

Separable Sequent Calculus for First-order Classical Logic (Work in Progress)

Dominic J. D. Hughes*

11 June 2014

This paper presents *Separable Sequent Calculus* as an extension of Gentzen's first-order classical sequent calculus with Herbrand-style structural rules (subformula contraction and weakening). Every proof can be transformed into *separated* form, in which all logical rules precede all structural rules, a result in the spirit of Gentzen's midsequent theorem or *sharpened Hauptsatz* (where all propositional rules precede all quantifier rules), and Herbrand's theorem.

Every separated proof has a direct graph-theoretic semantics as a *combinatorial proof*. Since a standard Gentzen proof is a special case of a separable proof, every Gentzen proof has a combinatorial proof semantics via its separated form. The semantics identifies sequent proofs which differ by rule transpositions (such as pushing a contraction below a weakening). Since combinatorial proofs can be verified in polynomial time (conjectured linear time), they constitute a syntax in their own right, arguably a canonical abstraction of sequent calculus modulo inessential rule transposition.

1 Introduction

Gentzen's celebrated midsequent theorem (or *sharpened Hauptsatz*) for first-order classical sequent calculus [Gen35] states that every proof of a sequent of prenex formulae can be transformed to the form below-left, in which propositional rules are separated above quantifier rules:¹

Axioms	Axioms
Propositional & structural rules	Logical rules
Propositional midsequent	Multiplicative midsequent
Quantifier & structural rules	Structural rules
Concluding prenex sequent	Concluding sequent

Gentzen [Gen35] This paper

This paper presents a separation in the same spirit, in which all logical rules precede all structural rules (see above-right). To achieve this separation we extend sequent calculus with Herbrand-style structural rules: subformula contraction [Her30]² (which Herbrand called *generalized simplification*) and weakening. We refer to the calculus as *Separable Sequent Calculus*.

*This research is being conducted at Stanford University as a Visiting Scholar in the Mathematics Department. I am very grateful to my host, Sol Feferman. The work began whilst a Visiting Scholar in the Computer Science Department, hosted by Vaughan Pratt, to whom I am also very grateful.

¹Herbrand's theorem for prenex formulae follows from the midsequent theorem (see e.g. [TS96]).

²See [Bus94, §2] for a summary.

An illustrative example is below. Consider the following proof of in first-order classical sequent calculus with multiplicative $\wedge$ and $\vee$ (i.e., as in multiplicative linear logic [Gir87]), where $\bar{P}c$ is the negation of Pc :³

$$\begin{array}{c}
\frac{\overline{\vdash \bar{P}a, Pa}}{\vdash \bar{P}a, (\bar{P}a \wedge \bar{P}b), Pa} W \quad \frac{\overline{\vdash \bar{P}b, Pb}}{\vdash \bar{P}b, (\bar{P}a \wedge \bar{P}b), Pb} W \\
\frac{\vdash \bar{P}a, (\bar{P}a \wedge \bar{P}b), Pa}{\vdash \bar{P}a, (\bar{P}a \wedge \bar{P}b) \vee Pa} \vee \quad \frac{\vdash \bar{P}b, (\bar{P}a \wedge \bar{P}b), Pb}{\vdash \bar{P}b, (\bar{P}a \wedge \bar{P}b) \vee Pb} \vee \\
\frac{\vdash \bar{P}a, (\bar{P}a \wedge \bar{P}b) \vee Pa}{\vdash \bar{P}a, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} \exists \quad \frac{\vdash \bar{P}b, (\bar{P}a \wedge \bar{P}b) \vee Pb}{\vdash \bar{P}b, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} \exists \\
\frac{\vdash \bar{P}a, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px) \quad \vdash \bar{P}b, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash \bar{P}a \wedge \bar{P}b, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} \wedge \\
\frac{\vdash \bar{P}a \wedge \bar{P}b, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash (\bar{P}a \wedge \bar{P}b), Px, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} C \\
\frac{\vdash (\bar{P}a \wedge \bar{P}b), Px, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash (\bar{P}a \wedge \bar{P}b) \vee Px, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} \vee \\
\frac{\vdash (\bar{P}a \wedge \bar{P}b) \vee Px, \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} \exists \\
\frac{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} C
\end{array}$$

