Frobenius Proof Nets and Resource Semantics for Classical Logic

Novak Novaković2,1
INRIA Saclay - Île de France, Campus de l’École Polytechnique, 91120 Palaiseau, France

François Lamarche1
INRIA Saclay - Île de France, Campus de l’École Polytechnique, 91120 Palaiseau, France

Abstract

The once-intractable problem “is there such a thing as a Boolean category” — a category which is to a Boolean algebra what a cartesian-closed category with coproducts is to a Heyting algebra — has been given several solutions during the last six years or so. But, we still cannot say we have a really satisfying solution.

The cleanest way to represent a category of logical formulas and proofs is by the means of so-called proof nets, where the objects are ordinary formulas but the maps belong to a category $C$ where some logically distinct objects have been identified (e.g., conjunction and disjunction). The original structure of the logical formulas introduces constraints on these maps, so there is a faithful functor from the proof net category to $C$, but one which is neither full nor injective on objects.

In this work we construct a new category of proof nets for classical logic where the category $C$ is the “Free Frobenius category”, the free symmetric monoidal category generated by one object equipped with a Frobenius algebra structure [1]. We devise a correctness procedure for the maps in the category, i.e., we show a decision procedure for the maps in that category which come from proofs in the sequent calculus. We show how by forgetting some information, one obtains a category for which we have property that every map comes from a proof, so called full completeness. This use of Frobenius algebras in classical logic is quite different from the one proposed by Hyland [2] and furthered by Garner [3]. It more resembles the work of Lamarche-Strassburger [4], where the category $C$ is built from Sets and relations using an “interaction category” construction [2, Section 3], where composition is obtained by the means of a trace operator. Our new category has the desirable property of being resource-sensitive; we are able to capture complexity information inside the semantics and we show how topological characteristics of our proof nets, such as Euler characteristic, are good measure of complexity of a proof. Finally, we define a category that generalizes the Free Frobenius category, which carries the same complexity data, but which enjoys the property of full completeness.

1. Introduction

There are several logical calculi which have been endowed with categories of proof representations that have these common characteristics:

• The objects of such a category are just the formulas of the logic.

1The authors are partly supported by the ANR, project “STRUCTURAL”
2The first author is partly supported by the Ministry of Education and Science of Serbia, project ON 174026
• The maps (i.e. the proof representations) are graph-like structures that use the occurrences of atoms and negated atoms in the formulas as vertices.
• Every such graph is the denotation of a proof in the logic (say for a sequent calculus). This property is often called full completeness. Several proofs in a conventional proof formalism (e.g., sequent calculus) are identified, and thus a highly desirable reduction of bureaucracy is achieved.
• There is a “semantical” definition of the composition of these graphs which is independent from the cut-elimination process of the sequent calculus, although sometimes a precise correspondence can be established with the inductive steps in the definition of cut-elimination.

This is what is meant in this paper by a theory of proof nets for a given logic. We will use two examples as paradigms:

1. Multiplicative Linear Logic (without the constants), the original example [5],
2. Classical Logic, as presented in [4].

There are other examples, in particular for Multiplicative-Additive Linear Logic [6] and in the realm of non-commutative (and even non-associative), purely multiplicative logics [7].

Given such a theory of proof nets, the structure which best describes the associated class of proof nets is actually more than a category, in the sense that a map in a category is a proof of a two-formula sequent, while we always should be able to get proof nets for sequents with an arbitrary number of formulas. Moreover experience has shown that one-sided sequent calculi are much better suited for proof net construction, so an involutive negation is a built-in primitive, while categories better correspond to two-sided calculi. But to simplify presentation we will be a little vague about those distinctions and use the term “category” for something which should be a little different in practice, but can be approximated well by adding more structure to a category. The important common features are that composition is associative and equipped with units.

In our two main examples, the proof net structure on a sequent is a relation on the set of literal (atomic and negatomic) occurrences of the sequent. That is, denoting the literal occurrences on a sequent \( \Gamma \) by \( \mathcal{L}(\Gamma) \), the proof net structure is a set \( P \subseteq \mathcal{L}(\Gamma) \times \mathcal{L}(\Gamma) \) satisfying certain conditions. In particular, the following always holds: an element of \( P \) has to be a pair made of an atomic and a negatomic occurrence. In addition

• in Example 1, the relation \( P \) has to be a permutation of the set \( \mathcal{L}(\Gamma) \) (i.e., it has to be functional in both variables), and it has to satisfy a correctness criterion, which can be expressed in many different, equivalent ways [8],
• in Example 2, the relation \( P \) does not have to satisfy any functionality requirement, but it also has to obey a form of correctness [4].

One important point we want to make is that each of these two categories of proof nets can be embedded in a larger category where the distinction between conjunction and disjunction is erased, but where the important process of “semantical cut-elimination” (or interaction) is still defined. In a sense this larger category is the natural habitat of the interaction process. More precisely, in both cases we can define a category whose objects are simply sets labeled with atomics and negatomic formulas, and a map \( A \to B \) is relation \( P \subseteq (A + B) \times (A + B) \) where every pair in \( P \) links an atomic and a negatomic of the same sort.

• In the case of Example 1. a map \( P \) is just a partial bijection and composition is obtained by the famous interaction formula [9]. Partiality of the bijections is a result of the loss of a correctness criterion.
• In the case of Example 2. there are no additional requirements whatsoever for the relation \( P \) to obey, and the equivalent to the interaction process is given by formula on p. 12 in [4].

In this paper we present a new theory of proof nets for Classical Propositional Logic, which is very much in the mold of the above two examples. More precisely we will start by construct a category whose objects are just labeled finite sets (actually, we’ll first dispense with the labeling) and maps are some kinds of graphs on these sets. And then we will use this category to define a class of proof nets for propositional classical
logic, which naturally will introduce issues of correctness. These proof nets have the very desirable property of taking account of proof resources, unlike the nets presented in [4]. Interestingly they carry both more information (the counting of resources) and less information (explicit correspondences between atoms and negatoms) than the aforementioned proof nets.

1.1. The basic “semantical” category.

In this subsection, by graph we mean a category theorist’s graph, which in combinatorics is also called an oriented multigraph. That is, it is given by a set $V$ of vertices, a set $E$ of edges, and two functions $d_0, d_1 : E \to E$ for source and target.

**Definition 1.1.** A finite topological graph is a topological space which is (homeomorphic to one) obtained by the topological realization of a finite graph: given a graph as above, we take the disjoint sum $V \sqcup E \times [0, 1]$, where $V$ is seen as a discrete topological space and quotient it under the equivalence generated by the equations $d_0(e) = (e, 0), d_1(e) = (e, 1)$ for all $e \in E$.

The built-in notion of equivalence between topological graphs is homeomorphism; notice that since the map $x \mapsto 1 - x$ is an order-inverting homeomorphism on $[0, 1]$ a topological graph is not oriented. But we are interested in a coarser notion of equivalence.

**Definition 1.2.** Let $A$ be a topological space. An $A$-based space is a pair $(X, a_X)$ where $X$ is a topological space and $a_X : A \to X$ an embedding. An $A$-based map $f : (X, a_X) \to (Y, a_Y)$ is just a continuous function $f$ making the obvious triangle commute; in other words the category of $A$-based spaces is just the full subcategory of the co-slice category $A/\text{Top}$ determined by the embeddings. Given $A$-based maps $f, g : (X, a_X) \to (Y, a_Y)$, a relative homotopy between $f, g$ is a continuous map $X \times [0, 1] \to Y$ which is constant on the subspace determined by $a_X \times 1 : A \times [0, 1] \to X \times [0, 1]$.

It is well-known that relative homotopy defines an equivalence relation on the set of $A$-based maps $(X, a_X) \to (Y, a_Y)$, which is compatible with both left and right composition of maps, thus allowing to quotient the whole category of $A$-based spaces under that relation. Thus we can define a homotopy equivalence as an isomorphism in that quotient category, or equivalently as a pair $f : X \to Y, k : Y \to X$ of $A$-based maps such that $kf$ is relative homotopic to $1_X$ and $fk$ relative homotopic to $1_Y$.

From now on we assume that $A$ is a (finite) set seen as a discrete topological space. Thus a topological embedding $A \to X$ is just an injective function.

**Theorem 1.3.** Given an $A$-based topological graph $(X, a_X)$, then it is relative homotopy equivalent to a topological graph constructed as follows:

- The set of vertices is $A \sqcup \text{Comp}_X$, where $\text{Comp}_X$ is the set of connected components of $X$.
- Every element of $A$ is connected by exactly one edge to its connected component in $X$.
- The only other vertices are loops on the connected components.

Examples of these topological graphs are depicted in Figure 1. A full proof of this theorem is given in [10]. It uses the better-known (but much more technical) result that every topological graph-with-an-embedding is homology equivalent to one in that “normal form” [1, 11].

We are now ready to define the category that will structure our proof nets. We will actually define two categories whose maps are relative homotopy equivalence classes of finite topological graphs. In both categories the objects are finite sets.

- in category $\text{FFrob}$ a map $A \to B$ is a relative (with respect to $A \sqcup B$) homotopy equivalence of topological graphs $X$ equipped with an embedding from the disjoint sum $A \sqcup B \to X$. The composite of two maps $G_1 : A \to B$ and $G_2 : B \to C$ is obtained by gluing the graphs along their common border $B$ to get a graph $H$ and discarding the set $B$ to get an embedding $A \sqcup C \to H$. 

3
• in category $\text{FThFrob}$ a map $A \to B$ is a relative (with respect to $A \sqcup B$) homotopy equivalence of topological graphs $X$ equipped with an embedding from the disjoint sum $A \sqcup B \to X$, such that every connected component of $X$ is reached by an element of $A \sqcup B$. The composite of two maps is defined as in the previous case, except that in the graph $H$ that is obtained by gluing, the components that are not reached by $A, C$ (i.e., that are only reached by $B$) have to be discarded.

Notice that we could define the skeletal versions of these categories, by restricting ourselves to sets of the form $[n] = \{0, 1, \ldots, n-1\}$. Both $\text{FFrob}, \text{FThFrob}$ are symmetric monoidal categories for the operation of coproduct (disjoint sum): given $G_1: A_1 \to B_1$ and $G_2: A_2 \to B_2$ in either category then $G_1 \sqcup G_2: A_1 \sqcup A_2 \to B_1 \sqcup B_2$ is a map in the same category. Notice also that these two categories are actually compact-closed [10], with a contravariant involution which is identity on objects: A map $G: A \to B$ can also be considered to be a map $G: B \to A$, and a map $H: A \sqcup B \to C$ can also be considered to be a map $H: A \to B \sqcup C$.

Definition 1.4. Let $(C, \otimes, I)$ be a symmetric monoidal category. A Frobenius algebra is a sextuple $(A, \Delta, \Pi, \nabla, \nabla', \Delta')$, where $(A, \nabla, \Delta')$ is a commutative monoid, $(A, \Pi, \Delta)$ a co-commutative comonoid, and the following equations hold:

Every object of $\text{FFrob}$ is equipped with a Frobenius algebra structure. In the case of a one-element set the maps are generated by

\[
\begin{align*}
\Delta \otimes \text{id} & \quad \nabla & \quad \text{id} \otimes \Delta \\
A \otimes A & \quad A & \quad A \\
\text{id} \otimes \nabla & \quad \Delta & \quad \nabla \otimes \text{id} \\
A \otimes A & \quad A & \quad A \otimes A
\end{align*}
\]

where the maps $\Delta$ (diagonal), $\Pi$ (projection), $\nabla$ (co-diagonal) and $\Pi$ (co-projection), respectively, are depicted horizontally. Notice the use of a ‘terminator’ triangle symbol to denote the single graph vertex that is not an object generator. If a set has $n$ elements then the diagonal and co-diagonal are obtained by using $n$ disjoint copies of the maps for one-element sets as in $\nabla$ here

\[
\begin{align*}
\Delta & \quad \Pi & \quad \nabla \\
\text{Object} & \quad \text{Object} & \quad \text{Object}
\end{align*}
\]

Theorem 1.5. [10] The category $\text{FFrob}$ is the free symmetric monoidal category with an object equipped with a Frobenius algebra structure.

Naturally the meaning of the term “free” is adapted to the context of categories, where free objects are defined up to equivalence and not up to (unique) iso.

Definition 1.6. Given a symmetric monoidal category $C$, a Frobenius algebra $(A, \Delta, \Pi, \nabla, \Pi)$ in it is said to be thin if for all $k \geq 0 \Pi \circ (\nabla \circ \Delta) \circ \cdots \circ (\nabla \circ \Delta) \circ \Pi$ is the identity.

Theorem 1.7. The category $\text{FThFrob}$ is the free symmetric monoidal category with an object equipped with a thin Frobenius algebra structure.

The full proof is given in [10]. Thus, the $P$ component of our proof nets will be given by the category $\text{FFrob}$. 4
2. F-Prenets

We are working in the standard language for propositional classical logic one-sided calculi (but without the two logical constants). Formulas/types are built with atomics $a,b,a_1,\ldots$, negatomics $\overline{a}, \overline{b}, \overline{a}_1,\ldots$ and the binary conjunction and disjunction connectives $\land, \lor$. Negation is defined as usual in this context, by induction:

$$\overline{a} = a; \overline{\overline{a}} = a; (A \land B) = \overline{A} \lor \overline{B}; (A \lor B) = \overline{A} \land \overline{B}.$$  

A type for us is the same as a formula, but we use the term sort to express what a formula and its negation have in common: $a$ and $\overline{a}$ have the same sort, but opposite polarities (and different types).

Uppercase Roman letters like, $A, B, C, X, Y, \ldots$ will denote formulas, and uppercase Greek letters $\Gamma, \Sigma, \ldots$ sequents. In addition, we stick to the convention that when we use end-of-the alphabet letters $x,y,z,X,Y\ldots$ the letter itself contains no information on the type, while it is the case for beginning-of-the-alphabet letters $a,b,c,A,B,C$. For example we can contract occurrences $X,Y$ in a sequent (this supposes, naturally, that they denote the same type) while we will never try to contract $A,B$ because this choice of letters carries the information that they have different types, the same going with different subscripts $A_1, A_2$.

When we need to distinguish different occurrences of a same type (literal, formula or sequent), we use superscripts $A^1, A^2, \ldots$.

The notation and terminology we use here follows as much as it can [4]. The first proof system for Classical Logic we will consider appears in that same paper. It is a one-sided variant of Gentzen’s LK with the rule Mix added. The complete calculus is as follows (later the Exchange rule will often be implicit).

<table>
<thead>
<tr>
<th>Rule</th>
<th>Premises</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ax$</td>
<td>$\vdash \pi, a$</td>
<td>$\vdash \Gamma, A, B$</td>
</tr>
<tr>
<td>$\lor$</td>
<td>$\vdash \Gamma, A \lor B$</td>
<td>$\lor \Gamma_1, A \lor B, \Gamma_2$</td>
</tr>
<tr>
<td>$\land$</td>
<td>$\vdash \Gamma_1, A \land B, \Gamma_2$</td>
<td>$\land \Gamma, A, B, \Sigma$</td>
</tr>
<tr>
<td>$Exch$</td>
<td>$\vdash \Gamma, A, B, \Sigma$</td>
<td>$\vdash \Gamma, B, A, \Sigma$</td>
</tr>
<tr>
<td>$Weak$</td>
<td>$\vdash \Gamma, A$</td>
<td>$\vdash \Gamma, A, A$</td>
</tr>
<tr>
<td>$\lor$</td>
<td>$\lor \Gamma, A$</td>
<td>$\lor \Gamma, \overline{A}, \Sigma$</td>
</tr>
<tr>
<td>$Contr$</td>
<td>$\lor \Gamma, A, A$</td>
<td>$\lor \Gamma, \overline{A}, \Sigma$</td>
</tr>
<tr>
<td>$Cut$</td>
<td>$\lor \Gamma, A, \overline{A}, \Sigma$</td>
<td>$\lor \Gamma, \overline{A}, \Sigma$</td>
</tr>
<tr>
<td>$Mix$</td>
<td>$\lor \Gamma, \overline{A}, \Sigma$</td>
<td>$\lor \Gamma, \overline{A}, \Sigma$</td>
</tr>
</tbody>
</table>

Figure 2: System CL.

