Combinatorial Proofs and Decomposition Theorems for First-order Logic

Dominic J. D. Hughes

Lutz Straßburger
Inria, Equipe Partout
Ecole Polytechnique, LIX
France

Jui-Hsuan Wu
Ecole Normale Supérieure
France

***Abstract*—We uncover a close relationship between combinatorial and syntactic proofs for first-order logic. Whereas syntactic proofs are formalized in some deductive proof system based on inference rules, a combinatorial proof is a "syntax-free" presentation of a proof that is independent from any set of inference rules. We show that the two proof representations are related via a deep inference decomposition theorem that establishes a new kind of normal form of syntactic proofs. This yields (a) a simple proof of the soundness and completeness of first-order combinatorial proofs, and (b) a full completeness theorem: every combinatorial proof is the image of a syntactic proof.**

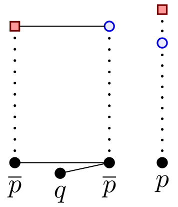
[[[**TODO:** Examples,examples,examples]]]

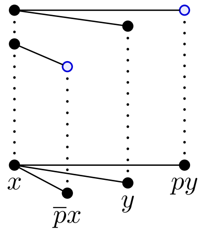
I. Introduction

First-order predicate logic is a cornerstone of modern logic. Since its formalisation by Frege [1] it has seen a growing usage in many fields of mathematics and computer science. Upon the development of proof theory by Hilbert [2], *proofs* became first-class citizens as mathematical objects that could be studied on their own. Since Gentzen's *sequent calculus* [3], [4], many other proof systems have been developed that allow the implementation of efficient proof search, for example *analytic tableaux* [5] or *resolution* [6]. Despite the immense progress made in proof theory in general and in the area of automated and interactive theorem provers in particular, we still have no satisfactory notion of proof identity for first-order logic. In this respect, proof theory is quite different from any other mathematical field. For example in group theory, two groups are *the same* iff they are isomorphic; in topology, two spaces are *the same* iff they are homeomorphic; etc. In proof theory, we have no such notion telling us when two proofs are *the same*, even though Hilbert was considering this problem as a possible 24th problem for his famous lecture [7] in 1900 [8], before proof theory existed as a mathematical field.

The main reason for this problem is that formal proofs, as they are usually studied in logic, are inextricably tied to the syntactic (inference rule based) proof system in which they are carried out. And it is difficult to compare two proofs that are produced within two different syntactic proof systems, based on different sets of inference rules. [[[**Lutz:** an example here?]]]

This is where *combinatorial proofs* come in. They have been introduced by Hughes [9] for classical propositional logic as a "syntax-free" notion of proof, and as a potential solution to Hilbert's 24th problem [10] (see also [11]). The basic idea is to abstract away from the syntax of the inference rules used in the proof and consider the proof as a combinatorial object, more precisely as a special kind of graph homomorphism. For example, a propositional combinatorial proof of Peirce's law $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$ is depicted below-left, a homomorphism from a coloured graph to a graph labelled with propositional variables.





Several authors have illustrated how syntactic proofs in various proof systems can be translated to propositional combinatorial proofs: for sequent proofs in [10], for deep inference proofs in [12], for Frege systems in [13], and for tableaux systems and resolution in [14]. This enables a natural definition of proof identity for propositional logic: two proofs are *the same*, if they are mapped to the same combinatorial proof.

Recently, Acclavio and Straßburger extended this notion to relevant logics [15] and to modal logics [16], and Heijlties, Hughes and Straßburger have provided combinatorial proofs for intuitionistic propositional logic [17].

In this paper we advance the idea that combinatorial proofs can provide a notion of proof identity for first-order logic. *First-order combinatorial proofs* were introduced by Hughes in [18]. For example, a first-order combinatorial proof of Smullyan's "drinker paradox" $\exists x(px \Rightarrow \forall y\, py)$ is shown above-right, a homomorphism from a partially coloured graph to a labelled graph. However, even though Hughes proves soundness and completeness, the proof is highly unsatisfactory: (1) the soundness argument is extremely long, intricate and cumbersome, and (2) the completeness proof does not allow a syntactic proof to be read back from a combinatorial proof, i.e., completeness is not *full* [**?**].[[[**TODO:** cite something for full comp?]]][[[**Lutz:** is that enough?]]] A fundamental problem is that not all combinatorial proofs can be obtained as translations of sequent calculus proofs.

In this paper we solve this issue by moving to a deep inference system. More precisely, we introduce a new proof system, called KS1, for first-order logic, that (a) reflects every combinatorial proof, i.e., there is a surjective mapping from proofs in KS1 to combinatorial proofs, and (b) yields far simpler proofs of soundness and completeness of combinatorial proofs, and (c) admits new decomposition theorems establishing a precise correspondence between certain syntactic inference rules and certain combinatorial notions.

In general, a *decomposition theorem* provides normal forms of proofs, separating subsets of inference rules of a proof system. A prominent example of a decomposition theorem is Herbrand's theorem [19], which allows a separation between the propositional part and the quantifier part in a first-order proof [4], [20]. Through the advent of deep inference, new kinds of proof decompositions became possible, most notably the separation between the linear part of a proof and the resource management of a proof. It has been shown by Straßburger [21] that a proof in classical propositional logic can be decomposed into a proof of (multiplicative) linear logic, followed by a proof consisting only of contractions and weakenings. In this paper we show that the same is possible for first-order logic.

Combinatorial proofs and deep inference can be seen as opposite ends of a spectrum: whereas deep inference allows for a very fine granularity of inference rules—one inference rule in a standard formalism, like sequent calculus or semantic tableaux, is usually simulated by a sequence of different deep inference rules—have combinatorial proofs completely abolished the concept of inference rule. Nonetheless, there is a close relationship between the two, realized through a decomposition theorem, as we establish it in this paper.

A. *Pictures that could be used elsewhere in the paper*

Condensed combinatorial proof of drinker formula(s):

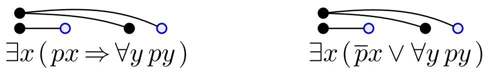
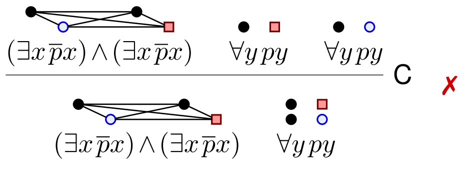
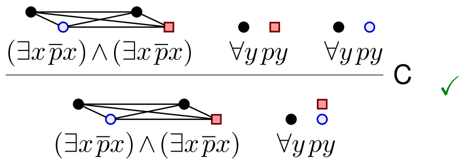

$\exists x\,(px \Rightarrow \forall y\, py)$
$\exists x\,(\overline{p}x \vee \forall y\, py)$

Fig. 1 is a floating figure with four combinatorial proofs.

Fig. 2 is a floating figure with the condensed forms of the four combinatorial proofs in Fig. 1.

Illustrating why we need to collapse twins during contraction, to preserve the target as a fograph:


$(\exists x\,\overline{p}x) \wedge (\exists x\,\overline{p}x)$
$\forall y\, py$
$\forall y\, py$
C
✗
$(\exists x\,\overline{p}x) \wedge (\exists x\,\overline{p}x)$
$\forall y\, py$


$(\exists x\,\overline{p}x) \wedge (\exists x\,\overline{p}x)$
$\forall y\, py$
$\forall y\, py$
C
✓
$(\exists x\,\overline{p}x) \wedge (\exists x\,\overline{p}x)$
$\forall y\, py$

Aligning both ● ■ and ● ○ over a single copy of $\forall y py$ results in two uncoloured vertices $\genfrac{}{}{0pt}{}{\bullet}{\bullet}$ over $\forall y$. The cover therefore fails to be a fograph: both uncoloured vertices are implicitly labelled with y and so are outer y-binders in the scope of each other. The correct operation is shown above-right, in which the troublesome pair is collapsed to a single uncoloured vertex ● over $\forall y$.

II. Preliminaries: First-order Logic

A. *Terms and Formulas*

Fix pairwise disjoint countable sets $\textsc{var} = \{x, y, z, \ldots\}$ of variables, $\textsc{fun} = \{f, g, \ldots\}$ of function symbols, and $\textsc{pred} = \{p, q, \ldots\}$ of predicate symbols. Each function symbol and each predicate symbol has a finite arity. The grammars below generate the set TERM of ***terms***, denoted by $s, t, u, \ldots$, the set ATOM of ***atoms***, denoted by $a, b, c, \ldots$, and the set FORM of ***formulas***, denoted by $A, B, C, \ldots$:

$$\begin{aligned} t &::= \ x \mid f(t_1, \ldots, t_n) \\ a &::= \ \mathsf{t} \mid \mathsf{f} \mid p(t_1, \ldots, t_n) \mid \overline{p}(t_1, \ldots, t_n) \\ A &::= \ a \mid A \wedge A \mid A \vee A \mid \exists x.A \mid \forall x.A \end{aligned}$$

where the arity of f and p is n. For better readability of often omit parentheses and write simply $ft_1, \ldots, t_n$ or $pt_1, \ldots, t_n$. We consider the truth constants t (*true*) and f (*false*) as additional atoms, and we consider all formulas in negation normal form, where ***negation*** $(\bar{\cdot})$ is defined on atoms and formulas via De Morgan's laws:

$$\begin{array}{lll} \overline{\overline{a}} = a & \overline{\mathsf{t}} = \mathsf{f} & \overline{p(t_1, \ldots, t_n)} = \overline{p}(t_1, \ldots, t_n) \\ & \overline{\mathsf{f}} = \mathsf{t} & \overline{\overline{p}(t_1, \ldots, t_n)} = p(t_1, \ldots, t_n) \\ \multicolumn{2}{l}{\overline{\exists x.A} = \forall x.\overline{A}} & \overline{A \wedge B} = \overline{A} \ \vee \ \overline{B} \\ \multicolumn{2}{l}{\overline{\forall x.A} = \exists x.\overline{A}} & \overline{A \vee B} = \overline{A} \ \wedge \ \overline{B} \end{array}$$

Then we write $A \Rightarrow B$ as abbreviation for $\overline{A} \vee B$.

A formula is ***rectified*** if all bound variables are distinct from one another and from all free variables. Every formula can be transformed into a logically equivalent rectified form, by simply renaming the bound variables. If we consider formulas equivalent modulo α-conversion (renaming of bound variables), then the rectified form of a formula A is uniquely defined, and we denote it by $\widehat{A}$.

A ***substitution*** is a function $\sigma\colon \textsc{var} \to \textsc{term}$ that is the identity almost everywhere. We denote substitutions as $\sigma = [x_1/t_1, \ldots, x_n/t_n]$, where $\sigma(x_i) = t_i$ for $i = 1..n$ and $\sigma(x) = x$ for all $x \notin \{x_1, \ldots, x_n\}$. We write $A\sigma$ for the formula obtained from A by applying σ, i.e., by simultaneously replacing all occurrences of x_i by t_i. A ***variable renaming*** is a substitution ρ with $\rho(x) \in \textsc{var}$ for all variables x.

B. *Sequent Calculus* LK1

Sequents, denoted by $\Gamma, \Delta, \ldots$, are finite multisets of formulas, written as lists, separated by comma. The ***corresponding formula*** of a (non-empty) sequent $\Gamma = A_1, A_2, \ldots, A_n$ is the disjunction of its formulas: $\mathrm{fm}(\Gamma) = A_1 \vee A_2 \vee \cdots \vee A_n$. A sequent is ***rectified*** iff its corresponding formula is.

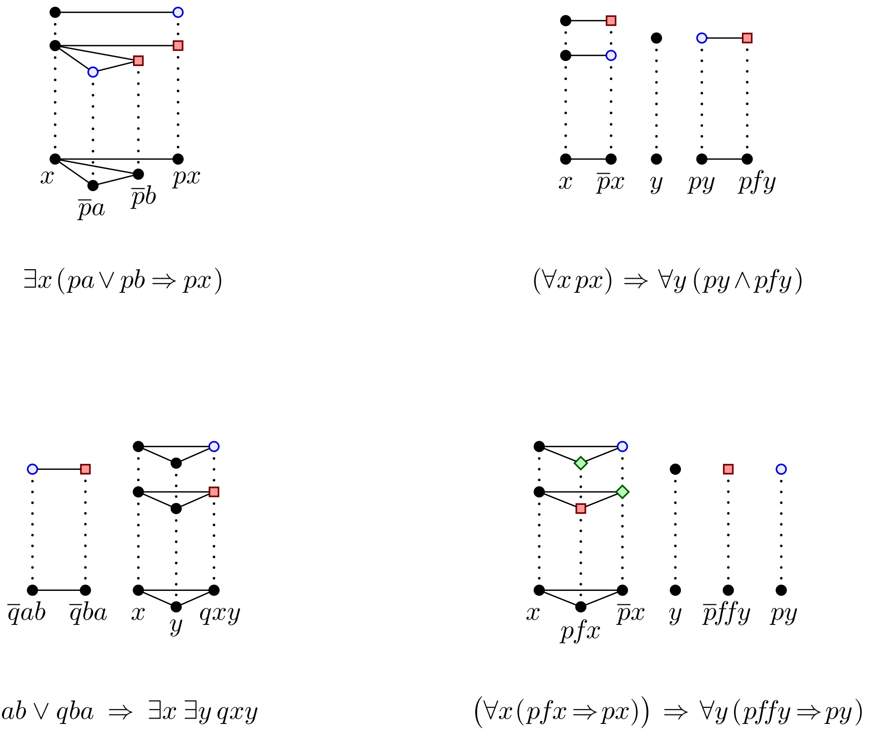

x
$\overline{p}a$
$\overline{p}b$
px
$\exists x\,(pa \vee pb \Rightarrow px)$
x
$\overline{p}x$
y
py
pfy
$(\forall x\,px) \Rightarrow \forall y\,(py \wedge pfy)$
$\overline{q}ab$
$\overline{q}ba$
x
y
qxy
$qab \vee qba \;\Rightarrow\; \exists x\,\exists y\,qxy$
x
pfx
$\overline{p}x$
y
$\overline{p}ffy$
py
$\big(\forall x\,(pfx \Rightarrow px)\big) \Rightarrow \forall y\,(pffy \Rightarrow py)$

Fig. 1. Four combinatorial proofs, each shown above the formula proved. Here x and y are variables, f is a unary function symbol, a and b are constants (nullary function symbols), p is a unary predicate symbol, and q is a binary predicate symbol.

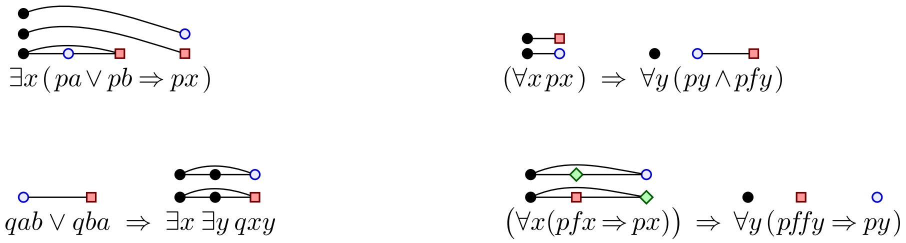

$\exists x\,(pa \vee pb \Rightarrow px)$
$(\forall x\,px) \Rightarrow \forall y\,(py \wedge pfy)$
$qab \vee qba \;\Rightarrow\; \exists x\,\exists y\,qxy$
$\big(\forall x(pfx \Rightarrow px)\big) \Rightarrow \forall y\,(pffy \Rightarrow py)$

Fig. 2. Condensed forms of the four combinatorial proofs in Fig. 1.

$$\mathsf{ax}\,\frac{}{\vdash a, \overline{a}} \qquad \mathsf{t}\,\frac{}{\vdash \mathsf{t}} \qquad \mathsf{mix}\,\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}$$

$$\vee\,\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B} \qquad \wedge\,\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \wedge B, \Delta}$$

$$\exists\,\frac{\vdash \Gamma, A[x/t]}{\vdash \Gamma, \exists x.A} \qquad \forall\,\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.A}\;(x \text{ not free in } \Gamma)$$

$$\mathsf{ctr}\,\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} \qquad \mathsf{wk}\,\frac{\vdash \Gamma}{\vdash \Gamma, A}$$

Fig. 3. Sequent calculi LK1 (all rules) and $\mathsf{MLL1}^{\mathsf{X}}$ (rules in the dashed box)

In this paper we use the sequent calculus LK1, shown in Figure 3, which is a one-sided variant of Gentzen's original calculus [3] for first-order logic. To simplify some technicalities later in this paper, we include the mix-rule.

Theorem 1. LK1 *is sound and complete for first-order logic.*

For a proof we refer to reader to any standard textbook, e.g. [22].

The linear fragment of LK1, i.e., the fragment without the rules ctr (*contraction*) and wk (*weakening*) defines *first-order multiplicative linear logic* [23], [24] *with mix* [25], [26] (MLL1+mix). We denote that system here with $\mathsf{MLL1}^{\mathsf{X}}$ (shown in Figure 3 in the dashed box).

We will employ the cut elimination theorem. The ***cut*** rule is

$$\mathsf{cut}\,\frac{\vdash \Gamma, A \quad \vdash \overline{A}, \Delta}{\vdash \Gamma, \Delta} \tag{1}$$

Theorem 2. *If a sequent $\vdash \Gamma$ is provable in* $\mathsf{LK1}{+}\mathsf{cut}$ *then it is also provable in* $\mathsf{LK1}$. *Furthermore, if $\vdash \Gamma$ is provable in* $\mathsf{MLL1}^{\mathsf{X}}{+}\mathsf{cut}$ *then it is also provable in* $\mathsf{MLL1}^{\mathsf{X}}$.

As before, this is standard, see e.g. [22] for a proof.

III. Preliminaries: First-order Graphs

A. Graphs

A ***graph*** $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ is a pair where $V_\mathcal{G}$ is a finite set of ***vertices*** and $E_\mathcal{G}$ is a finite set of ***edges***, which are two-element subsets of $V_\mathcal{G}$. We write vw for an edge $\{v, w\}$.

The ***complement*** of a graph $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ is the graph $\mathcal{G}^{\mathsf{C}} = \langle V_\mathcal{G}, E^{\mathsf{C}}_\mathcal{G}\rangle$ where $vw \in E^{\mathsf{C}}_\mathcal{G}$ iff $vw \notin E_\mathcal{G}$.

Let $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ and $\mathcal{H} = \langle V_\mathcal{H}, E_\mathcal{H}\rangle$ be graphs such that $V_\mathcal{G} \cap V_\mathcal{H} = \emptyset$. A ***homomorphism*** $\varphi\colon \mathcal{G} \to \mathcal{H}$ is a function $\varphi\colon V_\mathcal{G} \to V_\mathcal{H}$ such that if $vw \in E_\mathcal{G}$ then $\varphi(v)\varphi(w) \in E_\mathcal{H}$. The ***union*** $\mathcal{G} + \mathcal{H}$ is the graph $\langle V_\mathcal{G} \cup V_\mathcal{H}, E_\mathcal{G} \cup E_\mathcal{H}\rangle$ and the ***join*** $\mathcal{G} \times \mathcal{H}$ is the graph $\langle V_\mathcal{G} \cup V_\mathcal{H}, E_\mathcal{G} \cup E_\mathcal{H} \cup \{vw \mid v \in V_\mathcal{G}, w \in V_\mathcal{H}\}\rangle$. A graph $\mathcal{G}$ is ***disconnected*** if $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_2$ for two non-empty graphs $\mathcal{G}_1, \mathcal{G}_2$, otherwise it is ***connected***. It is ***coconnected*** if its complement is connected.

