G' \parr P(\!|M_1, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)\| < \|G \parr P(\!|M_1, \ldots, M_n|\!)\|$$

Therefore, by the above inequality, we can apply the induction hypothesis to $K_Y \parr G' \parr P(\!|M_1, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)$ to obtain one of the following three sub-cases:

(b.III.C.A) Consider when there is a context C' and there are graphs $K_1, \ldots, K_{m-1}, L, K_{m+1}, \ldots, K_n$ such that there are derivations

$$\begin{array}{c} C''[P^\perp(\!|K_1, \ldots, K_{m-1}, L, K_{m+1}, \ldots, K_n|\!)]_{R''} \\ \mathcal{D}'''_G \,\|\, \mathsf{GS} \\ K'_Y \parr G' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_m \,\|\, \mathsf{GS} \\ L \parr M' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

for $i \in \{1, \ldots, n\}$ such that $i \neq m$.

We conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ (see Lemma 6.6), and $K_m = K'_X \parr L$ and we let the derivations $\mathcal{D}_G$ is defined as

$$\mathsf{ss}{\downarrow}\dfrac{C'\left[\mathsf{ss}{\downarrow}\dfrac{C''[P^\perp(\!|K_1, \ldots, K'_X \parr L, K_{m+1}, \ldots, K_n|\!)]_{R''}}{K'_X \parr \begin{array}{c} C''[P^\perp(\!|K_1, \ldots, L, K_{m+1}, \ldots, K_n|\!)]_{R''} \\ \mathcal{D}'''_G \,\|\, \\ K'_Y \parr G' \end{array}}\right]_{R'}}{\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'}$$

and $\mathcal{D}_m$ is defined as follows

$$\mathsf{ss}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}'_m\,\|\\ L \parr M'\left[\begin{array}{c}\varnothing\\ \mathcal{D}'_X\,\|\\ K'_X \parr N_\ell\end{array}\right]_{R_m}\end{array}}{K'_X \parr L \parr M'[N_\ell]_{R_m}}$$

(b.III.C.B) there is a context $C''[\cdot]_{R''}$ and there are graphs K_X and K''_Y such that there are derivations

$$\begin{array}{c}C''[K_X \parr K''_Y]_{R''}\\ \mathcal{D}'''_G\,\|\,\mathsf{GS}\\ K'_Y \parr G'\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}_X\,\|\,\mathsf{GS}\\ K_X \parr M_1\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}''_Y\,\|\,\mathsf{GS}\\ K''_Y \parr P(\!|\varnothing, M_2, \ldots, M_{m-1}, M', M_{m+1}, \ldots M_n|\!)\end{array}\quad\text{and}\quad\begin{array}{c}\varnothing\\ \mathcal{D}''_C\,\|\,\mathsf{GS}\\ C''\end{array}$$

We conclude by setting $K_Y = K'_X \parr K''_Y$ and $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$, where the derivations $\mathcal{D}_G$ is defined as

$$\mathsf{ss}{\downarrow}\dfrac{C'\left[K'_X \parr \begin{array}{c}K_X \parr K''_Y\\ \mathcal{D}'''_G\,\|\\ K'_Y \parr G'\end{array}\right]_{R'}}{\begin{array}{c}C'[K'_X \parr K'_Y]_{R'}\\ \mathcal{D}''_G\,\|\\ G''\end{array}\parr G'}$$

and the derivation $\mathcal{D}_Y$ is defined as

$$\mathsf{ss}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}''_Y\,\|\\ K''_Y \parr P(\!|\varnothing, \ldots, M_{m\bar{1}}, M'[\begin{array}{c}\varnothing\\ \mathcal{D}'_X\,\|\\ K'_X \parr N_\ell\end{array}]_{R_m}, M_{m+1}, \ldots M_n|\!)\end{array}}{K'_X \parr K''_Y \parr P(\!|\varnothing, M_2, \ldots, M'[N_\ell]_{R_m}, M_{m+1}, \ldots M_n|\!)}$$

where $K_Y = K'_X \parr K''_Y$ and $M'[N_\ell]_{R_m} = M_m$.
(b.III.C.C) there is a context C' and there are graphs K''_X and K_Y such that, there are derivations

$$\begin{array}{c}C''[K''_X \parr K_Y]_{R''}\\ \mathcal{D}'''_G\,\|\\ K'_Y \parr G'\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}''_X\,\|\,\mathsf{GS}\\ K''_X \parr M'\end{array}\quad,\quad\begin{array}{c}\varnothing\\ \mathcal{D}_Y\,\|\,\mathsf{GS}\\ K_Y \parr P(\!|\varnothing, M_2, \ldots, M_{m-1}, \varnothing, M_{m+1}, \ldots M_n|\!)\end{array}\quad\text{and}\quad\begin{array}{c}\varnothing\\ \mathcal{D}''_C\,\|\,\mathsf{GS}\\ C''\end{array}$$

We conclude by letting $K_X = K'_X \parr K''_X$ and $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$, where the derivations $\mathcal{D}_G$ and $\mathcal{D}_X$ are defined as follows

$$C'\left[\mathsf{ss}{\downarrow}\,\frac{C''[K'_X \parr K''_X \parr K_Y]_{R''}}{K'_X \parr \begin{array}{c} C''[K''_X \parr K_Y]_{R'} \\ \mathcal{D}'''_G \,\|\, \\ K'_Y \parr G' \end{array}}\right]_{R''} \mathsf{ss}{\downarrow}\,\frac{}{\begin{array}{c} C'[K'_X \parr K'_Y]_{R'} \\ \mathcal{D}''_G \,\|\, \\ G'' \end{array} \parr G'} \qquad \text{and} \qquad \mathsf{ss}{\downarrow}\,\frac{\begin{array}{c} \varnothing \\ \hat{\mathcal{D}_X} \,\|\, \\ K''_X \parr M'\left[\begin{array}{c} \varnothing \\ \mathcal{D}'_X \,\|\, \\ K'_X \parr N_\ell \end{array}\right]_{R_m} \end{array}}{K''_X \parr K'_X \parr M'[N_\ell]_{R_m}}$$

(c) Consider the case where the $\mathsf{ss}{\downarrow}$ is applied and $\mathcal{D}$ is of shape

$$\mathsf{ss}{\downarrow}\,\frac{\begin{array}{c} \varnothing \\ \mathcal{D}' \,\|\, \\ G[P(\!|M_1, \ldots, M_n|\!)]_S \end{array}}{G \parr P(\!|M_1, \ldots, M_n|\!)}$$

By Lemma 6.4 there is a graph K and a context $C'[\cdot]_R$ such that there are derivations

$$\begin{array}{c} \varnothing \\ \mathcal{D}_A \,\|\, \mathsf{GS} \\ K \parr P(\!|M_1, \ldots, M_n|\!) \end{array}, \quad \begin{array}{c} C'[K \parr X]_R \\ \mathcal{D}^X_G \,\|\, \mathsf{GS} \\ G[X]_S \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}$$

for any graph X.

Because of $\mathcal{D}^X_G$ we have that $\|K\| < \|G\|$ hence we can apply the induction hypothesis to $K \parr P(\!|M_1, \ldots, M_n|\!)$ to obtain two following cases:

- either there is a context $C''[\cdot]_R$ and there are graphs K_1, …, K_n, such that $\|P(\!|M_1, \ldots, M_n|\!)\| < \|K\|$ and there are derivations

$$\begin{array}{c} C''[P^\perp(\!|K_1, \ldots, K_n|\!)]_R \\ \mathcal{D}'_K \,\|\, \mathsf{GS} \\ K \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

for all $i \in \{1, \ldots, n\}$, Therefore there is a context $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and a derivation

$$C'\left[\begin{array}{c} C''[P^\perp(\!|K_1, \ldots, K_n|\!)]_{R''} \\ \mathcal{D}'_K \,\|\, \\ K \end{array}\right]_{R'} \\ \mathcal{D}^\varnothing_G \,\|\, \\ G$$

- or there is a context $C''[\cdot]_R$ and there are graphs K_X and K_Y and there are derivations

$$\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}_K \,\|\, \mathsf{GS} \\ K \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_i \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|M_1, \ldots, M_{i-1}, \varnothing, M_{i+1}, \ldots, M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

for some $i \in \{1, \ldots, n\}$.
Therefore there is a context $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and a derivation

$$C'\left[\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}'_K \,\| \\ K \end{array}\right]_{R'} \quad \begin{array}{c} \\ \mathcal{D}^{\varnothing}_G \,\| \\ G \end{array}$$

(d) Consider when we have $G = G' \parr P^{\perp}(\!|N_1, \ldots, N_n|\!)$ and $\mathcal{D}$ is of shape

$$\mathsf{p}\!\downarrow \frac{\begin{array}{c} \varnothing \\ \mathcal{D}' \,\| \\ G' \parr ((N_1 \parr M_1) \otimes \cdots \otimes (N_n \parr M_n)) \end{array}}{G' \parr P^{\perp}(\!|N_1, \ldots, N_n|\!) \parr P(\!|M_1, \ldots, M_n|\!)} .$$

By Lemma 6.5 there are a context $C[\cdot]_R$ and graphs $L_1, \ldots, L_n$ such that

$$\begin{array}{c} C[L_1 \parr \cdots \parr L_n]_R \\ \mathcal{D}'_G \,\| \\ G' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\| \\ L_i \parr N_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \,\| \\ C \end{array}$$

for all $i \in \{1, \ldots, n\}$. We let $K_i = L_i \parr N_i$. Then there is a derivation $\mathcal{D}_G$ defined as

$$\mathsf{ss}\!\downarrow \frac{C\left[\begin{array}{c} P^{\perp}(\!|L_1 \parr N_1, \ldots, L_n \parr N_n|\!) \\ \|\, \{\mathsf{ss}\!\downarrow\} \\ L_1 \parr \cdots \parr L_n \parr P^{\perp}(\!|N_1, \ldots, N_n|\!) \end{array}\right]_R}{\begin{array}{c} L_1 \parr \cdots \parr L_n \\ \mathcal{D}'_G \,\| \\ G' \end{array} \quad \parr P^{\perp}(\!|N_1, \ldots, N_n|\!)} .$$

(e) In this case, the $\mathsf{p}\!\downarrow$ rule is applied to a larger prime graph Q in the context of the principal prime graph P. We have in this case that $\mathcal{D}$ is of shape

$$\mathsf{p}\!\downarrow \frac{\begin{array}{c} \varnothing \\ \mathcal{D}' \,\| \\ G'' \parr (N_1 \otimes (N_2 \parr L_2) \otimes \cdots \otimes (N_k \parr L_k)) \end{array}}{G'' \parr Q(\!|N_1, \ldots, N_k|\!) \parr P(\!|M_1, \ldots, M_n|\!)}$$

An essential assumption is that all modules of Q must be non-empty. Hence, in this case we assume $G = G'' \parr Q(\!|N_1, \ldots, N_k|\!)$ where $N_1, \ldots, N_k$ are non-empty graphs,

Q is a prime graph with $k = |V_Q| > |V_P|$ such that w.l.o.g. $P(\!|M_1, \ldots, M_n|\!) = Q^{\perp}(\!|\varnothing, L_2, \ldots L_k|\!)$ for some possibly empty graphs $L_2, \ldots L_k$. Observe at least one module of the prime connective $Q^{\perp}$ must be empty for this equality to hold and we set that w.l.o.g. to be the first module, otherwise $P^{\perp}$ and Q are isomorphic, contradicting $|V_Q| > |V_P|$.

We apply Lemma 6.5 and we get a context $C'[\cdot]_{R'}$ and graphs $K'_1, \ldots, K'_k$

$$\begin{array}{c} C'[K'_1 \parr \cdots \parr K'_k]_R \\ \mathcal{D}''_G \,\|\, \mathsf{GS} \\ G'' \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_1 \,\|\, \mathsf{GS} \\ K'_1 \parr N_1 \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_j \,\|\, \mathsf{GS} \\ K'_j \parr (N_j \parr L_j) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}$$

for all for all $j \in \{2, \ldots, k\}$.

Now observe, for some graph H (that is not necessarily prime), we have that $Q(\!|\varnothing, K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) = H(\!|K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!)$ and $Q^{\perp}(\!|\varnothing, L_2, \ldots L_k|\!) = H^{\perp}(\!|L_2, \ldots L_k|\!)$. Hence, since for all $i \in \{2, \ldots, k\}$ we have $N_i \neq \varnothing$, we can apply Lemma 5.1 (notice the non-emptyness of modules of H is required in this lemma) to construct the following proof.

$$\dfrac{\left(\begin{array}{c} \varnothing \\ \mathcal{D}'_2 \,\| \\ K'_2 \parr N_2 \parr L_2 \end{array} \otimes \ldots \otimes \begin{array}{c} \varnothing \\ \mathcal{D}'_k \,\| \\ K'_k \parr N_k \parr L_k \end{array}\right)}{\begin{array}{c} \mathcal{D}_H \,\|\, \text{Lemma 5.1} \\ H(\!|K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \parr H^{\perp}(\!|L_2, \ldots, L_k|\!) \end{array}}$$

Now, observe that we have $H^{\perp}(\!|L_2, \ldots, L_k|\!) = P(\!|M_1, \ldots, M_n|\!)$. Furthermore, $\|K'_1 \parr \cdots \parr K'_k\| \leq \|G''\|$, and hence we have $\|H(\!|K'_1 \parr N_1, \ldots, K'_k \parr N_k|\!)\| \leq \|G\|$. Also $N_1 \neq \varnothing$ and N_1 does not appear in the conclusion of the proof immediately above; hence we have the following inequality.

$$\|H(\!|K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \parr P(\!|M_1, \ldots, M_n|\!)\| < \|G \parr P(\!|M_1, \ldots, M_n|\!)\|$$

- there is a context $C''[\cdot]_{R''}$ and graphs $K_1 \parr \ldots K_n$ such that there are derivations

$$\begin{array}{c} C''[P^{\perp}(\!|K_1 \parr \ldots K_n|\!)]_{R''} \\ \mathcal{D}'''_G \,\|\, \mathsf{GS} \\ Q(\!|\varnothing, K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}$$

and, for all $i \in \{1, \ldots, n\}$. We conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and the derivation $\mathcal{D}_G$ be defined as

$$\dfrac{C'\left[\dfrac{\begin{array}{c} C''[P^{\perp}(\!|K_1 \parr \ldots K_n|\!)]_{R''} \\ \mathcal{D}'''_G \,\|\, \\ Q(\!|\begin{array}{c}\varnothing \\ \mathcal{D}'_1 \,\|\, \\ K'_1 \parr N_1\end{array}, K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \end{array}}{\mathsf{ss}\downarrow\ K'_1 \parr K'_2 \parr \ldots K'_k \parr Q(\!|N_1, \ldots, N_k|\!)}\right]_{R'}}{\mathsf{ss}\downarrow\ \begin{array}{l} C'[K'_1 \parr K'_2 \parr \ldots K'_k]_{R'} \\ \mathcal{D}'' \,\|\, \\ G'' \end{array} \quad \parr Q(\!|N_1, \ldots, N_k|\!)}$$

- or we have a context $C''[\cdot]_{R''}$ and graphs K_X and K_Y such that, w.l.o.g. there are derivations

