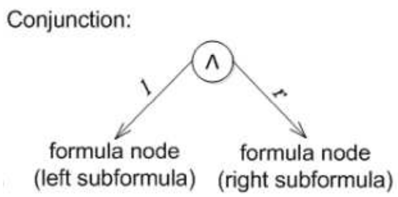
5
Extending mimp-graphs

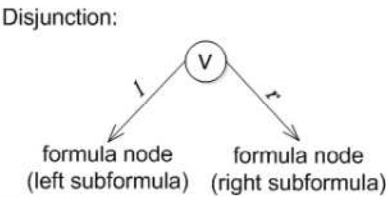
Mimp-graphs provide a formalism for Natural Deduction where the use of a "mixed" graph representation of formulas and inferences (in the purely implicational minimal logic) serve as a way to study the complexity of proofs and to provide more efficient theorem provers. We presented the main notion of mimp-graphs in Chapter 3. In this chapter, we wish to extent this formalism in graphs for full minimal propositional calculus and for first order logic.

5.1
Proof-graphs for propositional logic

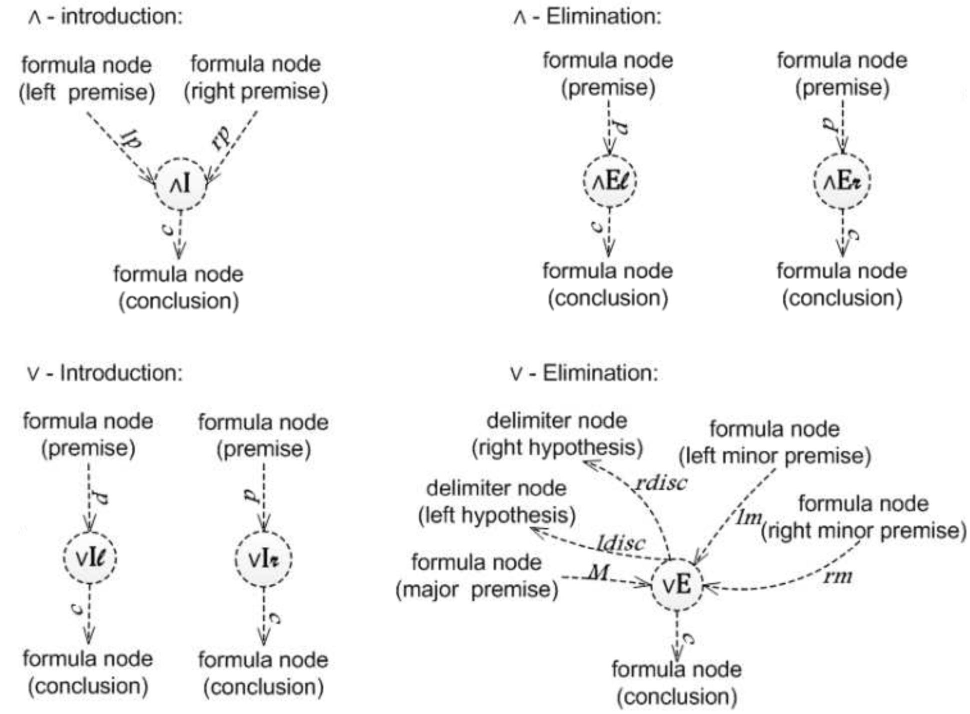
In Chapter 3 we considered implication as the only logic connective. Let us now turn to a more general presentation of proof-graphs for full minimal propositional logic that includes $\rightarrow$ (implication), $\vee$ (disjunction) and $\wedge$ (conjunction). We also develop the normalization procedure for these proof-graphs. Mimp-graphs for propositional logic will be defined along with 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.

- the set of formula labels *F-Labels* in Definition 5 has two added labels $\vee$, $\wedge$;
- the set of inference labels *R-Labels* has the added labels: {$\wedge$I, $\wedge$El, $\wedge$Er, $\vee$Il, $\vee$Ir, $\vee$E};
- the set of edge labels E_M-*Labels* has the added labels: {lp (left premise), rp (right premise), lm (left minor premise), rm (right minor premise) , $ldisc$ (discharge to the left), $rdisc$ (discharge to the right)};
- the definition of formula graphs has two added inductive graphs,

  i.e. 



- inference rules $\wedge$-Introduction, $\wedge$-Elimination, $\vee$-Introduction and $\vee$-Elimination in proof-graphs are as follows:



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 (subdeduction), 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".*

Definition 22 *Let* $G_1 = \langle \mathrm{V}^1, \mathrm{E}^1, \mathrm{L}^1 \rangle$ *and* $G_2 = \langle \mathrm{V}^2, \mathrm{E}^2, \mathrm{L}^2 \rangle$ *be two graphs, where:* V^1 *and* V^2 *are sets of vertices,* E^1 *and* E^2 *are sets of labeled* edges, L^1 *and* L^2 *are subsets of* LBL. *The operation* $G_1 \odot G_2 := \langle \mathrm{V}^1 \sqcup \mathrm{V}^2, \mathrm{E}^1 \ddagger \mathrm{E}^2, \mathrm{L}^1 \cup \mathrm{L}^2 \rangle$ *equalizes R-nodes of* G_1 *with 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 F-nodes of* G_2 *that have the same label, and equalizes edges with the same source, target and label into one.*

Definition 23 *A* mimp-graph *for propositional logic G is a directed graph* $\langle$V, E, L$\rangle$ *where:* V *is a set of nodes,* L *is a set of labels,* E *is a set of labeled* edges $\langle v \in \mathrm{V}, t \in \mathrm{L}, v' \in \mathrm{V}\rangle$*, of source v, of target v' and label t.*

Propositional mimp-graphs are recursively defined as follows:

mimp *Every construction rule for mimp-graphs (Definition 9) is a construction rule for propositional mimp-graph.*

$\wedge$I *If G_1 and G_2 are propositional mimp-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*

1. *$G := G_1 \oplus G_2 \oplus G_3$ with the removal of ingoing edges in the node C which were generated in the intermediate step (see Figure 5.1, dotted area in $G_1 \oplus G_2$);*
2. *an R-node $\wedge \mathrm{I}_i$ at the top position;*
3. *edges: $\alpha_m \xrightarrow{lp_i} \wedge\mathrm{I}_i$, $\beta_n \xrightarrow{rp_i} \wedge\mathrm{I}_i$, $\wedge\mathrm{I}_i \xrightarrow{c_i} \wedge_t$ and $\wedge_t \xrightarrow{conc} C$,*

is a mimp-graph (see Figure 5.1).

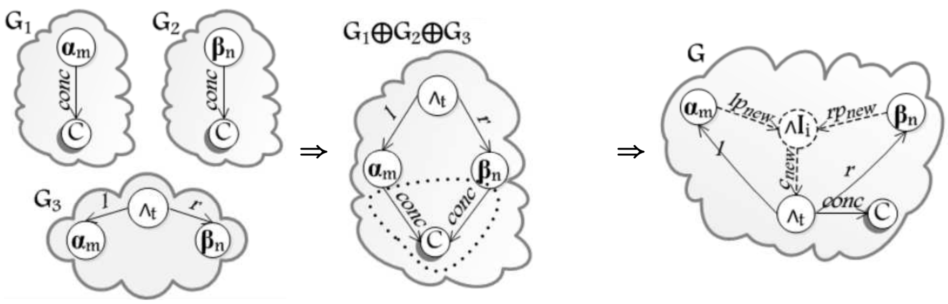


Figure 5.1: The $\wedge$I rule of the propositional mimp-graph.

$\wedge$E*l* *If G_1 is a propositional mimp-graph and contains edges $\wedge_t \xrightarrow{l} \alpha_m$, $\wedge_t \xrightarrow{r} \beta_n$) and the node $\wedge_t$ linked to the D-node C then the graph G that is defined as G_1 with*

1. *the removal of the ingoing edge in the node C;*
2. *an R-node $\wedge\mathrm{E}l_i$ at the top position;*
3. *edges: $\wedge_t \xrightarrow{p_i} \wedge\mathrm{E}l_i$, $\wedge\mathrm{E}l_i \xrightarrow{c_i} \alpha_m$ and $\alpha_m \xrightarrow{conc} C$;*

is a mimp-graph. There is a symmetric case for $\wedge Er$.

$\vee$I*l* *If G_1 is a propositional mimp-graph and contains nodes α_m linked to the D-node C then the graph G that is defined as G_1 with*

1. *the removal of the ingoing edge in the node C.*

2. *an R-node* $\vee\mathrm{I}l_i$ *at the top position;*
3. *edges:* $\vee\mathrm{I}l_i \xrightarrow{c_i} \vee_t$, $\alpha_m \xrightarrow{p_i} \vee\mathrm{I}l_i$, $\vee_t \xrightarrow{l} \alpha_m$, $\vee_t \xrightarrow{r} \beta_n$ *and* $\vee_t \xrightarrow{conc} C$,

is a mimp-graph. There is a symmetric case for $\vee\mathrm{I}r$.

