Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 323 (2016) 181–196

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On Strong Normalization in Proof-Graphs for Propositional Logic

Marcela Quispe-Cruz[1]

Universidad Católica San Pablo (Perú)

Edward Haeusler[2]

Pontificia Universidade de Rio de Janeiro (Brazil)

Lew Gordeev[3]

Tuebingen University, Ghent University

Abstract

Traditional proof theory of Propositional Logic deals with proofs whose size can be huge. Proof theoretical studies discovered exponential gaps between normal or cut free proofs and their respective non-normal proofs. 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 study complexity of propositional proofs and to provide more efficient theorem provers, concerning size of propositional proofs.
Fpl-graphs were initially developed for minimal implicational logic representing proofs through references rather than copy. Thus, formulas and sub-deductions preserved in the graph structure, can be shared deleting unnecessary sub-deductions resulting in the reduced proof. In this work, we consider full minimal propositional logic and show how to reduce (eliminating maximal formulas) these representations such that strong normalization theorem can be proved by simply counting the number of maximal formulas in the original derivation. In proof-graphs, the main reason for obtaining the strong normalization property using such simple complexity measure is a direct consequence of the fact that each formula occurs only once in the proof-graph and the case of the hidden maximum formula that usually occurs in the tree-form derivation is already represented in the fpl-graph.

Keywords: Proof Theory, Proof Graphs, N-Graphs, Intuitionistic Logic, Sequent Calculus, Multiple-Conclusion Systems.

1 Introduction

Recently the use of graphs instead of trees to represent proofs has been shown to be more efficient[2][1], while also being helpful to better address the lack of symmetry

[1] Email: mquispec@ucsp.edu.pe

[2] Email: hermann@inf.puc-rio.br

[3] Email: lew.gordeew@uni-tuebingen.de

http://dx.doi.org/10.1016/j.entcs.2016.06.012

1571-0661/

in classical ND logic [4] and the complexity of the proof normalization process. Previously we have already presented mimp-graphs as a new proof system developed for minimal implicational logic [6], whose deductions are structured as proof-graph. The point is that in mimp-graphs it is easy to determine maximal formulas [4] and upper bounds on the length of reduction sequences leading to normal proofs. Thus a normalization theorem is proved by counting the number of maximal formulas in the original derivation. The strong normalization property is a direct consequence of such normalization, since any reduction decreases the corresponding measure of derivation complexity. In the present paper we wish to explain this procedure more clearly and expand it onto full propositional logic.

Mimp-graphs are directed graphs whose nodes and edges are labelled. Moreover we distinguish two parts, one representing the inferences of a proof, and the other the formulas. For the formula-part of a mimp-graph, we use directed acyclic graphs, that we denominated formula graphs, consist of basis in the mimp-graph construction and contain only formula nodes sharing formula nodes, thus each formula node only need to occur once in the graph, an example is shown in the left-hand side of Figure 1: the propositions P and Q occur once in the graph.

For the inference-part of a mimp-graph we have the rule nodes (R-nodes) that are labelled by the names of the inference rules. The logic connectives and inference names may be indexed, in order to achieve a 1-1 correspondence between formulas (inferences) and their representations (names), an shown in the right-hand side of Figure 1: the R-node $\rightarrow E_1$ has as major premise the formula graph $(P\rightarrow Q)\rightarrow(P\rightarrow Q)$ and as minor premise the formula graph $P\rightarrow Q$.

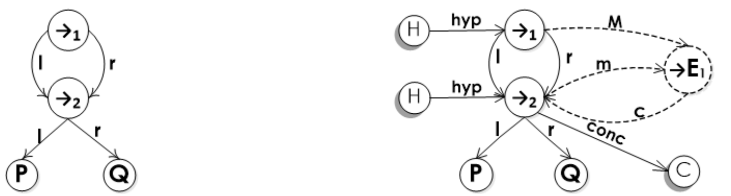


Fig. 1. Formula $(P \rightarrow Q) \rightarrow (P \rightarrow Q)$ depicted as a formula graph (left-hand side) and as major premise of the R-node $\rightarrow E_1$ (right-hand side).

Any strong normalization (SN) proof has to take care of every possible detour (maximal formula) that appears in a derivation. If one considers the normalization as a dynamic process, not all detours that are eliminated in a derivation are explicitly present since the beginning of the process. For example, the permutation-conversions used by Prawitz in the (weak) normalization of intuitionistic logic were designed to take care of hidden maximal formulas (see discussion below). Hiding a detour is a feature of elimination rules similar to $\vee$. As far as we know, Natural Deduction systems with rules similar to $\vee$-elimination use permutation-conversions to prove (weak) normalization. SN should deal with these permutation-conversions as well as systems that admit it. Another instance of hidden maximal formula is when after a conversion (reduction), new maximal formulas can appear. This already

[4] A *maximal formula* is a formula occurrence that is consequence of a introduction rule and the major premise of a elimimination rule.

happens in the case of the $\rightarrow$-reduction. In the derivation below, after eliminating the maximal formula $A \rightarrow B$, every discharged occurrence of A in Π_2 that is the major premise of a rule elimination will be a maximal formula. Since the number of such occurrences is unbounded, the number of maximal formulas in the original derivation is not a good upper-bound for the number of reduction applications. This is a key point when discussing SN proofs.

$$\dfrac{\dfrac{\begin{matrix}\Pi_1\\ A'\end{matrix}}{A}\,r\text{-intro} \qquad \dfrac{\begin{matrix}[A]\\ \Pi_2\\ B\end{matrix}}{A \rightarrow B}}{B}$$

The permutation-conversions attack another situation, as seen in the permutation-conversion below. $C \rightarrow D$ is a hidden maximal formula that becomes a maximal formula after the permutation-conversion (right-hand side).

$$\dfrac{\begin{matrix}\Pi\\ C\end{matrix} \quad \dfrac{A\vee B \quad \dfrac{\begin{matrix}[A,C]\\ \Pi_1\\ D\end{matrix}}{C\rightarrow D} \quad \dfrac{\begin{matrix}[B,C]\\ \Pi_2\\ D\end{matrix}}{C\rightarrow D}}{C\rightarrow D}}{D} \quad \triangleright \quad \dfrac{A\vee B \quad \dfrac{\begin{matrix}\Pi\\ C\end{matrix}\quad \dfrac{\begin{matrix}[A,C]\\ \Pi_1\\ D\end{matrix}}{C\rightarrow D}}{D} \quad \dfrac{\begin{matrix}\Pi\\ C\end{matrix}\quad \dfrac{\begin{matrix}[B,C]\\ \Pi_2\\ D\end{matrix}}{C\rightarrow D}}{D}}{D}$$

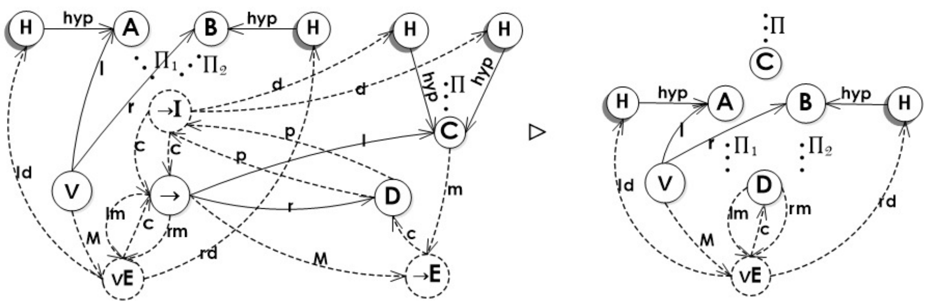


Fig. 2. Fpl-graph of the derivation above and its reduction.

