A Graphical Proof Theory of Logical Time

Matteo Acclavio
Department of Computer Science, University of Luxembourg, Luxembourg

Ross Horne
Department of Computer Science, University of Luxembourg, Luxembourg

Sjouke Mauw
Department of Computer Science, University of Luxembourg, Luxembourg

Lutz Straßburger
Inria-Saclay, Palaiseau, France

Abstract

Logical time is a partial order over events in distributed systems, constraining which events precede others. Special interest has been given to series-parallel orders since they correspond to formulas constructed via the two operations for "series" and "parallel" composition. For this reason, series-parallel orders have received attention from proof theory, leading to pomset logic, the logic BV, and their extensions. However, logical time does not always form a series-parallel order; indeed, ubiquitous structures in distributed systems are beyond current proof theoretic methods. In this paper, we explore how this restriction can be lifted. We design new logics that work directly on graphs instead of formulas, we develop their proof theory, and we show that our logics are conservative extensions of the logic BV.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Linear logic; Theory of computation → Process calculi

Keywords and phrases proof theory, causality, deep inference

Digital Object Identifier 10.4230/LIPIcs.FSCD.2022.22

1 Introduction

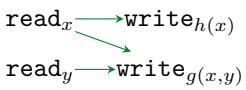
This paper is about the design of graphical proof systems that have an expressive power beyond logics constrained by the syntax of formulas. We explain, in this introduction, our original observations that compelled us to explore this untapped region in logic, whilst exploring a proof theory of logical time. Logical time is a partial order constraining events. It is employed because some events have to precede others (for instance due to read/write dependencies) and because synchronising clocks is infeasible in distributed systems.

A special case is given by series-parallel orders [39, 18, 6] which gave rise to a family of non-commutative logics, including pomset logic [36] and BV [19]. A series-parallel order is a partial order that can be constructed by composing smaller components, in series ($\lhd$) and in parallel ($\parr$). For example, the following two series-parallel orders $a \longrightarrow b \longrightarrow c \longrightarrow d$ and  correspond to the formulas $(a \lhd b) \lhd (c \lhd d)$ and $(a \parr b) \lhd (c \parr d)$, respectively.

Pomset logic and BV give series-parallel orders a first-class status inside a logical system. Both logics extend the core of linear logic, hence feature a multiplicative conjunction ($\otimes$) that is de Morgan dual to $\parr$. Implication, $A \multimap B$ is internalised as $A^\perp \parr B$, where $A^\perp$ is linear negation, with respect to which sequential composition is self dual, that is, $(A \lhd B)^\perp = A^\perp \lhd B^\perp$. For example, in both BV and pomset logic we have $(a \lhd b) \lhd (c \lhd d) \multimap (a \parr b) \lhd (c \parr d)$, and hence the above-mentioned series-parallel orders are related by implication.

Modelling logical time, where the causal relation "happens before" constrains events in a distributed system in such a way that they certainly precede others [29], has always been a core motivation for BV [8]. Implication can be a sound tool for reasoning about behavioural preorders between processes such as simulation [11, 26, 24, 25], and it can be used to devise powerful notions of multiparty compatibility and subtyping in session types [23].

However, all the above-mentioned work is limited by the series-parallel restriction, that is, on orders which can be decomposed using series and parallel compositions. Yet, not all patterns of causality can be represented by series-parallel orders. Consider for example read-write dependencies, where shapes such as



(1)

can occur. In this diagram there are two data write operations where one is dependent on two previous read operations, while the other is dependent on a single read. The diagram in (1) above is not a series-parallel order and indeed is out-of-scope of BV and pomset logic. In contrast, the diagram is in scope of general process models, such as Pratt's original pomsets and event structures [33, 34].

Clearly, such non-series-parallel structures arise ubiquitously. As a running example, let us consider consumer-producer queues which are used to preserve message order when composing asynchronous distributed components, as depicted in Figure 1. In these diagrams, the labels on the nodes represent the events of enqueueing or dequeueing four particular messages (a, b, c, and d). Then, an enqueue event must happen before its corresponding dequeue event. Secondly, if one enqueue event happens before another enqueue event, then the corresponding dequeue events occur in the same order. This preserves a first-in-first-out order for messages exchanged using the queue. Finally, queues with capacity n have the restriction that the message i must be dequeued before message $i + n$ is enqueued.

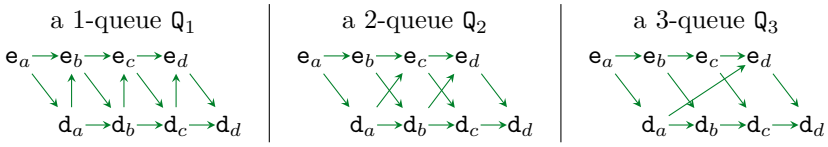


Figure 1 Causality patterns for consumer-producer n-queues, where n is the bound on the number of messages that can be enqueued. Nodes labelled by $\mathtt{e}_x$ and $\mathtt{d}_x$ respectively represent the enqueuing and dequeuing of the message x. In all three queue types we only represent the first four messages (a, b, c, and d) inserted into the queue.

The edges in the diagrams in Figure 1 should not be seen as the possible paths when "executing" the system, but they describe the dependencies that have to be met before executing each event. If there are no edges between a given pair of events and if all dependencies of these two events are met, then they can be executed in any order. In other words, they describe the Hasse diagram of the induced partial order. In this way, one can see that dependencies are relaxed from left to right in Figure 1, permitting more concurrency.

The first example is a 1-queue, allowing at most one message to be enqueued at any time. The result is a simple total order over events, alternating between enqueue and dequeue events, represented by the formula $\mathtt{e}_a \lhd \mathtt{d}_a \lhd \mathtt{e}_b \lhd \mathtt{d}_b \lhd \mathtt{e}_c \lhd \mathtt{d}_c \lhd \mathtt{e}_d \lhd \mathtt{d}_d$. This helps explain why some verification frameworks for distributed systems, e.g. [15, 12], assume 1-queues as a

simplified approximation of queues. The 2-queue, allowing two messages to be buffered, is also a series-parallel order, represented by $\mathtt{e}_a \triangleleft (\mathtt{e}_b \parr \mathtt{d}_a) \triangleleft (\mathtt{e}_c \parr \mathtt{d}_b) \triangleleft (\mathtt{e}_d \parr \mathtt{d}_c) \triangleleft \mathtt{d}_d$. Hence we can use BV or pomset logic to reason about these queues, for example proving that the 1-queue implies the 2-queue, serving as a proof that a 2-queue can simulate behaviours of a 1-queue.[1] In the third example, up to three messages can be buffered in the queue. However, this 3-queue is not a series-parallel order,[2] nor is any other n-queue with $n > 2$. Therefore the existing proof-theoretical tools cannot be used to reason about them.

This is the main motivation for this paper. Can we design proof systems that work directly on graphs that are not bound by the series-parallel restriction, i.e., go beyond formulas?

This problem turned out to be more challenging than expected. A first step was communicated in [1], which devised a minimal logic, called GS, over general undirected graphs, where the symmetric edges generalise the multiplicative connectives of linear logic, without the presence of directed edges representing logical time. The proof theory of GS requires deep inference, where inference rules may be applied deep in any context, not just at the root connective of a formula as for example in the sequent calculus. For developing a structural proof theory in the absence of formulas, we employed the notion of *modular decomposition* from graph theory, which enables any graph to be represented as a tree of *prime graphs*.

The main contribution of this paper is a graphical logic that generalizes both, GS and BV. Following the recent discovery that BV and pomset logic are not the same [32], we also made the observation that there is not one canonical choice about what is the "right" logic. We present here two proof systems GV and $\mathsf{GV}^{\mathsf{sl}}$, that both (i) are analytic (i.e., suitable for proof search) and (ii) obey cut elimination. In this respect, we can indeed speak of a proper proof theory. We present our logics using *open deduction*, which is a deep inference formalism. Furthermore, we show (iii) that both logics, GV and $\mathsf{GV}^{\mathsf{sl}}$, are conservative extensions of GS and of BV, and (iv) that the provability problem in GV and in $\mathsf{GV}^{\mathsf{sl}}$ is **NP**-complete.

Structure of the paper. We begin by recalling some preliminary notions about graphs and their modular decomposition in Section 2 and some preliminaries about deep inference and open deduction in Section 3. In Section 4 we introduce the inference rules for our proof systems and show some of their properties. Then, in Section 5 and Section 6 we prove cut elimination and some of its consequences, including the results on conservativity and complexity mentioned above.

2 Preliminaries on Graphs and their Modular Decomposition

A ***directed graph*** $G = \langle V_G, \overset{G}{\curvearrowright} \rangle$ is a set V_G of *vertices* equipped with an irreflexive binary *edge relation* $\overset{G}{\curvearrowright} \subseteq V_G \times V_G$. An ***undirected graph*** $G = \langle V_G, \overset{G}{\frown} \rangle$ is a set V_G of *vertices* equipped with an irreflexive and symmetric binary *edge relation* $\overset{G}{\frown} \subseteq V_G \times V_G$. A ***mixed graph*** (or simply ***graph***) is a triple $G = \langle V_G, \overset{G}{\frown}, \overset{G}{\curvearrowright} \rangle$ where $\langle V_G, \overset{G}{\frown} \rangle$ is an undirected graph and $\langle V_G, \overset{G}{\curvearrowright} \rangle$ is a directed graph, such that $\overset{G}{\frown} \cap \overset{G}{\curvearrowright} = \varnothing$. In a mixed graph we define the following additional edge-relations:

[1] The proof for the implication $\mathtt{Q}_1 \multimap \mathtt{Q}_2$ is shown in Figure 3.
[2] It suffices to remark that the vertices in $\{\mathtt{e}_b, \mathtt{e}_c, \mathtt{e}_d, \mathtt{d}_a\}$ induce an N-shaped subgraph similar to the one from Equation (1).

$$\begin{array}{lcl} \overset{G}{\not\frown} & = & \{(v,w) \mid v \neq w \text{ and } (v,w) \notin \overset{G}{\frown}\} \\ \overset{G}{\frown} & = & \{(w,v) \mid (v,w) \in \overset{G}{\frown}\} \end{array} \qquad \begin{array}{lcl} \overset{G}{\not\smile} & = & \{(v,w) \mid v \neq w \text{ and } (v,w) \notin \overset{G}{\smile}\} \\ \overset{G}{\smile} & = & \{(v,w) \mid v \overset{G}{\not\frown} w \text{ and } v \overset{G}{\not\frown} w \text{ and } w \overset{G}{\not\frown} v\} \end{array} \tag{2}$$

We say that a graph G is ***n*-colour** (for $n \leq 3$) if at most n of the three edge relations $\smile$ (white edges), $\frown$ (green edges) and $\frown$ (red edges) are non-empty. In particular, we say that G is a ***complete graph*** if $\overset{G}{\smile} = \varnothing$, a directed graph if $\overset{G}{\frown} = \varnothing$, and an undirected graph if $\overset{G}{\frown} = \varnothing$. The empty graph is denoted as $\varnothing$.

In the following we may omit the index/superscript G when it is clear from the context. When drawing a graph we draw v—w whenever $v \frown w$, we draw $v \rightarrow w$ whenever $v \frown w$, and we draw no edge at all whenever $v \smile w$.

▶ **Definition 1.** *Let G and H be graphs. If $V' \subseteq V_G$, then we define the* ***subgraph induced by V'*** *as the graph $G|_{V'} = \langle V', \overset{G}{\frown} \cap (V' \times V'), \overset{G}{\frown} \cap (V' \times V') \rangle$. We write $H \sqsubseteq G$ if there is an injective homomorphism f from H to G such that $f(H)$ is an induced subgraph of G, and we say that G is* ***H*-free** *whenever $H \sqsubseteq G$ does not hold.*

As shown previously [1, 9], graphs can be used as operators to compose graphs, playing a similar role to connectives in formulas.