$\vee$**E** *If* G_1, G_2 *and* G_3 *are propositional mimp-graphs, and the graph obtained by* $(G_1 \odot G_2) \oplus G_3$ *(intermediate step) contains 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*

1. *the removal of ingoing edges in the node* C *which were generated in the intermediate step (see Figure 5.2);*
2. *an R-node* $\vee E_i$ *at the top position;*
3. *edges:* $\sigma_r \xrightarrow{lm_i} \vee\mathrm{E}_i$, $\sigma_r \xrightarrow{rm_i} \vee\mathrm{E}_i$, $\vee_t \xrightarrow{M_i} \vee\mathrm{E}_i$, $\vee\mathrm{E}_i \xrightarrow{c_i} \sigma_r$, $\vee\mathrm{E}_i \xrightarrow{ldisc_i} H_u$, $\vee\mathrm{E}_i \xrightarrow{rdisc_i} H_s$ *and* $\sigma_r \xrightarrow{conc} C$,

is a mimp-graph (see Figure 5.2).

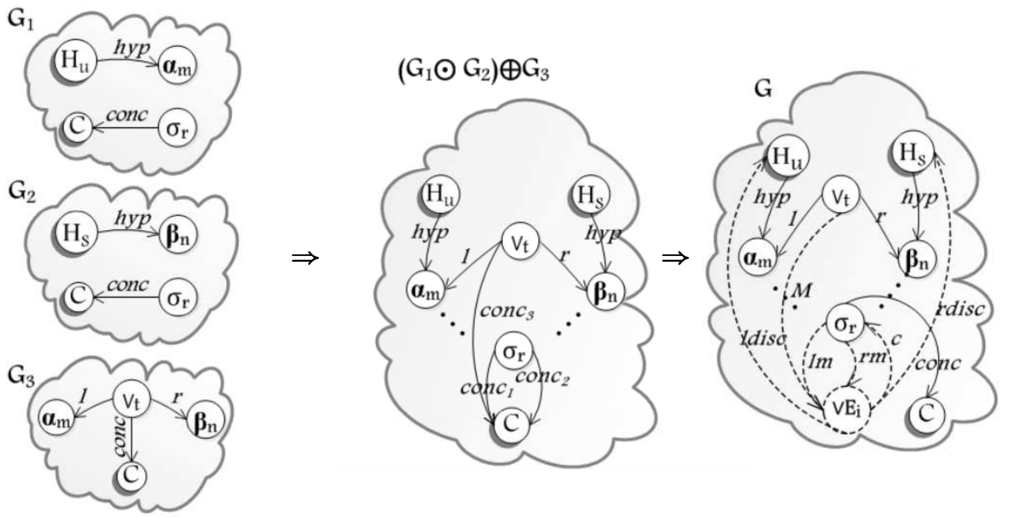


Figure 5.2: The $\vee$E rule of the propositional mimp-graph.

Lemma 5 enables us to prove that a given graph G is a propositional mimp-graph without explicitly supplying a construction. Among others it says that we have to check that each node of G is of one of possible types that generate the construction cases of Definition 23.

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

Lemma 5 *G is a propositional mimp-graph if and only if the following hold:*

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

2. *Every node N of G is of one of the following ten types:*

P *It is as in the Lemma 1.*

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.*

I$^\wedge$ *N has label $\wedge\mathrm{I}_i$, one outgoing edge $\wedge\mathrm{I}_i \xrightarrow{c} \wedge_t$ and exactly two ingoing edges: $\alpha_m \xrightarrow{lp} \wedge I_i$ and $\beta_n \xrightarrow{rp} \wedge\mathrm{I}_i$, 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 $\wedge\mathrm{E}_i$, one outgoing edge $\wedge\mathrm{E}l_i \xrightarrow{c} \alpha_m$ where α_m (or β_n in the case $\wedge\mathrm{E}r_i$ is a node type* **P** *or* **F** *and has exactly one ingoing edge: $\wedge_t \xrightarrow{p} \wedge\mathrm{E}_i$. 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 $\vee\mathrm{I}l_i$, one outgoing edge $\vee\mathrm{I}l_i \xrightarrow{c} \vee_t$ and has exactly one ingoing edge: $\alpha_m \xrightarrow{p} \vee\mathrm{I}l_i$ where α_m (or β_n in the case $\vee\mathrm{I}r_i$) 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 $\vee\mathrm{E}_i \xrightarrow{c} \sigma_r$, $\vee\mathrm{E}_i \xrightarrow{ldisc} H_u$ and $\vee\mathrm{E}_i \xrightarrow{rdisc} H_s$; and it has exactly three ingoing edges: $\vee_t \xrightarrow{M} \vee\mathrm{E}_i$, $\sigma_r \xrightarrow{lm} \vee E_i$, $\sigma_r \xrightarrow{rm} \vee\mathrm{E}_i$ where α_m (or β_n in the case $\vee\mathrm{E}_i$) 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 hypothesis edges: $H_u \xrightarrow{hyp} \alpha_m$ and $H_s \xrightarrow{hyp} \beta_n$.*

I, E, H, C *They are as in the Lemma 1.*

Proof:

$\Rightarrow$: Argue by induction on the construction of propositional mimp-graph (Definition 23). For every construction case for propositional mimp-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 propositional mimp-graph. In the new construction cases $\wedge$I, $\wedge$El, $\wedge$Er, $\vee$Il, $\vee$Ir 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 $\mathrm{I}^{\vee}$) and the node type C is linked to the premise of the R-node, we obtain a graph G' that satisfies properties listed in Lemma. By induction hypothesis we see that G' is a propositional mimp-graph. Now we can add the node n again, using one of construction cases for propositional mimp-graphs: *mimp* if n is a L node, F node, $\rightarrow$E node or $\rightarrow$I node, $\mathrm{I}^{\wedge}$ if n is a $\wedge$I node, $\mathrm{E}^{\wedge}$ if n is a $\wedge$El node or $\wedge$Er node, $\mathrm{I}^{\vee}$ if n is a $\vee$Il node or $\vee$Ir node, $\mathrm{E}^{\wedge}$ if n is a $\vee$E node. ∎

5.2 Normalization for propositional mimp-graphs

5.2.1 Elimination of maximal formula

In this section, we describe the normalization process for propositional mimp-graphs. Eliminating a maximal formula is very similar to the procedure for mimp-graphs described in Chapter 3, where we considered only the case of implication, now we define maximal formulas in conjunction, disjunction and implication:

Definition 24 *A* maximal formula m *in a propositional mimp-graph* G *is a sub-graph of* G *as follows:*

– ***$\wedge$I followed by $\wedge$El.*** *It is composed of (see Figure 5.3):*

1. *F-nodes: α_m, β_n and $\wedge_q$, where $\wedge_q$ has zero or more ingoing/outgoing edges*[1]*, e.g. $\wedge_q$ could be premise or conclusion of others R-nodes;*
2. *R-nodes: $\wedge I_i$ and $\wedge \mathrm{E}l_l$, where $\wedge I_i$ has an inferential order lower than $\wedge \mathrm{E}l_l$ and there are zero or more maximal formulas between them*[2]*. If*

[1]Represented in the figure by double-headed arrows
[2]The maximal formulas are represented in the figure by nodes labeled with I and E

these nodes occur in different branches, a branch must be insertable[3] *in the other branch or bifurcated by an R-node* $\vee E$*;*

3. *edges:* $\wedge_q \xrightarrow{l} \alpha_m$, $\wedge_q \xrightarrow{r} \beta_n$, $\alpha_m \xrightarrow{lp} \wedge I_i$, $\beta_n \xrightarrow{rp} \wedge I_i$, $\wedge I_i \xrightarrow{c} \wedge_q$, $\wedge_q \xrightarrow{p} \wedge \mathrm{E}l_l$ *and* $\wedge \mathrm{E}l_l \xrightarrow{c} \alpha_m$.

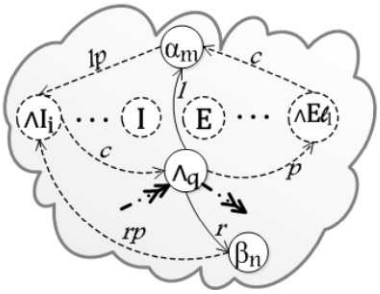


Figure 5.3: The maximal formula: $\wedge I$ followed by $\wedge El$.

There is a symmetric case for $\wedge I$ *followed by* $\wedge Er$.

– $\vee \mathrm{I}l$ ***followed by*** $\vee \mathbf{E}$. *It is composed of (see Figure 5.4):*

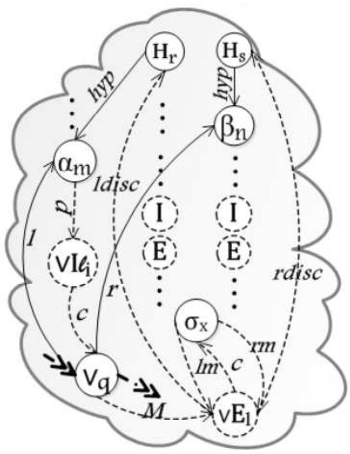


Figure 5.4: The maximal formula: $\vee Il$ followed by $\vee E$.