In the case of our graphs, since every formula labels one and only one vertex of the graph, all the situations discussed above are explicit. That is, every possible maximal formula that can appear in the normalization proof is already there, and there is no hidden maximal formulas, as shown in Figure 2, where $\rightarrow_q$ is twice conclusion of the R-node $\rightarrow I_i$, thus it is representing already two maximal formulas. The reduction, shown on the right-hand side of figure, decreases the number of maximal formulas. Consequently, we can prove SN by induction on the number of maximal formulas.

2 Proof-graphs for full propositional logic

In [6] we considered implication as the only logic connective. Let us now turn to a more general presentation of proof-graphs for propositional calculus that includes implication, conjunction and disjunction that we called proof-graph for full propositional logic (fpl-graph). First, we define sets of labels to nodes and edges of the graph, along with a partial ordering on its R-nodes that allows to pass through the nodes of the structure. We will also develop the normalization procedure for these proof-graphs.

We want to emphasize that the fpl-graphs put together information on formula graphs and R-nodes. To make it more transparent we can use different types of lines. In this way F-nodes and edges between them are used solid lines, whereas R-nodes and edges between them and adjacent premises and/or conclusions are used dashed lines and additionally delimiter nodes have been shaded.

Definition 2.1 [Label types] There are five types of labels:

- *R-Labels* is the set of labels for *rule nodes*: $\{\to E_{m\in\mathbb{N}}, \to I_{n\in\mathbb{N}}, \wedge I_{o\in\mathbb{N}}, \wedge E_{p\in\mathbb{N}}, \vee I_{q\in\mathbb{N}}, \vee E_{r\in\mathbb{N}}\}$,
- *F-Labels* is the set of labels for *formula nodes*: $\{\to_{i\in\mathbb{N}}, \wedge_{j\in\mathbb{N}}, \vee_{k\in\mathbb{N}}\}$ and the propositional letters $\{P, Q, R, ...\}$,
- *D-Labels* is the set of labels for *delimiter nodes*: $\{H_{k\in\mathbb{N}}, C\}$.
- E_F*-Labels* is the set of labels for *formula edges*: $\{l$ (left), r (right)$\}$,
- E_M*-Labels* is the set of labels for *rule edges*: $\{$ $p_{i\in\mathbb{N}}$ (premise), $rp_{j\in\mathbb{N}}$ (right premise), $lp_{k\in\mathbb{N}}$ (left premise), $rm_{l\in\mathbb{N}}$ (right minor premise), $lm_{m\in\mathbb{N}}$ (left minor premise), $m_{n\in\mathbb{N}}$ (minor premise), $M_{o\in\mathbb{N}}$ (major premise), $c_{p\in\mathbb{N}}$ (conclusion), $d_{q\in\mathbb{N}}$ (discharge), $ld_{r\in\mathbb{N}}$ (left discharge), $rd_{s\in\mathbb{N}}$ (right discharge), $hyp_{t\in\mathbb{N}}$ (hypothesis), $conc$ (final conclusion)$\}$,

The union of these five sets of label types will be called LBL. We will use the terms α_m, β_n and γ_r to represent the principal connective of the formula α, β and γ respectively.

Definition 2.2 A proof-graph for full propositional logic (fpl-graph) G is a directed graph $\langle V, E, L\rangle$ where: V is a set of nodes, L is a subset of LBL, E is a set of labelled *edges* $\langle v \in V, t \in E_{F\cup M}$-Labels, $v' \in V\rangle$, of source v, of target v' and label t and identified with the arrow $v \xrightarrow{t} v'$.

Fpl-graphs are recursively defined as follows:

Basis If G_1 is a formula graph with root node α_m then the graph G_2 defined as G_1 with delimiter nodes H_n and C and edges $\alpha_m \xrightarrow{conc} C$ and $H_n \xrightarrow{hyp} \alpha_m$ is a fpl-graph.

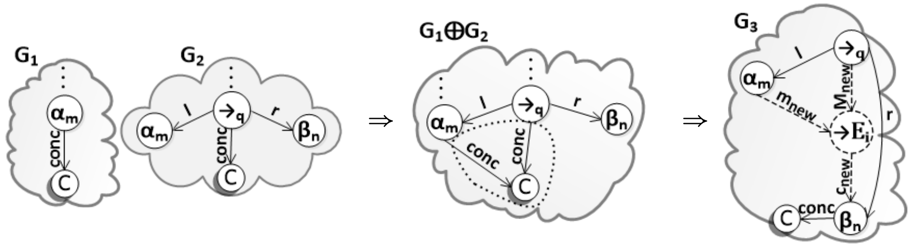
→E If G_1 and G_2 are fpl-graphs, and the graph (intermediate step) obtained by $G_1 \oplus G_2$ contains the edge $\to_q \xrightarrow{l} \alpha_m$ and two nodes $\to_q$ and α_m linked to the delimiter node C, then the graph G_3 that is defined as $G_1 \oplus G_2$ with

(i) the removal of ingoing edges in the node C which were generated in the intermediate step (see the figure below, dotted area in $G_1 \oplus G_2$);

(ii) an R-node $\to E_i$ at the top position;

(iii) the edges: $\alpha_m \xrightarrow{m_{new}} \to E_i$, $\to_q \xrightarrow{M_{new}} \to E_i$, $\to E_i \xrightarrow{c_{new}} \beta_n$ and $\beta_n \xrightarrow{conc} C$, where new is a fresh (new) index ranging over all edges of kind c, m and M ingoing and/or outgoing of the formula-nodes α_m, β_n and $\to_q$;

is a fpl-graph (see figure below).



