HAL
archives-ouvertes.fr

Three Dimensional Proofnets for Classical Logic

Clement Houtmann

▶ **To cite this version:**

Clement Houtmann. Three Dimensional Proofnets for Classical Logic. 2009. inria-00426469v2

HAL Id: inria-00426469
https://hal.inria.fr/inria-00426469v2

Preprint submitted on 8 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Three Dimensional Proofnets for Classical Logic

Clément Houtmann

Université Henri Poincaré - Nancy 1 - LORIA*

Abstract

Classical logic and more precisely classical sequent calculi are currently the subject of several studies that aim at providing them with an algorithmic meaning. They are however ruled by an annoying syntactic bureaucracy which is a cause of pathologic non-confluence. An interesting patch consists in representing proofs using proofnets. This leads (at least in the propositional case) to cut-elimination procedures that remain confluent and strongly normalising without using any restricting reduction strategy. In this paper we describe a presentation of sequents in a two-dimensional space as well as a presentation of proofnets and sequent calculus derivations in a three-dimensional space. These renderings admit interesting geometrical properties: sequent occurrences appear as parallel segments in the case of three-dimensional sequent calculus derivations and the De Morgan duality is expressed by the fact that negation stands for a ninety degree rotation in the case of two-dimensional sequents and three-dimensional proofnets.

1 Introduction

The witness property is an important advantage of intuitionistic logic over classical logic. Indeed witnesses of existential statements can always be obtained when the statement is proved in the former. Nevertheless and from the programming point of view, constructivity is only required for Σ_1^0-statements for which Friedman demonstrated that classical and intuitionistic provability coincide [Fri78]. Initiated by Griffin in 1990, studies on the algorithmic meaning of classical constructions is today an active and successful field of research. Griffin first demonstrated that control operators can be typed by Pierce's law [Gri90]. This foundational work was then followed by a range of studies about computational interpretations of classical logic as well as sequent calculus. While beta-reduction is usually considered as *the* cut-elimination procedure for natural deduction, a wide range of cut-elimination procedures can be imagined for sequent calculus [Gen35, CH00, UB01, BL08]. Strong normalisation or confluence may not hold for these reduction procedures. Yet they lead to extractions of programs from classical proofs, as demonstrated for instance by [Miq09].

By several aspects, proofnets appear as a good solution to the problem of representing proofs in classical logic. Indeed they contain and

*UMR 7503 CNRS-INPL-INRIA-Nancy2-UHP

organize all the necessary information that a proof must include without getting tangled in the bureaucracy of syntax. Proofnets originate from Girard's Linear Logic [Gir87] and elegantly adapt to classical logic [Mil87, Rob03, LS05]. In particular Lamarche and Straßburger demonstrate that confluence and strong normalisation of the unrestricted cut-elimination procedure can be recovered in the propositional case [LS05]. As usual with classical logic, problems arise when adding quantifiers, as shown by the ongoing work of McKinley and Heijltjes [McK09, Hei09].

The various existing approaches to proofnets for classical logic significantly differ by the criterion that they adopt to characterize *correct* proofnets. For example Robinson [Rob03] uses switchings *à la* Danos-Regnier whereas Lamarche and Straßburger use conjunctive resolutions [LS05]. This latter criterion nicely relates to sequent calculus proofs in the following way: when considering a proofnet associated with some sequent S, conjunctive resolutions exactly correspond to the atomic sequents that one could obtain from S by decomposing all its connectives using sequent calculus inferences. Therefore the criterion states that if all these conjunctive resolutions are actually proved, then the proofnet is correct. It corresponds for instance to the tableaux method criterion *if all branches of a tableau are closed, the formula represented by the tableau is unsatisfiable*. The advantage of this last criterion is that branches of a tableau as well as branches of a sequent calculus derivation are easily extractable. On the contrary, the extraction of subnets using conjunctive resolutions is not trivial: a proofnet may have an exponantial number of conjunctive resolutions whereas a derivation always have a linear number of branches. These complexities reflect the fact that proofnets are compressed derivations. In order to check their value as a certificate, their information must be decoded through conjunctive resolutions.

In this paper, we propose a rendering of proofnets in a three dimensional space. This rendering is meant to ease the extraction of subnets using conjunctive resolutions. It is based on the De Morgan duality: the (horizontal) axis which is usually employed in order to write sequents is split into an axis for conjunctions and an axis for disjunctions. Sequents and proofnets therefore become respectively two-dimensional and three-dimensional objects. Conjunctive resolutions appear as paths and surfaces which follow to the disjunction axis. Checking the validity of a proofnet therefore becomes *geometrical*. As far as we know, our idea of using the De Morgan duality to represent proofnets and sequent calculus derivations in a three-dimensional space is completely novel. The only other instance of a rendering of proofs in a three-dimensional space is Yves Guiraud's work which represents SKS derivations as three-dimensional penrose diagrams [Gui06]. This paper is organized as follows. In section 2, we reformulate the definitions of classical proofnets and conjunctive resolutions. Then in section 3, we present successively the rendering of sequents as two-dimensional planar acyclic graphs and the rendering of proofnets and sequent calculus derivations in a three-dimensional space. Finally we present the ongoing implementation of a prototype in section 4.

