1 Introduction

The use of proof-graphs, instead of trees or lists, for representing proofs is getting popular among proof-theoreticians. Proof-graphs serve as a way to provide a better symmetry to the semantics of proofs (Oliveira & Queiroz 2003) and a way to study complexity of propositional proofs and to provide more efficient theorem provers, concerning size of propositional proofs. In (Bonet & Buss 1993), one can find a complexity analysis of the size of Frege systems, Natural Deduction systems and Sequent Calculus concerning their tree-like and list-like representation. This leads to $O(nlog(n))$ improvement in the size of the list-based proofs compared to tree-like proofs, which is based on the observation that the hypotheses occur only once in the lists and more than once in the trees. Thus sharing formulas helps to reduce the size of proofs. There are related works, e.g. (Alves, Fernández & Mackie 2011), that use graphs for representing proofs, pointing out that proof-graphs offer a better way to facilitate the visualisation and understanding of proofs in the underlying logic.

On the other hand (Finger 2005), (da Costa 2007) and (Gordeev, Haeusler & Costa 2009) show that the use of Directed Acyclic Graphs (DAGs) together with mechanisms of unification/substitution in proof representations has compacting/compressing factor equivalent to cut-introduction. And, obviously, graphs can save space by means of reference, instead of plain copying. This work shows yet another advantage of using graphs for representing proofs. First, we show that using “mixed” graph representations of formulas and inferences in Natural Deduction in the purely implicational minimal logic one can obtain a (weak) normalization theorem that, in fact, is a strong normalization theorem. Moreover the corresponding normalization procedure does not exceed the size of the input, which sharply contrasts to the well-known exponential speed-up of standard normalization. The choice of purely implicational minimal logic ($M^{\rightarrow}$) is motivated by the fact that the computational complexity of the validity of $M^{\rightarrow}$ is PSPACE-complete and can polynomially simulate classical, intuitionistic and full minimal logic (Statman 1979) as well as any propositional logic with a Natural Deduction system satisfying the subformula property (Haeusler 2013). Then we extend this result to propositional logic obtaining also strong normalization.

In a more general context, this work has been conducted as part of a bigger tree-to-graph proof compressing research project. The purpose of such proof compression is:

1. To construct small (if possible, minimal) graph-like representations of standard tree-like proofs in a given proof system and – in the propositional case – investigate the corresponding short graph-like theorem provers.

2. To find short (say, polynomial-size) graph-like analogous of standard tree-like proof theoretic operations like e.g. normalization in Natural Deduction and/or cut-elimination in Sequent Calculus.

Note that the present work fulfills both conditions with regard to the mimp-graph representation (see below) of chosen Natural Deduction and the corresponding notion of formula-minimality (see Theorems 3, **??**, 4 and 5).

Back to the proof normalization, recall the following properties of a given structural deductive system (Natural Deduction, Sequent Calculus, etc):

– Normal form: To each derivation of α from Δ there is a normal derivation of α from $\Delta' \subseteq \Delta$.

– Normalization: To each derivation of α from Δ there is a normal derivation of α from $\Delta' \subseteq \Delta$, obtained by a particular strategy of reductions application.

– Strong Normalization: To each derivation of α from Δ there is a normal derivation of α from $\Delta' \subseteq \Delta$. This normal form can be obtained by applying reductions to the original derivation in any ordering.

The strong normalization property for a natural deduction system is usually proved by the so-called semantical method:

– Define a property $P(\pi)$ on derivations π in the Natural Deduction system;

– Prove that this property implies strong normalization, that is $\forall\pi(P(\pi) \rightarrow SN(\pi))$, where $SN(X)$ means that X is strongly normalizable;

– Prove that $\forall\pi P(\pi)$.

There are well-known examples of this property $P(X)$: (1) Prawitz's "strong validity"; (2) Tait's "convertibility"; (3) Jervell's "regularity"; (4)

Leivant's "stability"; (5) Martin-Löf's "computability"; (6) Girard's "candidate de reducibilité". Note that such semantical method to prove strong normalization is unconstructive and even in the case of purely implicational fragment of minimal logic it provides no combinatorial insight into the nature of strong normalization. Another, more constructive strategy would be to show that there is a worst sequence of reductions always produces a normal derivation. Let us call it a syntactic method of proving the strong normalization theorem. The method used in the present research is that any sequence of reductions always produces a normal derivation. This means that the order in which cuts (maximal formulas) are eliminated has no impact on the end-result. This is obtained by brute force: the proof consists of an exhaustive case-analysis.