→I If G_1 is a fpl-graph and contains a node β_n linked to the delimiter node C and the node α_m linked to the delimiter node H_k, then the graph G that is defined as

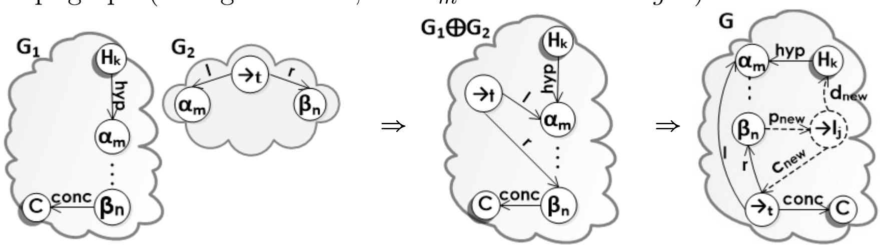
(i) $G := G_1 \oplus G_2$, such that G_2 is a formula graph with root node $\to_t$ linked to F-nodes α_m and β_n by the edges: $\to_t \xrightarrow{l} \alpha_m$, $\to_t \xrightarrow{r} \beta_n$;

(ii) with the removal of the edges: $\beta_n \xrightarrow{conc}$C;

(iii) an R-node $\to I_j$ at the top position;

(iv) the edges: $\beta_n \xrightarrow{p_{new}} \to I_j$, $\to I_j \xrightarrow{c_{new}} \to_t$, $\to_t \xrightarrow{conc} C$ and $\to I_j \xrightarrow{d_{new}} H_k$, where *new* is a fresh (new) index considering all edges of kind p, d and c ingoing and/or outgoing of the formula-nodes α_m, β_n and $\to_q$;

is a fpl-graph (see figure below; the α_m-node is *discharged*).



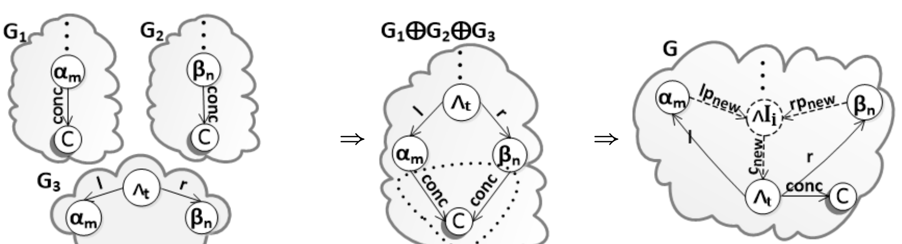
∧I If G_1 and G_2 are propositional fpl-graphs and G_1 contains α_m linked to the D-node C and G_2 contains β_n linked to the D-node C, then the graph G that is defined as

(i) $G := G_1 \oplus G_2 \oplus G_3$ with the removal of the ingoing edges in the node C which were generated in the intermediate step (see figure below, dotted area in $G_1 \oplus G_2 \oplus G_3$);

(ii) an R-node $\wedge I_i$ at the top position;

(iii) the edges: $\alpha_m \xrightarrow{lp_{new}} \wedge I_i$, $\beta_n \xrightarrow{rp_{new}} \wedge I_i$, $\wedge I_i \xrightarrow{c_{new}} \wedge_t$ and $\wedge_t \xrightarrow{conc} C$,

is a fpl-graph, see figure below.



∨E If G_1, G_2 and G_3 are propositional fpl-graphs, and the graph obtained by

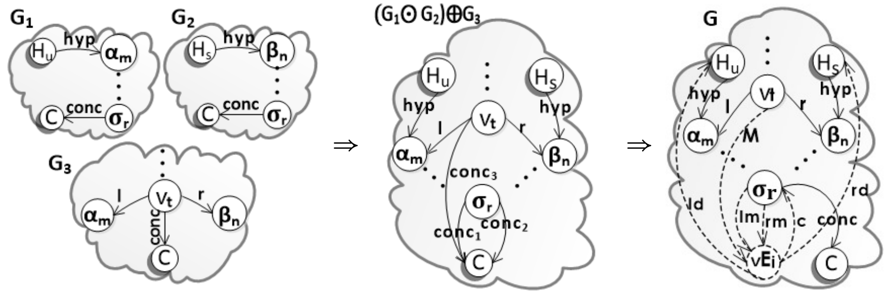
$(G_1 \odot G_2) \oplus G_3$ [5] (intermediate step) contains the nodes: $\vee_t$ and σ_r linked to the D-node C (σ_r twice); and α_m and β_n are subformulas of $\vee_t$ and are linked to D-nodes H, then the graph G that is defined as $(G_1 \odot G_2) \oplus G_3$ with

(i) the removal of the ingoing edges in the node C which were generated in the intermediate step (see figure below);

(ii) an R-node $\vee E_i$ at the top position;

(iii) the edges: $\sigma_r \xrightarrow{lm_w} \vee E_i$, $\sigma_r \xrightarrow{rm_w} \vee E_i$, $\vee_t \xrightarrow{M_w} \vee E_i$, $\vee E_i \xrightarrow{c_w} \sigma_r$, $\vee E_i \xrightarrow{ld_w} H_u$, $\vee E_i \xrightarrow{rd_w} H_s$ and $\sigma_r \xrightarrow{conc}$C, where w is a fresh (new) index considering all edges of kind p, d and c ingoing and/or outgoing of the formula-nodes α_m, β_n and $\rightarrow_q$;

is a fpl-graph, see figure below.



→Iv, $\wedge E$**l,** $\vee I$**l,** $\vee I$**r** Similar to other cases of construction (for an expanded version see [5]).

In the terminology about inference rules or R-nodes, when an R-node has more than one incoming edge, these are distinguished by calling them left, right, major or minor, or a combination of these terms and so also the F-node 'premise' associated with these edges. Thus, the major premise in R-node contains the connective that is eliminated; the other premise in R-node is called 'minor'. Two premises that play a more or less equal role in the inference are called 'left' and 'right'. For instance, an R-node ∨E has a major premise, a left minor premise and a right minor premise; an R-node ∧I has a left premise and a right premise.

The term *R-node sequence* is representing a deduction, and if it is a smaller part of another R-node sequence (deduction), then it is called *a subsequence* of the latter. A subsequence that derives a premise of the last R-node application in an R-node sequence is called a direct R-node subsequence. Instead of writing "the direct R-node subsequence that derives the minor premise of the last inference of an R-node sequence D", we simply write *"the minor subsequence of D"*.

Fpl-graphs need to conform a number of restrictions. To formulate the first one, acyclicity, we need the notion of *inferential order* on R-nodes that allows to pass through the nodes of the structure preventing nodes from infinitely reoccurring in a *path*.

[5] By definition $G_1 \odot G_2$ equalizes or collapses the R-nodes of G_1 with the R-nodes of G_2 that have the same set of premises and conclusion keeping the inferential order of each node, and equalizes F-nodes of G_1 with the F-nodes of G_2 that have the same label, and equalizes edges with the same source, target and label into one.

Definition 2.3 Let G be a fpl-graph. An *inferential order* $<$ on nodes of G is a partial ordering of the R-nodes of G such that $n < n'$ iff n and n' are R-nodes and there is an F-node f such that $n \xrightarrow{lbl_1} \mathrm{f} \xrightarrow{lbl_2} n'$ and lbl_1 is c and lbl_2 is m, or lbl_1 is c and lbl_2 is M, or lbl_1 is c and lbl_2 is p. Node n is a *top position* node if n is maximal w.r.t. $<$.