▶ **Definition 2.** *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, with disjoint vertices. We define the* ***composition of $H_1, \ldots, H_n$ via G*** *as the graph $G(\!|H_1, \ldots, H_n|\!)$ obtained by replacing each vertex v_i of G by the graph H_i. That is, the graph with vertices the union of the vertices of $H_1, \ldots, H_n$ and with an edge $x \frown y$ (resp. $x \frown y$) if either x and y are in the same H_i and $x \overset{H_i}{\frown} y$ (resp. $x \overset{H_i}{\frown} y$), or $x \in V_{H_i}$ and $y \in V_{H_j}$ for $i \neq j$ and $v_i \overset{G}{\frown} v_j$ (resp. $v_i \overset{G}{\frown} v_j$). Formally,*

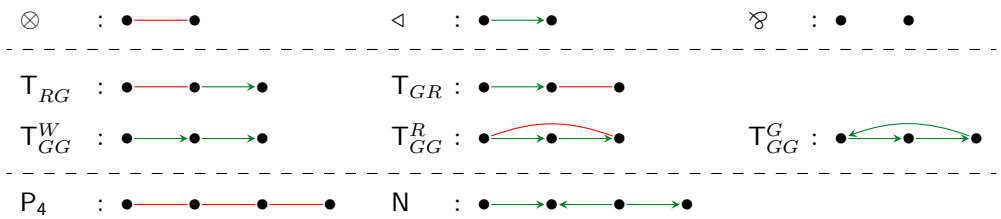
$$\begin{array}{lcl} V_{G(\!|H_1,\ldots,H_n|\!)} & = & \bigcup_{1 \leq i \leq n} V_{H_i} \\ \overset{G(\!|H_1,\ldots,H_n|\!)}{\frown} & = & \left(\bigcup_{1 \leq i \leq n} \overset{H_i}{\frown}\right) \cup \left\{ (x,y) \mid x \in V_{H_i} \text{ and } y \in V_{H_j} \text{ and } v_i \overset{G}{\frown} v_j \text{ and } i,j \in \{1,\ldots,n\} \right\} \\ \overset{G(\!|H_1,\ldots,H_n|\!)}{\frown} & = & \left(\bigcup_{1 \leq i \leq n} \overset{H_i}{\frown}\right) \cup \left\{ (x,y) \mid x \in V_{H_i} \text{ and } y \in V_{H_j} \text{ and } v_i \overset{G}{\frown} v_j \text{ and } i,j \in \{1,\ldots,n\} \right\} \end{array}$$

In developing our proof systems, we use the notion of *modular decomposition* [16, 27, 22, 30, 31, 13, 7] in order to assign to each graph a tree-like structure where leaves are single vertex graphs, and modules are sub-trees.[3]

▶ **Definition 3.** *A* ***module*** *of a graph G is a subset M of V_G such that $x R z$ iff $y R z$ for any $x, y \in M$, $z \in V_G \setminus M$ and $R \in \{\frown, \frown, \frown\}$. We say that a module M is* ***trivial*** *if $M = \varnothing$, $M = V_G$, or $M = \{x\}$ for some $x \in V_G$. A graph G is* ***prime*** *if $|V_G| > 1$ and all its modules are trivial.*

▶ **Notation 4.** *In this paper we identify a module M of a graph G and its induced subgraph $G|_M$. We introduce the following notation for relevant prime graphs on 2, 3 and 4 vertices, respectively:*

[3] The notion of module we use here coincides with the one of *clans* from the literature on 2-structures [13].



(3)

We use the following notation for the composition via (prime) graphs with two vertices:

$$G \parr H = \parr(\!|G, H|\!) \qquad G \otimes H = \otimes(\!|G, H|\!) \qquad G \triangleleft H = \triangleleft(\!|G, H|\!) \qquad .$$

Prime graphs play a special role in modular decomposition and can be considered as generalized (non-decomposable) logic connectives in the sense of [17, 10, 3].

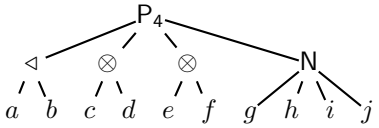
▶ **Theorem 5** ([16, 27, 13])**.** *Let G be a graph with at least two vertices. Then there are some non-empty graphs $H_1, \ldots, H_n$ and a prime graph P such that $G = P(\!|H_1, \ldots, H_n|\!)$.*

▶ **Corollary 6.** *Each graph G admits a **modular decomposition** by means of prime graphs and single-vertex graphs. Such a decomposition is unique modulo associativity of $\otimes$, $\parr$ and $\triangleleft$, and prime graphs isomorphisms.*

▶ **Example 7.** Consider the graph on the left below.

(4)

The representation on the right above relies on modular decomposition to reduce the number of edges to draw: the vertices inside a rectangle belong to a same module. Therefore an edge touching a rectangle represents a bundle of edges (of the same type) connecting with each vertex in the module. We can associate a ***modular decomposition tree*** to each graph in the same way we associate an abstract syntax tree to a formula. Below is a tree representation of the modular decomposition of the graph above and its formula-like denotation, which uses the prime graphs from (3) and the composition notation from Definition 2.



$$\mathsf{P}_4(\!|(a \triangleleft b), (c \otimes d), (e \otimes f), \mathsf{N}(\!|g, h, i, j|\!)|\!) \qquad (5)$$

Indeed, prime graphs can be thought as primitive operators or connectives for graphical logic (see Section 9 of [2]), and the notion of module can be seen as a generalization of subformulas.

In this paper, we use the graphical representation on the right of (4) and the formula-like representation on the right of (5) interchangeably.

▶ **Definition 8.** *A prime graph P is a **connector** of a graph G (or **P-connector** if we want to specify the prime graph P) if there is an occurrence of the graph P in the modular decomposition of G.*

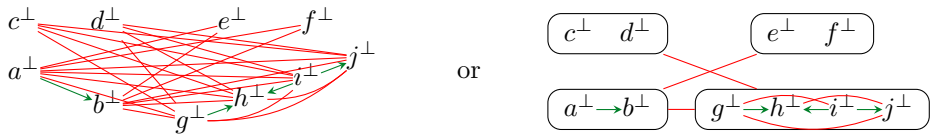
For example, in the graph in Example 7 above, we have two connectors $\otimes$, one $\triangleleft$, one N and one P_4.

We assume graphs are labelled. That is, each vertex v of a graph G carries a label $\ell_G(v)$ selected from a label set $\mathcal{L}$. In particular in this paper we assume $\mathcal{L} = \mathcal{A} \cup \mathcal{A}^\perp$ where $\mathcal{A}$ is a set of atoms $\{a, b, c, \ldots\}$ and $\mathcal{A}^\perp$ is the set of negated atoms $\{a^\perp, b^\perp, c^\perp, \ldots\}$.

▶ **Definition 9.** *Let $G = \langle V_G, \overset{G}{\frown}, \overset{G}{\curvearrowright} \rangle$ be a graph, we define its **dual** graph $G^\perp = \langle V_{G^\perp}, \overset{G^\perp}{\frown}, \overset{G^\perp}{\curvearrowright} \rangle$ with $V_{G^\perp} = V_G$ and $\overset{G^\perp}{\frown} = \overset{G}{\smile}$ and $\overset{G^\perp}{\curvearrowright} = \overset{G}{\curvearrowright}$. The label of each vertex in $V_{G^\perp}$, is the dual of the label of the corresponding vertex in G, that is, if $\ell_G(v) = x$, then $\ell_{G^\perp}(v) = x^\perp$. For this purpose we assume negation to be involutive, that is, $x^{\perp\perp} = x$. The **implication** $G \multimap H$ is defined as $G^\perp \parr H$.*

The dual graph operation above flips labels on vertices, exchanges $\frown$ (red) and $\smile$ (white) symmetric edges, but preserves $\curvearrowright$ (green) directed edges.

▶ **Example 10.** The graph $\mathsf{P}_4{}^\perp(\!|(a^\perp \triangleleft b^\perp), (c^\perp \parr d^\perp), (e^\perp \parr f^\perp)|\!), \mathsf{N}^\perp(\!|g^\perp, h^\perp, i^\perp, j^\perp|\!))$ is the dual of the graph from Example 7 and is represented as follows:



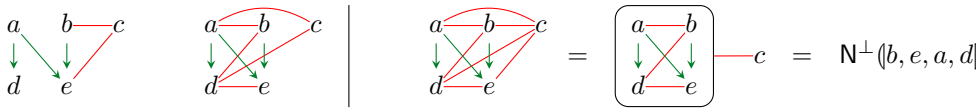
In this work we are mainly interested in how vertices are labelled, and not in the identity of the underlying vertex. For this reason we rely on the notion of graph isomorphism.

▶ **Definition 11.** *Let G and G' be two graphs. We say that G and G' are **isomorphic** if there is a bijection $f : V_G \to V_{G'}$ such that $x \frown y$ iff $f(x) \frown f(y)$ and $x \curvearrowright y$ iff $f(x) \curvearrowright f(y)$ for any $x, y \in V_G$. If the graphs are labelled, then we additionally require that $\ell(v) = \ell(f(v))$. If G and G' are isomorphic we write $G \sim_f G'$, or simply $G \sim G'$ if the isomorphism f is clear form the context.*

▶ **Definition 12.** *A **web** is a graph such that $\curvearrowright$ is transitive and such that each connector occurring in its modular decomposition is a 2-colour prime graph. A **relation web** is a web such that each connector is in $\{\parr, \triangleleft, \otimes\}$. A **cograph** is a relation web with $\curvearrowright = \varnothing$. A **series-parallel order** is a relation web with $\frown = \varnothing$.*

It is worth noticing that the above graphs may be characterised by ruling out the presence of specific induced subgraphs (see, e.g., [37, 19, 14]). In particular, cographs are P_4-free undirected graphs, series-parallel orders are N-free directed graphs with $\curvearrowright$ being transitive, and relation webs are T_{GR}-free, P_4-free, and N-free graphs with transitive $\curvearrowright$. Note that requiring $\curvearrowright$ to be transitive is equivalent to requiring that a graph is T^W_{GG}-free, T^G_{GG}-free and T^R_{GG}-free. Moreover, graphs with no 3-colour connectors are by definition T_{GR}-free and T_{RG}-free, however the converse does not hold.

▶ **Example 13.** The two graphs below on the left are not webs since they are three-colour prime graphs, while the complete graph on the right is a web since its modular decomposition only contains 2-colour connectors.



▶ **Notation 14.** *In drawing webs we may omit directed edges whenever they are derivable from the drawn ones via the transitivity of $\curvearrowright$. This means, we can represent the web $u \triangleleft v \triangleleft w$ as $u \to v \to w$, in which we implicitly assume the directed edge from u to w. For example, the web Q_1 in Figure 1 can be represented as a single thread $\mathsf{e}_a \to \mathsf{d}_a \to \mathsf{e}_b \to \mathsf{d}_b \to \mathsf{e}_c \to \mathsf{d}_c \to \mathsf{e}_d \to \mathsf{d}_d$.*

▶ Remark 15. The operation of composing graphs via a graph (Definition 2) is closed with respect to the sets of cographs, series-parallel orders, relation webs and webs. That is, if all $H_1, \ldots, H_n$ and G are cographs (resp. series-parallel orders, relation webs, webs), then so is $G\langle\!\langle H_1, \ldots, H_n \rangle\!\rangle$.

The correspondence between cographs and formulas containing only connectives for conjunction and disjunction is well-known since the 60s [16, 27]. We here recall the correspondence between relation webs and the formulas for the non-commutative logic BV from [19, 35].

▶ **Definition 16.** *The set of* BV*-formulas is defined by a countable set of propositional* ***atoms*** $\mathcal{A} = \{a, b, \ldots\}$*, a special symbol* $\circ$ *called* ***unit****, and the following grammar.*

$$\phi, \psi := a \mid a^\perp \mid \circ \mid \phi \parr \psi \mid \phi \triangleleft \psi \mid \phi \otimes \psi$$

The binary connectives $\parr$*,* $\triangleleft$ *and* $\otimes$ *are called* ***par****,* ***seq*** *and* ***tensor****. Par and tensor correspond to disjunction and conjunction. We associate to each formula* ϕ *a relation web* $[\![\phi]\!]$ *defined as follows:*

$$\begin{array}{ccc} [\![\circ]\!] = \varnothing & [\![a]\!] = a & [\![a^\perp]\!] = a^\perp \\ [\![\phi \triangleleft \psi]\!] = [\![\phi]\!] \triangleleft [\![\psi]\!] & [\![\phi \parr \psi]\!] = [\![\phi]\!] \parr [\![\psi]\!] & [\![\phi \otimes \psi]\!] = [\![\phi]\!] \otimes [\![\psi]\!] \end{array} \tag{6}$$

We identify an atom a *(resp. its negation* $a^\perp$*) with a single vertex graph labelled by the same atom* a *(resp.* $a^\perp$*).*

▶ **Theorem 17** ([35, 19])**.** *A graph* G *is a relation web iff there is a* BV*-formula* ϕ *such that* $G = [\![\phi]\!]$*.*

We will make use of this result later in Section 4, when formulating the proof system for BV as a proof system on relation webs rather than a proof system on formulas.

3 Preliminaries on Open Deduction

Open deduction [20] is a proof formalism based on deep inference [5]. It has originally been defined for formulas, but it is abstract enough such that it can equally well be used for graphs, as already done in [2]. We begin here by defining the notion of *context*, which is a graph with a hole into which another graph can be plugged, generalising the similar notion of context for formulas.

▶ **Definition 18** (Context)**.** *A* ***context*** $\mathcal{C}[\Box]$ *is a graph containing a single occurrence of a special vertex* $\Box$*. If* $\mathcal{C}[\Box]$ *is a context and* G *a graph, we define* $\mathcal{C}[G]$ *as the graph obtained by replacing* $\Box$ *by* G*, that is, the graph obtained by replacing* $\Box$ *by (the modular decomposition of)* G *in the modular decomposition of* $\mathcal{C}[\Box]$*.*

Note that the notion of context is strongly connected to the one of module: M is a module of a graph G iff $G = \mathcal{C}[M]$ for some context $\mathcal{C}[\Box]$.