The following definition should come as no surprise given the previous section.
Definition 2.1 (Linking, general case). Let $P$ be a set whose elements are labeled by atom and negatom symbols. We define a linking to be a triple

$$P = (P, \text{Comp}_P, \text{Gen}_P)$$

where

- $\text{Comp}_P$ is partition of the set $P$ of literals such that each class $C \in \text{Comp}_P$, to which we refer as a component, contains literals of arbitrary polarity, but all of the same sort,
- $\text{Gen}_P : \text{Comp}_P \to \mathbb{N}$ is a function that assigns a natural number (its genus) to each component of the class $\text{Comp}_P$.

Notice the standard abuse of notation in the overloading of the symbol $P$, which stands at the same time for the underlying set and the whole set-with-structure.

Definition 2.2 (F-prenet). Let $P = (P, \text{Comp}_P, \text{Gen}_P)$ be a linking. We define an F-prenet to be a pair $P \triangleright \Gamma$ consisting of a linking $P$ and a sequent $\Gamma$, where there is a bijective, type-preserving correspondence between the literal occurrences in $\Gamma$ and the elements of $P$.

There is no need to have a name for that bijection, since it will always be clear what it is.

We will make much use of an alternative definition to the previous, obviously equivalent to it.

Definition 2.3.

- A component with genus is a pair $K = (C, G)$, where $G$ is a natural number, and $C$ is a non-empty set labeled with atoms and/or negatoms of the same sort.
- A linking $P$ is a finite set $P = \{(C_1, G_1), \ldots, (C_k, G_k)\}$ of components-with-genera, where sets $C_1, \ldots, C_k$ are pairwise disjoint.
- An F-prenet is a pair $P \triangleright \Gamma$ where $P$ is a linking and $\Gamma$ is a sequent such that the union $C_1 \cup \cdots \cup C_k$ of all of the components of $P$ is equipped with a bijection with the set of literals in $\Gamma$.

Again, there is no need to explicitly name that bijection. We will give no preference to any of the equivalent notations, and will use the one that facilitates easier notation in a given situation.

Graphical representation of F-prenets is done as is Figure 3: the sequent is represented as its syntactic forest with (neg)atoms as leaves. Each literal is connected to a base point representing a component of the linking, while the number (genus) assigned to a component is represented by the same number of loops sharing the same base point.

2.0.1. Operations on Linkings

In the course of transforming a sequent calculus proof into a proof net, we need to define a set of operations on linkings, as well as a good notation for them. This section is devoted to this, and to giving some elementary properties of these operations. Notice that a linking $P = (P, \text{Comp}_P, \text{Gen}_P)$ can be seen as a map $\emptyset \to P$ in the category $\text{FThFrob}$ (where we forget about the labelling of $P$ with literals).

The first operations we will define are those associated with Contraction. Let us start with trying to contract two (occurrences of) atomic formulas. We obtained this by composing with map $\nabla$ of (1.1) above.

Definition 2.4. Let $P = (P, \text{Comp}_P, \text{Gen}_P)$ be a linking and $x, y$ two distinct occurrences of the same literal in $P$. We define a new linking

$$P(x \triangleright y), \text{Comp}_{P(x \triangleright y)}, \text{Gen}_{P(x \triangleright y)}$$

whose underlying set of literal occurrences $P(x \triangleright y)$ is obtained from $P$ by identifying $x$ and $y$. Two cases may happen.
Let \( x, y \) belong to the same component of \( \text{Comp}_P \). We write \( P(x \ useful x y) \) to emphasize that this happens; the set \( \text{Comp}_P(x \ useful x y) \) obviously has the same cardinality as \( \text{Comp}_P \) and for a component \( C \in \text{Comp}_P(x \ useful x y) \) we define
\[
\text{Gen}_{P(x \ useful x y)}(C) = \begin{cases} 
\text{Gen}_P(C), & \text{if } C \text{ is not affected by identifying } x, y \\
\text{Gen}_P(C) + 1, & \text{otherwise}.
\end{cases}
\]

Let \( x, y \) belong to different components of \( \text{Comp}_P \). We write \( P(x \ useful x y) \) to emphasize that this happens. The set \( \text{Comp}_P(x \ useful x y) \) obviously has the cardinality of \( \text{Comp}_P \) minus one, and for a component \( C \in \text{Comp}_P(x \ useful x y) \) we define
\[
\text{Gen}_{P(x \ useful x y)}(C) = \begin{cases} 
\text{Gen}_P(C), & \text{if } C \text{ is not affected by identifying } x, y \\
\text{Gen}_P(C_1) + \text{Gen}_P(C_2), & \text{otherwise}.
\end{cases}
\]

Thus \( P(x \ useful x y) \) is either \( P(x \ useful x y) \) or \( P(x \ useful x y) \), depending.

Given \( 2n \) pairwise distinct occurrences \( x_1, y_1, \ldots, x_n, y_n \), with \( x_i \) having the same type as \( y_i \), we denote \( P(x_1 \ useful y_1, \ldots, x_n \ useful y_n) \) for \( P(x_1 \ useful y_1) \cdot \ldots \cdot (x_n \ useful y_n) \).

It is not difficult to see that the order is irrelevant here, i.e. \( P(x_1 \ useful y_1, x_2 \ useful y_2) = P(x_2 \ useful y_2, x_1 \ useful y_1) \); this is due to the underlying Frobenius algebra structure.

Let \( A \) now be a formula, and \( x_1, \ldots, x_n \) the ordered list of all its literal occurrences. Supposing \( P \vdash \Gamma, A^1, A^2 \) is an F-prenet over a sequence that contains two occurrences of that formula, we define the linking
\[
P(A^1 \ useful A^2) = P(x_1^1 \ useful x_2^1, \ldots, x_n^1 \ useful x_n^2),
\]
which also obviously defines a prenet over the sequent \( \Gamma, A \). Similarly, if \( \Gamma = A_1, \ldots, A_m \) is a list of formulas and \( P \vdash \Delta, \Gamma^1, \Gamma^2 \) is a prenet over a sequent that contains twice that list, we define the linking
\[
P(\Gamma^1 \ useful \Gamma^2) = P(A_1^1 \ useful A_1^2, \ldots, A_m^1 \ useful A_m^2),
\]
which is a prenet over the sequent \( \Delta, \Gamma \).

Another similar operation is for the Cut.

**Definition 2.5** (\( \sim \) transformation). Let \( P = (P, \text{Comp}_P, \text{Gen}_P) \) be a linking
1. Let \( \{x^1, x^2\} \) be a component in a linking \( P \), we denote \( P(x^1 \ useful x^2) \) to be the linking defined by:
   - \( \text{Comp}_P(x^1 \ useful x^2) = \text{Comp}_P \setminus \{x^1, x^2\} \)
   - \( \text{Gen}_P(x^1 \ useful x^2) \) agrees with \( \text{Gen}_P \) on its domain.

Figure 3: Graphical representation of an F-prenet (left) and a generic 'icicle' representation of a sequent forest (right)
(Here, the v in \(\prec\) stresses the fact that the induced component vanishes in the transformation)

- In the case \(x^i, x^j\) are two occurrences of literals from a same component in \(\text{Comp}_P\), such that they are not the only literals in the common component, i.e. \(\{x^i, x^j\} \notin \text{Comp}_P\). We define linking \(P(x^i \prec_s x^j)\) as follows (the s in \(\prec_s\) stresses the fact that the literal occurrences are from the same component):
  - the set of literals \(P(x^i \prec_s x^j)\) is obtained from \(P\) by removing the two literals \(x^i\) and \(x^j\),
  - \(\text{Comp}_{P(x^i \prec_s x^j)}\) is partition of \(P(x^i \prec_s x^j)\) obtained from \(\text{Comp}_P\) by removing \(x^i\) and \(x^j\) from the component they both belong to

- \(\text{Gen}_{P(x^i \prec_s x^j)}\) is defined as follows: for a \(C \in \text{Comp}_{P(x^i \prec_s x^j)}\),
  \[
  \text{Gen}_{P(x^i \prec_s x^j)}(C) = \begin{cases} 
  \text{Gen}_P(C) & \text{if } C \text{ is not altered by } x^i \prec_s x^j \\
  \text{Gen}_P(C) + 1, & \text{otherwise}
  \end{cases}
  \]

- Let now \(x^i, x^j\) be two literals from different components in \(\text{Comp}_P\), i.e. \(\{x^i, x^j\} \notin \text{Comp}_P\). We define linking \(P(x^i \prec_d x^j)\) as follows (the d in \(\prec_d\) stresses the fact that the literals are from different components):
  - the set of literals \(P(x^i \prec_d x^j)\) is obtained from \(P\) by removing the two literals \(x^i\) and \(x^j\),
  - \(\text{Comp}_{P(x^i \prec_d x^j)}\) is partition of \(P(x^i \prec_d x^j)\) obtained from \(\text{Comp}_P\) by joining the two classes \(x^i\) and \(x^j\) belong to and removing both \(x^i\) and \(x^j\) from the newly formed class

- \(\text{Gen}_{P(x^i \prec_d x^j)}\) is defined as follows: for a \(C \in \text{Comp}_{P(x^i \prec_d x^j)}\),
  \[
  \text{Gen}_{P(x^i \prec_d x^j)}(C) = \begin{cases} 
  \text{Gen}_P(C) & \text{if } C \text{ is not affected by } x^i \prec_d x^j \\
  \text{Gen}_P(C_1) & \text{if } C \text{ is obtained by merging}\n  + \text{Gen}_P(C_2), & C_1, C_2 \text{and removing } x^i, x^j.
  \end{cases}
  \]

- One defines \(P(x^i \prec x^j)\) to be
  \[
  P(x^i \prec x^j) := \begin{cases} 
  P(x^i \prec_s x^j) & x^i, x^j \text{ are from the same component in } P \\
  P(x^i \prec_d x^j) & \text{otherwise}.
  \end{cases}
  \]

- \(P(x^i \prec x^j, \ldots, y^r \prec y^s)\) is defined as \(P(x^i \prec x^j) \ldots (y^r \prec y^s)\). Again, the order is irrelevant, i.e.
  \(P(x^i \prec x^j, y^r \prec y^s) = P(x^i \prec x^j)(y^r \prec y^s) = P(y^r \prec y^s)(x^i \prec x^j)\) and we choose to omit an easy proof-by-cases.

- If \(x^1, \ldots, y^r\) are all of the literals in \(A^1\) and \(x^j, \ldots, y^s\) are all of the respective literals in \(A^2\), we define
  \(P(A^1 \prec A^2) := P(x^1 \prec x^j, \ldots, y^r \prec y^s)\).

**Definition 2.6** (\(\uparrow\) transformation). Let \(P = (P, \text{Comp}_P, \text{Gen}_P)\) be a linking and \(x^i, \ldots, y^j\) fresh literals. One defines:

\[
P \uparrow \{x^i, \ldots, y^j\} := P \cup \{(\{x^i\}, 0), \ldots, (\{y^j\}, 0)\}.
\]

If \(x^i, \ldots, y^r\) are all literals in \(A\), we define \(P \uparrow A := P \uparrow \{x^i, \ldots, y^r\}\). Also, if \(\Gamma = A, \ldots, B\), we define \(P \uparrow \Gamma := P \uparrow A \ldots \uparrow B\).

**Definition 2.7** (Restriction). Let \(P = \{(C_1, G_1), \ldots, (C_k, G_k)\}\) be a linking, and let \(Q\) be an arbitrary subset of the set \(P\) of literals, and let. One defines the restriction of \(P\) to \(Q\), denoted by \(P|_Q\) as:

\[
P|_Q := \bigcup_{C, C \cap Q \neq \emptyset} \{(C \cap Q, G_i)\}.
\]

In particular, given an F-net \(P \triangleright \Gamma\), and an arbitrary subforest \(\Gamma'\) of the syntactic forest of \(\Gamma\) induced by choice of subset \(Q\) of the set of literals in \(\Gamma\), we define \(P|_{\Gamma'}\) to be \(P|_Q\). Notice that \(\Gamma'\) need not (and it frequently won’t) be a sequent itself.

Notice that \((P \uparrow \{x\})(x \gamma y)\) is equal to \(P\), in other words that the operation \(\uparrow\) acts as a unit for \(\Upsilon\).
2.0.2. Syntactic category of F-prenets

We are in the process of constructing several categories in which F-prenets are maps. F-prenets can also be seen as objects in a quite different category.

**Definition 2.8.** Let $\mathcal{FSynt}$ denote the syntactic category of F-prenets, where

1. an object is an F-prenet
2. a map $f : P \to Q$ is a function between sets $f : P \to Q$ (i.e. $\mathcal{L}(\Gamma) \to \mathcal{L}(\Delta)$) s.t.
   (a) for every formula $A$, $f$ maps $\mathcal{L}(A)$ to a subset of $\mathcal{L}(\Delta)$ which is the set of literals of (and thus defines) a subformula in $\Delta$, and the restriction of $f$ to $\mathcal{L}(A)$ preserves the syntactical left-to-right order of the literals on these formulas;
   (b) for every component $(C,m)$ in the linking $P$ there is a $(C',n)$ in $Q$ such that the direct image $f(C) \subseteq C'$ and $m \leq n$;
   (c) given a component $(C',n)$ in $Q$, then
   \[ \sum_{\{ (C,m) | f(C) \subseteq C' \}} m \leq n. \]

**Example 2.9.** If $\Gamma$ has only one formula, there can be at most one syntactic map $P \to Q$ because the map $P \to Q$ is necessarily determined by the identity map on $\Gamma$.

**Example 2.10** (Isomorphisms in $\mathcal{FSynt}$). Consider an iso $f$ in $\mathcal{FSynt}$. Property 2. in the definition of $\mathcal{FSynt}$ requires $f$ restricted to a component to be a bijection, i.e. for an iso $\subseteq$ is an equality. Similarly, $\leq$ is equality, hence $f$ preserves linkings. Since literals of a formula are mapped to literals defining a subformula, formulas are mapped to formulas. Therefore, an iso in $\mathcal{FSynt}$ is a map that permutes formulas of an F-prenet.

On the other hand, all Exchange-induced maps are isos in $\mathcal{FSynt}$, as mentioned before. Thus isos in $\mathcal{FSynt}$ are precisely the maps induced by Exchange.

Since in our definition of linking $(P,\text{Comp}_P, \text{Gen}_P)$ the set $P$ is arbitrary, for any F-prenet $P \to \Gamma$ there
are many other prenets $P' \triangleright \Gamma$ where the function $P \rightarrow P'$ that corresponds to the identity on $\Gamma$ is an isomorphism of linkings. Thus there are many copies lying around of "the same" F-Prenet. This situation is useful technically, because it simplifies many constructions. This can be related to the following observation: given a natural number $n$, in some circumstances it is preferable to have a single totally ordered set with $n$ elements (usually the ordinal $[n] = \{0, 1, \ldots, n-1\}$, cf. example in the previous section) but in other circumstances (like here) it is better to have arbitrary totally ordered sets, whose elements can have a range of names. As with total orderings the presence of many copies of an F-prenet isomorphism type is not a problem because the isomorphism that relates two of them is uniquely defined.

2.0.3. From classical sequent proofs to F-prenets

In this section we show how to assign an F-prenet to a classical sequent proof. We start with our initially chosen proof system $CL$, to which we add a binary connective $\rightarrow$ for interpreting Cut, to allow us to delay the cut-elimination process. It is only allowed to appear in as a root in a sequent, and if a sequent $\Gamma$ contains a cut formula $A \rightarrow B$, literals of both $A, B$ are formally considered to be in $L(\Gamma)$, but are invisible in the sequent, since they cannot appear as premisses of rules.