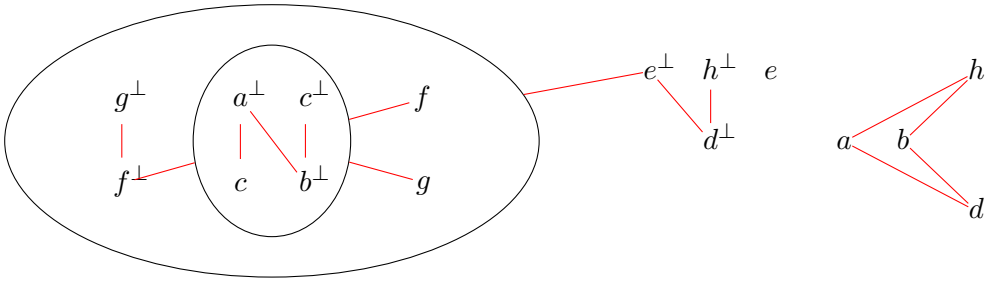
$$\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}'''_G \,\|\, \mathsf{GS} \\ Q(\!|\varnothing, K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \end{array}, \quad \begin{array}{c}\varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ K_X \parr M_1\end{array}, \quad \begin{array}{c}\varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ K_Y \parr P(\!|M_2, \ldots, M_n|\!)\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C''\end{array}$$

We conclude by letting $C[\cdot]_R = C'[C''[\cdot]_{R''}]_{R'}$ and the derivation $\mathcal{D}_G$ be defined as

$$\dfrac{C'\left[\dfrac{\begin{array}{c} C''[K_X \parr K_Y]_{R''} \\ \mathcal{D}'''_G \,\|\, \\ Q(\!|\begin{array}{c}\varnothing \\ \mathcal{D}'_1 \,\|\, \\ K'_1 \parr N_1\end{array}, K'_2 \parr N_2, \ldots, K'_k \parr N_k|\!) \end{array}}{\mathsf{ss}\downarrow\ K'_1 \parr K'_2 \parr \ldots K'_k \parr Q(\!|N_1, \ldots, N_k|\!)}\right]_{R'}}{\mathsf{ss}\downarrow\ \begin{array}{l} C'[K'_1 \parr K'_2 \parr \ldots K'_k]_{R'} \\ \mathcal{D}'' \,\|\, \\ G'' \end{array} \quad \parr Q(\!|N_1, \ldots, N_k|\!)}$$

□

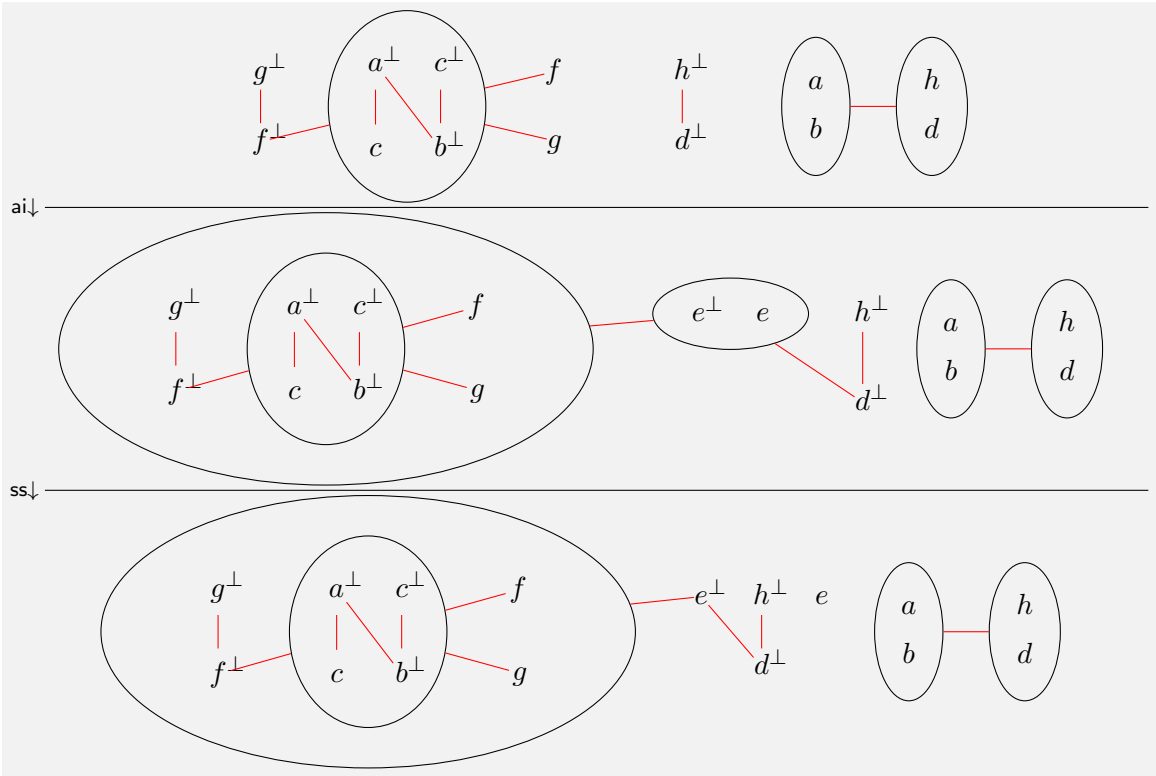
To understand why the cases in the previous proof can become complex consider the following rather exotic example.



(6.8)

One possible way to prove the above graph is to first apply the ss↓-rule to move the vertex labelled e so that it is attached by an edge only to h. This way we obtain exactly the graph in the conclusion of Figure 2, therefore we know already how to complete the proof via that strategy.

Now observe that the above graph 6.8 is of the form $G \parr (a \parr b) \otimes (h \parr d)$. Thus it is in a form to which we could apply splitting (Lemma 6.1). Applying splitting to this $\otimes$ operator as the principal graph forces the proof to be renormalised such that firstly e moves next to $e^\perp$ as shown in the derivation below.

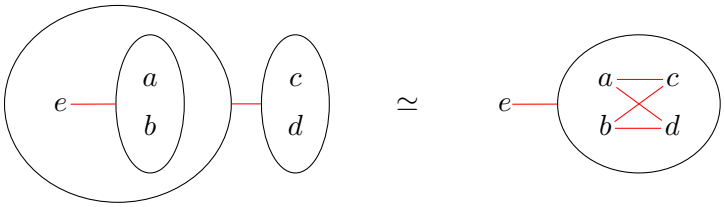


Observe that in the topmost graph in the derivation above we have broken the graph on the left into two disjoint graphs. The first is provable when composed with $a \parr b$, while the second is provable when composed with $d \parr h$.

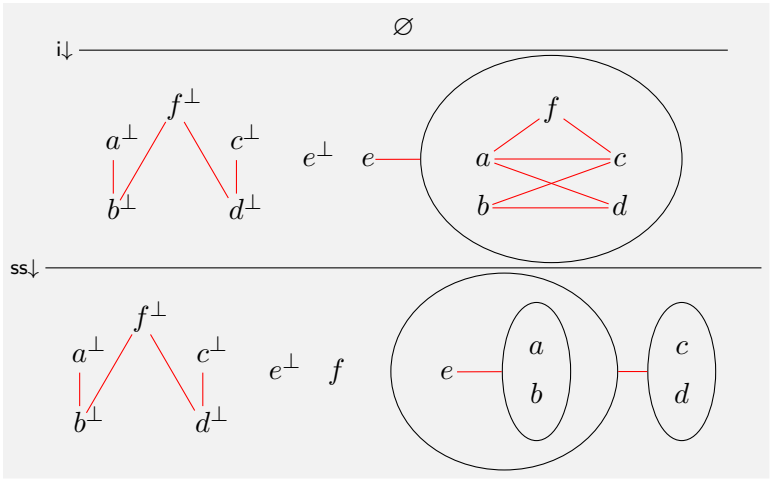
The choices between the different rules to first apply to graph 6.8, where ss↓ is applied to e in order to either create a larger prime graph on the right (leading to the proof in Figure 2)

or to break up the structure on the left as shown above, leads to quite a different proof of the structure. When moving from the former to the latter proof appeals to Case (b.III) in the proof of Lemma 6.2 during the normalisation procedure, which is the most involved sub-case in the proof of that lemma given above.

The Case (b.II) in the proof of Lemma 6.2 is not immediately obvious but is usually present in splitting proofs. In ordinary splitting proofs for logics on formulas, similar cases are induced by associativity, whereas here graph isomorphism plays a role. Firstly, observe that clearly the two presentations of the graphs below are isomorphic, even if they are composed differently.

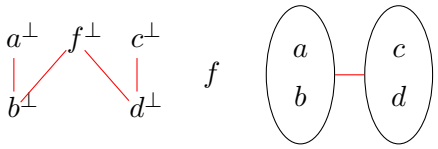


To understand the necessity of Case (b.II) in the proof of Lemma 6.2 consider the following proof of a graph, where the above graph isomorphism is appealed to in order to enable the ss↓ to be applied as shown below.



(6.9)

We can apply splitting to the above graph in various ways. One possibility is to apply splitting to the tensor on the right as shown in the conclusion of the above proof. In order to apply splitting in that way, we appeal to Case (b.II) in the proof of Lemma 6.2, which firstly observes that there is a proof of $e^\perp \parr e$ and a proof of the following induced sub-graph.



(6.10)

Case (b.III) is then appealed to to renormalise the proof of 6.10 such that the P_5 on the left is split into two parts, from which we can reassemble a proof of the graph in the conclusion of 6.9 which satisfies the conditions of splitting for the tensor that we selected. In particular, if $A = e \otimes (a \parr b)$ and $B = c \parr d$ in Lemma 6.1, then we obtain $K_A = (a^\perp \otimes b^\perp) \parr e^\perp$ and $K_B = c^\perp \parr d^\perp$, as required (the context can be empty for this example).

Beyond the above two illustrative examples of non-trivial normalisation steps, observe that the deepest nesting in the proof of splitting for prime graphs is inside Case (b.III), which has two more levels of nested case analysis to cover all ways in which we can handle situations where the principal graph is turned into a larger prime graph using the $\mathsf{ss}{\downarrow}$ rule.

Finally, it remains to give the proof of Atomic Splitting (Lemma 6.3), which follows from context reduction (Lemma 6.4) and splitting for prime graphs (Lemma 6.2), as indicated in Figure 3.

Proof of Atomic Splitting (Lemma 6.3). The proof is similar to the one of Lemma 6.2. We assume $\vdash_{\mathsf{GS}} G \parr a$ and aim to construct $C[\cdot]_R$, $\mathcal{D}_G$, $\mathcal{D}_C$ as in the statement of the lemma. Observe that $G \neq \varnothing$, otherwise $G \parr a$ would not be probable in GS. We make a case analysis on the bottommost rule instance r in $\mathcal{D}$.

(a) If rule r acts inside G, then the derivation $\mathcal{D}$ is of shape

$$\mathsf{r}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\\ G'\end{array}}{G} \parr a$$

for some $\mathcal{D}'$. By Observation 5.14 we know that $\|G'\| < \|G\|$, then we apply the induction hypothesis on $G' \parr a$. This gives us a context $C'[\cdot]_{R'}$ and two derivations

$$\begin{array}{c} C'[a^\perp]_R \\ \mathcal{D}'_G \,\|\, \mathsf{GS} \\ G' \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array} \quad .$$

We conclude by applying the rule r to the conclusion G' of $\mathcal{D}_{G'}$.

(b) If the last rule in $\mathcal{D}$ is a $\mathsf{ss}{\downarrow}$, then $\mathcal{D}$ is of the following shape $\mathcal{D}$ is of shape

$$\mathsf{ss}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\\ G[a]_R\end{array}}{G \parr a} \quad \text{or} \quad \mathsf{ss}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \,\|\, \mathsf{GS}\\ a[G]_R\end{array}}{G \parr a}$$

By condition on $\mathsf{ss}{\downarrow}$, $R \neq \varnothing$ and then $a[G]_R = a \otimes G = G[a]_{V_G}$. It is therefore sufficient to discuss the first case only. By Lemma 6.4 there is a graph K and a context $C'[\cdot]'_R$ such that there are derivations

$$\begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_a \,\|\, \mathsf{GS} \\ K \parr a \end{array} \quad \text{and} \quad \begin{array}{c} C'[K \parr X]_R \\ \mathcal{D}^X_G \,\|\, \mathsf{GS} \\ G[X]_S \end{array}$$

for any graph X. We let $C[\cdot]_R$ and we conclude since $K = a^\perp$ and there is a derivation

$$\begin{array}{c} C'[a^\perp \parr \varnothing]'_R \\ \mathcal{D}^{\varnothing}_G \,\| \\ G[\varnothing]_S \end{array} = \begin{array}{c} C[a^\perp]_R \\ \mathcal{D}_G \,\| \\ G \end{array}$$

(c) If the last rule is a $\mathsf{p}\downarrow$ which does not act inside G, then w.l.o.g. $\mathcal{D}$ is of the shape

$$\begin{array}{c} \varnothing \\ \mathcal{D}' \,\|\, \mathsf{GS} \\ G'' \parr \mathsf{p}\downarrow \dfrac{((N_1 \parr a) \otimes N_2 \otimes \cdots \otimes N_m)}{P(\!|N_1, \ldots, N_m|\!) \parr a} \end{array}$$

Then by Lemma 6.5 there is a context $C'[\cdot]_{R'}$ and there are graphs $L_1, \ldots, L_m$ such that there are derivations

$$\begin{array}{c} C'[L_1 \parr \cdots \parr L_m]_{R'} \\ \mathcal{D}'_G \,\|\, \mathsf{GS} \\ G'' \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_1 \,\|\, \mathsf{GS} \\ L_1 \parr (a \parr N_1) \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GS} \\ L_i \parr N_i \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_C \,\|\, \mathsf{GS} \\ C' \end{array}.$$

for all $i \in \{2, \ldots, m\}$. By inductive hypothesis on $L_1 \parr (a \parr N_1)$ we have

$$\begin{array}{c} C''[a^\perp]_{R''} \\ \mathcal{D}'_1 \,\|\, \mathsf{GS} \\ L_1 \parr N_1 \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}''_C \,\|\, \mathsf{GS} \\ C'' \end{array}.$$

We conclude by setting $C = C'[C''[\cdot]_{R''}]_{R'}$ since

$$\mathsf{ss}\downarrow \dfrac{C'\left[\mathsf{ss}\downarrow \dfrac{\begin{array}{c} C''[a^\perp]_{R''} \\ \mathcal{D}'_1 \,\| \\ L_1 \parr N_1 \end{array} \otimes \begin{array}{c} \varnothing \\ \mathcal{D}'_2 \,\| \\ L_2 \parr N_2 \end{array} \otimes \cdots \otimes \begin{array}{c} \varnothing \\ \mathcal{D}'_m \,\| \\ L_m \parr N_m \end{array}}{L_1 \parr \cdots \parr L_m \parr \mathsf{p}\downarrow \dfrac{N_1 \otimes N_2 \otimes \cdots \otimes N_m}{P(\!|N_1, \ldots, N_m|\!)}}\right]_{R'}}{\begin{array}{c} C'[L_1 \parr \cdots \parr L_m]_{R'} \\ \mathcal{D}''_G \,\| \\ G'' \end{array} \parr P(\!|N_1, \ldots, N_m|\!)}$$

(d) if the last rule in $\mathcal{D}$ is an $\mathsf{ai}\downarrow$, then $\mathcal{D}$ is of shape

$$\begin{array}{c} \varnothing \\ \mathcal{D}'_G \,\|\, \mathsf{GS} \\ G' \parr \mathsf{ai}\downarrow \dfrac{\varnothing}{a^\perp \parr a} \end{array}$$

and we conclude by letting $C = \varnothing$ (hence $\mathcal{D}_C$ is trivial) and $\mathcal{D}_G = \mathcal{D}'_G$. □

7. Elimination of the Up-Fragment

In this section we discuss how we use splitting and context reduction to prove Theorem 5.5, i.e., the admissibility of the rules $\mathsf{ai}{\uparrow}$, $\mathsf{ss}{\uparrow}$, and $\mathsf{p}{\uparrow}$. The procedure is similar to ordinary deep inference systems (see, e.g., [Str03, GS11, HTAC19, CGS11]. To prove up-rule elimination we show that if the conclusion of such a rule is derivable, then its premise is. The strategy this procedure consists in applying the context reduction (Lemma 6.4) to extract the conclusion of the rule from the context in which it is applied. Then the use of splitting (Lemmas 6.2 or 6.3) allows us to find derivations which can be used to to construct a derivation of the premise of the rule.