Definition 2.4 (1) For $n_i \in V$, a *path* in fpl-graph is a sequence of vertices and edges of the form: $n_1 \xrightarrow{lbl_1} n_2 \xrightarrow{lbl_2} \ldots \xrightarrow{lbl_{k-2}} n_{k-1} \xrightarrow{lbl_{k-1}} n_k$, such that n_1 is a hypothesis formula node, n_k is the conclusion formula node, n_i alternating between a rule node and a formula node. The edges lbl_i alternate between two types of edges: the first is $lbl_j \in \{rm, lm, m, M, rp, lp, p\}$ and the second $lbl_j = c$. (2) A *branch* in fpl-graph is an initial part of a *path* which stops at the conclusion F-node of the graph or at the first minor (or left) premise whose major (or right) premise is the conclusion of a rule node. (3) An *insertable branch* in fpl-graph is a branch that is bifurcated by a maximal formula: $\rightarrow I$ followed by $\rightarrow E$.

The Lemma 2.5 below enables us to prove that a given graph G is a fpl-graph. Among others it says that we have to check that each node of G is of one of the possible types that generate the construction cases of Definition 2.2.

In order to avoid overloading of indexes, we will omit whenever possible, the indexing of edges of kind lm, rm, lp, rp, ld and rd, keeping in mind that the coherence of indexing is established by the kind of rule-node to which they are linked.

Lemma 2.5 *G is a fpl-graph if and only if the following hold:*

(i) *There exists a well-founded (hence acyclic) inferential order $<$ on all rule nodes of the fpl-graph.*

(ii) *Every node N of G is of one of the following ten types:*

P *N is labelled with one of the propositional letters: $\{P,\ Q,\ R,\ \ldots\ \}$. N has no outgoing edges l and r.*

F *N has one of the following labels: $\rightarrow_i$, $\wedge_j$ or $\vee_k$, and has exactly two outgoing edges with label l and r. N has outgoing edges with labels p, m, M, lm, rm, lp, rp; and ingoing edges with label c and hyp.*

E$^{\rightarrow}$ *N has label $\rightarrow E_i$ and has exactly one outgoing edge $\rightarrow E_i \xrightarrow{c} \beta_n$, where β_n is a node type* **P** *or* **F**. *N has exactly two ingoing edges $\alpha_m \xrightarrow{m} \rightarrow E_i$ and $\rightarrow_q \xrightarrow{M} \rightarrow E_i$, where α_m is a node type* **P** *or* **F**. *There are two outgoing edges from the node $\rightarrow_q$: $\rightarrow_q \xrightarrow{l} \alpha_m$ and $\rightarrow_q \xrightarrow{r} \beta_n$.*

I$^{\rightarrow}$ *N has label $\rightarrow I_j$ (or $\rightarrow Iv_j$, if discharges an hypothesis vacuously), has one outgoing edge $\rightarrow I_j \xrightarrow{c} \rightarrow_t$, and one (or zero for the case $\rightarrow Iv$) outgoing edge $\rightarrow I_j \xrightarrow{d} H_k$. N has exactly one ingoing edge: $\beta_n \xrightarrow{p} \rightarrow I_j$, where β_n is a node type* **P** *or* **F**. *There are two outgoing edges from the node $\rightarrow_t$: $\rightarrow_t \xrightarrow{l} \alpha_m$ and $\rightarrow_t \xrightarrow{r} \beta_n$ such that there is one (or zero for the case $\rightarrow Iv$) ingoing edge to the node α_m: $H_k \xrightarrow{hyp} \alpha_m$.*

I$^\wedge$ *N has label* $\wedge I_i$*, one outgoing edge* $\wedge I_i \xrightarrow{c} \wedge_t$ *and exactly two ingoing edges:* $\alpha_m \xrightarrow{lp} \wedge I_i$ *and* $\beta_n \xrightarrow{rp} \wedge I_i$*, where* α_m *and* β_n *are nodes type* **P** *or* **F**. *There are two outgoing edges from the node* $\wedge_t$*:* $\wedge_t \xrightarrow{l} \alpha_m$ *and* $\wedge_t \xrightarrow{r} \beta_n$*.*

E$^\wedge$ *N has label* $\wedge E_i$*, one outgoing edge* $\wedge El_i \xrightarrow{c} \alpha_m$ *where* α_m *(or* β_n *in the case* $\wedge Er_i$ *is a node type* **P** *or* **F** *and has exactly one ingoing edge:* $\wedge_t \xrightarrow{p} \wedge E_i$*. There are two outgoing edges from the node* $\wedge_t$*:* $\wedge_t \xrightarrow{l} \alpha_m$ *and* $\wedge_t \xrightarrow{r} \beta_n$*.*

I$^\vee$ *N has label* $\vee Il_i$*, one outgoing edge* $\vee Il_i \xrightarrow{c} \vee_t$ *and has exactly one ingoing edge:* $\alpha_m \xrightarrow{p} \vee Il_i$ *where* α_m *(or* β_n *in the case* $\vee Ir_i$*) is a node type* **P** *or* **F**. *There are two outgoing edges from the node* $\vee_t$*:* $\vee_t \xrightarrow{l} \alpha_m$ *and* $\vee_t \xrightarrow{r} \beta_n$*.*

E$^\vee$ *N has label* $\vee E_i$*, three outgoing edges* $\vee E_i \xrightarrow{c} \sigma_r$*,* $\vee E_i \xrightarrow{ld} H_u$ *and* $\vee E_i \xrightarrow{rd} H_s$*; and it has exactly three ingoing edges:* $\vee_t \xrightarrow{M} \vee E_i$*,* $\sigma_r \xrightarrow{lm} \vee E_i$*,* $\sigma_r \xrightarrow{rm} \vee E_i$ *where* α_m *(or* β_n *in the case* $\vee E_i$*) is a node type* **P** *or* **F**. *There are two outgoing edges from the node* $\vee_t$*:* $\vee_t \xrightarrow{l} \alpha_m$*,* $\vee_t \xrightarrow{r} \beta_n$ *and the hypothesis edges:* $H_u \xrightarrow{hyp} \alpha_m$ *and* $H_s \xrightarrow{hyp} \beta_n$*.*

H *N has label* H_k *and has exactly one outgoing edge hyp.*

C *N has label C and has exactly one ingoing edge conc.*

Proof.