The essential point point of the definition of syntactic map is that, whatever proof system we use, every unary rule $\Pi \triangleright \Gamma$ will give rise to a syntactic map $P' \triangleright \Gamma' \rightarrow P \triangleright \Gamma$ and every binary rule $\Pi \triangleright \Gamma \rightarrow \Pi' \triangleright \Gamma'$ will give rise to two syntactic maps $P' \triangleright \Gamma' \rightarrow P \triangleright \Gamma$ and $P'' \triangleright \Gamma'' \rightarrow P \triangleright \Gamma$. We name these maps by the rule they are assigned to, i.e. we speak of a $\lor$-map, of $\land$-maps, etc.. In the table below we give the two ways we have to interpret Cut: with delayed reduction, which is just the introduction of a $\lor$ connective, and with immediate reduction, which uses the semantics to reduce the Cut.

\[
\begin{array}{c|c|c|c|c}
Ax: & \lor: & \land: & \downarrow: & \tt Mix: \\
\{(\pi, a), 0\} \triangleright \pi, a & P \triangleright \Gamma, A, B & P \triangleright \Gamma, A \lor B & P' \triangleright \Gamma, A & P'' \triangleright \Gamma, A, \Sigma \\
\downarrow: & P' \triangleright \Gamma, A, B & P' \triangleright \Gamma, \Sigma & P' \triangleright \Gamma, A, B, \Sigma & P' \triangleright \Gamma, A, B, \Sigma \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A^1, A^2 & P \triangleright \Gamma, A^1, A^2 \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A, B, \Sigma & P \triangleright \Gamma, A, B, \Sigma \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A, B, \Sigma & P \triangleright \Gamma, A, B, \Sigma \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A, B, \Sigma & P \triangleright \Gamma, A, B, \Sigma \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A, B, \Sigma & P \triangleright \Gamma, A, B, \Sigma \\
\downarrow: & P \triangleright \Gamma, A, B & P \triangleright \Gamma, B, A & P \triangleright \Gamma, A, B, \Sigma & P \triangleright \Gamma, A, B, \Sigma \\
\end{array}
\]

Thus we have a construction that assigns a well defined F-prenet to a given $CL$ proof. The inverse, the problem of existence of a $CL$ proof for given an F-prenet is more challenging, and will need more work.

As an example, Figure 5 shows a cut-free $CL$ proof which utilizes two axiom links, proving the same sequent as a single axiom link. Such proof can be iterated (concatenated). The assigned F-prenet is shown to the right. It is worth noticing that had we iterated the proof, the corresponding F-prenet would be almost the same, with the difference of genus of the single component being equal to the number of iterations. This means that F-prenets can count applications of the axiom rule.
2.1. Accommodating the cut. Calculus FL.

The real challenge in dealing with Cut is handling its elimination. Naturally we already made a choice on how it should be done, i.e., using composition in FThFrob:

Definition 2.11. Given an F-prenet with one cut formula, a cut-reducing step \( \rightarrow \) acts on linkings as follows: \( P \vdash \Gamma, \neg A \) \( \rightarrow \) \( P(A \rightarrow \neg A) \vdash \Gamma \).

As an illustration, look at the cut wire in the F-prenet to the left on Figure 6, corresponding to the classical proof above it. Notice that the proof is the notorious case of contraction-contraction step in cut elimination. The proof to the right is cut-free and shares the same F-prenet as the one to the right. The treatment of cut in F-prenets, therefore, provides a novel way of eliminating cut in this difficult case.

![Figure 7: Cut elimination performed on two CL-correct F-nets that results in an F-prenet not corresponding to a CL proof.](image)

This is where we first run into problems. We obviously want that the cut reduction process applied to an F-prenet obtained from a sequent proof should yield an F-prenet which also can be obtained from a sequent proof (and without using the Cut rule). However, consider the Example of Figure 7. The resulting F-prenet cannot come from a proof in CL, since it contains a single-atom component with a non-zero genus; it is very easy to show that such a component cannot be produced by the process given in Table 2.1.

A solution to this problem lies in a more creative treatment of Weakening. For that purpose, we go to the syntax and change our calculus for propositional classical logic; the result is system FL on Figure 8.
The deductive system $\text{FL}$ comes from a simple idea: we want to be able to weaken several literals at once, which will form a single component in the interpretation partition. Such weakened literals should be able to form compound formulas. This requires special treatment, and thus the presence of the stoop symbol `;` to separate from the rest of the sequent a part of it which is guaranteed to come from weakenings.

**Proposition 2.12.** $\text{FL}$ is a sound and complete calculus for classical propositional formulas. More precisely, if $\vdash \Gamma ; \Sigma$ is provable in $\text{FL}$, $\forall \Gamma$ is a valid propositional formula, as is $\forall \Gamma, \Sigma$. $(\forall(-))$ denotes the formula which is disjunction of the formulas in the argument. Conversely, for a valid propositional formula $\Lambda, \vdash \Lambda ;$ is derivable in $\text{FL}$.

**Proof.** Soundness of the calculus is straightforwardly checked by induction, showing that every rule is valid from the point of view of provability. Completeness is also unproblematic, since $\text{CL}$ can be seen as a fragment of $\text{FL}$. Indeed, by forgetting the stoop, every $\text{CL}$ rule other than $\text{Weak}$ appears as a rule in $\text{FL}$. Weakening in $\text{CL}$ can be viewed in $\text{FL}$ as a sequence of $\text{MulWeak}$ rule applications that introduce the literals, followed by a sequence of $\text{Cut}$ rules which introduce the connectives of the weakened formula. Since $\text{CL}$ is complete for classical propositional logic, so is $\text{FL}$.

The intended interpretation of rules largely coincides with the one for rules of $\text{CL}$, where one disregards the stoop symbol—the axiom rule is interpreted the same way, so are contractions, $\text{Cut}$ introduces a cut formula, while logical rules do not affect the linkings, only the sequents. The main difference is in the interpretation of $\text{Weakening}$:

$$\vdash \Gamma ; \Delta$$

$\text{MulWeak}$

Weakening in the case of $\text{FL}$ is interpreted as adding a single component of genus 0 consisting of the weakened literals

$$P \vdash \Gamma ; \Delta$$

$$P \vdash \{\{a, a, \ldots, a, \bar{a}, \bar{a}, \ldots, \bar{a}\}, 0\} \vdash \Gamma ; \Delta, a, \ldots, a, \bar{a}, \bar{a}, \ldots, \bar{a}.$$
Definition 2.13. Let $\mathbf{CL}^+$ be the calculus obtained from $\mathbf{CL}$ by replacing $\text{Weak}$ by the following rule (with the obvious breach of syntactic independence of semantics):

$$
\frac{\vdash \Gamma}{Q \vdash \Gamma, \Sigma} \text{Weak}^+ ,
$$

where $Q$ is an arbitrary linking over the literals of $\Sigma$.

The intended interpretation of this rule is obviously: $\frac{P \triangleright \Gamma}{P \uplus Q \triangleright \Gamma, \Sigma}$.

The proof theories of $\mathbf{FL}$ and $\mathbf{CL}^+$ are very similar. But, to be able to express the precise connection between the two, we need to introduce the following notation: given an $\mathbf{FL}$ proof of $\vdash \Gamma; \Sigma$ and its assigned $\mathbf{F}$-prenet $P \triangleright \Gamma, \Sigma, \Pi^\phi$ for the sequence $\Pi^\phi$ of cut formulas, we split the cut formulas into the left- and right-cut formulas. Eg.

$$
P_1 \triangleright \Gamma, A, \Delta_1, \Pi_1^\phi, \Pi_2^\phi, P_2 \triangleright \overline{A}, \Sigma, \Delta_2, \Pi_3^\phi, \Pi_4^\phi \implies \text{Cut}_l
$$

A formula remains left or right, respectively, when an $\mathbf{FL}$ rule is interpreted, except in the case of $\land_c$ and $\text{Cut}_r$ rules. In these cases, all left-cut formulas assigned to the branch of the proof of the left premise become right cut formulas. Eg.

$$
P_1 \uplus P_2 \triangleright \Gamma, \Sigma, \Delta_1, \Delta_2, A \lor \rho, A, \Pi_1^\phi, \Pi_2^\phi, \Pi_3^\phi, \Pi_4^\phi \implies \land_c^l
$$

Notice that we are only discussing notation here that is going to be used in the following proposition, $\mathbf{FL}$ proofs are still assigned $\mathbf{F}$-prenets in the usual sense.

Proposition 2.14. Given a proof of a sequent $\vdash \Gamma$ in $\mathbf{CL}^+$, there exists a decomposition of $\Gamma = \Gamma', \Gamma''$ such that there is a proof of $\vdash \Gamma'; \Gamma''$ in $\mathbf{FL}$ that yields the same prenet.

Conversely, let an $\mathbf{FL}$ proof of $\vdash \Gamma; \Sigma$ be given and let $P \triangleright \Gamma, \Sigma, \Delta^\phi_l, \Delta^\phi_r$ be the assigned $\mathbf{F}$-net, for sequences of left- and right-cut formulas $\Delta^\phi_l$ and $\Delta^\phi_r$. Then, there is a $\mathbf{CL}^+$ proof of $\vdash \Gamma$ with the assigned $\mathbf{F}$-net $P|_{\Gamma, \Delta^\phi_l} \triangleright \Gamma, \Delta^\phi_l$.

Proof. As usual, we omit explicit mention of Exchange in what follows.

The easy part is showing how a proof in $\mathbf{CL}^+$ can be seen as a proof in $\mathbf{FL}$. To that purpose, notice that every rule of $\mathbf{CL}^+$ is a rule of $\mathbf{FL}$ when one disregards the stoup symbol, with the exception of $\text{Weak}^+$. Indeed, in $\mathbf{CL}$ one is allowed to introduce an arbitrary configuration in the linking over the weakened literals, whereas $\mathbf{FL}$ allows only a single component with the genus 0. But

Proposition 2.15. The rule $\text{Weak}^+$ is derivable in $\mathbf{FL}$, with exactly the same semantics.

Proof. Let $\{ (C_1,G_1), \ldots, (C_n,G_n) \} \vdash \Gamma, \Sigma$ be an occurrence of an application of that rule, and suppose we have an $\mathbf{F}$-prenet $P \triangleright \Gamma$. For every $i \leq n$ let $\Delta_i$ be the sequent which is just made of the literals of the component $C_i$. In other words, choosing an arbitrary $i$, $\Delta_i$ is of the form $x_1, \ldots, x_m$, where there is an atomic a such that every $x_j$ is either a $a$ or $\pi$. Choose an arbitrary literal occurrence, say $x_{i_0}$, which we can
suppose w.l.o.g. to be the atomic \( a \). Performing \textit{MulWeak} on \( P \vdash \Gamma \) with \( \Delta \), disjointly adds the F-prenet \( \{(C_i,0)\} \vdash \Delta_i \) to \( P \vdash \Gamma \). After this is done, perform a \textit{MulWeak} with the two-atomic sequent \( a,a \), and then contract these two together. We have obtained the F-prenet \( P \uplus \{(C_i, 1)\} \vdash \Gamma, \Delta_i,a \). Then contracting that added \( a \) on \( x_1 \) give us \( P \uplus \{(C_i, 1)\} \vdash \Gamma, \Delta_i \), thus we have increased the genus by one. We can iterate this genus-increment trick \( G_i \) times to obtain the sequent \( P \uplus \{(C_i, G_i)\} \vdash \Gamma, \Delta_i \). Now that we know how to add a component with arbitrary genus we can iterate the procedure, one time each for every \( (C_i,G_i) \), and we can produce the F-prenet

\[
P \uplus \{(C_1,G_1), \ldots, (C_n, G_n)\} \vdash \Gamma, \Delta_1, \ldots, \Delta_n .
\]

Finally we apply logical \( \neg r \) rules to turn all these added literals into the sequent \( \Sigma \).

Rule \textit{Weak+} officially belongs \( \text{CL}+ \), but as we will soon see it is quite useful to consider it also as a rule of \( \text{FL} \), since it is technically powerful but doesn't change the logical power of the calculus.

The second part of the proposition can be shown by induction on an \( \text{FL} \) proof. The base of the induction, an instance of the axiom rule is immediate. As for the induction step, notice that every \( \neg r \) rule, can be replicated in \( \text{CL}+ \). The induction step follows also immediately for the \textit{MulWeak}, \textit{Exch}, \textit{Mix}, \textit{\land c}, \textit{Cut}i and all of the \( \land r, \lor r, \text{Cut}r \). For instance, in the case of \textit{Cut}r, the induction hypothesis for premisses gives \( \text{CL}+ \)-proofs of \( \vdash \Gamma, A \) and \( \vdash \Sigma \) whose F-prenets are \( P_1 \vdash \Gamma, A, P_1^{\lor} \) and \( P_2 \vdash \Sigma, P_1^{\lor} \). Now, by the above definition of left- and right-cuts, what we need is a \( \text{CL}+ \) proof of \( \vdash \Sigma \) with a F-prenet \( P_2 \vdash \Sigma, P_1^{\lor} \), which entails keeping the one of subproofs we already have.

Therefore, what it remains to show is that \textit{Contr}c, \textit{Contr}r, \textit{\lor c}, can be simulated in \( \text{CL}+ \) in the sense of the claim.

In the cases of \textit{Contr}c and \textit{Contr}r, one is required to produce a \( \text{CL}+ \) proof of the sequents \( \vdash \Gamma, A \) and \( \vdash \Gamma \), respectively, form the given proofs of the same sequents, but whose assigned F-prenets are possibly more connected or with larger genus. How this can be done in the case of \( \text{FL} \) is shown in Lemma 3.12 on page 27 and in Lemma 3.13 on page 27 (the result presented there is independent of the current discussion and could be replicated here).

Finally, for e.g. \( \lor c \), one can simulate the \( \text{FL} \) derivation in the sense of the claim by extending the \( \text{CL}+ \)-proof of the induction hypothesis as below

\[
\begin{align*}
\vdash \Gamma, A & \quad \text{Weak+} \\
\vdash \Gamma, A, B, \Gamma & \quad \text{Contr} \\
\vdash \Gamma, A, B & \quad \lor c.
\end{align*}
\]

We also need to describe the linking in the \textit{Weak+} step. For a every component of the F-net assigned to the \( \text{FL} \) proof connected to the literals of \( B \) we choose a literal of \( \Gamma \). The linking of \textit{Weak+} contains zero-genus components consisting of these literals of \( \Gamma \) and literals of \( B \) they are connected to. Every other literal of the introduced \( B, \Gamma \) is in its own single-element-zero-genus component. This completes the proof of the proposition.

With an appropriate use of \textit{MulWeak} and perhaps subsequent contractions, one has:

\begin{corollary}
For every cut-free proof of a sequent \( \vdash \Gamma ; \Sigma \) in \( \text{FL} \) there exists a cut-free proof of \( \vdash \Gamma, \Sigma \) in \( \text{CL}+ \) that produces the same F-prenet.

Conversely, given a cut-free proof of \( \vdash \Gamma \) in \( \text{CL}+ \) there exists a decomposition of the sequent \( \Gamma = \Gamma_1, \Gamma_2 \) such that there exists a proof of \( \vdash \Gamma_1 ; \Gamma_2 \) in \( \text{FL} \) that produces the same F-prenet.
\end{corollary}
We see that the systems FL and CL+ are not exactly equivalent in terms of the F-prenets they produce, but in practice they are. This is because the extra stuff that appears in a prenet that comes from a proof in FL is disjoint from the rest, and has no real logical contents whatsoever, since it comes from a succession of $\neg r$ introduction rules applied to weakened formulas. Thus something logically equivalent could be produced without using the $Cut_r$ formula, making for a prenet that is much easier to deal with.

One can design a version of FL where this discrepancy would disappear, but at the cost of much added complexity.

Thus the system FL should actually be seen as a syntactic justification for CL+. The latter cannot be seen as a formal system in the usual sense, but it is what we need in practice to prove the important results that follow. We will see that what will be called FL-correctness (soon to be defined) could perhaps be better called CL+-correctness.

2.2. Correctness for F-prenets

We have seen that going from a sequent proof to an F-prenet is straightforward. The inverse question ‘when does an F-prenet correspond to a proof?’ seems to be a difficult one. In fact, as argued in [4] for N-nets, one should not hope for an easy correctness criterion. The authors argue that in the case of N-nets, ability to count axiom links of these nets, i.e. to keep track of resources, makes the problem close to the $NP=coNP$? problem (cf. [12]). The same is bound to happen here, since we have seen that the F-prenets are also capable of counting axiom links.

Example 2.17. To illustrate how delicate the issue of correctness is, consider the F-prenets in Figure 9. The figure to the left corresponds to no CL proof, but it corresponds to a proof in a deep inference system SK from [13], while there is neither CL nor SK proof for the F-prenet to the right.

We show the claims on existence/non-existence of proofs in respective systems for the given F-prenets in Example 3.18.

As the previous example shows, any correctness criterion one may devise needs to be tailored to a specific deductive system. Therefore, in what immediately follows, we restrict our attention to the correctness of F-prenets for the classical sequent system FL we have just constructed.

Notation 2.18. Let $P \vdash \Gamma$ be an F-prenet and let $\pi$ be a FL proof without cuts whose F-prenet is $P \vdash \Gamma$, with $P = (P, Compp, Gen_P)$.