1. *F-nodes:* α_m, β_n, $\vee_q$ *and* σ_x, *where* $\vee_q$ *has zero or more ingoing/outgoing edges*[4]*;*

2. *D-nodes:* H_r *and* H_s*;*

3. *R-nodes in ascending inferential order:* $\vee \mathrm{I}l_i$ *and* $\vee E_l$, *and there are zero or more maximal formulas in branches between them*[5]. *If these nodes occur in different branches, a branch must be insertable in the other branch or bifurcated by an R-node* $\vee E$*;*

[3] A branch is insertable in other branch when it is bifurcated by a maximal formula: →I followed by →E

[4] Represented in the figure by double-headed arrows

[5] Maximal formulas are represented in the figure by nodes labeled with I and E

4. *edges:* $\vee_q \xrightarrow{l} \alpha_m$, $\vee_q \xrightarrow{r} \beta_n$, $\alpha_m \xrightarrow{p} \vee Il_i$, $\vee Il_i \xrightarrow{c} \vee_q$, $\vee_q \xrightarrow{M} \vee E_l$, $\sigma_x \xrightarrow{lm} \vee E_l$, $\sigma_x \xrightarrow{rm} \vee E_l$, $\vee E_l \xrightarrow{c} \sigma_x$, $\vee E_l \xrightarrow{ldisc} H_r$ *and* $\vee E_l \xrightarrow{rdisc} H_s$.

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

– **➔I *followed by* ➔E.** *It is composed of (see Figure 5.5):*

1. *F-nodes:* α_m, β_n *and* $\to_q$, *where* $\to_q$ *has zero or more ingoing/outgoing edges*[6]*;*
2. *the D-node:* H_u*;*
3. *R-nodes in ascending inferential order:* $\to I_i$ *and* $\to E_l$, *and there are zero or more maximal formulas between them*[7]. *If these nodes occur in different branches, a branch must be insertable in the other branch or bifurcated by an R-node* $\vee E$*;*
4. *edges:* $\to_q \xrightarrow{l} \alpha_m$, $\to_q \xrightarrow{r} \beta_n$, $\beta_n \xrightarrow{p} \to I_i$, $\to I_i \xrightarrow{c} \to_q$, $\to I_i \xrightarrow{disc} H_u$, $\to_q \xrightarrow{M} \to E_l$, $\alpha_m \xrightarrow{m} \to E_l$ *and* $\to E_l \xrightarrow{c} \beta_n$.

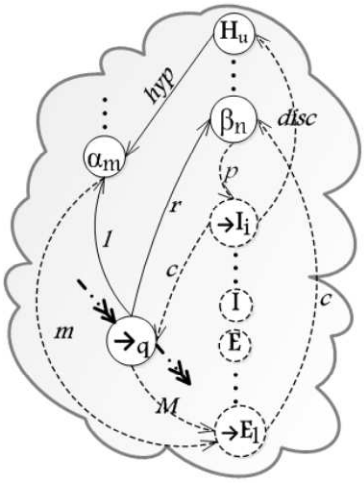


Figure 5.5: The maximal formula: ➔I followed by ➔E.

Definition 25 *The operation* incorporate *adds an R-node sequence inside other R-node sequence where it shares the same formula-graphs premise and conclusion, then apply the operation defined in Definition 13 to the resulting graph. Note that Proposition 1 ensures that the result is a mimp-graph.*

Note that the actual situation is more complicated than those sketched in Figures 5.3, 5.4 and 5.5. There are five sub-cases for each maximal formula due to the presence of disjunction and other maximal formulas. These sub-cases are treated in the Definition 26 as follows.

[6]Represented in the figure by double-headed arrows

[7]Maximal formulas are represented in the figure by nodes labeled with I and E

Definition 26 *Given a propositional mimp-graph G with a maximal formula m,* eliminating a maximal formula *is the following transformation of a propositional mimp-graph:*

Elimination of $\wedge I$ followed by $\wedge El$. *There is a symmetric case for the elimination of $\wedge I$ followed by $\wedge \mathrm{E}r$. The elimination of the maximal formula $\wedge I$ followed by $\wedge \mathrm{El}$ is the following operation on a propositional mimp-graph:*

1. *If there are no* maximal formulas *between R-nodes $\wedge \mathrm{I}_i$ and $\wedge \mathrm{El}_l$ then follow these steps:*
   - (a) *If $\wedge \mathrm{I}_i$ and $\wedge \mathrm{El}_l$ are* not bifurcated *by one $\vee \mathrm{E}$ then (see cases 1 and 2 in Figure 5.6).*
     - i. *Remove R-nodes $\wedge \mathrm{I}_i$ and $\wedge \mathrm{El}_l$ and their edges.*
     - ii. *If the F-node $\wedge_q$ only has outgoing edges to sub-formulas then remove it (see case 2 in Figure 5.6).*

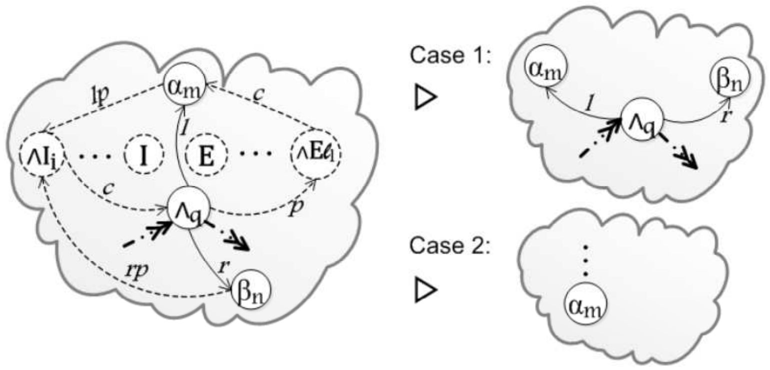


Figure 5.6: Elimination of $\wedge I$ followed by $\wedge El$: Cases 1 and 2.

   - (b) *Else If $\wedge \mathrm{I}_i$ represents two R-nodes then (see case 3 in Figure 5.7):*
     - i. *Remove the R-node $\wedge \mathrm{I}_i$ and its edges.*
     - ii. *Eliminate edges: $\wedge_q \xrightarrow{lm} \vee \mathrm{E}_k$, $\wedge_q \xrightarrow{rm} \vee \mathrm{E}_k$ and $\vee \mathrm{E}_k \xrightarrow{c} \wedge_q$.*
     - iii. *If the F-node $\wedge_q$ only has outgoing edges to sub-formulas then remove it (see case 4 in Figure 5.7).*
     - iv. *Add edges: $\alpha_m \xrightarrow{lm} \vee \mathrm{E}_k$, $\alpha_m \xrightarrow{rm} \vee \mathrm{E}_k$ and $\vee \mathrm{E}_k \xrightarrow{c} \alpha_m$.*

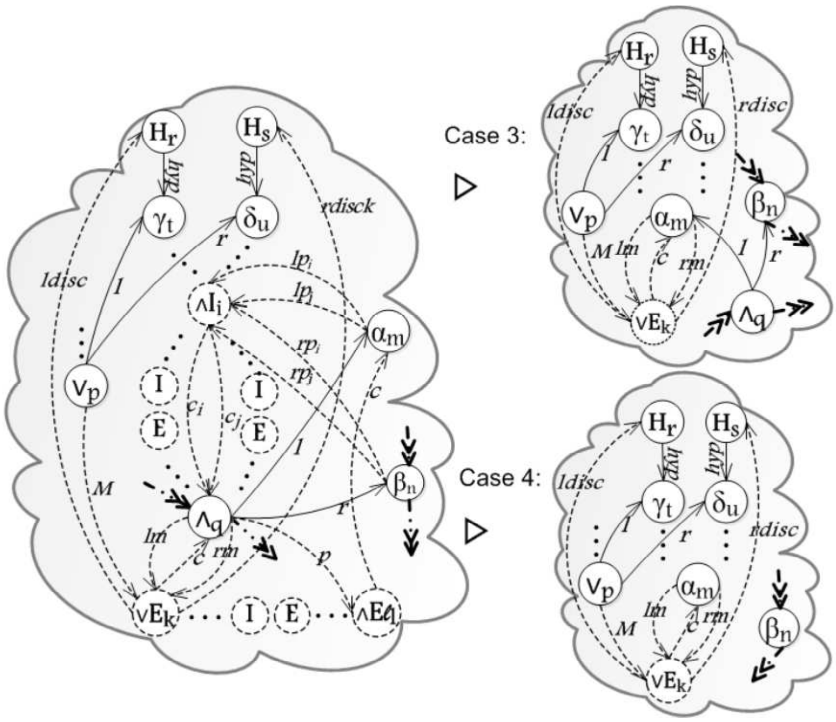


Figure 5.7: Elimination of $\wedge I$ followed by $\wedge El$: Cases 3 and 4.