▶ **Definition 19.** *An* ***inference system*** S *is a set of inference rules (as for example shown in Figure 2). A* ***derivation*** $\mathcal{D}$ *in* S *with premise* G *and conclusion* H *is denoted* $\begin{array}{c} G \\ \mathcal{D} \Vert \mathsf{S} \\ H \end{array}$ *and is defined inductively as follows:*

- *Every graph* G *is a (****trivial****) derivation (also denoted* G*) with premise* G *and conclusion* G*.*

- *Every instance of a rule* $\mathsf{r}\dfrac{G}{H}$ *in* S *is a derivation with premise* G *and conclusion* H*.*
- *If* $\mathcal{D}_1$ *is a derivation with premise* G_1 *and conclusion* H_1*, and* $\mathcal{D}_2$ *is a derivation with premise* G_2 *and conclusion* H_2*, and* $H_1 \sim_f G_2$*, then the composition of* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *is a derivation* $\mathcal{D}_2 \circ_f \mathcal{D}_1$ *denoted as below on the left.*

$$\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \\ f \ \cdots\cdots \\ G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array} \quad \Bigg| \quad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \\ \cdots\cdots \\ G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array} \quad \textit{or} \quad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array} \quad \textit{or} \quad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ G_2 \\ \mathcal{D}_2 \Vert \mathsf{S} \\ H_2 \end{array} \tag{7}$$

  If f *is clear from context we may simply write* $\mathcal{D}_2 \circ \mathcal{D}_1$ *and denote it as above on the right in order to ease readability. However, even if the isomorphism* f *is not written, we always assume it is part of the derivation and explicitly given.*
- *If* G *is a graph with* n *vertices and* $\mathcal{D}_1, \ldots, \mathcal{D}_n$ *are derivations with premise* G_i *and conclusion* H_i *for each* $i \in \{1, \ldots, n\}$*, then* $G(\!|\mathcal{D}_1, \ldots, \mathcal{D}_n|\!)$ *is a derivation with premise* $G(\!|G_1, \ldots, G_n|\!)$ *and conclusion* $G(\!|H_1, \ldots, H_n|\!)$ *denoted as below on the left (if* $G = \otimes$ *or* $G = \triangleleft$ *or* $G = \parr$ *we may write the derivations as below on the right).*

$$G\left(\begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \mathsf{S} \\ H_1 \end{array}, \ldots, \begin{array}{c} G_n \\ \mathcal{D}_n \Vert \mathsf{S} \\ H_n \end{array}\right) \quad \Bigg| \quad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array} \otimes \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array} \qquad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array} \triangleleft \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array} \qquad \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array} \parr \begin{array}{c} G_1 \\ \mathcal{D}_1 \Vert \\ H_1 \end{array}$$

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* $\vdash_{\mathsf{S}} G$*.*

Thus, if $\begin{array}{c} G \\ \mathcal{D} \Vert \mathsf{S} \\ H \end{array}$ is a derivation and $\mathcal{C}[\Box]$ a context, then we have a derivation $\mathcal{C}\left[\begin{array}{c} G \\ \mathcal{D} \Vert \mathsf{S} \\ H \end{array}\right] = \begin{array}{c} \mathcal{C}[G] \\ \mathcal{C}[\mathcal{D}] \Vert \mathsf{S} \\ \mathcal{C}[H] \end{array}$.

4 Proof Systems on Webs

In this section, we recall the proof systems BV from [19] (that we here formulate as a proof system on relation webs) and GS from [2] (which is a proof system on undirected graphs), and we introduce two new proof system on webs using the rules from Figure 2.[4]

$$\begin{array}{lcllcl} \mathsf{BV} & = & \{\mathsf{ai}{\downarrow}, \mathsf{s}, \mathsf{q}_{\mathsf{BV}}{\downarrow}\} & \mathsf{GV} & = & \{\mathsf{ai}{\downarrow}, \mathsf{s}_{\parr}, \mathsf{s}_{\otimes}, \mathsf{p}{\downarrow}, \mathsf{q}{\downarrow}, \mathsf{qm}\} \\ \mathsf{GS} & = & \{\mathsf{ai}{\downarrow}, \mathsf{s}_{\parr}, \mathsf{p}{\downarrow}\} & \mathsf{GV}^{\mathsf{sl}} & = & \{\mathsf{ai}{\downarrow}, \mathsf{s}_{\parr}, \mathsf{s}_{\otimes}, \mathsf{p}{\downarrow}, \mathsf{q}{\downarrow}, \mathsf{qm}, \mathsf{sl}\} \end{array} \tag{8}$$

[4] Note that in [2, 1] the system GS is defined with the rule $\mathsf{s.sw}_{\parr}\dfrac{\mathcal{C}[M]}{\mathcal{C}[\varnothing] \parr M}\, M \neq \varnothing$ instead of $\mathsf{s}_{\parr}$. However it is easy to show that $\mathsf{s.sw}_{\parr}$ is derivable using $\mathsf{s}_{\parr}$.

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

$$\mathsf{q_{BV}}{\downarrow}\,\frac{(N_1 \parr N_3) \triangleleft (N_2 \parr N_4)}{(N_1 \triangleleft N_2) \parr (N_3 \triangleleft N_4)} \qquad \mathsf{s}\,\frac{(M_1 \parr N) \otimes M_2}{M_1 \parr (N \otimes M_2)} \qquad \mathsf{q_{BV}}{\uparrow}\,\frac{(N_1 \otimes N_2) \otimes (N_3 \otimes N_4)}{(N_1 \otimes N_3) \triangleleft (N_2 \otimes N_4)}$$

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

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

$$\mathsf{q}{\downarrow}\,\frac{Q^{\perp}(\!|L_1 \parr J_1, \ldots, L_n \parr J_n|\!)}{Q^{\perp}(\!|L_1, \ldots, L_n|\!) \parr Q(\!|J_1, \ldots, J_n|\!)} \qquad \mathsf{q}{\uparrow}\,\frac{Q(\!|L_1, \ldots, L_n|\!) \otimes Q^{\perp}(\!|J_1, \ldots, J_n|\!)}{Q(\!|L_1 \otimes J_1, \ldots, L_n \otimes J_n|\!)}$$

$$\mathsf{qm}\,\frac{Q(\!|L_1 \parr J_1, \ldots, L_n \parr J_n|\!)}{Q(\!|L_1, \ldots, L_n|\!) \parr Q(\!|J_1, \ldots, J_n|\!)}$$

$$\mathsf{sl}\,\frac{Q(\!|M_1, \ldots, M_k, \varnothing, \ldots, \varnothing|\!) \triangleleft Q(\!|\varnothing, \ldots, \varnothing, M_{k+1}, \ldots, M_n|\!)}{Q(\!|M_1, \ldots, M_n|\!)}$$

For all rules we assume P, Q and R prime webs with $\overset{R}{\frown} = \varnothing$, $\overset{Q}{\frown} = \varnothing$, and $M_i \neq \varnothing \neq L_i \parr J_i$ for all i.
For the rule sl *we also require $y \not\curvearrowright x$ for all $x \in M_i, y \in M_j$ with $0 \leq i \leq k < j \leq n$.*

Figure 2 Inference rules for our family of proof systems on webs.

Following the deep inference tradition, the rules come in pairs, a down-rule (marked with ↓) and an up-rule (marked with ↑), which are dual to each other. The proof systems in (8) above only contain the down-rules, as the up-rules are admissible, which we will show in the next section.[5]

The rules ai are called ***atomic interactions*** and allow to derive the atomic implication $a \multimap a$. The rules s (called ***switch***), $\mathsf{q_{BV}}{\downarrow}$ and $\mathsf{q_{BV}}{\uparrow}$ are the rules obtained by adapting the standard proof system BV on formulas to relation webs. When generalising s from relation webs to webs, this rule admits two formulations $\mathsf{s}_{\parr}$ and $\mathsf{s}_{\otimes}$, which are respectively called ***switch par*** and ***switch tensor***. Intuitively, $\mathsf{s}_{\parr}$ allows to move a connected component M_i inside a context of the shape $\mathcal{C}[\Box] = P(\!|M_1, \ldots, M_{i-1}, N \parr \Box, M_{i+1}, \ldots M_n|\!)$. Dually, $\mathsf{s}_{\otimes}$ allows to extract a module M_i from a context $\mathcal{C}[\Box] = P(\!|M_1, \ldots, M_{i-1}, N \otimes \Box, M_{i+1}, \ldots M_n|\!)$.[6] The rules p are called ***prime graph*** rules.[7] Each instance of $\mathsf{p}{\downarrow}$ creates disjunctions of the modules in a prime connector R and the corresponding modules in an occurrence of its dual

[5] The rules sl and qm are in fact down-rules, but we omitted the ↓ as we do not need the corresponding up-rules in this paper.

[6] From inspecting the rules, one can observe that $\mathsf{s}_{\parr}$ and $\mathsf{s}_{\otimes}$ are dual to each other. And indeed in simpler systems like BV and GS, the $\mathsf{s}_{\otimes}$ is admissible like all other "up-rules". However, for GV and $\mathsf{GV}^{\mathsf{sl}}$, this is no longer the case (see Example 44 in the appendix).

[7] Differently from [1, 2] in this paper we drop the side condition $|V_P| \geq 4$ in p-rules. This choice restrains the set of rules required to simulate the general i-rules (see Lemma 24). Note that this makes the s-rule also a special case of $\mathsf{p}{\downarrow}$.

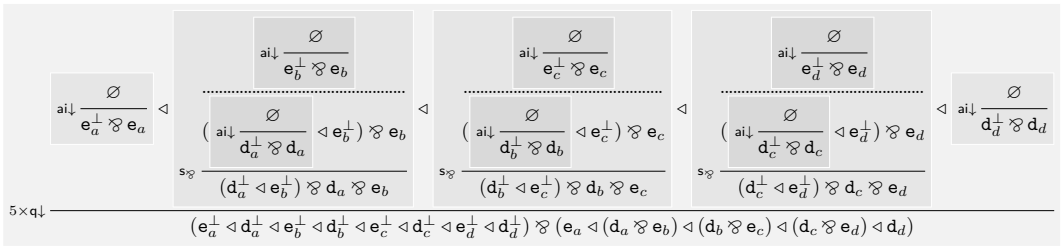
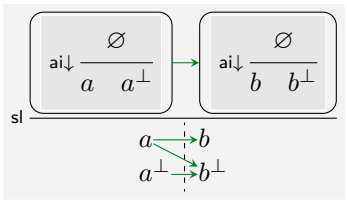


Figure 3 Proof of $\mathtt{Q}_1 \multimap \mathtt{Q}_2$ from the introduction. This is a correct proof in BV, in GV, and in GV$^{\mathsf{sl}}$.

$R^\perp$ and puts them in conjunction. The rules q and qm are called **q-*rules*** and generalise the two rules $\mathsf{q}_{\mathsf{BV}}{\downarrow}$ and $\mathsf{q}_{\mathsf{BV}}{\uparrow}$ from BV to general prime webs with an empty set of $\frown$-edges. The rule sl is called ***slice*** and is yet another generalisation the $\mathsf{q}_{\mathsf{BV}}{\downarrow}$-rule. It formalises the idea that during proof search we aim at slicing a directed graph into a "before" and an "after" part by introducing additional $\curvearrowright$-edges. This mimics a longstanding idea in distributed systems: a partial order is a representation of many possible linear orders in which concurrent events may occur [29]. For example, below is a proof of a small web provable in GV$^{\mathsf{sl}}$ but not provable in GV.



Other examples of derivations are given in Figure 3, Figure 4, Figure 5, and Figure 6.

▶ Remark 20. The rules $\mathsf{p}{\downarrow}$, $\mathsf{p}{\uparrow}$, $\mathsf{q}{\downarrow}$ and $\mathsf{q}{\uparrow}$ are crucial to obtain a proof system containing only atomic interactions (see Proposition 28). These rules gather two dual connectors by merging their set of edges, therefore obtaining a complete graph. In particular, the formulation of the rule $\mathsf{q}{\downarrow}$ exploits the fact that if Q is a prime web such that $\overset{Q}{\frown} \neq \varnothing$, then either Q or $Q^\perp$ is a complete graph. This will be crucial for proving the results in this paper. However, as Example 13 shows, this fact is not true for general 3-colour prime graphs. The question whether the results on webs, that are presented in this paper, can be generalized to arbitrary graphs is an open problem.

Let us now show that GV and GV$^{\mathsf{sl}}$ do indeed extend BV beyond the restriction of relation webs (i.e., formulas).

▶ **Lemma 21.** *Let A and B be relation webs.*

$$\textit{if } \mathsf{s}\frac{B}{A} \textit{ , then } \mathsf{s}_{\parr}\frac{B}{A} \qquad \textit{if } \mathsf{q}_{\mathsf{BV}}{\downarrow}\frac{B}{A} \textit{ , then } \mathsf{r}\frac{B}{A} \qquad \textit{if } \mathsf{q}_{\mathsf{BV}}{\uparrow}\frac{B}{A} \textit{ , then } \mathsf{t}\frac{B}{A} \tag{9}$$

where $\mathsf{r} \in \{\mathsf{q}{\downarrow}, \mathsf{s}_{\parr}\}$ *and* $\mathsf{t} \in \{\mathsf{q}{\uparrow}, \mathsf{s}_{\otimes}\}$.