1. We denote the number of literal occurrences in $\Gamma$ (= elements of $P$) by $|L(\Gamma)| = |P|$.
2. $|Compp|$ stands for number of components of the linking $P$.
3. The sum of all genera in $P$ is denoted by $|Gen_P|$, i.e.

$$|Gen_P| = \sum_{(C_i, G_i) \in P} G_i = \sum_{(C_i, G_i) \in P} Gen_P(C_i).$$
|        | $|P|$  | $|\text{Comp}_P|$ | $|\text{Gen}_P|$ |
|--------|-------|------------------|-----------------|
| $Ax$   | +2    | +1               | 0               |
| MulWeak| $\geq +1$ | +1              | $\geq 0$        |
| $Y_d$  | −1    | 0                | +1              |
| $\sim_c$ | −2    | −1               | $\leq 0$        |
| $\sim_s$ | −2    | 0                | +1              |
| $\sim_d$ | −2    | −1               | 0               |

Figure 10: Action of the FL rules and transformations on linkings. The precise use of $\sim$ transformations will be demonstrated later.

4. Given the FL proof $\pi$, the number of applications of axioms in $\pi$ is denoted by $|Ax|$, while $|\text{MulWeak}|$ and $|\mathcal{Y}|$ are defined to be

$$|\text{MulWeak}| = \sum_{\Delta \text{ is introduced in } \pi \text{ by MulWeak}} |\mathcal{L}(\Delta)|, \quad |\mathcal{Y}| = \sum_{A^{l-m} \text{ obtained by Contron}A^{l-m}} |\mathcal{L}(A^{l-m})|,$$

respectively. As $\mathcal{Y}$ includes two different types of contractions, we denote accordingly the number of corresponding rule applications, i.e. we have $|\mathcal{Y}| = |\mathcal{Y}_s| + |\mathcal{Y}_d|$.

Figure 10 represents the action of an individual rule applications on a linking (we ignore the transformation $\sim$ for the moment) as a combination of increases/decreases of $|P|$, $|\text{Comp}_P|$, $|\text{Gen}_P|$, number of literals in the linking, number of components, and/or genus, respectively.

**Definition 2.19 (Characteristic of an F-prenet).** Given an F-prenet $P \rhd \Gamma$, we call the value

$$\text{Char}(P \rhd \Gamma) = |P| - |\text{Comp}_P| + |\text{Gen}_P|$$

the characteristic of the prenet.

If one views linking of an F-prenet as a one-dimensional topological space, characteristic of a prenet is close to the Euler’s characteristic of a space. It is an intrinsic property of an F-prenet which is read-out immediately from it.

The following observation is easy to deduce from Figure 10.

**Proposition 2.20 (Counting axiom links in an F-prenet).** If an F-prenet $P \rhd \Gamma$ corresponds to a FL proof, then

$$|Ax| \leq |P| - |\text{Comp}_P| + |\text{Gen}_P| = \text{Char}(P \rhd \Gamma).$$

Thus, given a (cut-free) F-prenet, one can conclude immediately an upper bound on the number of applications of the axiom rule in the corresponding FL proof, if such exists.

Some more work needs to be done to be able to formulate the correctness procedure. To that purpose, we look at $\mathcal{FSynt}$.

**Definition 2.21.** Let $\Sigma^\bigcirc$, with possible subscripts, denote a sequence of cut formulas and $\Gamma, A, B$ be with no cut formulas.
1. (M): In the category \( \mathcal{F}\text{Synt} \), we define a family of elementary Mix cospans or elementary M cospans, i.e. diagrams of the form \( Y \to X \leftarrow Z \) to be

\[
\begin{array}{c}
P_1 \triangleright \Gamma, \Sigma_1^\triangleright & \xrightarrow{M : f} & P \triangleright \Gamma, \Sigma_1^\triangleright, \Sigma_2^\triangleright \\
\downarrow & & \downarrow \\
P_2 \triangleright \Gamma, \Sigma_2^\triangleright & \xleftarrow{M : r} & P_3 \triangleright \Gamma, \Sigma_2^\triangleright
\end{array}
\]

where \( P = P_1 \uplus P_r(\Gamma \vee \Gamma) \).

A general M-cospan, abbreviated to just M-cospan, denoted also by \( M \), is any cospan of \( \mathcal{F}\text{Synt} \) isomorphic to an elementary M-cospan, i.e. a cospan to the right of the following commutative diagram

\[
\begin{array}{c}
P_1 \triangleright \Gamma, \Sigma_1^\triangleright & \xrightarrow{M : f} & P_r \triangleright \Gamma, \Sigma_1^\triangleright, \Sigma_2^\triangleright \\
\downarrow & & \downarrow \\
P_2 \triangleright \Gamma, \Sigma_2^\triangleright & \xleftarrow{M : r} & P_3 \triangleright \Delta_1
\end{array}
\]

where the cospan to the left is an elementary M cospan.

2. (∧) We define an elementary ∧-cospans in \( \mathcal{F}\text{Synt} \) to be:

\[
\begin{array}{c}
P_1 \triangleright \Gamma, \Sigma_1^\triangleright, A \vee B, A & \xrightarrow{\Lambda : f} & P \triangleright \Gamma, \Sigma_1^\triangleright, A \vee B, \Sigma_2^\triangleright, \Gamma \\
\downarrow & & \downarrow \\
\Lambda : r & & \Lambda : r
\end{array}
\]

where \( P = P_1 \uplus P_r(\Gamma \vee \Gamma) \).

3. (⊕Y) The elementary ⨁Y cospan in \( \mathcal{F}\text{Synt} \) is:

\[
\begin{array}{c}
P_1 \triangleright \Gamma, \Sigma_1^\triangleright, A & \xrightarrow{\Phi : f} & P \triangleright \Gamma, \Sigma_1^\triangleright, \Phi, \Sigma_2^\triangleright \\
\downarrow & & \downarrow \\
\Phi : r & & \Phi : r
\end{array}
\]

where one has \( P = P_1 \uplus P_r(\Gamma \vee \Gamma) \).

By the analogy with the general M-cospan, a (general) ∧-cospan and a (general) ⨁Y-cospan, is any cospan of \( \mathcal{F}\text{Synt} \) isomorphic to an elementary ∧- or a ⨁Y-cospan, respectively.

The intention behind the definition of cospans in \( \mathcal{F}\text{Synt} \) is to handle places where a classical proof branches, i.e. the ∧-rule and the Mix rule cases. Contractions are implicitly accounted, and are permitted in restricted cases where the proof branchings occur. The following concept encompass precisely the invertible rules, those that have no logical content.

**Definition 2.22.** An anodyne map \( P \triangleright \Gamma \Rightarrow P \triangleright \Delta \) is a syntactic map that can be decomposed as an isomorphism followed by a sequence of ∨-introduction maps (which do not affect the linking, only the sequent)

\[
P \triangleright \Gamma \Rightarrow P \triangleright \Delta_1 \Rightarrow \cdots \Rightarrow P \triangleright \Delta_n = \Delta.
\]

There is an important anodyne map, \([P \triangleright \Gamma] = P \triangleright [\Gamma] \Rightarrow P \triangleright \Delta\) whose domain is the sequent \([\Gamma]\) where all outer disjunctions have been removed.

**Definition 2.23** (Correct F-nets). A correctness tree or correctness diagram \( \mathcal{T} \) is a diagram (functor) \( \mathcal{T} : \mathcal{T} \to \mathcal{F}\text{Synt} \) where \( \mathcal{T} \) is of the form.
Observation 2.26. If an F-prenet is correct, then so is any F-prenet occurring in its correctness diagram.

The definition we just gave should be understood as follows: an F-prenet is an F-net (as will be shown soon, this is same as saying 'comes from a proof'), if one can construct a canonical proof directed by the F-prenet. In a certain sense the definition of correctness tree corresponds to a logic where the only rules are axiom with weakening applied only in the beginning of a proof followed by a contraction, while cut, \( \land \)
and Mix coupled with contractions according to the cospan definitions, are followed by \( \lor \)-introductions in a specific order.

Since these rules are derivable in FL, it follows immediately that a correct net is provable in FL, and the details can be found in the proof of the following theorem. The converse is the proof that these rules are sufficient and just as powerful as FL and more, that the proof theory of the restricted logic is the same.

**Theorem 2.27 (Sequentialization).** For every F-net there is an FL proof that corresponds to that F-net. The converse also holds, i.e. given an FL proof, its F-prenet is in fact a correct F-net.

**Proof.** That a correct F-net corresponds to an FL proof is argued in the comment above, and the construction of a proof can be carried out as in Example 2.25.

For the interesting direction, we relay on the characterization of the FL proofs as CL+ proofs giving rise to same F-nets. Therefore, we give a proof by induction on length of a CL+ proof \( \pi \) whose F-prenet we want to show to be a correct F-net. Base of the induction, the case where \( \pi \) consists of a proof of the form \( \Gamma \vdash \Phi \). Fix a branch in \( \Gamma \) and let \( \check{\Gamma} \) be the tree obtained from \( \Gamma \) by replacing every node \( R \vdash \Phi, \Delta \) of the fixed branch by \( R \vdash Q \vdash \Delta, \Lambda \), and every other node \( S \vdash \Pi, \Psi \) of \( \Gamma \) by \( S \vdash \Sigma \vdash \Pi, \Psi, \Delta \). The claim follows from the virtue of the last rule applied in \( \pi \).

- **Weak+**: the last rule application is \( Q \vdash \Gamma, \Sigma \).

  By induction hypothesis, there is a correctness diagram \( \mathcal{T} \) for the F-prenet \( P \vdash \Gamma \) corresponding to the proof of \( Q \vdash \Gamma, \Sigma \). Fix a branch in \( \mathcal{T} \) and let \( \check{\mathcal{T}} \) be the tree obtained from \( \mathcal{T} \) by replacing every node \( \mathcal{T} \vdash \Delta, \Lambda \) of the fixed branch by \( R \vdash Q \vdash \Delta, \Lambda \), and every other node \( S \vdash \Pi, \Psi \) of \( \mathcal{T} \) by \( S \vdash \Sigma \vdash \Pi, \Psi, \Delta \). The claim follows from the virtue of the last rule applied in \( \pi \).

- **\( \lor \)-rule**: \( \vdash \Gamma, \Delta \lor \Sigma \) was concluded from \( \vdash \Gamma, \Delta \lor \Sigma \). By induction hypothesis, there is a tree \( \mathcal{T} \) from the definition of F-nets. The correctness diagram for \( \vdash \Gamma, \Delta \lor \Sigma \) is obtained from \( \mathcal{T} \) simply by replacing the root \( \vdash \Gamma, \Delta \lor \Sigma \) by \( \vdash \Gamma, \Delta \lor \Sigma \).

- **\( \land \)-rule**: \( \vdash \Gamma, \Delta \land \Sigma \) was concluded from \( \vdash \Gamma, \Delta \land \Sigma \). Again, by induction hypothesis, there are two correctness diagrams \( \mathcal{T}' \) and \( \mathcal{T}'' \) from the definition of F-nets for \( \vdash \Gamma, \Delta \land \Sigma \) and \( \vdash \Gamma, \Delta \lor \Sigma \) for proofs of \( \vdash \Gamma, \Delta \land \Sigma \) and \( \vdash \Gamma, \Delta \lor \Sigma \), respectively. We already showed in the **Weak+** case that \( \vdash \Gamma, \Delta \land \Sigma \) and \( \vdash \Gamma, \Delta \lor \Sigma \) are also correct with their correctness diagrams \( \vdash \Gamma, \Delta \land \Sigma \) and \( \vdash \Gamma, \Delta \lor \Sigma \) can now be obtained by pasting a \( \land \) cospan:

  A correctness diagram for \( \vdash \Gamma, \Delta \land \Sigma \) and \( \vdash \Gamma, \Delta \lor \Sigma \) can now be obtained by pasting a \( \land \) cospan:

- **Mix and Cut**: These cases are not that different to the conjunction case, due to the uniform nature of the definitions.

- **Contr**: Suppose \( \vdash \Gamma, A \) is inferred from \( \vdash \Gamma, A \lor A \). This case is a delicate one, and it should be obvious that if we prove the following
Lemma 2.28. If \( P \triangleright \Gamma, \Lambda, A, \Delta^{\Diamond} \) is correct, then \( P(A \land A) \triangleright \Gamma, \Lambda, \Delta^{\Diamond} \) is correct too.

then we have proved the induction step for \textit{Contr.} \hfill \Box

Proof of Lemma 2.28. This will be done by induction on the number \( \text{Char}(P \triangleright \Gamma, \Lambda, A, \Delta^{\Diamond}) \) of the leaves in the underlying graph of the correctness diagram \( \mathcal{T} \) of \( P \triangleright \Gamma, \Lambda, A, \Delta^{\Diamond} \).

For the induction base one has \( \mathcal{T} \) to be \( P \triangleright \Gamma, \Lambda, A^{1-2}, A^{3-4} \) \( \equiv \) \( P \triangleright \Gamma, \Lambda, A^{1-2}, A^{3-4} \), where \( P \) is \( \{\{(a, \pi), 0\}, \{(b), 0\}, \ldots, \{(b_k), 0\}\} \) and \( \{\Gamma, \Lambda, A, A^2\} \). This means that either \( a, \pi \) are not literals of \( A \), or \( A \) is precisely one of those two. In either case \( P\Lambda^{1-2} \triangleright \Gamma, \Lambda, A^{1-2}, A^{3-4} \) is a leaf of a correctness tree, with a linking \( \{\{(a, \pi), 0\}, \{(b), 0\}, \ldots, \{(b_k), 0\}\} \Lambda^1 \triangleright \Lambda^2 \Lambda^4 \), since the first set is of the form \( \{\{(a, \pi), 0\}, \{(c), 0\}, \ldots, \{(m), 0\}\} \).

For the induction step let \( P \triangleright \Gamma, \Lambda, A, \Delta^{\Diamond}_1, \Delta^{\Diamond}_2 \) be an F-net whose correctness diagram \( \mathcal{T} \) has \( n \) leaves. Three cases can be distinguished, depending on the nature of the branching of \( [P \triangleright \Gamma, \Lambda, A, \Delta^{\Diamond}_1, \Delta^{\Diamond}_2] \) in \( \mathcal{T} \).

1. (M) - the reduction is of \( M \) type.

\[
\begin{array}{c}
L \triangleright [\Gamma] \triangleright [\Lambda] \triangleright [A]_{11}, [A]_{21}, \Delta^{\Diamond}_1 \\
L \triangleright [\Gamma] \triangleright [\Lambda]_{11}, [A]_{21}, \Delta^{\Diamond}_1 \quad \text{M : } l \\
R \triangleright [\Gamma] \triangleright [\Lambda]_{11}, [A]_{21}, \Delta^{\Diamond}_2 \quad \text{M : } r
\end{array}
\]

Here, \( P = (L \cup R)(\Gamma \gamma \Gamma, [\Lambda]_{11} \gamma [A]_{11}, [A]_{21} \gamma [A]_{21}) \).

The two subtrees with \( L \triangleright [\Gamma] \triangleright [\Lambda]_{11}, [A]_{21}, \Delta^{\Diamond}_1 \) and \( R \triangleright [\Gamma] \triangleright [\Lambda]_{11}, [A]_{21}, \Delta^{\Diamond}_2 \) as roots are both correctness diagrams with strictly smaller number of leaves than \( \mathcal{T} \). By induction hypothesis, \( L([\Lambda] \gamma [A]) \triangleright [\Gamma], [\Lambda], \Delta^{\Diamond}_1 \) and \( R([A] \gamma [A]) \triangleright [\Gamma], [\Lambda], \Delta^{\Diamond}_2 \) are correct F-nets, and we have the correctness diagram

\[
\begin{array}{c}
\begin{array}{c}
L([\Lambda] \gamma [A]_{21}) \triangleright [\Gamma], [\Lambda]_{11-21}, \Delta^{\Diamond}_1 \\
\text{M : } l \\
R([A]_{11} \gamma [A]_{21}) \triangleright [\Gamma], [\Lambda]_{11-21}, [A]_{11} \gamma [A]_{11} \gamma [A]_{21} \gamma [A]_{21}
\end{array}
\end{array}
\]

To save space, we will leave out the F-prenet in the root of the correctness diagram, as it differs form the penultimate F-prenet by the lack of the square brackets on certain formulas.