A graph $\mathcal{G}$ is ***labelled*** in a set L if each vertex $v \in V_\mathcal{G}$ has an element $\ell(v) \in L$ associated with it, its ***label***. A graph $\mathcal{G}$ is (partially) ***coloured*** if it carries a partial equivalence relation $\sim_\mathcal{G}$ on $V_\mathcal{G}$; each equivalence class is a ***colour***. A ***vertex renaming*** of $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ along a bijection $(\hat{\cdot})\colon V_\mathcal{G} \to \hat{V_\mathcal{G}}$ is the graph $\hat{\mathcal{G}} = \langle \hat{V_\mathcal{G}}, \{\hat{v}\hat{w} \mid vw \in E_\mathcal{G}\}\rangle$, with colouring and/or labelling inherited (i.e., $\hat{v} \sim \hat{w}$ if $v \sim w$, and $\ell(\hat{v}) = \ell(v)$). Following standard graph theory, we identify graphs modulo vertex renaming.

A ***directed graph*** $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ is a set $V_\mathcal{G}$ of ***vertices*** and a set $E_\mathcal{G} \subseteq V_\mathcal{G} \times V_\mathcal{G}$ of ***direct edges***. A ***directed graph homomorphism*** $\varphi\colon \langle V_\mathcal{G}, E_\mathcal{G}\rangle \to \langle V_\mathcal{H}, E_\mathcal{H}\rangle$ is a function $\varphi\colon V_\mathcal{G} \to V_\mathcal{H}$ such that if $(v, w) \in E_\mathcal{G}$ then $(\varphi(v), \varphi(w)) \in E_\mathcal{H}$.

B. Cographs

A graph $\mathcal{H} = \langle V_\mathcal{H}, E_\mathcal{H}\rangle$ is a ***subgraph*** of a graph $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ if $V_\mathcal{H} \subseteq V_\mathcal{G}$ and $E_\mathcal{H} \subseteq E_\mathcal{G}$. It is ***induced*** if $v, w \in V_\mathcal{H}$ and $vw \in E_\mathcal{G}$ implies $vw \in E_\mathcal{H}$. An induced subgraph of $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G}\rangle$ is uniquely determined by its set of vertices V and we denote it by $\mathcal{G}[V]$. A graph is ***$\mathcal{H}$-free*** if it does not contain $\mathcal{H}$ as an induced subgraph. The graph $\mathbf{P_4}$ is the (undirected) graph $\langle\{v_1, v_2, v_3, v_4\}, \{v_1v_2, v_2v_3, v_3v_4\}\rangle$. A ***cograph*** is a P_4-free undirected graph. The interest in cographs for our paper comes from the following well-known fact.

Theorem 3 ([27])**.** *A graph is a cograph iff it can be constructed from the singletons via the operations $+$ and $\times$.*

In a graph $\mathcal{G}$, the ***neighbourhood*** $N(v)$ of a vertex $v \in V_\mathcal{G}$ is defined as the set $\{w \mid vw \in E_\mathcal{G}\}$. A ***module*** is a set $M \subseteq V_\mathcal{G}$ such that $N(v) \setminus M = N(w) \setminus M$ for all $v, w \in M$. A module M is ***strong*** if for every module M', we have $M' \subseteq M$, $M \subseteq$ or $M \cap M' = \emptyset$. A module is ***proper*** if it has two or more vertices.

C. Fographs

A cograph is ***logical*** if every vertex is labelled by either an atom or variable, and it has at least one atom-labelled vertex. We write $\bullet\alpha$ for an α-labelled vertex. An atom-labelled vertex is called a ***literal*** and a variable-labelled vertex is called a ***binder***. A binder labelled with x is called an ***x-binder***. The ***scope*** of a binder b is the smallest proper strong module containing b. An ***x-literal*** is a literal whose atom contains the variable x. An x-binder ***binds*** every x-literal in its scope. In a logical cograph $\mathcal{G}$, a binder b is ***existential*** (resp. ***universal***) if, for every other vertex v in its scope, we have $bv \in E_\mathcal{G}$ (resp. $bv \notin E_\mathcal{G}$). An x-binder is ***legal*** if its scope contains no other x-binder and at least one literal.

Definition 4. A ***first-order graph*** or ***fograph*** is a logical cograph with legal binders. The ***binding graph*** of a fograph $\mathcal{G}$ is the directed graph $\vec{\mathcal{G}} = \langle V_\mathcal{G}, \{(b, l) \mid b \text{ binds } l\}\rangle$.

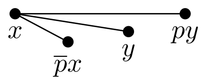
We define a mapping $[\![\cdot]\!]$ from formulas to (labelled) graphs, inductively as follows:

$$[\![a]\!] = \bullet a \quad \text{(for any atom } a\text{)}$$

$$\begin{array}{ll} [\![A \vee B]\!] = [\![A]\!] + [\![B]\!] & \qquad [\![\exists x.A]\!] = \bullet x \times [\![A]\!] \\ [\![A \wedge B]\!] = [\![A]\!] \times [\![B]\!] & \qquad [\![\forall x.A]\!] = \bullet x + [\![A]\!] \end{array}$$

where $\bullet a$ (resp. $\bullet x$) is a single-vertex graph whose vertex is labelled by a (resp. x).

Example 5. Below is the fograph of drinker formula $\exists x\,(px \Rightarrow \forall y\,py) = \exists x\,(\bar{p}x \vee \forall y\,py)$:



Lemma 6. *If A is a rectified formula then $[\![A]\!]$ is a fograph.*

Proof. That $[\![A]\!]$ is a logical cograph follows immediately from the definition and Theorem 3. The fact that every binder of $[\![A]\!]$ is legal can be proved by structural induction on A. □

Remark 7. Note that $[\![A]\!]$ is not necessarily a fograph if A is not rectified. If $A = (\forall x.p(x)) \vee (\forall x.q(x))$, then $[\![A]\!] = \bullet x \;\; \bullet p(x) \;\; \bullet x \;\; \bullet q(x)$, the scope of each x-binder contains all the vertices, in particular, the two x-binders. On the other hand, there are non-rectified formulas which are translated to fographs by $[\![\cdot]\!]$. For example, in the graph of $(\exists x.p(x)) \vee (\exists x.q(x))$, both x-binders are legal, as they are not in each other's scope: $x\bullet\!\!-\!\!\bullet px \;\; x\bullet\!\!-\!\!\bullet qx$.

We define a congruence relation $\equiv$ on formulas by the following equations:

$$\begin{array}{cc} A \wedge B \equiv B \wedge A & (A \wedge B) \wedge C \equiv A \wedge (B \wedge C) \\ A \vee B \equiv B \vee A & (A \vee B) \vee C \equiv A \vee (B \vee C) \\ \forall x.\forall y.A \equiv \forall y.\forall x.A & \forall x.(A \vee B) \equiv (\forall x.A) \vee B \\ \exists x.\exists y.A \equiv \exists y.\exists x.A & \exists x.(A \wedge B) \equiv (\exists x.A) \wedge B \end{array} \qquad (2)$$

where $x \notin fv(B)$ in the last two equations. Two formulas A and B are ***equivalent*** if $A \equiv B$. The following theorem

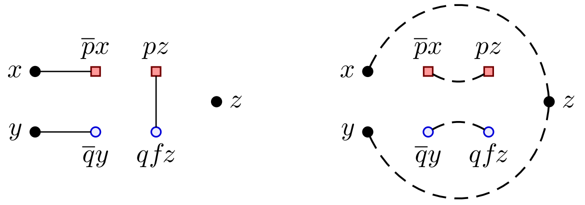


Fig. 4. A fonet (left) with dualizer $[x/z, y/fz]$ and its leap graph (right).

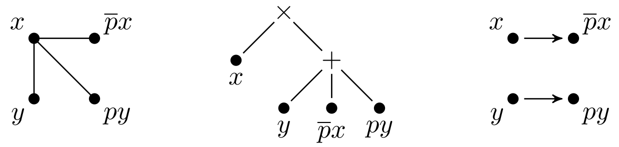
shows that the set of fographs can been seen as the quotient FORM$/\equiv$.

Theorem 8. *Let A, B be rectified formulas. Then*

$$A \equiv B \iff [\![A]\!] = [\![B]\!]$$

Proof. By a straightforward induction on A. □

Example 9. Both $\exists x(\bar{p}x \vee \forall y\, py)$ and $\exists x \forall y\, (py \vee \bar{p}x)$ (which are equivalent modulo $\equiv$) have the same (rectified) fograph $\mathcal{D}$, shown below-left.



In the middle above we show the *cotree* of the underlying cograph (illustrating the idea behind Theorem 3) and on the right above is its binding graph $\vec{\mathcal{D}}$.

IV. First-order Combinatorial Proofs

A. Fonets

Two atoms are ***pre-dual*** if their predicate symbols are dual (e.g. $p(x, y)$ and $\bar{p}(y, z)$) and two literals are ***pre-dual*** if their labels (atoms) are pre-dual. A ***linked fograph*** $\langle \mathcal{C}, \sim_{\mathcal{C}} \rangle$ is a coloured fograph $\mathcal{C}$ such that every colour (i.e., equivalence class of $\sim_{\mathcal{C}}$), called a ***link***, consists of two pre-dual literals, and every literal is either t-labelled or in a link.

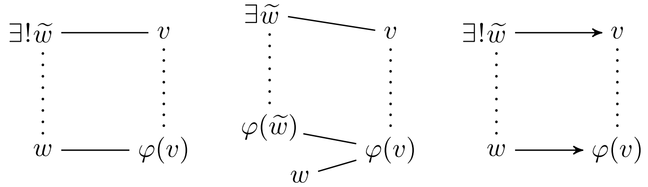
Let $\mathcal{C}$ be a linked fograph. The set of links can be seen as a unification problem by identifying dual predicate symbols. A ***dualizer*** of $\mathcal{C}$ is a substitution δ unifying all the links of $\mathcal{C}$. Since a first-order unification problem is either unsolvable or has a most general unifier, we can define the notion of ***most general dualizer***. A ***dependency*** is a pair $\{\bullet x, \bullet y\}$ of an existential binder $\bullet x$ and a universal binder $\bullet y$ such that the most general dualizer assigns to x a term containing y. A ***leap*** is either a link or a dependency. The ***leap graph*** $\mathcal{C}^{\mathsf{L}}$ of $\mathcal{C}$ is the undirected graph $\langle V_{\mathcal{C}}, L_{\mathcal{C}} \rangle$ where $L_{\mathcal{C}}$ is the set of leaps of $\mathcal{C}$. A vertex set $W \subseteq V_{\mathcal{C}}$ induces a ***matching*** in $\mathcal{C}$ if for all $w \in W$, $N(w) \cap W$ is a singleton. We say that W induces a ***bimatching*** in $\mathcal{C}$ if it induces a matching in $\mathcal{C}$ and a matching in $\mathcal{C}^{\mathsf{L}}$.

Definition 10. A ***first-order net*** or ***fonet*** is a linked fograph which has dualizer but no induced bimatching.

Figure 4 shows a fonet with a unique dualizer, and its leap graph.

B. Skew Bifibrations

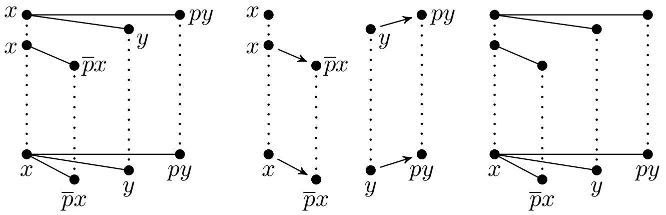
A graph homomorphism $\varphi\colon \langle V_{\mathcal{G}}, E_{\mathcal{G}} \rangle \to \langle V_{\mathcal{H}}, E_{\mathcal{H}} \rangle$ is a ***fibration*** if for all $v \in V_{\mathcal{G}}$ and $w\varphi(v) \in E_{\mathcal{H}}$, there exists a unique $\tilde{w} \in V_{\mathcal{G}}$ such that $\tilde{w}v \in E_{\mathcal{G}}$ and $\varphi(\tilde{w}) = w$ (indicated on the left below), and is a ***skew fibration*** if for all $v \in V_{\mathcal{G}}$ and $w\varphi(v) \in E_{\mathcal{H}}$ there exists $\tilde{w} \in V_{\mathcal{G}}$ such that $\tilde{w}v \in E_{\mathcal{G}}$ and $\varphi(\tilde{w})w \notin E_{\mathcal{H}}$ (indicated in the middle below). A directed graph homomorphism is a ***fibration*** if for all $v \in V_{\mathcal{G}}$ and $(w, \varphi(v)) \in E_{\mathcal{H}}$, there exists a unique $\tilde{w} \in V_{\mathcal{G}}$ such that $(\tilde{w}, v) \in E_{\mathcal{G}}$ and $\varphi(\tilde{w}) = w$ (indicated o the right below).



A ***fograph homomorphism*** $\varphi = \langle \varphi, \rho_{\varphi} \rangle$ is a pair where $\varphi\colon \mathcal{G} \to \mathcal{H}$ is a graph homomorphism between the underlying graphs, and ρ_{φ}, also called the ***substitution induced by*** φ is a variable renaming such that for all $v \in V_{\mathcal{G}}$ we have $\ell(\varphi(v)) = \rho_{\varphi}(\ell(v))$. Note that this entails that φ maps binders to binders and literals to literals. We say that a fograph homomorphism $\varphi\colon \mathcal{G} \to \mathcal{H}$ is ***existential-preserving*** if for all existential binders b in $\mathcal{G}$, the vertex $\varphi(b)$ is an existential binder in $\mathcal{H}$.

Definition 11. Let $\mathcal{G}$ and $\mathcal{H}$ be fographs. A ***skew bifibration*** $\varphi\colon \mathcal{G} \to \mathcal{H}$ is an existential-preserving fograph homomorphism that is a skew fibration on $\langle V_{\mathcal{G}}, E_{\mathcal{G}} \rangle \to \langle V_{\mathcal{H}}, E_{\mathcal{H}} \rangle$ and a fibration on the binding graphs $\vec{\mathcal{G}} \to \vec{\mathcal{H}}$.

Example 12. Below on the left is a skew bifibration, with its binding fibration in the middle. We often omit the labeling in the upper graph, as done on the right below.



Definition 13. A ***first-order combinatorial proof*** (***FOCP***) of a fograph $\mathcal{G}$ is a skew bifibration $\varphi\colon \mathcal{C} \to \mathcal{G}$ where $\mathcal{C}$ is a fonet. A ***first-order combinatorial proof*** of a formula A is a combinatorial proof of its graph $[\![A]\!]$.

Theorem 14 ([18])**.** *FOCPs are sound and complete for first-order logic.*

Remark 15. In or definition of FOCP, we are slightly laxer than the original definition of [18], as we allow for a variable renaming σ_{φ} which was forced to be the identity in [18].

V. First-order Deep Inference system KS1

Contrary to standard proof formalisms, like sequent calculi or tableaux, where inference rules decompose the principal

$$
\begin{array}{|c|}
\hline
\forall\,\dfrac{S\{\mathsf{t}\}}{S\{\forall x.\mathsf{t}\}} \qquad \mathsf{ai}\,\dfrac{S\{\mathsf{t}\}}{S\{a\vee\bar{a}\}} \qquad \mathsf{t}\,\dfrac{S\{A\}}{S\{A\wedge\mathsf{t}\}} \\[2ex]
\mathsf{s}\,\dfrac{S\{A\wedge(B\vee C)\}}{S\{(A\wedge B)\vee C\}} \qquad \mathsf{mix}\,\dfrac{S\{A\wedge B\}}{S\{A\vee B\}} \\[2ex]
\exists\,\dfrac{S\{A[x/t]\}}{S\{\exists x.A\}} \qquad \equiv\dfrac{S\{B\}}{S\{A\}}\,\text{(where } A\equiv B\text{)} \\
\hline
\end{array}
$$

$$
\mathsf{m}_\forall\,\dfrac{S\{(\forall x.A)\vee(\forall x.B)\}}{S\{\forall x.(A\vee B)\}} \qquad \mathsf{m}_\exists\,\dfrac{S\{(\exists x.A)\vee(\exists x.B)\}}{S\{\exists x.(A\vee B)\}}
$$

$$
\mathsf{m}\,\dfrac{S\{(A\wedge C)\vee(B\wedge D)\}}{S\{(A\vee B)\wedge(C\vee D)\}}
$$

$$
\mathsf{w}\,\dfrac{S\{A\}}{S\{A\vee B\}} \qquad \mathsf{ac}\,\dfrac{S\{a\vee a\}}{S\{a\}} \qquad \mathsf{c}_\forall\,\dfrac{S\{\forall x.\forall x.A\}}{S\{\forall x.A\}}
$$

Fig. 5. Deep inference systems KS1 (all rules) and MLS1$^{\mathsf{X}}$ (rules in the dashed box)

formula along its root connective, can *deep inference rules* be applied like rewriting rules inside any (positive) formula or sequent ***context***, which is denoted as $S\{\cdot\}$, and which is a formula (resp. sequent) with exactly one occurrence of the ***hole*** $\{\cdot\}$ in the position of an atom. Then $S\{A\}$ is the result of replacing the hole $\{\cdot\}$ in $S\{\cdot\}$ with A.

Figure 5 shows the inference rules for the deep inference system KS1 that we are using in this paper. It is a slight variation of the systems presented by Brünnler [28] and Ralph [29] in their PhD-theses. The main differences being that we have (i) the explicit presence of the mix-rule, (ii) a different choice of how the formula equivalence $\equiv$ is defined, and (iii) an explicit rule for the equivalence.

We consider here only the cut-free fragment, as cut-elimination for deep inference systems has already been discussed elsewhere (e.g. [20], [30]).[1]

As with the sequent system LK1, we also need for KS1 the *linear fragment*, that we call here MLS1$^{\mathsf{X}}$, and that is shown in Figure 5 in the dashed box.

We write $\begin{array}{c}B\\ \mathsf{S}\,\Big\|\,\Phi\\ A\end{array}$ to denote a derivation Φ from B to A using the rules from system S. A formula A is ***provable*** in a system S if there is a derivation in S from t to A.

In the course of this paper we are also going to make use of the general (non-atomic) version of the contraction rule:

$$
\mathsf{c}\,\dfrac{S\{A\vee A\}}{S\{A\}}
$$

[1] In the deep inference literature, the cut-free fragment is also called the *down-fragment*. But as we do not discuss the *up-fragment* here, we omit the down-arrows $\downarrow$ in the rule names.

VI. Main Results

We are now ready to see the main results of this paper. We only state them here and give the proofs in the later sections of the paper. The first one is routine and expected, but needs to be proved nonetheless:

Theorem 16. KS1 *is sound and complete for first-order logic.*

Our second result is more surprising, as it is a very strong decomposition result for first order logic.

Theorem 17. *For every derivation* $\begin{array}{c}\mathsf{t}\\ \mathsf{KS1}\,\Big\|\,\Phi\\ A\end{array}$ *there are formulas* $A_1,\ldots,A_5$*, such that there is a derivation:*

$$
\begin{array}{c}
\mathsf{t}\\
\{\forall,\mathsf{ai},\mathsf{t}\}\,\Big\|\\
A_5\\
\{\mathsf{s},\mathsf{mix},\equiv\}\,\Big\|\\
A_4\\
\{\exists\}\,\Big\|\\
A_3\\
\{\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\,\Big\|\\
A_2\\
\{\mathsf{ac},\mathsf{c}_\forall\}\,\Big\|\\
A_1\\
\{\mathsf{w},\equiv\}\,\Big\|\\
A
\end{array}
$$

This theorem is stronger than the existing decompositions for first-order logic, which either separated only atomic contraction and atomic weakening [28] or only contraction [29] or only the quantifiers in form of a Herbrand theorem [31], [29].

There is a weaker version of Theorem 17 that will also be useful:

Theorem 18. *For every derivation* $\begin{array}{c}\mathsf{t}\\ \mathsf{KS1}\,\Big\|\,\Phi\\ A\end{array}$ *there is a formula* A_1*, such that there is a derivation:*

$$
\begin{array}{c}
\mathsf{t}\\
\mathsf{MLS1}^{\mathsf{X}}\,\Big\|\\
A_1\\
\{\mathsf{w},\mathsf{c},\equiv\}\,\Big\|\\
A
\end{array}
$$

Let us now establish the connection between derivations in KS1 and combinatorial proofs.