Theorem 7.1. *The rule* $\mathsf{ai}{\uparrow}$ *is admissible for* GS.

Proof. Assume we have a proof of $G[a \otimes a^{\perp}]_S$. By Lemma 6.4 we have a graph L and a context $C_1[\cdot]_{R_1}$, such that there are derivations

$$\begin{array}{c}\varnothing\\ \mathcal{D}_1 \,\|\, \mathsf{GS}\\ C_1\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_2 \,\|\, \mathsf{GS}\\ L \parr (a \otimes a^{\perp})\end{array}\quad \text{and} \quad \begin{array}{c}C_1[L \parr X]_{R_1}\\ \mathcal{D}_3 \,\|\, \mathsf{GS}\\ G[X]_S\end{array}$$

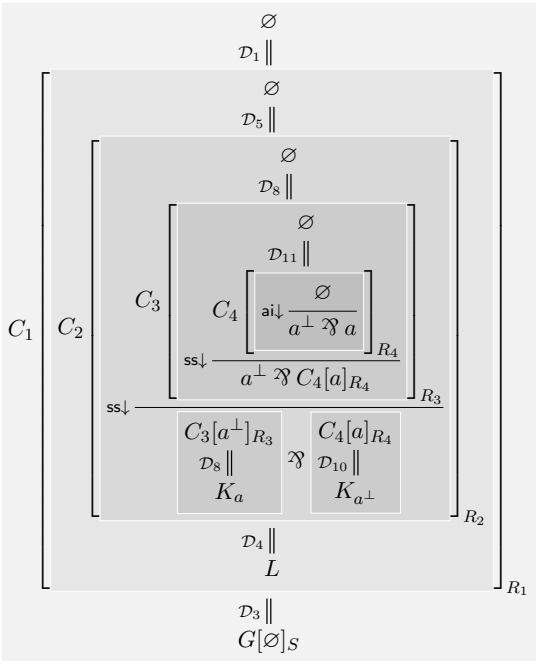
for any graph X. We apply Lemma 6.1 to $L \parr (a \otimes a^{\perp})$ and get K_a and $K_{a^{\perp}}$ and a context $C_2[\cdot]_{R_2}$ such that

$$\begin{array}{c}C_2[K_a \parr K_{a^{\perp}}]_{R_2}\\ \mathcal{D}_4 \,\|\, \mathsf{GS}\\ L\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_6 \,\|\, \mathsf{GS}\\ K_a \parr a\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_7 \,\|\, \mathsf{GS}\\ K_{a^{\perp}} \parr a^{\perp}\end{array}\quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_5 \,\|\, \mathsf{GS}\\ C_2\end{array}$$

Applying Lemma 6.3 to $K_a \parr a$ and $K_{a^{\perp}} \parr a^{\perp}$ gives us $C_3[\cdot]_{R_3}$ and $C_4[\cdot]_{R_4}$ such that

$$\begin{array}{c}C_3[a^{\perp}]_{R_3}\\ \mathcal{D}_8 \,\|\, \mathsf{GS}\\ K_a\end{array}\ ,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_9 \,\|\, \mathsf{GS}\\ C_3\end{array}\ ,\quad \begin{array}{c}C_4[a]_{R_4}\\ \mathcal{D}_{10} \,\|\, \mathsf{GS}\\ K_{a^{\perp}}\end{array}\quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_{11} \,\|\, \mathsf{GS}\\ C_4\end{array}$$

We can now give the following derivation



that proves $G = G[\varnothing]_S$ in GS. □

Theorem 7.2. *The rule* ss↑ *is admissible for* GS.

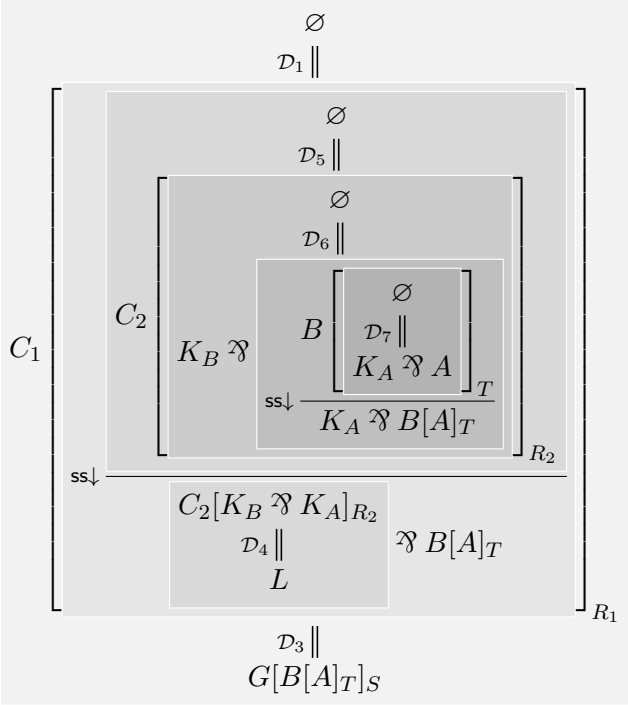
Proof. Assume we have a proof of $G[B \otimes A]_S$ in GS. By Lemma 6.4 we have a graph L and a context $C_1[\cdot]_{R_1}$, such that there are derivations

$$\begin{array}{c}\varnothing\\ \mathcal{D}_1 \,\|\, \mathsf{GS}\\ C_1\end{array}\quad,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_2 \,\|\, \mathsf{GS}\\ L \parr (B \otimes A)\end{array}\quad \text{and} \quad \begin{array}{c}C_1[L \parr X]_{R_1}\\ \mathcal{D}_3 \,\|\, \mathsf{GS}\\ G[X]_S\end{array}$$

for any graph X. We apply Lemma 6.1 to $L \parr (B \otimes A)$ and get K_B and K_A and a context $C_2[\cdot]_{R_2}$ such that

$$\begin{array}{c}C_2[K_B \parr K_A]_{R_2}\\ \mathcal{D}_4 \,\|\, \mathsf{GS}\\ L\end{array}\quad,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_6 \,\|\, \mathsf{GS}\\ K_B \parr B\end{array}\quad,\quad \begin{array}{c}\varnothing\\ \mathcal{D}_7 \,\|\, \mathsf{GS}\\ K_A \parr A\end{array}\quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_5 \,\|\, \mathsf{GS}\\ C_2\end{array}\ .$$

We can now give a proof of $G[B[A]_T]_S$ as follows:



for any $\varnothing \subseteq T \subset |V_B|$. □

Theorem 7.3. *The rule* $\mathsf{p}\uparrow$ *is admissible for* GS.

Proof. Assume we have a proof of $G[P(\!|M_1, \ldots, M_n|\!) \otimes P^\perp(\!|N_1, \ldots, N_n|\!)]_S$ with $M_1, \ldots, M_n$ non-empty graphs. We apply Lemma 6.4 and get a graph L and a context $C_1[\cdot]_{R_1}$, such that there are derivations

$$\begin{array}{c}\varnothing\\ \mathcal{D}_1 \,\|\, \mathsf{GS}\\ C_1\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_2 \,\|\, \mathsf{GS}\\ L \parr (P(\!|M_1, \ldots, M_n|\!) \otimes P^\perp(\!|N_1, \ldots, N_n|\!))\end{array} \quad \text{and} \quad \begin{array}{c}C_1[L \parr X]_{R_1}\\ \mathcal{D}_3 \,\|\, \mathsf{GS}\\ G[X]_S\end{array}$$

for any graph X.

If $P^\perp(\!|N_1, \ldots, N_n|\!) \neq \varnothing$ and we can apply Lemma 6.1 to $L \parr (P(\!|M_1, \ldots, M_n|\!) \otimes P^\perp(\!|N_1, \ldots, N_n|\!))$ and get graphs L_P and $L_{P^\perp}$ and a context $C_2[\cdot]_{R_2}$ such that

$$\begin{array}{c}C_2[L_P \parr L_{P^\perp}]_{R_2}\\ \mathcal{D}_4 \,\|\, \mathsf{GS}\\ L\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_5 \,\|\, \mathsf{GS}\\ L_P \parr P(\!|M_1, \ldots, M_n|\!)\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_6 \,\|\, \mathsf{GS}\\ L_{P^\perp} \parr P^\perp(\!|N_1, \ldots, N_n|\!)\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_7 \,\|\, \mathsf{GS}\\ C_2\end{array}.$$

Otherwise, if $P^\perp(\!|N_1, \ldots, N_n|\!) = \varnothing$ and by context reduction we already have a derivation $\mathcal{D}_2 = \mathcal{D}_5$ of $L \parr (P(\!|M_1, \ldots, M_n|\!) \otimes P^\perp(\!|N_1, \ldots, N_n|\!)) = L_P \parr P(\!|M_1, \ldots, M_n|\!)$. Hence we discuss this case together with the previous one by considering $C_2 = \varnothing$ and the derivations $\mathcal{D}_4$, $\mathcal{D}_5$ and $\mathcal{D}_6$ to be trivial.

Applying Lemma 6.2 to $L_P \parr P(\!|M_1, \ldots, M_n|\!)$ gives us two different cases.

(A) We get $K_1, \dots, K_n$ a context $C_3[\cdot]_{R_3}$ such that

$$\begin{array}{c} C_3[P^\perp(\!|K_1, \dots, K_n|\!)]_{R_3} \\ \mathcal{D}_8 \,\|\, \mathsf{GS} \\ L_P \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_i \,\|\, \mathsf{GS} \\ K_i \parr M_i \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_9 \,\|\, \mathsf{GS} \\ C_3 \end{array} \quad ,$$

for all $i \in \{1, \dots, n\}$. Then, our derivation of $G[(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)]_S$ is defined as follows

$$\begin{array}{l} \varnothing \\ \mathcal{D}_1 \,\| \\ C_1 \left[\begin{array}{l} \varnothing \\ \mathcal{D}_7 \,\| \\ C_2 \left[\begin{array}{l} \varnothing \\ \mathcal{D}_6 \,\| \\ L_{P^\perp} \parr P^\perp(\!| \dots, \ssdown\dfrac{N_1 \otimes \begin{array}{l}\varnothing \\ \mathcal{D}'_1 \,\| \\ K_1 \parr M_1\end{array}}{(M_1 \otimes N_1) \parr K_1}, \dots, \ssdown\dfrac{N_n \otimes \begin{array}{l}\varnothing \\ \mathcal{D}'_n \,\| \\ K_n \parr M_n\end{array}}{(M_n \otimes N_n) \parr K_n} \dots |\!) \end{array} \right]_{R_2} \end{array} \right]_{R_1} \\ \ssdown\!\!= \\ C_1 \left[\begin{array}{l} C_2 \left[\begin{array}{l} C_3[P^\perp(\!|K_1, \dots, K_n|\!)]_{R_3} \\ \mathcal{D}_8 \,\| \\ L_P \end{array} \right]_{R_2} \parr L_{P^\perp} \\ \mathcal{D}_4 \,\| \\ L \end{array} \right]_{R_1} \parr (M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n) \\ \mathcal{D}_3 \,\| \\ G[(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)]_S \end{array}$$

(B) We get H_X and H_Y and and $C_4[\cdot]_{R_4}$, such that

$$\begin{array}{c} C_4[H_X \parr H_Y]_{R_4} \\ \mathcal{D}_{10} \,\|\, \mathsf{GS} \\ L_P \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GS} \\ H_X \parr M_j \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GS} \\ H_Y \parr P(\!|M_1, \dots, M_{i-1}, \varnothing, M_{i+1}, \dots M_n|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_{11} \,\|\, \mathsf{GS} \\ C_4 \end{array}$$

for some $i \in \{1, \dots, n\}$.

(B.i) If $N_j \neq \varnothing$, then we apply Lemma 6.4 to the (GS-derivable) graph

$$H[N_j]_{S'} = L_P \parr P(\!|M_1, \dots, N_{j-1}, N_j, N_{j+1}, \dots, N_n|\!)$$

to obtain a context C_5 and a graph K_{N_j} such that

$$\begin{array}{c} C_5[K_{N_j} \parr X]_{R_5} \\ \mathcal{D}_{12} \,\|\, \mathsf{GS} \\ L_{P^\perp} \parr P^\perp(\!|N_1, \dots, N_{j-1}, X, N_{j+1}, \dots, N_n|\!) \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_{13} \,\|\, \mathsf{GS} \\ N_j \parr K_{N_j} \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_{14} \,\|\, \mathsf{GS} \\ C_5 \end{array}$$

Then, our proof of $G[(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)]_S$ is shown in Figure 5,

(B.ii) If $N_j = \varnothing$, then our proof of $G[(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)]_S$ is similar to the one of the previous case, by considering $C_5 = K_{N_j} = \varnothing$ (hence $\mathcal{D}_{12}$, $\mathcal{D}_{13}$ and $\mathcal{D}_{14}$ to be trivial), that is, as shown in Figure 6.

