Annals of Mathematics, **164** (2006), 1065–1076

Proofs without syntax

By Dominic J. D. Hughes

[M]athematicians care no more for logic
than logicians for mathematics.

Augustus de Morgan, 1868

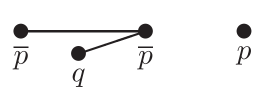
Abstract

Proofs are traditionally syntactic, inductively generated objects. This paper presents an abstract mathematical formulation of propositional calculus (propositional logic) in which proofs are combinatorial (graph-theoretic), rather than syntactic. It defines a *combinatorial proof* of a proposition ϕ as a graph homomorphism $h : C \to G(\phi)$, where $G(\phi)$ is a graph associated with ϕ and C is a coloured graph. The main theorem is soundness and completeness: ϕ is true if and only if there exists a combinatorial proof $h : C \to G(\phi)$.

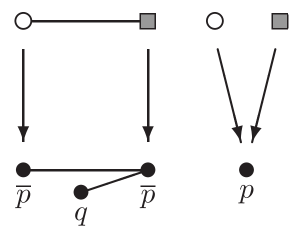
1. Introduction

In 1868, de Morgan lamented the rift between mathematics and logic [deM68]: "*[M]athematicians care no more for logic than logicians for mathematics.*" The dry syntactic manipulations of formal logic can be off-putting to mathematicians accustomed to beautiful symmetries, geometries, and rich layers of structure. Figure 1 (see Section 2) shows a syntactic proof in a standard Hilbert system taught to mathematics undergraduates [Hil28], [Joh87]. Although the system itself is elegant (just three axiom schemata suffice), the syntactic proofs generated in it need not be. Other syntactic systems include [Fr1879], [Gen35].

This paper presents an abstract mathematical formulation of propositional calculus (propositional logic) in which proofs are combinatorial (graph-theoretic), rather than syntactic. It defines a *combinatorial proof* of a proposition ϕ as a graph homomorphism $h : C \to G(\phi)$, where $G(\phi)$ is a graph associated with ϕ and C is a coloured graph. For example, if $\phi = ((p \Rightarrow q) \Rightarrow p) \Rightarrow p$ then $G(\phi)$ is:



A combinatorial proof $h : C \to G(\phi)$ of ϕ is shown below:



The upper graph C has two colours (white ○ and grey ■), and the arrows define h. The same proposition is proved syntactically in Figure 1.

The main theorem of the paper is soundness and completeness:

A proposition is true if and only if it has a combinatorial proof.

As with conventional syntactic soundness and completeness, this theorem matches a universal quantification with an existential one: a proposition ϕ is true if it evaluates to 1 *for all* 0/1 assignments of its variables, and ϕ is provable if *there exists* a proof of ϕ. However, where conventional completeness provides an inductively generated *syntactic* witness (e.g. Figure 1), this theorem provides an abstract *mathematical* witness for every true proposition (e.g. the homomorphism h drawn above).

Just three conditions suffice for soundness and completeness: a graph homomorphism $h : C \to G(\phi)$ is a combinatorial proof of ϕ if (1) C is a suitable coloured graph, (2) the image of each colour class is labelled appropriately, and (3) h is a *skew fibration*, a lax form of graph fibration. Each condition can be checked in polynomial time, so combinatorial proofs constitute a formal *proof system* [CR79].

Acknowledgements. Nil Demirçubuk, Vaughan Pratt, Julien Basch, Rajat Bhattacharjee, Rob van Glabbeek, Sol Feferman, Grisha Mints and Don Knuth. Stanford grant 1DMA644.

2. Notation and terminology

Graphs. An ***edge*** on a set V is a two-element subset of V. A ***graph*** (V, E) is a finite set V of ***vertices*** and a set E of edges on V. Write $V(G)$ and $E(G)$ for the vertex set and edge set of a graph G, respectively, and vw for $\{v, w\}$. The ***complement*** of (V, E) is the graph (V, E^{c}) with $vw \in E^{\mathsf{c}}$ if and only if $vw \notin E$. A graph (V, E) is ***coloured*** if V carries an equivalence relation $\sim$ such that $v \sim w$ only if $vw \notin E$; each equivalence class is a ***colour class***. Given a set L, a graph is L-***labelled*** if every vertex has an element of L associated with it, its ***label***. Let $G = (V, E)$ and $G' = (V', E')$ be graphs. A ***homomorphism*** $h : G \to G'$ is a function $h : V \to V'$ such that $vw \in E$ implies $h(v)h(w) \in E'$.