(c) Else (see case 5 in Figure 5.8)

i. Remove the R-node $\wedge \mathrm{I}_i$*, and its edges.*

ii. Eliminate edges: $\wedge_q \xrightarrow{lm} \vee \mathrm{E}_k$, $\wedge_q \xrightarrow{rm} \vee \mathrm{E}_k$ *and* $\vee \mathrm{E}_k \xrightarrow{c} \wedge_q$.

iii. Add edges: $\alpha_m \xrightarrow{lm} \vee \mathrm{E}_k$, $\alpha_m \xrightarrow{rm} \vee \mathrm{E}_k$ *and* $\vee \mathrm{E}_k \xrightarrow{c} \alpha_m$.

iv. Incorporate the R-node $\wedge \mathrm{El}_l$*, as defined in Definition 25, in the right minor sequence of* $\vee \mathrm{E}_k$ *as last inference.*

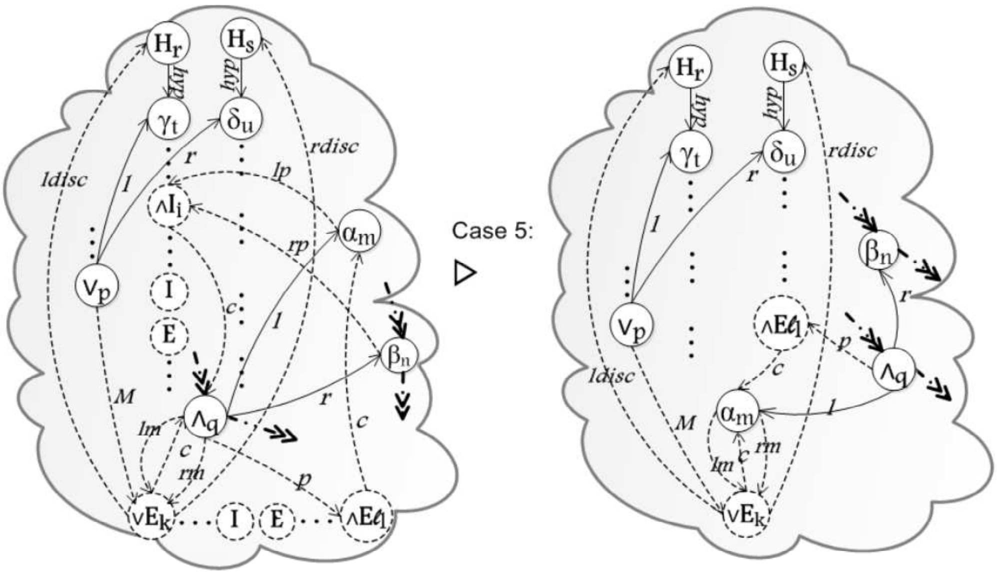


Figure 5.8: Elimination of $\wedge I$ followed by $\wedge El$: Case 5.

2. *Otherwise eliminate* maximal formulas *between R-nodes* $\wedge\mathrm{I}_i$ *and* $\wedge\mathrm{E}l_l$.

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

1. *If there are no* maximal formulas *in branches between R-nodes* $\vee\mathrm{I}l_i$ *and* $\vee\mathrm{E}_l$ *then follow these steps:*
   (a) *If* $\vee\mathrm{I}l_i$ *and* $\vee\mathrm{E}_l$ *are* not bifurcated *by one* $\vee\mathrm{E}$ *then (see cases 1 and 2 in Figure 5.9).*
      i. *Remove R-nodes* $\vee\mathrm{I}l_i$, $\vee\mathrm{E}_l$, H_r *and* H_s, *and their edges.*
      ii. *If the F-node* $\vee_q$ *only has outgoing edges to sub-formulas then remove it (see case 2 in Figure 5.9).*

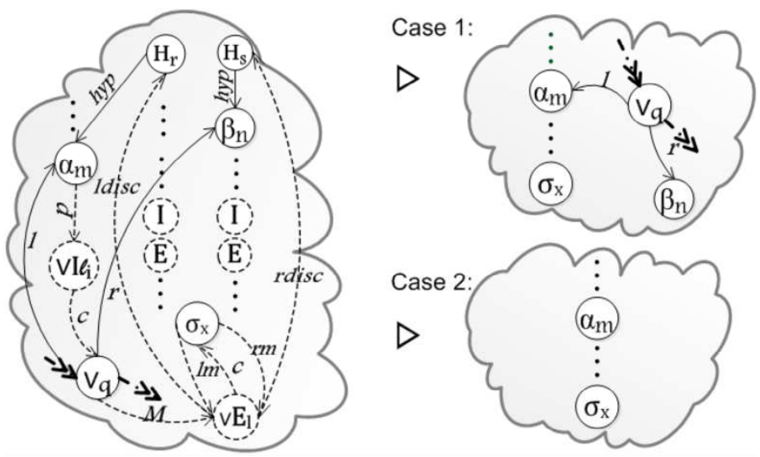


Figure 5.9: Elimination of $\vee Il$ followed by $\vee E$: Cases 1 and 2.

   (b) *Else If* $\vee\mathrm{I}l_i$ *represents two R-nodes then (see case 3 in Figure 5.10):*
      i. *Remove R-nodes* $\vee\mathrm{I}l_i$, $\vee\mathrm{E}_l$, H_t *and* H_u, *and their edges.*
      ii. *Eliminate edges:* $\vee_q \xrightarrow{lm} \vee\mathrm{E}_k$, $\vee_q \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \vee_q$.
      iii. *If the F-node* $\vee_q$ *only has outgoing edges to sub-formulas then remove it (see case 4 in Figure 5.10).*
      iv. *Add edges:* $\sigma_x \xrightarrow{lm} \vee\mathrm{E}_k$, $\sigma_x \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \sigma_x$.
      v. *Incorporate the sequence* Π_x^m *of the Figure 5.10, as defined in Definition 25, in left and right minor subsequences of* $\vee\mathrm{E}_k$.

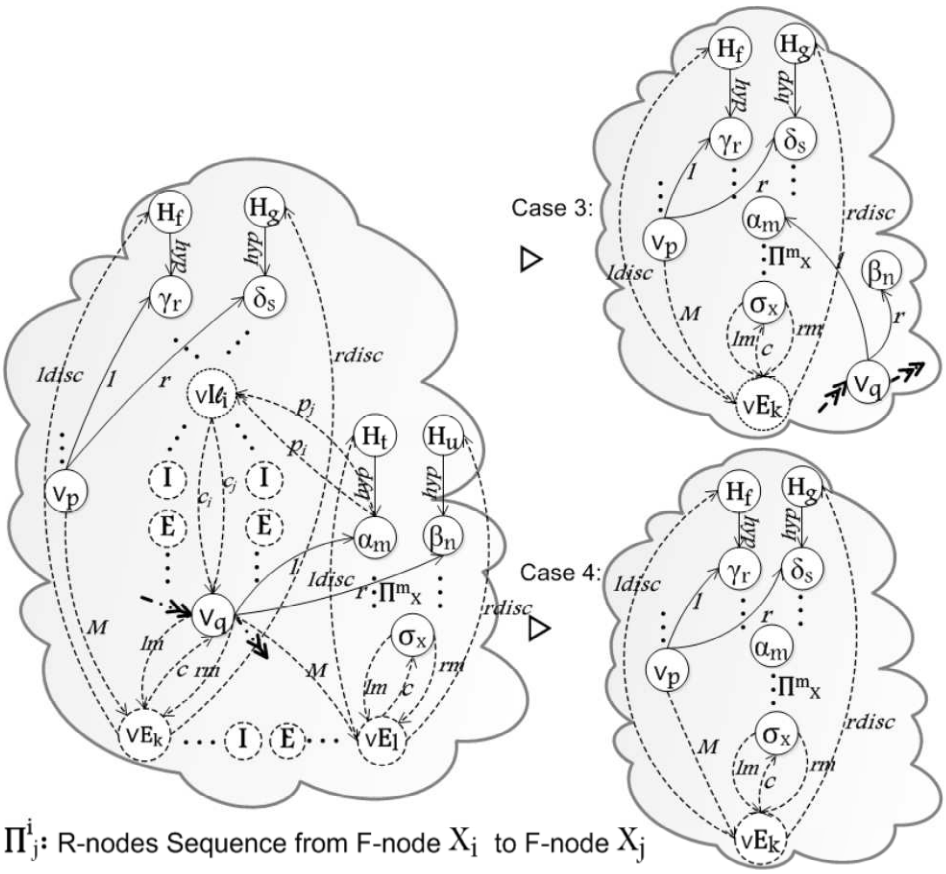


Figure 5.10: Elimination of $\vee Il$ followed by $\vee E$: Cases 3 and 4.

(c) *Else (see case 5 in Figure 5.11)*

i. *Remove the R-node* $\vee\mathrm{I}l_i$*, and its edges.*

ii. *Eliminate edges:* $\vee_q \xrightarrow{lm} \vee\mathrm{E}_k$, $\vee_q \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \vee_q$.

iii. *Add edges:* $\sigma_x \xrightarrow{lm} \vee\mathrm{E}_k$, $\sigma_x \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \sigma_x$.

iv. *Incorporate the R-node* $\vee\mathrm{E}_l$ *with its sub-sequences* Π_x^m *and* Π_x^n *showed in Figure 5.11, as defined in Definition 25, in the right minor subsequence of* $\vee\mathrm{E}_k$ *and incorporate the R-node sequence* Π_x^m *in the left minor premise of* $\vee\mathrm{E}_k$.

2. *Otherwise eliminate the* maximal formulas *in branches between R-nodes* $\vee\mathrm{I}l_i$ *and* $\vee E_l$.