Theorem 19. *Let* $\varphi\colon \mathcal{C}\to\mathcal{A}$ *be a combinatorial proof and let* A *be a formula with* $\mathcal{A}=[\![A]\!]$*. Then there is a derivation*

$$
\begin{array}{c}
\mathsf{t}\\
\mathsf{MLS1}^{\mathsf{X}}\,\Big\|\,\Phi_1\\
A'\\
\{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\,\Big\|\,\Phi_2\\
A
\end{array}
\qquad (3)
$$

for some $A' \equiv C\sigma_\varphi$ where C is a formula with $[\![C]\!] = \mathcal{C}$ and σ_φ is the variable renaming substitution induced by φ. Conversely, whenever we have a derivation as in (6) *above, then there is a combinatorial proof $\varphi\colon \mathcal{C} \to [\![A]\!]$ such that $\mathcal{C} = [\![\widehat{A'}]\!]$.*

Furthermore, in the proof of Theorem 19, we will see that (i) the links in the fonet $\mathcal{C}$ correspond precisely to the pairs of atoms that meet in the instances of the ai-rule in the derivation Φ_1, and (ii) the "flow-graph" of Φ_2 that traces the quantifier- and atom-occurrences in the derivation corresponds exactly to the vertex-mapping induced by φ.

Thus, combinatorial proofs are closely related to derivations of the form (6), and since by Theorem 17 every derivation can be transformed into that form, we can say that combinatorial proofs form a canonical proof representation for first order logic, similarly to what proof nets are for linear logic [32].

Finally, Theorems 16, 17 and 19 imply Theorem 14, which means that we have here an alternative proof of the soundness and completeness of first-order combinatorial proofs which is simpler than the one given in [18].

VII. Translating between LK1 and KS1

In this section we prove Theorems 16, 17, and 18, mainly by translating derivations to and from the sequent calculus, and by rule permutation arguments.

A. *The Linear Fragments* $\mathsf{MLL1^X}$ *and* $\mathsf{MLS1^X}$

In this section we show the equivalence of $\mathsf{MLL1^X}$ and $\mathsf{MLS1^X}$.

Lemma 20. *If $\vdash \Gamma$ is provable in $\mathsf{MLL1^X}$ then $\mathrm{fm}(\Gamma)$ is provable in $\mathsf{MLS1^X}$.*

Proof. This is a straightforward induction on the proof of $\vdash \Gamma$ in $\mathsf{MLL1^X}$, making a case analysis on the bottommost rule instance. We show here only the case of $\forall\,\dfrac{\vdash \Delta, A}{\vdash \Delta, \forall x.A}$ (all other cases are simpler and have been shown before, e.g. [28]): By induction hypothesis, there is a proof of $\mathrm{fm}(\Delta) \vee A$ in $\mathsf{MLS1^X}$. We can prefix every line in that proof by $\forall x$ and then compose the following derivation:

$$\begin{array}{c} \forall\,\dfrac{\mathsf{t}}{\forall x.\mathsf{t}} \\ \mathsf{MLS1^X}\,\Big\Vert \\ \equiv\,\dfrac{\forall x.\mathrm{fm}(\Delta) \vee A}{\mathrm{fm}(\Delta) \vee \forall x.A} \end{array}$$

where we can apply the $\equiv$-rule because x is not free in Δ. □

Lemma 21. *Let $\mathsf{r}\,\dfrac{S\{A\}}{S\{B\}}$ be an inference rule in $\mathsf{MLS1^X}$ other than* ai. *Then the sequent $\vdash \overline{A}, B$ is provable in $\mathsf{MLL1^X}$.*

Proof. This is a straightforward exercise that we leave to the reader. (Note that the ax-rule can be applied to $\vdash \mathsf{f}, \mathsf{t}$ in the cases of $\mathsf{r} = \forall$.) □

Lemma 22. *Let A, B be formulas, and let $S\{\cdot\}$ be a (positive) context. If $\vdash \overline{A}, B$ is provable in $\mathsf{MLL1^X}$, then so is $\vdash \overline{S\{A\}}, S\{B\}$.*

Proof. Straightforward induction on $S\{\cdot\}$. (see e.g. [33]) □

Lemma 23. *If a formula C is provable in $\mathsf{MLS1^X}$ then $\vdash C$ is provable in $\mathsf{MLL1^X}$.*

Proof. We proceed by induction on the number of inference steps in the proof of C in $\mathsf{MLS1^X}$. Consider the bottommost rule instance $\mathsf{r}\,\dfrac{S\{A\}}{S\{B\}}$. By induction hypothesis we have a $\mathsf{MLL1^X}$ proof Π of $\vdash S\{A\}$. If r is $\mathsf{ai}\,\dfrac{S\{\mathsf{t}\}}{S\{a \vee \overline{a}\}}$, we replace in Π all corresponding occurrences of t with $a \vee \overline{a}$ and the rule instance $\mathsf{t}\,\dfrac{}{\vdash \mathsf{t}}$ with the derivation $\vee\,\dfrac{\mathsf{ax}\,\dfrac{}{\vdash a, \overline{a}}}{\vdash a \vee \overline{a}}$. This gives us a proof of $\vdash S\{a \vee \overline{a}\}$. In all other cases, by Lemmas 21 and 22, we have a $\mathsf{MLL1^X}$ proof of $\vdash \overline{S\{A\}}, S\{B\}$. We can compose them via cut:

$$\mathsf{cut}\,\frac{\vdash S\{A\} \quad \vdash \overline{S\{A\}}, S\{B\}}{\vdash S\{B\}}$$

and then apply Theorem 2. □

B. *Contraction and Weakening*

The first observation here is that Lemmas 20–23 from above also hold for LK1 and KS1. We therefore immediately have:

Theorem 24. *For every sequent Γ, we have that $\vdash \Gamma$ is provable in* LK1 *if and only if $\mathrm{fm}(\Gamma)$ is provable in* KS1.

Then Theorem 16 is an immediate consequence. Let us now proceed with providing further lemmas that will be needed for the other results.

Lemma 25. *The* c*-rule is derivable in $\{\mathsf{ac}, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$.*

Proof. We show that there is always a derivation

$$\begin{array}{c} A \vee A \\ \mathsf{S}\,\Big\Vert \\ A \end{array}$$

, where $\mathsf{S} = \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$, by induction on A:

- If $A = a$, then we have $\mathsf{ac}\,\dfrac{a \vee a}{a}$.
- If $A = B \wedge C$, then we have $\begin{array}{c} \mathsf{m}\,\dfrac{(B \wedge C) \vee (B \wedge C)}{(B \vee B) \wedge (C \vee C)} \\ \mathsf{S}\,\Big\Vert \\ B \wedge (C \vee C) \\ \mathsf{S}\,\Big\Vert \\ B \wedge C \end{array}$.

- If $A = B \vee C$, then we have $$\begin{array}{rc} & (B\vee C)\vee(B\vee C)\\ \equiv & \overline{(B\vee B)\vee(C\vee C)}\\ \mathsf{s} & \Vert\\ & B\vee(C\vee C)\\ \mathsf{s} & \Vert\\ & B\vee C\end{array}\ .$$
- If $A = \exists x.A'$, then we have $$\begin{array}{rc} & (\exists x.A')\vee(\exists x.A')\\ \mathsf{m}_\exists & \overline{\exists x.(A'\vee A')}\\ \mathsf{s} & \Vert\\ & \exists x.A'\end{array}\ .$$
- If $A = \forall x.A'$, then we have $$\begin{array}{rc} & (\forall x.A')\vee(\forall x.A')\\ \mathsf{m}_\exists & \overline{\forall x.(A'\vee A')}\\ \mathsf{s} & \Vert\\ & \forall x.A'\end{array}\ .$$

⟦**TODO:** ⟧ ⟦**Jui-Hsuan:** done. Maybe just keep one case.⟧⟦**Lutz:** yes, but we do that at the end. don't think about space right now.⟧ □

Lemma 26. *$\mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists$ are derivable in $\{\mathsf{w}, \mathsf{c}, \equiv\}$.*

Proof. ⟦**TODO:** ⟧

We have the following derivations:

$$\begin{array}{rc} & \forall x.\forall x.A\\ \mathsf{w} & \overline{\forall x.((\forall x.A)\vee A)}\\ \equiv & \overline{(\forall x.A)\vee(\forall x.A)}\\ \mathsf{c} & \overline{\forall x.A}\end{array}\ (x\notin fv(\forall x.A))$$

.

$$\begin{array}{rc} & (A\wedge C)\vee(B\wedge D)\\ \mathsf{w} & \overline{((A\vee B)\wedge C)\vee(B\wedge D)}\\ \mathsf{w} & \overline{((A\vee B)\wedge(C\vee D))\vee(B\wedge D))}\\ \mathsf{w} & \overline{((A\vee B)\wedge(C\vee D))\vee((B\vee A)\wedge D))}\\ \mathsf{w} & \overline{((A\vee B)\wedge(C\vee D))\vee((B\vee A)\wedge(D\vee C))}\\ \equiv & \overline{((A\vee B)\wedge(C\vee D))\vee((A\vee B)\wedge(C\vee D))}\\ \mathsf{c} & \overline{(A\vee B)\wedge(C\vee D)}\end{array}$$

.

$$\begin{array}{rc} & (\exists x.A)\vee(\exists x.B)\\ \mathsf{w} & \overline{(\exists x.(A\vee B))\vee(\exists x.B)}\\ \mathsf{w} & \overline{(\exists x.(A\vee B))\vee(\exists x.(B\vee A))}\\ \equiv & \overline{\exists x.(A\vee B))\vee(\exists x.(A\vee B))}\\ \mathsf{c} & \overline{\exists x.(A\vee B)}\end{array}$$

.

$$\begin{array}{rc} & (\forall x.A)\vee(\forall x.B)\\ \mathsf{w} & \overline{(\forall x.(A\vee B))\vee(\forall x.B)}\\ \mathsf{w} & \overline{(\forall x.(A\vee B))\vee(\forall x.(B\vee A))}\\ \equiv & \overline{(\forall x.(A\vee B))\vee(\forall x.(A\vee B))}\\ \mathsf{c} & \overline{\forall x.(A\vee B)}\end{array}$$

. ⟦**Jui-Hsuan:** done. If needed, we can introduce the notion of open deduction to reduce the size of derivations...⟧⟦**Lutz:** I was thinking about that, but (i) it is probably not worth the effort, as we wont have many derivation, and (ii) it is hard to define rectified derivations this way.⟧ □

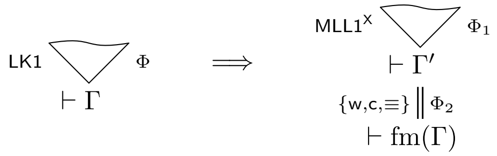
Lemma 27. *Let A and B be formulas. Then*

$$\begin{array}{rc} & A\\ \{\mathsf{w},\mathsf{c},\equiv\} & \Vert\\ & B\end{array}\quad\Longleftrightarrow\quad\begin{array}{rc} & A\\ \{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\} & \Vert\\ & B\end{array}$$

Proof. This follows immediately from Lemmas 25 and 26. □

C. *Rule Permutations*

Theorem 28. *Let Γ be a sequent. If $\vdash\Gamma$ is provable in $\mathsf{LK1}$ (as depicted on the left below) then there is a sequent Γ' such that there is a derivation as shown on the right below:*



Proof. Note that the instances of w, c in Φ_2 are deep, but inside sequent contexts.

First, if an instance of $\mathsf{wk}\ \dfrac{\vdash\Gamma}{\vdash\Gamma, A}$ is followed by a rule in which A is not in the principal formula, it can be permuted downwards. Otherwise, the proof can be transformed using the following rewriting rules.

$$\wedge\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdash\Gamma,A}\quad\vdash B,\Delta}{\vdash\Gamma,A\wedge B,\Delta}\ \rightsquigarrow\ \mathsf{wk}\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdots}}{\vdash\Gamma,A\wedge B,\Delta}$$

$$\vee\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma,A}{\vdash\Gamma,A,B}}{\vdash\Gamma,A\vee B}\ \rightsquigarrow\ \mathsf{w}\,\dfrac{\vdash\Gamma,A}{\vdash\Gamma,A\vee B}$$

$$\exists\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdash\Gamma,A[x/t]}}{\vdash\Gamma,\exists x.A}\ \rightsquigarrow\ \mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdash\Gamma,\exists x.A}$$

$$\forall\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdash\Gamma,A}}{\vdash\Gamma,\forall x.A}\ \rightsquigarrow\ \mathsf{wk}\,\dfrac{\vdash\Gamma}{\vdash\Gamma,\forall x.A}$$

$$\mathsf{ctr}\,\dfrac{\mathsf{wk}\,\dfrac{\vdash\Gamma,A}{\vdash\Gamma,A,A}}{\vdash\Gamma,A}\ \rightsquigarrow\ \vdash\Gamma,A$$

Note that in the case of $\vee$, we use the deep rule w which can be permuted down over all the rules. By using these rewriting rules, we can eventually get a derivation with all the instances of wk and w at the bottom. Now observe that the instances of ctr in Φ can be transformed using the following rule:

$$\mathsf{ctr}\,\dfrac{\vdash\Gamma,A,A}{\vdash\Gamma,A}\ \rightsquigarrow\ \mathsf{c}\,\dfrac{\vee\,\dfrac{\vdash\Gamma,A,A}{\vdash\Gamma,A\vee A}}{\vdash\Gamma,A}$$

Knowing that c can be permuted down over all the rules of $\mathsf{MLL1^X}$, we eventually obtain a derivation:

$$\begin{array}{c} \mathsf{MLL1^X}\ \bigtriangledown\ \Phi_1' \\ \vdash \Gamma_0 \\ {\scriptstyle \{\mathsf{wk},\mathsf{w},\mathsf{c},\equiv\}}\Big\Vert\,\Phi_2' \\ \vdash \Gamma \end{array}$$

Note that $\equiv$ is required here since the permutation of formulas is implicit in $\mathsf{MLL1^X}$.

By transforming each sequent of Φ_2' into its corresponding formula, and by considering the following rewriting rule:

$$\mathsf{wk}\,\frac{\vdash \Gamma}{\vdash \Gamma, A} \rightsquigarrow \mathsf{w}\,\frac{\vdash \mathrm{fm}(\Gamma)}{\vdash \mathrm{fm}(\Gamma) \vee A}$$

, we obtain a derivation

$$\begin{array}{c} \mathsf{MLL1^X}\ \bigtriangledown\ \Phi_1 \\ \vdash \Gamma' \\ {\scriptstyle \{\mathsf{w},\mathsf{c},\equiv\}}\Big\Vert\,\Phi_2 \\ \vdash \mathrm{fm}(\Gamma) \end{array}$$

where $\Gamma' = \mathrm{fm}(\Gamma_0)$ and Φ_1 can be obtained from Φ_1' by applying the $\vee$ rule. ⟦**TO CHECK:** ⟧ ⟦**Jui-Hsuan:** This might be a bit long...⟧ □

Lemma 29. *For every derivation* $\begin{array}{c} \mathsf{t} \\ {\scriptstyle \mathsf{MLS1^X}}\Big\Vert \\ A \end{array}$ *there are formulas A' and A'' such that*

$$\begin{array}{c} \mathsf{t} \\ {\scriptstyle \{\forall,\mathsf{ai},\mathsf{t}\}}\Big\Vert \\ A'' \\ {\scriptstyle \{\mathsf{s},\mathsf{mix},\equiv\}}\Big\Vert \\ A' \\ {\scriptstyle \{\exists\}}\Big\Vert \\ A \end{array}$$

Proof. First, observe that the $\exists$ rule can be permuted downwards over all the other rules since $A[x/t]$ has the same structure as A and none of the other rules has a premise of the form $S\{\exists x.A\}$. It suffices now to prove that for all $r_1 \in \{\forall, \mathsf{ai}, \mathsf{t}\}$, for all $r_2 \in \{\mathsf{s}, \mathsf{mix}, \equiv\}$, we can permute r_1 upwards over r_2. We show some cases here, and leave the others to the reader.

$$\mathsf{ai}\,\dfrac{\mathsf{s}\,\dfrac{S\{A \wedge (\mathsf{t} \vee C)\}}{S\{(A \wedge \mathsf{t}) \vee C\}}}{S\{(A \wedge (a \vee \bar{a})) \vee C\}} \rightsquigarrow \mathsf{s}\,\dfrac{\mathsf{ai}\,\dfrac{S\{A \wedge (\mathsf{t} \vee C)\}}{S\{A \wedge ((a \vee \bar{a}) \vee C)\}}}{S\{(A \wedge (a \vee \bar{a})) \vee C\}}$$

$$\mathsf{t}\,\dfrac{\mathsf{mix}\,\dfrac{S\{A \wedge B\}}{S\{A \vee B\}}}{S\{(A \vee (B \wedge \mathsf{t}))\}} \rightsquigarrow \mathsf{mix}\,\dfrac{\mathsf{t}\,\dfrac{S\{A \wedge B\}}{S\{A \wedge (B \wedge \mathsf{t})\}}}{S\{(A \vee (B \wedge \mathsf{t})\}}$$

⟦**TO CHECK:** ⟧ □

Lemma 30. *For every derivation* $\begin{array}{c} A \\ {\scriptstyle \{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}}\Big\Vert \\ B \end{array}$ *there are formulas A' and B' such that*

$$\begin{array}{c} A \\ {\scriptstyle \{\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}}\Big\Vert \\ A' \\ {\scriptstyle \{\mathsf{ac},\mathsf{c}_\forall\}}\Big\Vert \\ B' \\ {\scriptstyle \{\mathsf{w},\equiv\}}\Big\Vert \\ B \end{array}$$

Proof. First, a derivation consisting of an instance of w followed by $\mathsf{r} \in \{\mathsf{ac}, \mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists\}$ can be either replaced by a derivation consisting of w only or the instance of w can be permuted downwards. We show some cases here, and leave the others to the reader.

$$\mathsf{m}_\forall\,\dfrac{\mathsf{w}\,\dfrac{S\{\forall x.A\}}{S\{(\forall x.A) \vee (\forall x.B)\}}}{S\{\forall x.(A \vee B)\}} \rightsquigarrow \mathsf{w}\,\dfrac{S\{\forall x.A\}}{S\{\forall x.(A \vee B)\}}$$

$$\mathsf{m}\,\dfrac{\mathsf{w}\,\dfrac{S\{A \wedge C\}}{S\{(A \wedge C) \vee (B \wedge D)\}}}{S\{(A \vee B) \wedge (C \vee D)\}} \rightsquigarrow \mathsf{w}\,\dfrac{\mathsf{w}\,\dfrac{S\{A \wedge C\}}{S\{(A \vee B) \wedge C\}}}{S\{(A \vee B) \wedge (C \vee D)\}}$$

$$\mathsf{ac}\,\dfrac{\mathsf{w}\,\dfrac{S\{a\}}{S\{a \vee a\}}}{S\{a\}} \rightsquigarrow S\{a\}$$

For $\mathsf{r}_1 \in \{\mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists\}$, $\mathsf{r}_2 \in \{\mathsf{ac}, \mathsf{c}_\forall\}$, r_1 can be permuted upwards over r_2 (with some $\equiv$ inserted). The only non-trivial case is shown below:

$$\mathsf{m}_\forall\,\dfrac{\mathsf{c}_\forall\,\dfrac{S\{(\forall x.\forall x.A) \vee (\forall x.B)\}}{S\{(\forall x.A) \vee (\forall x.B)\}}}{S\{\forall x.(A \vee B)\}} \rightsquigarrow \mathsf{m}_\forall\,\dfrac{\equiv\,\dfrac{\mathsf{m}_\forall\,\dfrac{S\{(\forall x.\forall x.A) \vee (\forall x.B)\}}{S\{\forall x.(\forall x.A \vee B)\}}}{S\{(\forall x.A) \vee (\forall x.B)\}}}{S\{\forall x.(A \vee B)\}}$$

⟦**TODO:** permutation with $\equiv$⟧ □

We can now complete the proof of Theorems 17 and 18.