$\Rightarrow$: Argue by induction on the construction of fpl-graph (Definition 2.2). For every construction case for fpl-graphs we have to check the three properties stated in Lemma. Property (2) is immediate. For property (1), we know from the induction hypothesis that there is an inferential order $<$ on R-nodes of the fpl-graph. In construction cases $\rightarrow I$, $\rightarrow E$, $\wedge I$, $\wedge E$l, $\wedge E$r, $\vee I$l, $\vee I$r or $\vee E$, we make the new R-node that is introduced highest in the $<$-ordering, which yields an inferential ordering on R-nodes. In the construction case $\wedge I$, when we have two inferential orderings, $<_1$ on G_1 and $<_2$ on G_2. Then $G_1 \oplus G_2$ can be given an inferential ordering by taking the union of $<_1$ and $<_2$ and in addition putting $n < m$ for every R-node n, m such that $n \in G_1, m \in G_2$. In the construction case $\vee$E, when we have three inferential orderings, $<_1$ on G_1, $<_2$ on G_2 and $<_3$ on G_3. Then $(G_1 \odot G_2) \oplus G_3$ can be given an inferential ordering by taking the union of $<_1$, $<_2$ and $<_3$ and in addition putting $n < m < p$ for every R-node n, m, p such that $n \in G_1, m \in G_2, p \in G_3$.

$\Leftarrow$: Argue by induction on the number of R-nodes of G. Let $<$ be the topological order that is assumed to exist. Let n be the R-node that is maximal w.r.t. $<$. Then n must be on the top position. When we remove node n, including its edges linked (if n is of type **I**$^\vee$) and the node type **C** is linked to the premise of the R-node, we obtain a graph G' that satisfies the properties listed in Lemma. By induction hypothesis we see that G' is a fpl-graph. Now we can add the node n again, using one of the construction cases for fpl-graphs: *Basis* if n is a **L** node, **F** node, $\rightarrow E$ node or $\rightarrow I$ node, **I**$^\wedge$ if n is a $\wedge I$ node, **E**$^\wedge$ if n is a $\wedge E$l node or $\wedge E$r node, **I**$^\vee$ if n is a $\vee I$l node or $\vee I$r node, **E**$^\vee$ if n is a $\vee E$ node. $\square$

3 Normalization for fpl-graphs

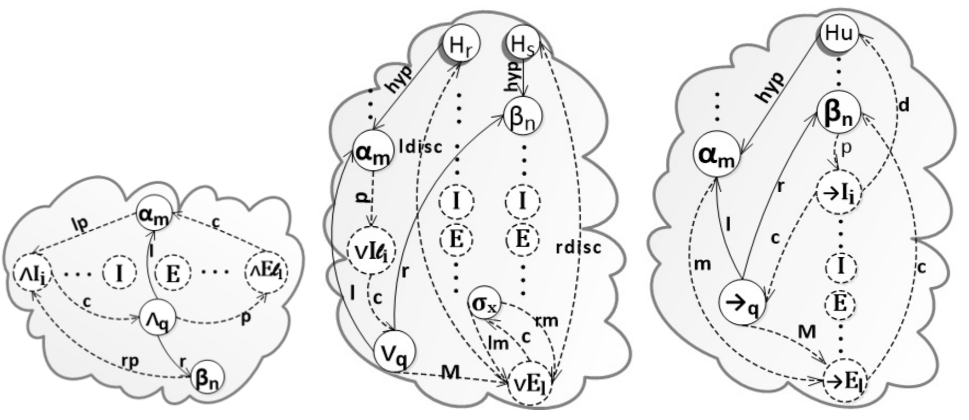
3.1 Elimination of maximal formula

In this section, we describe the normalization process for fpl-graphs. Eliminating a maximal formula is very similar to the procedure for mimp-graphs described in [6], where we considered only the case of implication, now we define the maximal formulas in conjunction, disjunction and implication. The notion of *reordering* is provided as well, because when the maximal formula is removed a reordering of nodes occurs.

Definition 3.1 A *maximal formula* m in a fpl-graph G is a sub-graph of G as follows:

- $\wedge I$ **followed by** $\wedge E$**l.** It is composed of (see Figure 3(a)):
  (i) the F-nodes: α_m, β_n and $\wedge_q$, where $\wedge_q$ has zero or more ingoing/outgoing edges, e.g. $\wedge_q$ could be premise or conclusion of others R-nodes;
  (ii) the R-nodes: $\wedge I_i$ and $\wedge El_l$, where $\wedge I_i$ has an inferential order lower than $\wedge El_l$ and there are zero or more maximal formulas between them [6]. If these nodes occur in different branches, a branch must be insertable in the other branch or bifurcated by an R-node $\vee$E;
  (iii) the edges: $\wedge_q \xrightarrow{l} \alpha_m$, $\wedge_q \xrightarrow{r} \beta_n$, $\alpha_m \xrightarrow{lp} \wedge I_i$, $\beta_n \xrightarrow{rp} \wedge I_i$, $\wedge I_i \xrightarrow{c} \wedge_q$, $\wedge_q \xrightarrow{p} \wedge El_l$ and $\wedge El_l \xrightarrow{c} \alpha_m$.

  There is a symmetric case for $\wedge I$ followed by $\wedge E$r.



(a) $\wedge I$ followed by $\wedge E$l (b) $\vee I$l followed by $\vee E$ (c) $\rightarrow I$ followed by $\rightarrow E$

Fig. 3. Maximal Formulas

- $\vee I$**l followed by** $\vee E$**.** It is composed of (see Figure 3(b)):
  (i) the F-nodes: α_m, β_n, $\vee_q$ and σ_x, where $\vee_q$ has zero or more ingoing/outgoing edges;
  (ii) the D-nodes: H_r and H_s;

[6] The maximal formulas are represented in the figure by nodes labelled with I and E

(iii) the R-nodes in ascending inferential order: $\vee Il_i$ and $\vee E_l$, and there are zero or more maximal formulas in branches between them. If these nodes occur in different branches, a branch must be insertable in the other branch or bifurcated by an R-node $\vee E$;

(iv) the edges: $\vee_q \xrightarrow{l} \alpha_m$, $\vee_q \xrightarrow{r} \beta_n$, $\alpha_m \xrightarrow{p} \vee Il_i$, $\vee Il_i \xrightarrow{c} \vee_q$, $\vee_q \xrightarrow{M} \vee E_l$, $\sigma_x \xrightarrow{lm} \vee E_l$, $\sigma_x \xrightarrow{rm} \vee E_l$, $\vee E_l \xrightarrow{c} \sigma_x$, $\vee E_l \xrightarrow{ld} H_r$ and $\vee E_l \xrightarrow{rd} H_s$.

There is a symmetric case for $\vee Ir$ followed by $\vee E$.

- **$\to I$ followed by $\to E$.** It is composed of (see Figure 3(c)):

(i) the formula nodes: α_m, β_n and $\to_q$, where $\to_q$ has zero or more ingoing/outgoing edges;

(ii) the D-node: H_u;

(iii) the R-nodes in ascending inferential order: $\to I_i$ and $\to E_l$, and there are zero or more maximal formulas between them. If these nodes occur in different branches, a branch must be insertable in the other branch or bifurcated by an R-node $\vee E$;

(iv) the edges: $\to_q \xrightarrow{l} \alpha_m$, $\to_q \xrightarrow{r} \beta_n$, $\beta_n \xrightarrow{p} \to I_i$, $\to I_i \xrightarrow{c} \to_q$, $\to I_i \xrightarrow{d} H_u$, $\to_q \xrightarrow{M} \to E_l$, $\alpha_m \xrightarrow{m} \to E_l$ and $\to E_l \xrightarrow{c} \beta_n$.