Proof. This can be seen immediately by inspecting the rules in Figure 2. ◀

We define the additional (non-atomic) ***interaction*** rules

$$\mathsf{i}{\downarrow}\frac{\varnothing}{G^\perp \parr G} \qquad \text{and} \qquad \mathsf{i}{\uparrow}\frac{G^\perp \otimes G}{\varnothing} \tag{10}$$

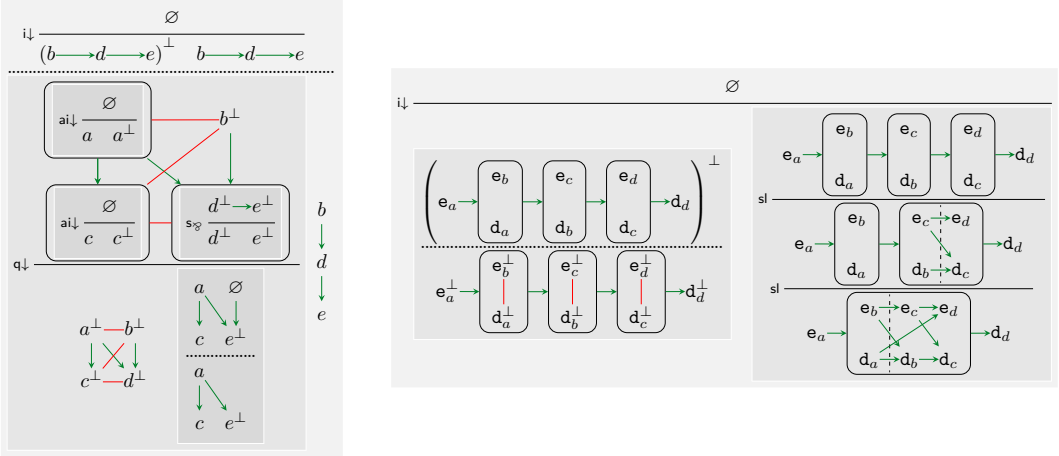


Figure 4 Examples of proofs in GV and $\mathsf{GV}^{\mathsf{sl}}$. As the conclusions are not relation webs, these proofs cannot be carried out in BV. The left proof is valid in GV and $\mathsf{GV}^{\mathsf{sl}}$. The right proof uses the slice rule and is only valid in $\mathsf{GV}^{\mathsf{sl}}$. It proves $\mathtt{Q}_2 \multimap \mathtt{Q}_3$ from the introduction.

The $\mathsf{i}\uparrow$ rule is the deep inference equivalent to the cut rule from sequent calculus and proving its admissibility is equivalent to proving the cut-elimination result. We recall some admissibility results for rules in BV and GS from the literature where we notice that the admissibility of cuts can be achieved by proving the admissibility of up-rules.

▶ **Theorem 22** ([19]). *Let G be a relation web. Then $\vdash_{\mathsf{BV}\cup\{\mathsf{i}\uparrow\}} G$ iff $\vdash_{\mathsf{BV}\cup\{\mathsf{ai}\uparrow,\mathsf{q}_{\mathsf{BV}}\uparrow\}} G$ iff $\vdash_{\mathsf{BV}} G$.*

▶ **Theorem 23** ([1]). *Let G be an undirected graph. Then $\vdash_{\mathsf{GS}\cup\{\mathsf{i}\uparrow\}} G$ iff $\vdash_{\mathsf{GS}\cup\{\mathsf{ai}\uparrow,\mathsf{s}_\otimes,\mathsf{p}\uparrow\}} G$ iff $\vdash_{\mathsf{GS}} G$.*

In the next section we are going to generalize these two theorems to GV and $\mathsf{GV}^{\mathsf{sl}}$. For doing so, we end this section with the following two auxiliary lemmas.

▶ **Lemma 24.** *Let $M_1, \ldots, M_n, N_1, \ldots, N_n$ and G be webs such that $|V_G| = n$. Then there is a complete graph $\mathbb{C}$ with $|V_{\mathbb{C}}| = n$ such that there are derivations*

$$\dfrac{\mathbb{C}\langle\!\langle M_1 \parr N_1, \ldots, M_n \parr N_n\rangle\!\rangle}{\mathcal{D}^{\downarrow} \,\|\, \{\mathsf{s}_\otimes,\mathsf{p}\downarrow,\mathsf{q}\downarrow\}} \\ G^{\perp}\langle\!\langle M_1, \ldots, M_n\rangle\!\rangle \parr G\langle\!\langle N_1, \ldots, N_n\rangle\!\rangle$$

and

$$\dfrac{G^{\perp}\langle\!\langle M_1, \ldots, M_n\rangle\!\rangle \otimes G\langle\!\langle N_1, \ldots, N_n\rangle\!\rangle}{\mathcal{D}^{\uparrow} \,\|\, \{\mathsf{s}_\parr,\mathsf{p}\uparrow,\mathsf{q}\uparrow\}} \\ \mathbb{C}^{\perp}\langle\!\langle M_1 \otimes N_1, \ldots, M_n \otimes N_n\rangle\!\rangle$$

Moreover, if $M_1, \ldots, M_n, N_1, \ldots, N_n$ are non-empty, then $\mathcal{D}^{\downarrow}$ contains no $\mathsf{s}_\otimes$ and $\mathcal{D}^{\uparrow}$ contains no $\mathsf{s}_\parr$.

Proof. We proceed by induction on the modular decomposition of G. If G is atomic, then we conclude by letting $\mathbb{C}$ be atomic and $\mathcal{D}^{\downarrow}$ trivial. Otherwise, without loss of generality we can assume $M_i \parr N_i \neq \varnothing$ for all $i \in \{1, \ldots, n\}$. In fact, if $M_i \parr N_i = \varnothing$ for some $i \in \{1, \ldots, n\}$, then we could consider a graph H with $|V_H| = |V_G| - 1$ such that

$$H^{\perp}\langle\!\langle M_1, \ldots, M_{i-1}, M_{i+1}, \ldots, M_n\rangle\!\rangle = G^{\perp}\langle\!\langle M_1, \ldots, M_{i-1}, \varnothing, M_{i+1}, \ldots, M_n\rangle\!\rangle \\ H\langle\!\langle N_1, \ldots, N_{i-1}, N_{i+1}, \ldots, N_n\rangle\!\rangle = G\langle\!\langle N_1, \ldots, N_{i-1}, \varnothing, N_{i+1}, \ldots, N_n\rangle\!\rangle$$

satisfying the condition. Then, by Theorem 5, we can write $G = P(\!|G_1, \ldots, G_m|\!)$ for a prime web P and webs $G_1, \ldots, G_m$. We can, without loss of generality, write

$$G(\!|N_1, \ldots, N_n|\!) = P(\!|G_1(\!|N_1, \ldots N_{k_1}|\!), \ldots, G_m(\!|N_{k_{m-1}+1}, \ldots N_n|\!)\!|)$$

- If $G_i^\perp(\!|M_{k_{i-1}+1}, \ldots, M_{k_i}|\!) = \varnothing$ for some $i \in \{1, \ldots, m\}$, then we can assume w.l.o.g. that $i = 1$. Then $N_1, \ldots, N_{k_1}$ must be non-empty since we are assuming $M_j \parr N_j \neq \varnothing$ for all $j \in \{1, \ldots, n\}$. Hence we can apply $\mathsf{s}_\otimes$ as follows and conclude by induction hypothesis:

$$\mathsf{2{\times}s}_\otimes \dfrac{\dfrac{\mathbb{C}_1(\!|N_1, \ldots, N_{k_1}|\!) \;\|\, IH \;\; G_1(\!|N_1, \ldots, N_{k_1}|\!)}{\mathbb{C}_1(\!|N_1, \ldots, N_{k_1}|\!)} \otimes \dfrac{\dfrac{\mathbb{C}_k(\!|M_{k_1+1} \parr N_{k_1+1}, \ldots, M_{k_2} \parr N_{k_2}|\!) \;\|\, IH \;\; G_1^\perp(\!|M_{k_1+1}, \ldots, M_{k_2}|\!) \parr G_1(\!|N_{k_1+1}, \ldots, N_{k_2}|\!)}{\mathbb{C}_k(\!|M_{k_1+1} \parr N_{k_1+1}, \ldots, M_{k_2} \parr N_{k_2}|\!)} \otimes \cdots \otimes \dfrac{\mathbb{C}_k(\!|M_{k_{n-1}+1} \parr N_{k_{n-1}+1}, \ldots, M_n \parr N_n|\!) \;\|\, IH \;\; G_1^\perp(\!|M_{k_{n-1}+1}, \ldots, M_n|\!) \parr G_1(\!|N_{k_{n-1}+1}, \ldots, N_n|\!)}{\mathbb{C}_k(\!|M_{k_{n-1}+1} \parr N_{k_{n-1}+1}, \ldots, M_n \parr N_n|\!)}}{\dfrac{H^\perp(\!|G_2^\perp(\!|M_{k_1+1}, \ldots M_{k_2}|\!), \ldots, G_m^\perp(\!|M_{k_{m-1}+1}, \ldots M_n|\!)|\!)}{P^\perp(\!|\varnothing, G_2^\perp(\!|M_{k_1+1}, \ldots M_{k_2}|\!), \ldots, G_m^\perp(\!|M_{k_{m-1}+1}, \ldots M_n|\!)|\!)} \parr \dfrac{H(\!|G_2(\!|N_{k_1+1}, \ldots, N_{k_2}|\!), \ldots, G_m(\!|N_{k_{m-1}+1}, \ldots, N_n|\!)|\!)}{P(\!|\varnothing, G_2(\!|N_{k_1+1}, \ldots, N_{k_2}|\!), \ldots, G_m(\!|N_{k_{m-1}+1}, \ldots, N_n|\!)|\!)} \;\|\, IH}}{P^\perp(\!|\varnothing, G_2^\perp(\!|M_{k_1+1}, \ldots M_{k_2}|\!), \ldots, G_m^\perp(\!|M_{k_{m-1}+1}, \ldots M_n|\!)|\!) \parr P(\!|G_1(\!|N_1, \ldots N_{k_1}|\!), G_2(\!|N_{k_1+1}, \ldots N_{k_2}|\!), \ldots, G_m(\!|N_{k_{m-1}+1}, \ldots N_n|\!)|\!)}$$

- If $G_i^\perp(\!|M_{k_{i-1}+1}, \ldots, M_{k_i}|\!) \neq \varnothing$ for all $i \in \{1, \ldots, m\}$, and $\overset{P}{\frown} = \varnothing$ then we can apply $\mathsf{p}{\downarrow}$ as follows

$$\mathsf{p}{\downarrow} \dfrac{\otimes_m \left(\dfrac{\mathbb{C}_1(\!|M_1 \parr N_1, \ldots, M_{k_1} \parr N_{k_1}|\!) \;\|\, IH \;\; G_1^\perp(\!|M_1, \ldots, M_{k_1}|\!) \parr G_1(\!|N_1, \ldots, N_{k_1}|\!)}{\mathbb{C}_1(\!|M_1 \parr N_1, \ldots, M_{k_1} \parr N_{k_1}|\!)}, \ldots, \dfrac{\mathbb{C}_m(\!|M_{k_{m-1}+1} \parr N_{k_{m-1}+1}, \ldots, M_n \parr N_n|\!) \;\|\, IH \;\; G_m^\perp(\!|M_{k_{m-1}+1}, \ldots, M_n|\!) \parr G_m(\!|N_{k_{m-1}+1}, \ldots, N_n|\!)}{\mathbb{C}_m(\!|M_{k_{m-1}+1} \parr N_{k_{m-1}+1}, \ldots, M_n \parr N_n|\!)} \right)}{P^\perp(\!|G_1^\perp(\!|M_1, \ldots, M_{k_1}|\!), \ldots, G_m^\perp(\!|M_{k_{m-1}+1}, \ldots, M_n|\!)|\!) \parr P(\!|G_1(\!|N_1, \ldots, N_{k_1}|\!), \ldots, G_m(\!|N_{k_{m-1}+1}, \ldots, N_n|\!)|\!)}$$

  where $\otimes_m$ is the complete undirected graph with m vertices, and conclude by induction hypothesis.

- If $\overset{P}{\frown} \neq \varnothing$, we conclude similarly to the previous case by using $\mathsf{q}{\downarrow}$ instead of $\mathsf{p}{\downarrow}$ and therefore replacing the $\otimes_m$ with $\mathbb{C} = P$ (if $\overset{P}{\smile} = \varnothing$) or $\mathbb{C} = P^\perp$ (if $\overset{P}{\frown} = \varnothing$). ◀

▶ **Lemma 25.** *For every* $\mathsf{r}{\downarrow} \in \mathsf{GV}$ *we have that* $\mathsf{r}{\uparrow}$ *is derivable in* $\mathsf{GV} \cup \{\mathsf{i}{\uparrow}\}$.

Proof. This follows in exactly the same way as in any other deep inference system [21]. ◀

5 Cut elimination

The $\mathsf{i}{\uparrow}$ rule in (10) is the deep inference equivalent to the cut rule. We show in this section, that this rule is admissible for our systems, i.e., every web provable with cut is also provable without.

▶ **Theorem 26.** *The rule* $\mathsf{i}{\uparrow}$ *is admissible for* GV *and for* $\mathsf{GV}^{\mathsf{sl}}$.

The main consequence of this theorem is the transitivity of the consequence relation defined by implication $A \multimap B = A^\perp \parr B$ is transitive.

▶ **Corollary 27.** *Let* A, B, *and* C *be any webs. If* $A \multimap B$ *and* $B \multimap C$ *are provable in* GV *(resp.* $\mathsf{GV}^{\mathsf{sl}}$*), then so is* $A \multimap C$.