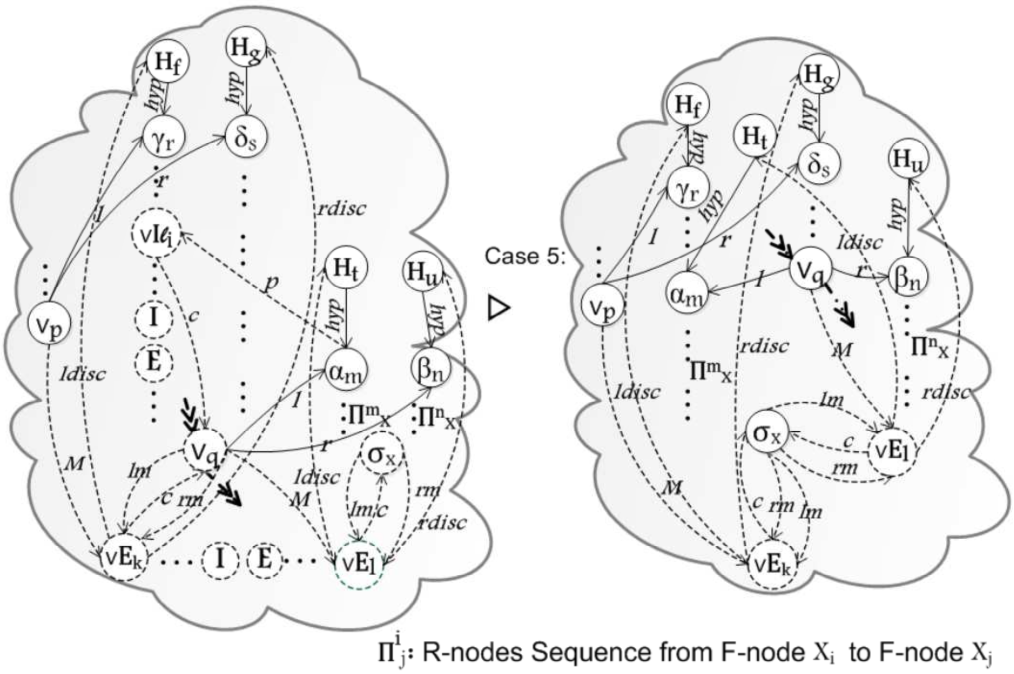


Figure 5.11: Elimination of $\vee Il$ followed by $\vee E$: Case 5.

Elimination of $\rightarrow I$ followed by $\rightarrow E$ *In order to reduce this, we need the following rule:*

1. *If there are no* maximal formulas *between R-nodes* $\rightarrow \mathrm{I}_i$ *and* $\rightarrow \mathrm{E}_l$ *then follow these steps:*

   (a) *If* $\rightarrow \mathrm{I}_i$ *and* $\rightarrow \mathrm{E}_l$ *are* not bifurcated *by one* $\vee$E *then (see cases 1 and 2 in Figure 5.12).*

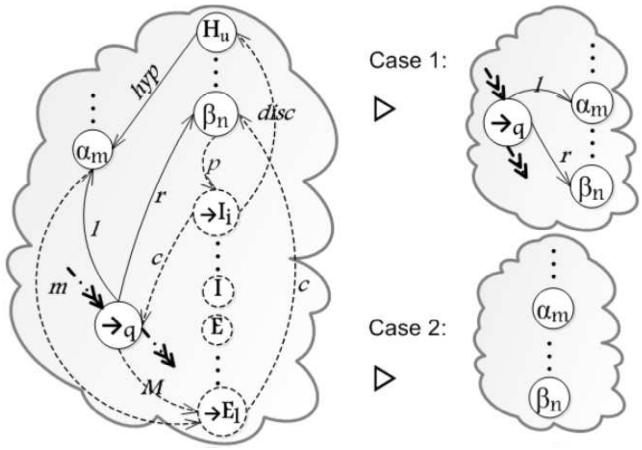


Figure 5.12: Elimination of $\rightarrow I$ followed by $\rightarrow E$: Cases 1 and 2.

i. *If the D-node* H_u*, discharged by* $\to\mathrm{I}_i$*, has* n *outgoing edges with label hyp then repeat* n*-times edges in the minor sub-sequence of* $\to\mathrm{E}_l$*.*

ii. *Remove R-nodes* $\to\mathrm{I}_i$*,* $\to\mathrm{E}_l$ *and* H_u*, and their edges.*

iii. *Remove R-nodes* $\to\mathrm{I}_i$ *and* $\to\mathrm{E}_l$*, and their edges.*

iv. *If the F-node* $\wedge_q$ *only has outgoing edges to sub-formulas then remove it (see case 2 in Figure 5.12).*

(b) *Else If* $\to\mathrm{I}_i$ *represents two R-nodes then (see case 3 in Figure 5.13):*

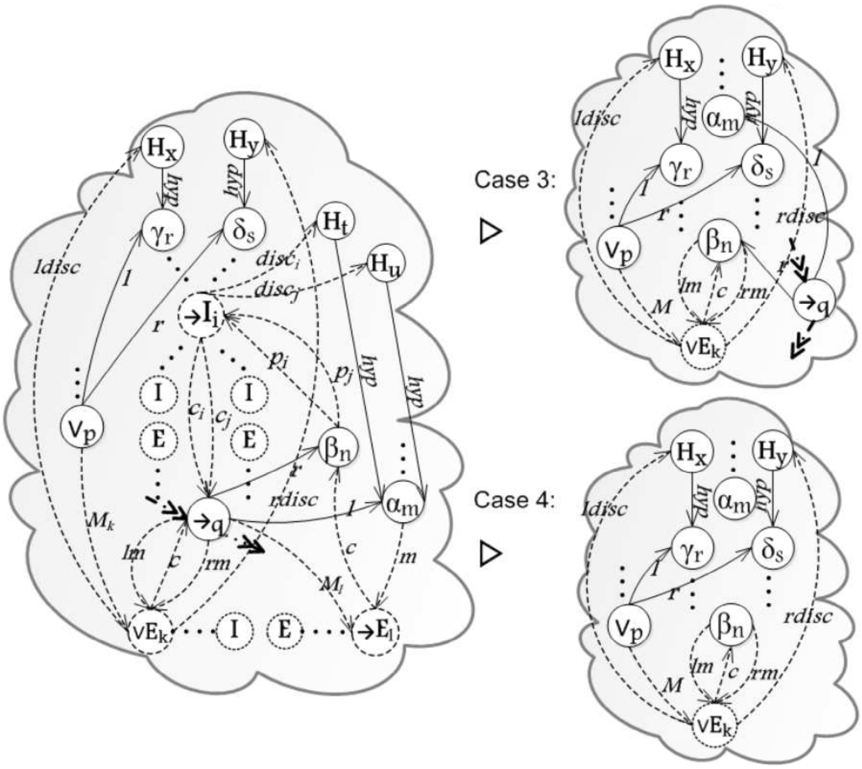


Figure 5.13: Elimination of $\to$I followed by $\to$E: Cases 3 and 4.

i. *Remove R-nodes* $\to\mathrm{I}_i$*,* $\to\mathrm{E}_l$*,* H_t *and* H_u*, and their edges.*

ii. *Eliminate edges:* $\to_q \xrightarrow{lm} \vee\mathrm{E}_k$*,* $\to_q \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \to_q$*.*

iii. *If the F-node* $\to_q$ *only has outgoing edges to sub-formulas then remove it (see case 4 in Figure 5.13).*

iv. *Add edges:* $\beta_n \xrightarrow{lm} \vee\mathrm{E}_k$*,* $\beta_n \xrightarrow{rm} \vee\mathrm{E}_k$ *and* $\vee\mathrm{E}_k \xrightarrow{c} \beta_n$*.*

v. *Incorporate the R-node sequence with conclusion* α_m*, as defined in Definition 25, in left and right minor sub-sequences of* $\vee\mathrm{E}_k$*.*

(c) *Else (see case 5 in Figure 5.14)*

i. *Remove R-nodes* $\to\mathrm{I}_i$ *and* H_t*, and their edges.*

ii. *Eliminate edges:* $\rightarrow_q \xrightarrow{lm} \vee \mathrm{E}_k$, $\rightarrow_q \xrightarrow{rm} \vee \mathrm{E}_k$ *and* $\vee \mathrm{E}_k \xrightarrow{c} \rightarrow_q$.

iii. *Add edges:* $\beta_n \xrightarrow{lm} \vee \mathrm{E}_k$, $\beta_n \xrightarrow{rm} \vee \mathrm{E}_k$ *and* $\vee \mathrm{E}_k \xrightarrow{c} \beta_n$.

iv. *Incorporate the node* $\rightarrow \mathrm{E}_l$, *as defined in Definition 25, in the right minor subsequence of* $\vee \mathrm{E}_k$ *as last inference and the R-node sequence with conclusion* α_m *in the left minor subsequence.*

2. *Otherwise eliminate* maximal formulas *between R-nodes* $\rightarrow \mathrm{I}_i$ *and* $\rightarrow \mathrm{E}_l$.