Other methods use assignments of rather complicated measures to derivations such that arbitrary reductions decrease the measure, which by standard inductive arguments yields a desired proof of the strong normalization. In this thesis we show how to represent derivations in a graph-like form to $M^{\rightarrow}$ and full propositional logic, and how to reduce (eliminating maximal formulas) these representations such that a normalization theorem can be proved by counting the number of maximal formulas in the original derivation. The strong normalization is a direct consequence of such normalization, since any reduction decreases the corresponding measure of derivation complexity. The underlying intuition comes from the fact that our graph representations use only one node for any two identical formulas occurring in the original Natural Deduction derivation (see Theorem 3 for a more precise description).

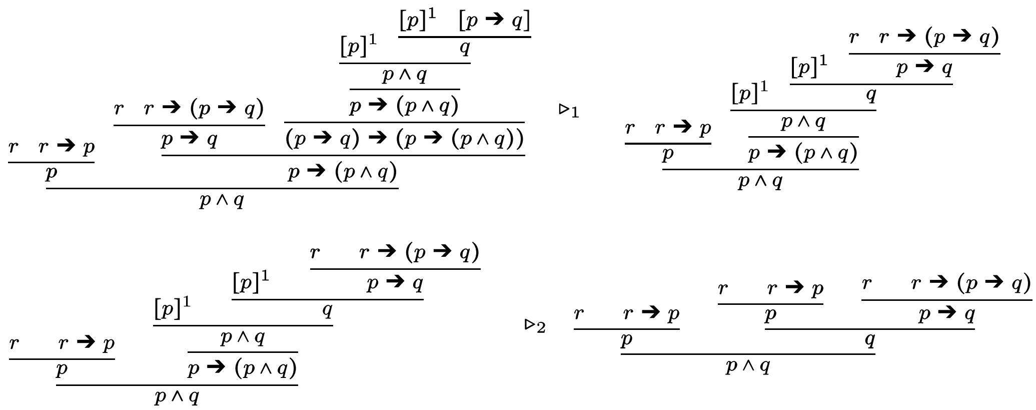

Figure 1.1: Example of derivation with two steps of reduction.

We show in Figure 1.1 an example of the eliminating a maximal formula in a derivation in Natural Deduction. The formula $p \rightarrow (p \wedge q)$ is not a maximal formula before a reduction ($\triangleright_1$) is applied to eliminate the maximal formula $(p \rightarrow q) \rightarrow (p \rightarrow (p \wedge q))$. This possibility of having hidden maximal formulas in

ND is the main reason to use more sophisticated methods whenever proving strong normalization.

In Figure 1.2 we show an embedding of this derivation into a mimp-graph. This example shows the reason why our normalization procedure is directly a strong normalization. We remark that there is no possibility to hide a maximal formula because all formulas are represented only once in the graph (see Figure 1.2). In this graph $p \to (p \wedge q)$ is already a maximal formula. We can choose to remove any of the maximal formulas. If $p \to (p \wedge q)$ is chosen to be eliminated, by the mimp-graph elimination procedure, its reduction eliminates the $(p \to q) \to (p \to (p \wedge q))$ too. On the other hand, the choice of $(p \to q) \to (p \to (p \wedge q))$ to be reduced only eliminates itself. In any case the number of maximal formulas decreases and the mimp-graph becomes as shown in Figure 1.3.

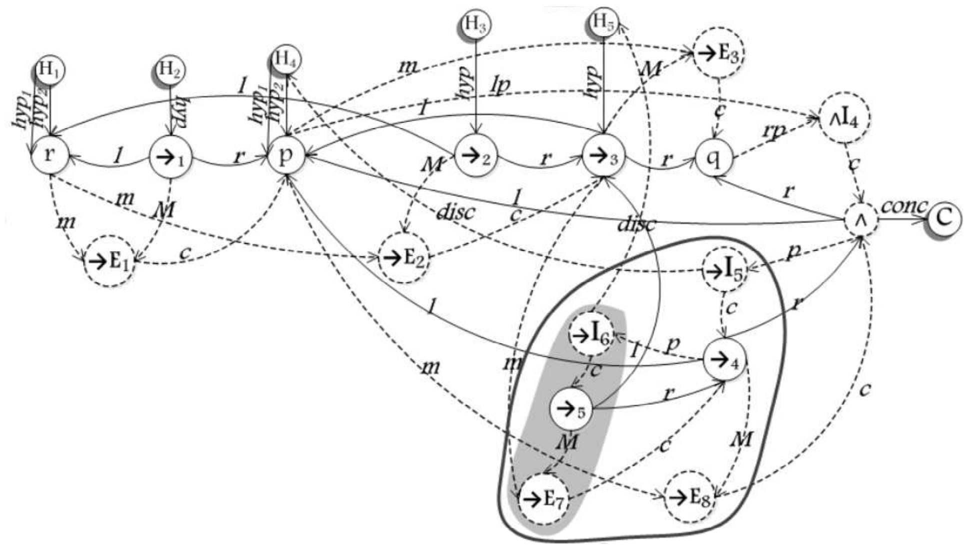

Figure 1.2: Mimp-graph translation of derivation in Figure 1.1.

