7 Conclusions and future works

Our proof-graph representation, which we call mimp-graph, defined in Chapter 3, was introduced through definitions and examples, mainly devised for extracting proof-theoretic properties from proof system.

That is, we have tackled one of our research tracks. Mimp-graph preserves the ability to represent proofs in Natural Deduction and its minimal formula representation is a key feature of the mimp-graph structure, because as we saw earlier, it is easy to determine maximal formulas and upper bounds on the length of reduction sequences leading to normal proofs.

Thus 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 measures of derivation complexity.

The results presented for mimp-graph are naturally extended for propositional mimp-graphs in Chapter 5. We also get strong normalization for normalization process that is proved by counting the number of maximal formulas in the original derivation.

Other merit of the present thesis is the treatment of sharing for inference rules, in addition to formula sharing, developed in Chapter 4, representing graphs more compactly, which is performed during the construction process of the graph. This feature is very important, since we intend to use this mimp-graph approach in automatic theorem provers.

In the construction process when similar formulas are found, the process of producing the unifier would consume linear time. We do not implement a process of searching for similar formulas in our graph but we estimate that the resources consumed in such searching would be compensated by the reduction of necessary resources to build the proof-graph.

Other contribution is the representation in graphs for first order logic. With our approach, we get a better view on the behaviour of variables inside a proof: variable binding in both quantifiers and inferences.

Our experimentation with other logics as deep inference has resulted in a representation for Calculus of Structures and that it preserves the symmetry of Calculus of Structures: (i) all rules have one premise and one conclusion (vertical symmetry), (ii) there are dual rules, e.g. the identity rule and cut rule, weakening and co-weakening, (iii) the constant node f is symmetrical with t. Our proof-graph representation also has the ability to access subformulas, allowing the inference rules to be applied in any place deep inside a formula graph, just like deep inference. So, we get for our graph the same good characteristics of deep inference. We intend, as future work, to propose a normalization procedure for deep-graphs, where we will use the technique of reducing cuts similar to what is done in normalization for propositional mimp-graphs.

The application of a mimp-style representation to Bi-intuitionistic logic (2Int) aims to verify that using this graph representation results in a reduced size of the proofs with respect to traditional ways of presentation in Natural Deduction (N2Int). It also allows a better understanding of the proving process, due to the intuitive graphical interpretation the graphs provide. For example, the use of the delimiter node hypothesis (assumptions and counter-assumptions) has also proven useful. In particular, it has made duality identification manageable and more elegant, in such way semantic properties of logical connectives are determined by the rules nodes.

Finally, we left the details of an efficient implementation of these graphs for future work. This is a preliminary step into investigating how a theorem prover based on graphs can be more efficient than usual theorem provers.