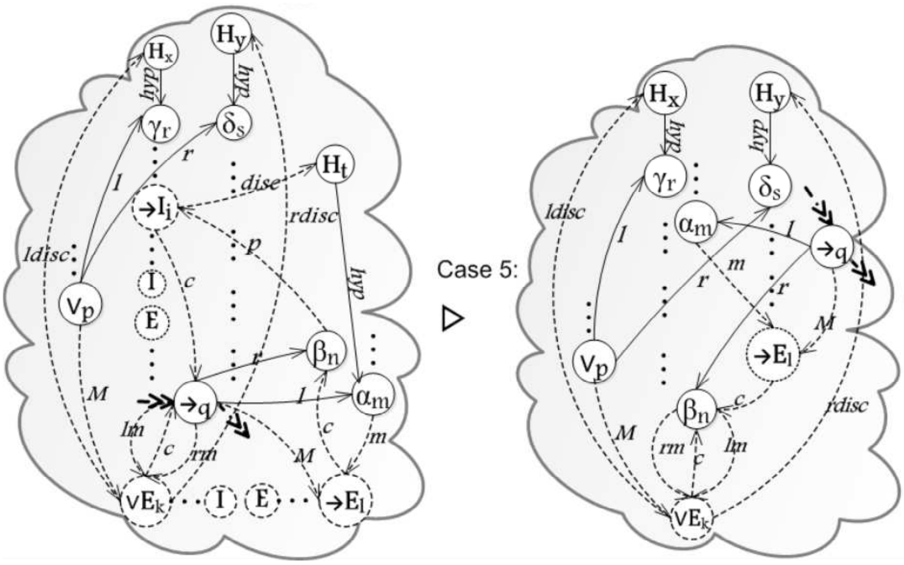


Figure 5.14: Elimination of →I followed by →E: Case 5.

Definition 27 *(1) For* $n_i \in V$, *a* p-path *in propositional mimp-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. 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 propositional mimp-graph is an initial part of a* p-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.*

Lemma 6 *If* G *is a propositional mimp-graph with a maximal formula* m *and* G' *is obtained from* G *by eliminating* m, *then* G' *is also a propositional mimp-graph.*

Proof: We use Lemma 5. All nodes in G' are of the right form: P, K, E, I, $\mathrm{E}^{\vee}$, $\mathrm{I}^{\vee}$, $\mathrm{E}^{\wedge}$, $\mathrm{I}^{\wedge}$, H or 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 13) between rule nodes must preserve. ■

5.2.2 Normalization proof

Just as with mimp-graphs, we shall also construct the normalization proof for these extended mimp-graphs. This proof is guided by the normalization measure. That is, the general mechanism from the proof determines that a given mimp-graph G should be transformed into a non-redundant mimp-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 mimp-graph.

Theorem 5 (Normalization) *Every propositional mimp-graph G can be reduced to a normal propositional mimp-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 mimp-like representation of Π, cf. Theorem 3), then the size of G' does not exceed the size of G, and hence also Π.*

Remark 2 *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 mimp-like representations are pairwise distinct.*

Proof: This characteristic of preservation of 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 26) of the propositional mimp-graph does not change the set of premises and conclusions (indicated by D-nodes H and C) of the derivation that is being reduced.

In addition, the demonstration of this theorem has two primary requirements. First, we guarantee that through the elimination of maximal formulas in the propositional mimp-graph, 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 5.2.2) and demonstrated through an analysis of existing cases in the elimination of maximal formulas in mimp-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 mimp-graph representations we can use as optional inductive parameter the ordinary size of mimp-graphs. ■

Normalization Process

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

1. Choose a maximal formula represented by M_k.
2. Identify the respective number of maximal formulas $Nmax(G)$.
3. Eliminate M_k as defined in Definition 26, creating a new graph G.
4. 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.
5. Repeat this process until the normalization measure $Nmax(G)$ is reduced to 0 and G becomes *a normal propositional mimp-graph.*

Since the process of the eliminating a maximal formula on propositional mimp-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.*

5.3
Proof-graphs for first order logic

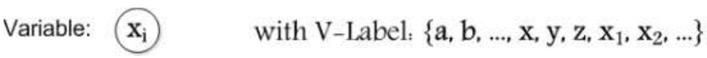
In this section we extend mimp-graphs for propositional logic defined in Section 5.1 to first order logic. This extension can be carried through without much ado and thus we show the robustness of the concept of mimp-graphs.

In Section 5.3.1 we give the definition of these mimp-graphs for first order logic, called *mimp-fol*, starting from definitions for terms, formula graphs of first order logic, and rule nodes for mimp-fol, then the Section 5.3.2 show two examples of mimp-fol. The set of transformations for normalization in mimp-fol is given in Section 5.3.3.

5.3.1
Definition

According to the language of first order logic for Gentzen-Prawitz style natural deduction (Gentzen 1969) (Prawitz 2006) introduced in Chapter 2, we extend mimp-graphs (defined in Chapter 3) to first order logic as follows.

– Variables are represented by nodes as follows:



It is not necessary to differentiate between free and bound variables, since this will be made explicit by label in edges of the graph.

– Formulas are represented by formula graphs that are composed by formula nodes; thus the definition of formula graphs has three added inductive graphs:

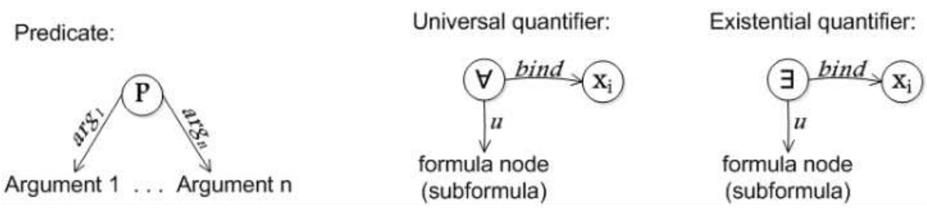


Figure 5.15 shows how bound variables appear in formula graphs; they are to be shared as much as possible. Like hypothesis (linked to D-node with label H) may only be discharged once, variables may not be bound by more than one quantifier. Also, quantifiers may not bind variables outside their direct subformula, their scope.

In mimp-fol, rule nodes operate conveniently on the root node (primary connective) of a formula-graph. Since quantifier rules of mimp-fol affect variables and we are sharing subformulas and since variables are different before and after substitution, we cannot share one formula graph as the "before and

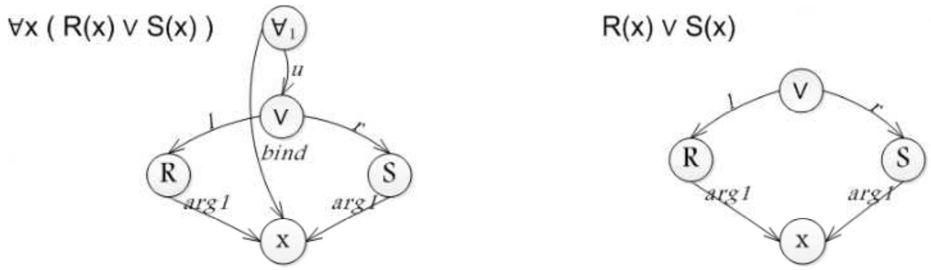


Figure 5.15: Bound and free variables in formula graphs.

after" of a substitution. Instead, two subgraphs are required that are different in the variable that is substituted for. The substitution explicit by an edge labeled *subs.by* (see Definition 28). In the following example we can identify two formula graphs as the premise and conclusion of a substitution, P and $P[x/t]$.

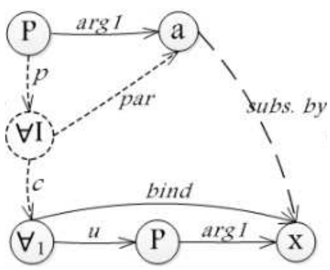


Figure 5.16: Premise and conclusion of a substitution.

Now we have the set of rules added in the extension:

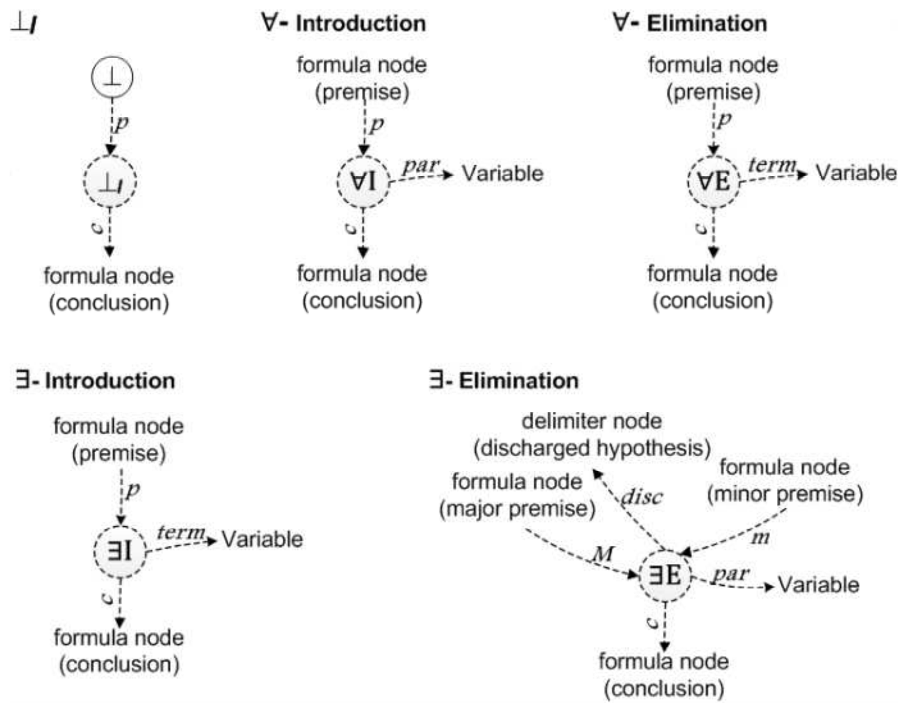
Definition 28 *A* mimp-fol *G is a directed graph* ⟨V, E, L⟩ *where:* V *is a set of nodes,* L *is a set of labels,* E *is a set of* edges ⟨$v \in$ V, $t \in$ L, $v' \in$ V⟩, *where v is the source and v' the target.*