What is left to show is that
\[
(L([\Lambda]_{11} \gamma [A]_{21}) \cup R([A]_{11} \gamma [A]_{21})) (\Gamma \gamma \Gamma, [\Lambda]_{11-21} \gamma [A]_{11-21} \gamma [A]_{11} \gamma [A]_{21} \gamma [A]_{21}) \gamma \Gamma, [\Lambda]_{11-21} \gamma [A]_{11} \gamma [A]_{21} \gamma [A]_{21} \gamma [A]_{21} \gamma \Gamma
\]

is equal to \( (L \cup R)([\Lambda]_{11} \gamma [A]_{11} \gamma [A]_{21} \gamma [A]_{21} \gamma [A]_{21} \gamma [A]_{21} \gamma [A]_{21}) \), but this follows from symmetry and associativity of \( \gamma \).

20
2. (Cut) This case is analogous to the previous one, as no contractions are applicable on cut formulas

\[ \vdash [A]^1, [A]^2, [\Gamma], \Delta^\Phi \]

\[ \vdash \Gamma \rightarrow [A]^1, [A]^2, [\Gamma], \Delta^\Phi, \Delta_2^\Phi \]

This case is very similar to the M case, with the difference being that $[\Gamma]$ and $\Gamma_r$ are not two occurrences of the same sequent, rather they differ in light of the $\land$ cospan family. Consequently, $[\Gamma]$ is not obtained by applying $Y$ to the two, it is obtained by conjunction and subsequent contractions, as presented in the $\land$-type cospan definition. The notation we use for the linkings is $(\Gamma \land \Gamma_r)$, rather than $(\Gamma \land \Gamma_r)$. Thus, we have that in the diagram above $P = (L \sqcup R) ([\Gamma] \land \Gamma, [A]^1, [A]^2, [A]^{2r})$.

In this case, the induction hypothesis states that $L([A] \land [A]) \triangleright [A], [\Gamma], \Delta_1^\Phi$ and $R([A] \land [A]) \triangleright [\Gamma_r], [A], [\Delta_2^\Phi]$. We then have the correctness diagram consisting of two diagrams with roots $L([A]^1 \land [A]^2) \triangleright [A]^{1-2r}, [\Gamma], \Delta_1^\Phi$ and $R([A]^{2r} \land [A]^2) \triangleright [\Gamma_r], [A]^{1r-2r}, [\Delta_2^\Phi]$, followed by abd $\land$-cospan with codomain

\[ L([A]^1 \land [A]^2) \sqcup R([A]^{1r} \land [A]^2r) \]

\[ ([\Gamma] \land \Gamma, [A]^{1-2r}, [A]^{1r-2r}) \triangleright [A]^{1-2r-1r-2r}, [\Gamma], \Delta_1^\Phi, \Delta_2^\Phi \]

followed by an anodyne map. The claim follows from symmetry and associativity of $Y$, since

\[ L([A]^1 \land [A]^2) \sqcup R([A]^{1r} \land [A]^2r) \]

\[ ([\Gamma] \land \Gamma, [A]^{1-2r}, [A]^{1-2r}) \triangleright [A]^{1-2r-1r-2r}, [\Gamma], \Delta_1^\Phi, \Delta_2^\Phi \]

4. ($\land$, second case) - the reduction is of $\land$ type, and the active conjunction formula is in, say $[A]^2$, i.e. $[A]^2$ is of the form $[B]^2$, $C \land D$ and we have the correctness diagram consisting of two diagrams with roots $L \triangleright [\Gamma], [B]^1, [C] \land [D], [B]^1l \land [D]^1l, \Delta_1^\Phi$ and $R \triangleright [D]^2, [C] \land [D]^2, [\Gamma], [B]^1, [C] \land [D]^1r, [B]^2r, \Delta_2^\Phi$, followed by an $\land$-cospan with codomain $P \triangleright [\Gamma]_l, [B]^{1r-2r}, [C]^{1r-1r} \land [D]^{1r-1r}, [B]^{2r-2r}, [C]^{2r-2r} \land [D]^{2r-2r}, [\Delta_1^\Phi, \Delta_2^\Phi])$.

Here,

\[ P = (L \sqcup R) ([\Gamma], [B]^1, [C] \land [D], [B]^1l \land [D]^1l, [B]^{2r} \lor [\Gamma], [B]^1, [C] \land [D]^1r, [B]^2r, [C] \land [D]^2) \]

\[ ([B]^{2r} \land [D]^{2r}) \triangleright [C]^{2r} \land [D]^{2r} \lor [\Gamma], [B]^{2r-2r} \land [D]^{2r-2r} \]

\[ ([C]^{2r} \land [D]^{2r}) \triangleright [C]^{2r} \land [D]^{2r} \lor [\Gamma], [B]^{2r-2r} \land [D]^{2r-2r} \]

\[ [\Delta_1^\Phi, \Delta_2^\Phi]) \]

The induction hypothesis yields correctness of $L([B] \land [B])(C \land D \land C \land D) \triangleright [\Gamma], [B], [C] \land [D], [C] \land [D]$, and $R([B] \land [B])(C \land D \land C \land D) \triangleright [\Gamma], [B], [C] \land [D], [C] \land [D]$, and we have the correctness diagram consisting of two diagrams with roots $R([B]^{2r} \land [B]^2r)\triangleright [D]^2r, [C] \land [D]^2r \land [D]^{2r-1r}, [\Gamma], [B]^{1r-2r}, [\Delta_2^\Phi]$ and $L([B]^{2r} \land [B]^2r)(C \land [D]^{2r} \land [D]^{2r}) \triangleright [\Gamma], [B]^{1r-2r}, [C]^{1r-1r} \land [D]^{1r-1r}, [\Delta_1^\Phi, \Delta_2^\Phi])$, followed by a general $\land$-cospan with codomain

\[ L([B]^{1r} \land [B]^2r)(C \land [D]^{2r} \land [D]^{2r}) \]

\[ \sqcup R([B]^{1r} \land [B]^2r)(C \land [D]^{2r} \land [D]^{2r}) \]

\[ ([\Gamma], [B]^{1r-2r}, [C]^{1r-1r} \land [D]^{1r-1r}, [\Delta_1^\Phi, \Delta_2^\Phi]) \]

and an anodyne map.
The claim now follows from symmetry and associativity of $\vee$, since
\[
\begin{align*}
((L \cup R) ([\Gamma^r], [B]^1, [C^1] \land [D^1], [B]^{2r} \land [\Gamma^r], [D^{2r}]) & \cup R([B]^1 \land [C^1])
\end{align*}
\]

is equal to
\[
\begin{align*}
P = (L \cup R) ([\Gamma^r], & [B]^1, [C^1] \land [D^1], [B]^{2r} \land [\Gamma^r], [D^{2r}])
\end{align*}
\]

\[
\begin{align*}
&\cup (C^{2l} \land D^{2r} \lor C^{2rr} \land D^{2rr}) (C^{2l} \land C^{2ll} \land D^{2rr} \lor C^{2l})
\end{align*}
\]

\[
\begin{align*}
&(D^{2l} \land D^{2r} \lor D^{2rr} \land D^{2rr}) (C^{2r} \land D^{2rr} \lor D^{2rr}).
\end{align*}
\]

\[\square\]

The proof above reflects the intention of the correctness procedure - it amounts to having a proof net-guided proof search procedure. The key point of the procedure is splitting of the linkings to facilitate branchings of $\land$, $Mix$ and $Cut$. The branchings are the only places where contractions are allowed, and they are performed in a uniform way. This is possible due to the fact that if we ‘overduplicate’, the excess formulas can be eliminated by weakenings in the leaves. Also, the correctness diagram represents a canonical proof for a given F-prenet. If such a proof does not exist, than no proof corresponds to the F-prenet.

In principle, on may try to establish a correctness procedure by taking one proof after another and checking if the assigned F-net is the give one. The obvious problem one faces there is the infinite search space; the number of proofs of a sequent is infinite. Even once we know the number of axioms is finite, there has to be a way to control the lengths of branches in a proof, i.e. one needs to be able to control the number of weakenings in a proof, which is potentially infinite. Once one tries to control these, one is bound to go back to something akin to the correctness criterion we have formulated.

It may appear that the proof we have just given is surprisingly unproblematic, even though there are some technical intricacies involved, and one needs to be pedantic with the nested inductions. However, it is the proper definitions that are essential, those which facilitate proofs. Here, had we chosen only slightly different way to construct the correctness diagram, e.g. had we chosen to make seemingly innocent modifications of the way we handle conjunctions:

\[
\begin{align*}
P_1 \triangleright \Gamma, A & \quad \land \quad A \land \Gamma
\end{align*}
\]

\[
\begin{align*}
P_2 \triangleright \Gamma, A \land B & \quad \land \quad P_2 \triangleright B, \Gamma
\end{align*}
\]

that is, without two repetitions of the conjunction formula of the codomain in the domains of the cospan maps, it is unclear how could the induction step be shown.

The procedure relies on ability to construct a correctness diagram given an F-prenet. The number of branchings/size of a correctness diagram is finite and is read-out from the F-net directly. We start by the F-prenet itself and its child node and proceed by expanding the tree. Having an intermediate tree, the choice for expansion is finite, as the number of candidates for active conjunction/cut formulas is finite, and the number of possible linkings for domains of the corresponding cospans is finite as well. The anodyne steps in construction are deterministic. Finally, as a consequence of Proposition 2.20 that connects a characteristic of an F-prenet with number of axioms of its proof, if such exist, in order to have a correctness diagram, the number of cospan constructions has a finite upper bound.

The discussion above is summed-up in the following:

**Theorem 2.29.** Given an F-prenet, its FL-correctness (FL-sequentializability) can be checked in finite time, i.e. the FL-correctness procedure is a decision procedure for the set of F-nets.

Notice how we choose to speak of a correctness *procedure*, rather than a *criterion*. This choice of terminology acknowledges the fact that the procedure is not formulated as a universally quantified sentence, the way correctness criteria are usually presented.
3. Cut Elimination.

Let us have another look now at cut elimination on F-nets. Following the cut elimination-as-composition ideology we follow, we have

\[ P \vdash \Gamma, A \bigotimes \overline{A} \rightarrow P(A \sim \overline{A}) \vdash \Gamma. \] (3.1)

The following follows trivially from the underlying categorical structure and the definition of the \( \sim \) transformation.

Observation 3.1. Cut elimination for F-prenets is terminating and confluent and yields a unique normal form for an F-prenet with cuts.

We now know that cut elimination for an F-prenet yields a normal form, i.e., given an F-prenet with cuts, there is unique F-prenet without cuts that the original one reduces to. However, the important question is: does the same apply to FL-correct F-nets? More concretely, starting from an F-net with cuts, does the cut elimination procedure result in another FL-correct F-net? The direct answer to the most ambitious version of this question is, somewhat disappointingly, negative. The counter-examples are many, and one is given in 11.

Figure 11: Cut elimination on FL-correct F-nets does not yield an FL-correct F-net. The dotted loop that is added is required to get an FL-correct net, but is not obtained from the composition.

The F-prenet obtained after eliminating the cut does not come from a proof, neither in the FL system, nor in a (non-symmetric) deep inference version of FL we could formulate. There is, however, an observation we can make about the counter-example: had we added a loop on the single component of the resulting F-prenet while keeping the partition of the literals in the linking the same, it would be FL-correct.

We perform further analysis along these lines. Before doing so, we some need additional conceptual tools.

Definition 3.2. Given a cut-free F-prenet \( P \vdash \Gamma \), by conjunctive switching we understand a choice of an immediate subtree for every \( \land \) node in the syntactic tree of \( \Gamma \). Given a conjunctive switching, let \( \Gamma' \) be the tree where all of the subtrees chosen by the switching in \( \Gamma \) are erased, and let \( L(\Gamma') \) denote the set of leaves of the tree. (Notice that \( \Gamma' \) is not a sequence of well-formed propositional formulas, it is rather a graph with nodes decorated by connectives and literals.) By switching of the F-prenet \( P \vdash \Gamma \) we understand the pair \( P \mid_{L(\Gamma')} \vdash \Gamma' \), for the choice of conjunctive switching of \( \Gamma \).

We say an F-prenet \( P \vdash \Gamma \) is sound if for any switching \( \Gamma' \) the linking \( P \mid_{L(\Gamma')} \) contains a component with atoms of opposite polarity. We refer to these atoms of opposite polarity as soundness witnesses for the switching.

Soundness of an F-prenet with cut formulas is defined in the same way, where, in addition, each switching deletes a subtree for every \( \bigotimes \) connective, i.e. by treating the cut symbol as conjunction.

The reader can recognize the Lamarche-Straßburger condition on B-nets appearing in [4], whose possible predecessor is the Danos-Regnier condition for proof nets for multiplicative linear logic, appearing in [8].

Proposition 3.3. A correct F-net is sound.
Proof. By induction on the correctness diagram. Leaves of the correctness diagram always contain a component with two atoms of the opposite polarity that are present in every switching. As for the inductive step, notice that Anodyne maps are unproblematic - disjunction introduction or a formula permutation does not affect switchings. As for cospans, we consider them as a rule application, Mix, conjunction, or a cut, followed by a series of contractions. By applying a Mix one notices that switching of the codomain F-prenet is obtained by pairing switchings of the premisses, with components that are soundness witnesses coming from either of the witnesses of the premisses. When encountering a conjunction rule, each switching of the F-prenet in the conclusion is a switching of one of the premisses, restricted to the corresponding F-prenet, and thus the soundness witness for the switching will come from one of the premisses. The same argument is applied to a cut. As for contractions, \( P(A \lor A) \Rightarrow \Gamma, A \Rightarrow A \), a switching \( \Gamma', A', A' \) of the conclusion generates a switching in the premise \( \Gamma', A', A' \), by duplicating the switching of the contracted formula. Then, for the switching of the conclusion witnessing components may come from the premise directly, if soundness witness includes literals of \( \Gamma' \) of opposite polarity. Otherwise, if soundness witnesses for the switching are literals of one of the \( A' \) in the premise, the same literals after contraction will provide a witness for the prenet in conclusion, since the literals will belong to the same component after switching.

In the following, when we write \( \Gamma \), we assume that \( \Gamma \) may contain cut formulas.

**Proposition 3.4.** Given a sound F-prenet \( P \Rightarrow \Gamma, A \Rightarrow \overline{A} \), performing the

\[
P \Rightarrow \Gamma, A \Rightarrow \overline{A} \Rightarrow P(A \Rightarrow \overline{A}) \Rightarrow \Gamma
\]

cut elimination step preserves its soundness.

**Proof.** We show that \( P(A \Rightarrow \overline{A}) \Rightarrow \Gamma \) is sound by induction on the complexity of \( A \).

Assume \( A = a \), the cut is on literals. Let \( \Gamma' \) be a switching of \( \Gamma \). The switching is extended to the switching of \( \Gamma, a \Rightarrow \pi \) by choice of either \( a \) or \( \pi \). For each of the two choices there is a pair of literals that are witnesses for soundness of the sequent given the switching. If it happens that a pair of witnesses is from \( \Gamma \), it is a pair of witnesses for the switching \( \Gamma' \) in \( P(A \Rightarrow \overline{A}) \Rightarrow \Gamma \). Alternatively, for the switching with \( a \), there is an atom \( x \) of \( \Gamma' \) in the same component as \( a \), with whom it witnesses the soundness for the switching. The switching with \( \pi \) similarly gives a witness \( y \) in the same component as \( \pi \). Finally, \( x \) and \( y \) are in the same component in \( P(A \Rightarrow \overline{A}) \), and are witnesses for soundness for the switching \( \Gamma' \).

Assume \( A = C \land D \) (the case \( A = C \lor D \) is the same one, since if \( A = C \land D \), then \( \overline{A} = \overline{C} \lor \overline{D} \)). Let again \( \Gamma' \) be a switching of \( \Gamma \). By assumption, \( P \Rightarrow \Gamma, (C \land D) \Rightarrow (\overline{C} \lor \overline{D}) \) is sound. The switching \( \Gamma' \) is extended to a switching of \( \Gamma, (C \land D) \Rightarrow (\overline{C} \lor \overline{D}) \) by a choice of exactly one of the following: \( C \) or \( D \) or both \( C \) and \( D \). Once the choice is made, it suffices to choose a switching for the chosen formula(s), one of the \( C \lor D \) or \( \overline{C} \lor \overline{D} \) for the respective formulas.