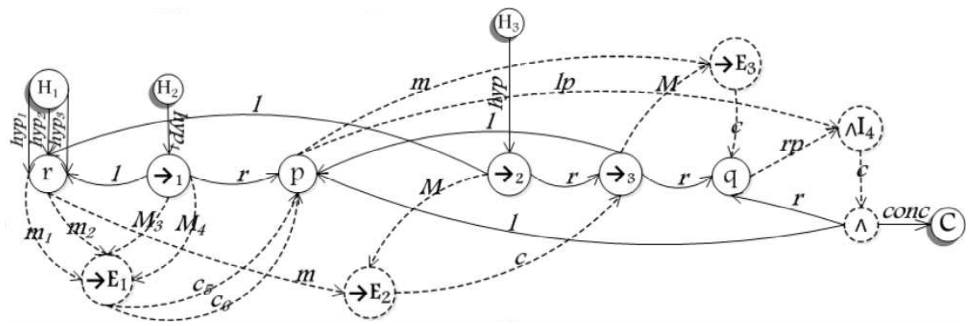

Figure 1.3: Normalized mimp-graph of the example in Figure 1.2.

Continuing with our aim of studying the complexity of proofs, the current approach also give graph representations for first order logic, deep inference and bi-intuitionistic logic.

1.0.1
Related work

The related idea of proof-graphs has been investigated in the last decade, for classical and intuitionistic logic. More recently, N-graphs introduced by de Oliveira (Oliveira & Queiroz 2003) is a proof system originally developed for classical logic, as a suitable solution to the lack of symmetry in classical ND logic. Mainly because N-Graphs use a multiple conclusion proof structure. N-Graphs have also been adapted for intuitionistic logic (Quispe-Cruz, de Oliveira, de Queiroz & de Paiva 2014).

In (Geuvers & Loeb 2007) another approach to represent Natural Deduction using graphs is proposed. It reports a graph-representation of Natural Deduction, in Gentzen as well as Fitch's style. In fact the proofs are represented as hypergraphs, or boxed-graphs, with possibility of sharing subproofs. It is developed not only for the implicational fragment, although the representation of linear logic proofs is related as further work. Our approach is different from (Geuvers & Loeb 2007) in that we include graph-representations of formulas in the proofs. The fact that our normalization procedure leads to strong-normalization is a consequence of sharing subformulas, and hence subproofs, in our proof-graph representations. It is unclear whether a similar result is available using (Geuvers & Loeb 2007).

Other previous research concerning the use of graphs to represent proofs was developed on connections to substructural logics as Linear Logic, see (Girard 1996) and (Girard, Lafont & Regnier 1995) for example. The main motivation of this just mentioned investigations is to provide a sound way of representing Linear Logic proofs without dealing with unique labeling and complicated rules for relabeling and discharging mechanisms need to represent Linear Logic proofs as trees in Natural Deduction style as well as in Sequent Calculus.

Proof-nets were such representations and a syntactical criteria on the possible paths on them were considered as a soundness criteria for a proof-graph to be a proof-net. Proof-nets have a cut-rule quite similar to the cut in Sequent Calculus. For the Multiplicative fragment of Classical Linear Logic, there is a linear time cut-elimination theorem. However, when the additive versions of the connectives are considered, the usual complexity of the cut-elimination raises up again. Linear Logic is an important Logic whenever we consider the study of a concurrent computational system and semantics of it strongly uses concurrency theory concepts. Our investigation, on the other hand, is not motivated by proof-theoretical semantics [1]. From the purely proof-

[1] The name that nowadays is used to denote the kind of research pioneered by Jean-Yves

theoretical point of view, we use graphs to reduce the redundancy in proofs in such a way that we do not allow hidden maximal formulas in our graph representation of a Natural Deduction proof.

Finally, we mention Alessio-Gundersen (Guglielmi & Gundersen 2008) work, where a kind of flow graph is used to present an abstract graphical framework capable of representing the calculus of structures in deep inference and is also presented a normalization mechanism via an abstract graphical framework for SKS system.

1.1 Organization

Chapter 2 gives the basic notions of natural deduction that is considered in the thesis.

Chapter 3 introduces our proof-graphs for minimal implicational propositional logic.

Chapter 4 proposes proof-graphs with explicit sharing of sub-proofs by means of boxes to border the set of shared rules.

Chapter 5 adds the full minimal propositional logic to the mimp-graph formalism (propositional mimp-graph) with its normalization procedure and then adds a version for first order logic and the set of transformations that we need to describe the normalization.

Chapter 6 starts with a brief overview of deep inference and the *calculus of structures* by Guglielmi (Guglielmi 2007) and then presents a proof-graph representation for this calculus; and Section 6.2 introduces the Bi-intuitionistic Logic and then shows how the formalism N2Int can be embedded in proof-graphs.