Proof. Given proofs of $A^\perp \parr B$ and $B^\perp \parr C$, we also have a proof of $(A^\perp \parr B) \otimes (B^\perp \parr C)$. Applying $\mathsf{s}_\parr$ twice gives us $A^\perp \parr (B \otimes B^\perp) \parr C$, and then we can obtain $A^\perp \parr C$, via $\mathsf{i}{\uparrow}$. ◀

To prove Theorem 26, we rely on methods developed for deep inference in general [38, 21, 19, 4] and for graphical systems in particular [1]. There are two steps:

$$\vdash_{\mathsf{GX} \cup \{\mathsf{i}{\uparrow}\}} G \iff \vdash_{\mathsf{GX} \cup \{\mathsf{ai}{\uparrow}, \mathsf{p}{\uparrow}, \mathsf{q}{\uparrow}\}} G \iff \vdash_{\mathsf{GX}} G \tag{11}$$

where $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$ and G is an arbitrary web. The first step is decomposing the general $\mathsf{i}\uparrow$ (which introduces an arbitrary web when going up in a proof) into smaller steps that are easier to control, and it is achieved with Lemma 25 and the following proposition, that is an immediate consequence of Lemma 24.

▶ **Proposition 28.** *The rule* $\mathsf{i}\uparrow$ *can be derived with* $\{\mathsf{ai}\uparrow, \mathsf{p}\uparrow, \mathsf{q}\uparrow\}$. *And dually,* $\mathsf{i}\downarrow$ *can be derived with* $\{\mathsf{ai}\downarrow, \mathsf{p}\downarrow, \mathsf{q}\downarrow\}$.

It remains to show the second step in (11), i.e., the following theorem.

▶ **Theorem 29.** *The rules* $\mathsf{ai}\uparrow$, $\mathsf{p}\uparrow$, *and* $\mathsf{q}\uparrow$ *are admissible for* GV *and* $\mathsf{GV}^{\mathsf{sl}}$.

Proving this theorem follows the same outline as in other deep inference systems, via *splitting* and *context reduction* stated below.

▶ **Lemma 30** (Tensor Splitting). *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$, *let* A *and* B *be nonempty webs, and let* G *be an arbitrary web. If* $\vdash_{\mathsf{GX}} G \parr (A \otimes B)$, *then there are webs* K_A *and* K_B *such that there are derivations*

$$\begin{array}{c} K_A \parr K_B \\ \mathcal{D}_G \parallel \mathsf{GX} \\ G \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_A \parallel \mathsf{GX} \\ K_A \parr A \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_B \parallel \mathsf{GX} \\ K_B \parr B \end{array} .$$

Note that the well-known splitting tensor lemma for linear logic proof-nets [10] states the existence of a splitting tensor, whereas Lemma 30 (and any splitting tensor lemma in deep inference in general) says that every tensor can be made splitting by first reducing the context.

▶ **Lemma 31** (Atomic Splitting). *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$ *and* G *be a web. If* $\vdash_{\mathsf{GX}} G \parr a$ *for an atom* a, *then there is a derivation*

$$\begin{array}{c} a^\perp \\ \mathcal{D}_G \parallel \mathsf{GX} \\ G \end{array} .$$

▶ **Lemma 32** (Context Reduction). *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$, *Let* A *be a web and* $\mathcal{C}[\Box]$ *be a context, such that* $\vdash_{\mathsf{GX}} \mathcal{C}[A]$. *Then there is a web* K, *such that for any web* $\mathcal{X}$ *there are derivations*

$$\begin{array}{c} K \parr \mathcal{X} \\ \mathcal{D}_{\mathcal{X}} \parallel \mathsf{GX} \\ \mathcal{C}[\mathcal{X}] \end{array} \quad \textit{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_A \parallel \mathsf{GX} \\ K \parr A \end{array}$$

These three lemmas are already enough to prove the admissibility of $\mathsf{ai}\uparrow$ for GV and $\mathsf{GV}^{\mathsf{sl}}$: Assume we have a proof of $\mathcal{C}[a^\perp \otimes a]$. By the three lemmas above, we get webs K, $K_{a^\perp}$, K_a and derivations

$$\begin{array}{c} K \parr \varnothing \\ \mathcal{D}_\varnothing \parallel \mathsf{GX} \\ \mathcal{C}[\varnothing] \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_K \parallel \mathsf{GX} \\ K \parr (a^\perp \otimes a) \end{array} \quad \begin{array}{c} K_{a^\perp} \parr K_a \\ \mathcal{D}_K \parallel \mathsf{GX} \\ K \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_{a^\perp} \parallel \mathsf{GX} \\ K_{a^\perp} \parr a^\perp \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_a \parallel \mathsf{GX} \\ K_a \parr a \end{array} \quad \begin{array}{c} a \\ \mathcal{D}_1 \parallel \mathsf{GX} \\ K_{a^\perp} \end{array} \quad \begin{array}{c} a^\perp \\ \mathcal{D}_2 \parallel \mathsf{GX} \\ K_a \end{array}$$

With the derivations $\mathcal{D}_\varnothing$, $\mathcal{D}_K$, $\mathcal{D}_1$, $\mathcal{D}_2$ and an instance of $\mathsf{ai}\downarrow$, we can now obtain a proof of $\mathcal{C}[\varnothing]$, as desired.

In order to prove admissibility of $\mathsf{p}\uparrow$ and of $\mathsf{q}\uparrow$ in a similar way, we need two more variants of the splitting lemma. The first (Lemma 33) deals with prime webs in general, and the second (Lemma 34) deals with complete webs, i.e., webs in which for any two vertices x, y, we either have $x \frown y$ or $x \curvearrowright y$ or $x \curvearrowleft y$.

▶ **Lemma 33** (Prime Web Splitting). *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$*, let* $P \neq \parr$ *be a prime web with* $|V_P| = n$*, let* $M_1, \ldots, M_n$ *be non-empty webs, and let* G *be an arbitrary web. If* $\vdash_{\mathsf{GX}} G \parr P(\!|M_1, \ldots, M_n|\!)$*, then one of the following holds:*

(A) *there are webs* $K_1, \ldots, K_n$*, such that there are derivations*

$$\begin{array}{c} P^{\perp}(\!|K_1, \ldots, K_n|\!) \\ \mathcal{D}_G \,\|\, \mathsf{GX} \\ G \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GX} \\ K_i \parr M_i \end{array} \qquad \text{for all } i \in \{1, \ldots, n\};$$

(B) *there are webs* K_X *and* K_Y*, such that there are derivations*

$$\begin{array}{c} K_X \parr K_Y \\ \mathcal{D}_G \,\|\, \mathsf{GX} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GX} \\ K_X \parr M_i \end{array} \text{ and } \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GX} \\ K_Y \parr P(\!|M_1, \ldots, M_{i-1}, \varnothing, M_{i+1}, \ldots, M_n|\!) \end{array} \quad \text{for some } i \in \{1, \ldots, n\};$$

(C) *only if* $\mathsf{GX} = \mathsf{GV}^{\mathsf{sl}}$ *and* $\overset{P}{\frown} = \varnothing$*: there are webs* K_X *and* K_Y*, and some* $k \in \{2, \ldots, n-1\}$ *such that, w.l.o.g.,* $y \not\curvearrowright x$ *for all* $x \in \bigcup_{i=1}^{k} M_i$ *and* $y \in \bigcup_{j=k+1}^{n} M_j$*, and there are derivations*

$$\begin{array}{c} K_X \triangleleft K_Y \\ \mathcal{D}_G \,\|\, \mathsf{GX} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GX} \\ K_X \parr P(\!|M_1, \ldots, M_k, \varnothing, \ldots, \varnothing|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GX} \\ K_Y \parr P(\!|\varnothing, \ldots, \varnothing, M_{k+1}, \ldots, M_n|\!) \end{array}$$

▶ **Lemma 34** (Complete Web Splitting). *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$*. Let* $\mathbb{C}$ *be a complete web with* $|V_{\mathbb{C}}| = n \geq 2$*, let* $M_1, \ldots, M_n$ *be non-empty webs, and let* G *be an arbitrary web. If* $\vdash_{\mathsf{GV}} G \parr \mathbb{C}(\!|M_1, \ldots, M_n|\!)$*, then there are webs* $K_1, \ldots, K_n$ *such that there are derivations*

$$\begin{array}{c} \mathbb{C}^{\perp}(\!|K_1, \ldots, K_n|\!) \\ \mathcal{D}_G \,\|\, \mathsf{GX} \\ G \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GX} \\ K_i \parr M_i \end{array} \qquad \text{for all } i \in \{1, \ldots, n\}.$$

These lemmas are now sufficient to complete the proof of rule elimination (Theorem 29).

Proof of Theorem 29. For $\mathsf{ai}\uparrow$, this has been shown above. For $\mathsf{p}\uparrow$, we proceed similarly as in [1, 2] for GS. For $\mathsf{q}\uparrow$, assume $\vdash_{\mathsf{GX}} \mathcal{C}\left[Q(\!|M_1, \ldots, M_n|\!) \otimes Q^{\perp}(\!|N_1, \ldots, N_n|\!)\right]$. By Lemma 32 there is a web K, such that for any web $\mathcal{X}$ there are the derivations

$$\begin{array}{c} K \parr \mathcal{X} \\ \mathcal{D}_{\mathcal{X}} \,\|\, \mathsf{GX} \\ \mathcal{C}[\mathcal{X}] \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}' \,\|\, \mathsf{GX} \\ K \parr Q(\!|M_1, \ldots, M_n|\!) \otimes Q^{\perp}(\!|N_1, \ldots, N_n|\!) \end{array}$$

By Lemma 30 we have webs K_X and K_Y and the derivations below left

$$\begin{array}{c} K_X \parr K_Y \\ \mathcal{D}_K \,\|\, \mathsf{GX} \\ K \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \,\|\, \mathsf{GX} \\ K_X \parr Q(\!|M_1, \ldots, M_n|\!) \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \,\|\, \mathsf{GX} \\ K_Y \parr Q^{\perp}(\!|N_1, \ldots, N_n|\!) \end{array} \quad \Bigg| \quad \begin{array}{c} Q(\!|K_1, \ldots, K_n|\!) \\ \mathcal{D}_Q \,\|\, \mathsf{GX} \\ K_Y \end{array} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_i \,\|\, \mathsf{GX} \\ K_i \parr N_i \end{array}$$

Since $\overset{Q}{\frown} = \varnothing$, we have that $Q^\perp$ is a complete web, and we can apply Lemma 34 to $\mathcal{D}_Y$ giving us $K_1, \ldots, K_n$ and the derivations above right (for all $i \in \{1, \ldots, n\}$). We conclude with

$$\dfrac{\dfrac{\varnothing}{\mathcal{D}_X \,\|\, K_X \parr Q}\left(\dfrac{\dfrac{\varnothing}{\mathcal{D}_1 \,\|\, M_1 \otimes N_1}}{K_1 \parr (M_1 \otimes N_1)}\mathsf{s}_\parr, \ldots, \dfrac{\dfrac{\varnothing}{\mathcal{D}_n \,\|\, M_n \otimes N_n}}{K_n \parr (M_n \otimes N_n)}\mathsf{s}_\parr\right)}{\dfrac{K_X \parr \dfrac{Q(\!|K_1, \ldots, K_n|\!)}{\mathcal{D}_Q \,\|\, K_Y}}{\mathcal{D}_K \,\|\, K} \parr Q(\!|M_1 \otimes N_1, \ldots, M_n \otimes N_n|\!)}\mathsf{qm}}{\mathcal{D}_{Q(\!|M_1 \otimes N_1, \ldots, M_n \otimes N_n|\!)} \,\|\, \mathcal{C}\,[Q(\!|M_1 \otimes N_1, \ldots, M_n \otimes N_n|\!)]} \tag{12}$$

◀

Observe that Lemma 30 is just a special case of Lemma 34 and also a special case of Lemma 33, as cases (A) and (B) of Lemma 33 collapse both to Lemma 30 if $P = \otimes$. Therefore, it only remains to prove Lemma 31, Lemma 32, Lemma 33, and Lemma 34, which is done by induction on the size of the web to be proved.

▶ **Definition 35.** *We define the* ***size*** *of a web G as the pair $\|G\| = \langle |V_G|, |\frown| + 2|\smile| \rangle$, ordered lexicographically, i.e., if $\|G\| = \langle n_G, m_G \rangle$ and $\|H\| = \langle n_H, m_H \rangle$, then we have $G < H$ if $n_G < n_H$ or if $n_G = n_H$ and $m_G < m_H$.*

Even though the general pattern of the overall argument to prove Theorem 26 is similar to other deep inference systems, the details are much more involved. Some of the problems have already been observed for the system GS in [1]. For example, Lemma 32, Lemma 33 and Lemma 34 are not proved one after the other, but simultaneously, by mutual induction on the size (defined above) of the conclusion. An additional difficulty for GV and $\mathsf{GV}^{\mathsf{sl}}$ is the explosion of the cases to consider. When proving Lemma 33, we have to do a case analysis on the last rule applied to the proof of $G \parr P(\!|M_1, \ldots, M_n|\!)$. More details on the splitting proof can be found in Appendix A.