□

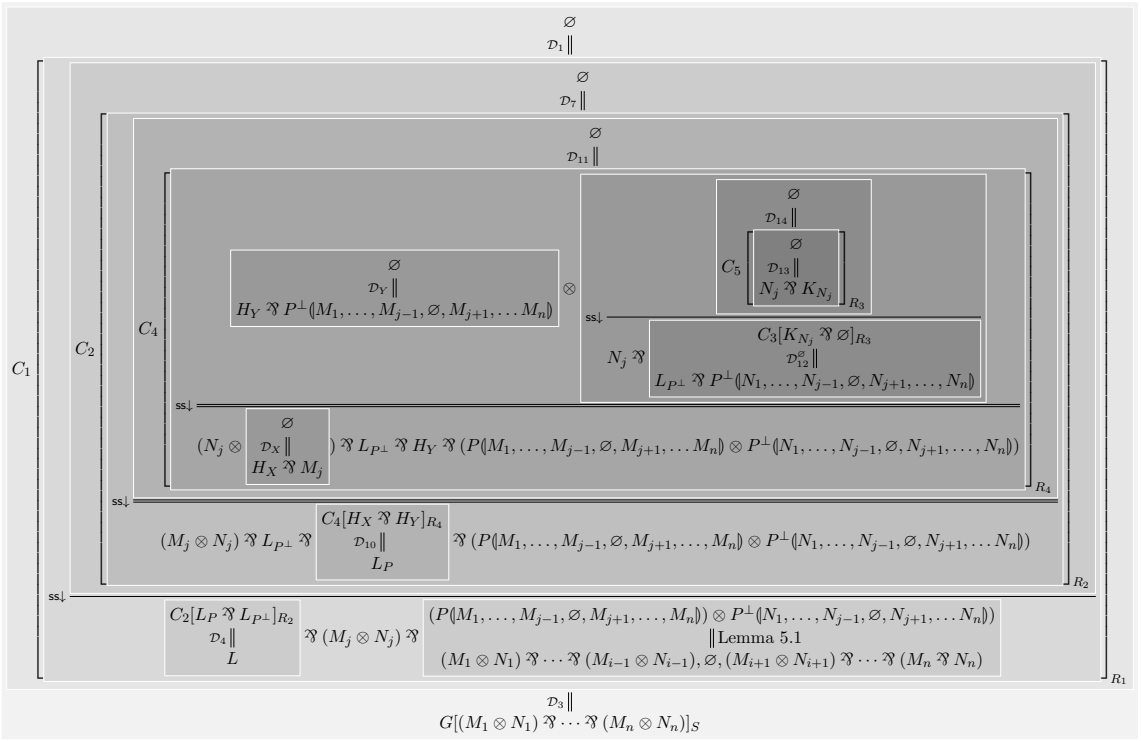


Figure 5: Derivation for case (B.i) in the proof of Theorem 7.3

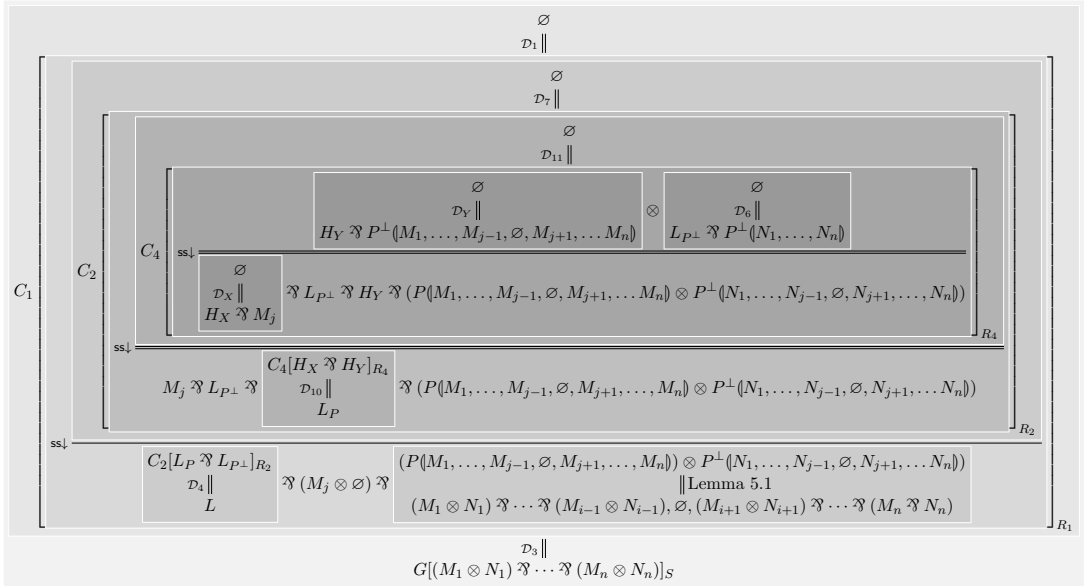
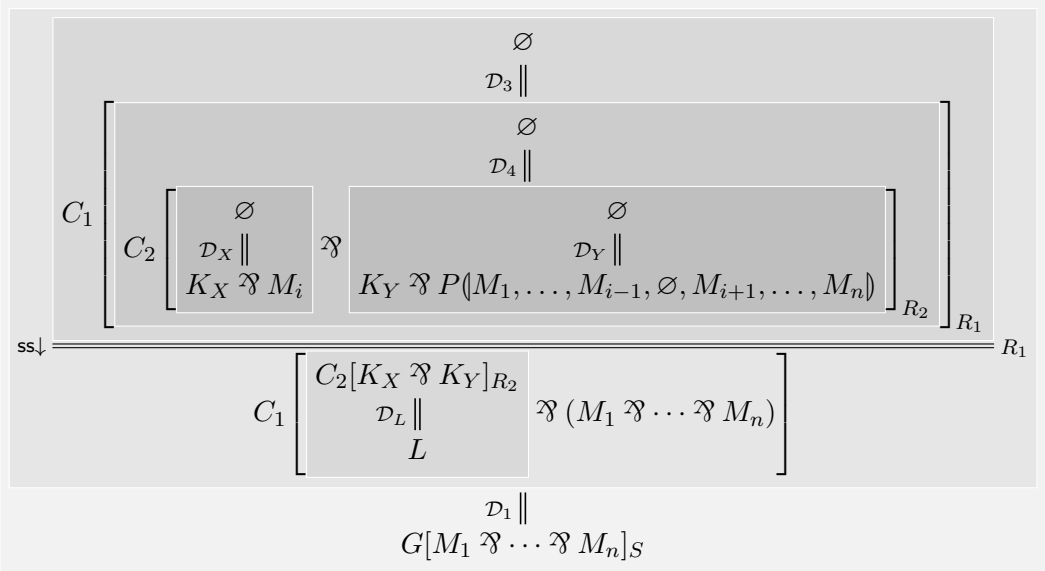


Figure 6: Derivation for case (B.ii) in the proof of Theorem 7.3 where $N_j = \varnothing$.

Because of the constraints on $\mathsf{p}\uparrow$, this rule can also be applied if $N_1, \ldots, N_n$ are empty. In these particular cases, the $\mathsf{p}\uparrow$-elimination procedure is simpler since some of the contexts and derivations involved in the previous above become respectively empty and trivial. In

particular, we need no more Lemma 5.1 to construct such derivations, as shown in the following derivation (case (B))



The three admissibility theorems in this section Theorem 7.1, Theorem 7.2 and Theorem 7.3 were stated previously as Theorem 5.5. Recall that from Theorem 5.5 and Lemma 5.2 we obtain the main result of this paper: the rule i↑, also known as the cut rule, is admissible as stated previously in Theorem 5.4. We have already stated some corollaries of cut elimination (Section 5), the rest of this paper will explore some further desirable consequences of cut elimination.

Alternative formulations of the rules of GS. Further admissibility results follow immediately now that we have established cut elimination, without having to pass via splitting and context reduction. We can in fact considerably strengthen Theorem 5.5, as follows.

Corollary 7.4. *Let* r *be an inference rule. If for every instance* $\mathsf{r}\dfrac{A}{B}$ *we have* $\vdash_{\mathsf{GS}} A \multimap B$ *then* r *is admissible for* GS.

Proof. Assume we have $\vdash_{\mathsf{GS}} A$. By Corollary 5.7 we have a derivation in SGS from A to B. Hence, we have $\vdash_{\mathsf{SGS}} B$, and therefore by Corollary 5.6 also $\vdash_{\mathsf{GS}} B$. □

Now consider the following two inference rules:

$$\mathsf{g}{\downarrow}\frac{(M_1 \parr N_1) \otimes \cdots \otimes (M_n \parr N_n)}{G(\!|M_1, \ldots, M_n|\!) \parr G^{\perp}(\!|N_1, \ldots, N_n|\!)} \qquad \mathsf{g}{\uparrow}\frac{G(\!|M_1, \ldots, M_n|\!) \otimes G^{\perp}(\!|N_1, \ldots, N_n|\!)}{(M_1 \otimes N_1) \parr \cdots \parr (M_n \otimes N_n)} \tag{7.1}$$

Their shape is similar to $\mathsf{p}{\downarrow}$ and $\mathsf{p}{\uparrow}$, but there are no side conditions, i.e., G can be any graph, and there are no emptiness or non-emptiness conditions for M_i or N_i.

Corollary 7.5. *The two rules* $\mathsf{g}{\downarrow}$ *and* $\mathsf{g}{\uparrow}$ *are admissible for* GS.

Proof. We are going to show that for all graphs G with $|V_G| = n$ and graphs $M_1, N_1, \ldots, M_n, N_n$ we have that

$$\vdash_{\mathsf{GS}} (M_1^{\perp} \otimes N_1^{\perp}) \parr \cdots \parr (M_1^{\perp} \otimes N_1^{\perp}) \parr G(\!|M_1, \ldots, M_n|\!) \parr G^{\perp}(\!|N_1, \ldots, N_n|\!) \quad . \tag{7.2}$$

Then this corollary follows via Corollaries 5.8 and 7.4. To prove (7.2), we use the following derivation

$$\dfrac{\dfrac{\varnothing}{\begin{array}{c}((M_1^\perp \otimes N_1^\perp) \parr M_1 \parr N_1) \otimes \cdots \otimes ((M_n^\perp \otimes N_n^\perp) \parr M_n \parr N_n) \\ \mathcal{D} \,\|\, \mathsf{GS} \\ G(\!|(M_1^\perp \otimes N_1^\perp) \parr M_1, \ldots, (M_n^\perp \otimes N_n^\perp) \parr M_n|\!) \parr G^\perp(\!|N_1, \ldots, N_n|\!)\end{array}}\, n{\times}\mathsf{i}{\downarrow}}{(M_1^\perp \otimes N_1^\perp) \parr \cdots \parr (M_n^\perp \otimes N_n^\perp) \parr G(\!|M_1, \ldots, M_n|\!) \parr G^\perp(\!|N_1, \ldots, N_n|\!)}\, n{\times}\mathsf{ss}{\downarrow}$$

where $\mathcal{D}$ exists by Lemma 5.1. To see this, observe that for each $i \in \{1, \ldots, n\}$, there are two cases: either $(M_i^\perp \otimes N_i^\perp) \parr M_i \neq \varnothing$ or $(M_i^\perp \otimes N_i^\perp) \parr M_i = \varnothing$. In the second case we also have $N_i^\perp = \varnothing$, and therefore also $N_i = \varnothing$. Hence, the condition for applying Lemma 5.1 is fulfilled. □

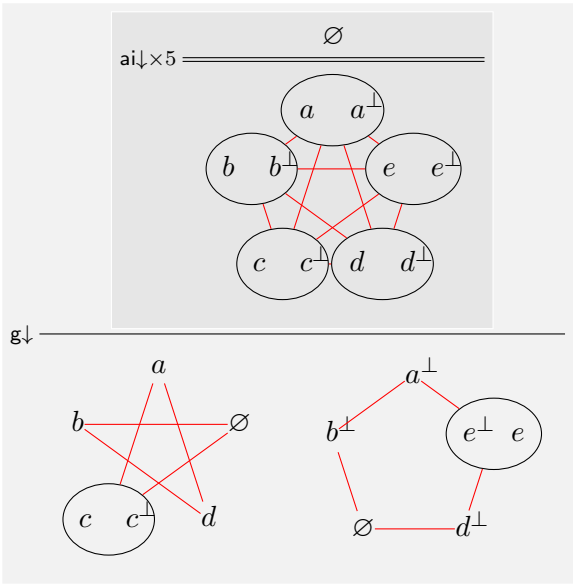
The above corollary, shows that a variant of GS where we use g↓ instead of p↓ is equivalent to the system GS using only p↓, as it is defined in Figure 1. Indeed, as noted previously in Section 4, a weaker constraint on the p↓ and p↑ was used in the conference version of this paper [AHS20], which did not insist that all of the components of one graph are non-empty when applying these rules. Thus we have tightened the proof search space of GS in this journal version, but have not changed the expressive power of the logic compared to the conference version.

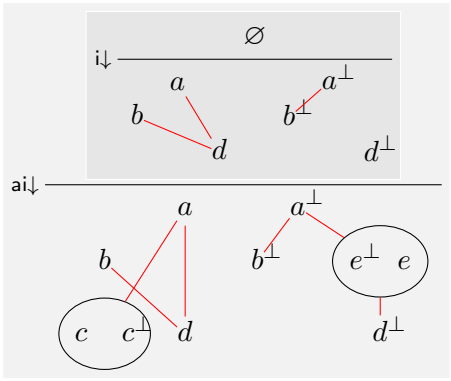
The reason why it is not immediately obvious whether the stronger side conditions in this version of paper can be used, is that Corollary 7.5 is a proper admissibility result rather than a derivability result. As shown in Lemma 5.1, if we drop the condition in the p↓ that P is prime then the rule for general graphs is derivable (that is we can produce a series of rules that have the effect of applying the rule to general graphs). However, the condition that all components of one graph are non-empty is more subtle, since the resulting variant of p↓ rules without that side-condition are not derivable using the rules of GS with the side conditions we fixed in Figure 1. For a counter example to the derivability of the unconstrained g↓ rule in GS, consider the proof on the left below and observe that the instance of g↓ cannot be

$$\mathsf{ax}\,\frac{}{a, a^{\perp}} \qquad \otimes\,\frac{\Gamma, \phi \quad \Delta, \psi}{\Gamma, \phi \otimes \psi, \Delta} \qquad \parr\,\frac{\Gamma, \phi, \psi}{\Gamma, \phi \parr \psi} \qquad \mathsf{mix}\,\frac{\Gamma \quad \Delta}{\Gamma, \Delta}$$

Figure 7: The inference rules of the system MLL°

derived using the rules of GS.





However, observe that there is the proof of the same conclusion on the right above, using the rules of GS with the correct side conditions. Corollary 7.5 proves that, although g↓ is not derivable in GS, it is admissible, which is, of course, a stronger property that ensures that if g↓ is used in a proof, then the proof could be rewritten using only the rules of GS.

8. Conservativity

We are now able to show that GS is a conservative extension of unit-free multiplicative linear logic with mix (MLL°) [Gir87a]. The formulas of MLL° are as in (2.1), but without the unit, and inference rules of MLL° were presented in Figure 7 where Γ and Δ are ***sequents***, i.e. multisets of formulas, separated by commas. We write $\vdash_{\mathsf{MLL}^\circ} \Gamma$ if the sequent Γ is provable in MLL°, i.e. if there is a finite proof tree in MLL° with conclusion Γ, where every leaf of the tree is an axiom.

Lemma 8.1. *If $\vdash_{\mathsf{GS}} A$ and A is a cograph, then there is a derivation*

$$\begin{array}{c} \varnothing \\ \mathcal{D} \,\|\, \mathsf{GS} \\ A \end{array} \tag{8.1}$$

such that every graph occurring in $\mathcal{D}$ is a cograph.

Proof of Lemma 8.1. By way of contradiction, assume there is a cograph that is not provable with passing through a non-cograph. Let A be such a cograph of minimal size $\|A\|$ (see

Definition 5.13). The only way to create a non-cograph from a cograph while going up in a derivation is via the $\mathsf{ss}\downarrow$ as in

$$\mathsf{ss}\downarrow \frac{P(\!|M_1, \ldots, M_{i_1}, M_i, M_{i+1}, \ldots, M_n|\!)}{M_i \parr P(\!|M_1, \ldots, M_{i_1}, \varnothing, M_{i+1}, \ldots, M_n|\!)}$$

where M_i and $P(\!|M_1, \ldots, M_{i_1}, \varnothing, M_{i+1}, \ldots, M_n|\!)$ are cographs and $P(\!|M_1, \ldots, M_n|\!)$ is not. Without loss of generality, we assume $i = 1$. By minimality of A, we can assume that this $\mathsf{ss}\downarrow$ occurs as bottommost rule instance in $\mathcal{D}$, and we can also assume that it occurs in a shallow context, i.e., we have

$$A = G \parr M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)$$

for some G. Otherwise $A = G \parr C[M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)]_R$ for some nontrivial context $C[\cdot]_R$, and we could apply context reduction to get a K with $\vdash K \parr M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)$ contradicting the minimality of A. Hence, $\mathcal{D}$ is of shape

$$\mathsf{ss}\downarrow \frac{\begin{array}{c}\varnothing \\ \mathcal{D}' \Vert \mathsf{GS} \\ G \parr P(\!|M_1, M_2, \ldots, M_n|\!)\end{array}}{G \parr M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)}$$

and we apply splitting (Lemma 6.2) to $G \parr M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)$, yielding 2 possibilities, of which we show here only the first, the second one being simpler.

