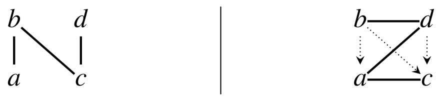
Multiplicative Logic Beyond Cographs

Matteo Acclavio

1 Introduction

In theoretical computer science, formulas are used to describe complex structures using elementary operators such as logical connectives and modalities. In particular, the proof theory of propositional logic typically considers formulas built from a very limited palette of binary (connectives) and unary (modalities) operators. Beside the restriction on the basic operators does not generally limit the expressiveness of the language, as soon as proof theory is used to define paradigms as "formulas-as-types", "formulas-as-programs", or "formulas-as-processes", this limitation leads to a payout in term of efficiency whenever we aim at providing efficient implementations: in order to describe complex interaction, ad-hoc encodings need to be put in place. As a consequence, automated tools relying on formula-based proof systems are either sub-optimal, because of the blow-up in computational complexity due to the use of encodings, or sacrifice the quality of information, by reducing their scope to only considering simpler configurations. This latter possibility may lead to information loss, potentially causing, among others, security issues or imprecise results in AI for decision systems.

For this reason, graphs are often used in computer science practice from abstract definitions to practical implementation to describe systems with complex interactions: it is often the case that "a picture is worth a thousand words". By means of example, consider a system consisting of four processes a, b, c and d racing to access shared resources, and assume that the pairs of processes a and b, b and c, and c and d share the access to a same resource. This configuration can be represented by the graph below on the left (called P_4) where vertices represent processes and an edge is drawn whenever two processes share the access to a same resource.



(1)

Similarly, we could consider a dependency relation (e.g., causality) in a system with where a depends from b, and c depends from both b and d. In this case, again, the binary relation of "non-causal dependency" can be represented by a graph with similar shape (see the graph above on the right). It is well-known that the graph P_4 cannot be represented by a formula containing only binary connectives and with a one-to-one correspondence between atoms and vertices of the graph [15, 20]. At the same time, the patter P_4 occurs in any graph representing a non series-parallel relation, which are ubiquitous in distributed systems (see, e.g., non-transitive conflict of interest relations in control access models such as [7], in dependency graphs, or in producer-consumer queues).

It is worthy notice that the use of graph-based syntaxes are used in logic and proof theory exactly for their expressiveness: a same object may admit multiple representations, but graphs allows us to provide more canonical ones. By means of example, graphs are largely used in defining semantics (see, e.g., Kripke semantics for modal logics [6]), in proof systems capturing those semantical structures (see, e.g., nested sequents [22, 8, 25]), and in proof systems capturing proof equivalence (e.g., proof nets [16] or combinatorial proofs [19, 19]).

However, proof theory has rarely considered graphs as primitive terms to reason on: prior to [3, 4, 2] we cannot find proof systems conceived to handle graphs as terms of an inference system defined with proof-theoretical purposes.[1] In these works, the authors move from the well-known correspondence between classical propositional formulas and cographs (graphs containing no induced subgraph isomorphic to a P_4) [20] to generalize proof theoretical methodologies for inference systems on formulas to graphs. In fact, we could say that inference systems operating on formulas can be seen as inference systems operating on cographs, that is, on graphs with "less complex" structure where no induced subgraph isomorphic to P_4 occurs[2]. In these works, the authors consider only *deep inference* [18, 5] formalism to design proof systems

[1] Another line of works [9, 29, 10, 12, 13] explored the extensions of the semantics of boolean logic from cographs enconding formulas to graphs. These works are motivated by the study of valid linear inferences extending the semantics of boolean logic, and the proof theory for this logic has only been developed after its semantics.

[2] Several NP-hard optimization problems on graphs become solvable in polynomial time if restricted to cographs [21].

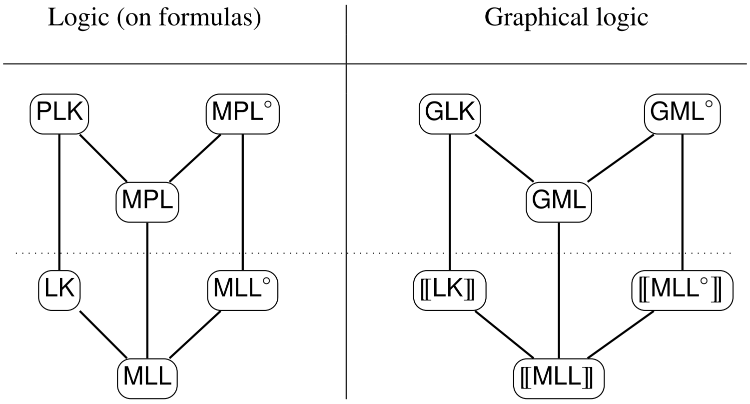


Figure 1: The lattice of the logics on formulas we discuss in this talk and the lattice of graphical logics defined by the interpretation of graphical connectives as prime graphs. The logics below the dotted line contain formulas where only the binary connectives for conjunction and disjunction occurs; similarly, the graphical logics below the dotted line are families of cographs.