▶ Remark 36. The $\mathsf{s}_\otimes$ is usually admissible in a deep inference system, as it is an *up-rule* (and dual to $\mathsf{s}_\parr$). This was also the case for GS [1]. However, when we consider mixed graphs, the rule is no longer admissible. Or, put differently, cut elimination does not hold if we remove the rule from the system (see Appendix B).

▶ **Example 37.** To illustrate the dynamics of splitting, consider the alternative proofs in Figure 5 which prove the same web as found in the conclusion of the proof on the left of Figure 4. The proofs in Figure 5 can be obtained by applying splitting to the first proof in Figure 4. For the proof on the left of Figure 5, apply Lemma 34 to the leftmost disjoint graph, which ensures the existence of the derivation found below the instance of the $\mathsf{q}{\downarrow}$ rule, that does not change the sub-graph selected. Similarly, the second proof in Figure 5 is obtained by applying splitting to the connected sub-graph labelled with a, c and $e^\perp$.

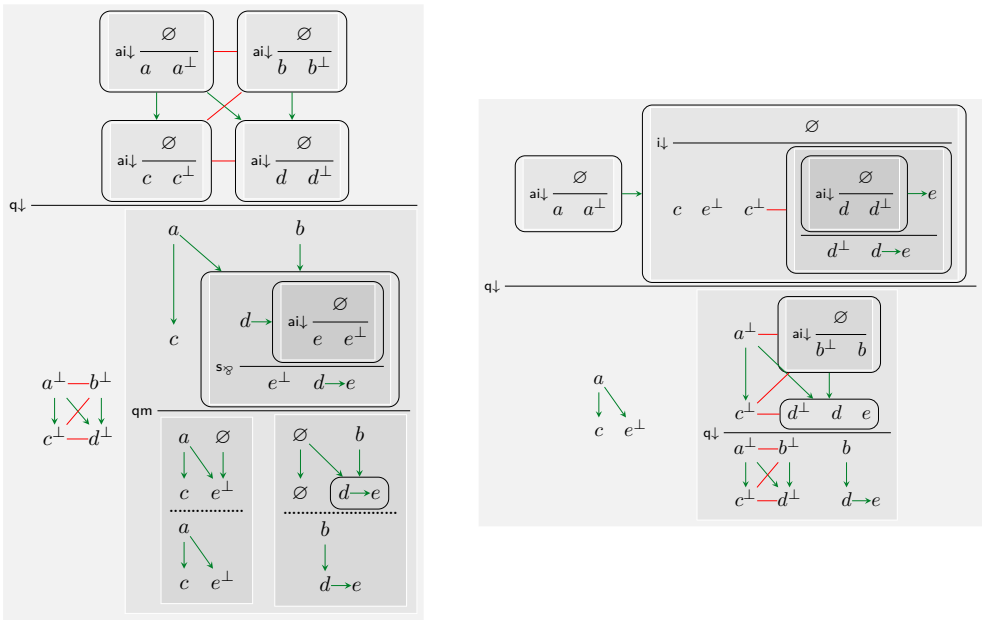


Figure 5 Alternative proofs of the web proved by the left derivation in Figure 4.

6 Analyticity, Conservativity, Complexity

In this section, we discuss some consequences of the cut-elimination result obtained in the previous section. The most important is that GV and GV$^{\mathsf{sl}}$ are both analytic. Since deep inference systems in general, and our graphical proof systems in particular, do not have a "subformula property" in the traditional sense, we present here a formal notion of what we mean by analyticity of a graphical proof system. For this, we start from the notion of *connector* given in Definition 8.

▶ **Definition 38.** *A **subconnector** of G is a prime web that is an induced subgraph of a connector in G. A connector (or subconnector) P is called **proper** if $\overset{P}{\frown} \neq \varnothing$. A proof $\mathcal{D}$ of G is **analytic** if every proper connector of a web in $\mathcal{D}$ is a subconnector of G.*

▶ **Theorem 39** (Analyticity). *If $\vdash_{\mathsf{GX}} G$ for $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$, then G admits an analytic proof in GX.*

Proof Sketch. This is proved similarly to the same result for GS given in [2], by taking into account the additional rules in GV and GV$^{\mathsf{sl}}$. When going bottom-up in a derivation (as in proof search), the only two rules that can introduce a new proper connector that is not a subconnector of the conclusion are $\mathsf{s}_{\parr}$ (when the N in Figure 2 is empty) and $\mathsf{q}{\downarrow}$ (when L_i in Figure 2 is empty for some $i \in \{1, \ldots, |V_Q|\}$). Since the premise of every proof is $\varnothing$, all these connectors have to be destroyed eventually. This can happen via the rules $\mathsf{ai}{\downarrow}$ or $\mathsf{s}_{\otimes}$ (one immediate submodule of the connector is removed), or via the rule $\mathsf{p}{\downarrow}$ or $\mathsf{q}{\downarrow}$ (the connector itself is destroyed, the submodules remain). It can now be shown via rule permutations that the rule instance that creates the connector and the rule instance that destroys it can be brought together, such that we can rearrange the derivation in a way that the offending connector does not occur. ◀

▶ **Corollary 40.** *Let* $\mathsf{GX} \in \{\mathsf{GV}, \mathsf{GV}^{\mathsf{sl}}\}$ *and let* G *be an undirected graph. If* $\vdash_{\mathsf{GX}} G$*, then* G *admits a derivation* $\mathcal{D}$ *in* GX *such that every connector of a web in* $\mathcal{D}$ *is a subconnector of* G*.*

Proof Sketch. For proper connectors this has been shown in Theorem 39. Hence we only need to consider P-connectors that are directed prime graphs. The only way to create such a new P-connector is via $\mathsf{s}_{⅋}$, $\mathsf{q}{\downarrow}$,or qm. Then P is either destroyed by destroying all its modules, or it is destroyed after merging via qm (or $\mathsf{q}{\downarrow}$ if $P = \triangleleft$) with another P-connector. In both cases we use a similar rule permutation argument as in the proof of Theorem 39 to eliminate the occurrence of the P-connector. ◀

As a consequence of Theorem 39, we know that, in an analytic proof, any proper connector occurring in the conclusion of a $\mathsf{q}{\downarrow}$ also occurs in its premise. This condition is not required by definition for the rule $\mathsf{q}{\downarrow}$ (see Figure 2) where the side condition only asks that $L_i ⅋ J_i$ is non-empty: the rule might introduce bottom-up a new connector whenever a L_i is empty. The inclusion of the additional restriction $L_i \neq \varnothing$ for all $i \in \{1, \ldots, n\}$ for $\mathsf{q}{\downarrow}$ in the definition of the rule would require a more complex formulation of the splitting statement.

To better understand that subtlety consider the example in Figure 6. The proof on the left is obtained by applying splitting to the leftmost connected component of the conclusion; while the proof on the right has a shape obtained by applying splitting to the rightmost connected component. In the latter proof, the bottommost $\mathsf{q}{\downarrow}$ introduces a new connector establishing relations between the modules of the leftmost and central components of the conclusion. As indicated in the figure, this is achieved by introducing empty modules in different places in both webs affected by that instance of the $\mathsf{q}{\downarrow}$ rule, making use of the weak side condition mentioned above. The application of a restricted version of the $\mathsf{q}{\downarrow}$ would force the merge of the two webs to establish additional relations which later would prevent a successful proof. However, observe that this proof violates analyticity.

In contrast, the proof on the left is analytic. Even though that proof also appeals to the same weak side conditions and introduces in the premise of the qm-rule instance a new connector that is not a subconnector of the conclusion, this new connector is not proper because it contains no ⌒-edge. This is the reason for restricting our analyticity result to proper connectors.

Nonetheless, this restricted form of analyticity is enough to show that both, GV and $\mathsf{GV}^{\mathsf{sl}}$, are conservative extensions of GS and of BV.

▶ **Theorem 41** (Conservativity)**.** *Let* G *be an undirected graph. Then* $\vdash_{\mathsf{GS}} G$ *iff* $\vdash_{\mathsf{GV}} G$ *iff* $\vdash_{\mathsf{GV}^{\mathsf{sl}}} G$*.*

Proof. Assume we have a derivation of G in GV or $\mathsf{GV}^{\mathsf{sl}}$. By Corollary 40, there is also a derivation in which all occurring webs are also undirected graphs. Hence all occurring rules are $\mathsf{ai}{\downarrow}$, $\mathsf{s}_{⅋}$, $\mathsf{s}_{\otimes}$, and $\mathsf{p}{\downarrow}$. Now the result follows from admissibility of $\mathsf{s}_{\otimes}$ for GS (see Theorem 23). ◀

▶ **Theorem 42** (Conservativity)**.** *Let* G *be a relation web. Then* $\vdash_{\mathsf{BV}} G$ *iff* $\vdash_{\mathsf{GV}} G$ *iff* $\vdash_{\mathsf{GV}^{\mathsf{sl}}} G$*.*

Proof. In Lemma 21, we have shown that every proof of BV is also a proof in GV, and therefore also in $\mathsf{GV}^{\mathsf{sl}}$. Conversely, assume we have a proof $\mathcal{D}$ of G in GV or $\mathsf{GV}^{\mathsf{sl}}$. Since G is a relation web, all its connectors are in $\{\otimes, \triangleleft, ⅋\}$. Hence, by Corollary 40, there is a derivation $\tilde{\mathcal{D}}$ in which each occurring web is a relation web. Therefore, each instance of $\mathsf{p}{\downarrow}$ can be simulated by two instances of s; each $\mathsf{s}_{⅋}$ is an instance of s or $\mathsf{q}_{\mathsf{BV}}{\downarrow}$; each instance of $\mathsf{s}_{\otimes}$ is an instance of s or $\mathsf{q}_{\mathsf{BV}}{\uparrow}$; each instance of $\mathsf{q}{\downarrow}$ or qm or sl is an instance of $\mathsf{q}_{\mathsf{BV}}{\downarrow}$, and each instance of $\mathsf{q}{\uparrow}$ is an instance of $\mathsf{q}_{\mathsf{BV}}{\uparrow}$. Now the result follows from the fact that $\mathsf{q}_{\mathsf{BV}}{\uparrow}$ is admissible for BV (see Theorem 22). ◀

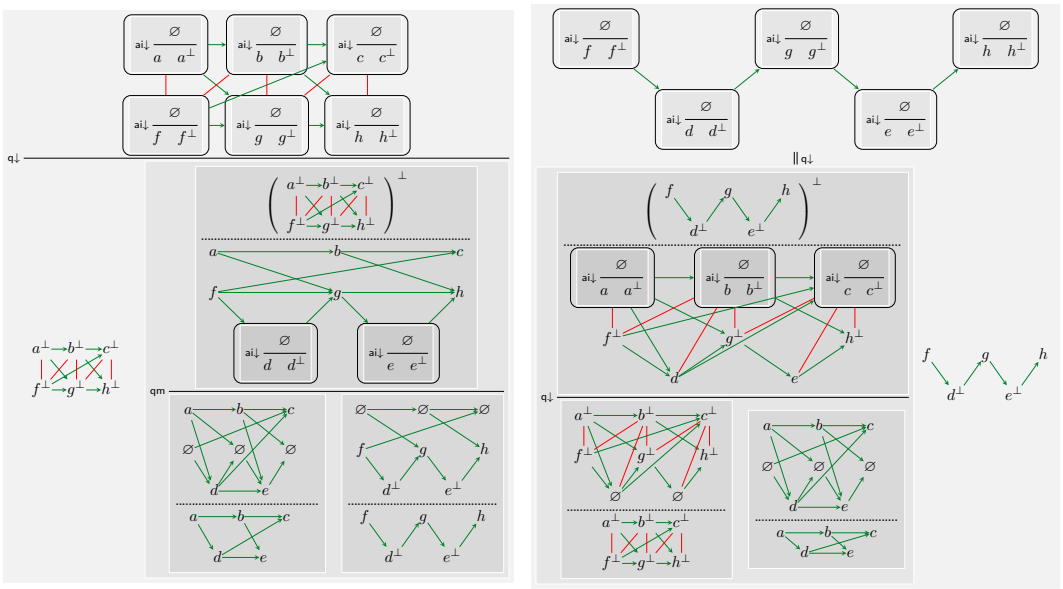


Figure 6 Two proofs of the same web. Only the derivation on the left is analytic: the qm in the derivation on the left introduces a new connector, which is not proper; the bottommost $\mathsf{q}{\downarrow}$ in the derivation on the right introduces a new proper connector, violating analyticity.

▶ **Corollary 43.** *Provability in* GV *and in* $\mathsf{GV}^{\mathsf{sl}}$ *is* **NP***-complete.*

Proof. It has been shown before that provability in BV is **NP**-complete [28]. Hence, by Theorem 42 we obtain **NP**-hardness of provability in GV and $\mathsf{GV}^{\mathsf{sl}}$. For containment in **NP**, observe that for any rule instance $\mathsf{r}\dfrac{H}{G}$ in $\mathsf{GV}^{\mathsf{sl}}$ we have that $\|H\| < \|G\|$. For a web G we have $|\overset{G}{\frown}| + 2|\overset{G}{\smile}| < 2(|V_G|^2)$. Therefore any derivation in $\mathsf{GV}^{\mathsf{sl}}$ (or in GV) of a web G has at most length $\mathcal{O}(|V_G|^3)$. ◀

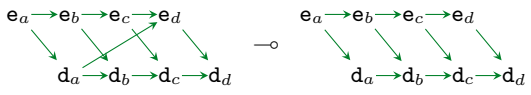
7 Conclusion

We presented two analytic proof systems, GV and $\mathsf{GV}^{\mathsf{sl}}$ (Figure 2), that conservatively extend the logic BV to graphs (Theorem 42). The graphs, called webs (Definition 12), feature both undirected and directed edges, which generalise, respectively, the multiplicative conjunction of linear logic, and the non-commutative connective *seq* of BV. In addition, our systems are conservative extensions of GS (Theorem 41), which features general simple graphs. This is no coincidence, since GS was conceived precisely as the minimal core of graphical logics like GV for which it was already clear that we needed to generalise the tools of proof theory. Indeed, we were able to adapt the deep inference technology developed for GS to prove cut elimination for GV and $\mathsf{GV}^{\mathsf{sl}}$ (Theorem 26 and Section 5).