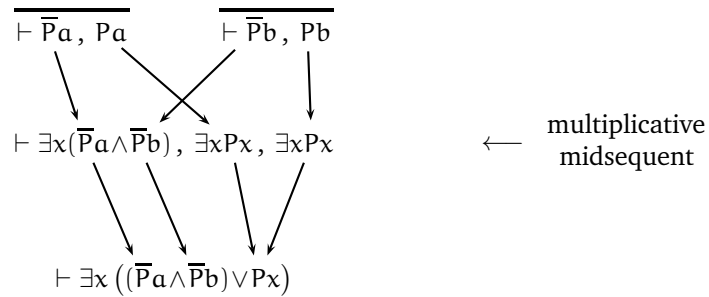
A *separated form* of the above example, with all logical rules ($\wedge \vee \exists$) preceding all structural rules (C W), is below-left:

$$\begin{array}{c}
\frac{\overline{\vdash \bar{P}a, Pa}}{\vdash \bar{P}a, \exists x Px} \exists \quad \frac{\overline{\vdash \bar{P}b, Pb}}{\vdash \bar{P}b, \exists x Px} \exists \\
\frac{\vdash \bar{P}a, \exists x Px \quad \vdash \bar{P}b, \exists x Px}{\vdash \bar{P}a \wedge \bar{P}b, \exists x Px, \exists x Px} \wedge \\
\frac{\vdash \bar{P}a \wedge \bar{P}b, \exists x Px, \exists x Px}{\vdash \exists x (\bar{P}a \wedge \bar{P}b), \exists x Px, \exists x Px} \exists \\
\frac{\vdash \exists x (\bar{P}a \wedge \bar{P}b), \exists x Px, \exists x Px}{\vdash \exists x (\bar{P}a \wedge \bar{P}b), \exists x Px} C \\
\frac{\vdash \exists x (\bar{P}a \wedge \bar{P}b), \exists x Px}{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x Px} W \\
\frac{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x Px}{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} W \\
\frac{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px), \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)}{\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)} C
\end{array}$$

Axioms
Logical rules
Multiplicative midsequent
Structural rules
Concluding sequent

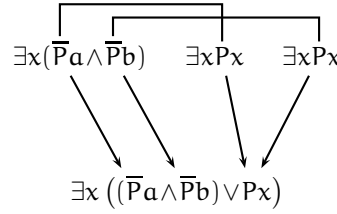
Sequent shades match the separated structure above-right (copied from page 1). The weakenings W are Herbrand-style structural rules, subformula rewrites $\bar{P}a \wedge \bar{P}b \rightarrow (\bar{P}a \wedge \bar{P}b) \vee Px$ and $Px \rightarrow (\bar{P}a \wedge \bar{P}b) \vee Px$. Since the midsequent between the logical rules and structural rules is a theorem of first-order multiplicative linear logic, we call it the *multiplicative midsequent*.

Every separated proof has a canonical semantics as a *combinatorial proof*, introduced in the propositional case in [Hug06a, Hug06b]. The example above becomes a combinatorial proof in two steps. First, abstract both the upper logical half and the lower structural half to a *leaf function*, where a *leaf* is a predicate symbol occurrence:

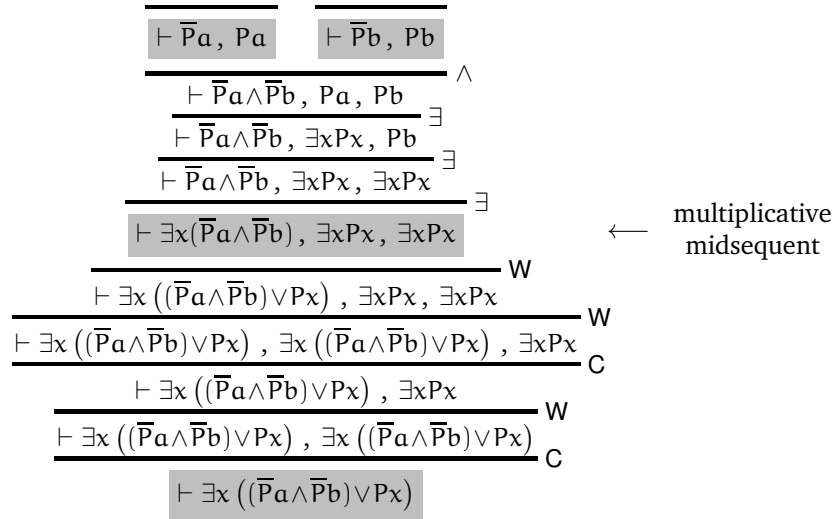


³We work with a right-sided sequent calculus in which a sequent is a multiset of formulas, obviating the need for an exchange rule. Formulas are in negation normal form.