Definition 3.2 A reordering of a given fpl-graph G is obtaining by supplying G with the following (new) inferential order on the R-nodes of G.

- $o(t_m) = 0$ for an R-node t_m starting with hypothesis.
- $o(t) = o(t') + 1$ if the conclusion formula of R-node t' is premise, right premise or major premise of t.

Proposition 3.3 *A graph obtained by a reordering according to Definition 3.2 is a fpl-graph.*

Note that the actual situation is more complicated than those sketched in Figures 3(a), 3(b) and 3(c). There are five sub-cases for each maximal formula due to the presence of disjunction and other maximal formulas. For brevity we only show how subcases of the elimination of $\vee I$l followed by $\vee E$ are treated (for an expanded version see [5]).

Definition 3.4 Given a fpl-graph G with a maximal formula m, *eliminating a maximal formula* is the following transformation of a fpl-graph:

Elimination of $\vee I$l followed by $\vee E$ There is a symmetric case for $\vee I$r followed by $\vee E$. The elimination of this maximal formula is the following operation on a fpl-graph:

(i) If there are no *maximal formulas* in branches between the R-nodes $\vee Il_i$ and $\vee E_l$ then follow these steps:

(a) If $\vee Il_i$ and $\vee E_l$ are *not bifurcated* by one $\vee E$ then (see cases 1 and 2 in Figure 4).

Remove the R-nodes $\vee Il_i$ and $\vee E_l$, and their edges.

If the F-node $\vee_q$ only has outgoing edges to sub-formulas then remove it (see case 2 in Figure 4).

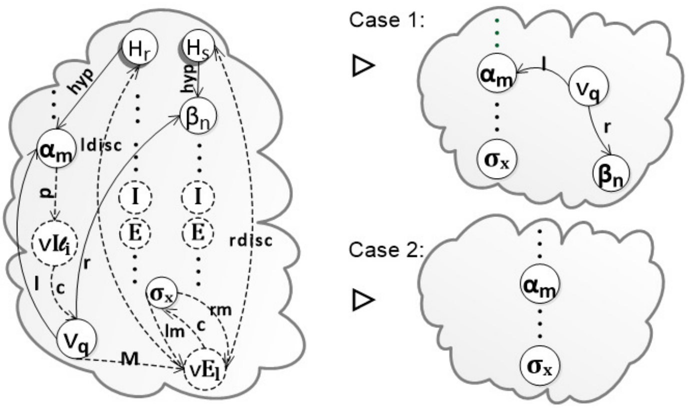


Fig. 4. Elimination of $\vee Il$ followed by $\vee E$: Cases 1 and 2.

(b) Else If $\vee Il_i$ represents two R-nodes then (see case 3 in Figure 5):
Remove R-nodes $\vee Il_i$ and $\vee E_l$, and their edges.
Eliminate edges: $\vee_q \xrightarrow{lm} \vee E_k$, $\vee_q \xrightarrow{rm} \vee E_k$ and $\vee E_k \xrightarrow{c} \vee_q$.
If the F-node $\vee_q$ only has outgoing edges to sub-formulas then remove it (see case 4 in Figure 5).
Add the edges: $\sigma_x \xrightarrow{lm} \vee E_k$, $\sigma_x \xrightarrow{rm} \vee E_k$ and $\vee E_k \xrightarrow{c} \sigma_x$.
Incorporate the inference orders of sequence Π_x^m of the Figure 5 in the minor subsequence of $\vee E_k$ (left and right).

(c) Else (see case 5 in Figure 6)
Remove the R-node $\vee Il_i$, and its edges.
Eliminate edges: $\vee_q \xrightarrow{lm} \vee E_k$, $\vee_q \xrightarrow{rm} \vee E_k$ and $\vee E_k \xrightarrow{c} \vee_q$.
Add the edges: $\sigma_x \xrightarrow{lm} \vee E_k$, $\sigma_x \xrightarrow{rm} \vee E_k$ and $\vee E_k \xrightarrow{c} \sigma_x$.
Incorporate the inference order of node $\vee E_l$ with its subsequences Π_x^m and Π_x^n as shown in Figure 6 in the right minor subsequence of $\vee E_k$ and incorporate the R-node sequence Π_x^m in the left minor premise of $\vee E_k$.

(d) Apply the operation defined in Definition 3.2 to the resulting graph. Note that Proposition 3.3 ensures that the result is a fpl-graph.

(ii) Otherwise eliminate the *maximal formulas* in branches between the R-nodes $\vee Il_i$ and $\vee E_l$.

Lemma 3.5 *If G is a fpl-graph with a maximal formula m and G' is obtained from G by eliminating m, then G' is also a fpl-graph. Moreover G and G' both have the same conclusion, i.e. the F-label being the source of conc.*

Proof. We use Lemma 2.5. All nodes in G' are of the right form: $\mathbf{P}$, $\mathbf{F}$, $\mathbf{E}^{\rightarrow}$, $\mathbf{I}^{\rightarrow}$, $\mathbf{E}^{\vee}$, $\mathbf{I}^{\vee}$, $\mathbf{E}^{\wedge}$, $\mathbf{I}^{\wedge}$, $\mathbf{H}$ or $\mathbf{C}$. We verify that G' has one ingoing edge with label *conc* to the D-node with label C and that is acyclic and connected. Finally, an inferential order on G' (as defined in Definition 3.2) between rule nodes must preserve the derivability and the conclusions. □

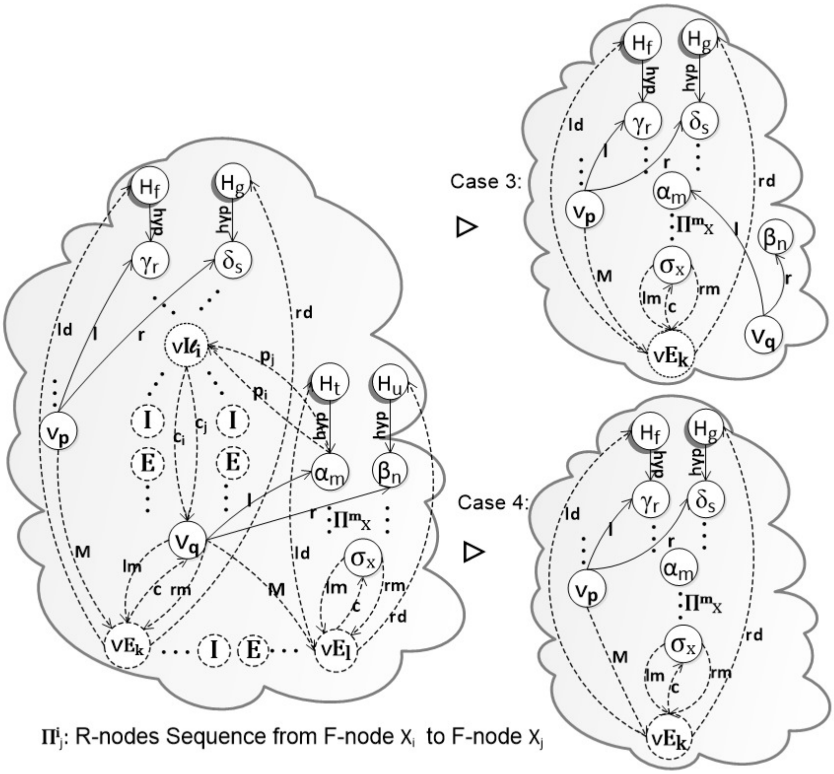


Fig. 5. Elimination of $\vee I$l followed by $\vee E$: Cases 3 and 4.