- there is a context $C[\cdot]_R$ and graphs K_1, …, K_n, such that

$$\begin{array}{c} C[P^\perp(\!|K_1, \ldots, K_n|\!)]_R \\ \mathcal{D}_G \Vert \mathsf{GS} \\ G \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \Vert \mathsf{GS} \\ K_i \parr M_i \end{array} \quad , \quad \begin{array}{c} \varnothing \\ \mathcal{D}_C \Vert \mathsf{GS} \\ C \end{array}$$

  for all $i \in \{1, \ldots, n\}$. If $|\mathcal{D}_G| = 0$ then $G = C[P^\perp(\!|K_1, \ldots, K_n|\!)]_R$ and we can have a derivation

$$\mathsf{ss}\downarrow \frac{C\left[P^\perp(\!|\begin{array}{c}\varnothing \\ \mathcal{D}_1 \Vert \mathsf{GS} \\ K_1 \parr M_1\end{array}, K_2, \ldots, K_n|\!)\right]_R}{C[P^\perp(\!|K_1, \ldots, K_n|\!)]_R \parr M_1} \parr P(\!|\varnothing, M_2, \ldots, M_n|\!) \tag{8.2}$$

  contradicting the minimality of A. If $|\mathcal{D}_G| \neq 0$ and there is a cograph G' occurring in $\mathcal{D}_G$, then G' has smaller size than G, contradicting the minimality of $G \parr M_1 \parr P(\!|\varnothing, M_2, \ldots, M_n|\!)$. If there is no smaller cograph G' occurring in $\mathcal{D}_G$ in, then the bottommost rule instance in $\mathcal{D}_G$ is a $\mathsf{ss}\downarrow$ creating a non-cograph. By the same argument as above, we can conclude that it must be in a shallow context. Then $G = K_i \parr C[P^\perp(\!|K_1, \ldots, K_{i-1}, \varnothing, K_{i+1}, \ldots, K_n|\!)]_R$. If $i = 1$ we apply $\mathsf{ss}\downarrow$ to move M_1 and K_1 inside $C[\cdot]_R$ and conclude by a similar reasoning as with (8.2). If $i \neq 1$ we have $P^\perp(\!|K_1, \ldots, K_{i-1}, \varnothing, K_{i+1}, \ldots, K_n|\!)$ and $P(\!|\varnothing, M_2, \ldots, M_n|\!)$ are cographs and again we get a contradiction to the minimality of A. □

Lemma 8.2. *Let A and B be cographs. Then*

$$\mathsf{ss}{\downarrow}\,\frac{A}{B} \quad \Longrightarrow \quad \begin{array}{l} A \\ \Big\| \{\mathsf{s}\} \\ B \end{array} \tag{8.3}$$

Proof. By Theorem 2.6, the graphs A and B are cographs iff there are formulas ϕ and ψ with $[\![\phi]\!] = A$ and $[\![\psi]\!] = B$. Now the statement follows from the corresponding statement for formulas (see e.g., Lemma 4.3.20 in [Str03]). □

Theorem 8.3. *Let A be a cograph. Then $\vdash_{\mathsf{GS}} A$ iff $\vdash_{\{\mathsf{ai}{\downarrow},\mathsf{s}\}} A$.*

Proof. The implication from right to left follows immediately from the fact that s is a special case of $\mathsf{ss}{\downarrow}$ (see Lemma 4.10). For the implication from left to right, apply Lemma 8.1 to get a derivation $\mathcal{D}$ that only uses cographs. Hence the rule $\mathsf{p}{\downarrow}$ is not used in $\mathcal{D}$. Therefore, by Lemma 8.2, we can get a derivation $\mathcal{D}'$ that uses the rules $\mathsf{ai}{\downarrow}$ and s. □

Corollary 8.4. *For any unit-free formula ϕ,*

$$\vdash_{\mathsf{MLL}^\circ} \phi \quad \Longleftrightarrow \quad \vdash_{\mathsf{GS}} [\![\phi]\!]$$

Proof. It has been shown before (see, e.g., [GS01, Str03] that a unit-free formula ϕ is provable in MLL° iff it is provable in $\{\mathsf{ai}{\downarrow},\mathsf{s}\}$ (note that in (4.10) we can have $B = \varnothing$). Now the statement follows from Theorem 8.3 and Theorem 2.6. □

Corollary 8.5. *Provability in GS is* **NP***-complete.*

Proof. Since MLL° is **NP**-complete, we can conclude from Corollary 8.4 that GS is **NP**-hard. Containment in **NP** has been proved in Theorem 5.15. □

9. Graphs as Generalised Connectives

In this paper we have defined a proof system on graphs starting form logical principles. In the previous Section 8 we have shown that our system is a conservative extension of multiplicative linear logic with mix MLL°: the logic MLL° is the restriction of GS to P_4-free graphs. Lemma 3.9 provides a modular decomposition result for graphs, which allows us to associate a modular decomposition-tree to a graph, in the same way we associate a formula-tree to a cograph. This suggests the use of graphs as connectives in order to define *generalized formulas* that are encoding graphs. Our choices to name the two graphs on two vertices $⅋$ and $\otimes$ (see 3.3) and to denote the graph operations of union and join with the same symbols (Definition 2.3) are coherent with this intent. In fact, according with the notation for the composition-via-graph (Definition 3.8) we have

$$G \otimes H = \otimes(\!|G, H|\!) \quad \text{and} \quad G ⅋ H = ⅋(\!|G, H|\!).$$

Definition 9.1 (Connective-as-graph)**.** Any graph G with $|V_G| = n$ *describes* an n-ary connective G.

It appears clear that this notion of connective goes beyond the one of *synthetic connective*, which is a general connective definable as composition of standard binary connectives (see e.g. [And92, Gir00, MP13]). In this section we discuss the exact relation between our *connectives-as-graphs* notion with the notion of *(multiplicative) generalized connectives* from [Gir87b, DR89, Mai19, AM20]. In particular we show that our notion fits the definition

of multiplicative connective, but is describes different mathematical objects from the ones known in the literature. For this, we first recall the notion of *multiplicative generalized connective* from the early work in linear logic [Gir87b, DR89], and their description as sets of partitions. Then we show that the two notions are different. More precisely, we show that

(1) our system proves different generalized formulas than MLL° extended with generalized connectives,
(2) the symmetries induced by our generalized connectives-as-graphs are different from the generalized connectives-as-partitions from [Gir87b, DR89, Mai19, AM20],
(3) the notion of *decomposable connective* differs in the two settings, and
(4) our generalized connectives-as-graphs do not suffer from the so called *packaging problem* [DR89].

Definition 9.2 (multiplicative generalized connective [Gir87b, DR89])**.** A *multiplicative generalized connective* is an n-ary connective C which admits a *linear* and *context-free* (introduction) rule in the sequent calculus, that is, a rule of the form

$$\frac{\vdash \Gamma_1, \phi_{i_1}, \dots, \phi_{j_1} \quad \cdots \quad \vdash \Gamma_k, \phi_{i_k}, \dots, \phi_{j_k}}{\vdash \Gamma, \mathsf{C}(\phi_1, \dots, \phi_n)} \quad \text{with } \Gamma = \Gamma_1 \uplus \cdots \uplus \Gamma_k \tag{9.1}$$

The linearity condition demands that each occurrence of a formula in Γ or sub-formula of the *active formula* $\mathsf{C}(\phi_1, \dots, \phi_n)$ occurs exactly once in the premise, while the context-freeness condition demands that the application of the rule does not depend on the shape of the premises.

By means of example, consider the connectives of linear logic. The multiplicative conjunction $\otimes$ and $\parr$ disjunction are multiplicative connectives, while the additive conjunction $\&$ and disjunction $\oplus$ of linear logic are not.

$$\otimes \frac{\vdash \Gamma_1, \phi \quad \vdash \Gamma_2, \psi}{\vdash \Gamma, \phi \otimes \psi} \text{ where } \Gamma = \Gamma_1 \uplus \Gamma_2 \qquad \parr \frac{\vdash \Gamma, \phi, \psi}{\vdash \Gamma, \phi \parr \psi} \qquad \& \frac{\vdash \Gamma, \phi \quad \vdash \Gamma, \psi}{\vdash \Gamma, \phi \,\&\, \psi} \qquad \oplus \frac{\vdash \Gamma, \phi}{\vdash \Gamma, \phi \oplus \psi}$$

In fact, each formula in Γ and the two sub-formula ϕ and ψ of the active formula occurs exactly once in the premises of the rules for $\otimes$ and $\parr$. On the contrary, the rule for $\&$ is not context-free since it can be applied only if the premises are of the form Γ, ϕ and Δ, ψ with $\Gamma = \Delta$, that is, the rule depends on the context. Moreover this rule is not linear since each occurrence of formula in Γ occurs twice in the premises. The rule for $\oplus$ is not linear since it produces a sub-formula ψ which does not occur in the premise.

Following Definition 9.2, our connectives-as-graphs (Definition 9.1) are multiplicative connectives. In fact, the n-ary connective described by a graph G with $|V_G| = n$ admits a unique sequent calculus (introduction) rule of the form

$$G \frac{\vdash \Gamma_1, \phi_1 \quad \cdots \quad \vdash \Gamma_n, \phi_n}{\vdash \Gamma, G(\!|\phi_1, \dots, \phi_n|\!)} \quad \text{with } \Gamma = \Gamma_1 \uplus \cdots \uplus \Gamma_n$$

as it is derivable in GS if we interpret formulas as graphs and sequents of formulas as their disjoint union, and "premises concatenation" as the joint of the sequents in the pemises, that is

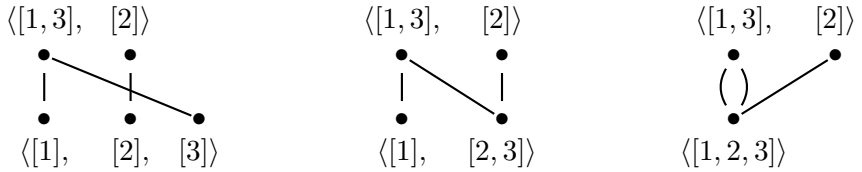
$$G \frac{(\llbracket \Gamma_1 \rrbracket \parr \llbracket \phi_1 \rrbracket) \otimes \cdots \otimes (\llbracket \Gamma_n \rrbracket \parr \llbracket \phi_n \rrbracket)}{\llbracket \Gamma_1 \rrbracket \parr \cdots \parr \llbracket \Gamma_n \rrbracket \parr G(\!|\llbracket \phi_1 \rrbracket, \dots, \llbracket \phi_n \rrbracket|\!)} \sim \begin{array}{c} (\llbracket \Gamma_1 \rrbracket \parr \llbracket \phi_1 \rrbracket) \otimes \cdots \otimes (\llbracket \Gamma_n \rrbracket \parr \llbracket \phi_n \rrbracket) \\ \mathcal{D} \,\|\, \mathsf{GS} \\ \mathsf{ss}{\downarrow} \dfrac{G^\perp(\!|\llbracket \Gamma_1 \rrbracket, \dots, \llbracket \Gamma_n \rrbracket|\!) \parr G(\!|\llbracket \phi_1 \rrbracket, \dots, \llbracket \phi_n \rrbracket|\!)}{\llbracket \Gamma_1 \rrbracket \parr \cdots \parr \llbracket \Gamma_n \rrbracket, \parr G(\!|\llbracket \phi_1 \rrbracket, \dots, \llbracket \phi_n \rrbracket|\!)} \end{array}$$

where $\mathcal{D}$ exists by Lemma 5.1.

Let us now recall the description of generalized connectives by means of *sets of partitions* from [DR89, Mai19, AM20]. We denote by $\mathbb{P}_n$ the set of partitions of the set $\{1,\dots,n\}$ and we say that an n-ary connective is described by a non-empty subset of $\mathbb{P}_n$. To improve readability, we follow the following policy for parenthesis: $\{\ \}$ for sets (of partitions), $\langle\ \rangle$ for partitions (i.e. family of disjointed covering sets of subsets of $\{1,\dots,n\}$) and $[\ ]$ for the elements of each partition (i.e. subsets of $\{1,\dots,n\}$). Each of these partitions can be interpreted intuitively as a way in which it is possible to gather the sub-formulas of the active formula into the premises of a multiplicative rule. By means of example, the binary rules for $\otimes$ and $\parr$ are described respectively by the (singleton) partition sets $\{\langle[1],[2]\rangle\}$ and $\{\langle[1,2]\rangle\}$. If we define a generalized (synthetic) connective $\mathsf{C}(a,b,c) = (a \otimes b) \parr c$, then we can associate to it the multiple rules corresponding to all possible ways to produce this formula in multiplicative linear logic:

$$\dfrac{\dfrac{\vdash a, c \quad \vdash b}{\vdash (a \otimes b), c}\otimes}{\vdash (a \otimes b) \parr c}\parr \rightsquigarrow \mathsf{C}^{[1,3],[2]}\ \dfrac{1,3 \quad 2}{\mathsf{C}(1,2,3)} \rightsquigarrow \langle[1,3],[2]\rangle \qquad \dfrac{\dfrac{\vdash a \quad \vdash b, c}{\vdash (a \otimes b), c}\otimes}{\vdash (a \otimes b) \parr c}\parr \rightsquigarrow \mathsf{C}^{[1],[2,3]}\ \dfrac{1 \quad 2,3}{\mathsf{C}(1,2,3)} \rightsquigarrow \langle[1],[2,3]\rangle$$

Because of De Morgan duality, we need for every connective a dual connective. The interactions of dual connectives by means of the cut-rule, together with the resulting cut-elimination procedure, enforces some additional structure between the sets of partitions. Given two partitions $p, q \in \mathbb{P}_n$, we define their *incidence graph* as the multi-graph whose vertices are the elements of p and q, such that there is an edge between two different vertices for every element in their intersection. For example, we have the following incidence graphs between $p = \langle[1,3],[2]\rangle$ and each of $q_1 = \langle[1],[2],[3]\rangle$, $q_2 = \langle[1],[2,3]\rangle$ and $q_3 = \langle[1,2,3]\rangle$:



We say that two partitions $p, q \in \mathbb{P}_n$ are *orthogonal* if their incidence graph is connected and acyclic. In the example above, p and q_2 are orthogonal, but p and q_1 are not (because the incidence graph is not connected) and p and q_3 are not (because the incidence graph is not acyclic). We say tha $P, Q \subset \mathbb{P}_n$ are *orthogonal* if every partition in P is orthogonal to every partition in Q.

Definition 9.3 (Connective-as-partitions)**.** Let P be a set of partitions in $\mathbb{P}_n$ such that there is a set of partitions $Q \subset \mathbb{P}_n$ that is orthogonal to P. An n-ary connective C *is described* by P if C admits exactly one multiplicative inference rule C^p of shape (9.1) for each $p \in P$, such that any two subformulas ϕ_i and ϕ_j of the principal formula $\mathsf{C}(\phi_1,\dots,\phi_n)$ belong to the same premise iff i and j belong to a same element in p.