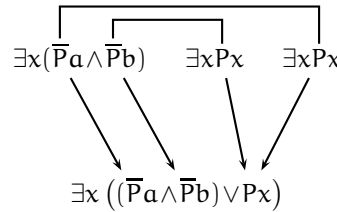
(The arrows simply result from tracing the leaves down the proof in the obvious way.) Second, abstract the upper half still further, replacing each axiom by an edge on the multiplicative midsequent (and remove turnstiles and commas):



This is the *unique* (cut-free) combinatorial proof of $\vdash \exists x ((\bar{P}a \wedge \bar{P}b) \vee Px)$. *Every* (cut-free) separated proof translates to it, exemplifying the sense in which combinatorial proofs provide a canonical representation of sequent calculus proofs. For example, the following separated proof, with a different sequence of contractions and weakenings and a transposition of the $\wedge$ rule up over the top two $\exists$ rules, translates to the same combinatorial proof:



When drawing a combinatorial proof, the particular layout of the multiplicative midsequent formulas on the page is immaterial. For example the following diagram depicts the same combinatorial proof:



A combinatorial proof of a sequent $\vdash \Gamma$ comprises:

- a multiplicative midsequent M ;⁴
- edges between leaves in M ;
- a leaf function $M \rightarrow \Gamma$.

These data must satisfy certain combinatorial conditions, which can be checked in polynomial time.

CONJECTURE 1 (LINEAR TIME COMPLEXITY)

The correctness of a combinatorial proof can be checked in linear time.

⁴We formalize the multiplicative midsequent as a parse tree with ordered $\wedge$ nodes and unordered $\vee$ nodes of unbounded arity. This provides a canonical representation modulo associativity of $\vee$ and $\wedge$ and commutativity of $\vee$, identifying comma and $\vee$. The ordering on $\wedge$ leads to the conjecture that a combinatorial proof can be verified in linear time.

The main theorems are:

THEOREM 1 (SEPARATION)

Every proof of separable sequent calculus can be transformed into separated form by transposing rules.

THEOREM 2 (SURJECTIVITY)

The translation from separated sequent calculus proofs to combinatorial proofs is a surjection.

Since a combinatorial proof can be verified in polynomial time (conjectured linear time), combinatorial proofs constitute a well-defined syntax in their own right. In combination with the surjectivity theorem, this suggests that combinatorial proofs provide an abstract representation of first-order classical sequent calculus modulo inessential local rule commutations.

Related Work. Herbrand-style subformula rewrites are common in categorical logic (where in addition coherence rules provide equalities on proofs). Propositional systems of subformula rewrites were used in [DHPP99] and [Hug06b]. Subformula rewrite rules are sometimes called *deep* rules [Brü04].

Propositional combinatorial proofs were introduced in [Hug06a] and studied as invariants of syntactic proofs in [Hug06b]. Combinatorial proofs build on a long history of work in the area of graphical invariants of proofs. The idea of linking dual occurrences (as on the multiplicative midsequent of a combinatorial proof) is widespread in the study of various forms of syntax, going back many decades in many disciplines, including category theory (compatibility graphs) [EK66, EML42], automated theorem proving (matings and connections) [Dav71, Bib74, And76, And81] and linear logic (axiom links) [Gir87]. A key difference with the mating and connection methods is that their correctness is exponential time. Buss analyses the flow of atoms down from axioms in a proof [Bus89], without a correctness criterion, in the propositional case; a translation from logical flow graphs to propositional combinatorial proofs is studied in [Car10]. The quantifier jumps in the correctness criterion are based on those for proof nets [Gir95]. Propositional proof nets with weakening are presented in [Bel03]. More recently, Heijltjes and McKinley have studied graphical representations of first-order classical proofs [Hei10, McK10]. The expansion proof nets in [McK13] are a restriction of the class of propositional combinatorial proofs [Hug06a, Hug06b] lying in the image of standard sequent calculus (*i.e.*, without subformula rewrite rules).