2 Proofnets in Classical Logic

This section introduces proofnets for classical logic. Our presentation is greatly inspired by the work of Lamarche and Straßburger [LS05]. The

set of *formulæ* is defined by the grammar

$$\varphi, \psi \quad ::= \quad \underbrace{P \mid \bar{P} \mid \top \mid \bot}_{\text{atoms}} \mid \varphi \wedge \psi \mid \varphi \vee \psi \ .$$

Formulæ are built from atomic predicates (P), negated atomic predicates ($\bar{P}$), false ($\bot$), true ($\top$), conjunctions ($\varphi \wedge \psi$) and disjunctions ($\varphi \vee \psi$). The *negation* $\neg\varphi$ of a formula φ is defined inductively by

$$\neg P = \bar{P} \ , \qquad \neg\bar{P} = P \ , \qquad \neg\bot = \top \ , \qquad \neg\top = \bot \ ,$$
$$\neg(\varphi \wedge \psi) = (\neg\varphi) \vee (\neg\psi) \ , \qquad \neg(\varphi \vee \psi) = (\neg\varphi) \wedge (\neg\psi) \ .$$

We suppose given an infinite countable set of names. Symbols x, y, z will range over this set of names. A named formula is a pair containing a name x and a formula φ and denoted $x : \varphi$. A *sequent* L is a set of named formulæ such that to any name x corresponds at most one formula φ such that $x : \varphi \in L$. Such a sequent is denoted $\vdash x_1 : \varphi_1, \ldots, x_n : \varphi_n$ or $\vdash \varphi_1, \ldots, \varphi_n$ when names are insignificant. A *position* is a finite binary sequence. The empty position will be denoted ε. If φ is a formula and l is a position, the subformula of φ occurring at position l, denoted $\varphi_{|l}$, is defined by

$$P_{|\varepsilon} = P \ , \qquad \bar{P}_{|\varepsilon} = \bar{P} \ , \qquad \bot_{|\varepsilon} = \bot \ , \qquad \top_{|\varepsilon} = \top \ ,$$
$$(\psi_0 \wedge \psi_1)_{|i\cdot l} = \psi_{i|l} \ , \qquad (\psi_0 \vee \psi_1)_{|i\cdot l} = \psi_{i|l} \qquad (\ i \in \{0,1\}\ ) \ .$$

Remark that $\varphi_{|l}$ is not defined for all formulæ and all positions. If φ is a formula, the set of positions of φ, denoted $\mathscr{P}_\varphi$ is the set of positions l such that $\varphi_{|l}$ is actually defined. Let us consider for example the formula $\top \vee (P \wedge \top)$. It admits five positions, namely ε, 0, 1, 10 and 11. They correspond respectively to the subformulæ $\top \vee (P \wedge \top)$, $\top$, $P \wedge \top$, P and $\top$. A *sequent position* is a pair containing a name x and a position l and denoted $x \circ l$. In what follows, we will improperly say *position* instead of *sequent position*. Let L be a sequent. The set of (sequent) positions of L is the set

$$\mathscr{P}_L = \{ \ \ x \circ l \ \ / \ \ x : \varphi \in L \text{ and } l \in \mathscr{P}_\varphi \ \ \} \ .$$

Therefore if $L = \ \vdash x_1 : \varphi_1, \ldots, x_n : \varphi_n$ is a sequent and $x_i \circ l \in \mathscr{P}_L$, the subformula of L occurring at position $x_i \circ l$, denoted $L_{|x_i \circ l}$, is the formula $\varphi_{i|l}$. We are now able to define proofnets.

Definition 1 (Proofnets)**.** *A proofnet is a pair $(L, \sim)$ where L is a sequent and $\sim$ is symmetric relation on $\mathscr{P}_L$ such that whenever $l \sim m$, then either $L_{|l} = \neg L_{|m}$ or $L_{|l} = L_{|m} = \top$.*

Figure 1 depicts a proofnet whose sequent is $\vdash x : \top \vee (P \wedge \top), y : (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)$ and whose relation is the symmetric closure of $\{(x \circ 11, x \circ 11), (x \circ 1, y \circ 01), (y \circ 00, y \circ 010)\}$. Proofnets are meant to represent possibly unfinished derivations in the cut-free classical sequent calculus. In a proofnet $(L, \sim)$, a link $x \circ l \sim y \circ m$ with $\varphi = L_{|x\circ l} = \neg L_{|y\circ m}$ represents the use of the rule *axiom* and a link $l \sim m$ with $L_{|x\circ l} = L_{|y\circ m} = \top$ represents the use of the rule introducing $\top$.