Proof of Theorem 18. Assume we have a proof of A in KS1. By Theorem 24 we have a proof of $\vdash A$ in LK1 to which we can apply Theorem 28. Finally, we apply Lemma 20 to get the desired shape. □

Proof of Theorem 17. Assume we have a proof of A in KS1. We first apply Theorem 18, and then Lemma 29 to the upper half and Lemma 30 to the lower half. □

VIII. Fonets and Linear Proofs

A. *From* $\mathsf{MLL1^X}$ *Proofs to Fonets*

Let Π be a $\mathsf{MLL1^X}$ proof of a rectified sequent $\vdash \Gamma$. We now show how Π is translated into a linked fograph $[\![\Pi]\!] = \langle [\![\Gamma]\!], \sim_\Pi \rangle$. We proceed inductively, making a case analysis on the last rule in Π. At the same time we are constructing a dualizer δ_Π, so that in the end we can conclude that $[\![\Pi]\!]$ is in fact a fonet.

1) Π is $\mathsf{ax}\,\dfrac{}{\vdash a, \bar{a}}$: Then the only link is $\{a, \bar{a}\}$, and δ_Π is empty.
2) Π is $\mathsf{t}\,\dfrac{}{\vdash \mathsf{t}}$: Then $\sim_\Pi$ and δ_Π are both empty.
3) The last rule in Π is $\mathsf{mix}\,\dfrac{\vdash \Gamma' \quad \vdash \Gamma''}{\vdash \Gamma', \Gamma''}$: By induction hypothesis, we have proofs Π' and Π'' of Γ' and Γ'', respectively. We have $[\![\Gamma]\!] = [\![\Gamma']\!] + [\![\Gamma'']\!]$ and let
$$\sim_\Pi = \sim_{\Pi'} \cup \sim_{\Pi''} \qquad \text{and} \qquad \delta_\Pi = \delta_{\Pi'} \cup \delta_{\Pi''}$$
4) The last rule in Π is $\vee\,\dfrac{\vdash \Gamma_1, A, B}{\vdash \Gamma_1, A \vee B}$: By induction hypothesis, there is proofs Π' of $\Gamma' = \Gamma_1, A, B$. We have $[\![\Gamma]\!] = [\![\Gamma']\!]$ and let $\sim_\Pi = \sim_{\Pi'}$ and $\delta_\Pi = \delta_{\Pi'}$.
5) The last rule in Π is $\wedge\,\dfrac{\vdash \Gamma_1, A \quad \vdash B, \Gamma_2}{\vdash \Gamma_1, A \wedge B, \Gamma_2}$: By induction hypothesis, we have proofs Π' and Π'' of $\Gamma' = \Gamma_1, A$ and $\Gamma'' = B, \Gamma_2$, respectively. We have $[\![\Gamma]\!] = [\![\Gamma_1]\!] + ([\![A]\!] \times [\![B]\!]) + [\![\Gamma_2]\!]$ and we let
$$\sim_\Pi = \sim_{\Pi'} \cup \sim_{\Pi''} \qquad \text{and} \qquad \delta_\Pi = \delta_{\Pi'} \cup \delta_{\Pi''}$$
6) The last rule in Π is $\exists\,\dfrac{\vdash \Gamma_1, A[x/t]}{\vdash \Gamma_1, \exists x.A}$: By induction hypothesis, there is a Π' of $\Gamma' = \Gamma_1, A[x/t]$. For each atom in $\Gamma' = \Gamma_1, A[x/t]$, there is a corresponding atom in $\Gamma = \Gamma_1, \exists x.A$. We can therefore define the linking $\sim_\Pi$ from the linking $\sim_{\Pi'}$ via this correspondence. Then, we let δ_Π be $\delta_{\Pi'} + [x/t]$. Since Γ is rectified x does not yet occur in $\delta_{\Pi'}$. Hence δ_Π is a dualizer of $[\![\Pi]\!]$.
7) The last rule in Π is $\forall\,\dfrac{\vdash \Gamma_1, A}{\vdash \Gamma_1, \forall x.A}$ (x not free in Γ_1) : By induction hypothesis, there is a proof Π' of $\Gamma' = \Gamma_1, A$, which has the same atoms as in $\Gamma = \Gamma_1, \forall x.A$. Hence, we can let $\sim_\Pi = \sim_{\Pi'}$ and $\delta_\Pi = \delta_{\Pi'}$.

Theorem 31. *If Π is a $\mathsf{MLL1^X}$ proof of a rectified sequent $\vdash \Gamma$, then $[\![\Pi]\!]$ is a fonet and δ_Π is a dualizer for it.*

Proof. We have to show that none of the operations above can introduce a bimatching. For cases 1–6, this is immediate. For case 7, observe that there is a potential dependency from each existential binder in $[\![\Gamma']\!]$ to the new x-binder $\bullet x$ in $[\![\Gamma]\!]$. However, observe that this $\bullet x$ vertex is not connected to any vertex in $[\![\Gamma']\!]$, and hence no such new dependency can be extended to a bimatching. That δ_Π is a dualizer for $[\![\Pi]\!]$ follows immediately from the construction. Hence, $[\![\Pi]\!]$ is a fonet. □

B. *From* $\mathsf{MLS1^X}$ *Proofs to Fonets*

There is a more direct path from a $\mathsf{MLL1^X}$ proof Π of a rectified sequent Γ to the linked fograph $[\![\Pi]\!]$: simply take the fograph $[\![\Gamma]\!]$, and let the equivalence classes of $\sim_\Pi$ be all the atom pairs that meet in an instance of ax, and δ_Π is simply the collection of all substitutions of all the instances of the $\exists$-rule in Π. We have chosen the more cumbersome path above because it gives us a direct proof of Theorem 31. However, for translating $\mathsf{MLS1^X}$ derivation into fonets, we employ exactly that direct path.

A derivation Φ in $\mathsf{MLS1^X}$ is ***rectified*** if every line in Φ is rectified.

Lemma 32. *Let Φ be a $\mathsf{MLS1^X}$ proof of a formula A. Then Φ is rectified iff A is rectified.*

Proof. The only rules involving bound variables are $\forall$ and $\exists$ which both remove a binder (and all occurrences of the variable it binds). □

Hence, for a non-rectified $\mathsf{MLS1^X}$ derivation Φ in $\mathsf{MLS1^X}$ we can define its ***rectification*** $\widehat{\Phi}$ inductively, by rectifying each line, proceeding step-wise from conclusion to premise.[2]

A rectified derivation ${}^{\mathsf{MLS1^X}}\underset{A}{\overset{\mathsf{t}}{\Vert}}\,\Phi$ determines a substitution which maps the existential bound variables occurring in A to the terms substituted for them in the instances of the $\exists$-rule in Φ. We denote this substitution with δ_Φ and call it the ***dualizer*** of Φ. Furthermore, every atom occurring in the conclusion A must be consumed by a unique instance of the rule ai in Φ. This allows us to define a (partial) equivalence relation $\sim_\Phi$ on the atom occurrences in A by $a \sim_\Phi b$ if a and b are consumed by the same instance of ai in Φ. We call $\sim_\Phi$ the ***linking*** of Φ, and define $[\![\Phi]\!] = \langle [\![A]\!], \sim_\Phi \rangle$.

[[TODO: example here**]]**

Theorem 33. *Let ${}^{\mathsf{MLS1^X}}\underset{A}{\overset{\mathsf{t}}{\Vert}}\,\Phi$ be a rectified derivation. Then $[\![\Phi]\!]$ is a fonet and δ_Φ a dualizer for it.*

For proving this theorem, we have to show that no inference rule in $\mathsf{MLS1^X}$ can introduce a bimatching. To simplify the argument, we introduce the ***frame*** [34] of the fograph $\mathcal{C}$, which is a linked (proposional) cograph in which the dependencies between the binders in $\mathcal{C}$ are encoded as links.

More formally, let C be a formula with $[\![C]\!] = \mathcal{C}$, to which we exhaustively apply the following subformula rewriting steps, to obtain a sequent $C^\star$:

1) **Encode dependencies as fresh links.** For each dependency $\{\bullet x_i, \bullet y_j\}$ in $\mathcal{C}$, with corresponding subformulas $\exists x_i.A$ and $\forall y_j.B$ in C, we pick a fresh (nullary) predicate symbol $q_{i,j}$, and then replace $\exists x_i.A$ by $\bar{q}_{i,j} \wedge \exists x_i.A$, and replace $\forall y_j.B$ by $q_{i,j} \vee \forall y_j.B$.

[2] As for formulas, the rectification of a derivation is unique up to renaming of bound variables.

2) **Erase quantifiers.** After step 1, remove all the quantifiers, i.e., replace $\exists x_i.A$ by A and replace $\forall y_j.B$ by B everywhere.

3) **Simplify atoms.** After step 2, replace every predicate $p(t_1 \cdots t_n)$ (resp. $\overline{p}(t_1 \cdots t_n)$) with a nullary predicate symbol p (resp. $\overline{p}$)

The $\sim_{C^\star}$ consists of the pairs induced by $\sim_{\mathcal{C}}$ and the new pairs $\{q_{i,j}, \overline{q}_{i,j}\}$ introduced in step 1 above. We call $C^\star$ the ***frame*** of C and we define the ***frame*** of $\mathcal{C}$, denoted $\mathcal{C}^\star$, as $\langle [\![C^\star]\!], \sim_{C^\star}\rangle$.

Lemma 34. *A linked fograph $\mathcal{C}$ has an induced bimatching iff its frame $\mathcal{C}^\star$ has an induced bimatching.*

Proof. This immediately follows from the construction of the frame.⟦[**Lutz:** is it really an "iff"? It is easy to construct from a bimatching in $\mathcal{C}$ a bimatching in the frame. (and I think we only need that direction). But what about the other direction?]⟧ □

Proof of Theorem 33. From Φ we construct a derivation $\Phi^\star$ of $A^\star$ in the propositional fragment of $\mathsf{MLS1^X}$, such that $[\![\Phi^\star]\!] = [\![\Phi]\!]^\star$. The rules $\mathsf{ai}, \mathsf{t}, \mathsf{mix}$ and s are translated trivially, and for $\equiv$, it suffices to observe that the frame construction is invariant under $\equiv$. Finally, for the rules $\forall$ and $\exists$, proceed as follows. Every instance of $\forall$ is replaced by the derivation on the right below:[3]

$$\forall\,\frac{S\{\mathsf{t}\}}{S\{\forall y_j.\mathsf{t}\}} \quad\rightsquigarrow\quad \begin{array}{c} \mathsf{t} \\ \{\mathsf{ai},\mathsf{t}\}\,\Big\Vert\,\Psi_1 \\ S\{(q_{h_1,j} \vee \overline{q}_{h_1,j}) \wedge \cdots \wedge (q_{h_j,j} \vee \overline{q}_{h_j,j}) \wedge \mathsf{t}\} \\ \{\mathsf{s},\equiv\}\,\Big\Vert\,\Psi_2 \\ S\{q_{h_1,j} \vee \cdots \vee q_{h_j,j} \vee (\overline{q}_{h_1,j} \wedge \cdots \wedge \overline{q}_{h_j,j} \wedge \mathsf{t})\} \end{array}$$

where $h_1, \ldots, h_j$ range over the indices of the existential binders dependend on that y_j. It is easy to see how Ψ_1 is constructed, and for Ψ_2 see, e.g. [**?**], [33], [35]⟦[**Lutz:** check if it is really there. otherwise [36]]⟧ Then, every occurrence of $\forall y_j.F$ is replaced by $q_{h_1,j} \vee \cdots \vee q_{h_j,j} \vee (\overline{q}_{h_1,j} \wedge \cdots \wedge \overline{q}_{h_j,j} \wedge F)$ in the derivation below that $\forall$-instance. Now, observe that all instances of the $\exists$-rule introducing x_i dependend on y_j must occur below in the derivation (otherwise Φ would not be recified). Now consider such an instance $\exists\,\dfrac{S\{B[x_i/t]\}}{S\{\exists x_i.B\}}$. Its context $S\{\cdot\}$ must contain all the $\forall y_j$ the $\exists x_i$ depends on, such that B is in their scope. Following the translation of the $\forall$ rules above, we can therefore translate the $\exists$-rule instance by the following derivation

$$\begin{array}{c} S_0\{\overline{q}_{i,k_1} \wedge S_1\{\overline{q}_{i,k_2} \wedge \cdots S_{k_i-1}\{\overline{q}_{i,k_i} \wedge S_{k_i}\{B'\}\} \cdots\}\} \\ \{\mathsf{s},\equiv\}\,\Big\Vert\,\Psi_3 \\ S_0\{S_1\{\cdots S_{k_i-1}\{S_{k_i}\{\overline{q}_{i,k_1} \wedge \overline{q}_{i,k_2} \wedge \cdots q_{i,k_i} \wedge B'\}\} \cdots\}\} \end{array}$$

where $k_1, \ldots, k_i$ are the indices of of the universal binders on which that x_i depends, and B' is B in which all predicates are replaced by nullary one (step 3 in the frame construction). The derivation Ψ_3 can be constructed in the same way as Ψ_2 above.

Doing this to all instances of the rules $\forall$ and $\exists$ in Φ yields indeed a propositional derivation $\Phi^\star$ with $[\![\Phi^\star]\!] = [\![\Phi]\!]^\star$. It has been shown by Retoré [**?**] and rediscovered by Straßburger [36] that $[\![\Phi^\star]\!] = \langle [\![C^\star]\!], \sim_{\Phi^\star}\rangle$ can not contain an induced bimatching. By Lemma 36, $[\![\Phi]\!]$ does not have an induced bimatching either. Furthemore, it followed from the definition of δ_Φ that it is a dualizer for $[\![\Phi]\!]$. Hence $[\![\Phi]\!]$ is a fonet. □

Remark 35. There is an alternative path of proving Theorem 33 by translating Φ to an $\mathsf{MLL1^X}$-proof Π, observing that this process preserves the linking and the dualizer. However, for this, we have to extent the construction above to the cut-rule, and then show that linking and dualizer of a sequent proof Π are invariant under cut elimination. This can be done similarly to unification nets in [34].

C. From Fonets to $\mathsf{MLL1^X}$ Proofs

Now we are going to show how from a given fonet $\langle \mathcal{C}, \sim_{\mathcal{C}}\rangle$ we can construct a sequent proof Π in $\mathsf{MLL1^X}$ such that $[\![\Pi]\!] = \langle \mathcal{C}, \sim_{\mathcal{C}}\rangle$. In the proof net literature, this operation is also called *sequentialization*. The basic idea behind our sequentialization is to construct a propositional linked cograph, called the ***frame*** [34] of $\mathcal{C}$, in which the dependencies between the binders in $\mathcal{C}$ are encoded as links. Then we can apply the *splitting tensor theorem* to the frame, and then reconstruct the sequent proof Π. ⟦[**Lutz:** if the proof of thm 33 is verified, we can delete the frame-def here]⟧

More formally, let Γ be a sequent with $[\![\Gamma]\!] = \mathcal{C}$, to which we exhaustively apply the following subformula rewriting steps, to obtain a sequent $\Gamma^\star$:

1) **Encode dependencies as fresh links.** For each dependency $(\bullet x, \bullet y)$ in $\mathcal{C}$, with corresponding subformulas $\exists x A$ and $\forall y B$ in Γ, we pick a fresh (nullary) predicate symbol q, and then replace $\exists x.A$ by $q \wedge \exists x.A$, and replace $\forall y.B$ by $\overline{q} \vee \forall y.B$.

2) **Erase quantifiers.** After step 1, remove all the quantifiers, i.e., replace $\exists x.A$ by A and replace $\forall y.B$ by B everywhere.

3) **Simplify atoms.** After step 2, replace every predicate $p(t_1 \cdots t_n)$ (resp. $\overline{p}(t_1 \cdots t_n)$) with a nullary predicate symbol p (resp. $\overline{p}$)

The $\sim_{\Gamma^\star}$ consists of the pairs induced by $\sim_{\mathcal{C}}$ and the new pairs $\{q, \overline{q}\}$ introduced in step 1 above. We call $\Gamma^\star$ the ***frame*** of Γ and we define the ***frame*** of $\mathcal{C}$, denoted $\mathcal{C}^\star$, as $\langle [\![\Gamma^\star]\!], \sim_{\Gamma^\star}\rangle$, and we immediately have the following:

Lemma 36. *A linked fograph $\mathcal{C}$ induces a bimatching iff its frame $\mathcal{C}^\star$ has an induced bimatching.*

Let Γ be a propositional sequent and $\sim_\Gamma$ be a linking for $[\![\Gamma]\!]$. A conjunction formula $A \wedge B$ is ***splitting*** or a ***splitting tensor*** if $\Gamma = \Gamma', A \wedge B, \Gamma''$ and $\sim_\Gamma = \sim_1 \cup \sim_2$, such that $\sim_1$ is a linking for $[\![\Gamma', A]\!]$ and $\sim_2$ is a linking for $[\![B, \Gamma'']\!]$, i.e., removing the $\wedge$ from $A \wedge B$ splits the linked fograph

[3] For better readability we omit superfluous parentheses, knowing that we always have $\equiv$ incorporating associativity and commutativity of $\wedge$ and $\vee$.

$\langle \llbracket \Gamma \rrbracket, \sim_\Gamma \rangle$ into two fographs. We say that $\langle \llbracket \Gamma \rrbracket, \sim_\Gamma \rangle$ is ***mixed*** iff $\Gamma = \Gamma', \Gamma''$ and $\sim_\Gamma = \sim_1 \cup \sim_2$, such that $\sim_1$ is a linking for $\llbracket \Gamma' \rrbracket$ and $\sim_2$ is a linking for $\llbracket \Gamma'' \rrbracket$. Finally, $\langle \llbracket \Gamma \rrbracket, \sim_\Gamma \rangle$ is ***splittable*** if it is mixed or has a splitting tensor.

The purpose of introducing the frame is the following theorem.

Theorem 37. *Let Γ be a propositional sequent containing only atoms and $\wedge$-formulas, and $\sim_\Gamma$ be a linking for $\llbracket \Gamma \rrbracket$. If $\langle \llbracket \Gamma \rrbracket, \sim_\Gamma \rangle$ does not induce a bimatching then it is splittable.*

This is the well-know splitting-tensor-theorem [23], [37], adapted for the presence of mix. In the setting of linked cographs, it has first been proved by Retoré [38], [39]. We use it now for our sequentialization:

Theorem 38. *Let $\langle \mathcal{C}, \sim_\mathcal{C} \rangle$ be a fonet, and let Γ be a sequent with $\llbracket \Gamma \rrbracket = \mathcal{C}$. Then there is an $\mathsf{MLL1}^\mathsf{X}$-proof Π of Γ, such that $\llbracket \Pi \rrbracket = \langle \mathcal{C}, \sim_\mathcal{C} \rangle$.*

Proof. Let $\delta_\mathcal{C}$ be the dualizer of $\langle \mathcal{C}, \sim_\mathcal{C} \rangle$. We proceed by induction on the size of Γ (i.e., the number of symbols in it, without counting the commas). If Γ contains a formula with $\vee$-root, or a formula $\forall x.A$, we can immediately apply the $\vee$-rule or the $\forall$-rule of $\mathsf{MLL1}^\mathsf{X}$ and proceed by induction hypothesis. If Γ contains a formula $\exists x.A$ such that the corresponding binder $\bullet x$ in $\mathcal{C}$ has no dependency, then we can apply the $\exists$-rule, choosing the term t as determined by $\delta_\mathcal{C}$, and proceed by induction hypothesis. Hence, we can now assume that Γ contains only atoms, $\wedge$-formulas, or formulas of shape $\exists x.A$, where the vertex $\bullet x$ has dependencies. Then the frame $\langle \llbracket \Gamma^\star \rrbracket, \sim_{\Gamma^\star} \rangle$ does not induce a bimatching and contains only atoms and $\wedge$-formulas, and is therefore splittable. If it is mixed, then we can apply the mix-rule to Γ and apply the induction hypothesis to the two components. If it is not mixed then there must be a splitting tensor. If the splitting $\wedge$ is already in Γ, then we can apply the $\wedge$-rule and proceed by induction hypothesis on the two branches. However, if $\Gamma^\star$ is not mixed and all splitting tensors are $\wedge$-formulas introduced in step 1 of the frame construction, then we get a contradiction as in that case there must be a $\vee$- or $\forall$-formula in Γ. [[[**Lutz:** can anyone give a good argument here?]]] □

D. From Fonets to $\mathsf{MLS1}^\mathsf{X}$ Proofs

We can now straightforwardly obtain the same result for $\mathsf{MLS1}^\mathsf{X}$:

Theorem 39. *Let $\langle \mathcal{C}, \sim_\mathcal{C} \rangle$ be a fonet, and let C be a formula with $\llbracket C \rrbracket = \mathcal{C}$. Then there is a derivation $\mathsf{MLS1}^\mathsf{X} \begin{array}{c}\mathsf{t}\\ \Vert\\ C\end{array} \Phi$ such that $\llbracket \Phi \rrbracket = \langle \mathcal{C}, \sim_\mathcal{C} \rangle$.*

Proof. We apply Theorem 38 to obtain a sequent proof Π of $\vdash C$ with $\llbracket \Pi \rrbracket = \langle \mathcal{C}, \sim_\mathcal{C} \rangle$. Then we apply Lemma 20, observing that the translation from $\mathsf{MLL1}^\mathsf{X}$ to $\mathsf{MLS1}^\mathsf{X}$ preserves linking and dualizer. □

Remark 40. Note that it is also possible to do a direct "sequentialization" into the deep inference system $\mathsf{MLS1}^\mathsf{X}$, using the techniques presented in [36] and [40].