Consider now \( P \Rightarrow \Gamma, C \Rightarrow \overline{C}, D \Rightarrow \overline{D} \). A switching for \( \Gamma, C \Rightarrow \overline{C}, D \Rightarrow \overline{D} \) consists of a switching \( \Gamma' \), a choice of either \( C \) or \( \overline{C} \) and either \( D \) or \( \overline{D} \), and a subsequent switchings of the chosen formulas. When compared to the switchings of \( \Gamma, (C \land D) \Rightarrow (\overline{C} \lor \overline{D}), \) one sees readily that each switching of \( \Gamma, C \Rightarrow \overline{C}, D \Rightarrow \overline{D} \) yields a set of literals that is a superset of the set of literals in a switching of \( \Gamma, (C \land D) \Rightarrow (\overline{C} \lor \overline{D}) \). Thus, if \( P \Rightarrow \Gamma, (C \land D) \Rightarrow (\overline{C} \lor \overline{D}) \) is sound, so must be \( P \Rightarrow \Gamma, C \Rightarrow \overline{C}, D \Rightarrow \overline{D} \).

Now, the induction hypothesis on the complexity of cut formula applied twice shows that \( P(A \Rightarrow \overline{A}) \Rightarrow \Gamma \) which is \( P(C \land D \Rightarrow \overline{C} \lor \overline{D}) \Rightarrow \Gamma \) or \( P(C \Rightarrow \overline{C})(D \Rightarrow \overline{D}) \Rightarrow \Gamma \) is sound.

**Convention 3.5.** As we have established, there is a canonical way to perform cut elimination on F-prenets which yields normal forms that are attainable in a single step. Our main interest in the remaining sections of this chapter is to investigate the conditions under which FL-correct F-prenets with cuts yield cut-free
\textbf{FL-correct F-nets via cut elimination.} For this purpose it suffice to assume that the cut elimination is done on a pair of two cut-free F-prenets, i.e. it suffice to consider

\[ P \uplus Q \triangleright \Gamma, \Sigma, AX \uplus A \rightarrow P \uplus Q(AX \sim A) \triangleright \Gamma, \Sigma, \]

where \( P \triangleright \Gamma, AX \) and \( Q \triangleright A, \Sigma \) are without cut formulas. This is slightly less general than the cut elimination applied on an F-prenet with several cut formulas, but for the purposes stated here the assumption does not cause loss of generality. To see that this is the case, it suffice to look at correctness diagrams where cuts can be eliminated in succession right after they are introduced.

We finally take a look at the categorical structure (cut-free) F-prenets stand for.

\textbf{Definition 3.6 (Category of F-prenets).} The category of F-prenets, written \( F\text{-prenet} \), has sequences of propositional formulas as its objects, together with an empty formula \( \emptyset \).

A map \( A \rightarrow B \) is a cut-free F-prenet \( P \triangleright AX, B \).

The identity F-prenet for \( A \) is the F-prenet \( \text{Id}_A \triangleright AX, A \), where \( \text{Id}_A \) is the linking consisting of two-element components connecting the \( i \)-th literal (counted, say from left to right) in \( A \) with the \( i \)-th in \( AX \), with genus 0 everywhere.

Given \( P \triangleright AX, B : A \rightarrow B \) and \( Q \triangleright BY, C : B \rightarrow C \), the composition map is the F-prenet

\[ P \uplus Q(B \sim BY) \triangleright AX, C. \]

Every F-prenet \( P \triangleright \Gamma, \Sigma \) for any partition and any permutation of the formulas in \( \Gamma \cup \Sigma \) into \( \Gamma, \Sigma \), determines a map \( \Gamma \rightarrow \bigvee \Sigma \) in \( F\text{-prenet} \), where \( (\cdot) \) is the negation extended to sequents by \( AX, B := AX \land BY \), and where \( \bigvee (\cdot) \) is a transformation of a sequent into a disjunction formula replacing comma symbols by \( \lor' \).

Structure of \( F\text{-prenet} \) is well examined, up to equivalence of categories. A detailed investigation is carried out in [10] and we only highlight the key points here. The underlying categorical structure corresponds to the free category \( \text{FThFrob}(ATYP) \), which is the category \( \text{FThFrob} \) where every object generator is, in addition, assigned a label of an atom from \( ATYP \), or its negation. Also, as in definition of a F-prenet, we require connected components in maps of \( \text{FThFrob}(ATYP) \) to connect only nodes which are labelled by a same atom or its negation. Again, such a freely generated category is uniquely determined up to an equivalence by a choice of Frobenius algebras on literals. \( \text{FThFrob}(ATYP) \).

The previous proposition now shows that the category \( F\text{-prenet} \) has a subcategory of sound F-prenets, \( sF\text{-prenet} \).
Going back to the definition of a sound F-prenet, the reader may notice that the notion of the soundness depends only on components of a linking, but not on the genera.

Let us have a look at the forgetful functor from $F\text{ThFrob}(AT_{yp})$ to the category we call $BF\text{ThFrob}(AT_{yp})$, which forgets about genera. In other words, $BF\text{ThFrob}(AT_{yp})$ is $F\text{ThFrob}(AT_{yp})$ deprived of the genus information. The same way $F\text{ThFrob}(AT_{yp})$ gives rise to the category of the assigned prenets with linkings standing for maps from the monoidal unit in $BF\text{ThFrob}(AT_{yp})$. We call these $FB$-prenets, and their category is denoted by $FB$-prenet. We refer to $FB : \{ (C_1, G_1), \ldots, (C_k, G_k) \} \mapsto \{ C_1, \ldots, C_k \} \mapsto \Gamma$ as to a forgetful functor from $F$-prenet to $FB$-prenet.

The notion of a sound prenet can be repeated for $FB$-prenets, i.e. it is stable under the forgetful functor we have just defined. Consequently, the sound $FB$-prenets constitute a subcategory $sFB$-prenet of $FB$-prenet. (We remind the reader that the composition in $FB$-prenet differs from the composition in $F$-prenet as no information on genus is present).

We have already established that $FL$-correct F-nets are sound, and therefore, the image under the functor $FB$ into $FB$-prenet falls into the subcategory $sFB$-prenet, which is precisely the claim of the previous proposition.

To sum up, the following diagram commutes:

\[
\begin{array}{ccc}
\text{sF-prenet} & \xrightarrow{FB} & \text{F-prenet} \\
\downarrow{FB|sF-prenet} & & \downarrow{FB} \\
\text{sFB-prenet} & \xrightarrow{FB} & \text{FB-prenet}.
\end{array}
\]

We continue our investigation of cut elimination by introducing some more concepts.

**Definition 3.7.** We say that a category is order-enriched or ordered if there is a partial order defined on every hom-set, such that given three objects $X, Y, Z$ the composition function $\circ : \text{Hom}(X, Y) \times \text{Hom}(Y, Z) \to \text{Hom}(X, Z)$ is monotone in both variables.

Using the terminology of the enriched category theory, an ordered category is enriched over the monoidal category $(\text{Poset}, \times, \emptyset)$ of partial orders and monotone functions.

The following proposition is an exercise in understanding of the above definition and the definition of composition in F-prenet.

**Proposition 3.8.** Let $\leq$ be a partial order defined on the set of all F-prenets such that

1. $P \triangleright \Gamma \leq P' \triangleright \Gamma'$ implies $\Gamma = \Gamma'$,
2. given cut-free $P \triangleright \Gamma, A \leq P' \triangleright \Gamma', A$ and $Q \triangleright A, \Sigma$, one has

$$P \uplus Q(\overline{A} \sim A) \triangleright \Gamma, \Sigma \leq P \uplus Q(\overline{A} \sim A) \triangleright \Gamma', \Sigma.$$

Then $\leq$ induces an order enrichment on F-prenet.

The previous proposition enables us to generalize the notion of an order enrichment to the set of all F-prenets, not only on the two-formula F-prenets of F-prenet. Therefore, by “enrichment on the set of F-prenets” we mean the presence of an order satisfying the conditions of the previous proposition.
Definition 3.9. Given two F-prenets \( P \triangleright \Gamma \) and \( Q \triangleright \Gamma \) over the same sequent we write

\[ P \triangleright \Gamma \leq Q \triangleright \Gamma \]

when

1. the partitions of literals of \( \Gamma \) coincide in \( P \) and \( Q \),
2. for every component \((C, G)\) of \( P \), the corresponding component \((C, G')\) of \( Q \) is such that \( G \leq G' \).

If the sequent belongs to a calculus with a stoup the definitions are the same, with the stoup’s semicolon replaced by a comma.

To show that the above order is indeed an enrichment, we show

Proposition 3.10. For every three F-prenets \( P \triangleright \Gamma, A; Q \triangleright \Gamma, A \) and \( R \triangleright A, \Delta \);

\[ P \triangleright \Gamma, A \leq Q \triangleright \Gamma, A \text{ implies } P \triangleright R(A \sim \bar{A}) \triangleright \Gamma, \Delta \leq Q \triangleright R(A \sim \bar{A}) \triangleright \Gamma, \Delta. \]

Proof. By the definition of \( \sim \), the partitions of literals after the both \( \sim \) transformations remain the same, the genera created by the composition are the same, whereas those coming from the factors in the composition are added. The claim thus follows from the monotonicity of addition.

The following is a direct consequence of the previous proposition.

Theorem 3.11. The order \( \leq \) on F-prenets in F-prenet defines an order-enrichment of the category.

The enrichment we have just defined behaves well with respect to the FL-correctness.

Lemma 3.12. If an F-prenet \( P \triangleright \Gamma \) comes from a classical FL proof, that is, if it is FL-correct, then so is any F-prenet \( Q \triangleright \Gamma \), for which \( P \triangleright \Gamma \leq Q \triangleright \Gamma \).

Proof. Let \( P \) be \( \{(C_1, G_1), \ldots, (C_k, G_k)\} \) and \( Q \) be \( \{(C_1, G_1'), \ldots, (C_k, G_k')\} \), with \( G_i \leq G_i' \). It suffices to show the claim for the case \( G_1' = G_1 + 1, G_i' = G_i \) for \( i > 1 \).

Given \( \{(C_1, G_1), \ldots, (C_k, G_k)\} \triangleright \Gamma \), one can consider an FL proof that corresponds to it (out of many). Now, for \( (C_1, G_1) \), one can choose an axiom (out of possibly several), such that one of the literals of the rule is a domain of a proof-induced map in \( \mathcal{FS}ynt \) which has the codomain form the set \( C_1 \). In other words, one can pick an instance of the axiom rule that introduced an atom which is an ancestor in the history of the proof of an atom in \( C_1 \). Now, the proof whose F-prenet is \( \{(C_1, G_1'), \ldots, (C_k, G_k')\} \triangleright \Gamma \) is obtained by replacing the axiom by the proof corresponding to the doubling map from Figure 5, which yields correctness of the F-net \( \{(C_1, G_1 + 1), \ldots, (C_k, G_k)\} \triangleright \Gamma \). The process illustrated in the picture to the right of Figure 13.

The previous lemma shows that the set of FL-correct F-nets is up-closed for the \( \leq \) relation. The same set will also be closed under the following transformation.

Lemma 3.13. If an F-prenet \( \{(C_1, G_1), \ldots, (C_k, G_k)\} \triangleright \Gamma \) is FL-correct, then so is any F-prenet obtained by connecting several components, i.e. any

\[ \{(C_1, G_1), \ldots, (C_{i-1}, G_{i-1}), (C_i \cup C_{i+1}, G_i + G_{i+1}), (C_{i+2}, G_{i+2}), \ldots, (C_k, G_k)\} \triangleright \Gamma \]

is also FL-correct.
Proof. Given \( \{(C_1, G_1), \ldots, (C_k, G_k)\} \triangleright \Gamma \), consider a corresponding FL proof of the sequent \( \vdash \Gamma \). We know that for this proof there is one in \( \mathcal{CL}+ \) with the same F-net. Now, extend the \( \mathcal{CL}+ \) proof by the weakening \( \text{Weak}+ \):

\[
\vdash \Gamma \\
\{(0, \{a_1\}), \ldots, (0, \{a_{k-1}\}), (0, \{a_k, a_{k+1}\}), (0, \{a_{k+2}\}), \ldots, (0, \{a_n\})\} \vdash \Gamma, \Gamma
\]

where \( a_1, \ldots, a_m = \mathcal{L}(\Gamma^2) \), and \( a_k \) is a copy of an atom form \( C_i \), while \( a_{k+1} \) is a copy of an atom form \( C_{i+1} \). Now, for this \( \mathcal{CL}+ \) proof there is an FL proof with the same F-net. Finally, by performing contractions:

\[
\vdash \Gamma, \Gamma \\
\vdash \Gamma, \Gamma \text{ Contr} \]

one obtains an FL proof whose F-net is the desired one. The outline of the proof is depicted in Figure 13.

The previous lemmas motivates the following definition.

**Definition 3.14.** Given two F-prenets \( P \triangleright \Gamma \) and \( Q \triangleright \Gamma \) over the same sequent we write \( P \triangleright \Gamma \preceq Q \triangleright \Gamma \) when

1. the partition of literals in \( \Gamma \) is finer in \( P \) than in \( Q \), i.e. each component of \( Q \) can be obtained by connecting several components of \( P \)
2. for every component \((C, G)\) of \( Q \) for which \( C = C_1 \cup \cdots \cup C_k \), for components \((C_1, G_1), \ldots, (C_n, G_n)\) of \( P \), we have

\[
G \geq \sum_{i=1}^{n} G_i.
\]

If the sequent belongs to a calculus with a stoup the definitions are the same, with the stoup’s semicolon replaced by a comma.

By the previous lemmas, one has

**Theorem 3.15.** The set of FL-correct F-nets is closed under the \( \leq \) and \( \preceq \) orders.

Notice that if \( P \triangleright \Gamma \preceq Q \triangleright \Gamma \), then \( \text{Char}(P \triangleright \Gamma) \preceq \text{Char}(Q \triangleright \Gamma) \), but converse is not generally true.

**Theorem 3.16.** The order \( \preceq \) on F-prenets an order-enrichment on F-prenet.

**Proof.** What needs to be shown, by Proposition 3.8, is that the composition in F-prenet is monotone w.r.t. \( \preceq \). Indeed, let \( P \triangleright \Gamma, A \preceq Q \triangleright \Gamma, A \). Then, after composing with \( R \triangleright \Delta, \Delta \), finer partitions of literals

![Figure 13: Illustration of the proof of Lemma 3.12 (left) and Lemma 3.13 (right).](image-url)
in of $P$ remain finer in $P \sqsupset R(\Lambda \sim \Lambda) \triangleright \Gamma, \Delta$, and the increase of genus due to the composition is larger with $Q \sqsupset R(\Lambda \sim \Lambda) \triangleright \Gamma, \Delta$, since there is fewer cases where composition connects different components (see Figure 14). In the case the partitions of $P, Q$ coincide, $Q$ contributes, by definition, with more genera to the composition with $R \triangleright \Lambda, \Delta$.

![Figure 14](image-url)

Figure 14: If $P \times Q$ (both are over the same sequent) and $Q$ is more connected, by composition, $P$ can become as connected, but in that case $Q$ will keep larger genera. In this example: $l \geq k + m$, and after composition $l + 1 > k + m$.

\[ \square \]

**Theorem 3.17.** For every sound $F$-prenet $P \triangleright \Gamma$ there is an $F$-prenet $Q \triangleright \Gamma$ for which

\[ P \triangleright \Gamma \leq Q \triangleright \Gamma \]

and which is FL-correct.

**Proof.** By induction on number of connectives in $\Gamma$.

Let $P$ be $\{(C_1, G_1), \ldots, (C_k, G_k)\}$ and let $Q$ be $\{(C_1, G'_1), \ldots, (C_k, G'_k)\}$, with $G_i \leq G'_i$.