Below is a proof of Peirce's law $((p\Rightarrow q)\Rightarrow p)\Rightarrow p$ in a standard Hilbert formulation of propositional logic, taught to mathematics undergraduates [Joh87], with axiom schemata

$$\begin{array}{ll} (a) & x\Rightarrow(y\Rightarrow x) \\ (b) & (x\Rightarrow(y\Rightarrow z))\Rightarrow((x\Rightarrow y)\Rightarrow(x\Rightarrow z)) \\ (c) & ((x\Rightarrow\bot)\Rightarrow\bot)\Rightarrow x \end{array}$$

and where (m^i_j) marks *modus ponens* with hypotheses numbered i and j. Hilbert systems tend to emphasise the elegance of the schemata (just (a)–(c) suffice) over the elegance of the proofs generated by the schemata. (Note: there may exist a shorter proof of Peirce's law in this system.)

1 (c) $((q\Rightarrow\bot)\Rightarrow\bot)\Rightarrow q$

2 (a) $(((q\Rightarrow\bot)\Rightarrow\bot)\Rightarrow q)\Rightarrow(\bot\Rightarrow(((q\Rightarrow\bot)\Rightarrow\bot)\Rightarrow q))$

3 (m^1_2) $\bot\Rightarrow(((q\Rightarrow\bot)\Rightarrow\bot)\Rightarrow q)$

4 (b) $(\bot\Rightarrow(((q\Rightarrow\bot)\Rightarrow\bot)\Rightarrow q))\Rightarrow((\bot\Rightarrow((q\Rightarrow\bot)\Rightarrow\bot))\Rightarrow(\bot\Rightarrow q))$

5 (m^3_4) $(\bot\Rightarrow((q\Rightarrow\bot)\Rightarrow\bot))\Rightarrow(\bot\Rightarrow q)$

6 (a) $\bot\Rightarrow((q\Rightarrow\bot)\Rightarrow\bot)$

7 (m^6_5) $\bot\Rightarrow q$

8 (a) $(\bot\Rightarrow q)\Rightarrow(p\Rightarrow(\bot\Rightarrow q))$

9 (m^7_8) $p\Rightarrow(\bot\Rightarrow q)$

10 (b) $(p\Rightarrow(\bot\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))$

11 (m^9_{10}) $(p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)$

12 (a) $((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))$

13 (b) $((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p))$

14 (a) $(((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))\Rightarrow$
$(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))$

15 (m^{13}_{14}) $((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))$

16 (b) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))\Rightarrow$
$((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))$

17 (m^{15}_{16}) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow((p\Rightarrow q)\Rightarrow p)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))$

18 (m^{12}_{17}) $((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p))$

19 (b) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))\Rightarrow$
$((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))$

20 (m^{18}_{19}) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))$

21 (a) $((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))$

22 (a) $((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))\Rightarrow$
$(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))$

23 (m^{20}_{22}) $((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))$

24 (b) $(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q)))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))\Rightarrow((((p\Rightarrow\bot)\Rightarrow$
$(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))))$

25 (m^{23}_{24}) $(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))))\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p)))$

26 (m^{21}_{25}) $((p\Rightarrow\bot)\Rightarrow(p\Rightarrow q))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))$

27 (m^{11}_{26}) $((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p)$

28 (a) $(p\Rightarrow\bot)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot))\Rightarrow(p\Rightarrow\bot))$

29 (b) $((p\Rightarrow\bot)\Rightarrow(((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot))\Rightarrow(p\Rightarrow\bot)))\Rightarrow(((p\Rightarrow\bot)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot)))\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot)))$

30 (m^{28}_{29}) $((p\Rightarrow\bot)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot)))\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot))$

31 (a) $(p\Rightarrow\bot)\Rightarrow((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot))$