It is obviously important to characterize which proofnets represent completed derivations. For that we reformulate the notion of conjunctive resolution that is introduced by Lamarche and Straßburger [LS05]. The *conjunctive positions* of a sequent L are the positions $x \circ l$ such that $L_{|x\circ l}$ is a conjunction. The set of conjunctive positions of a sequent L is denoted $\mathscr{P}^\wedge_L$. A *conjunctive resolution* for a sequent L is a function $s : \mathscr{P}^\wedge_L \to \mathbb{B}$.

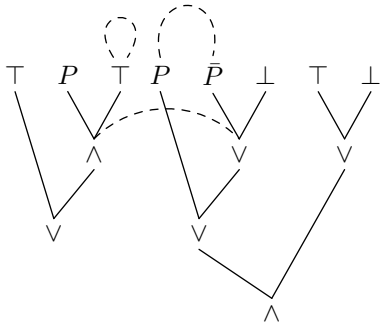


Figure 1: A proofnet for $\vdash x : \top \vee (P \wedge \top), y : (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)$.

If s is a conjunctive resolution for L, the set of positions of s, denoted $\mathscr{P}_L^s$, is the subset of positions $(x \circ l) \in \mathscr{P}_L$ such that for all $(x \circ m) \in \mathscr{P}_L^\wedge$, if m is a strict prefix of l, then the binary sequence $m \cdot s(m)$ is also a prefix of l.

Definition 2 (Valid Proofnets)**.** *A proofnet $(L, \sim)$ is valid if for all conjunctive resolution s of L, the restriction of $\sim$ to $\mathscr{P}_L^s$ (that is to say the set $\sim \cap(\mathscr{P}_L^s \times \mathscr{P}_L^s)$) is not empty.*

Let us consider for example the proofnet depicted in figure 1. Its conjunctive positions are $x \circ 1$ and y. Therefore the sequent corresponding to this proofnet admits four conjunctive resolutions which are

$$\left\{ \begin{array}{lcl} x \circ 1 & \mapsto & 0 \\ y & \mapsto & 0 \end{array} \right. , \qquad \left\{ \begin{array}{lcl} x \circ 1 & \mapsto & 1 \\ y & \mapsto & 0 \end{array} \right. ,$$

$$\left\{ \begin{array}{lcl} x \circ 1 & \mapsto & 0 \\ y & \mapsto & 1 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{lcl} x \circ 1 & \mapsto & 1 \\ y & \mapsto & 1 \end{array} \right. .$$

These conjunctive resolutions correspond to the positions of the subtrees depicted in figure 2. This figure also represents the restriction of the relation associated with each conjunctive resolution. Notice that the third conjunctive resolution returns an empty restriction of the proofnet relation. Consequently the proofnet of figure 1 is not a valid proofnet. This can easily be corrected though, for example by linking the position $x \circ 0$ to itself.

In order to represent proofnets with cuts, Lamarche and Straßburger use a new binary connective [LS05]. This connective is in fact just a conjunction (and is indeed treated as a conjunction), consequently replacing the cut rule

$$\textsc{Cut}\ \frac{\vdash \Gamma, \varphi, \Delta \qquad \vdash \Gamma, \neg\varphi, \Delta}{\vdash \Gamma, \Delta} \qquad \text{by the rule} \qquad \textsc{Cut}',\ \frac{\vdash \Gamma, \varphi \wedge (\neg\varphi), \Delta}{\vdash \Gamma, \Delta}\ .$$

Therefore cut-formulæ are special conjunctions that are denoted $\varphi \sqcap \psi$ instead of $\varphi \wedge \psi$. They cannot occur inside another (cut)-formula: in a sequent $L = \ \vdash x_1 : \varphi_1, \ldots, x_n : \varphi_n$, the only positions corresponding to cut-formulæ are of the form $x_i \circ \varepsilon$. In addition, if $(L, \sim)$ is a proofnet and $x_i \circ \varepsilon$ is a position corresponding to a cut-formula, then for all $l \in \mathscr{P}_L$, $(x_i \circ \varepsilon) \not\sim l$. Cut-formulæ are handled in the same way as conjunctions with respect to conjunctive resolutions. If L is a sequent containing no

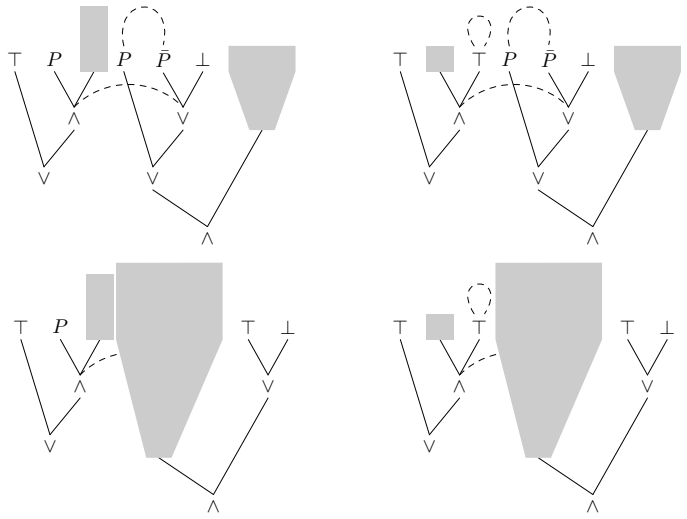


Figure 2: Conjunctive resolutions for the proofnet in figure 1

cut-formula and L' is a sequent only containing cut-formulæ, then any proofnet $(L \cup L', \sim)$ will be said to be a *proofnet for L with cuts L'*. If L' is empty, the proofnet is *cut-free*.