2 Separable Sequent Calculus

Cut-free first-order multiplicative linear logic [Gir87], extended with rules for contraction and weakening, constitutes a one-sided sequent calculus for first-order classical logic:

$$\begin{array}{c}
\frac{}{\vdash \overline{P} \vec{t}, P \vec{t}} \quad \frac{}{\vdash \top} \quad \text{(no rule for } \perp \text{)} \\
\\
\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \wedge B, \Delta} \wedge \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B} \vee \\
\\
\frac{\vdash \Gamma, A(t)}{\vdash \Gamma, \exists x A(x)} \exists \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x A} \forall \quad (x \text{ not free in } \Gamma) \\
\\
\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} C \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, A, B} W
\end{array}$$

Here $\vec{t}$ denotes any sequence of terms, P is any predicate symbol, $\overline{P}$ is the formal dual of P , A and B are formulas, and Γ and Δ are sequents. We take a sequent to be a multiset (disjoint union) of

formulas, obviating the need for the exchange rule on sequents (cf. [TS96]). **Separable sequent calculus** is the extension of the above system with the following Herbrand-style [Her30] subformula rewrite rules:

$$\begin{array}{c}
\dfrac{\dots \dfrac{A \vee A}{A} \dots C}{\dots \dfrac{A_i}{A_1 \vee A_2} \dots W} \\
\dfrac{\dots \dfrac{A \vee B}{B \vee A} \dots E}{\dots \dfrac{A \vee (B \vee C)}{(A \vee B) \vee C} \dots A} \quad \dfrac{\dots \dfrac{(A \wedge B) \wedge C}{A \wedge (B \wedge C)} \dots A}
\end{array}$$

C is *contraction* (which Herbrand calls *generalized simplification* [Her30]) W is *weakening*, E is *exchange* and A is *associativity*. The ellipsis notation indicates that the rule applies to any subformula. Examples of the application of subformula weakening W can be found in the Introduction. It is possible to add a cut rule (not explored in this paper).

A **separated proof** is a proof of separable sequent calculus in which all logic rules precede all structural rules.

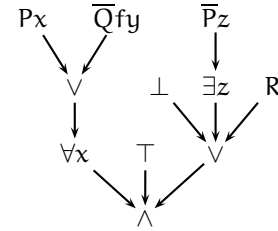
THEOREM 3 (SEPARATION)

Every proof of separable sequent calculus can be transformed into separated form by transposing rules.

For example, the second proof in the Introduction is the separated form of the first proof in the Introduction.

3 Combinatorial proofs

Henceforth we work with formulas formalized as labelled trees, such as the one shown right. Formulas are canonical modulo commutativity of $\vee$ and associativity of $\wedge$ and $\vee$: the children (inbound edges) of $\wedge$ nodes are ordered⁵ (but those of $\vee$ nodes are not), and every $\vee$ (resp. $\wedge$) node has at least two children, none of which is a $\vee$ (resp. $\wedge$). We identify formulas modulo renaming of bound variables (α -conversion).



We employ the following conventions:

$x \ y \ z$	variables	$s \ t \ u$	terms
$f \ g \ h$	n -ary function symbols ($n \geq 1$)	$P \ Q \ R$	predicate symbols
$a \ b \ c$	0-ary function symbols	$A \ B \ C$	formulas

Fix an arity-preserving **negation** function ($\bar{}$) on predicate symbols such that $\bar{\bar{P}} = P$ for all P . A **literal** is an expression $Pt_1 \dots t_n$ for any n -ary predicate symbol P and terms t_i . Parentheses and commas may be inserted to increase readability, e.g., $Paffy = P(a, ffy) = P(a, f(f(y)))$. An **atom** is a literal or either of the logical constants **true** $\top$ or **false** $\perp$.

A **label** is any of: (a) an atom, (b) the connectives **and** $\wedge$ and **or** $\vee$, (c) for any variable x , a quantifier $\forall x$ or $\exists x$. A **formula** is a finite directed tree labelled with atoms on leaves and connectives/quantifiers on internal nodes, with edges directed towards the root, such that (1) the inbound edges of $\wedge$ nodes are strictly linearly ordered⁶, and (2) every $\vee$ (resp. $\wedge$) node has at least two inbound edges, none from a $\vee$ (resp. $\wedge$) node, (3) every quantifier node has just one inbound

⁵The conjecture that a combinatorial proof can be verified in linear time is a direct result of ordered $\wedge$ nodes. One can define combinatorial proofs modulo commutativity of $\wedge$, with unordered $\wedge$ nodes (as in the original propositional version [Hug06a]), but correctness is then apparently not linear (merely polynomial).