- there is a disjunction in $\Gamma$, i.e. $\Gamma$ is of the form $\Sigma, A \lor B$. Then,

  \[ \{(C_1, G_1), \ldots, (C_k, G_k)\} \triangleright \Sigma, A, B \text{ is also sound, since the switching of the two F-prenets coincide. By induction, there is } \{(C_1, G'_1), \ldots, (C_k, G'_k)\} \triangleright \Sigma, A, B \text{, with } G_i \leq G'_i, \text{ which is an FL-correct F-net, and then so is } \{(C_1, G'_1), \ldots, (C_k, G'_k)\} \triangleright \Sigma, A \lor B, \text{ by the disjunction rule.} \]

- there is a conjunction in $\Gamma$, i.e. $\Gamma$ is of the form $\Sigma, A \land B$. This time, one can notice, form a simple arguments on switchings that

  \[ \{(C_1^1, G_1), \ldots, (C_k^1, G_k)\} \triangleright \Sigma^1, A^1; \]

  \[ \{(C_1^2, G_1), \ldots, (C_k^2, G_k)\} \triangleright \Sigma^2, B^2; \]

  \[ \{(C_1^3, G_1), \ldots, (C_k^3, G_k)\} \triangleright \Sigma^3, A^3, B^3 \]

are also sound. By induction, there are

\[ \{(C_1^1, H_1), \ldots, (C_k^1, H_k)\} \triangleright \Sigma^1, A^1; \]

\[ \{(C_1^2, H_{k+1}), \ldots, (C_k^2, H_{2k})\} \triangleright \Sigma^2, B^2; \]

\[ \{(C_1^3, H_{2k+1}), \ldots, (C_k^3, H_{3k})\} \triangleright \Sigma^3, A^3, B^3 \]

with $G_i \leq H_{m,k+i}$, which are FL-correct F-nets. By performing two $\land$ and contractions on $\Sigma$’s, $A$’s and $B$’s in an underlying FL proof, one obtains a linking of an FL-correct F-net

\[ \{(C_1^1, H_1), \ldots, (C_k^3, H_{3k})\} \triangleright \Sigma^1, A^1, \Sigma^2, B^2, \Sigma^3, A^3, B^3 \]

\( (\Sigma^1 \land \Sigma^2 \land \Sigma^3, A^1 \land B^2 \land A^3 \land B^2) \).

The crucial observation here is that the contractions respect the linking partition with genera augmented, i.e. the result can be seen as

\[ \{(C_1, F_1), \ldots, (C_k, F_k)\}, \]

which shows that $\{(C_1, F_1), \ldots, (C_k, F_k)\} \triangleright \Sigma, A \land B$ is FL-correct, with $G_i \leq F_i$.  

29
• if $\Gamma$ is a set of literals, then one of the components, say $(C_1, G_1)$ is of the form

$$(\{a_1, \ldots, a^m, \pi^1, \ldots, \pi^n\}, G_1),$$

with, say, $m \geq n > 0$. Now, by taking $2n$ instances of the axiom rule, performing Mix $2m - 1$ times, contracting positive atoms $2m - 1$ times to obtain a single component with $m$ occurrences of $a$ and $2m$ occurrences of $\pi$, of genus 0, and finally, contracting $\pi$ occurrences $2m - n$ times, one obtains a proof whose FL-correct F-prenet is

$$(\{(a_1, \ldots, a^m, \pi^1, \ldots, \pi^n), 2m - n\} \triangleright a_1, \ldots, a^m, \pi^1, \ldots, \pi^n).$$

By what we have previously shown,

$$(\{(a_1, \ldots, a^m, \pi^1, \ldots, \pi^n), \text{max}(2m - n, G_1)\} \triangleright a_1, \ldots, a^m, \pi^1, \ldots, \pi^n)$$

is an FL-correct F-net. Finally, $(C_1, \text{max}(2m - n, G_1)), \ldots, (C_k, G_k)) \triangleright \Gamma$ is also FL-correct, since it comes from a Weak+ rule of CL+ applied in the end.

Let us reflect for a while on the last observations we made. To that purpose, consider the following: let $\Gamma$ be a sequent with cut formulas. We define $|\text{Cut}|$ to be

$$|\text{Cut}| = \sum_{A^k \in \pi^r} |\mathcal{L}(A^k)|.$$

Figure 10 shows the effect of operations on linkings, and in particular, the last two rows show the effect of cut elimination. More concretely, if $P_1 \triangleright \Gamma_1$ is an FL-correct F-net with cuts, according to Proposition 2.20 we have

$$|A_{\triangleright} \leq |P_1| - |\text{Comp}_{P_1}| + |\text{Gen}_{P_1}| = \text{Char}(P_1 \triangleright \Gamma_1).$$

Each time a $\triangleright$ is performed on a pair of formulas, the left side of the inequality decreases by the number of eliminated pairs of literals, thus, if $P_2 \triangleright \Gamma_2$ is the F-prenet obtained after cut elimination, one has

$$|A_{\triangleright} \leq |P_1| - |\text{Comp}_{P_1}| + |\text{Gen}_{P_1}| - |\text{Cut}|$$

$$\leq |P_2| - |\text{Comp}_{P_2}| + |\text{Gen}_{P_2}| = \text{Char}(P_2 \triangleright \Gamma_2).$$

The first $\leq$ is a consequence of the fact that characteristic of an F-prenet is an upper bound in the case of FL, as already established, while the second $\leq$ stands instead of the equation because cut elimination can remove components altogether in view of $\triangleright$, due to the thinness of the underlying category. This situation is depicted in Figure 15.
The proof nets we have defined can be seen as generalizations of those for multiplicative linear logic. The proof theory of multiplicative linear logic is sensitive to resources, understood as number of applications of the axiom rule in a proof. By design, F-nets differentiate between proofs that differ in number of axioms used. Moreover, we have defined the characteristic of an F-prenet, which is increased by 1 by a single instance of the axiom rule. This characteristic represents a lower-bound on the number of axioms used in an FL proof corresponding to the F-prenet, if such a proof exists. Cut elimination on a pair of literals, on the other hand, decreases the characteristic by 1 through connection of literals of opposite polarities, dually to the axiom, and perhaps more in view of thinness. Therefore, it makes sense to state

characteristic of an F-prenet keeps track of resources used in a proof.

Indeed, the first lemma out of the last two shows that if we have succeeded in building a proof out of pieces offered by an F-prenet, we are sure to succeed with an F-prenet with larger genera. Similar with the lemma that follows. Since increase of the genera means increase of the characteristic, just as connecting two mutually disconnected components, these two lemmas can be rephrased to saying that if a proof can be constructed using resources coming from an F-prenet, then, naturally, a proof can be constructed using more resources. An axiom introduces a unit of resources, whereas cut elimination is consuming resources. Finally, the previous theorem can be rephrased, with slight generalization

**Theorem 3.17., restated.** A sound F-prenet can be turned into an FL-correct F-net by adding resources.

Notice that one can speak of three dimensions of a resource - (number of) components, genera, and (number of) literals, with the last one being constant for all proofs of the given sequent. This, of course, concerns the cut free F-prenets/proofs.

Going back to FB-prenet, the category of F-prenets without the genus information, and its subcategory sFB-prenet of sound prenets from FB-prenet, the previous Theorem precisely states that sFB-prenet yields an interpretation of classical proofs for which the property of full completeness holds.

We have already established that the notion of correctness for F-prenets is heavily dependent on the deductive system in question. In the case of CL, we have encountered F-prenets for which there is no derivation in the calculus, but for which there exist derivations in a deep inference system. The same is for the case of FL and its deep inference version.

**Example 3.18.** We go back to the two F-prenets in Example 2.17. The F-prenet to the right in the example has the following KS proof (notice that we make no effort towards reducing the proof bureaucracy):
As for the F-prenet to the right, in a similar way as in the previous case, we realize that any Mix or conjunction cospan that would start a hypothetical correctness tree has to have non-sound F-prenets as domains of its legs. Thus, no (shallow) FL proof is possible. Any deep proof assigned to the F-prenet to the right has to have at most 3 \( \alpha \) \( \downarrow \) rules. To make the three components created by the rule applications into the required linking, one either performs two contractions on literals or a single contraction on two conjunctions of two literals. In both cases, obtaining a single component through conjunction there will leave a literal in a disjunctive context.

One possible view of the concept of resources as we have defined them is that they provide a semantical account of relative efficiency of deductive systems. More precisely, as we have shown, certain deductive systems, such as the deep inference ones, can provide proofs for certain configurations of linkings, whereas the standard sequent ones require additional resources. There could be certain potential for this line of investigation, and some remotely related ideas can be found in the work of Carbone [14].

***

Let us now summarize. What we have so far is a category of F-prenets F-prenet and its subcategory of sound F-prenets sF-prenet. Taking the image of sF-prenet under the forgetful functor that forgets about genera one gets a category of classical proof nets, for which full completeness holds. This category contains less information on proofs than the category of \( \beta \)-nets from [4]. In fact, a linking in a sound FB-prenet is a transitive closure of a linking in a B-net, hence one loses information on axiom links in this category.

FL-correct F-nets themselves do not form a category, at least not in the standard way of composing nets by connecting linkings, but we know that we can construct an FL-correct F-net out of a sound one by adding genera. In addition, FL-correct F-nets are closed under this adding of resources, so the question is what is the minimal amount of resources one needs to add to obtain an FL-correct F-net. While we do not have a generic answer to this question, we are still able to compute the minimal amount of resources for every sound F-prenet separately. Indeed, due to the decision procedure for FL-correct F-nets, by incremental adding of genera, we can find minimal FL-correct F-nets w.r.t. \( \leq \) relation.

Figure 16: The sound F-prenet without both the dotted loops is not FL-correct. If either of the dotted ones is added, resulting F-prenet is FL-correct.

These minimal FL-correct F-nets can be several, as shown in Figure 16.
Figure 17: Depending on the order of elimination of cuts, one has to add resources or not to. If the cut to the right is eliminated first on two FL-correct F-nets, genus needs to increased by 1 to get an FL-correct F-net. If the cut to the left is eliminated first, the dotted loop need not be added.

One may be tempted to start with FL-correct F-nets and then for every sound F-prenet that is obtained in the composition / cut elimination, to establish a policy of adding resources to it to remain in the realm of FL-correct F-nets. Unfortunately, the straightforward attempt proves to be faulty, since the composition/ cut elimination of ≤-minimal FL-correct F-nets turns to be non-associative, as in the F-prenets of Figure 17

Figure 18: The F-prenet obtained by composition/cut elimination is not the infimum of all FL-correct F-nets it is smaller than. The infimum is the F-net without the added dotted loop, which is not correct.

Also, one can hope that the sound F-prenet obtained from cut elimination on an FL-correct F-net is the infimum of all FL-correct nets it is ⪯-smaller than, but this is also not the case, as shown in the Figure 18

3.1. Cut elimination for F-nets

At this point our theory of proof nets for classical logic suggests that normal forms we assign to proofs with cuts, i.e. sound F-prenets obtained via cut-as composition normalization of F-prenets, do not correspond to any proofs that may be obtained by an effective cut elimination procedure at the level of proofs. In [2] Hyland argues that classical proof theory requires this kind of approach to cut elimination and proof identification, and certain denotational interpretations that have motivated this work suggest the same [15].

For these reasons, we are interested in a possibility of defining a way to eliminate cut in a correct F-net that is guaranteed to yield another correct F-net. Naturally, we want the cut elimination to be associative, and to have a unit. This requires defining a new composition in the category of F-prenets.

We remind the reader here that the focus of our interest here are the cut elimination steps performed on single cuts arising from pairs of cut-free F-prenets. Equivalently, one may assume that in what follows we restrict the logic so that once a Cut rule is applied, the only rule that can be used afterwards is Cut. This is the standard use of Cut in categorical interpretations.

Definition 3.19. A bonus $B$ is a function which is defined on all pairs $(P \triangleright \Gamma, C)$ where $P \triangleright \Gamma$ is a sound F-prenet and $C$ a component of $P$, and whose value $B(P \triangleright \Gamma, C)$ is a natural number.

When the context is clear we allow ourselves to drop either the components or the prenets they belong to.

Let $P \triangleright \Gamma$ be a sound F-prenet and $((C_1, G_1)...(C_k, G_k))$ be its linking. We define $[P]^B$ to be the linking $((C_1, G_1 + B(P \triangleright \Gamma, C_1)), (C_2, G_2 + B(P \triangleright \Gamma, C_2)), ..., (C_k, G_k + B(P \triangleright \Gamma, C_k)))$
and \([P \triangleright \Gamma]^B\) to be the corresponding sound F-prenet with the linking \([P]^B\).

A bonus is said to be fair if it is constant on components for a given sound F-prenet, in which case it depends only on the F-prenet, and we write \(B(P \triangleright \Gamma)\).

A bonus \(B\) is said to be winning if \([P \triangleright \Gamma]^B\) is always an FL-correct net.

According to Lemma 3.12, it is safe to assume that given a sound F-prenet a winning bonus is strictly positive on the components of the given F-prenet. Such a winning bonus \(B\) with values \(\geq 1\) on components of every F-prenet is said to be strictly winning.

In addition to the definition of \([P \triangleright \Gamma]^B\) we define \([P \triangleright \Gamma]_B\) as the F-prenet obtained by subtracting \(B(P \triangleright \Gamma)\) many loops from every component of \(P \triangleright \Gamma\), if possible, \(P \triangleright \Gamma\) otherwise.

**Definition 3.20.** For F-prenets \(P \triangleright \Gamma, A\) and \(Q \triangleright \bar{A}, \Sigma\) and a winning bonus \(B\), we define

\[
(P \triangleright \Gamma, A) \diamond (Q \triangleright \bar{A}, \Sigma) = \begin{cases} 
(P \triangleright \Gamma, A) \circ (Q \triangleright \bar{A}, \Sigma), & \text{if } (P \triangleright \Gamma, A) \text{ or } (Q \triangleright \bar{A}, \Sigma) \text{ is an identity F-prenet} \\
\lceil [P \triangleright \Gamma, A]_B \circ [Q \triangleright \bar{A}, \Sigma]_B \rceil^B, & \text{otherwise}
\end{cases}
\]

where the \(\circ\) is the standard ‘\(\triangleright\)’ composition.

So the diamond “eats” the last formula of the left sequent and the first one of the right sequent, leaving a concatenation of what is left. this notation is not completely satisfactory in general, but good enough for the purposes of this work.

The following is obvious.

**Proposition 3.21.** Given a winning bonus \(B\), its \(\diamond\) cut-elimination applied to two FL-correct F-nets yields a correct net.

**Theorem 3.22.** Given a strictly winning bonus \(B\), the \(\diamond\) cut-elimination on FL-correct F-nets is associative. Also, the \(\diamond\) cut-elimination on an F-prenet and an identity F-prenet yields the former one.

**Proof.** From the definition of the \(\diamond\) cut-elimination, it is clear that the identity map in F-prenet is the identity with respect to the \(\diamond\) cut-elimination as well. What is left to show is the associativity of the \(\diamond\) cut-elimination.

So, let \(P_3 \triangleright \Delta, B, P_2 \triangleright \bar{B}, \Sigma, C\) and \(P_1 \triangleright \bar{C}, \Gamma\) be the F-prenets on which two \(\diamond\) cut-eliminations are performed in different order. The first case to distinguish is the one where at least one of the \(\diamond\) cut-eliminations in \((P_3 \triangleright \Delta, B) \circ ((P_2 \triangleright \bar{B}, \Sigma, C) \circ (P_1 \triangleright \bar{C}, \Gamma))\) or \(((P_3 \triangleright \Delta, B) \circ (P_2 \triangleright \bar{B}, \Sigma, C)) \circ (P_1 \triangleright \bar{C}, \Gamma)\) is the upper case in the definition, i.e., one of the maps is the identity (which is the same for \(\circ\) and \(\diamond\)). From the definition, this results in the ‘\(\triangleright\)’ cut-elimination whose result is equal to the one of the maps composed. Whichever of the \((P_3 \triangleright \Delta, B), (P_2 \triangleright \bar{B}, \Sigma, C), (P_1 \triangleright \bar{C}, \Gamma)\) is the identity, both \((P_3 \triangleright \Delta, B) \circ ((P_2 \triangleright \bar{B}, \Sigma, C) \circ (P_1 \triangleright \bar{C}, \Gamma))\) and \(((P_3 \triangleright \Delta, B) \circ (P_2 \triangleright \bar{B}, \Sigma, C)) \circ (P_1 \triangleright \bar{C}, \Gamma)\) reduce to the cut-elimination of the remaining two maps, showing the equality of the two F-prenets obtained by the cut-eliminations.