For the purpose of this section, we consider the pair of dual generalized 4-ary connectives first introduced in [Gir87b] by means of permutations and later reformulated in [DR89] by means of partitions. Following the formalism in [AM20], we denote by $\mathsf{G_4}$ and $\mathsf{G_4^\perp}$ the connectives respectively described by the following two sets of partitions in $\mathbb{P}_4$

$$\mathsf{G_4}\colon \{\langle[1,2],[3,4]\rangle, \langle[1,4],[2,3]\rangle\} \quad \text{and} \quad \mathsf{G_4^\perp}\colon \{\langle[1,3],[2],[4]\rangle, \langle[2,3],[1],[3]\rangle\} \tag{9.2}$$

That is, for the connective $\mathsf{G_4}$ we have the following two inference rules

$$\mathsf{G_4}^{[1,2],[3,4]}\,\frac{\Gamma,\phi,\psi \qquad \Delta,\rho,\chi}{\Gamma,\Delta,\mathsf{G_4}(\phi,\psi,\rho,\chi)} \qquad \mathsf{G_4}^{[1,4],[2,3]}\,\frac{\Gamma,\phi,\chi \qquad \Delta,\psi,\rho}{\Gamma,\Delta,\mathsf{G_4}(\phi,\psi,\rho,\chi)} \tag{9.3}$$

Let $\mathcal{C}$ be the set of non-decomposable (multiplicative) connectives-as-partitions. We call a *generalized formula* a formula that is generated by a countable set of positive or negative propositional atoms $\{a, a^\perp, b, b^\perp, \ldots\}$ via the following grammar:

$$\phi, \psi ::= a \mid a^\perp \mid \phi \parr \psi \mid \phi \otimes \psi \mid \mathsf{C}(\phi_1, \ldots, \phi_n)$$

where $\mathsf{C} \in \mathcal{C}$ is any n-ary connective. We denote by $\mathsf{MLL}^\circ_{\mathcal{C}}$ the proof system over generalized formulas extending MLL° with, for each $\mathsf{C} \in \mathcal{C}$, all the sequent rules of C.

Comparing Theorems. We can now prove that $\mathsf{MLL}^\circ_{\mathcal{C}}$ and GS deal with different objects by showing that the connective $\mathsf{G_4}$ does not correspond to any instance of the connective P_4, which is described by the only prime graph with 4 vertices. From now on, we refer to $\mathsf{P}_4(\!|a, b, c, d|\!)$ as the graph a—b—c—d.

Proposition 9.4. *In $\mathsf{MLL}^\circ_{\mathcal{C}}$, none of the following formulas or their equivalent sequents are provable:*

$c \otimes (d \parr (a \otimes b)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), c^\perp \parr (d^\perp \otimes (a^\perp \parr b^\perp))$
$d \otimes (c \parr (a \otimes b)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), d^\perp \parr (c^\perp \otimes (a^\perp \parr b^\perp))$
$a \otimes (c \parr (b \otimes d)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), a^\perp \parr (c^\perp \otimes (b^\perp \parr d^\perp))$
$c \otimes (a \parr (b \otimes d)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), c^\perp \parr (a^\perp \otimes (b^\perp \parr d^\perp))$
$a \otimes (d \parr (b \otimes c)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), a^\perp \parr (d^\perp \otimes (b^\perp \parr c^\perp))$
$d \otimes (a \parr (b \otimes c)) \multimap \mathsf{G_4}(a,b,c,d)$	$\mathsf{G_4}(a,b,c,d), d^\perp \parr (a^\perp \otimes (b^\perp \parr c^\perp))$

(9.4)

Proof. We show that the first sequent is not provable. The same reasoning applies to the other sequents. In a derivation of $\mathsf{G_4}(a,b,c,d), c^\perp \parr (d^\perp \otimes (a^\perp \parr b^\perp))$ we can apply as first rule a $\parr$. Any derivation continuing with a $\otimes$ rule must have in one branch $\mathsf{G_4}(a,b,c,d)$ and either $d^\perp$ or $a^\perp \parr b^\perp$. This makes it impossible to match all pairs of dual atoms in axioms. Similarly, starting with one of the two $\mathsf{G_4}$ rules in 9.3, or starting with the $\parr$ rule followed by one of these two rules, would occur in the same problem of mismatched atoms. □

Proposition 9.5. *In GS, the graphs corresponding to the following formulas are provable:*

$c \otimes (d \parr (a \otimes b)) \multimap \mathsf{P}_4(\!|a,b,c,d|\!)$	$\mathsf{P}_4(\!|a,b,c,d|\!), c^\perp \parr (d^\perp \otimes (a^\perp \parr b^\perp))$
$c \otimes (d \parr (a \otimes b)) \multimap \mathsf{P}_4(\!|b,a,c,d|\!)$	$\mathsf{P}_4(\!|b,a,c,d|\!), c^\perp \parr (d^\perp \otimes (a^\perp \parr b^\perp))$
$d \otimes (c \parr (a \otimes b)) \multimap \mathsf{P}_4(\!|b,a,d,c|\!)$	$\mathsf{P}_4(\!|b,a,d,c|\!), d^\perp \parr (c^\perp \otimes (a^\perp \parr b^\perp))$
$d \otimes (c \parr (a \otimes b)) \multimap \mathsf{P}_4(\!|a,b,d,c|\!)$	$\mathsf{P}_4(\!|a,b,d,c|\!), d^\perp \parr (c^\perp \otimes (a^\perp \parr b^\perp))$
$a \otimes (c \parr (b \otimes d)) \multimap \mathsf{P}_4(\!|c,a,d,b|\!)$	$\mathsf{P}_4(\!|c,a,d,b|\!), a^\perp \parr (c^\perp \otimes (b^\perp \parr d^\perp))$
$a \otimes (c \parr (b \otimes d)) \multimap \mathsf{P}_4(\!|c,a,b,d|\!)$	$\mathsf{P}_4(\!|c,a,b,d|\!), a^\perp \parr (c^\perp \otimes (b^\perp \parr d^\perp))$
$c \otimes (a \parr (b \otimes d)) \multimap \mathsf{P}_4(\!|a,c,b,d|\!)$	$\mathsf{P}_4(\!|a,c,b,d|\!), c^\perp \parr (a^\perp \otimes (b^\perp \parr d^\perp))$
$c \otimes (a \parr (b \otimes d)) \multimap \mathsf{P}_4(\!|a,c,d,b|\!)$	$\mathsf{P}_4(\!|a,c,d,b|\!), c^\perp \parr (a^\perp \otimes (b^\perp \parr d^\perp))$
$a \otimes (d \parr (b \otimes c)) \multimap \mathsf{P}_4(\!|c,b,a,d|\!)$	$\mathsf{P}_4(\!|c,b,a,d|\!), a^\perp \parr (d^\perp \otimes (b^\perp \parr c^\perp))$
$a \otimes (d \parr (b \otimes c)) \multimap \mathsf{P}_4(\!|b,c,a,d|\!)$	$\mathsf{P}_4(\!|b,c,a,d|\!), a^\perp \parr (d^\perp \otimes (b^\perp \parr c^\perp))$
$d \otimes (a \parr (b \otimes c)) \multimap \mathsf{P}_4(\!|a,d,c,b|\!)$	$\mathsf{P}_4(\!|a,d,c,b|\!), d^\perp \parr (a^\perp \otimes (b^\perp \parr c^\perp))$
$d \otimes (a \parr (b \otimes c)) \multimap \mathsf{P}_4(\!|a,d,b,c|\!)$	$\mathsf{P}_4(\!|a,d,b,c|\!), d^\perp \parr (a^\perp \otimes (b^\perp \parr c^\perp))$

(9.5)

Proof. The proof of $(d \parr (b \otimes (a \parr c))) \multimap \mathsf{P}_4(\!|c, a, d, b|\!)$ is shown in 4.9. The other implications are proven similarly. □

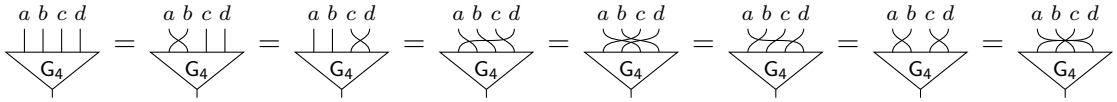
Both $\mathsf{MLL}^\circ_{\mathcal{C}}$ (by definition) and GS (by Corollary 8.4) are conservative extensions of MLL. The non $\otimes/\parr$-decomposable 4-ary connective $\mathsf{G_4}$ should be translated as a non-decomposable graph with 4 vertices, that is, a P_4. Thus, the translation of $\mathsf{G_4}(a, b, c, d)$ should be of the shape $\mathsf{P}_4(\!|\sigma(a), \sigma(b), \sigma(c), \sigma(d)|\!)$ for some permutation σ over the set $\{a, b, c, d\}$. However for each possible translation of $\mathsf{G_4}$, one of the implications in 9.4, which are not derivable in $\mathsf{MLL}^\circ_{\mathcal{C}}$ and contain $\mathsf{G_4}$, is translated into a GS-derivable implication. We can conclude that the systems $\mathsf{MLL}^\circ_{\mathcal{C}}$ and GS are not equivalent.

Symmetries of a connective. We now calculate the symmetries of the connectives $\mathsf{G_4}$ and P_4 and show that they are different in a way such that GS and $\mathsf{MLL}^\circ_{\mathcal{C}}$ are not comparable as logical systems. To understand what we mean by symmetries of a connective, consider for example the $\otimes$, which is commutative—as the formula $A \otimes B$ is logically equivalent to $B \otimes A$—that is, we can permute A and B without changing the formula. On the other hand, the connective $\multimap$ is not commutative. But we can define a connective $\circ\!\!-$ distinct from $\multimap$ but definable from $\multimap$ as $A \circ\!\!- B = B \multimap A$.

Once we start to analyse n-ary connectives with $n > 2$, we can no more categorize connectives only as commutative and non-commutative. Consider the 4-ary connectives $\mathsf{G_4}$ and $\mathsf{G_4^\perp}$ and their describing sets of partitions in 9.2. We have

$$\mathsf{G_4}(a, b, c, d) = \mathsf{G_4}(b, a, c, d) = \mathsf{G_4}(a, b, d, c) = \mathsf{G_4}(b, c, d, a) =$$
$$\mathsf{G_4}(c, d, a, b) = \mathsf{G_4}(d, a, b, c) = \mathsf{G_4}(b, a, d, c) = \mathsf{G_4}(d, c, b, a)$$

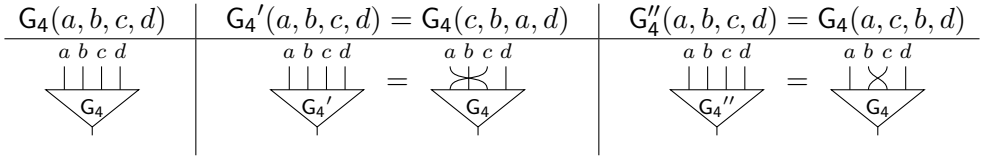
which, using interaction net syntax [Laf95], could be represented as follows



This follows the fact that the sets of partitions describing $\mathsf{G_4}$ and $\mathsf{G_4^\perp}$ are *stable* under the permutation[8] in the set $\mathfrak{S}(\mathsf{G_4})$, shown below:

$$\mathfrak{S}(\mathsf{G_4}) = \mathfrak{S}(\mathsf{G_4^\perp}) = \{(1), (1,2), (3,4), (1,2,3,4), (1,3)(2,4), (1,4,2,3), (1,2)(3,4), (1,4)(2,3)\}$$

We conclude that there are $\frac{|\mathfrak{S}_4|}{|\mathfrak{S}(\mathsf{G_4})|} = 3$ non-isomorphic instances of $\mathsf{G_4}$ defining as many 4-ary connectives:



Moreover, we observe that $\mathsf{G_4^\perp}$ cannot be expressed as a function of $\mathsf{G_4}$ since the two describing sets are made of partitions of different cardinality. The connective $\mathsf{G_4}$ is a non-decomposable 4-ary connective [Mai19, AM20]. We can conclude that there are only three pairs of dual non-decomposable 4-ary connectives-as-partitions. In fact, a properly defined duality forces the dual of a set P of partitions to be exactly the set all partitions that are orthogonal to

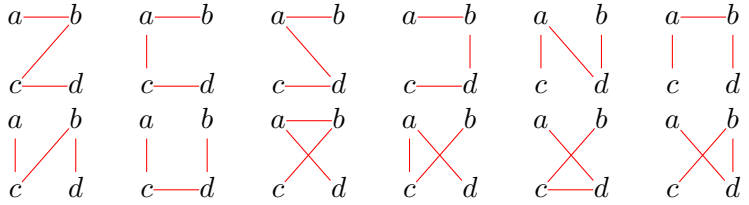
[8]We write permutations using the *cycle notation*, that is, the permutation is written as a product of cycles, and each element is mapped to the following element in the same cycle.

the ones in P. This restrains the subsets of $\mathbb{P}_4$ which can be used to describe connectives, leaving only to 6 non-decomposable connectives.

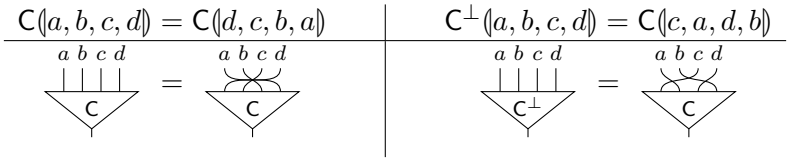
At the same time, P_4 is the unique prime graph on four vertices, hence any non-decomposable 4-ary connective-as-graph has to be an instance of P_4. Since the symmetry group of P_4, that is, the group of its isomorphisms is the following

$$\mathfrak{S}(\mathsf{P}_4) = \{(1), (1,4)(2,3)\}$$

we conclude that there are $\frac{|\mathfrak{S}_4|}{|\mathfrak{S}(\mathsf{P}_4)|} = 12$ different instances of P_4 defining as many 4-ary connectives



If we denote $\mathsf{Z}(\!|a,b,c,d|\!) = \mathsf{P}_4(\!|a,b,c,d|\!)$ and $\mathsf{N}(\!|a,b,c,d|\!) = \mathsf{P}_4(\!|c,a,d,b|\!)$ (the first and second-to-last graph in the first line in the above figure), we have that $\mathsf{Z}^\perp = \mathsf{N}$. More generally, if $\mathsf{C}(\!|a,b,c,d|\!) = \mathsf{P}_4(\!|\sigma(a),\sigma(b),\sigma(c),\sigma(d)|\!)$ for a permutation σ of the set $\{a,b,c,d\}$, then



We conclude that there are 6 pairs of dual non-decomposable 4-ary connectives-as-graphs.

Summing up, the symmetry group of G_4 is different from the one of P_4. Hence, the two connectives have to be distinct. Moreover $\mathsf{G}_4^\perp(a,b,c,d)$ cannot be expressed as $\mathsf{G}_4(\sigma(a),\sigma(b),\sigma(c),\sigma(d))$ for any σ permutation over $\{a,b,c,d\}$, while $\mathsf{P}_4(\!|a,b,c,d|\!) = \mathsf{P}_4^\perp(\!|b,d,a,c|\!)$. In $\mathsf{MLL}^\circ_{\mathcal{C}}$ there are 3 pairs of dual non-decomposable 4-ary connectives, in GS there are 6 of such pairs.

Decomposable and non-decomposable connectives. In the connectives-as-partitions setting, a generalized n-ary connective described by a partition $P \subset \mathbb{P}_n$ is *decomposable* if there is a $\otimes/\parr$-formula F such that P is the set of partitions associated to all possible derivations of F [DR89, AM20]. For example, the connective described by $P = \{\langle[1,2],[2]\rangle, \langle[1],[2,3]\rangle\}$ is decomposable since it is associated to the formula $F = (a \otimes b) \parr c$ as shown above. However, the generalized connective which corresponds to the formula $\mathsf{G}_4(1,2,3,\mathsf{G}_4(4,5,6,7))$ is considered to be non-decomposable, even if we define it by using other connectives. A possible motivation for such a choice may be the lack of an efficient decomposition algorithm in the literature. In fact, even from the set of partitions defined by a $\otimes/\parr$-formula, it is not trivial to backtrack the original formula.