operating on graphs. Such unconventional choice with respect to, e.g., sequent calculi or natural deduction, pays off in [2], where a proof system operating on graphs with both symmetric and non-symmetric edges defines a conservative extension of the non-commutative logic BV[3], for which a cut-free sequent calculus cannot exist [28].

Main contributions

In this talk we discuss the results in [1] including:

- how to extend the methodologies used in proof theory to reason on formulas, to reason on graphs;
- sequent calculi operating graphs;
- the graphical logic GS from [3, 4] admits a sound and complete sequent system;
- a syntax for proof nets for the substructural logics operating on graphs we introduce.

For this purpose, we introduce the notion of *graphical connectives* to define formulas whose purpose is to represent graphs via the *graph modular decompositions*, that is, abstract syntax trees describing graphs with a term of linear size with respect to the number of vertices of the graph. This provides foundation to the methodologies used in [3, 4, 10] to design proof systems operating on graphs by handling their modular decomposition trees.

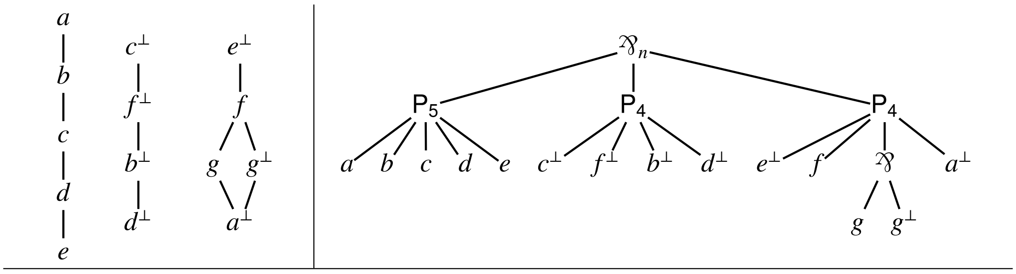
We then introduce proof systems in which graphical connectives, are used as *generalized multiplicative connectives* (in the sense of [11, 17]). We recall the basic proof theoretical properties of these systems, such as cut-elimination, initial coherence and a weaker notion of the analyticity condition taking into account the richer structure of non-binary connectives. Then, relying on these results, we prove that these logics are conservative extensions of multiplicative linear logic with and without mix and that the latter logic capture graph isomorphism as a logical equivalence between formulas.

We prove that the sequent system MPL° is sound and complete with respect to the set of graphs in the graphical logic GS from [3, 4]. This result indirectly provides a proof of of existence of a sequent system for the logic GS, as well as a proof of analyticity and transitiveness of implication for this GS relying on more standard techniques[4].

Then in the second part of the talk we discuss a formalism for proof nets for the MPL and MPL° extending Retoré's syntax of RB-proof nets [26]. For this purpose, we extend this syntax with new gate types

[3] The logic BV is a NP-time decidable fragment of Pomset logic [24, 23]. This logic is sound and complete with respect to series-parallel order refinements: if ϕ and ψ are formulas encoding series-parallel orders, then the order encoded by ϕ is a refinement of the order encoded by ψ iff $\vdash_{\mathsf{BV}} \phi \multimap \psi$.