IX. Skew Bifibrations and Resource Management

In this section we establish the relation between skew bifibrations and derivations in $\{\mathsf{w}, \mathsf{ac}, \mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$. However, if a derivation Φ contains instances of the rules $\mathsf{c}_\forall$, $\mathsf{m}_\forall$, and $\mathsf{m}_\exists$ we can no longer naively define the rectification $\widehat{\Phi}$ as in the previous section for $\mathsf{MLS1}^\mathsf{X}$, as these two rules cannot be applied if premise and conclusion are rectified. For this reason we define here rectified versions $\widehat{\mathsf{c}_\forall}$, $\widehat{\mathsf{m}_\forall}$ and $\widehat{\mathsf{m}_\exists}$, shown below:

$$\widehat{\mathsf{c}_\forall}\ \frac{S\{\forall y.\forall x.Ax\}}{S\{\forall x.Ax\}} \qquad \widehat{\mathsf{m}_\forall}\ \frac{S\{(\forall y.Ay) \vee (\forall z.Bz)\}}{S\{\forall x.(Ax \vee Bx)\}} \qquad \widehat{\mathsf{m}_\exists}\ \frac{S\{(\exists y.Ay) \vee (\exists z.Bz)\}}{S\{\exists x.(Ax \vee Bx)\}}$$

Here, we use the notation $A\cdot$ for a formula A with occurrences of a placeholder $\cdot$ for a variable. Then Ax stands for the results of replacing that placeholder with x, and also indicating that x must not occur in $A\cdot$. Then $\forall x.Ax$ and $\forall y.Ay$ are the same formula modulo renaming of the bound variable bound by the outermost $\forall$-quantifier. We also demand that the variables x, y, and z do not occur in the context $S\{\cdot\}$.

Note that in an instance of $\widehat{\mathsf{m}_\forall}$ or $\widehat{\mathsf{m}_\exists}$ (as shown above), we can have $x = y$ or $x = z$, but not both if the premise is rectified. If $x = y$ and $x = z$ we have $\mathsf{m}_\forall$ and $\mathsf{m}_\exists$ as special cases of $\widehat{\mathsf{m}_\forall}$ and $\widehat{\mathsf{m}_\exists}$, respectively. And similarly, if $x = y$ then $\mathsf{c}_\forall$ is a special case of $\widehat{\mathsf{c}_\forall}$.

For a derivation Φ in $\{\mathsf{w}, \mathsf{ac}, \mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$, we can now construct the ***rectification*** $\widehat{\Phi}$ by rectifying each line of Φ, yielding a derivation in $\{\mathsf{w}, \mathsf{ac}, \mathsf{m}, \widehat{\mathsf{m}_\forall}, \widehat{\mathsf{m}_\exists}, \equiv\}$.

For each instance $\mathsf{r}\,\dfrac{Q}{P}$ of an inference rule in $\{\mathsf{w}, \mathsf{ac}, \widehat{\mathsf{c}_\forall}, \mathsf{m}, \widehat{\mathsf{m}_\forall}, \widehat{\mathsf{m}_\exists}, \equiv\}$ we can define the ***induced map*** $\lfloor \mathsf{r} \rfloor \colon V_{\llbracket Q \rrbracket} \to V_{\llbracket P \rrbracket}$ which acts as the identity for $\mathsf{r} \in \{\mathsf{m}, \equiv\}$ and as the canonical injection for $\mathsf{r} = \mathsf{w}$. For $\mathsf{r} = \mathsf{ac}$ it maps the vertices corresponding to the two atoms in the premise to vertex of the contracted atom in the conclusion, and for $\mathsf{r} \in \{\widehat{\mathsf{c}_\forall}, \widehat{\mathsf{m}_\forall}, \widehat{\mathsf{m}_\exists}\}$ it maps the two vertices corresponding to the quantifiers in the premise to the one in the conclusion (as acts as the identity on all other vertices). For a derivation Φ in $\{\mathsf{w}, \mathsf{ac}, \widehat{\mathsf{c}_\forall}, \mathsf{m}, \widehat{\mathsf{m}_\forall}, \widehat{\mathsf{m}_\exists}, \equiv\}$ we can then define the ***induced map*** $\lfloor \Phi \rfloor$ as the composition of the induced maps of the rule instances in Φ. [[[**Jui-Hsuan:** maybe mention at least that the induced maps define graph homomorphisms. Do we need to talk about the contexts $S\{\cdot\}$ here (induced maps act clearly as the identity on contexts but we need them for the composition)?]]] [[[**Lutz:** For the context, I aleady say it is the identity. For the homom, it comes later]]]

Lemma 41. *Let* $\{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\} \begin{array}{c}A\\ \Vert\\ B\end{array} \Phi$ *be a derivation. Then there is a recified derivation* $\{\mathsf{w},\widehat{\mathsf{ac}},\widehat{\mathsf{c}_\forall},\mathsf{m},\widehat{\mathsf{m}_\forall},\widehat{\mathsf{m}_\exists},\equiv\} \begin{array}{c}\widehat{A}\\ \Vert\\ \widehat{B}\end{array} \widehat{\Phi}$, *such that* *the induced maps $\lfloor\Phi\rfloor\colon [\![A]\!] \to [\![B]\!]$ and $\lfloor\widehat{\Phi}\rfloor\colon [\![\widehat{A}]\!] \to [\![\widehat{B}]\!]$ are equal up to a variable renaming of the vertex labels.*

Proof. Immediate from the definition. □

[[**TODO:** example]]

A. From Contraction and Weakening to Skew Bifibrations

We say that a derivation Φ is ***sane*** if for every line Q in Φ we have that $[\![D]\!]$ is a fograph (i.e., all binders are legal). Clearly, every rectified derivation is sane, but not vice versa, as we might have multiple occurrences of bound variables in Q, such that $[\![Q]\!]$ is still a fograph.

Lemma 42. *Let $\begin{array}{c}A\\ \{\mathsf{w},\mathsf{ac},\widehat{\mathsf{c}_\forall},\mathsf{m},\widehat{\mathsf{m}_\forall},\widehat{\mathsf{m}_\exists},\equiv\}\big\Vert\,\Phi\\ B\end{array}$ be a sane derivation. Then the induced map $\lfloor\Phi\rfloor\colon [\![A]\!] \to [\![B]\!]$ is a skew bifibration.*

Before we show the proof of this lemma, we introduce another useful concept: the ***propositional encoding*** A° of a formula A, which is a propositional formula with the property that $[\![A^\circ]\!] = [\![A]\!]$. For this, we introduce new propositional variables that have the same names as the (first-order) variables $x \in \textsc{var}$. Then A° is defined inductively by:

$$\begin{array}{rclcrcl} a^\circ &=& a & \qquad & (\forall x A)^\circ &=& x \vee A^\circ \\ (A \vee B)^\circ &=& A^\circ \vee B^\circ & & (\exists x A)^\circ &=& x \wedge A^\circ \\ (A \wedge B)^\circ &=& A^\circ \wedge B^\circ & & & & \end{array}$$

Lemma 43. *For every formula A, we have $[\![A^\circ]\!] = [\![A]\!]$.*

Proof. Straightforward induction on A. □

We use $\equiv^\circ$ to denote the restriction of $\equiv$ to propositional formulas, i.e., the first two lines in (2).

Proof of Lemma 42. First, observe that for every inference rule $\mathsf{r} \in \{\mathsf{w},\mathsf{ac},\widehat{\mathsf{c}_\forall},\mathsf{m},\widehat{\mathsf{m}_\forall},\widehat{\mathsf{m}_\exists},\equiv\}$ the induced map $\lfloor\mathsf{r}\rfloor\colon V_{[\![Q]\!]} \to V_{[\![P]\!]}$ defines a existential preserving graph homomorphism $[\![Q]\!] \to [\![P]\!]$ and a fibration on the corresponding binding graphs. [[**Jui-Hsuan:** we may need to have some explication here.]][[**Lutz:** no]] Therefore, their composition $\lfloor\Phi\rfloor$ has the same properties fibration.

For showing that it is also a skew fibration, we construct for Φ its propositional encoding Φ° by translating every line into its propositional encoding. [[**Jui-Hsuan:** maybe mention that an instance of one of the other rules can be translated into an instance of the same rule. It's trivial but may be worth mentioning.]][[**Lutz:** done below]] The instances of the rules $\widehat{\mathsf{m}_\forall}$ and $\widehat{\mathsf{m}_\exists}$ are replaced in two steps by:

$$\equiv\frac{\widehat{\mathsf{ac}}\dfrac{S\{(y \vee (Ay)^\circ) \vee (z \vee (Bz)^\circ)\}}{S\{(y \vee z) \vee ((Ay)^\circ \vee (Bz)^\circ)\}}}{S\{x \vee ((Ax)^\circ \vee (Bx)^\circ)\}}$$

and

$$\mathsf{m}\frac{\widehat{\mathsf{ac}}\dfrac{S\{(y \wedge (Ay)^\circ) \vee (z \wedge (Bz)^\circ)\}}{S\{(y \vee z) \wedge ((Ay)^\circ \vee (Bz)^\circ)\}}}{S\{x \wedge ((Ax)^\circ \vee (Bx)^\circ)\}}$$

respectively, where $\widehat{\mathsf{ac}}$ is a ac that renames the variables—the propositional variable, as well as the first-order variable of the same name—as everything is recified, there is no ambiguity here. Any instance of a rule w, ac, m, or $\equiv$ is translated to an instance of the same rule, and $\widehat{\mathsf{c}_\forall}$ is translated to $\widehat{\mathsf{ac}}$.

This gives us a derivation $\begin{array}{c}A^\circ\\ \{\mathsf{w},\mathsf{ac},\widehat{\mathsf{ac}},\mathsf{m},\equiv^\circ\}\big\Vert\,\Phi^\circ\\ B^\circ\end{array}$ such that $\lfloor\Phi^\circ\rfloor = \lfloor\Phi\rfloor$. It has been shown in [21] that $\lfloor\Phi^\circ\rfloor$ is a skew fibration (see also [10], [41], [13]). Hence, $\lfloor\Phi\rfloor$ is a skew fibration. □

B. From Skew Bifibrations to Contraction and Weakening

Lemma 44. *Let $\mathcal{A}$ and $\mathcal{B}$ be fographs, let $\varphi\colon \mathcal{A} \to \mathcal{B}$ be a skew bifibration, and let A and B be formulas with $[\![A]\!] = \mathcal{A}$ and $[\![B]\!] = \mathcal{B}$. Then there are derivations*

$$\begin{array}{c}A\\ \{\mathsf{w},\mathsf{ac},\widehat{\mathsf{c}_\forall},\mathsf{m},\widehat{\mathsf{m}_\forall},\widehat{\mathsf{m}_\exists},\equiv\}\big\Vert\,\widehat{\Phi}\\ B\end{array} \quad\text{and}\quad \begin{array}{c}A\sigma_\varphi\\ \{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\big\Vert\,\Phi\\ B\end{array}$$

such that $\lfloor\widehat{\Phi}\rfloor = \varphi$ and $\widehat{\Phi}$ is a rectification of Φ, and σ_φ is the substitution induced by φ.

In the proof of this lemma, we make use of the following concept: Let $\begin{array}{c}P\\ \mathsf{S}\big\Vert\,\Psi\\ Q\end{array}$ be a derivation where P and Q are propositional formulas (possibly using variable $x \in \textsc{var}$ at the places of atoms). We say that Ψ can be ***lifted*** to S' if there are (first-order) formulas C and D such that $P = C^\circ$ and $Q = D^\circ$ and there is a derivation $\begin{array}{c}C\\ \mathsf{S}'\big\Vert\,\Psi'\\ D\end{array}$.

Proof of Lemma 44. By Lemma 43 we have $\mathcal{A} = [\![A^\circ]\!]$ and $\mathcal{B} = [\![B^\circ]\!]$. Let $V'_{\mathcal{B}} \subseteq V_{\mathcal{B}}$ be the image of φ, and let $\mathcal{B}_1$ be the subgraph of $\mathcal{B}$ induced by $V'_{\mathcal{B}}$. Hence, we have two maps $\varphi''\colon \mathcal{A} \to \mathcal{B}_1$ being a surjection and $\varphi'\colon \mathcal{B}_1 \to \mathcal{B}$ being an injection that reflects edges. [[**Jui-Hsuan:** what do you mean by "reflect edges"?]][[**Lutz:** edge downstairs implies edge upstairs]] Both, φ' and φ'' remain skew bifibrations. Let us first look at φ'. Let $\tilde{B}_1$ be the propositional formula obtained from B° by removing all atoms that are not represented by vertices in $V'_{\mathcal{B}}$. Then $[\![\tilde{B}_1]\!] = \mathcal{B}_1$. By [21, Proposition 7.6.1], we have a derivation $\begin{array}{c}\tilde{B}_1\\ \{\mathsf{w},\equiv\}\big\Vert\,\Phi_1^\circ\\ B^\circ\end{array}$. A subformula of B° is called *weak* if it has been introduced by a weakening. All subformulas of a weak formula are also weak. Two weak subformulas B' and B'' of B° form a *weak pair* if $B^\circ \equiv S\{B' \vee B''\}$ for some context $S\{\cdot\}$. We can assume without loss of generality that whenever weak subformulas B' and B'' form a weak pair, they have been introduced by the same instance of w in Φ_1°.[4] Now we show that Φ_1° can be lifted. For this, observe that whenever

[4] If Φ_1° is not of that shape, it can brought into this form by simple rule permutations, and then collapsing several weakenings into a single one.

a weakening in Φ_1° deletes an atom $x \in \mathsf{VAR}$, it must also delete all atoms in the scope of the corresponding quantifier, because φ' is a fibration on the binding graph. Hence, each line in Φ_1° is the propositional encoding P° of a first-order formula P. We now have to show that each instance of w is indeed a correct application in first-order logic. For this we have to inspect the cases a weakening happens inside a subformula $x \vee C$ or $x \wedge C$ in Φ_1°. There are the following cases:

$$\frac{S\{x \vee C\}}{S\{x \vee D \vee C\}} \qquad \frac{S\{x \wedge C\}}{S\{x \wedge (D \vee C)\}} \qquad \frac{S\{x \wedge C\}}{S\{(x \vee D) \wedge C\}}$$

In the first case the weakening happens inside the scope of a $\forall$-quantifier, and in the second case inside the scope of a $\exists$-quantifier. Both cases are unproblematic in first-order logic. However, in the third case an $\exists$-quantifier would be transformed into an $\forall$-quantifier. But as φ has to preserve existentials, this third case cannot occur. Thus we have a first order derivation $\begin{array}{c} B_1 \\ {\scriptstyle \{\mathsf{w},\equiv\}}\Big\Vert{\scriptstyle \Phi_1} \\ B \end{array}$ with $B_1^\circ = \tilde{B}_1$.

Let us now look at φ''. Let $\mathcal{A}_1 = \mathcal{A}\sigma_\varphi$ be the graph obtained from $\mathcal{A}$ by applying σ_φ to all the labels. Note that $\mathcal{A}_1$ is not necessarily a fograph, as binders might be illegal. But it still is a cograph, and we have a surjective skew fibration $\varphi'' \colon \mathcal{A}_1 \to \mathcal{B}_1$ that preserves the labels. Therefore, by [41, Proposition 7.5], there is a derivation $\begin{array}{c} A_1^\circ \\ {\scriptstyle \{\mathsf{ac},\mathsf{m},\equiv^\circ\}}\Big\Vert{\scriptstyle \Phi_2^\circ} \\ B_1^\circ \end{array}$, where $A_1^\circ = A^\circ\sigma_\varphi$ is the result of applying σ_φ to A°. Note that $A_1^\circ = (A\sigma_\varphi)^\circ$ and B_1° are both proposional encodings. We plan to show that Φ_2 can be lifted to $\{\mathsf{ac}, \mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$. However, observe that not every formula occurring in Φ_2 is a propositional encoding. There are two reasons for this: (i) we might have $P \equiv^\circ Q$ where P is a propositional encoding but Q is not, and (ii) the rule ac can duplicate an atom $x \in \mathsf{VAR}$. Let us write ac_x for such instances. To address (i), we consider here formulas equivalent modulo $\equiv$, always knowing that we can add instances of $\equiv$ as needed.[5] ⟦**Jui-Hsuan:** this does not seem clear to me. What if from A_1° to B_1° there are just a bunch of $\equiv^\circ$? What do we do in this case?⟧⟦**Lutz:** see footnote⟧ To address (ii), we apply a permutation argument, permuting all instances of ac_x up until they either reach the top of the derivation or an instance of m which separates the two atoms in the premise. More precisely, we consider the following inference rule

$$\mathsf{ac}_x^{\equiv}\,\frac{S_0\{S_1\{x\} \vee S_2\{x\}\}}{S\{x\}} \tag{4}$$

where $S_1\{\cdot\} \equiv \{\cdot\} \vee E$ and $S_2\{\cdot\} \equiv \{\cdot\} \vee F$ and $S\{\cdot\} \equiv S_0\{\{\cdot\} \vee E \vee F\}$ for some formulas E and F, where E or F or both might be empty. The rule $\mathsf{ac}_x^{\equiv}$ permutes over $\equiv$, ac, and other instances of $\mathsf{ac}_x^{\equiv}$, and over instances of m if they occur inside S_0 or S_1 or S_2. The only situation in which $\mathsf{ac}_x^{\equiv}$ cannot be permuted up is the following:

$$\begin{array}{r} \mathsf{m}\,\dfrac{S\{(R_1\{x\} \wedge C) \vee (R_2\{x\} \wedge D)\}}{S\{(R_1\{x\} \vee R_2\{x\}) \wedge (C \vee D)\}} \\ \mathsf{ac}_x^{\equiv}\,\dfrac{\phantom{S\{(R_1\{x\} \vee R_2\{x\}) \wedge (C \vee D)\}}}{S\{R\{x\} \wedge (C \vee D)\}} \end{array} \tag{5}$$

We can therefore assume that all instances of ac_x, that contract an atom $x \in \mathsf{VAR}$ are either at the top of Φ_2° or below a m-instance as in (5). We now lift Φ_2° to $\{\mathsf{ac}, \mathsf{c}_\forall, \mathsf{m}, \mathsf{m}_\forall, \mathsf{m}_\exists, \equiv\}$, proceed by induction on the height of Φ_2°, beginning at the top, making a case analysis on the topmost rule that is not a $\equiv$.