Valid proofnets are sound and complete with respect to the classical cut-free sequent calculus and valid proofnets with cuts are sound and complete with respect to the classical sequent calculus with cuts as demonstrated by the following theorem.

Theorem 1. *For some sequent L, there exists a valid proofnet for L if and only if there exists a derivation of L in the cut-free classical sequent calculus. For some sequent L, there exists a sequent L' as well as a valid proofnet for the sequent L with cuts L' if and only if there exists a derivation of L in the classical sequent calculus with cuts.*

Proof. Adaptation of the proof available in [LS05]. □

We will not formally introduce a cut-elimination procedure for classical proofnets in this paper. One is described by Lamarche and Straßburger in [LS05] for their presentation of proofnets. Let us remark that their $\mathbb{B}$-nets are a subclass of our valid proofnets: they just restrict the use of links $l \sim m$ to positions l and m corresponding to atoms. In addition, their cut-elimination procedure does not use any restricting strategy and remains confluent and strongly normalising on $\mathbb{B}$-nets. Therefore we believe that it could be extended to an unrestricted confluent and strongly normalising cut-elimination procedure for our presentation of proofnets. However we leave the formalisation of such a procedure for a subsequent paper.

3 Third Dimension

Before representing proofnets in a three-dimensional space, we represent sequents in a two-dimensional space. This is done using planar acyclic directed graphs having exactly one source and one sink. We first define two auxiliary operations on such graphs. If A and B are two acyclic

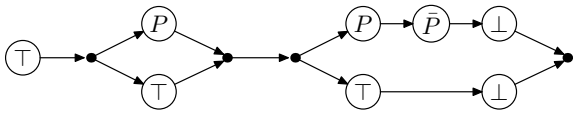


Figure 3: $\langle \vdash \top \vee (P \wedge \top), (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot) \rangle$

directed graphs, the graph $A|B$ is the acyclic directed graph obtained by linking the sink of A with the source of B; the graph $\frac{A}{B}$ is the graph obtained by adding a fresh source linked to the sources of A and B and a fresh sink to whom the sinks of A and B are linked. These operations preserve both properties of being acyclic and planar.

Now let us consider the list of atoms that occur in a sequent. Such a list is obviously a one-dimensional object that does not render the logical structure of the sequent. This logical structure is inherited from the connectives of the sequent and is translated into conjunctive resolutions that select sublists of the initial list of atoms. Then in order to prove the sequent, a link must be found for each sublist. We propose to replace the one-dimensional list of atoms by a two-dimensional object representing the whole sequent and therefore containing the logical structure which corresponds to the conjunctive resolutions: the rendering of a formula φ, denoted $\langle \varphi \rangle$, is the acyclic directed planar graph with exactly one source and one sink defined as

- a graph with a single node labelled by φ if φ is an atom ;
- $\langle \psi_1 \rangle | \langle \psi_2 \rangle$ if $\varphi = \psi_1 \vee \psi_2$;
- $\frac{\langle \psi_1 \rangle}{\langle \psi_2 \rangle}$ if $\varphi = \psi_1 \wedge \psi_2$.

The graph $\langle \vdash \varphi_1, \ldots, \varphi_n \rangle$ associated with a sequent $\vdash \varphi_1, \ldots, \varphi_n$ is the graph $\langle \varphi_1 \rangle | \ldots | \langle \varphi_n \rangle$. For example, the graph $\langle \vdash \top \vee (P \wedge \top), (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot) \rangle$ is represented in figure 3. Now let L be some sequent. Each directed path from the source of $\langle L \rangle$ to its sink corresponds to a sequent $\vdash a_1, \ldots, a_n$ where the a_i are the labels of the nodes of the path. These are exactly the sequents that one would obtain by decomposing all the connectives of L using sequent calculus rules. These also are precisely the sequents that appear in the subnets corresponding to each conjunctive resolution of L. In particular it holds for the graph in figure 3 and the conjunctive resolutions in figure 2: in this case, the sequents are

$$\vdash \top, P, P, \bar{P}, \bot \ , \qquad \vdash \top, \top, P, \bar{P}, \bot \ ,$$
$$\vdash \top, P, \top, \bot \quad \text{and} \quad \vdash \top, \top, \top, \bot \ .$$