The mimp-fol is defined recursively as follows:

pmimp *Every construction rule for propositional mimp-graphs (Definition 23) is a construction rule for mimp-fol.*

∀I *If G_1 is a mimp-fol, containing a node: α_m linked to the D-node C, then the graph G is defined as G_1 with*

1. *the removal of the ingoing edge in the node C;*
2. *an R-node* $\forall\mathrm{I}_i$ *at the top position;*
3. *duplicating: the graph of α_m with the substitution of x for t ($\alpha_m[x/a]$);*
4. *an F-node* $\forall_t$*;*



5. *edges:* $\alpha_m \xrightarrow{p_{new}} \forall\mathrm{I}_i$, $\forall\mathrm{I}_i \xrightarrow{c_{new}} \forall_t$, $\forall\mathrm{I}_i \xrightarrow{par_{new}} t$, $\forall_t \xrightarrow{bind} x$, $\forall_t \xrightarrow{u} \alpha_m[x/a]$, $a \xrightarrow{subs.\ by} x$ *and* $\forall_t \xrightarrow{conc} C$;

is a mimp-fol under the proviso that 'a' does not occur in any variable node of the branch (see Figure 5.17).

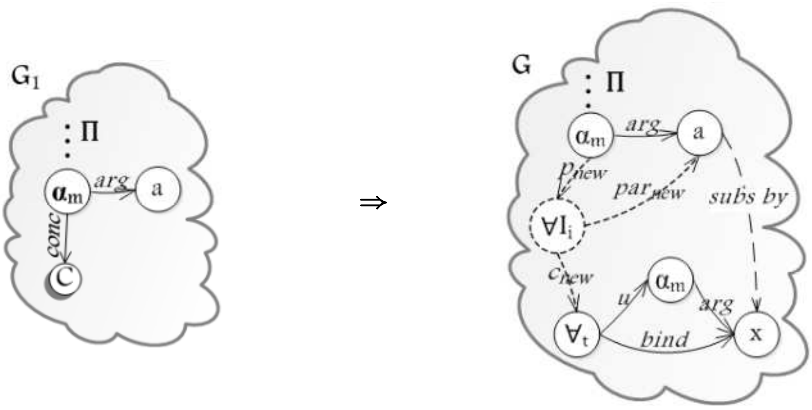


Figure 5.17: ∀-Introduction Rule.

∀E *If* G_1 *is a mimp-fol and contains the edge* $\forall_t \xrightarrow{u} \alpha_m$ *and the node* $\forall_t$ *linked to the delimiter node* C *then the graph* G *is defined as* G_1 *with*

1. *the removal of the ingoing edge in the node* C.
2. *an R-node* $\forall\mathrm{E}_i$ *at the top position;*
3. *duplicating: the graph of* α_m *with the substitution of* x *for* t– ($\alpha_m[t/x]$);

4. *edges:* $\forall_t \xrightarrow{p_{new}} \forall \mathrm{E}_i$, $\forall \mathrm{E}_i \xrightarrow{c_i} \alpha_m[t/x]$, $\forall \mathrm{E}_i \xrightarrow{term_{new}} t$, $x \xrightarrow{subs.\ by} t$ *and* $\alpha_m[t/x] \xrightarrow{conc} C$,

is a mimp-fol (see Figure 5.18).

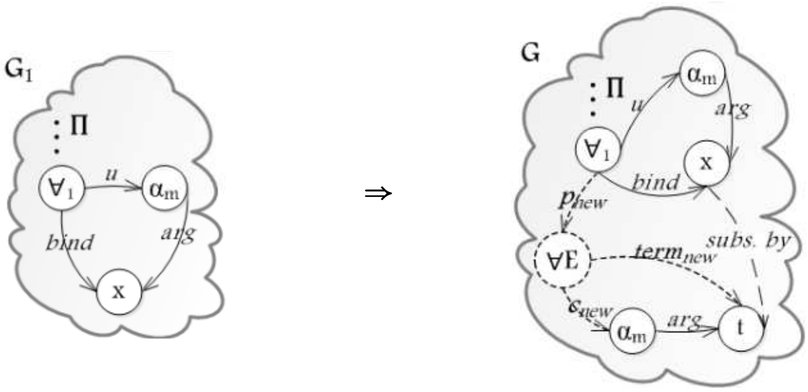


Figure 5.18: $\forall$-Elimination Rule.

∃I *If G_1 is a mimp-fol and contains the node α_m linked to the D-node C then the graph G is defined as G_1 with*

1. *the removal of the ingoing edge in the node C.*
2. *an R-node $\exists \mathrm{I}_i$ at the top position;*
3. *duplicating: the graph of α_m with the substitution of t for x ($\alpha_m[x/t]$);*
4. *edges:* $\alpha_m \xrightarrow{p_{new}} \exists \mathrm{I}_i$, $\exists \mathrm{I}_i \xrightarrow{c_{new}} \exists_t$, $\exists \mathrm{I}_i \xrightarrow{term_{new}} t$, $\exists_t \xrightarrow{bind} x$, $\exists_t \xrightarrow{u} \alpha_m[x/t]$, $t \xrightarrow{subs.\ by} x$ *and* $\exists_t \xrightarrow{conc} C$,

is a mimp-fol (see Figure 5.19).

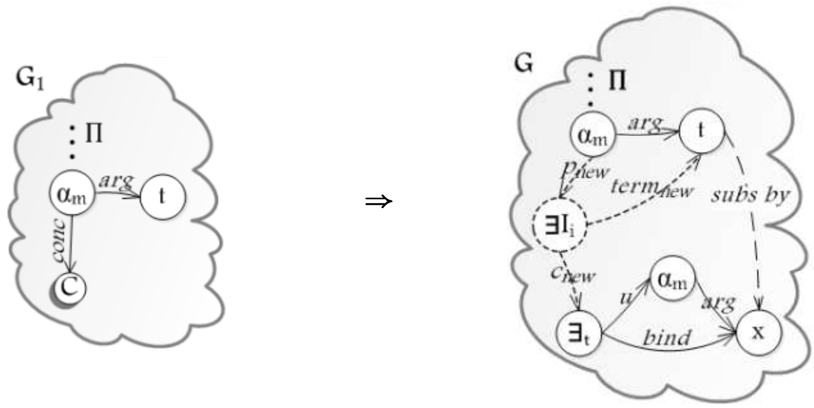


Figure 5.19: $\exists$-Introduction Rule.

$\exists$E *If G_1 and G_2 are mimp-fol, and the graph obtained by $G_1 \oplus G_2$ (intermediate step) contains nodes: $\exists_t$ and σ_r linked to the D-node C; and α_m is linked to $\exists_t$ by $\exists_t \xrightarrow{u} \alpha_m$ and α_m is linked to D-node H_u, then the graph G is defined as $G_1 \oplus G_2$ with*

1. *the removal of the ingoing edges in the node C which were generated in the intermediate step ($G_1 \oplus G_2$);*
2. *an R-node $\exists \mathrm{E}_i$ at the top position;*
3. *edges: $\sigma_r \xrightarrow{m_{new}} \exists \mathrm{E}_i$, $\exists_t \xrightarrow{M_{new}} \exists \mathrm{E}_i$, $\exists \mathrm{E}_i \xrightarrow{par_{new}} a$, $\exists \mathrm{E}_i \xrightarrow{c_{new}} \sigma_r$, $\exists \mathrm{E}_i \xrightarrow{disc_{new}} H_u$, $x \xrightarrow{subs.\ by} a$ and $\sigma_r \xrightarrow{conc} C$;*

is a mimp-fol (see Figure 5.20).

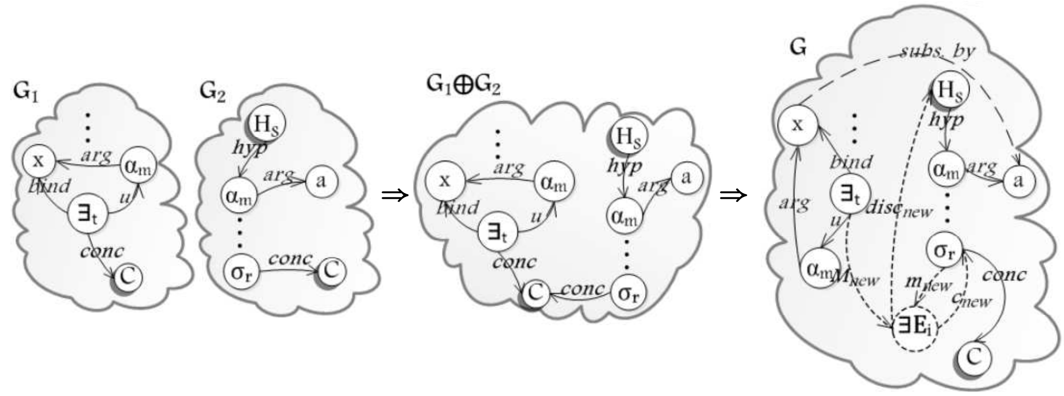


Figure 5.20: $\exists$-Elimination Rule.