- ac_x: We know that the premisse of (4) is a propositional encoding. Hence, $S_1\{\cdot\} = \{\cdot\} \vee E^\circ$ and $S_2\{\cdot\} = \{\cdot\} \vee F^\circ$ and both x are universals, and $E^\circ \vee F^\circ$ contains all occurrences of x bound by that universal. We have the following subcases:
  - E and F are both non-empty: We have
    $$\mathsf{ac}_x^{\equiv}\,\frac{S^\circ\{(x \vee E^\circ) \vee (x \vee F^\circ)\}}{S^\circ\{x \vee (E^\circ \vee F^\circ)\}}$$
    which can be lifted to
    $$\mathsf{m}_\forall\,\frac{S\{(\forall x.E) \vee (\forall x.F)\}}{S\{\forall x.(E \vee F)\}}$$
    where $S^\circ\{\cdot\}, E^\circ, F^\circ$ are the propositional encodings of $S\{\cdot\}, E, F$, respectively.
  - E° is empty and F° is non-empty: We have
    $$\mathsf{ac}_x^{\equiv}\,\frac{S^\circ\{x \vee (x \vee F^\circ)\}}{S^\circ\{x \vee F^\circ)\}}$$
    which can be lifted to
    $$\mathsf{c}_\forall\,\frac{S\{\forall x.\forall x.F\}}{S\{\forall x.F\}}$$
  - E° is non-empty and F° is empty: This is similar to the previous case.
  - E° and F° are both empty: This is impossible as the premise would not be a propositional encoding.
- ac (contracting an ordinary atom): This can trivially be lifted.
- m that is not in the situation of (5): Then now encoding of a quantifier is affected and the instance of m can be lifted.⟦**TODO:** medial permutation!!!⟧
- m/ac_x as in situation (5): We must have $R_1\{x\} \equiv x \vee E$ for some E and $R_2\{x\} \equiv x \vee F$ for some F with $R\{x\} \equiv x \vee E \vee F$. Otherwise, the application of $\mathsf{ac}_x^{\equiv}$ would not be correct. We have the following four cases:
  - E and F are both non-empty: Then (5) is (modulo omitted applications of $\equiv$):
    $$\begin{array}{r} \mathsf{m}\,\dfrac{S\{((x \vee E) \wedge C) \vee ((x \vee F) \wedge D)\}}{S\{((x \vee E) \vee (x \vee F)) \wedge (C \vee D)\}} \\ \mathsf{ac}_x^{\equiv}\,\dfrac{\phantom{S\{((x \vee E) \vee (x \vee F)) \wedge (C \vee D)\}}}{S\{(x \vee E \vee F) \wedge (C \vee D)\}} \end{array}$$

[5] Note that whenever we have formulas P and Q with $P^\circ \equiv^\circ Q^\circ$ then $P \equiv Q$.

which can be lifted to

$$\begin{array}{ll} & S\{((\forall x.E)\wedge C)\vee((\forall x.F)\wedge D)\} \\ \hline \mathsf{m} & S\{((\forall x.E)\vee(\forall x.F))\wedge(C\vee D)\} \\ \hline \mathsf{m}_\forall & S\{(\forall x.(E\vee F))\wedge(C\vee D)\} \end{array}$$

[[[**Jui-Hsuan:** maybe need some words to exclude the case in which C (or D) is a propositional variable.]]][[[**Lutz:** shit. (you mean a "first order variable") this actually can happen. then we have another $\mathsf{m}_\exists$]]]

- E is empty and F is not: Then (5) becomes

$$\begin{array}{ll} & S\{(x\wedge C)\vee((x\vee F)\wedge D)\} \\ \hline \mathsf{m} & S\{(x\vee(x\vee F))\wedge(C\vee D)\} \\ \hline \mathsf{ac}^{\equiv}_{x} & S\{(x\vee F)\wedge(C\vee D)\} \end{array}$$

The conclusion is the propositional encoding of $S\{(\forall x.F)\wedge(C\vee D)\}$ and the premise is the propositional encoding of $S\{(\exists x.C)\vee((\forall x.F)\vee D)\}$. Also note that no m-instance can break up the conjunction in $x\wedge C$ in the premise. Hence, φ maps an existential to a universal, which is ruled out by the definition. Hence, this case cannot occur.

- E is non-empty and F is empty: This case is similar to the previous subcase.
- E and F are both empty: Then (5) is

$$\begin{array}{ll} & S\{(x\wedge C)\vee(x\wedge D)\} \\ \hline \mathsf{m} & S\{(x\vee x)\wedge(C\vee D)\} \\ \hline \mathsf{ac}^{\equiv}_{x} & S\{x\wedge(C\vee D)\} \end{array}$$

which can be lifted immediately to

$$\begin{array}{ll} & S\{(\exists x.C)\vee(\exists x.D)\} \\ \hline \mathsf{m}_\exists & S\{\exists x.(C\vee D)\} \end{array}$$

Thus Φ_2° can be lifted to $\begin{array}{c} A\sigma_\varphi \\ \{\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\,\Big\|\,\Phi_2 \\ B_1 \end{array}$. We construct Φ by composing Φ_2 and Φ_1. Then $\widehat{\Phi}$ can be constructed by rectifying Φ, where the variables to be used in A are already given. That $\varphi=\lfloor\widehat{\Phi}\rfloor$ follows immediately from the construction. □

X. Summary and Proof of Main Result

The only theorem of Section VI that has not yet been proved is Theorem 19 establishing the full correspondence between decomposed proofs in $\mathsf{KS1}$ and combinatorial proofs. We show the proof here, by summarizing the results of the previous two Sections VIII and IX.

Proof of Theorem 19. First, asssume we have a combinatorial proof $\varphi\colon\mathcal{C}\to\mathcal{A}$ be a combinatorial proof and a formula A with $\mathcal{A}=[\![A]\!]$. Let C be a formula with $[\![C]\!]=\mathcal{C}$, and let σ_φ be the substitution induced by φ. By Lemma 44 there is a derivation

$$\begin{array}{c} C\sigma_\varphi \\ \{\mathsf{w},\mathsf{ac},\mathsf{c}_\forall,\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\,\Big\|\,\Phi_2 \\ A \end{array}$$

Since $\mathcal{C}$ is a fonet, we have by Theorem 39 a derivation

$$\begin{array}{c} \mathsf{t} \\ \mathsf{MLS1}^{\mathsf{X}}\,\Big\|\,\Phi'_1 \\ C \end{array}$$

This derivation remains valid if we apply the substitution σ_φ to every line in Φ'_1, yielding the derivation Φ_1 of $C\sigma_\varphi$ as desired.

Conversely, assume we have a decomposed derivation

$$\begin{array}{c} \mathsf{t} \\ \mathsf{MLS1}^{\mathsf{X}}\,\Big\|\,\Phi_1 \\ A' \\ \{\mathsf{w},\mathsf{ac},\mathsf{m},\mathsf{m}_\forall,\mathsf{m}_\exists,\equiv\}\,\Big\|\,\Phi_2 \\ A \end{array} \tag{6}$$

Then we can transform Φ_1 into a recified form $\widehat{\Phi}_1$, proving $\widehat{A}'$. By Theorem 33, the linked fograph $[\![\widehat{\Phi}_1]\!]=\langle[\![\widehat{A}']\!],\sim_{\widehat{\Phi}_1}\rangle$ is a fonet. Then, by Lemma 41, there is a rectified derivation $\begin{array}{c} \widehat{A'} \\ \{\mathsf{w},\widehat{\mathsf{ac}},\widehat{\mathsf{c}_\forall},\mathsf{m},\widehat{\mathsf{m}_\forall},\widehat{\mathsf{m}_\exists},\equiv\}\,\Big\|\,\widehat{\Phi_2} \\ \widehat{A} \end{array}$ whose induced map $\lfloor\widehat{\Phi_2}\rfloor\colon[\![\widehat{A'}]\!]\to[\![\widehat{A}]\!]$ is the same a the induced map $\lfloor\Phi_2\rfloor\colon[\![A']\!]\to[\![A]\!]$ of Φ_2. By Lemma 42, this map is a skew bifibration. Hence, we have a combinatorial proof $\varphi\colon\mathcal{C}\to[\![A]\!]$ with $\mathcal{C}=[\![\widehat{A'}]\!]$. [[[**Lutz:** shit, something's wrong...]]] □

Note that Theorem 19 shows at the same time soundness, completeness, and full completeness, as

1) every proof in $\mathsf{KS1}$ can be translated into a combinatorial proof, and
2) every combinatorial proof is the image of a $\mathsf{KS1}$-proof under that translation.

XI. Conclusion

In this paper we have uncovered a close correspondence between first-order combinatorial proofs and decomposed deep inference derivations of system $\mathsf{KS1}$, and we have shown that every proof in $\mathsf{KS1}$ has such a decomposed form.

The most surprising discovery for us was that all technical difficulties in our work could be reduced (in a non-trivial way) to the propositional setting.

The obvious next step in our research is to investigate proof composition and normalisation of first-order combinatorial proofs. Even in the propositional setting, the normalization of combinatorial proofs is underdevelopped. There exist two different procedures for cut elimination [10], [**?**], but both have their insufficiencies, and there is no general theory.

[[[**Lutz:** do we want/can say more here?]]]

References

[1] G. Frege, *Begriffsschrift*. Louis Nebert, Halle, 1879, English Translation in: J. van Heijenoort (ed.), From Frege to Gödel, Harvard University Press: 1977.

[2] D. Hilbert, "Die logischen Grundlagen der Mathematik," *Mathematische Annalen*, vol. 88, pp. 151–165, 1922.

[3] G. Gentzen, "Untersuchungen über das logische Schließen. I." *Mathematische Zeitschrift*, vol. 39, pp. 176–210, 1935.

[4] G. Gentzen, “Untersuchungen über das logische Schließen. II.” *Mathematische Zeitschrift*, vol. 39, pp. 405–431, 1935.

[5] R. M. Smullyan, *First-Order Logic*. Berlin: Springer-Verlag, 1968.

[6] J. A. Robinson, “A Machine-Oriented Logic Based on the Resolution Principle,” *Journal of the ACM*, vol. 12, pp. 23–41, 1965.

[7] D. Hilbert, “Mathematische Probleme,” *Nachrichten der Königlichen Gesellschaft der Wissenschaften zu Göttingen, mathematisch-physikalische Klasse*, vol. 3, pp. 253–297, 1900.

[8] R. Thiele, “Hilbert’s Twenty-fourth Problem,” *American Mathematical Monthly*, vol. 110, pp. 1–24, 2003.

[9] D. Hughes, “Proofs Without Syntax,” *Annals of Mathematics*, vol. 164, no. 3, pp. 1065–1076, 2006.

[10] D. Hughes, “Towards Hilbert’s 24th Problem: Combinatorial Proof Invariants:(preliminary version),” *Electronic Notes in Theoretical Computer Science*, vol. 165, pp. 37–63, 2006.

[11] L. Straßburger, “The Problem of Proof Identity, and Why Computer Scientists Should Care About Hilbert’s 24th Problem,” *Philosophical Transactions of the Royal Society A*, vol. 377, no. 2140, p. 20180038, 2019.

[12] L. Straßburger, “Combinatorial Flows and Their Normalisation,” in *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[13] L. Straßburger, “Combinatorial Flows and Proof Compression,” Inria Saclay, Research Report RR-9048, 2017. [Online]. Available: https://hal.inria.fr/hal-01498468

[14] M. Acclavio and L. Straßburger, “From Syntactic Proofs to Combinatorial Proofs,” in *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, D. Galmiche, S. Schulz, and R. Sebastiani, Eds., vol. 10900. Springer, 2018, pp. 481–497.

[15] M. Acclavio and L. Straßburger, “On Combinatorial Proofs for Logics of Relevance and Entailment,” in *26th Workshop on Logic, Language, Information and Computation (WoLLIC 2019)*, R. Iemhoff and M. Moortgat, Eds. Springer, 2019.

[16] M. Acclavio and L. Straßburger, “On Combinatorial Proofs for Modal Logic,” in *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. Springer, 2019, pp. 223–240.

[17] W. Heijltjes, D. Hughes, and L. Straßburger, “Intuitionistic Proofs Without Syntax,” in *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 2019, pp. 1–13.

[18] D. Hughes, “First-order Proofs Without Syntax,” *arXiv preprint arXiv:1906.11236*, 2019.

[19] J. Herbrand, “Recherches sur la Théorie de la Démonstration,” Ph.D. dissertation, University of Paris, 1930.

[20] K. Brünnler, “Cut Elimination Inside a Deep Inference System for Classical Predicate Logic,” *Studia Logica*, vol. 82, no. 1, pp. 51–71, 2006.

[21] L. Straßburger, “A Characterization of Medial as Rewriting Rule,” in *International Conference on Rewriting Techniques and Applications*. Springer, 2007, pp. 344–358.

[22] A. S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*. Cambridge University Press, 2000, no. 43.

[23] J.-Y. Girard, “Linear Logic,” vol. 50, pp. 1–102, 1987.

[24] J.-Y. Girard, “Quantifiers in Linear Logic,” *Temi e prospettive della logica e della filosofia della scienza comtemporanee*, vol. 1, pp. 95–130, 1988.

[25] A. Fleury and C. Retoré, “The Mix Rule,” *Math. Structures in Comp. Science*, vol. 4, no. 2, pp. 273–285, 1994.

[26] G. Bellin, “Subnets of Proof-nets in Multiplicative Linear Logic with MIX,” *Mathematical Structures in Computer Science*, vol. 7, no. 6, pp. 663–699, 1997.

[27] R. J. Duffin, “Topology of Series-parallel Networks,” *Journal of Mathematical Analysis and Applications*, vol. 10, no. 2, pp. 303–318, 1965.

[28] K. Brünnler, “Deep Inference and Symmetry for Classical Proofs,” Ph.D. dissertation, Technische Universität Dresden, 2003.

[29] B. Ralph, “Modular Normalisation of Classical Proofs,” Ph.D. dissertation, University of Bath, 2019.

[30] A. A. Tubella and A. Guglielmi, “Subatomic Proof Systems: Splittable Systems,” *ACM Transactions on Computational Logic (TOCL)*, vol. 19, no. 1, pp. 1–33, 2018.

[31] K. Brünnler, “Locality for Classical Logic,” *Notre Dame Journal of Formal Logic*, vol. 47, no. 4, pp. 557–580, 2006. [Online]. Available: http://www.iam.unibe.ch/ kai/Papers/LocalityClassical.pdf

[32] J.-Y. Girard, “Proof-nets: The Parallel Syntax for Proof-theory,” in *Logic and Algebra*, A. Ursini and P. Agliano, Eds. Marcel Dekker, New York, 1996.

[33] A. Guglielmi and L. Straßburger, “Non-commutativity and MELL in The Calculus of Structures,” in *Computer Science Logic, CSL 2001*, ser. LNCS, L. Fribourg, Ed., vol. 2142. Springer-Verlag, 2001, pp. 54–68.

[34] D. Hughes, “Unification Nets: Canonical Proof Net Quantifiers,” in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, 2018, pp. 540–549.

[35] K. Brünnler and A. F. Tiu, “A Local System for Classical Logic,” in *International Conference on Logic for Programming Artificial Intelligence and Reasoning*. Springer, 2001, pp. 347–361.

[36] L. Straßburger, “Linear logic and Noncommutativity in the Calculus of Structures,” Ph.D. dissertation, Technische Universität Dresden, 2003.

[37] V. Danos and L. Regnier, “The Structure of Multiplicatives,” *Archive for Mathematical Logic*, vol. 28, no. 3, pp. 181–203, 1989.

[38] C. Retoré, “Handsome Proof-nets: Perfect Matchings and Cographs,” *Theoretical Computer Science*, vol. 294, no. 3, pp. 473–488, 2003.

[39] C. Retoré, “Handsome Proof-nets: R&B-Graphs, Perfect Matchings and Series-parallel Graphs,” INRIA, Research Report RR-3652, 1999. [Online]. Available: https://hal.inria.fr/inria-00073020

[40] L. Straßburger, “Deep Inference and Expansion Trees for Second-order Multiplicative Linear Logic,” *Mathematical Structures in Computer Science*, vol. 29, pp. 1030–1060, 2019.

[41] B. Ralph and L. Straßburger, “Towards a Combinatorial Proof Theory,” in *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. Springer, 2019, pp. 259–276.

[[[**Lutz:** OLD STUFF, TO BE REMOVED EVENTUALLY]]]

A. Unification Nets

[[[**TODO:**]]]

In this paragraph, we associate each formula A with its ***formula tree*** $\mathcal{F}(A)$, a directed tree with leaves labelled by atoms, internal nodes labelled by connectives and quantifiers, and edges directed from leaves to the root. For a sequent $\Gamma = A_1, \dots, A_n$, we denote with $\mathcal{F}(\Gamma)$, the forest formed by $\mathcal{F}(A_1), \dots, \mathcal{F}(A_n)$, i.e., the disjoint union of $\mathcal{F}(A_i)$'s. The ***roots*** of $\mathcal{F}(\Gamma)$ are the roots of A_i's

Let Γ be a sequent in $\mathsf{MLL1}^{\mathsf{X}}$. Consider the forest $\mathcal{F}(\Gamma)$. A ***link*** on Γ is a pair of leaves whose atoms are pre-dual. A ***linking*** λ on Γ is a set of disjoint links such that each leaf of $\mathcal{F}(\Gamma)$ is either labelled by t or in exactly one link. Similar to the set of links in linked fographs, a linking can be seen as a unification problem, and a ***dualizer*** δ of the linking λ is an assignment unifying all the links in λ. There exists a ***most general dualizer*** of λ if λ has a dualizer. [[[**Jui-Hsuan:** Now I use the same terminology as for linked fographs]]][[[**Lutz:** use δ for the dualizer (or even better, make it a macro)]]] A ***dependency*** is a pair $(\bullet\exists x, \bullet\forall y)$ of nodes such that the most general dualizer assigns to x a term containing y.

Let λ is a linking on Γ that has a dualizer. The ***unification structure*** $\mathcal{U}(\lambda)$ associated with λ is the forest $\mathcal{F}(\Gamma)$ together with an undirected edge between leaves l and l' for every link $\{l, l'\}$ in λ and a directed edge from $\bullet\exists x$ to $\bullet\forall y$ for every dependency $(\bullet\exists x, \bullet\forall y)$.

A ***switching graph*** of a unification structure $\mathcal{U}(\lambda)$ is any derivative of $\mathcal{U}(\lambda)$ obtained by keeping only one edge into each $\vee$ and $\forall$ and undirecting remaining edges. A linking is ***correct*** if it is unifiable and all of the switching graphs of its associated unification structure are acyclic.

Definition 45. A ***unification net*** on a sequent Γ is a correct linking on Γ.

B. Translation between Unification Nets and $\mathsf{MLL1}^{\mathsf{X}}$

[[[**TODO:**]]]

Theorem 46. *If a sequent is provable in* $\mathsf{MLL1}^{\mathsf{X}}$*, then there exists a unification net on it.*

Proof. We proceed by induction on the proof of $\vdash \Gamma$ in $\mathsf{MLL1}^{\mathsf{X}}$, making a case analysis on the bottommost rule instance:

- $\mathsf{ax}\,\dfrac{}{\vdash a, \bar{a}}$: the linking $\{a, \bar{a}\}$ is correct.
- $\mathsf{t}\,\dfrac{}{\vdash \mathsf{t}}$: the empty linking is correct.
- $\mathsf{mix}\,\dfrac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}$: By induction hypothesis, there is a correct linking on Γ and another one on Δ, their union giving a correct linking on Γ, Δ.
- $\vee\,\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}$: By induction hypothesis, there is a correct linking on Γ, A, B, and it is correct on $\Gamma, A \vee B$ as well.
- $\wedge\,\dfrac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \wedge B, \Delta}$: By induction hypothesis, there is a correct linking on Γ, A and another one on B, Δ, their union giving a correct linking on $\Gamma, A \wedge B, \Delta$.
- $\exists\,\dfrac{\vdash \Gamma, A[x/t]}{\vdash \Gamma, \exists x.A}$: By induction hypothesis, there is a correct linking λ on $\Gamma, A[x/t]$. For each atom in $\Gamma, A[x/t]$, there is a corresponding atom in $\Gamma, \exists x.A$. There is therefore a linking λ' on $\Gamma, \exists x.A$ obtained from λ via this correspondence, and it is not difficult to check that λ' is correct as well.
- $\forall\,\dfrac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.A}$ (x not free in Γ) : By induction hypothesis, there is a correct linking on Γ, A, and it is easy to see that it is a correct linking on $\Gamma, \forall x.A$ as well.