For any sequent L, the graph $\langle L \rangle$ is meant to be rendered on the plane. If we choose the horizontal axis to represent disjunctions (following the tradition of writing sequent $\vdash \varphi_1, \ldots, \varphi_n$ *horizontally*) and the vertical axis to represent conjunctions. A graph $\langle \varphi \vee \psi \rangle = \langle \varphi \rangle | \langle \psi \rangle$ is then rendered as the horizontal sequence $\langle \varphi \rangle \to \langle \psi \rangle$ and a graph $\langle \varphi \wedge \psi \rangle = \frac{\langle \varphi \rangle}{\langle \psi \rangle}$ is rendered as the vertical fork $\cdot \begin{smallmatrix} \nearrow \langle \varphi \rangle \searrow \\ \searrow \langle \psi \rangle \nearrow \end{smallmatrix} \cdot$. This rendering behaves quite well with respect to negation which then stands for an exchange of the disjunction axis and the conjunction axis (for example a ninety degree rotation): the graph associated with the negation of the sequent $\vdash \top \vee (P \wedge \top), (P \vee (\bar{P} \vee$

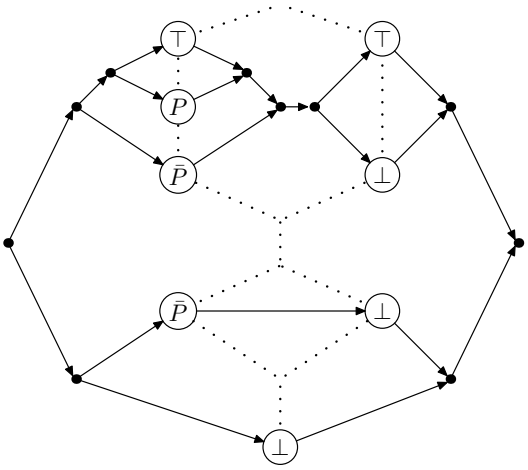
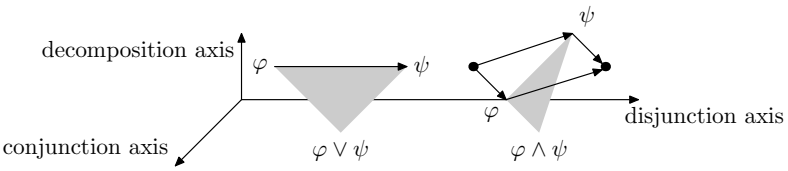


Figure 4: Negation of the graph in figure 3

$\bot)) \wedge (\top \vee \bot)$ is rendered in figure 4, on top of a dashed rotated version of the graph of figure 3.

We are now able to render sequents in a two-dimensional space. This rendering is a recursive decomposition of the connectives in the sequent. We can render this iteration with respect to a third axis called *decomposition axis* which is added to the *conjunction axis* and the *disjunction axis*. Disjunctions and conjunctions are rendered in the following way.



The rendering of a formula consists in the recursive rendering of its connectives, therefore obtaining a three-dimensional tree. The rendering of a whole sequent consists in the rendering of each of its formulæ side by side along the disjunction axis. Each step of such a rendering corresponds to a partially computed graph. Each decomposition of a connective corresponds to a computation further towards the final graph corresponding to the initial sequent. The final three-dimensional forest displays on top of it the completed graph corresponding to the initial sequent. For example, figure 5 displays the three-dimensional proofnet associated with the sequent $\vdash \top \vee (P \wedge \top), (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)$. It also displays several graphs corresponding to partial decompositions of its connectives, that is to say from bottom to top the graphs for

$$[\top \vee (P \wedge \top)] \mid [(P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)] \ , \quad [\top] \mid [(P \wedge \top)] \mid \frac{[P \vee (\bar{P} \vee \bot)]}{[\top \vee \bot]} \ ,$$

$$[\top] \mid \frac{[P]}{[\top]} \mid \frac{[P] \mid [\bar{P} \vee \bot]}{[\top] \mid [\bot]} \quad \text{and} \quad [\top] \mid \frac{[P]}{[\top]} \mid \frac{[P] \mid [\bar{P}] \mid [\bot]}{[\top] \mid [\bot]} \ .$$

Note that we could also have represented oblique graphs such as the one corresponding to the decomposition step $[\top] \mid \frac{[P]}{[\top]} \mid \frac{[P \vee (\bar{P} \vee \bot)]}{[\top \vee \bot]}$.

We have not dealt yet with the rendering of the relation $\sim$ of a proofnet $(L, \sim)$. If l and m are positions of L such that $l \sim m$, both positions l

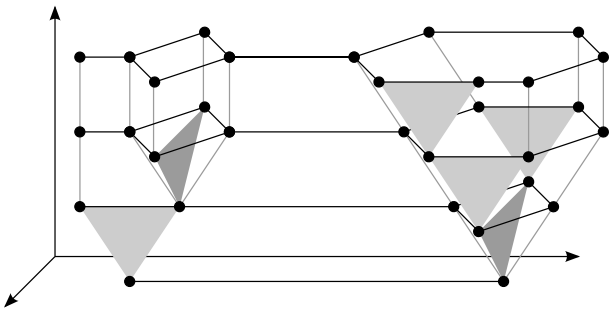


Figure 5: A three-dimensional proofnet

and m appear as vertices and the link $l \sim m$ is rendered as a curve between these vertices corresponding to l and m. Therefore the relation $\sim$ is rendered as a set of curves between vertices of the three-dimensional proofnet. Conjunctive resolutions correspond to maximal paths in the graph $\langle L \rangle$. Similarly they correspond to the selection of certain polygons and vertices of the rendered proofnet. If each selection contains a pair of linked vertices, then we are looking at a valid proofnet.