32 (m^{31}_{30}) $(p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot)$

33 (b) $((p\Rightarrow\bot)\Rightarrow(p\Rightarrow\bot))\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow\bot))$

34 (m^{32}_{33}) $((p\Rightarrow\bot)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow\bot)$

35 (c) $((p\Rightarrow\bot)\Rightarrow\bot)\Rightarrow p$

36 (a) $(((p\Rightarrow\bot)\Rightarrow\bot)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow\bot)\Rightarrow p))$

37 (m^{35}_{36}) $((p\Rightarrow\bot)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow\bot)\Rightarrow p)$

38 (b) $(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow\bot)\Rightarrow p))\Rightarrow((((p\Rightarrow\bot)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow\bot))\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p))$

39 (m^{37}_{38}) $(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow\bot))\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p)$

40 (m^{34}_{39}) $((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p$

41 (a) $(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p)\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p))$

42 (m^{40}_{41}) $((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p)$

43 (b) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow(((p\Rightarrow\bot)\Rightarrow p)\Rightarrow p))\Rightarrow((((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow p))$

44 (m^{42}_{43}) $(((p\Rightarrow q)\Rightarrow p)\Rightarrow((p\Rightarrow\bot)\Rightarrow p))\Rightarrow(((p\Rightarrow q)\Rightarrow p)\Rightarrow p)$

45 (m^{27}_{44}) $((p\Rightarrow q)\Rightarrow p)\Rightarrow p$

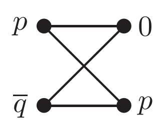
Figure 1: A syntactic proof in a standard Hilbert system

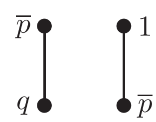
If V and V' are disjoint, the ***union*** $G \vee G'$ is $(V \cup V', E \cup E')$ and the ***join*** $G \wedge G'$ is $(V \cup V',\ E \cup E' \cup \{vv' : v \in V,\ v' \in V'\})$; colourings or labellings of G and G' are inherited. A graph (V, E) is a ***cograph*** [CLS81] if V is non-empty and for any distinct $v, w, x, y \in V$, the restriction of E to edges on $\{v, w, x, y\}$ is not $\{vw, wx, xy\}$. A set $W \subseteq V$ ***induces a matching*** if it is non-empty and for all $w \in W$ there is a unique $w' \in W$ such that $ww' \in E$.

Propositions. Fix a set $\mathcal{V}$ of ***variables***. A ***proposition*** is any expression generated freely from variables by the binary operations ***and*** $\wedge$, ***or*** $\vee$, and ***implies*** $\Rightarrow$, the unary operation ***not*** $\neg$, and the constants (nullary operations) ***true*** 1 and ***false*** 0. A ***valuation*** is a function $f : \mathcal{V} \to \{0, 1\}$. Write $\hat{f}$ for the extension of a valuation f to propositions defined by $\hat{f}(0) = 0$, $\hat{f}(1) = 1$, $\hat{f}(\neg\phi) = 1 - \hat{f}(\phi)$, $\hat{f}(\phi \wedge \rho) = \min\{\hat{f}(\phi), \hat{f}(\rho)\}$, $\hat{f}(\phi \vee \rho) = \max\{\hat{f}(\phi), \hat{f}(\rho)\}$, and $\hat{f}(\phi \Rightarrow \rho) = \hat{f}((\neg\phi) \vee \rho)$. A proposition ϕ is ***true*** if $\hat{f}(\phi) = 1$ for all valuations f. Variables $p \in \mathcal{V}$ and their negations $\overline{p} = \neg p$ are ***literals***; p and $\overline{p}$ are ***dual***, as are 0 and 1. An ***atom*** is a literal or constant, and $\mathcal{A}$ denotes the set of atoms.

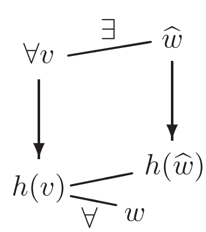
3. Combinatorial proofs

Given an $\mathcal{A}$-labelled graph G, define $\neg G$ as the result of complementing G and every label of G. For example, if G is the graph below-left, then $\neg G$ is the graph below-right.