Differences compared to GS surprised us. Notably, the $\mathsf{s}_{\otimes}$ rule is not admissible in GV (explained further in Appendix B), which in fact simplifies splitting (Lemmas 30-34), since $\mathsf{s}_{\otimes}$ can be used to remove contexts (Lemma 32). A nuance of GV is that rules for directed graphs require weaker side-conditions than similar rules for symmetric graphs. For example, if we restrict $\mathsf{q}{\downarrow}$ and qm such that all modules in one of the graphs are non-empty, then there is no proof of the web in Fig. 6 – an observation we can use to prove that such a restricted system cannot satisfy cut elimination. This forced us to take additional care defining analyticity Theorem 39, used to establish conservativity.

We draw attention to an open problem that we found more challenging than expected. Ideally, we would like to have an extension of GV admitting the ***homomorphism rule*** below on the left, making it possible to prove additional graphs beyond the scope of the systems proposed in this paper.

$$\mathsf{h}{\downarrow}\ \frac{H\langle\!|M_1,\ldots,M_n|\!\rangle}{G\langle\!|M_1,\ldots,M_n|\!\rangle}\ V_G{=}V_H,\ \overset{H}{\frown}{=}\overset{G}{\frown}{=}\varnothing,\ \overset{H}{\curvearrowright}{\supset}\overset{G}{\curvearrowright}$$



By means of example, the implication $\mathtt{Q}_3 \multimap \mathtt{Q}_4$ above on the right, internalising in the graphical logic the fact that a 4-queue can simulate the behaviour of a 3-queue, cannot be proven in $\mathsf{GV}^{\mathsf{sl}}$. The challenge we encounter in handling graph homomorphisms is due to its wild behaviour on graph modular decomposition. In fact, the rules $\mathsf{q}_{\mathsf{BV}}{\downarrow}$, qm and sl are special instances of this homomorphism rule with a controlled behaviour with respect to modular decomposition. Further insight is required to prove cut elimination for extensions of GV where the rule $\mathsf{h}{\downarrow}$ is admissible. The system $\mathsf{GV}^{\mathsf{sl}}$, which demanded a dedicated case in splitting (Lemma 33) to handle sl, is a first step towards that aim.

References

1 Matteo Acclavio, Ross Horne, and Lutz Straßburger. Logic beyond formulas: A proof system on graphs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 38–52, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394763.

2 Matteo Acclavio, Ross Horne, and Lutz Straßburger. An analytic propositional proof system on graphs. *Logical Methods in Computer Science*, 2022. to appear. doi:10.48550/arXiv.2012.01102.

3 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.

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

5 Andrea Aler Tubella and Lutz Straßburger. Introduction to deep inference. Lecture, August 2019. URL: https://hal.inria.fr/hal-02390267.

6 Denis Bechet, Philippe de Groote, and Christian Retoré. A complete axiomatisation of the inclusion of series-parallel partial orders. In H. Common, editor, *Rewriting Techniques and Applications, RTA 1997*, volume 1232 of *LNCS*, pages 230–240. Springer, 1997.

7 Carmen Bruckmann, Peter F. Stadler, and Marc Hellmuth. From modular decomposition trees to rooted median graphs. *Discrete Applied Mathematics*, 310:1–9, 2022. doi:10.1016/j.dam.2021.12.017.

8 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.

9 Cameron Calk, Anupam Das, and Tim Waring. Beyond formulas-as-cographs: an extension of Boolean logic to arbitrary graphs, 2020. doi:10.48550/arXiv.2004.12941.

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

11 Yuxin Deng, Robert J. Simmons, and Iliano Cervesato. Relating reasoning methodologies in linear logic and process algebra. *Mathematical Structures in Computer Science*, 26(5):868–906, 2016. doi:10.1017/S0960129514000413.

12 Pierre-Malo Deniélou and Nobuko Yoshida. Buffered communication analysis in distributed multiparty sessions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory*, pages 343–357, Berlin, Heidelberg, 2010. Springer. doi:10.1007/978-3-642-15375-4_24.

13 A. Ehrenfeucht, T. Harju, and G Rozenberg. *The Theory of 2-Structures A Framework for Decomposition and Transformation of Graphs*. World Scientific, 1999. doi:10.1142/4197.

14 Uli Fahrenberg, Christian Johansen, Georg Struth, and Ratan Bahadur Thapa. Generating posets beyond n. In Uli Fahrenberg, Peter Jipsen, and Michael Winter, editors, *Relational and Algebraic Methods in Computer Science*, pages 82–99, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-43520-2_6.

15 Xiang Fu, Tevfik Bultan, and Jianwen Su. Analysis of interacting BPEL web services. In *Proceedings of the 13th international conference on World Wide Web*, pages 621–630. ACM, 2004. doi:10.1145/988672.988756.

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

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

18 Jay L. Gischer. The equational theory of pomsets. *Theor. Comput. Sci.*, 61:199–224, 1988. doi:10.1016/0304-3975(88)90124-7.

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

20 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.

21 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.

22 Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010. doi:10.1016/j.cosrev.2010.01.001.

23 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.

24 Ross Horne, Sjouke Mauw, and Alwen Tiu. Semantics for specialising attack trees based on linear logic. *Fundam. Inform.*, 153(1-2):57–86, 2017. doi:10.3233/FI-2017-1531.

25 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.

26 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.

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

28 Ozan Kahramanoğulları. System BV is NP-complete. *Annals of Pure and Applied Logic*, 152(1-3):107–121, 2008. doi:10.1016/j.apal.2007.11.005.

29 Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978. doi:10.1145/359545.359563.

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

31 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.

32 Lê Thành Dũng Nguyễn and Lutz Straßburger. BV and Pomset Logic are not the same. In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2022.3.

33 Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13(1):85–108, 1981. Special Issue Semantics of Concurrent Computation. doi:10.1016/0304-3975(81)90112-2.

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

35 Christian Retoré. Perfect matchings and series-parallel graphs: multiplicative proof nets as R&B-graphs. *Electronic Notes in Theoretical Computer Science*, 3, 1996. doi:10.1016/S1571-0661(05)80416-5.

36 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.

37 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.

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

39 Jacobo Valdes, Robert E Tarjan, and Eugene L Lawler. The recognition of series parallel digraphs. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 1–12. ACM, 1979. doi:10.1137/0211023.

A Proof of Splitting and Context Reduction

Sketch of Proof of Splitting Lemma for Prime Webs (Lemma 33). The proof proceeds by case analysis of the last rule in a derivation of a graph $G \parr P(\!|M_1, \dots, M_n|\!)$.

- the last rule r acts inside G or any of the M_i,

$$\dfrac{\dfrac{\varnothing}{\mathcal{D}' \parallel \mathsf{GX}}}{\mathsf{r}\dfrac{G'}{G} \parr P(\!|M_1, \dots, M_n|\!)} \quad \text{or} \quad \dfrac{\dfrac{\varnothing}{\mathcal{D}' \parallel \mathsf{GX}}}{G \parr P(\!|M_1, \dots, M_{i-1}, \mathsf{r}\dfrac{M_i'}{M_i}, M_{i+1}, \dots, M_n|\!)}$$

  then we can conclude by inductive hypothesis using a rules permutation argument.

- the last rule is a $\mathsf{s}_\parr$ moving a connected component of $G = G'' \parr G'$ inside P, that is,

$$\mathsf{s}_\parr \dfrac{\dfrac{\varnothing}{\mathcal{D}'' \parallel \mathsf{GX}}}{G'' \parr P(\!|M_1, \dots, M_{i-1}, G' \parr M_i, M_{i+1} \dots, M_n|\!)}{G'' \parr G' \parr P(\!|M_1, \dots, M_n|\!)} \quad \text{or} \quad \mathsf{s}_\parr \dfrac{\dfrac{\varnothing}{\mathcal{D}'' \parallel \mathsf{GX}}}{G'' \parr R(\!|M_1', \dots, M_i', G', M_{i+1}', \dots, M_n'|\!)}{G'' \parr G' \parr \dfrac{R(\!|M_1', \dots, M_i', \varnothing, M_{i+1}', \dots, M_n'|\!)}{P(\!|M_1, \dots, M_k|\!)}}$$

  and we prove the result by inductive hypothesis on the conclusion of $\mathcal{D}''$

- the last rule is a $\mathsf{s}_{\parr}$ moving the connected component $P(\!|M_1, \dots, M_n|\!)$ inside G, that is,

$$\mathsf{s}_{\parr}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G'' \parr R(\!|N_1, \dots, N_{i-1}, P(\!|M_1, \dots, M_n|\!) \parr N_i, N_{i+1}, \dots, N_m|\!)\end{array}}{G'' \parr R(\!|N_1, \dots, N_m|\!) \parr P(\!|M_1, \dots, M_n|\!)}$$

  In this case, we conclude by inductive hypothesis using Lemma 32.

- the last rule is a $\mathsf{s}_{\otimes}$, leading to one of the following two cases:

$$\mathsf{s}_{\otimes}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}'' \parallel \mathsf{GX}\\ G \parr (M_i \otimes P(\!|M_1, \dots, M_{i-1}, \varnothing, M_{i+1}, \dots, M_n|\!))\end{array}}{G \parr P(\!|M_1, \dots, M_n|\!)} \quad \text{or} \quad \mathsf{s}_{\otimes}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}'' \parallel \mathsf{GX}\\ G \parr (P(\!|M_1, \dots, M_{i-1}, M_i, M_{i+1} \dots, M_n|\!) \otimes M_i')\end{array}}{G \parr P(\!|M_1, \dots, M_{i-1}, M_i \otimes M_i', M_{i+1} \dots, M_n|\!)}$$

  Using Lemma 30 the first case immediately leads to Case (B), while the second requires the use of inductive hypothesis to conclude.

- the last rule is a $\mathsf{p}{\downarrow}$, leading to one of the two following cases:

$$\mathsf{p}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G'' \parr ((N_1 \parr M_1) \otimes \dots \otimes (N_n \parr M_n))\end{array}}{G'' \parr P^{\perp}(\!|N_1, \dots, N_n|\!) \parr P(\!|M_1, \dots, M_n|\!)} \quad \text{or} \quad \mathsf{p}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G'' \parr (J_1 \otimes (J_2 \parr L_2) \otimes \dots \otimes (J_k \parr L_k))\end{array}}{G'' \parr R^{\perp}(\!|J_1, \dots, J_k|\!) \parr \dfrac{R(\!|\varnothing, L_2, \dots, L_k|\!)}{P(\!|M_1, \dots, M_n|\!)}}$$

  In the first case we conclude by Lemma 34, while the second requires Lemma 24 and inductive hypothesis to conclude.

- the last rule is a $\mathsf{q}{\downarrow}$ or qm, that is, $\mathcal{D}$ is of the following shape for a prime web Q with $\overset{Q}{\frown} \neq \varnothing$

$$\mathsf{q}{\downarrow}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G'' \parr Q^*(\!|J_1 \parr L_1, \dots, J_k \parr L_k|\!)\end{array}}{G'' \parr Q^{\perp}(\!|J_1, \dots, J_k|\!) \parr \dfrac{Q(\!|L_1, \dots, L_k|\!)}{P(\!|M_1, \dots, M_n|\!)}} \quad \text{or} \quad \mathsf{qm}\dfrac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G'' \parr Q(\!|J_1 \parr L_1, \dots, J_n \parr L_k|\!)\end{array}}{G'' \parr Q(\!|J_1, \dots, J_k|\!) \parr \dfrac{Q(\!|L_1, \dots, L_k|\!)}{P(\!|M_1, \dots, M_n|\!)}}$$

  where $J_i \parr L_i \neq \varnothing$ for all $i \in \{1, \dots, k\}$ and Q^* denotes $Q^{\perp}$ if $\overset{Q}{\frown} = \varnothing$ and Q otherwise. In this case, we conclude by induction hypothesis. Note that for the $\mathsf{q}{\downarrow}$ we do not have to take into consideration Case (C) since $\overset{Q^*}{\frown} \neq \varnothing$.