We have seen that in terms of two-dimensional graphs, negation stands for the exchange of the disjunction axis and the conjunction axis. It also remarkably holds for three-dimensional proofnets: the vertices corresponding to formulæ as well as the triangles corresponding to decompositions of connectives are unchanged. However all the parts of the three-dimensional proofnet only corresponding to the directed graphs have to be switched as done in figure 4.

If L is a sequent, each directed path from the source of $\langle L \rangle$ to its sink correspond to an atomic sequent. The list of such paths (or equivalently such sequents) is an expanded version of the graph $\langle L \rangle$. Let us remark that the size of the list of paths can be exponantial with respect to the size of the initial graph (think of a sequent $\vdash \varphi_1 \wedge \psi_1, \varphi_2 \wedge \psi_2, \ldots, \varphi_n \wedge \psi_n$). This expansion corresponds to the expansion that one must perform to transform a proofnet into a sequent calculus derivation. Each two-dimensional graph can be expanded into a two-dimensional list of sequents and each three-dimensional proofnet can be expanded into a three-dimensional sequent calculus derivation. This expansion is also known as *sequentialization*. The difference between proofnets and sequent calculus is that in the latter only one connective is decomposed at a time. The contexts (that is to say the other formulæ of the sequent) are just copied in the conclusion and in the premisses of the inference. Sequentialization therefore stands for transforming one big step containing several simultaneous decompositions into several steps each containing a decomposition and several duplications. These smaller steps correspond either to a sequent calculus introduction of the $\wedge$ connective or to a sequent calculus introduction of the $\vee$ connective. The following picture displays on its left a three-dimensional sequent calculus introduction of a conjunction and on its right a three-dimensional sequent calculus introduction of a disjunction.

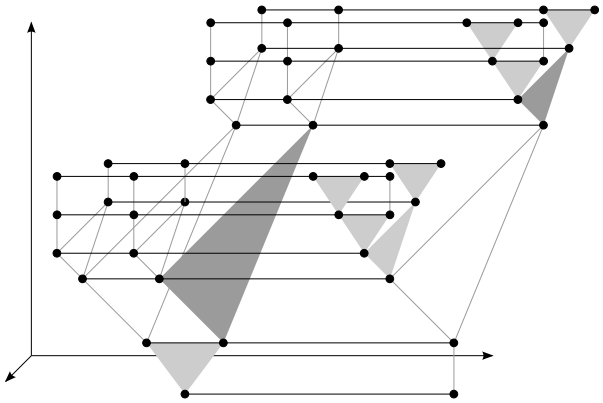
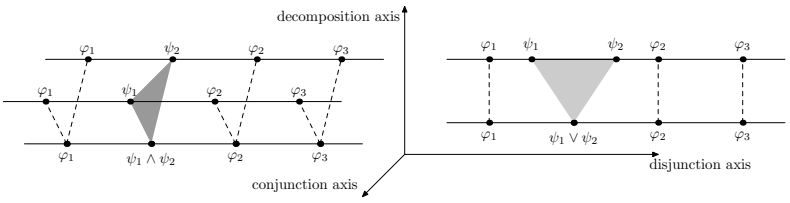


Figure 6: A three-dimensional sequent calculus derivation



In such three-dimensional inferences, a sequent is represented by a segment parallel to the disjunction axis. From the bottom-up point of view, on the left the decomposition of a segment $\dots \varphi_1, \psi_1 \wedge \psi_2, \varphi_2, \varphi_3 \dots$ creates two new segments $\dots \varphi_1, \psi_1, \varphi_2, \varphi_3 \dots$ and $\dots \varphi_1, \psi_2, \varphi_2, \varphi_3 \dots$; on the right the decomposition of a segment $\dots \varphi_1, \psi_1 \vee \psi_2, \varphi_2, \varphi_3 \dots$ creates one new segment $\dots \varphi_1, \psi_1, \psi_2, \varphi_2, \varphi_3 \dots$. The duplication of contexts is represented by the dashed lines. Using these three-dimensional inferences, we can build three-dimensional sequent calculus derivations. For example the three-dimensional derivation corresponding to