⁶ $\vee$ nodes remain unordered.

edge. Variable binding is defined in the standard way (viewing our formulas as parse trees of more standard syntactic formulas). We identify formulas modulo renaming of bound variables.

Define the **negation** $\bar{\alpha}$ of a label α by $\overline{pt_1 \dots t_n} = \bar{p}t_1 \dots t_n$, $\bar{\top} = \perp$, $\bar{\perp} = \top$, $\bar{\wedge} = \vee$, $\bar{\vee} = \wedge$, $\bar{\exists x} = \forall x$ and $\bar{\forall x} = \exists x$, and define the negation $\bar{A}$ of a formula A by negating its labels.

To write formulas conveniently inline, we identify an atom α with a single-node formula labelled α , interpret the symbols $\wedge$, $\vee$, $\forall x$ and $\exists x$ as operations on formulas in the obvious way, and define the abbreviation $A \Rightarrow B = \bar{A} \vee B$. For example, the following formula expressions each denote the formula whose tree was depicted near the beginning of this section:

$$(\forall x(Px \vee \bar{Q}fy)) \wedge \top \wedge (\perp \vee (\exists z\bar{P}z) \vee R)$$

$$(\forall x(Qfy \Rightarrow Px)) \wedge \top \wedge ((\exists z\bar{P}z) \vee R \vee \perp)$$

To reduce parentheses in such formula expressions, we assume quantifiers minimize scope, e.g., $\forall xPx \vee (\exists y\bar{P}y \wedge Q) = (\forall xPx) \vee ((\exists y\bar{P}y) \wedge Q)$.

The **meet** of two vertices is the vertex at which their paths to the root meet. Vertices v and w are **conjunctively related**, denoted $v \wedge w$, if their meet is a $\wedge$, and v is to the left of w (under hereditary child-orderings at $\wedge$ nodes). To emphasize the ambient formula B , we write $\wedge_B$. Vertices are **disjunctively related** if their meet is a $\vee$.

The **line** of a leaf v is the sequence of quantifiers along the path from the root to v , followed by the atom of v . For example, in $\exists x(\bar{P}x \vee \forall yPy)$ the line of Py is $\exists x\forall yPy$ and that of $\bar{P}x$ is $\exists x\bar{P}x$. A quantifier is **vacuous** if it binds no variable, e.g., $\exists x$ in $\exists x\forall y(Py \vee \perp)$.

A formula is **plain** if (a) no quantified variable occurs free, and (b) two quantifiers have the same variable only if they are disjunctively related. For example, $\forall xPx \vee \forall yQyz$, $\forall xPx \wedge \forall yQyz$ and $\forall xPx \vee \forall xQxz$ are plain, but $Px \vee \forall xQx$, $\forall xPx \wedge \forall xQxz$, $\forall x\forall xRx$ and $\forall x\forall x\top$ are not. For technical convenience, *henceforth we assume all formulas are plain* (by renaming bound variables if necessary). A formula is **clean** if all quantifiers have distinct variables, e.g. $\exists xPx \vee \exists yQy$ but not $\exists xPx \vee \exists xQx$.

A **morphism** $\theta : A \rightarrow B$ from a formula A to a clean formula B is a function θ from the leaves of A to the leaves of B such that:

(M1) θ preserves lines: for each leaf v in A the line of $\theta(v)$ equals the line of v .⁷

(M2) θ preserves conjunctive relationships: if $v \wedge_A w$ then $\theta(v) \wedge_B \theta(w)$.

(M3) θ is a **skew fibration**: if $\theta(v) \wedge_B w$ there exists w' with $v \wedge_A w'$ and $w \not\wedge_B \theta(w')$.⁸

Viewing $\wedge_A$ and $\wedge_B$ as simple directed graphs, (M2) requires θ to be a directed graph homomorphism, and (M3) requires θ to be a lax variant of a graph fibration; see [Hug06a, Hug06b].⁹ Condition (M1) in particular implies that atomic labels are preserved (i.e., the atomic label of $\theta(v)$ equals the atomic label of v).