$\bot$I *If G_1 is a mimp-fol and contains the node $\bot$ linked to the D-node C then the graph G is defined as G_1 with*

1. *the removal of the ingoing edge in the node C.*
2. *an R-node $\bot I_i$ at the top position;*
3. *edges: $\bot I_i \xrightarrow{c_{new}} \alpha_m$, $\bot \xrightarrow{p_{new}} \bot I_i$ and $\alpha_m \xrightarrow{conc} C$,*

is a mimp-fol.

5.3.2 Examples

By means of these examples we shall first of all show how our graphs represent deductions. Figure 5.21 shows a small proof with quantifiers, where the *term*-edge indicates the replaced term t in the inference scheme $\exists$I that has the conclusion $\exists y P(y)$ and the premise $P(y)[t/y] = P(t)$. The substitution is indicated by means of the *subs*-edge.

$$\dfrac{\exists x P(x) \qquad \dfrac{\dfrac{\dfrac{[P(t)]^1}{\exists y P(y)}\,\exists\text{I} \qquad B}{\exists y P(y) \wedge B}\,\wedge\text{I}}{\exists y P(y)}\,\wedge\text{E}}{\exists y P(y)}\,\exists\text{E},1 \quad \Rightarrow$$

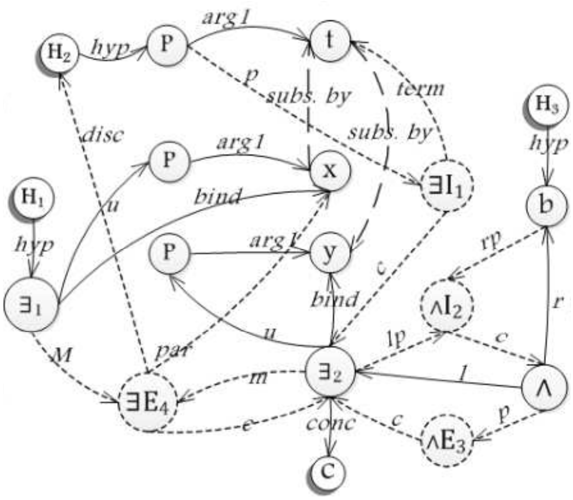


Figure 5.21: Example in proof-graphs for FOL

$$\dfrac{[\exists x \forall y F(x,y)]^1 \qquad \dfrac{\dfrac{\dfrac{[\forall y F(t_1,y)]^2}{F(t_1,t_2)}\,\forall\text{E}}{\exists x F(x_1,t_2)}\,\exists\text{I}}{\forall y_1 \exists x_1 F(x_1,y_1)}\,\forall\text{I}}{\dfrac{\forall y_1 \exists x_1 F(x_1,y_1)}{\exists x \forall y F(x,y) \rightarrow \forall y_1 \exists x_1 F(x_1,y_1)}\,\rightarrow\text{I}_2}\,\exists\text{E}_1$$

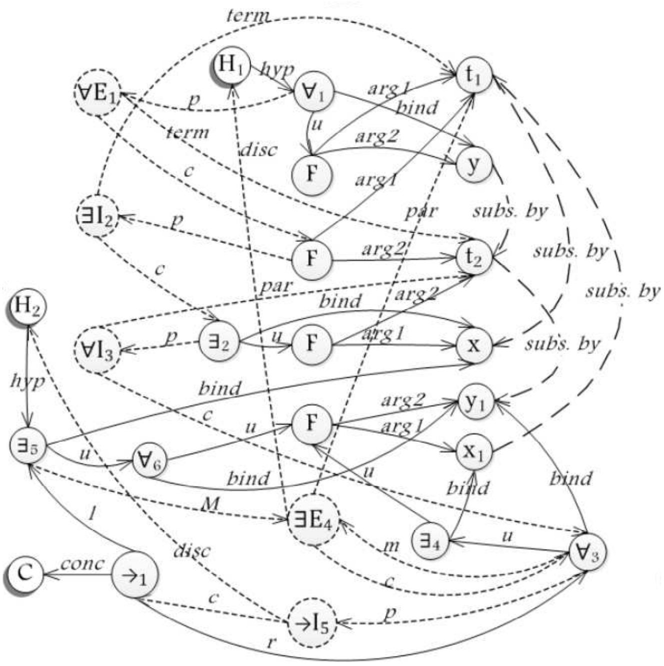
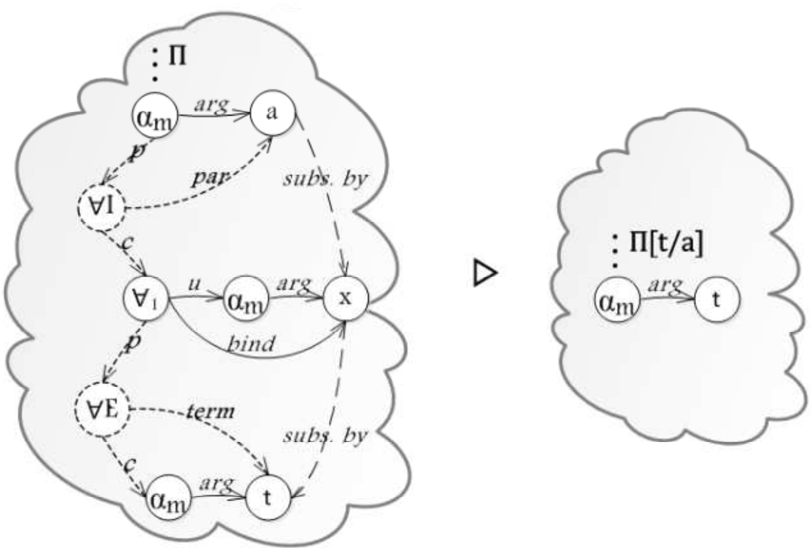


Figure 5.22: Example in mimp-graphs for FOL

5.3.3
Reductions for mimp-fol

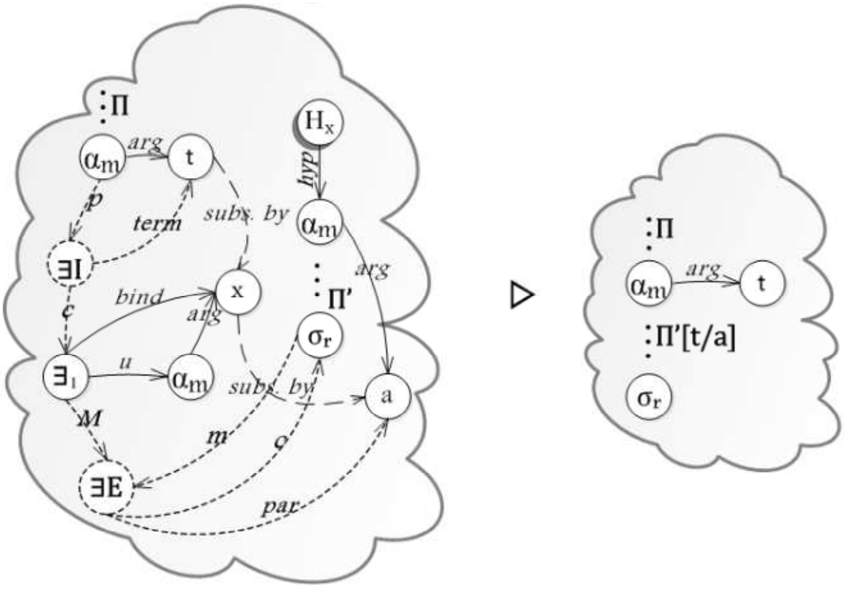
To obtain a normal derivation from any deduction, we transform it step by step, until no elimination rule is below an introduction rule. This process is called normalization, thus we give a set of transformations for first order logic proofs which preserve the information content of the original proof. We emphasize that for previous schemes of reduction (propositional logic), conclusions of maximal formulas remain shared, but now we have reduction schemes with added conclusions, we can compare it to the reduction scheme for natural deduction in Definition 3. Note that $\Pi[t/a]$ represents the resulting graph of replacing the label a on variable nodes by the label t.

Elimination of ∀I followed by ∀E In this reduction step are only preserved nodes and edges in the graph represented by Π, the formula graph α_m, the edge $\alpha_m \xrightarrow{arg} a$, the variable node with label a and the remaining graph represented by a cloud, then this label a on variable nodes is replaced by $t(\Pi[t/a])$.



Elimination of ∃I followed by ∃E Now, we preserve the graph represented by Π, the formula graph α_m, the edge $\alpha_m \xrightarrow{arg} t$, the variable node with label t, the graph Π', the formula graph σ_r, then the label a on variable nodes is replaced by t $(\Pi'[t/a])$.

then this label a on variable nodes is replaced by t.


Π
αm
arg
t
p
term
subs. by
Hx
hyp
αm
arg
∃I
bind
x
Π'
c
arg
σr
∃1
u
αm
subs. by
a
M
m
c
∃E
par
▷
Π
αm
arg
t
Π'[t/a]
σr