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \top, P, P, \bar{P}, \bot}{\vdash \top, P, P, (\bar{P} \vee \bot)}}{\vdash \top, P, (P \vee (\bar{P} \vee \bot))} \quad \dfrac{\vdash \top, P, \top, \bot}{\vdash \top, P, (\top \vee \bot)}}{\vdash \top, P, (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)} \quad \dfrac{\dfrac{\dfrac{\vdash \top, \top, P, \bar{P}, \bot}{\vdash \top, \top, P, (\bar{P} \vee \bot)}}{\vdash \top, \top, (P \vee (\bar{P} \vee \bot))} \quad \dfrac{\vdash \top, \top, \top, \bot}{\vdash \top, \top, (\top \vee \bot)}}{\vdash \top, \top, (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)}}{\dfrac{\vdash \top, P \wedge \top, (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)}{\vdash \top \vee (P \wedge \top), (P \vee (\bar{P} \vee \bot)) \wedge (\top \vee \bot)}}$$

is displayed in figure 6. Let us remark that between the sequent calculus derivation and its three-dimensional rendering, there is a one-to-one correspondence between vertices and formulæ occurrences as well as sequents occurrences and segments parallel to the disjunction axis. The same three-dimensional derivation can be obtained by sequentializing the three-dimensional proofnet of figure 5. Sequentialization mainly consists in the choice of an order for the decompositions of the connectives of the initial sequent. For example if a sequent contains both formulæ $\varphi_1 \wedge \varphi_2$ and $\psi_1 \vee \psi_2$, one can either choose to decompose the conjunction or the

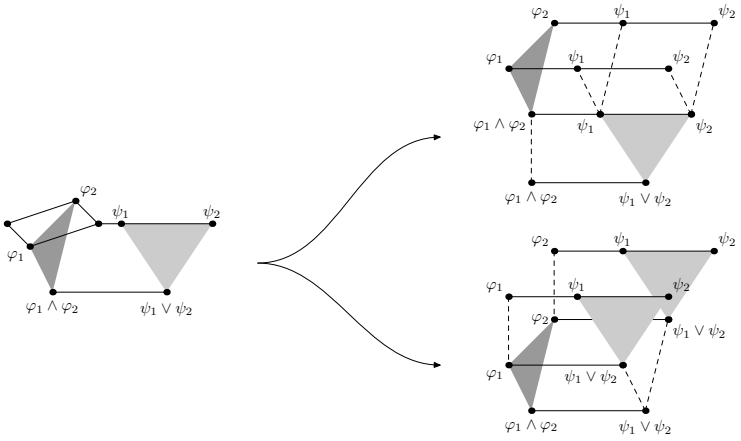


Figure 7: Sequentialization steps

conjunction first. It respectively leads to the sequent calculus derivations

$$\dfrac{\dfrac{\vdash \varphi_1, \psi_1, \psi_2 \qquad \vdash \varphi_2, \psi_1, \psi_2}{\vdash \varphi_1 \wedge \varphi_2, \psi_1, \psi_2}}{\vdash \varphi_1 \wedge \varphi_2, \psi_1 \vee \psi_2} \quad \text{and} \quad \dfrac{\dfrac{\vdash \varphi_1, \psi_1, \psi_2}{\vdash \varphi_1, \psi_1 \vee \psi_2} \qquad \dfrac{\vdash \varphi_2, \psi_1, \psi_2}{\vdash \varphi_2, \psi_1 \vee \psi_2}}{\vdash \varphi_1 \wedge \varphi_2, \psi_1 \vee \psi_2} \; .$$

The sequentialization of a three-dimensional proofnet is therefore not unique: the precise critical pair underlined by these two sequent calculus derivations is translated into sequentialization steps from three-dimensional proofnets to three-dimensional sequents in figure 7. Note that the sequents that appear on top of each sequent derivations are exactly those on top of each three-dimensional derivation and are also precisely the paths of the directed graph on top of the three-dimensional proofnet. This illustrates perfectly Kleene's well-known result [Kle52] which states that the order for decomposing propositional connectives is irrelevant in sequent calculus.

4 Prototype

We are currently developping a prototype for rendering derivations and proofnets using 3D computer graphics. It will include several features enhanced by a point-and-click interface based on the three-dimensional rendering. In particular we will implement methods for checking derivations and proofnets as well as procedures for automatic and interactive proof construction. Sequentialization will also be implemented. For this prototype, we have chosen the tom language[1] [BBK+07] which provides powerful associative rewriting capabilities and strategic programmation on top of java. It is especially designed to manipulate tree structures with the following advantages. First of all, the expressiveness of the language allows for clean and short code. This is particularly important because we wish to implement a proofchecking kernel. The smaller the proofchecker is, the easier it is to convince everyone that it is actually sound. Another interesting feature of tom is the expression of tacticals by strategies.

[1] `tom.loria.fr`

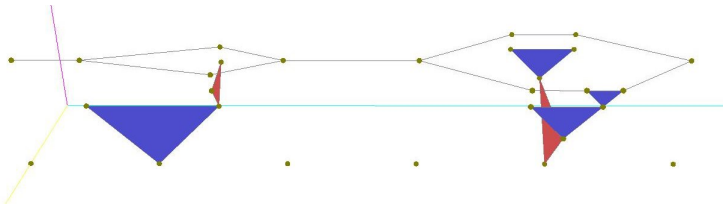


Figure 8: A 3d proofnet rendered by an early version of our prototype

The tom strategy language is directly inspired from early research on elan [VB98] and ρ-calculus and allows to compose basic strategies to express complex programs using strategy combinators. We therefore believe that tom is greatly adapted to the implementation of automatic and interactive proof-search strategies. Finally since tom works on top of java, implementing our prototype in tom allows us to access any java library designed to render three-dimensional objects such as OpenGL implementations. A picture produced by an early version of our prototyped are displayed in figure 8.