THEOREM 4 (CONTRACTION-WEAKENING) *A clean formula B can be derived from a formula A by the structural rewrite rules of separable sequent calculus if and only if there is a morphism $A \rightarrow B$.*

Two literals are **dual** if their predicate symbols are dual (e.g. Pxy and $\bar{P}yfa$) and two leaves are dual if they have dual literals. A **link** is a pair $\{v, w\}$ of dual leaves. A **linking** on a formula is a set of disjoint links whose union contains every literal-labelled leaf. We draw a link $\{v, w\}$ as an edge between v and w , e.g.

$$\overline{\forall y \exists x \bar{P}(gy, fx, a)} \quad \exists x' \forall y' P(x', fy', a) \wedge Qhza \quad \exists z' \bar{Q}z' \wedge \top$$

⁷By relaxing this condition to allow the line of v to be any result of deleting vacuous quantifiers from the line of $\theta(v)$ (e.g. $\forall x\exists y Pfx$ yields $\forall x Pfx$ by deleting the vacuous quantifier $\exists y$), we enlarge the set of combinatorial proofs while remaining sound and complete, and interpret a subformula prenex rule (rewrite $A \diamond \nabla x B$ to $\nabla x(A \diamond B)$ for $\diamond \in \{\wedge, \vee\}$ and $\nabla \in \{\forall, \exists\}$, with x not free in A) and a rule adding vacuous quantifiers (rewrite A to $\nabla x A$, for x not free in A).

⁸Equivalently, θ preserves maximal (directed) cliques, or, since $\wedge$ is a strict order, θ preserves maximal chains.

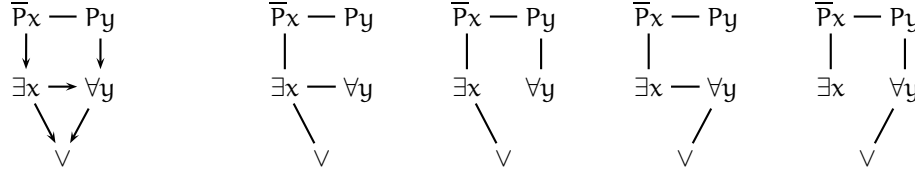
⁹The only difference here is that we take the edge relation $\wedge$ to be directed.

Let λ be a linking on a clean formula B . A **unifier** for λ is an assignment of terms to existentially quantified variables which equalizes the term sequences at either end of every link. For example, $U = \{x \mapsto y', x' \mapsto gy, z' \mapsto zha\}$ is a unifier for the linking above: upon substituting by U the first link has (gy, fy', a) at either end, and the second, zha . A linking is **unifiable** if it has a unifier.¹⁰

Let λ be a unifiable linking on a clean formula B . A **dependency** $x \curvearrowright y$ is an existentially quantified variable x and a universally quantified variable y such that every unifier of λ assigns to x a term containing y .¹¹ The dependencies of the linking drawn above are $x \curvearrowright y'$ and $x' \curvearrowright y$. The **graph** of λ is the directed tree B together with

- for every link $\{v, w\}$ in λ , an undirected edge $v - w$;
- for every dependency $x \curvearrowright y$, a directed edge $\exists x \rightarrow \forall y$.

For example, the graph of $\exists x \overline{P}x \vee \forall y Py$, whose unique unifier is $\{x \mapsto y\}$, is below-left.



A **switching** (cf. [DR89]) of λ is any derivative of its graph obtained by deleting all but one edge into each $\vee$ and $\forall$ vertex and undirecting all remaining edges. The four switchings of $\exists x \overline{P}x \vee \forall y Py$ are above-right. A linking is **correct** if all switchings are trees (acyclic and connected). The linking $\exists x \overline{P}x \vee \forall y Py$ is correct.

A **combinatorial proof** of a formula B is a formula A (the **multiplicative midsequent**), a correct linking on A , and a morphism $A \rightarrow B$.

CONJECTURE 2 *The correctness of a combinatorial proof can be verified in linear time.*

The graph can be constructed in linear time since each new edge is per axiom link or dependency. The graph determines a contractibility graph [Dan90] with $\vee$ s and $\forall$ s as switched nodes, and leaves, $\wedge$ s and $\exists$ s as unswitched nodes, checkable in linear time [Gue99]. Thus the crux of the conjecture is the fact that the correctness of a morphism appears to be checkable in linear time. For that, the ordering of $\wedge$ nodes seems crucial.