- We have the the special cases due to associativity of $\otimes$ and $\triangleleft$ concerning rules $\mathsf{s}_{\parr}$ and $\mathsf{q}{\downarrow}$.

$$\mathsf{s}_{\parr}\dfrac{G'' \parr (M_1 \otimes (G' \parr (M_2 \otimes M_3)))}{G'' \parr G' \parr \dfrac{M_1 \otimes (M_2 \otimes M_3)}{(M_1 \otimes M_2) \otimes M_3}} \quad \mathsf{q}{\downarrow}\dfrac{G'' \parr ((N_1 \parr M_1) \triangleleft (N_2 \parr (M_2 \triangleleft M_3)))}{G'' \parr (N_1 \triangleleft N_2) \parr \dfrac{M_1 \triangleleft (M_2 \triangleleft M_3)}{(M_1 \triangleleft M_2) \triangleleft M_3}} \quad \mathsf{q}{\downarrow}\dfrac{G'' \parr ((N_1 \parr (M_1 \triangleleft M_2) \triangleleft (N_2 \parr M_3))}{G'' \parr (N_1 \triangleleft N_2) \parr \dfrac{(M_1 \triangleleft M_2) \triangleleft M_3}{M_1 \triangleleft (M_2 \triangleleft M_3)}}$$

  Note that a case similar to the first one above can be produced using $\mathsf{p}{\downarrow}$ instead of $\mathsf{s}_{\parr}$ and similar cases to the ones above on the right can be produced using qm instead of $\mathsf{q}{\downarrow}$, and, whenever one between N_1 or N_2 is empty, using $\mathsf{s}_{\parr}$ instead of $\mathsf{q}{\downarrow}$.

- The last rule is a sl, that is, $\mathsf{GX} = \mathsf{GV}^{\mathsf{sl}}$, $\overset{P}{\frown} = \varnothing$ and

$$\mathsf{sl}\ \frac{\begin{array}{c}\varnothing\\ \mathcal{D}' \parallel \mathsf{GX}\\ G \parcons (P(\!|M_1,\dots,M_k,\varnothing,\dots,\varnothing|\!) \triangleleft P(\!|\varnothing,\dots,\varnothing,M_{k+1},\dots,M_n|\!))\end{array}}{G \parcons P(\!|M_1,\dots,M_n|\!)}$$

In this case, we can apply Lemma 34 and conclude by inductive hypothesis. ◀

Sketch of Proof of Splitting Lemma for Complete Webs (Lemma 34). By induction on $|V_{\mathbb{C}}|$ using Lemma 33. We know that all connectors and subconnectors of a complete graph are complete graphs, therefore webs. Therefore we can assume $\mathbb{C} = \mathbb{C}'(\!|\mathbb{C}_1,\dots,\mathbb{C}_m|\!)$ for some m and complete webs $\mathbb{C}', \mathbb{C}_1, \dots, \mathbb{C}_m$ where $\mathbb{C}'$ is also prime. We can apply Lemma 33. If Case (A) of Lemma 33 applies, then we conclude by induction hypothesis. Otherwise Case (B) applies and we can assume without of loss of generality that we have K_X and K_Y such that $\vdash_{\mathsf{GX}} K_X \parcons \mathbb{C}_1(\!|M_1,\dots,M_l|\!)$ and $\vdash_{\mathsf{GX}} K_Y \parcons \mathbb{C}''(\!|M_{l+1},\dots,M_n|\!)$ for some $1 \leq l \leq n$, where $\mathbb{C}'' = \mathbb{C}'(\!|\varnothing,\mathbb{C}_2,\dots,\mathbb{C}_m|\!)$. Since $\mathbb{C}''$ is a complete web, we can conclude by induction hypothesis. ◀

Proof of Context-Reduction (Lemma 32). If $A = \varnothing$ we conclude by letting $\mathcal{C}[\Box] = \varnothing$ and $K = \varnothing$. Otherwise, we can assume w.l.o.g. that $\mathcal{C}[A] = G \parcons \mathcal{C}'[A]$ for a web $\mathcal{C}'[A]$ which is neither a par nor empty. If $\mathcal{C}'[\varnothing] = \varnothing$, then we can let $K = G$ and the derivation $\mathcal{D}_{\mathcal{X}}$ is trivial. The derivation $\mathcal{D}_A$ is provided by assumption since $K \parcons A = G \parcons A = \mathcal{C}[A]$. If $\mathcal{C}'[\varnothing] = G'$ is non-empty, we proceed by induction on $\|\mathcal{C}'[A]\|$. Then w.l.o.g. we can assume that $\mathcal{C}[A] = G \parcons P(\!|M_1[A], M_2, \dots, M_n|\!)$ for a prime web $P \neq \parcons$ and we can apply Lemma 33. We have the following three cases:

(A) There are webs $K_1, \dots, K_n$ such that

$$\begin{array}{c}P^{\perp}(\!|K_1,\dots,K_n|\!)\\ \mathcal{D}_G \parallel \mathsf{GX}\\ G\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_1 \parallel \mathsf{GX}\\ K_1 \parcons M_1[A]\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_i \parallel \mathsf{GX}\\ K_i \parcons M_i\end{array}$$

for all $2 \leq i \leq n$.

(B) One of the two following sub-cases holds:

(I) there are webs K_X and K_Y such that

$$\begin{array}{c}K_X \parcons K_Y\\ \mathcal{D}_G \parallel \mathsf{GX}\\ G\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_X \parallel \mathsf{GX}\\ K_X \parcons M_1[A]\end{array} \quad \text{and} \quad \begin{array}{c}\varnothing\\ \mathcal{D}_Y \parallel \mathsf{GX}\\ K_Y \parcons P(\!|\varnothing, M_2, \dots, M_n|\!)\end{array}$$

(II) there are webs K_X and K_Y such that, w.l.o.g.,

$$\begin{array}{c}\varnothing\\ \mathcal{D}_X \parallel \mathsf{GX}\\ K_X \parcons M_n\end{array}, \quad \begin{array}{c}\varnothing\\ \mathcal{D}_Y \parallel \mathsf{GX}\\ K_Y \parcons P(\!|M_1[A], M_2, \dots, M_{n-1}, \varnothing|\!)\end{array} \quad \text{and} \quad \begin{array}{c}K_X \parcons K_Y\\ \mathcal{D}_G \parallel \mathsf{GX}\\ G\end{array}$$

(C) $\mathsf{GX} = \mathsf{GV}^{\mathsf{sl}}$ and one of the two following sub-cases holds:

(I) there are webs K_X and K_Y such that

$$\begin{array}{c} K_X \lhd K_Y \\ \mathcal{D}_G \parallel \mathsf{GX} \\ G \end{array}, \quad \begin{array}{c} \varnothing \\ \mathcal{D}_X \parallel \mathsf{GX} \\ K_X \parr P(\!|M_1, \ldots, M_{j-1}, M_j[A], M_{j+1}, \ldots, M_k, \varnothing, \ldots, \varnothing|\!) \end{array} \quad \text{and} \quad \begin{array}{c} \varnothing \\ \mathcal{D}_Y \parallel \mathsf{GX} \\ K_Y \parr P(\!|\varnothing, \ldots, \varnothing, M_k, \ldots, M_n|\!) \end{array}$$

(II) The other case is similar but assuming $j > k$.

In all cases we can conclude by applying the induction hypothesis. ◀

Sketch of Proof of Splitting Lemma for Atomic Webs (Lemma 31). As in the proof of Lemma 33, we proceed by case analysis on the last rule in a derivation $\mathcal{D}$ of $G \parr a$.

- If rule r acts inside G or the last rule in $\mathcal{D}$ is an $\mathsf{ai}{\downarrow}$, then the derivation $\mathcal{D}$ is of shape

$$\begin{array}{c} \varnothing \\ \mathcal{D}' \parallel \mathsf{GX} \\ \mathsf{r}\dfrac{G'}{G} \parr a \end{array} \quad \text{or} \quad \begin{array}{c} \varnothing \\ \mathcal{D}'_G \parallel \mathsf{GX} \\ G' \parr \mathsf{ai}{\downarrow}\dfrac{\varnothing}{a^\perp \parr a} \end{array}$$

  for some $\mathcal{D}'$. In the first case, we conclude by applying the induction hypothesis on $G' \parr a$ and in the second case, we let $\mathcal{D}_G = \mathcal{D}'_G \parr a^\perp$.

- If the last rule in $\mathcal{D}$ is a $\mathsf{s}_\parr$, then $\mathcal{D}$ is of one of the following shapes

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

  In both cases we conclude by applying context reduction.

- If the last rule is a $\mathsf{p}{\downarrow}$, then w.l.o.g. $\mathcal{D}$ is of the shape

$$\begin{array}{c} \varnothing \\ \mathcal{D}' \parallel \mathsf{GX} \\ G'' \parr \mathsf{p}{\downarrow}\dfrac{(M_1 \parr a) \otimes M_2 \otimes \cdots \otimes M_n}{P(\!|M_1, \ldots, M_n|\!) \parr P^\perp(\!|a, \varnothing, \ldots, \varnothing|\!)} \end{array}$$

  We conclude by applying Lemma 34 and the induction hypothesis.

- If the last rule is a $\mathsf{q}{\downarrow}$, then w.l.o.g. $\mathcal{D}$ is of the shape

$$\mathsf{q}{\downarrow}\dfrac{\begin{array}{c} \varnothing \\ \mathcal{D}' \parallel \mathsf{GX} \\ G'' \parr Q^\perp(\!|M_1 \parr a, M_2, \ldots, M_n|\!) \end{array}}{Q^\perp(\!|M_1, \ldots, M_n|\!) \parr Q(\!|a, \varnothing, \ldots, \varnothing|\!)}$$

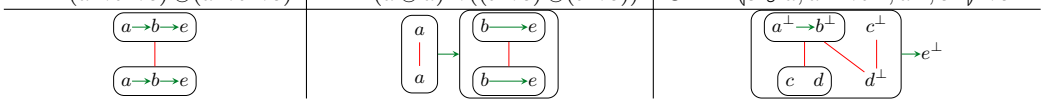
  for a complete prime web $Q^\perp$. We apply Lemma 33 and proceed as above.

- If the last rule is a $\mathsf{s}_\otimes$ or a sl then we conclude as in the first case. ◀

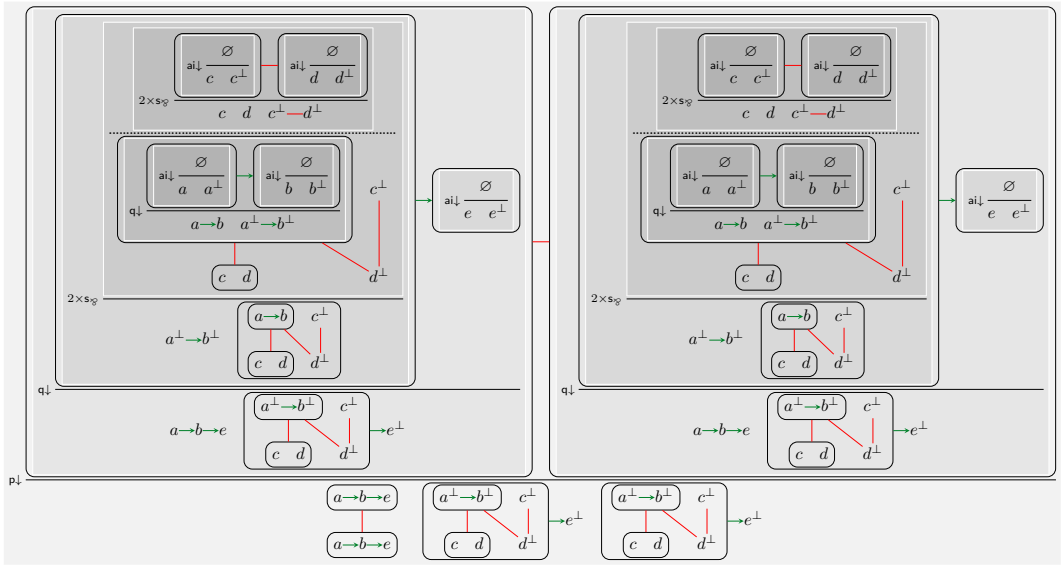
B The Need for $\mathsf{s}_{\otimes}$ in GV

▶ **Example 44** ($\mathsf{s}_{\otimes}$ is not admissible.)**.** The rule $\mathsf{q}_{\mathsf{BV}}{\uparrow}$ (which overlaps with $\mathsf{s}_{\otimes}$, as shown in Lemma 21) is admissible in BV [19]. Similarly, the rule $\mathsf{s}_{\otimes}$ is admissible in GS [1]. However, in [2] it was shown that $\mathsf{s}_{\otimes}$ is required to prove that GS is a conservative extension of the multiplicative linear logic (MLL). Furthermore, we are able to provide an example showing that $\mathsf{s}_{\otimes}$ is not admissible in any of our systems conservatively extending GS and BV. Consider the following webs.

$A = (a \triangleleft b \triangleleft e) \otimes (a \triangleleft b \triangleleft e)$ | $B = (a \otimes a) \triangleleft ((b \triangleleft e) \otimes (b \triangleleft e))$ | $C = \mathsf{N}(\!|c \parr d, a^{\perp} \triangleleft b^{\perp}, d^{\perp}, c^{\perp}|\!) \triangleleft e^{\perp}$



We have $\vdash_{\mathsf{BV}} A \multimap B$ and $\vdash_{\mathsf{GV}} A \parr C \parr C$ as shown below.



By cut elimination for GV, we get $\vdash_{\mathsf{GV}} B \parr C \parr C$. However, $\vdash_{\mathsf{GV}\setminus\{\mathsf{s}_{\otimes}\}} B \parr C \parr C$ does not hold.