This allows to define a translation $\lceil\cdot\rceil$ from proofs in $\mathsf{MLL1}^{\mathsf{X}}$ to unification nets. □

Theorem 47. *Any unification net can be obtained via the translation* $\lceil\cdot\rceil$ *given in Theorem 46.*

To prove this theorem, we need some basic lemmas about connected components in switching graphs of unification nets.

Lemma 48. *The number of connected components of an acylic graph* $\mathcal{G} = \langle V_{\mathcal{G}}, E_{\mathcal{G}} \rangle$ *is equal to* $|E_{\mathcal{G}}| - |V_{\mathcal{G}}|$.

Proof. By a straightforward induction on $|V_{\mathcal{G}}|$. □

Lemma 49. *The number of connected components is the same for any switching graph of a unification net.*

Proof. An immediate consequence of Lemma 48. □

In the proof, we also use the notion of ***frame*** introduced by Hughes in [34].

Definition 50. Let λ be a unification net on an $\mathsf{MLL1}^{\mathsf{X}}$ sequent Γ. We define the ***frame*** of λ by exhaustively applying the following subformula rewriting steps, to obtain a linking λ_m on an $\mathsf{MLL} + \mathsf{mix}$ sequent Γ_m:

1) **Encode dependencies as fresh links.** For each dependency $\exists x \to \forall y$, with corresponding subformulas $\exists x A$ and $\forall y B$, we add a fresh link as follows. Let P be a fresh (nullary) predicate symbol. Replace $\exists x A$ with $P \wedge \exists x A$ and $\forall y B$ with $\overline{P} \vee \forall y B$, and add an axiom link between P and $\overline{P}$.
2) **Erase quantifiers.** After step 1, erase all the quantifiers. (We no longer need their leaps since they are encoded as links in step 1.
3) **Simplify atoms.** After step 2, replace every predicate $Pt_1 \cdots t_n$ with a nullary predicate symbol P.

Note that the linking λ_m is a valid $\mathsf{MLL} + \mathsf{mix}$ proof net.

Lemma 51. *Suppose that λ is a* $\mathsf{MLL}+\mathsf{mix}$ *proof net which is connected and such that any of its switching graphs is not connected. Then there exists a $\vee$ node in $\mathcal{U}(\lambda)$ such that λ is correct on the sequent Γ' obtained from Γ by replacing this $\vee$ by a $\wedge$.*

Proof. Suppose that such a $\vee$ node does not exist. Then it is clear that for any two nodes, there exists a switching graph containing a path between them and this path corresponds to an AE-path in [38]. By [38, Propostion 3], λ corresponds to a sequent proof that does not use mix, which implies the connectedness of the switching graphs of λ. Contradiction. **⟦TO CHECK: ⟧** □

Lemma 52. *Suppose that λ is a* $\mathsf{MLL1}^{\mathsf{X}}$ *proof net which is connected and such that any of its switching graphs is not connected. Then there exists a $\vee$ node in $\mathcal{U}(\lambda)$ such that λ is correct on the sequent Γ' obtained from Γ by replacing this $\vee$ by a $\wedge$.*

Proof. Consider the frame λ_m of λ. The number of any switching graph of $\mathcal{U}(\lambda)$ is equal to that of $\mathcal{U}(\lambda_m)$. Apply Lemma 51 and it is clear that such $\vee$ cannot be one of the fresh $\vee$'s added during the frame construction. □

We can now give the proof of Theorem 47.

Proof of Theorem 47. Let λ be a unification net on Γ. We proceed by induction on the number of connected components of the unification structure $\mathcal{U}(\lambda)$:

- If there is only one connected component, we proceed by induction on the number k of connected components of any switching graph of $\mathcal{U}(\lambda)$. If $k=1$, we obtain a proof Φ in $\mathsf{MLL1}^{\mathsf{X}}$ such that $\lceil\Phi\rceil=\lambda$ by applying [34, Theorem 3]. If $k>1$, using the Lemma 52, we obtain a sequent Γ' on which λ is correct by transforming a $\vee$ node into a $\wedge$. By induction hypothesis, there is a proof Φ' in $\mathsf{MLL1}^{\mathsf{X}}$ whose translation is λ. By considering the $\wedge$ rule instance corresponding to the $\wedge$ node in Φ', we have: $$\Phi' = \cfrac{\cfrac{\overset{\Phi_1}{\vdash \Delta_1, A} \qquad \overset{\Phi_2}{\vdash B, \Delta_2}}{\vdash \Delta_1, A\wedge B, \Delta_2}\,\wedge}{\cfrac{\vdots}{\vdash \Gamma'}}$$ We can thus obtain a proof Φ of Γ: $$\Phi = \cfrac{\cfrac{\cfrac{\overset{\Phi_1}{\vdash \Delta_1, A} \qquad \overset{\Phi_2}{\vdash B, \Delta_2}}{\vdash \Delta_1, A, B, \Delta_2}\,\mathsf{mix}}{\vdash \Delta_1, A\vee B, \Delta_2}\,\vee}{\cfrac{\vdots}{\vdash \Gamma}}$$ such that $\lceil\Phi\rceil=\lambda$.
- If there are $n>1$ connected components, add a fresh $\vee$ node connecting two formulas belonging to different connected components of Γ to get a new sequent Γ'. Define a unification net λ' on Γ' using the same linking as λ. By induction hypothesis, since $\mathcal{U}(\lambda')$ has $n-1$ connected components, there is a $\mathsf{MLL1}^{\mathsf{X}}$ proof Φ' such that $\lceil\Phi'\rceil=\lambda'$. Consider the $\vee$ rule instance corresponding to the $\vee$ node in question. Since $\vee$ is invertible, we can permute downwards this rule instance until it becomes the last rule of the proof (note that this transformation does not change the image of the proof by the translation $\lceil\cdot\rceil$) to get a new proof Φ'' of Γ'. By deleting the last rule instance from Φ'', we obtain a proof Φ of Γ such that $\lceil\Phi\rceil=\lambda$. **⟦TO CHECK: ⟧**

We proceed by induction on the number of connectives in Γ. In the base case, Γ is of the form

$$p_1(t_{11},\dots,t_{1n_1}),\overline{p_1}(t_{11},\dots,t_{1n_1}),\dots,$$
$$p_k(t_{k1},\dots,t_{kn_k}),\overline{p_k}(t_{k1},\dots,t_{kn_k}),\underbrace{\mathsf{t},\dots,\mathsf{t}}_{m\text{ times}}$$

and λ is the linking $\{(a_1,\overline{a_1}),\dots,(a_k,\overline{a_k})\}$, where $a_i = p_i(t_{i1},\dots,t_{in_i})$, which equals to $\lceil\Pi\rceil$, where Π is the proof consisting of m instances of the t rule, n instances $\mathsf{ax}\,\dfrac{}{\vdash a_i,\overline{a_i}}$ of the ax rule, and followed by $m+k-1$ instances of the mix rule.

Now we consider the inductive cases:

- $\Gamma=\Delta, A\vee B$: Let $\Gamma'=\Delta, A, B$. Define λ' on Γ' using the same links as λ by identifying the leaves of $\mathcal{F}(\Gamma')$ with those of $\mathcal{F}(\Gamma)$. We now check that λ' is a unification net:
  - The most general dualizer of λ is also the most general dualizer of λ' as they correspond to the same unification problem. Hence, the unification structure $\mathcal{U}(\lambda')$ is equal to the restriction of $\mathcal{U}(\lambda)$ to the nodes of $\mathcal{F}(\Gamma')$.
  - Every switching graph of λ' is acyclic: if there were some switching graph of $\mathcal{U}(\lambda')$ containing a cycle, it would induce a switching graph of $\mathcal{U}(\lambda)$ containing also a cycle, by adding an edge from the root of $\mathcal{F}(A)$ to the $\vee$ node in question.
- $\Gamma=\Delta,\forall x.A$: Let $\Gamma'=\Delta, A$. Define λ' on Γ' using the same links as λ. We now check that λ' is a unification net:
  - The most general dualizer of λ is also the most general dualizer of λ' as they correspond to the same unification problem.
  - Every switching graph of $\mathcal{U}(\lambda')$ is acyclic: if there were some switching graph of $\mathcal{U}(\lambda')$ containing a cycle, it would induce a switching graph of $\mathcal{U}(\lambda)$ containg also a cycle, by adding an edge from the root of $\mathcal{F}(A)$ to the $\forall$ node in question.
- $\mathcal{F}(\Gamma)$ has a root $\exists x$ with no outgoing dependency edge:

□

C. Translation between Unification Nets and Fonets

[[TODO:]]

XII. First-order Combinatorial Proofs

A. First-order Logic

In this paper, we also use some ***deep inference*** [35] rules that are in the following form:

$$\frac{\vdash S\{A\}}{\vdash S\{B\}}$$

where $S\{\ \}$ stands for a ***context***, which corresponds to a sequent with a hole taking the place of an atom, and $S\{A\}$ represents the sequent or formula obtained by replacing the hole in $S\{\ \}$ with the formula A. Formally,

$$C ::= \Box \mid A \vee C \mid C \wedge A \mid \exists x C \mid \forall x C.$$

$$S ::= C \mid A, S \mid S, A$$

where A is a formula. The above rule can be thus seen as the rewriting rule $A \to B$.

We use the notation $\begin{array}{c} A \\ \|\mathcal{P} \\ B \end{array}$ for denoting that there is a derivation from premise $\vdash S\{A\}$ to conclusion $\vdash S\{B\}$ in system $\mathcal{P}$ for any context S.

B. Graphs

C. First-order combinatorial proofs

D. MLL1$^{\mathsf{X}}$ and Unification Nets

In MLL1$^{\mathsf{X}}$, terms, atoms, formulas are defined as in first-order logic. For simplicity, we choose to use $\vee$ and $\wedge$ instead of $⅋$ and $\otimes$ which are generally used in the presentation of linear logic. A formula A is identified with its ***formula tree*** $\mathcal{F}(A)$, a directed tree with leaves labelled by atoms, internal nodes labelled by connectives and quantifiers, and edges directed from leaves to the root. A ***sequent*** Γ is simply a disjoint union of formulas. We write comma for disjoint union.

Sequents are proved using the inference rules of MLL1$^{\mathsf{X}}$:

$$\frac{}{\vdash A, \neg A}\,\mathsf{ax} \qquad \frac{\vdash \Gamma, A \qquad \vdash \Delta, \neg A}{\vdash \Gamma, \Delta}\,\mathsf{cut}$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \wedge B}\,\wedge \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}\,\vee$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x A}\,\forall\ (x \notin fv(\Gamma)) \qquad \frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x A}\,\exists$$

Fig. 6. Sequent calculus for MLL1$^{\mathsf{X}}$

We also consider the mix rule:

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\,\mathsf{mix}$$

Let Γ be a sequent in $\mathsf{MLL1} + \mathsf{mix}$. A ***link*** on Γ is a pair of leaves whose atoms are pre-dual. A ***linking*** on Γ is a set of disjoint links such that each leaf of Γ is in exactly one link. Similar to the set of links in the linked fograph, a linking can be seen as a unification problem, and a link is said ***unifiable*** if the corresponding unification problem is solvable. ***Dependencies*** are defined as previously.

XIII. From first-order logic to combinatorial proofs

A. Decomposition Theorem

Consider the following deep inference rules [35]:

$$\frac{\vdash S\{A \vee A\}}{\vdash S\{A\}}\,\mathsf{c} \qquad\qquad \frac{\vdash S\{\mathsf{f}\}}{\vdash S\{A\}}\,\mathsf{w}$$

Note that the ctr (resp. wk) rule in LK is derivable in $\{\mathsf{c}, \vee\}$ (resp. $\{\mathsf{w}, \mathsf{f}\}$) and that c and w rules permute downwards with the non-structural rules of LK.

$$\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\,\mathsf{ctr} \quad\rightsquigarrow\quad \cfrac{\cfrac{\vdash \Gamma, A, A}{\vdash \Gamma, A \vee A}\,\vee}{\vdash \Gamma, A}\,\mathsf{c}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, A}\,\mathsf{wk} \quad\rightsquigarrow\quad \cfrac{\cfrac{\vdash \Gamma}{\vdash \Gamma, \mathsf{f}}\,\mathsf{f}}{\vdash \Gamma, A}\,\mathsf{w}$$

We also give an example to show how rule permutation works:

$$\cfrac{\cfrac{\Gamma, A \vee A}{\Gamma, A}\,\mathsf{c} \qquad \Delta, B}{\Gamma, \Delta, A \wedge B}\,\wedge \quad\rightsquigarrow\quad \cfrac{\cfrac{\Gamma, A \vee A \qquad \Delta, B}{\Gamma, \Delta, (A \vee A) \wedge B}\,\wedge}{\Gamma, \Delta, A \wedge B}\,\mathsf{c}$$

We want to establish the following theorem:

Theorem 53. *Let Γ be a sequent. Then there is a proof of Π in $\mathsf{LK} + \mathsf{mix}$ iff there is a proof of some sequent Δ in $\mathsf{MLL1} + \mathsf{mix}$ and a derivation from Δ to Γ consisting of the c and w rules only.*

Proof. ($\Rightarrow$) This direction comes from the above observation: it suffices to permute downwards all the instances of the c and w rules.

($\Leftarrow$) We regard the proof in $\mathsf{MLL1} + \mathsf{mix}$ as a proof in $\mathsf{LK} + \mathsf{mix}$. Then we put the derivation consisting of only c and w under the proof in $\mathsf{LK} + \mathsf{mix}$. Now we try to permute all the instances c and w upwards with the rules of LK and mix. For the c part, the only non-trivial case is the permutation with the $\vee$ rule where the formula generated is $A \vee A$.

$$\cfrac{\cfrac{\vdash \Gamma, A, A}{\vdash \Gamma, A \vee A}\,\vee}{\vdash \Gamma, A}\,\mathsf{c} \quad\rightsquigarrow\quad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\,\mathsf{ctr}$$

In this case, the permutation of this instance of c stops and we continue with the remaining instances.

For the w part, the only non-trivial case is the permutation with the f rule (or the instance of wk where f is introduced):

$$\cfrac{\cfrac{\vdash \Gamma}{\vdash \Gamma, f}\,\mathsf{f}}{\vdash \Gamma, A}\,\mathsf{w} \quad\rightsquigarrow\quad \frac{\vdash \Gamma}{\vdash \Gamma, A}\,\mathsf{wk}$$

In this case, the permutation of this instance of w stops and we continue with the remaining instances.

□

D. Hughes proves in [34] the soundness and completeness of unification nets with respect to $\mathsf{MLL1} + \mathsf{mix}$. In the following, we establish the equivalence between unification nets and fonets.

B. Equivalence between unification nets and fonets

In the following, we usually confound a vertex with its label.

Definition 54. A ***switching path*** of a unification structure $U(\lambda)$ is a path in a switching graph of $U(\lambda)$.

Definition 55. A ***switching path*** of a formula tree $\mathcal{F}(A)$ is a path in $\mathcal{F}(A)$ that does not go through both incoming edges of a $\vee$.

Proposition 56. *In a formula tree, the root is connected to every vertex by a switching path.*

Now we give the key proposition relating a fograph to its corresponding formula tree.

Proposition 57. *Let u and v be two distinct vertices of a fograph $\llbracket(\rrbracket A)$, then we have the equivalence between:*

- *u and v are adjacent in $\llbracket(\rrbracket A)$*
- *u and v are connected by a switching path of $\mathcal{F}(A)$, and if one of them is a universal quantifier, then the other is not a descendant of the former.*

Proof. By induction on A.

- If A is an atom, trivial.
- If $A = A_1 \wedge A_2$, then we distinguish two cases:
  - u and v are both in A_1 (resp. A_2): trivial by the induction hypothesis.
  - one of them is in A_1 and the other is in A_2: they are adjacent in $\llbracket(\rrbracket A)$ by definition. By Proposition 56, the one in A_1 (resp. A_2) is connected to the vertex representing A_1 (resp. A_2) by a switching path. Together with the two edges incident to $A_1 \wedge A_2$, we obtain a switching path connecting u and v.
- If $A = A_1 \vee A_2$, then we distinguish two cases:
  - u and v are both in A_1 (resp. A_2): trivial by the induction hypothesis.
  - one of them is in A_1 and the other is in A_2: they are not adjacent in $\llbracket(\rrbracket A)$ by definition. It is clear that they are not connected by a switching path.
- If $A = \exists x\ A'$, then we distinguish two cases:
  - u and v are both in A': trivial by the induction hypothesis.
  - one of them is $\exists x$ and the other is in A': trivial by Proposition 56
- If $A = \forall x\ A'$, then we distinguish two cases:
  - u and v are both in A': trivial by the induction hypothesis.
  - one of them is $\forall x$ and the other is in A': they are not adjacent in $\llbracket(\rrbracket A)$ by definition and it is clear that the former is a descendant of $\forall x$.

□

Proposition 58. *If there exists an induced bimatching of the linked fograph $G = \llbracket(\rrbracket A)$, then there exists a switching graph of the corresponding unification net which contains a cycle.*

Proof. Suppose that there exists a set W inducing a bimatching in G. Then (W, E_G) and (W, L_G) are matchings.
Let E_W (resp. L_W) be the restriction of E_G (resp. L_G) to W. If $E_W \cap L_W \neq \emptyset$, then there exist u and v such that $uv \in E_G$ and $uv \in L_G$. By Proposition 57, there exists a switching path of the formula tree of A. Together with the leap uv, this path induces a cycle in a switching graph of the corresponding unification structure.
We can now suppose that E_W and L_W are disjoint. It is not difficult to see the existence of an alternating and elementary cycle in the bicoloured graph $(W, E_W \uplus L_W)$, i.e. a cycle of which the edges are alternately in E_W and L_W and containing no two equal vertices. By Proposition 57, this cycle induces a cycle in the unification structure. Now we want to construct a switching graph that contains this cycle.
Consider a universal quantifier $\forall x$. If $\forall x \notin W$, then we keep the incoming edge from its direct subformula and remove all the dependencies. Otherwise, since (W, L_G) is a matching, there exists a unique existential quantifier adjacent to $\forall x$ and we keep thus the corresponding edge in the unification structure.
Now consider a $\vee$. We distinguish three cases:

- the cycle goes through none of the two branches (incoming edges) of the $\vee$: we can choose an arbitrary switching for this $\vee$
- the cycle goes through exactly one branch: we choose the corresponding switching
- the cycle goes through both branches: this means that there exist $v_L \in W$ (resp. v_R) in the left (resp. right) branch, u_L, $u_R \in W$, such that $u_L v_L, u_R v_R \in E_W$ and that the corresponding switching path from u_L to v_L (resp. from u_R to v_R) goes through the left (resp. right) edge of $\vee$.
  The red (resp. blue) path is the switching path corresponding to the edge $u_L v_L$ (resp. $u_R v_R$) in E_W.
  It is clear that u_L (resp. u_R) is not in the branches of the $\vee$. Otherwise, there will be no switching path from u_L to v_L
  By Proposition 57, we know that u_L and u_R are not universal quantifiers which are ancestors the $\vee$ and that there exist one switching path from u_L to v_L and one from u_R to v_R. In particular, there exist one switching path from u_L to the $\vee$ and one from the $\vee$ to v_R, and by concatenating the two, we obtain a switching path from u_L to v_R. By Proposition 57, u_L and v_R are thus adjacent in (W, E_G), which is impossible since (W, E_W) is a matching.