However, for our connectives-as-graphs setting, Lemma 3.9 provides a finer definition of connective decomposition. In particular, every prime graph defines a non-decomposable connective, and every connective is either non-decomposable or admits a decomposition into non-decomposable ones. That is, while the notion of decomposition for connectives-as-partitions is still rough, the one for connectives-as-graph is well studied and comes with a linear factorization algorithm [HdMP04].

The packaging problem. The so called *packaging problem* [DR89] is the impossibility of deriving the axiom $\mathsf{G}_4 \multimap \mathsf{G}_4$ in the sequent calculus. This is due to the lack of the *initial coherence* property [AL01, MP13] for the $\mathsf{MLL}^\circ_{\mathcal{C}}$ sequent calculus. This can easily be shown in the case of G_4 by remarking that the arities of the rules for G_4 and $\mathsf{G}_4^\perp$ make it impossible to gather together the premises as four smaller proofs. The initial coherence property can be recovered using the proof net syntax of $\mathsf{MLL}^\circ_{\mathcal{C}}$ [AM20].

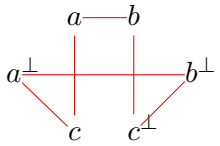
As shown in Lemma 5.2, the proof system GS has the initial coherence property, that is, there is no packaging problem in GS.

10. Related and Future Work

Here we draw attention to challenges surrounding GS. Using examples, such as (6.5) and (6.3), we have already explained why GS necessarily demands deep inference. Since no established deep inference system matches GS we have a fundamentally new proof system. Furthermore, we explain in this section that simply taking an established semantics for MLL° based on graphs and dropping the restriction to cographs does not immediately yield a semantics for GS.

Linear inferences in classical logic. The switch rule has the property that it reflects edges and maximal cliques. That is: if there is an edge in the conclusion it will also appear in the premise and every maximal clique in the premise is a superset of some maximal clique in the conclusion. Indeed, mappings reflecting maximal cliques and preserving stable sets (mutually independent vertices) have a long history in program semantics [Ber78] which led to coherence spaces and the discovery of linear logic [Gir87a]. Therefore it is a reasonable starting point to try generalising switch by using such maximal clique reflecting homomorphisms, instead of $\mathsf{ss}{\downarrow}$. Indeed this is how we discovered $\mathsf{ss}{\downarrow}$, which is sound with respect to such homomorphisms.

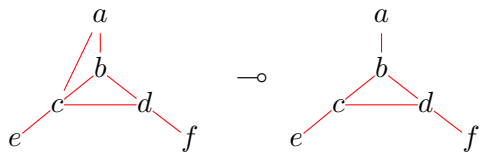
Unfortunately, replacing $\mathsf{ss}{\downarrow}$ with maximal clique reflecting homomorphisms yields a system distinct from our graphical system, for example the following would be provable, but is not provable in GS.



(10.1)

We may try replacing both $\mathsf{ss}{\downarrow}$ and $\mathsf{ss}{\uparrow}$ using a stronger symmetric notion of homomorphism where, in addition, every maximal stable set in the conclusion is a superset of some maximal stable set in the premise. When we use homomorphisms that are both maximal clique reflecting and stable set preserving as rules we call them *linear inferences* [DS15, DS16]. Using linear inferences, the above example is not provable. To see why, observe that at some point either a and $a^\perp$ or b and $b^\perp$ must be brought together into a module where they can interact, but this cannot be achieved while preserving the maximal stable set $\{a, b^\perp\}$.

Notice, however, that if we replace $\mathsf{ss}\downarrow$ and $\mathsf{ss}\uparrow$ by the linear inferences described above, the implication below would be provable.



(10.2)

In contrast, the above is not provable in GS, since both sides are distinct prime graphs; and there is no suitable way to apply $\mathsf{ss}\downarrow$. Thus, we would obtain a distinct system from GS by using such homomorphisms.

Below we provide two smaller examples that are provable using linear inferences and $\mathsf{ai}\downarrow$, neither of which are provable in GS.

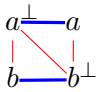
(10.3)

All the above separating examples made use of graphs that are not cographs – the smallest of which consisting of 6 vertices. The smallest example based on formulas (i.e., cographs) that separates GS from logics based on linear inferences consists of 16 vertices.[9] Indeed, studying logics defined using linear inferences that reflect maximal cliques and preserve maximal stable sets is currently a topic of active research and leads to possible extensions of Boolean logic to graphs [Cal16, War19, CDW20].

Logics resulting from directly generalising linear inferences inspired by Boolean logic are incomparable to GS. In one direction, the above examples show there are graphs that are not provable in GS that are provable using linear inferences; while in the converse direction the $\mathsf{p}\downarrow$ rule is not admissible in a system with linear inferences. To see why, observe that in Example 4.14 there is a clique $\{a^\perp, d^\perp, c, b\}$ in the premise of the $\mathsf{p}\downarrow$ that is not reflected in the conclusion. Since $\mathsf{p}\downarrow$ is necessary for examples such as in Figure 2, we know that the logics are incomparable. Hence, although these proposed logics on graphs appear to be close and the arguments are the same, each proposal must be founded on distinct logical principles.

Criteria for proof nets. Graphical approaches to proof nets such as $R\&B$-graphs [Ret03] have valid definitions when we drop the restriction to cographs. However, we show that (at least without strengthening established criteria), such definitions do not yield a semantics for a logic over graphs, since logical principles laid out in the introduction are violated.

Consider again graph (4.3), which is not provable in GS. In an $R\&B$-graph we draw blue edges representing the axiom links of proof nets, as shown below for graph (4.3).



(10.4)

If we remove the restriction to cographs when we apply the established correctness criterion for $R\&B$-graphs the above graph would be accepted, which is something we aim to avoid

[9] A new linear inference of size 8: `https://prooftheory.blog/2020/06/25/new-linear-inference/` and Linear inferences of size 7: `https://prooftheory.blog/2020/10/01/linear-inferences-of-size-7/`

for a semantics of GS. The reason is the cycle of 4 vertices alternating between red and blue edges has a chord. This observation is independent of the rules of the system GS, as we observed in Section 4 that graph (4.3) cannot be provable in a system subject to the logical principle of consistency. Future work includes defining a stronger criterion for $R\&B$-graphs that works for graphs that are not cographs, with the aim of obtaining a sound and complete semantics for GS.

11. Requirements of an Analytic Proof System on Graphs

In this section, we reflect on design decisions in order to support our claim that we have defined a logic where we reason about graphs rather than formulas. We present here an argument that is independent from the proof system that we developed, thus it is not necessary for the reader to accept deep inference a priori in order to accept the set of theorems proven by the logic GS. We reinforce the message that, for this logic without precedent – that is without a pre-existing semantics or proof system, we had success designing and justifying our system when we started from logical principles. Our approach of designing from principles is credible, since, as highlighted in the discussion on related work, there does not appear to be an immediate generalisation of the semantics for some established logic on formulas to graph that yields a system with the logical properties we desire.

From the title, we desire:

An analytic propositional proof system on graphs.

To understand fully the above statement we explain two aspects of this statement. Firstly, we make precise what we mean by a *propositional proof system*, by means of logical principles that such a proof system should conform to. Secondly, we must explain what it means for such a propositional proof system to be *analytic*, particularly since traditional definitions of analyticity do not lift immediately to our setting. In the discussion that follows we show that all design decisions made are widely accepted in logic, making is difficult to argue that GS is not a logic.

Graph isomorphism for propositional proofs. In this study, we have restricted ourselves to simple undirected graphs (see Definition 2.1), where vertices are labelled with positive or negative propositional atoms, such as a and $a^{\perp}$. This allows us to align with existing graphical representations of formulas if we restrict to *cographs*, which are exactly those graphs generated by propositional formulas using the operations illustrated in (2.2).

The first logical assumption we make is that isomorphic graphs (see Definition 2.2) are logically equivalent, which we expect to hold for all graphical logics.

> **Extensionality requirement:** For pairs of isomorphic graphs G and H, we have $\vdash G \multimap H$ holds.

Graph isomorphisms allows us to rename the underlying vertices of a graph, while preserving labels. Indeed in our diagrams of graphs, extensionality is implicitly appealed to, since we only show the labels of vertices, i.e., we quotient graphs by label-preserving isomorphisms. The term "extensionality" is consistent with the idea that objects are equivalent if their externally visible properties, i.e., their labels and edges shown in diagrams, are the same.

This is one of the few principles we expect to be common to all logics on graphs. To reinforce this belief, we acknowledge there are schools of philosophy that maintain that it is possible that $A \neq A$ [Mat68]. The essence of such arguments is that if A is not well typed

or does not exist then $A \neq A$ is a reasonable conclusion. But notice that, firstly, this is different from saying that $A = A$ does not hold, and, secondly, a metaphysical discussions on types or existence is perpendicular to the logic in this work.

Even if we assume extensionality as our sole logical principle, since we aim to define a propositional proof system for our logic on graphs, we must take additional care to ensure that we satisfy the following principle.

Cook-Reckhow requirement: Every rule is checkable in polynomial time.

The above is a fundamental property of proof systems for propositional logic [CR79]. As mentioned in Observation 5.12, since checking an explicit isomorphism is in P but there are no algorithms for finding graph isomorphisms in P, a formulation of a propositional proof system on graphs must make the isomorphisms explicit in the proof system (see Equation 4.5).

Involutive negation and consistency. The first proper design decision we make is that we insist on having a logic featuring an involutive negation, as found in most classical and linear logics (but not intuitionistic logic of course).

De Morgan requirement: negation should be involutive.

Formally, an involution on graphs is a unary operator $(\ \cdot\ )^{\perp}$ that satisfies the property $(G^{\perp})^{\perp} = G$ for all graphs G. The assumption that we have an involutive negation means that De Morgan dualities hold for pairs of connectives on graphs we define (Observation 4.3). For a proof system on are graphs, there are only two possible choices for an involution, namely the identity function and the graph complement function.

The use of the identity function to define an involutive negation is ruled out by the logical principle of consistency, which is that not all graphs are provable. The formulation of consistency that we achieve for GS is as follows.

Consistency requirement: For non-empty graphs G, if $\vdash G$ then $\nvdash G^{\perp}$.

To see why, the above principle rules out the identity function as negation, observe that extensionality ensures that there are some provable graphs say $\vdash G$. If the negation is defined by the identity, then $\nvdash G$ holds, which violates the above consistency requirement. Thereby, the assumptions thus far fix negation as *graph complement* (Definition 4.1). Observe also that the consistency of GS is indeed a corollary of cut elimination (Corollary 5.11).

Implication. Our next proper design decision is to materialise implication in our logic. We materialise implication "G implies H" as "not G or H", for some notion of disjunction, as in logics such as classical and linear logics. Having already fixed negation as graph complement, must first make a design decision in order to define disjunction. In order for there to be a single implication, the disjunction used must be commutative (implications materialised using non-commutative disjunction lead to a distinct left and right implication [Lam61]).

Commutativity requirement: disjunction is commutative and associative.

While there may be several elaborate choices for defining disjunction[10], we make the design decision that disjunction is disjoint union of graphs. This design decision aligns with an established culture of using cographs to represent formulas [Duf65, Ret03]. Note that, by

[10]For an example, consider the way multiplicative disjunction is defined on coherent spaces [GLT89].

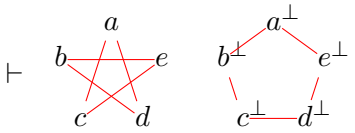
symmetry, we could have selected graph join as disjunction, which would simply interchange edges and non-edges throughout this paper without changing the meaning of the logic.

We are now able to materialise implication by means of negation and disjunction, as in classical and linear logic. More precisely, we have the following principle.

Material requirement: Implication $G \multimap H$ is defined as $G^{\perp} \parr H$.

This assumption should not be taken for granted, since various logics such as intuitionistic logic cannot materialise implication in this way. Also, this is not a guaranteed property of a logic satisfying De Morgan properties, since even if a disjunction is present it may not be the right disjunction to internalise negation. For example, linear implication cannot be materialised using additive disjunction and linear negation, hence the additive fragment of linear logic [Gir87a] has no material implication.

We can prove now further examples. By extensionality we have implication $\vdash a \multimap a$, where a is the singleton graph. Hence we know that $\vdash a^{\perp} \parr a$ should hold in our logic. This of course, agrees with the majority of proof systems. Moving beyond graphs that correspond to traditional formulas, we also have theorems such as the following, by the same reasoning, i.e., the two disjoint sub-graphs are dual.



For almost all logical systems, we require that implication is transitive, since transitivity, known since antiquity as the *hypothetical syllogism*, enables us to perform deduction.

Transitivity requirement: If $\vdash F \multimap G$ and $\vdash G \multimap H$, then $\vdash F \multimap H$.

Notice that transitivity allows us to apply *modus ponens*. To be explicit, observe that the empty graph (denoted $\varnothing$) is the unit of the disjoint union operation on graphs. Furthermore $\varnothing$ is self-dual. By the definition of implication, we have that $\vdash G$ holds whenever $\vdash \varnothing \multimap G$ does. Thereby, if we assume $\vdash G$ and $\vdash G \multimap H$ hold, then by the definition of implication $\vdash \varnothing \multimap G$ holds and by transitivity $\vdash \varnothing \multimap H$ holds. By the same argument we can conclude that $\vdash H$ holds. This is exactly modus ponens.

Like consistency, transitivity of implication is a default decision, although we acknowledge that transitivity can fail for some logics featuring negation-as-failure [BM90].

The final feature we add is the context-free assumption. For a logic with formulas this is the assumption that implication is preserved in all contexts, where a context is a formula with a hole in which another formula can be plugged, such as $\phi \otimes [\cdot]$. For example, if $\psi \multimap \chi$, then $\phi \otimes \psi \multimap \phi \otimes \chi$. More specifically, we mean all positive contexts, since negations reverse the direction of implication. However, in a logic satisfying the assumptions we have made, with De Morgan dualities and material implication, we can always reduce to a negation-normal-form, where all negations are pushed to the atoms, i.e., the labels on the vertices; thereby allowing us to range over all contexts. Similarly, in modal logics extending K we have that if $\psi \multimap \chi$ then $\Box\psi \multimap \Box\chi$. The same holds for the diamond modality. Indeed, this holds even for non-classical modal logics [MS14].

When we move from formulas to graphs the notion of a context must be generalised, where the obvious notion was introduced in Notation 3.2. In the broader philosophy of language (programming languages and natural languages), context freedom is not guaranteed, since the meaning of something may change in different contexts. However, logics are usually

designed such that if there is some context violating an implication, then implication does not hold; situations in knowledge representation where this fails is usually due to using moving between systems [Gab98]. Since we are hard pressed to find a recognisable logical system on formulas that does not satisfy context freedom, then we insist on the following principle holding for graphs.

> **Context-free requirement**: For contexts $C[\cdot]$, if $\vdash G \multimap H$, then $\vdash C[G] \multimap C[H]$.

The above formula allows ensures that if we prove a theorem it holds in any context. The above property is exploited explicitly in some proof systems to allow rules to be applied deep inside any module of a graph, which is the technique called *deep inference* employed in this paper. However, the above principle is not specific to deep inference.