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



$$
\text{wd}_\otimes\,\dfrac{
 \parr\,\dfrac{
  \text{wd}_\otimes\,\dfrac{
   \text{ax}\,\dfrac{}{\vdash a, a^\perp}
   \quad
   \text{d-}\kappa_{P_4}\,\dfrac{
    \text{ax}\,\dfrac{}{\vdash b, b^\perp}
    \quad
    \text{ax}\,\dfrac{}{\vdash c, c^\perp}
    \quad
    \text{ax}\,\dfrac{}{\vdash d, d^\perp}
    \quad
    \otimes\,\dfrac{\text{ax}\,\dfrac{}{\vdash e, e^\perp} \quad \text{ax}\,\dfrac{}{\vdash f, f^\perp}}{\vdash e, e^\perp \otimes f, f^\perp}
   }{\vdash \kappa_{P_4}\llparenthesis b, c, d, e \rrparenthesis, \kappa_{P_4}\llparenthesis c^\perp, f^\perp, b^\perp, d^\perp \rrparenthesis, e^\perp \otimes f}
  }{\vdash \kappa_{P_5}\llparenthesis a, b, c, d, e \rrparenthesis, \kappa_{P_4}\llparenthesis c^\perp, f^\perp, b^\perp, d^\perp \rrparenthesis, e^\perp \otimes f, a^\perp}
 }{\vdash \kappa_{P_5}\llparenthesis a, b, c, d, e \rrparenthesis, \kappa_{P_4}\llparenthesis c^\perp, f^\perp, b^\perp, d^\perp \rrparenthesis, (e^\perp \otimes f) \parr a^\perp}
 \quad
 \parr\,\dfrac{\text{ax}\,\dfrac{}{\vdash g, g^\perp}}{\vdash g \parr g^\perp}
}{\vdash \kappa_{P_5}\llparenthesis a, b, c, d, e \rrparenthesis, \kappa_{P_4}\llparenthesis c^\perp, f^\perp, b^\perp, d^\perp \rrparenthesis, \kappa_{P_4}\llparenthesis e^\perp \otimes f, g \parr g^\perp, a^\perp \rrparenthesis}
$$

Figure 2: A graph, the abstract syntax tree of its modular decomposition, and a derivation in MPL° of the formula encoding it.

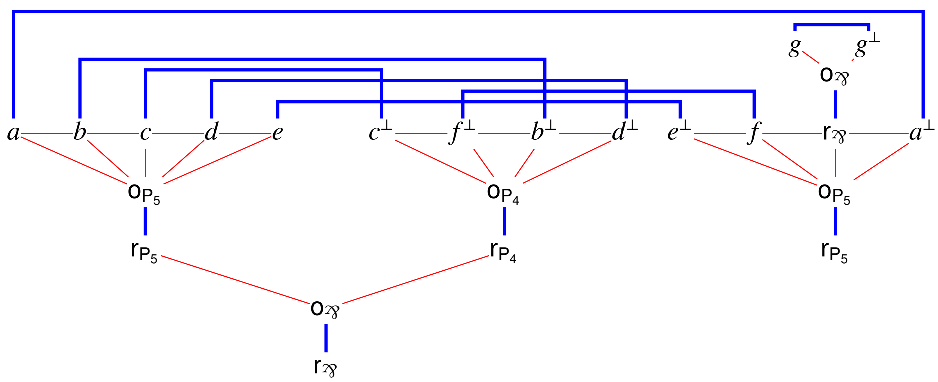


Figure 3: The RB-proof net encoding the derivation in Figure 2.

generalizing the gates for the binary connectives $\parr$ and $\otimes$ to the graphical connectives previously introduced. For this proof nets formalism, we provide correctness criteria to characterize both logics MPL and MPL° together with a sequentialization procedure obtained by refining Retoré'e criterion and considering additional information, which can be derived by the topology of the graph, which reminds *sequential edges* in *C-nets* [14].

References

[1] Matteo Acclavio. Graphical Proof Theory I: Multiplicative logic beyond cographs. Preprint, 2023.

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

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

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

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

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

[7] David FC Brewer and Michael J Nash. The chinese wall security policy. In *IEEE symposium on security and privacy*, volume 1989, page 206. Oakland, 1989.

[8] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009.

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

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

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

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

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

[14] Paolo Di Giamberardino and Claudia Faggian. Proof nets sequentialisation in multiplicative linear logic. *Annals of Pure and Applied Logic*, 155(3):173–182, 2008.

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

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

[17] 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.

[18] 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.

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

[20] 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.

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

[22] Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–136, 1994.

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

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

[25] Francesca Poggiolesi. The method of tree-hypersequents for modal propositional logic. In D. Makinson, J. Malinowski, and H. Wansing, editors, *Towards Mathematical Philosophy*, volume 28 of *Trends in Logic*, pages 31–51. Springer, 2009.

[26] Christian Retoré. Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs. Rapport de recherche 3652, INRIA, 1999. Appeared as [27].

[27] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. *Theoretical Computer Science*, 294(3):473–488, 2003.

[28] 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.

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