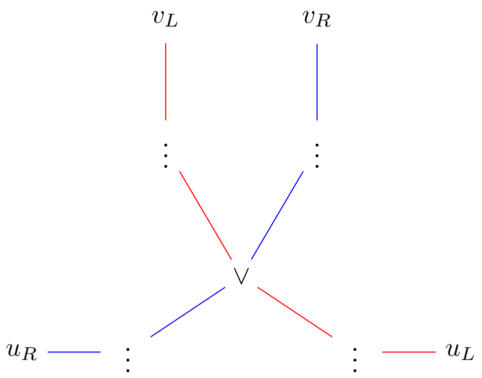


Fig. 7. A schema showing that the two branches of the same $\vee$ cannot be used in the cycle at the same time.

Notice that the switching paths here are in the underlying formula tree. We have to verify that they are compatible with the choices of switching made for universal quantifiers. That is, if $uv \in E_W$, then for all the universal quantifiers $\forall x$ on the switching path (in the formula tree), we have chosen in the first part of the proof to keep the only edge from the child of $\forall x$ to itself. In fact, if there exists a universal quantifier $w \in W$ on the switching path $u \to v$, then one of u and v is not a descendant of w. Moreover, if u (resp. v) is a universal quantifier, then w is not in its scope. By Proposition 57, $\{wu, wv\} \cap E_W \neq \emptyset$, which is impossible since (W, E_W) is a matching. We have thus constructed a switching graph containing this cycle. □

Proposition 59. *If one of the switching graphs of the unification structure of A contains a cycle or is not connected, then there exists an induced bimatching of the corresponding linked fograph.*

Proof. We use frames introduced by D. Hughes in Section 4 of [34].

Definition 60. Let θ be a unification structure on an $\mathsf{MLL1}^{\mathsf{X}}$ sequent Γ. We define the ***frame*** of θ by exhaustively applying the following subformula rewriting steps, to obtain a proof structure θ_m on an MLL sequent Γ_m:

1) **Encode dependencies as fresh links.** For each dependency $\exists x \to \forall y$, with corresponding subformulas $\exists x A$ and $\forall y B$, we add a fresh link as follows. Let P be a fresh (nullary) predicate symbol. Replace $\exists x A$ with $P \wedge \exists x A$ and $\forall y B$ with $\overline{P} \vee \forall y B$, and add an axiom link between P and $\overline{P}$.
2) **Erase quantifiers.** After step 1, erase all the quantifiers. (We no longer need their leaps since they are encoded as links in step 1.
3) **Simplify atoms.** After step 2, replace every predicate $Pt_1 \cdots t_n$ with a nullary predicate symbol P.

We have the following results:

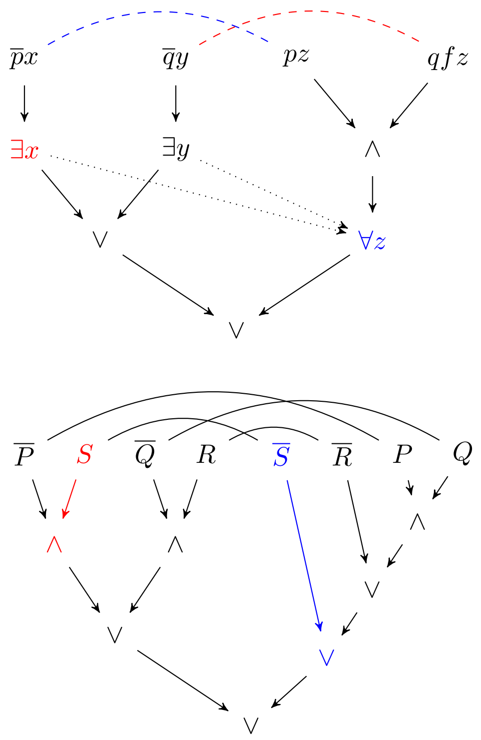


Fig. 8. A unification net and its frame. The colored part shows how the dependency $\exists x \to \forall z$ is transformed.

Let u and v be atoms or quantifiers in a unification structure θ. Then they are connected by a switching path in the unification structure if, and only if, their corresponding nodes are connected by a switching path in θ_m.

Consider now a switching graph H of a unification structure θ of A.

If H contains a cycle, then the corresponding switching graph of θ_m also contains a cycle. Hence, by applying the propositional results (Theorem 7) from [38], we conclude that there exists a chordless, alternating, and elementary cycle in the bicoloured graph $(W, E_W \uplus L_W)$, which corresponds to an induced bimatching in the linked fograph. (Note that the linked fograph (cograph) corresponding to θ_m is equivalent to the one corresponding to θ.)

□

C. From contraction/weakening to skew bifibrations

We first introduce the atomic contraction rule, the medial rule, and two rules on quantifiers.

$$\dfrac{\vdash S\{a \vee a\}}{\vdash S\{a\}}\ \mathsf{ac} \qquad \dfrac{\vdash S\{(A \wedge B) \vee (C \wedge D)\}}{\vdash S\{(A \vee C) \wedge (B \vee D)\}}\ \mathsf{m}$$

$$\dfrac{\vdash S\{\exists x A \vee \exists x B\}}{\vdash S\{\exists x (A \vee B)\}}\ \mathsf{m}_1\!\downarrow \qquad \dfrac{\vdash S\{\forall x A \vee \forall x B\}}{\vdash S\{\forall x (A \vee B)\}}\ \mathsf{m}_2\!\downarrow$$

Here, we also consider the equivalence generated by the associativity, commutativity of $\vee$ and the equations $\mathsf{t} \vee A \equiv \mathsf{t}$ and $\mathsf{f} \vee A \equiv A$.

Now we have the following lemma:

Lemma 61. *The contraction rule* c *is derivable for* $\{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\}$.

Proof. We prove that there is always $\begin{array}{c} A \vee A \\ \Vert\, {\scriptstyle \{\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ A \end{array}$ by structural induction on A.

- If $A = \mathsf{t}$ or $A = \mathsf{f}$, we have $\dfrac{\vdash S\{A \vee A\}}{\vdash S\{A\}}\ \equiv$. (the premiss and the conclusion are equivalent)
- If $A = a$, then we have $\dfrac{\vdash S\{a \vee a\}}{\vdash S\{a\}}\ \mathsf{ac}$
- If $A = A_1 \vee A_2$, then by the induction hypothesis, we have $\begin{array}{c} A_i \vee A_i \\ \Vert\, {\scriptstyle \{\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ A_i \end{array}$ for $i = 1, 2$.

  Hence, we have
  $$\begin{array}{c} \dfrac{\vdash S\{(A_1 \vee A_2) \vee (A_1 \vee A_2)\}}{\vdash S\{(A_1 \vee A_1) \vee (A_2 \vee A_2)\}}\ \equiv \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{A_1 \vee (A_2 \vee A_2)\} \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{A_1 \vee A_2\} \end{array}$$
- If $A = A_1 \wedge A_2$, then by the induction hypothesis, we have $\begin{array}{c} A_i \vee A_i \\ \Vert\, {\scriptstyle \{\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ A_i \end{array}$ for $i = 1, 2$.

  Hence, we have
  $$\begin{array}{c} \dfrac{\vdash S\{(A_1 \wedge A_2) \vee (A_1 \wedge A_2)\}}{\vdash S\{(A_1 \vee A_1) \wedge (A_2 \vee A_2)\}}\ \mathsf{m} \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{A_1 \wedge (A_2 \vee A_2)\} \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{A_1 \wedge A_2\} \end{array}$$
- If $A = \exists x A'$, then by the induction hypothesis, we have $\begin{array}{c} A' \vee A' \\ \Vert\, {\scriptstyle \{\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ A' \end{array}$.

  Hence, we have
  $$\begin{array}{c} \dfrac{\vdash S\{\exists x A' \vee \exists x A'\}}{\vdash S\{\exists x (A' \vee A')\}}\ \mathsf{m}_1\!\downarrow \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{\exists x A'\} \end{array}$$
- If $A = \forall x A'$, then by the induction hypothesis, we have $\begin{array}{c} A' \vee A' \\ \Vert\, {\scriptstyle \{\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ A' \end{array}$.

  Hence, we have
  $$\begin{array}{c} \dfrac{\vdash S\{\forall x A' \vee \forall x A'\}}{\vdash S\{\forall x (A' \vee A')\}}\ \mathsf{m}_2\!\downarrow \\ \vdots\ \{\mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\} \\ \vdash S\{\forall x A'\} \end{array}$$

□

Lemma 62. *The rules* $\mathsf{m}_1\!\downarrow$ *and* $\mathsf{m}_2\!\downarrow$ *are derivable for* $\{\mathsf{w}, \mathsf{c}\}$.

Proof. We have:

$$\dfrac{\dfrac{\vdash S\{\exists x A\}}{\vdash S\{\exists x (A \vee \mathsf{f})\}}\ \equiv}{\vdash S\{\exists x (A \vee B)\}}\ \mathsf{w} \qquad \text{and} \qquad \dfrac{\dfrac{\vdash S\{\exists x B\}}{\vdash S\{\exists x (\mathsf{f} \vee B)\}}\ \equiv}{\vdash S\{\exists x (A \vee B)\}}\ \mathsf{w}$$

Thus, we have:

$$\begin{array}{c} \vdash S\{\exists x A \vee \exists x B\} \\ \vdots \\ \dfrac{\vdash S\{\exists x (A \vee B) \vee \exists x (A \vee B)\}}{\vdash S\{\exists x (A \vee B)\}}\ \mathsf{c} \end{array}$$

Similar for $\mathsf{m}_2\!\downarrow$. □

Now we define a propositional encoding for first-order formulas.

Definition 63. The propositional encoding A° of a formula A is defined inductively by:

$$a^\circ = a \text{ for every atom } a$$
$$(A \vee B)^\circ = A^\circ \vee B^\circ \qquad (A \wedge B)^\circ = A^\circ \wedge B^\circ$$
$$(\forall x A)^\circ = U_x \vee A^\circ \qquad (\exists x A)^\circ = E_x \wedge A^\circ$$

where U_x and E_x are fresh nullary atoms.

Similarly, we can define the propositional encoding S° of a context S inductively by setting $\square^\circ = \square$. Note that S° is also a context.

We have the following facts:

Proposition 64. *For any context* S *and any formula* A*:*

- A° *is a formula containing no quantifier for any formula* A.
- $[\![( ]\!] A^\circ) = [\![( ]\!] A)$ *by confounding the atoms* U_x, E_x *with the variable* x*. Thus, a map* $f : [\![( ]\!] A^\circ) \to [\![( ]\!] B^\circ)$ *can be seen as a map* $f : [\![( ]\!] A) \to [\![( ]\!] B)$.
- $(S\{A\})^\circ = S^\circ\{A^\circ\}$.

Proposition 65. *Let* A *and* B *be two formulas such that* $\begin{array}{c} A \\ \Vert\, {\scriptstyle \{\mathsf{w},\mathsf{c}\}} \\ B \end{array}$. *Then* $\begin{array}{c} A^\circ \\ \Vert\, {\scriptstyle \{\mathsf{w},\mathsf{c}\}} \\ B^\circ \end{array}$.

Proof. Trivial by induction. □

Lemma 66. *Given two formulas* A *and* B *and a derivation* $\begin{array}{c} A \\ \Delta \Vert\, {\scriptstyle \{\mathsf{w},\mathsf{c}\}} \\ B \end{array}$*, then there exists a skew bifibration* $G(A) \to G(B)$.

Proof. By Lemma 61, there exists a derivation $\begin{array}{c} A \\ \Delta \Vert\, {\scriptstyle \{\mathsf{w},\mathsf{ac},\mathsf{m},\mathsf{m}_1\downarrow,\mathsf{m}_2\downarrow\}} \\ B \end{array}$.

For each rule from $\{\mathsf{w}, \mathsf{ac}, \mathsf{m}, \mathsf{m}_1\!\downarrow, \mathsf{m}_2\!\downarrow\}$, we define a map and show that it is a skew fibration.

- $\dfrac{\vdash S\{\mathsf{f}\}}{\vdash S\{A\}}\ \mathsf{w}$:

  the map wk maps f to anything and is identity elsewhere.

- $\dfrac{\vdash S\{a \vee a\}}{\vdash S\{a\}}$ ac:
  the map ac maps the two a-labelled literals in the premise to the a-labelled literal in the conclusion.
- $\dfrac{\vdash S\{(A \wedge B) \vee (C \wedge D)\}}{\vdash S\{(A \vee C) \wedge (B \vee D)\}}$ m:
  the map m is the canonical identity that maps A to A, $\cdots$, D to D.
- $\dfrac{\vdash S\{\exists x A \vee \exists x B\}}{\vdash S\{\exists x(A \vee B)\}}$ $\mathsf{m}_1\!\downarrow$:
  the map m_1 maps the two x-labelled binders in the premise to the x-labelled binder in the conclusion, A to A and B to B.
- $\dfrac{\vdash S\{\forall x A \vee \forall x B\}}{\vdash S\{\forall x(A \vee B)\}}$ $\mathsf{m}_2\!\downarrow$:
  the map m_2 maps the two x-labelled binders in the premise to the x-labelled binder in the conclusion, A to A and B to B.

By considering propositional encodings, the maps defined are label-preserving skew fibrations on the underlying fographs according to [21].

Now we prove that each map $g \in \{wk, ac, m, m_1, m_2\}$ is a skew bifibration. To do that, it suffices to prove that g is a fibration between the corresponding binding graphs since it is already a skew fibration on the corresponding fographs and it is label-preserving and existential-preserving.

for each x-binder b in $[\![(]\!]B^\circ)$, for each vertex
$v \in V([\![(]\!]A^\circ))$ such that $g(v)$ is bound by b, there exists a
unique binder b' such that b' binds v.

- wk and m are clearly fibrations: the binding relations of the premise and the conclusion are exactly the same.
- ac is a fibration: suppose that a that in the conclusion a is bound by some quantifier b in S, then for each of its preimages by ac, there exists exactly one binder (in fact, b) in S that binds it.
- m_1 and m_2 are fibrations: in the conclusion, for every atom a in $A \vee B$ bound by the x-labelled quantifier, a has exactly one preimage and it is bound by the x-llabelled quantifier in the premise.

Therefore, all of these maps are skew bifibrations and since skew bifibrations on fographs compose (Lemma 10.32, [18]), there exists a skew bifibration from $[\![(]\!]A)$ to $[\![(]\!]B)$.

□

Theorem 67. *If a formula A is provable in* LK*, then it has a combinatorial proof.*

Proof. By Theorem 53, there exists a formula A' such that there is a proof Π of A' in $\mathsf{MLL1}^{\mathsf{X}}$ and a derivation D from A' to A consisting of the w and c rules only. The proof Π corresponds to a unique unification net which is equivalent to the fonet corresponding to Π, i.e., the fograph $[\![(]\!]A')$ together with the links of Π. By Lemma 66, there exists a skew bifibration $[\![(]\!]A') \rightarrow [\![(]\!]A)$. We have thus a combinatorial proof of A.

□

D. From skew bifibrations to contraction/weakening

Theorem 68. *Let A and B be two formulas and $f : G(A) \rightarrow G(B)$ a skew bifibration. Then there exists a derivation*
$$\begin{array}{l} A \\ \Delta \Big\| \{\mathsf{w},\mathsf{c}\}. \\ B \end{array}$$

f can be seen as a skew fibration from $G(A^\circ)$ to $G(B^\circ)$, which gives the existence of the propositions A' and B', and of the following derivation:

$$\begin{array}{c} A^\circ \\ \Delta \Big\| \mathsf{m} \\ A' \\ \Delta' \Big\| \mathsf{ac} \\ B' \\ \Delta'' \Big\| \mathsf{w} \\ B^\circ \end{array}$$

Lemma 69. *there exists B'' such that $B''^\circ = B'$.*

Proof. Consider the dervation Δ''. If some U_x (or E_x) is introduced via weakening, then all the atoms it binds in B° should also be introduced via weakening. In fact, an atom of B° is introduced via weakening is equivalent to the fact that its corresponding vertex is not in the image of f. Since there is an edge from U_x (resp. E_x) to all the literals it binds in the binding graph $\overrightarrow{[\![(]\!]B)}$, if one of the atoms is in the image, U_x (resp. E_x) should also be in the image since f is a fibration on binding graphs.

This means that a such B'' can be obtained from B by erasing all the U_x and E_x introduced via weakening and all the atoms they bind. □

We introduce new (atomic) symbols E^*_x and U^*_x which are used to represent disjunctions of E_x and U_x respectiveley.

We define a translation $(\cdot)^*$ inductively by:

- $(E_x \vee \cdots \vee E_x)^* = E_x$
- $(U_x \vee \cdots \vee U_x)^* = U_x$
- structural recursion in all the other cases.

Then the derivation:

$$\begin{array}{c} A^\circ \\ \Delta \Big\| \mathsf{m} \\ A' \\ \Delta' \Big\| \mathsf{ac} \\ B''^\circ \end{array}$$

can be translated to the derivation:

$$\begin{array}{c} A^{\circ *} \\ \Delta^* \Big\| \\ B''^{\circ *} \end{array}$$

where Δ^* is the derivation obtained by replacing all the formulas F with F^* and by applying the following rule transformation:

$$\frac{S\{Q_x\}}{S\{Q_x\}}\,\mathsf{ac} \rightsquigarrow \frac{S\{Q_x\}}{S\{Q_x\}} =$$

$$\frac{S\{(E_x \wedge C) \vee (E_x \wedge D)\}}{S\{E_x \wedge (C \vee D)\}}\ \mathsf{m} \rightsquigarrow \frac{S\{(E_x \wedge C) \vee (E_x \wedge D)\}}{S\{E_x \wedge (C \vee D)\}}\ \mathsf{m}'$$

where Q_x stands for E_x or U_x.

Δ^* can now be transformed into a valid derivation Δ_1 by using the two transformation rules above and by applying them in a bottom-up style:

$$\begin{array}{c} A^{\circ *} \\ \Delta_1 \Big\| \mathsf{ac,m,m'} \\ B''^{\circ *} \end{array}$$

Lemma 70. *Every line of* Δ_1 *is a propositional encoding.*

Proof. We proceed by bottom-up induction in the derivation. Clearly, $(B''^{\circ})^*$ is a propositional encoding as there is no disjunction of Q_x in it.

First consider the ac rule: $\dfrac{C \vee C}{C}\ \mathsf{ac}$

It is clear that if C is a propositional encoding, then so is $C \vee C$.

Now consider the m rule:

$$\frac{S\{(C \wedge D) \vee (E \wedge F)\}}{S\{(C \vee E) \wedge (D \vee F)\}}\ \mathsf{m}$$

Suppose that $(C \vee E) \wedge (D \vee F) = G^\circ$ for some G. Since $C \vee E$ cannot be Q_x (otherwise, the rule applied would be m'), G can be written as $G_1 \wedge G_2$ with $C \vee E = G_1^\circ$ and $D \vee F = G_2^\circ$.

We have thus $G_i = \forall x_i H_i$ or $J_i \vee K_i (i = 1, 2)$.

If $G_i = \forall x H_i$ for some i, then there will be a conjunction of U_x and some formula which can never be eliminated by the rules m, m' and ac. However, there exists no such conjunction in $A^{\circ *}$, which leads to a contradiction.

Hence, G_i can be written as $J_i \vee K_i$ for $i = 1, 2$. We now have $(C \wedge D) \vee (E \wedge F) = ((J_1 \wedge J_2) \vee (K_1 \wedge K_2))^\circ$.

Finally, consider the m' rule:

$$\frac{S\{(E_x \wedge C) \vee (E_x \wedge D)\}}{S\{E_x \wedge (C \vee D)\}}\ \mathsf{m}'$$

Suppose that $E_X \wedge (C \vee D)$ = F° for some F. It is clear that $F = \exists x G$ with $G^\circ = C \vee D$ for some G. We distinguish two cases:

- $G = \forall y H$: in this case, $(E_x \wedge C) \vee (E_x \wedge D)$ has a subformula $(E_x \wedge U_y)$, which cannot be eliminated by the rules m, m', ac. It is clear that $A^{\circ *}$ does not have a subformula of this form, which leads to a contradiction.
- $G = G_1 \vee G_2$: in this case, $(E_x \wedge C) \vee (E_x \wedge D) = ((\exists x G_1) \vee (\exists x G_2))^\circ$.

□