THEOREM 5 *There is a surjection from separated sequent calculus proofs to combinatorial proofs.*

References

- [And76] Peter B. Andrews. Refutations by matings. *IEEE Transactions on Computers*, 25(8), 1976.
- [And81] Peter B. Andrews. Theorem proving via general matings. *J. ACM*, 28(2), 1981.
- [Bel03] Gianluigi Bellin. Two paradigms of logical computation in affine logic? de Queiroz, Ruy J. G. B. (ed.), *Logic for concurrency and synchronisation*. Trends Log. Stud. Log. Libr. 18, 2003.
- [Bib74] W. Bibel. An approach to a systematic theorem proving procedure in first-order logic. *Computing*, 12(1), 1974.
- [Brü04] Kai Brunnler. *Deep Inference and Symmetry in Classical Proofs*. PhD thesis, Technischen Universität Dresden, 2004.
- [Bus89] Samuel R. Buss. The undecidability of k -provability. *Annals of Pure and Applied Logic*, 53, 1989.

¹⁰Unifiability can be determined in linear time and space.

¹¹All dependencies can be determined in linear time, via a *most general unifier*.

- [Bus94] S. R. Buss. On Herbrand's theorem. In *Logic and Computational Complexity*, volume 960 of *Lec. Notes in Comp. Sci.*, pages 195–209. Springer-Verlag, 1994.
- [Car10] Alessandra Carbone. A new mapping between combinatorial proofs and sequent calculus proofs read out from logical flow graphs. *Inf. Comput.*, 208(5):500–509, 2010.
- [Dan90] V. Danos. *La logique linéaire appliquée à l'étude de divers processus de normalisation et principalement du lambda calcul*. PhD thesis, Univ. de Paris, 1990.
- [Dav71] G.V. Davydov. The synthesis of the resolution method with the inverse method. *Zapiski Nauchnykh Seminarov Lomi* 20 1971 24–35., 20:24–35, 1971. Translation in *J. Sov. Math.* 1(1) 1973 12–18.
- [DHPP99] H. Devarajan, D. J. D. Hughes, G. D. Plotkin, and V. R. Pratt. Full completeness of the multiplicative linear logic of Chu spaces. In *Proc. LICS'99*, pages 234–245. IEEE, 1999.
- [DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.
- [EK66] S. Eilenberg and G. M. Kelly. A generalization of the functorial calculus. *J. Algebra*, 3:366–375, 1966.
- [EML42] S. Eilenberg and S. Mac Lane. Group extensions and homology. *Ann. Math.*, 43:757–831, 1942.
- [Gen35] G. Gentzen. Untersuchungen über das logische Schließen I, II. *Mathematische Zeitschrift*, 39:176–210 405–431, 1935. Translation in [Gen69, 68–131].
- [Gen69] G. Gentzen. *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969. Edited and introduced by M.E. Szabo.
- [Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [Gir95] J.-Y. Girard. Geometry of interaction III: Accommodating the additives. In *Advances in Linear Logic*, volume 222 of *London Math. Soc. LNS*, pages 329–389. Cambridge University Press, 1995.
- [Gue99] S. Guerrini. Correctness of multiplicative proof nets is linear. In *Proc. Logic in Computer Science '99*, 1999.
- [Hei10] Willem Heijltjes. Classical proof forestry. *Annals of Pure and Applied Logic*, 161(11):1346–1366, 2010.
- [Her30] Jacques Herbrand. *Recherches sur la théorie de la démonstration*. PhD thesis, Sorbonne, Paris, 1930.
- [Hug06a] D. J. D. Hughes. Proofs Without Syntax. *Annals of Mathematics*, 143:1065–1076, 2006.
- [Hug06b] D. J. D. Hughes. Towards Hilbert's 24th Problem: Combinatorial Proof Invariants. In *Proc. WOLLiC'06*, volume 165 of *Lec. Notes in Comp. Sci.*, 2006.
- [McK10] Richard McKinley. Expansion nets: Proof nets for for propositional classical logic. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 17)*, volume 6397 of *Lecture Notes in Computer Science*, pages 535–549. Springer, 2010.
- [McK13] Richard McKinley. Canonical proof nets for classical logic. *Annals of Pure and Applied Logic*, 164(6):702–732, 2013.
- [TS96] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, U.K., 1996.