Define $G \Rightarrow G' = (\neg G) \vee G'$. Identify each atom a with a single vertex labelled a; thus, having defined operations $\neg$, $\vee$, $\wedge$ and $\Rightarrow$ on $\mathcal{A}$-labelled graphs, every proposition ϕ determines an $\mathcal{A}$-labelled graph, denoted $G(\phi)$. For example, $G\big((p \vee \neg q) \wedge (0 \vee p)\big)$ is above-left, $G\big((q \wedge \neg p) \vee (1 \wedge \neg p)\big)$ is above-right, and $G\big(((p \Rightarrow q) \Rightarrow p) \Rightarrow p\big)$ is as in the introduction. A colouring is ***nice*** if every colour class has at most two vertices and no union of two-vertex colour classes induces a matching. A graph homomorphism $h : G \to G'$ is a ***skew fibration*** (see figure below) if for all $v \in V(G)$ and $h(v)w \in E(G')$ there exists $v\widehat{w} \in E(G)$ with $h(\widehat{w})w \notin E(G')$.



Given a graph homomorphism $h : G \to G'$ with G' an $\mathcal{A}$-labelled graph, a vertex $v \in V(G)$ is ***axiomatic*** if $h(v)$ is labelled 1, and a pair $\{v, w\} \subseteq V(G)$ is axiomatic if $h(v)$ and $h(w)$ are labelled by dual literals.

Definition 3.1. A ***combinatorial proof*** of a proposition ϕ is a skew fibration $h : C \to G(\phi)$ from a nicely coloured cograph C to the graph $G(\phi)$ of ϕ, such that every colour class of C is axiomatic.

A combinatorial proof of $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$ is shown in the introduction. The reader may find it instructive to consider why $p \wedge \neg p$ has no combinatorial proof.

THEOREM 3.1 (Soundness and Completeness). *A proposition is true if and only if it has a combinatorial proof.*

Section 4 reformulates this theorem in terms of combinatorial (non-syntactic, non-inductive) notions of *proposition* and *truth*. Section 5 proves the reformulated theorem.

Notes. The map $\phi \mapsto G(\phi)$ is based on a well understood translation of a boolean formula into a graph [CLS81], and (up to standard graph isomorphism[1]) represents propositions modulo associativity and commutativity of $\wedge$ and $\vee$, double negation $\neg\neg\phi = \phi$, de Morgan duality $\neg(\phi \wedge \rho) = (\neg\phi) \vee (\neg\rho)$ and $\neg(\phi \vee \rho) = (\neg\phi) \wedge (\neg\rho)$, and $\phi \Rightarrow \rho = (\neg\phi) \vee \rho$. Perhaps the earliest graphical representation of propositions is due to Peirce [Pei58, vol. 4:2], dating from the late 1800s.

A skew fibration is a lax notion of graph fibration. A graph homomorphism $h : G \to G'$ is a ***graph fibration*** (see e.g. [BV02]) if for all $v \in V(G)$ and $h(v)w \in E(G')$ there is a unique $v\widehat{w} \in E(G)$ with $h(\widehat{w}) = w$.[2] The definition of skew fibration drops uniqueness and relaxes $h(\widehat{w}) = w$ to 'skewness' $h(\widehat{w})w \notin E(G')$.

Combinatorial proofs constitute a formal *proof system* [CR79] since correctness can be checked in polynomial time.[3] There is a polynomial-time computable function taking a propositional sequent calculus proof of ϕ with $n \geq 0$ cut rules [Gen35] to a combinatorial proof of ϕ ***with*** n ***cuts***: a combinatorial proof of $\phi \vee (\theta_1 \wedge \neg\theta_1) \vee \cdots \vee (\theta_n \wedge \neg\theta_n)$ for propositions θ_i.

In the example of a combinatorial proof drawn in the introduction, observe that the image of the colour class $\circ\;\circ$ under h is $\overset{\bullet}{\bar{p}}\;\overset{\bullet}{p}$. Think of the colour class as actively pairing an occurrence of a variable p with an occurrence of its

[1] Graphs (V, E) and (V', E') are isomorphic if there exists a bijection $h : V \to V'$ with $vw \in E$ if and only if $h(v)h(w) \in E'$.