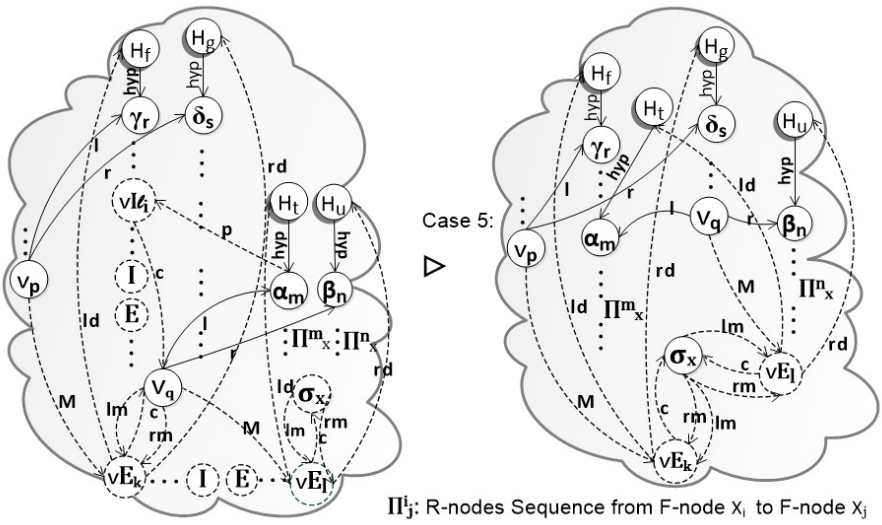


Fig. 6. Elimination of $\vee I$l followed by $\vee E$: Case 5.

3.2 Normalization proof

This proof is guided by the normalization measure. That is, a given fpl-graph G should be transformed into a non-redundant fpl-graph by applying of reduction steps and at each reduction step the measure must be decreased. The normalization measure will be the number of maximal formulas in the fpl-graph.

Theorem 3.6 (Normalization) *Every fpl-graph G can be reduced to a normal fpl-graph G' having the same hypotheses and conclusion as G. Moreover, for any standard tree-like natural deduction Π, if $G := G_\Pi$ (the F-minimal fpl-like representation of Π), then the size of G' does not exceed the size of G, and hence also Π.*

Remark 3.7 The second assertion sharply contrasts to the well-known exponential speed-up of standard normalization. Note that the latter is a consequence of the tree-like structure of standard deductions having different occurrences of equal hypotheses formulas, whereas all formulas occurring in F-minimal fpl-like representations are pairwise distinct.

Proof. This characteristic of preservation of the premises and conclusions of the derivation is proved naturally. Through an inspection of each elimination of maximal formula is observed that the reduction step (see Definition 3.4) of the fpl-graph does not change the set of premises and conclusions (indicated by the D-nodes H and C) of the derivation that is being reduced.

In addition, the demonstration of this theorem has two primary requirements to guarantee that through the elimination of maximal formulas in the fpl-graph, one cannot generate more maximal formulas.The second requirement is to guarantee that during the normalization process, the normalization measure adopted is always reduced.

The first requirement is easily verifiable through an inspection of each case in the elimination of maximal formulas. Thus, it is observed that no case produces more maximal formulas. The second requirement is established through the normalization procedure (see Section 3.2.1) and demonstrated through an analysis of existing cases in the elimination of maximal formulas in fpl-graphs. To support this statement, it is used the notion of normalization measure, we adopt as measure of complexity (induction parameter) the number of maximal formulas $Nmax(G)$. Besides, as already mentioned, working with F-mimimal fpl-graph representations we can use as optional inductive parameter the ordinary size of fpl-graphs. □

3.2.1 Normalization Process

We know that a specific propositional fpl-graph G can have one or more maximal formulas represented by $M_1, ..., M_n$. Thus, the normalization procedure is:

(i) Identify the number of maximal formulas $Nmax(G)$.

(ii) Choose a maximal formula represented by M_k.

(iii) Eliminate M_k as defined in Definition 3.4, creating a new graph G.

(iv) In this application one, of the following six cases may occur:
- **a)** The maximal formula is removed (case 1 in all eliminations of maximal formulas).
- **b)** The maximal formula is removed but the formula node is maintained, and, $Nmax(G)$ is decreased (case 2 in all eliminations of maximal formulas);

- **c)** Two maximal formula are removed (case 3 in all eliminations of maximal formulas).
- **d)** Two maximal formula are removed but the formula node is maintained, hence $Nmax(G)$ is decreased (case 4 in all eliminations of maximal formulas).
- **e)** The maximal formula is removed, the formula node is maintained and R-node sequence reordered, hence $Nmax(G)$ is decreased (case 5 in all eliminations of maximal formulas).
- **f)** All maximal formulas are removed.

(v) Repeat this process until the normalization measure $Nmax(G)$ is reduced to 0 and G becomes *a normal fpl-graph*.

Since the process of the eliminating a maximal formula on fpl-graphs always ends in the elimination of at least one maximal formula, and with the decrease in the number of vertices of the graph, we can say that this normalization theorem is directly a *strong normalization theorem*.

The following is an example illustrating the fact that exponential representation of proofs is avoided in this formalism.