Assume now that neither of the \((P_3 \triangleright \Delta, B), (P_2 \triangleright \bar{B}, \Sigma, C), (P_1 \triangleright \bar{C}, \Gamma)\) is the identity. Then, the \(\diamond\) cut-eliminations are always those that include \(\lceil - \rceil_B^B\) and \(\lfloor - \rfloor_B^B\). Indeed, it can never happen that a \(\diamond\) cut-elimination produces an identity F-prenet unless both F-prenets that are cut are identity F-prenets. If neither of the composing F-prenets is an identity F-prenet, by the definition of \(\lceil - \rceil_B^B\), every component in the resulting F-prenet has genera \(\geq 1\). This is the point where strictness of the winning bonus \(B\) comes into play.
So, if neither of the \((P_3 \triangleright A, B), (P_2 \triangleright B, \Sigma, C), (P_1 \triangleright C, \Gamma)\) is the identity, the cut-elimination of performed in either order \((P_3 \triangleright A, B) \circ ((P_2 \triangleright B, \Sigma, C) \circ (P_1 \triangleright C, \Gamma))\) or \(((P_3 \triangleright A, B) \circ (P_2 \triangleright B, \Sigma, C)) \circ (P_1 \triangleright C, \Gamma)\) can be seen as 

\[\left[\left[ P_3 \triangleright A, B \right]_B \circ \left[ P_2 \triangleright B, \Sigma, C \right]_B \circ \left[ P_1 \triangleright C, \Gamma \right]_B \right]_B.\]

\(\square\)

Naturally, we are interested in concrete examples of winning bonuses that define \(\circ\) cut eliminations.

**Definition 3.23.** For a sequent (or a formula) \(\Gamma\), let \(|\Gamma \wedge|\) denote the number of conjunctions in \(\Gamma\) and let \(|\Gamma|\) denote the number of literals in \(\Gamma\). For an F-prenet \(P \triangleright \Gamma\) we define a fair bonus \(\text{Bon}\) to be the value

\[\text{Bon}(P \triangleright \Gamma) = |\Gamma| \cdot \left(2^{\lceil |\Gamma|+1\rceil} - 1\right).\]

Let \([P \triangleright \Gamma]\)\textsubscript{Bon} be the F-prenet obtained by adding \(\text{Bon}(P \triangleright \Gamma)\) many loops to every connected component of \(P \triangleright \Gamma\), and let \([P \triangleright \Gamma]\)\textsubscript{Bon} be the F-prenet obtained by subtracting \(\text{Bon}(P \triangleright \Gamma)\) many loops from every component of \(P \triangleright \Gamma\), if possible, \(P \triangleright \Gamma\) otherwise.

Where it does not cause confusion and where the sequent of an F-prenet can be deduced from the context, we introduce the notion \([P]\)\textsubscript{Bon} and \([P]\)\textsubscript{Bon} on linkings with the same meaning as above.

**Theorem 3.24.** The bonus \(\text{Bon}\) is strictly winning.

*Proof.* What needs to be shown is that for any sound net \(P \triangleright \Gamma\), \([P \triangleright \Gamma]\)\textsubscript{Bon} is FL-correct. From the definition it is clear that \(\text{Bon}\) is fair and strictly positive.

Obviously, the most difficult case in the previous lemma is the one where \(P \triangleright \Gamma\) has genus 0 everywhere, so we will assume that each component in \([P \triangleright \Gamma]\)\textsubscript{Bon} is of the genus \(\text{Bon}(P \triangleright \Gamma)\).

We show the lemma by induction on number of conjunctions in \(\Gamma\).

For the case of no conjunctions, \(P \triangleright \Gamma\) is by itself its own single conjunctive switching. As a sound net, it has contain a component with literals of opposite polarities. Thus, the F-prenet is assigned to a proof consisting of an axiom introducing the two literals of opposite polarities witnessing the soundness, followed by some weakenings and contractions that introduce other components and (possibly) connect the axiom component with others, and finally some disjunction introductions. Adding a non-zero amount of loops to an FL-correct F-net does not change correctness, so \([P \triangleright \Gamma]\)\textsubscript{Bon} is FL-correct as well.

Assume now that the claim of the theorem holds for all F-prenets with less than \(n\) conjunctions, let \(\Gamma\) be with \(n\) conjunctions, and let \(P \triangleright \Sigma, A \wedge B\)  

\(\Rightarrow\)  \(P \triangleright \Gamma\).

Both F-prenets \(P|_{\Sigma, \Lambda} \triangleright \Sigma, A\) and \(P|_{\Sigma, B} \triangleright \Sigma, B\) are sound, any switching of either of the two can be extended to a switching of \(P \triangleright \Sigma, A \wedge B\), which will provide a witness for the switching. The induction hypothesis on number of conjunctions in \(\Gamma\) yields FL-correctness of \([P|_{\Sigma, \Lambda} \triangleright \Sigma, A]\)\textsubscript{Bon} and \([P|_{\Sigma, B} \triangleright \Sigma, B]\)\textsubscript{Bon}.

Let \(Q \triangleright \Gamma\) be the F-prenet for which

\[Q \triangleright \Gamma \quad \triangleleft \quad ([P|_{\Sigma, \Lambda}]\textsubscript{Bon} \lor [P|_{\Sigma, B}]\textsubscript{Bon}) \triangleright \Sigma \triangleright \Sigma, A \wedge B.\]

The F-prenet \(Q \triangleright \Gamma\) is obviously FL-correct – an assigned proof is obtained by performing a conjunction followed by contractions and disjunction introductions on two proofs assigned to FL correct F-nets \([P|_{\Sigma, \Lambda} \triangleright \Sigma, A]\)\textsubscript{Bon} and \([P|_{\Sigma, B} \triangleright \Sigma, B]\)\textsubscript{Bon}.

If we manage to show \(Q \preceq [P]\)\textsubscript{Bon}, then the induction step will follow from the fact that \(\preceq\) preserves FL-correctness.

35
Notice here that the set of literals in \( Q \) is obviously \( P \). Also, notice that for every component in \([P|_{\Sigma, A} \triangleright \Sigma, A]|_{\text{Bon}}\) there is a component of \([P|_{\Sigma, B} \triangleright \Sigma, B]|_{\text{Bon}}\) such that the two are (disregarding the genus for the moment) obtained by restrictions of the same component in \( P \). Consequently, components of the two \( F \)-nets agree on \( \Sigma \). We show by case analysis that every component \( C \) of \( P \) is a subcomponent of a component in \( P \), i.e. the partition of literals in \( Q \) is finer than in \( P \) (which is the same as in \([P]|_{\text{Bon}}\)).

So, let

1. \((C, G) \in Q\) be such that \( C \) contains at least one literal of \( \Sigma \). In this case \((C, G)\) is obtained by performing 
\((\Sigma \upharpoonright \Sigma)\) on disjoint union of two components, \((C_1, G_1) \in [P|_{\Sigma, A} \triangleright \Sigma, A]|_{\text{Bon}}\) and \((C_2, G_2) \in [P|_{\Sigma, B} \triangleright \Sigma, B]|_{\text{Bon}}\). As noticed, the two components must be obtained as restrictions of a same component in \( P \) and they agree on literals in \( \Sigma \). So, we conclude that \( C \) has exactly the same set of literals as the component in \( P \) whose restrictions are \( C_1 \) and \( C_2 \):

2. \((C, G) \in Q\) be such that \( C \) contains only literals of \( A \) and \( B \). First of all, as \([P|_{\Sigma, A} \triangleright \Sigma, A]|_{\text{Bon}}\) contains no literals of \( B \) and \([P|_{\Sigma, B} \triangleright \Sigma, B]|_{\text{Bon}}\) contains no literals of \( A \), \( C \) can contain literals from either \( A \), or literals form \( B \), but not form both in the same time. Moreover, \( C \) is equal to a component of one of the the two restriction linkings \([P|_{\Sigma, A} \triangleright \Sigma, A]|_{\text{Bon}}\) or \([P|_{\Sigma, B} \triangleright \Sigma, B]|_{\text{Bon}}\), and is thus a subset of a component in \( P \).

We have shown that partition of literals in \( Q \) is finer than the one of \( P \). What is left to show is that the relation on genera is the one required by the \( \preceq \) order. To that purpose, we revisit the two cases above:

1. \((C, G) \in Q\) is such that \( C \) contains at least one literal of \( \Sigma \). In this case we have shown that \( C \) is also a class in \( P \). Let us compute \( G \). First, we know genera of \([P|_{\Sigma, A} \triangleright \Sigma, A]|_{\text{Bon}}\) and \([P|_{\Sigma, B} \triangleright \Sigma, B]|_{\text{Bon}}\) (keep in mind, we have assumed genus in \( P \) to be 0 everywhere):

\[\begin{align*}
\text{Bon}(P|_{\Sigma, B} \triangleright \Sigma, B) &= ((|\Gamma| - |A|) \cdot (2^{r|\wedge| - |A|} - 1)) \quad \text{and} \\
\text{Bon}(P|_{\Sigma, A} \triangleright \Sigma, A) &= ((|\Gamma| - |B|) \cdot (2^{r|\wedge| - |B|} - 1)).
\end{align*}\]

Obviously, \( G \) is the the sum of the two values above, augmented by the number of loops created by composition. The latter is not greater than the number of the elements in \( C \) minus 1, so it can be bounded by \(|\Gamma| - 1\). So, one has:

\[\begin{align*}
(|\Gamma| - |A|) \cdot (2^{r|\wedge| - |A|} - 1) + (|\Gamma| - |B|) \cdot (2^{r|\wedge| - |B|} - 1) + |\Gamma| - 1 \\
&\leq 2 \cdot |\Gamma| \cdot 2^{r|\wedge| - 2} \cdot 2^{r|\wedge| - |\Gamma| + 1},
\end{align*}\]

where the last inequality follows from \( 1 < 2 \cdot 2^{r|\wedge|} \) for \(|r| \leq 1\). Thus, the component with genus in \([P]|_{\text{Bon}}\) that corresponds to \((C, G)\) has larger genus.

2. \((C, G) \in Q\) is such that \( C \) contains only literals of \( A \) and \( B \). As we have seen, this reduces to the case where \( C \) contains literals of only one of \( A \) and \( B \), and these in turn give rise to two cases:

(a) \( C \subseteq L(A) \) (symmetrically, \( C \subseteq L(B) \)) and \( C \) is equal to a component in \( P \). This case may arise when \( P \) contains a component from literals of \( A \) (resp. \( B \)) alone. In this case \( G \) is:

\[\text{Bon}(P|_{\Sigma, A} \triangleright \Sigma, A) = (|\Gamma| - |B|) \cdot (2^{r|\wedge| - |B|} - 1),\]

which is obviously not larger than \(|\Gamma| \cdot (2^{r|\wedge| + 1} - 1)\). Again, the component with genus in \([P]|_{\text{Bon}}\) that corresponds to \((C, G)\) has larger genus.

(b) \( C \subseteq L(A) \) (symmetrically, \( C \subseteq L(B) \)) and is a strict subset of a component in \( P \). This case arises when \( P \) contains a component from literals of both \( A \) and \( B \) and those alone. Here, \( C \) is a restriction of that component to \( A \) (resp. \( B \)) and the component of \( P \) can be recovered by merging its two restrictions, the one \( (\preceq C) \) and the one on \( B \). For the \( \preceq \) order, this translates to the requirement on genera:

\[\text{Bon}(P|_{\Sigma, A} \triangleright \Sigma, A) + \text{Bon}(P|_{\Sigma, B} \triangleright \Sigma, B) \leq \text{Bon}(P \triangleright \Gamma),\]

which follows from the case 1. above where we have proven a stronger claim:

\[\text{Bon}(P|_{\Sigma, A} \triangleright \Sigma, A) + \text{Bon}(P|_{\Sigma, B} \triangleright \Sigma, B) + |\Gamma| - 1 \leq \text{Bon}(P \triangleright \Gamma),\]

This shows that \( Q \triangleright \Gamma \preceq [P \triangleright \Gamma]|_{\text{Bon}} \), which yields FL-correctness of \([P \triangleright \Gamma]|_{\text{Bon}}\) in the induction step for the number of conjunctions, which completes the proof of the theorem. \(\square\)
Having that \( \leq \) order on F-prenets preserves correctness, one has

**Theorem 3.25.** *Bonus*  
\[ \text{Bon}(P \triangleright \Gamma) = |\Gamma| \cdot \left(2^{|\Gamma|} - 1\right). \]

is strictly winning.

This notion of a strongly winning bonus has the advantage of treating the both connectives identically.

The previous theorem yields a concrete example of an associative cut elimination \( \circ \) on the set of all F-prenets. There is also an identity with respect to the cut-elimination assigned to any F-prenet. So, if we consider only F-prenets with two-formula sequents, the cut-elimination entails a composition in the category of these prenets, the same way the \( \triangleright \) cut-elimination gives rise to F-prenet. The category of two-formula F-prenets and the concrete \( \circ \) composition induced by \( \text{Bon} \) (also, \( \text{Bon} \)) is denoted by \( \text{CorrFNet} \). The category \( \text{CorrFNet} \), according to the previous, enjoys a weak form of the full completeness property i.e. every map in \( \text{CorrFNet} \) is assigned to a proof.

There is, however, little we can say on the categorical structure of \( \text{CorrFNet} \). We know that it is not *-autonomous, as e.g. a map \( P \triangleright (A \lor C) \land \overline{B}, C \) with the same linking as \( P \triangleright A \lor C, B \lor C \) is not in general the result of the cut-elimination

\[
\left(P \triangleright Q \triangleright (A \lor C) \land \overline{B}, (B \lor C) \land \overline{B}\right) \circ \left(R \triangleright S \triangleright (B \lor C) \land \overline{B}, C\right),
\]

where it is clear what the \( Q, R, S \) are, they are linkings of some identity F-prenets. The composition here in general requires adding bonuses on components with literals from \( A \) and \( C \). Moreover, \( \text{CorrFNet} \) is not even monoidal. In a monoidal category the functoriality of the tensor (which is here the disjoint union) translates to the ‘locality’ of the composition, i.e. a composition should alter only those components that are being connected by it. Here, a component that does not play a role in the concatenation of maps may have its genus changed, because of fairness of \( \text{Bon} \) and \( \text{Bon} \). A possible clue suggesting that the same situation has to happen for any definition of a composition via a winning bonus is provided in the example of Figure 16.

Another possible flaw in the definition of \( \text{CorrFNet} \) is the fact that the only isos are identities. This is clearly undesirable, as we would like to identify at least sequents which differ in e.g., formula permutations or disjunction introductions. This flaw cannot be fixed a priori for an arbitrary notion of a winning bonus, as, for instance, a winning bonus may be sensitive to order of formulas in a sequent. Fortunately, in the case of \( \text{Bon} \), one can change the definition of the \( \circ \) composition

\[
(P \triangleright \Gamma, A) \circ (Q \triangleright \overline{A}, \Sigma) = \begin{cases} (P \triangleright \Gamma, A) \circ (Q \triangleright \overline{A}, \Sigma), & \text{if is an iso in sF-prenet} \\ \text{Bon}(P \triangleright \Gamma, A) \circ \text{Bon}(Q \triangleright \overline{A}, \Sigma), & \text{otherwise} \end{cases}
\]

Associativity and the identity laws follow trivially from the definition of \( \text{Bon} \).

So far, our algebraic analysis of the semantics has always dealt with categories as suitable structures reflecting the symmetries of classical logic. At this point, however, a more general algebraic setting proves to be instrumental.

Given a proof \( f \) : \( \Gamma \vdash \Sigma \) in a two-sided derivation system, the categorical approach suggests that \( f \) should be a map in a category whose domain object is \( \Gamma \) and codomain \( \Sigma \).

In the cases where \( \Gamma \) is a sequence of formulas, and \( \Sigma \) is a single formula, as in systems for intuitionistic logic, one may generalize the categorical approach by reverting to *multicategories*, where a map is a set of morphisms sharing a codomain (see e.g. [16]). An alternative generalization is the concept of polycategories
for truly two-sided derivation systems. There, one generalize an object to a list of objects, and a map becomes a map between the lists. The concept is originally due to Szabo (see [17]), and the context of classical proof theory appears in the work of Robinson, Bellin, Hyland, and Urban [18].

Finally, in the case of one-sided sequent calculi, which is what we have here, the proper algebra for structural contexts is the notion of Lamarche structads, as described in [19, 20].

Going back to the structure with bonuses, in all the models we have seen until now, the categorical structure in question was sufficient to derive the generalizing structad, and vice versa. In the case here, due to the notion of the bonus being ‘global’, more structure is contained in the structad than in just the category.

Making use of this generalization is an important major task yet to be undertaken.