5 Conclusion

We have presented a two-dimensional rendering of formulæ and sequents through planar acyclic directed graphs. Such a graph contains the whole logical structure of the corresponding sequent: conjunctive resolutions exactly match the maximal paths of the graph (also known as the paths from the source to the sink). Then we have presented a three-dimensional rendering of proofnets which depicts the iterative construction of a graph from a given sequent. The two-dimensional graphs as well as the three-dimensional proofnets are constructed on top of the De Morgan duality between conjunctions and disjunctions. Indeed the two-dimensional space that is used to render formulæ and sequents is defined by a *conjunction axis* and a *disjunction axis* respectively designed to represent conjunctions and disjunctions. Interestingly negation then stands for an exchange of these two axis and does not modify further the placement of the nodes in a graph neither the structure of a proofnet. We also have presented a three-dimensional rendering of sequent calculus derivations. It is based on the same duality between the conjunction axis and the disjunction axis and sequents are displayed as segments along this latter axis. This rendering of derivations does not behave so well with respect to negation. However its main advantage is the one of sequent calculus derivations over proofnets: the former stand for a fully expanded certificate. Indeed proofnets can be unfolded into sequent calculus derivations through a sequentialization process. We have shown how this process can be rendered into a three-dimensional sequentialization transforming three-dimensional proofnets into three-dimensional derivations. Finally we have presented the current state of the implementation of our prototype for a prover based on this three-dimensional rendering of proofnets and sequent calculus derivations. We also have explained our choice of tom on top of java for an implementation language.

This work opens several perspectives. First of all and without thinking about three-dimensional rendering, the proofnet approach remains fairly

unexplored, especially with respect to full predicate logic with quantifiers. In particular the question of the existence of a confluent and strongly normalising cut-elimination procedure for proofnets *with quantifiers* remains unanswered (as far as we know). Let us also remark that we leave the treatment of cut-elimination for a subsequent paper. These developments will then lead to developments with respect to our three-dimensional rendering. Indeed rendering proofnets with quantifiers as well as rendering cut-elimination are the next extensions that we wish to study. These extensions should also be implemented along all the implementation objectives listed in section 4.

References

[BBK+07] Emilie Balland, Paul Brauner, Radu Kopetz, Pierre-Etienne Moreau, and Antoine Reilles. Tom: Piggybacking rewriting on java. In *RTA*, pages 36–47, 2007.

[BL08] Steffen van Bakel and Pierre Lescanne. Computation with classical sequents. *Mathematical Structures in Computer Science*, 18(3):555–609, 2008.

[CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *ICFP*, pages 233–243, 2000.

[Fri78] Harvey Friedman. Classically and intuitionistically provably recursive functions. In *Higher set theory (Proc. Conf., Math. Forschungsinst., Oberwolfach, 1977)*, volume 669 of *Lecture Notes in Math.*, pages 21–27. Springer, Berlin, 1978.

[Gen35] Gehrard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The collected papers of Gerhard Gentzen*, pages 68–131. North Holland, 1935.

[Gir87] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[Gri90] Timothy G. Griffin. A formulae-as-types notion of control. In *POPL*, pages 47–58. ACM Press, 1990.

[Gui06] Yves Guiraud. The three dimensions of proofs. *Ann. Pure Appl. Logic*, 141(1-2):266–295, 2006.

[Hei09] Willem Heijltjes. Proof forests with cut based on Herbrand's theorem. available on the author's webpage, 2009.

[Kle52] Stephen C. Kleene. Permutability of inferences in gentzen's calculi lk and lj. *Memoires of the American Mathematical Society*, 10:1–26, 1952.

[LS05] François Lamarche and Lutz Straßburger. Naming proofs in classical propositional logic. In *TLCA*, pages 246–261, 2005.

[McK09] Richard McKinley. The alpha-epsilon calculus. In *Proceedings of Workshop on Structures and Deduction*, 2009.

[Mil87] Dale Miller. A compact representation of proofs. *Studia Logica*, 46(4):347–370, 1987.

[Miq09] Alexandre Miquel. Relating classical realizability and negative translation for existential witness extraction. In *TLCA*, pages 188–202, 2009.

[Rob03] Edmund Robinson. Proof nets for classical logic. *J. Log. Comput.*, 13(5):777–797, 2003.

[UB01] Christian Urban and Gavin M. Bierman. Strong normalisation of cut-elimination in classical logic. *Fundam. Inform.*, 45(1-2):123–155, 2001.

[VB98] Eelco Visser and Zine-el-Abidine Benaissa. A core language for rewriting. In C. Kirchner and H. Kirchner, editors, *Proceedings of WRLA*, volume 15 of *ENTCS*. Elsevier, sep 1998.