Example on Fibonacci numbers: Consider the formulas: 1) $\eta = A_1 \to A_2$, and 2) $\sigma_k = A_{k-2} \to (A_{k-1} \to A_k)$ for $k > 2$. Note that the size of a normal proof of $A_1 \to A_n$ from $\eta, \sigma_3, \ldots, \sigma_n$ has size bigger than $Fibonnaci(n)$

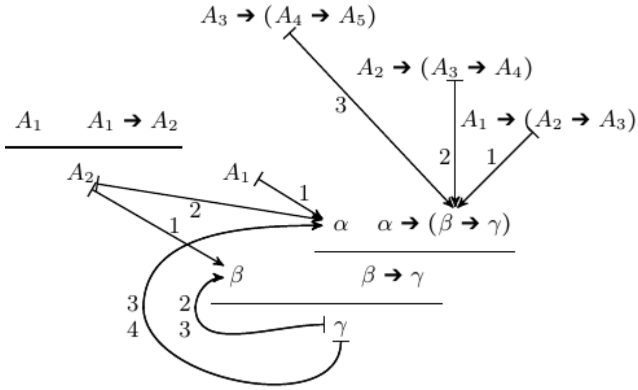
$$\dfrac{\dfrac{\dfrac{\begin{array}{c}[A_1]\\ A_1 \to A_2\\ A_1 \to (A_2 \to A_3)\\ \Pi_3\\ A_3\end{array} \quad \dfrac{\dfrac{[A_1] \quad A_1 \to A_2}{A_2} \quad A_2 \to (A_3 \to A_4)}{A_3 \to A_4}}{A_4} \quad \dfrac{\begin{array}{c}[A_1]\\ A_1 \to A_2\\ A_1 \to (A_2 \to A_3)\\ \Pi_3\\ A_3\end{array} \quad A_3 \to (A_4 \to A_5)}{A_4 \to A_5}}{A_5}}{A_1 \to A_5}$$

Generally, for each $5 \leq k$ we have

$$\dfrac{\dfrac{\begin{array}{c}[A_1]\\ \eta\\ \sigma_3, \ldots, \sigma_{k-1}\\ \Pi_{k-1}\\ A_{k-1}\end{array} \quad \dfrac{\begin{array}{c}[A_1]\\ \eta\\ \sigma_3, \ldots, \sigma_{k-2}\\ \Pi_{k-2}\\ A_{k-2}\end{array} \quad A_{k-2} \to (A_{k-1} \to A_k)}{A_{k-1} \to A_k}}{A_k}}{A_1 \to A_k}$$

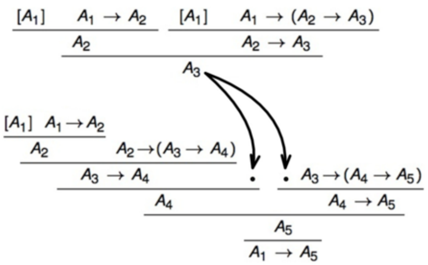
$$l(\Pi_2) = 1$$
$$l(\Pi_3) = l(\Pi_2) + 1$$
$$l(\Pi_k) = l(\Pi_{k-2}) + l(\Pi_{k-1}) + 2$$
$$Fibonacci(k) \leq l(\Pi_k)$$

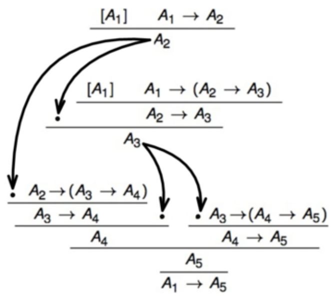
It is doubtful that interpolants would provide a polynomial proof for the same conclusion of this huge proof. However, using graph/dag representation it is possible to obtain a polynomial proof (cf. [3] and [2]). Now in the contexts of our present graph-representation allowing only one formula occurrence of each formula in the proof, we produce a following polynomial size graph-like proof

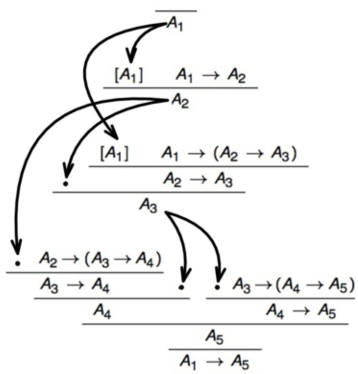


that actually can be obtained by merging distinct occurrences of identical formulas A_3, A_2, A_1 successively as shown below.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}[A_1]\\ A_1 \to A_2\\ A_1 \to (A_2 \to A_3)\\ \Pi_3\\ A_3\end{matrix} \quad \dfrac{\dfrac{[A_1] \quad A_1 \to A_2}{A_2} \quad A_2 \to (A_3 \to A_4)}{A_3 \to A_4}}{A_4} \quad \dfrac{\begin{matrix}[A_1]\\ A_1 \to A_2\\ A_1 \to (A_2 \to A_3)\\ \Pi_3\\ A_3\end{matrix} \quad A_3 \to (A_4 \to A_5)}{A_4 \to A_5}}{A_5}}{A_1 \to A_5}$$







This example shows that there are cases where a natural graph-like proof compressing allows to close exponential-size gaps between tree-like and graph-like (in fact dag-like) proof representations [7].

4 Conclusions

The results presented for mimp-graph in [6] are extended for fpl-graph. Thus, fpl-graph was introduced through definitions and examples preserving the ability to represent proofs in Natural Deduction. The minimal formula representation is a key feature of the fpl-graph structure, because as we saw earlier, it is easy to determine maximal formulas and upper bounds in the length of reduction sequences to leading to normal proofs. A normalization theorem was proved by counting the number of maximal formulas in the original derivation. The strong normalization property is a direct consequence of such normalization, since any reduction decreases the corresponding measures of derivation complexity. This is a preliminary step into investigating how a theorem prover based on graphs is more efficient than usual theorem provers.

We advice the reader, that although the example on Fibonacci numbers shows an exponential gap between tree and dag-like representations, this is not the general case. There are proofs that even in dag representation have exponentially many nodes regarding the size of their conclusions.

References

[1] M. Finger. Dag sequent proofs with a substitution rule. In *We will show Them – Essays in honour of Dov Gabbay 60th birthday*, volume 1 of *Kings College Publications*, pages 671–686. Kings College, London, 2005.

[2] L. Gordeev, E. H. Haeusler, and V. G. Costa. Proof compressions with circuit-structured substitutions. *Journal of Mathematical Sciences*, 158(5):645–658, apr 2009.

[3] Edward Hermann Haeusler and Vaston Gonçalves da Costa. A discussion on compressing proofs through proof-theoretical techniques. *Christiano Braga (Ed.)*, 2012.

[4] Anjolina Grisi Oliveira and Ruy J.G.B. Queiroz. Geometry of deduction via graphs of proofs. In Ruy J.G.B. Queiroz, editor, *Logic for Concurrency and Synchronisation*, volume 15 of *Trends in Logic*, pages 3–88. Springer Netherlands, 2003.

[5] Marcela Quispe-Cruz. On strong normalization in proof-graphs for propositional logic. http://www.tecmf.inf.puc-rio.br/MarcelaCruz?action=AttachFile&do=get&target=pmimp.pdf, 2015.

[6] Marcela Quispe-Cruz, Edward Hermann Haeusler, and Lew Gordeev. Proof-graphs for minimal implicational logic. In *Proceedings 9th International Workshop on Developments in Computational Models, DCM 2013, Buenos Aires, Argentina, 26 August 2013.*, pages 16–29, 2014.

[7] Some additional techniques with regard to the hypotheses discharging by $\rightarrow$-intro rules are also possible, but this topic is out of the scope of our present paper