Thus we expect the following to be a theorem of a logic satisfying the above principles.

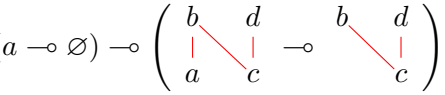
$$\vdash (G \multimap H) \multimap (C[G] \multimap C[H])$$

From the above, we can establish non-trivial facts that we expect to hold for a logic on graphs, that are beyond the scope of formulas. Consider for example when we instantiate the above theorem as follows.

$$G \triangleq a^{\perp} \qquad H \triangleq \varnothing \qquad C[\,] \triangleq$$

From the above we instantiation, we obtain the following theorem.

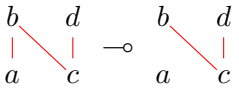
$$\vdash (a \multimap \varnothing) \multimap \Bigg($$



$$\Bigg)$$

Reorganising, according to the definitions of implication and negation that we have fixed we obtain one of our running examples (recall proofs 4.6, 4.9 and 6.2).

$\vdash$



The above proposition would hold for any logic satisfying the principles laid out in this section. There may be more propositions that hold other than those that are enforced by these principles; for instance, if we induce more principles from classical logic, we could prove strictly more theorems. However, in this work, we make the design decision of aiming for the minimal system, which we state as a concluding principle that fixes our target logic.

> **Minimality requirement:** No further propositions hold, other than those forced by the other requirements.

We have remarked throughout this section that GS satisfies all the principles laid out, mainly as a consequence of cut elimination. Minimality then follows from observing that all rules of GS are sound with respect to the same principles.

On analyticity. Throughout this paper we provide evidence that we can design *propositional proof systems* that achieve the above stated requirements. However, designing a proof system is not the main challenge. The main challenge is to design an *analytic* propositional proof system.

Analyticity, the idea that propositions contains all the information required to in order to judge their validity with respect to some logical system, has long been debated by

philosophers. The design of modern analytic proof calculi is widely considered to begin with the *sequent calculus*, as developed in 1935 [Gen35a, Gen35b], with improvements incorporated by Girard [Gir87a].

The rules of the sequent calculus, such as those for MLL° presented in Figure 7, are considered to be analytic since they satisfy the *subformula property*. The subformula property states that in each rule, every formula that occurs in one premise sequent, occurs as a subformula in the conclusion sequent. The subformula property facilitates proof search, since there are only finitely many subformulas to consider during proof search. The rules in Figure 7 clearly satisfy the subformula property; whereas the cut rule, below, which generalises transitivity and hence also *modus ponens*, does not in general satisfy the subformula property.

$$\mathsf{cut}\,\frac{\Gamma, \phi \quad \phi^\perp, \Delta}{\Gamma, \Delta}$$

If we start with some conclusion Γ, Δ and try to apply the cut rule above, there are infinitely many formula ϕ to chose from. Similarly, the transitivity assumption requires insight external to the system to know which formulas to introduce. Thus the subformula property is effectively avoiding infinite branching in the proof search space. This is a fundamental reason for proving cut elimination when designing a proof system: on one hand, we aim to ensure that the basic principles of deduction such as modus ponens may be applied; on the other hand, we also want to show that deductive rules such as cut and modus ponens, which are not well-behaved with respect to proof search are *admissible*.

The sub-formula property does not lift immediately to the setting of deep inference. Since deep inference is necessary for GS, we make use of an alternative definition of analyticity [BG09, BG16].

Analyticity requirement: For graph G and rule r, there is an n such that,

for all contexts $C[\cdot]$, there are at most n graphs H such that $\mathsf{r}\,\dfrac{C[H]}{C[G]}$.

This ensures a rule is guaranteed to be finitely branching regardless of the context in which the rule is applied. This property follows immediately for GS from inspecting the rules.

Thus, in this work, we have shown that we can design a proof system on graphs satisfying a notion of analyticity. The fact that traditional methods (such as the sequent calculus) for developing analytic proof calculi fail here is a surprise, particularly because we have targeted the minimal logic on graphs satisfying some widely accepted logical principles. The logical principles we have based our design decisions on we consider to be difficult to argue against — which does not prevent one from exploring alternative design decisions that may lead to further logics on graphs. The only assumption that may be strongly argued about is the use of the empty graph as a self-dual unit; which can be regarded as a simplifying assumption in our initial design. That design decision can be justified by *conservativity* (Theorem 8.3), in the sense that there are established formula-based logics featuring such a self-dual unit.

12. Conclusion

Guided by logical principles, we have devised a minimal proof system (GS in Figure 1) that operates directly over graphs, rather than formulas. Negation is given by graph duality, while disjunction is disjoint union of graphs, allowing us to define the implication "G implies H"

as the standard "not G or H" (see Definition 4.1). All other design decisions are then fixed by our guiding logical principles. Most of these principles follow from cut elimination (Theorem 5.4), to which the majority of this paper is dedicated. We also confirm that GS conservatively extends MLL$^\circ$ (Corollary 8.4) — a logic at the core of many proof systems.

Surprisingly, even for such a minimal generalisation of logic to graphs, deep inference is necessary. Proof systems for classical logic, intuitionistic logic, linear logic, and many other logics *may* be expressed using deep inference, but deep inference is generally not necessary, as presentations in the sequent calculus do exist. In contrast, for some logics (e.g., BV [Gug07, Tiu06] and modal logic S5 [Sto07, Pog08]), deep inference is necessary in order to define a proof system satisfying cut elimination. System GS goes further than the aforementioned systems in that all intermediate lemmas such as splitting (Lemmas 6.2 and 6.3) and context reduction (Lemma 6.4) demand a deep formulation that is more context aware than standard. As such we were required to generalise the basic mechanisms of deep inference itself in order to establish cut elimination (Theorem 5.4) for a logic over graphs. This need for deep inference for GS is due to a property of general graphs that is forbidden for graphs corresponding to formulas — that the shortest path between any two connected vertices may be greater than two; and hence, when we apply reasoning inside a module (i.e., a context), there may exist dependencies that indirectly constrain the module.

Acknowledgements. We are very grateful for insightful discussions with Anupam Das, particularly when exploring relationships between GS and linear inferences in classical logic.

References

[AHS20] Matteo Acclavio, Ross Horne, and Lutz Straßburger. Logic beyond formulas: A proof system on graphs. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 38–52. ACM, 2020. doi:10.1145/3373718.3394763. Full version with technical appendices available at `https://hal.inria.fr/hal-02560105`.

[AL01] 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.

[AM20] 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. doi:10.4230/LIPIcs.CSL.2020.6.

[And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[AT17] Andrea Aler Tubella. *A study of normalisation through subatomic logic*. PhD thesis, University of Bath, 2017.

[Bel97] Gianluigi Bellin. Subnets of proof-nets in multiplicative linear logic with mix. *Mathematical Structures in Computer Science*, 7(6):663–669, 1997. doi:10.1017/S0960129597002326.

[Ber78] Gérard Berry. Stable models of typed λ-calculi. In Giorgio Ausiello and Corrado Böhm, editors, *Automata, Languages and Programming*, pages 72–89, Berlin, Heidelberg, 1978. Springer.

[BG09] Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Trans. Comput. Logic*, 10(2), March 2009. doi:10.1145/1462179.1462186.

[BG16] Paola Bruscoli and Alessio Guglielmi. On analyticity in deep inference. Technical report, 2016. URL `http://cs.bath.ac.uk/ag/p/ADI.pdf`. note.

[BM90] Anthony J. Bonner and L. Thorne McCarty. Adding negation-as-failure to intuitionistic logic programming. In *Proceedings of the 1990 North American Conference on Logic Programming*, page 681–703, Cambridge, MA, USA, 1990. MIT Press.

[Bru02] Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming*, pages 302–316, Berlin, Heidelberg, 2002. Springer. doi:10.1007/3-540-45619-8_21.

[BS17] 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. doi:10.1007/978-3-662-55386-2_5.

[BT01] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In Robert Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 347–361, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-45653-8_24.

[Cal16] Cameron Calk. A graph theoretical extension of boolean logic. Bachelor's thesis, 2016. URL `http://www.anupamdas.com/graph-bool.pdf`.

[CDW20] Cameron Calk, Anupam Das, and Tim Waring. Beyond formulas-as-cographs: an extension of boolean logic to arbitrary graphs, 2020, 2004.12941.

[CGS11] Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. The focused calculus of structures. In Marc Bezem, editor, *CSL'11*, volume 12 of *LIPIcs*, pages 159–173, Dagstuhl, Germany, 2011. Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CSL.2011.159.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi:10.2307/2273702.

[DR89] Vincent Danos and Laurent Regnier. The structure of the multiplicatives. *Arch. Math. Log.*, 28(3):181–203, 1989. doi:10.1007/BF01622878.

[DS15] Anupam Das and Lutz Straßburger. No complete linear term rewriting system for propositional logic. In Maribel Fernández, editor, *26th International Conference on Rewriting Techniques and Applications (RTA 2015)*, volume 36 of *LIPIcs*, pages 127–142, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.RTA.2015.127.

[DS16] Anupam Das and Lutz Straßburger. On linear rewriting systems for boolean logic and some applications to proof theory. *Logical Methods in Computer Science*, 12(4):1–27, 2016. doi:10.2168/LMCS-12(4:9)2016.

[Duf65] R.J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303 – 318, 1965.

[FR94] Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2):273–285, 1994. doi:10.1017/S0960129500000451.

[Gab98] Dov M. Gabbay. *Fibring logics*. Oxford Logic Guides. Clarendon Press, 1998.

[Gal67] Tibor Gallai. Transitiv orientierbare Graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1–2):25–66, 1967.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[GGP10] 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. doi:10.4230/LIPIcs.RTA.2010.135.

[Gir87a] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987. doi:10.1016/0304-3975(87)90045-4.

[Gir87b] Jean-Yves Girard. Multiplicatives. In Gabriele Lolli, editor, *Logic and Computer Science: New Trends and Applications*, pages 11–34. Rosenberg & Sellier, 1987.

[Gir00] 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.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[GS01] 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. doi:10.1007/3-540-44802-0_5.

[GS02] 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. doi:10.1007/3-540-36078-6_16.

[GS11] Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure V: the exponentials and splitting. *Mathematical Structures in Computer Science*, 21(3):563–584, 2011. doi:https://doi.org/10.1017/S096012951100003X.

[Gug07] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007. doi:10.1145/1182613.1182614.

[HdMP04] Michel Habib, Fabien de Montgolfier, and Christophe Paul. A simple linear-time modular decomposition algorithm for graphs, using order extension. In Torben Hagerup and Jyrki Katajainen, editors, *Algorithm Theory - SWAT 2004*, pages 187–198, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[Hor19] Ross Horne. The sub-additives: A proof theory for probabilistic choice extending linear logic. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 23:1–23:16, Dagstuhl, Germany, 2019. Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSCD.2019.23.

[Hor20] Ross Horne. Session subtyping and multiparty compatibility using circular sequents. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:22, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2020.12.

[How80] William Alvin Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, London, 1980.

[HT19] Ross Horne and Alwen Tiu. Constructing weak simulations from linear implications for processes with private names. *Mathematical Structures in Computer Science*, 29(8):1275–1308, 2019. doi:10.1017/S0960129518000452.

[HTAC19] Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. De Morgan dual nominal quantifiers modelling private names in non-commutative logic. *ACM Trans. Comput. Log.*, 20(4):22:1–22:44, 2019. doi:10.1145/3325821.

[Hug06] Dominic Hughes. Proofs Without Syntax. *Annals of Mathematics*, 164(3):1065–1076, 2006. doi:10.4007/annals.2006.164.1065.

[KY93] Naoki Kobayashi and Akinori Yonezawa. ACL –a concurrent linear logic programming paradigm. In *Proceedings of the 1993 International Symposium on Logic Programming*, ILPS '93, pages 279–294, Cambridge, MA, USA, 1993. MIT Press.

[Laf95] Yves Lafont. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 225–247. Cambridge University Press, 1995.

[Lam61] Jim Lambek. On the calculus of syntactic types. In R. Jacobson, editor, *Proceedings of the Twelfth Symposium in Applied Mathematics*, volume XII, pages 166–178, 1961.

[LW00] Kamal Lodaya and Pascal Weil. Series–parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1):347 – 380, 2000. doi:https://doi.org/10.1016/S0304-3975(00)00031-1.

[Mai19] Roberto Maieli. Non decomposable connectives of linear logic. *Annals of Pure and Applied Logic*, 170(11):102709, 2019. doi:https://doi.org/10.1016/j.apal.2019.05.006.

[Mat68] Benson Mates. Leibniz on possible worlds. In B. Van Rootselaar and J.F. Staal, editors, *Logic, Methodology and Philosophy of Science III*, volume 52 of *Studies in Logic and the Foundations of Mathematics*, pages 507–529. Elsevier, 1968. doi:10.1016/S0049-237X(08)71214-X.

[Mil93] Dale Miller. The π-calculus as a theory in linear logic: Preliminary results. In E. Lamma and P. Mello, editors, *Extensions of Logic Programming*, pages 242–264, Berlin, Heidelberg, 1993. Springer. doi:10.1007/3-540-56454-3_13.

[MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied logic*, 51(1-2):125–157, 1991. doi:10.1016/0168-0072(91)90068-W.

[Möh89] Rolf H. Möhring. Computationally tractable classes of ordered sets. In Ivan Rival, editor, *Algorithms and Order*, pages 105–193, Dordrecht, 1989. Springer Netherlands. doi:10.1007/978-94-009-2639-4_4.

[MP13] Dale Miller and Elaine Pimentel. A formal framework for specifying sequent calculus proof systems. *Theoretical Computer Science*, 474:98–116, 2013.

[MS94] 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.

[MS14] Sonia Marin and Lutz Straßburger. Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In *Advances in Modal Logic 10*, Groningen, Netherlands, August 2014.

[NPW81] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13(1):85 – 108, 1981. doi:https://doi.org/10.1016/0304-3975(81)90112-2.

[Pet77] Carl Adam Petri. Interpretations of net theory. *ACM Computing Surveys*, 9(3):223–252, 1977.

[Pog08] Francesca Poggiolesi. A cut-free simple sequent calculus for modal logic S5. *The Review of Symbolic Logic*, 1(1):3–15, 2008. doi:10.1017/S1755020308080040.

[Pra86] Vaughan Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986. doi:10.1007/BF01379149.

[Ret97] Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Philippe de Groote and J. Roger Hindley, editors, *Typed Lambda Calculi and Applications*, pages 300–318, Berlin, Heidelberg, 1997. Springer. doi:10.1007/3-540-62688-3_43.

[Ret03] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. *Theoretical Computer Science*, 294(3):473–488, 2003. doi:10.1016/S0304-3975(01)00175-X.

[Sto07] Phiniki Stouppa. A deep inference system for the modal logic S5. *Studia Logica*, 85(2):199–214, 2007. doi:10.1007/s11225-007-9028-y.

[Str02] Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 388–402, Berlin, Heidelberg, 2002. Springer. doi:10.1007/3-540-36078-6_26.

[Str03] Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003.

[Str17] Lutz Straßburger. Combinatorial Flows and Their Normalisation. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, volume 84 of *LIPIcs*, pages 31:1–31:17, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSCD.2017.31.

[Tiu06] 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. doi:10.2168/LMCS-2(2:4)2006.

[War19] Timothy Waring. A graph theoretic extension of boolean logic. Master's thesis, 2019. URL `http://anupamdas.com/thesis_tim-waring.pdf`.