[2] This is simply a convenient restatement of the familiar notions of fibration in topology [Whi78] and category theory [Gro59], [Gra66]: a graph homomorphism is a graph fibration *if and only if* it satisfies the homotopy lifting property (when viewed as a continuous map by identifying each edge with a copy of the unit interval) *if and only if* it has all requisite cartesian liftings (when viewed as a functor by identifying each graph with its path category).

[3] The skew fibration and axiomatic conditions are clearly polynomial. Checking that a graph G is a cograph is polynomial by constructing its modular decomposition tree $T(G)$ [BLS99], and checking that G is nicely coloured is a simple breadth-first search on $T(G)$.

dual $\overline{p}$. The idea of pairing dual variable occurrences has arisen in the study of various forms of syntax, such as closed categories [KM71], contraction-free predicate calculus [KW84] and linear logic [Gir87]. Combinatorial proofs relate only superficially to the connection/matrix method [Dav71], [Bib74], [And81]; the latter fails to provide a proof system [CR79].

A partially combinatorial notion of proof for classical logic, called a *proof net*, was presented in [Gir91], though promptly dismissed by the author as overly syntactic: a proof net of a proposition ϕ has an underlying syntax tree containing not only $\wedge$'s and $\vee$'s from ϕ, but also auxiliary syntactic connectives which are not even boolean operations (*contraction* and *weakening*).

Nicely coloured cographs with two vertices in every colour class correspond to *unlabelled chorded R&B-cographs* [Ret03]. When labelled, the latter represent proof nets of mixed multiplicative linear logic [Gir87].

4. Combinatorial propositions and truth

A set $W \subseteq V(G)$ is ***stable*** if $vw \notin E(G)$ for all $v,w \in W$. A ***clause*** is a maximal stable set. A clause of an $\mathcal{A}$-labelled graph is ***true*** if it contains a 1-labelled vertex or two vertices labelled by dual literals; an $\mathcal{A}$-labelled graph is ***true*** if its clauses are true. For example, $\underset{\overline{p}}{\bullet}\ \underset{p}{\bullet}\!\!-\!\!\underset{1}{\bullet}$ $(= G(p \Rightarrow (p \wedge 1)))$ is true, with true clauses $\underset{\overline{p}}{\bullet}\ \underset{p}{\bullet}$ and $\underset{\overline{p}}{\bullet}\quad \underset{1}{\bullet}$.

Lemma 4.1. *A proposition ϕ is true if and only if its graph $G(\phi)$ is true.*

Proof. Exhaustively apply distributivity $\theta \vee (\psi_1 \wedge \psi_2) \to (\theta \vee \psi_1) \wedge (\theta \vee \psi_2)$ to ϕ modulo associativity and commutativity of $\wedge$ and $\vee$, yielding a conjunction ϕ' of syntactic clauses (disjunctions of atoms). The lemma is immediate for ϕ' since $G(\phi')$ is a join of clauses, and $G\big(\theta \vee (\psi_1 \wedge \psi_2)\big)$ is true if and only if $G\big((\theta \vee \psi_1) \wedge (\theta \vee \psi_2)\big)$ is true since for non-empty graphs G_1 and G_2, a clause of $G_1 \vee G_2$ (resp. $G_1 \wedge G_2$) is a clause of G_1 and (resp. or) a clause of G_2. $\square$

A ***combinatorial proposition*** is an $\mathcal{A}$-labelled cograph. Since a graph is a cograph if and only if it is derivable from individual vertices by union, join and complement [BLS99, §11.3], the graph $G(\phi)$ of any syntactic proposition ϕ is a combinatorial proposition; conversely, every combinatorial proposition is (isomorphic to) $G(\phi)$ for some ϕ.

Definition 4.1. A ***combinatorial proof*** of a combinatorial proposition P is a skew fibration $h : C \to P$ from a nicely coloured cograph C whose colour classes are axiomatic.

Thus a combinatorial proof of a syntactic proposition ϕ (Def. 3.1) is a combinatorial proof of $G(\phi)$ (Def. 4.1). By Lemma 4.1, the following is equivalent to Theorem 3.1.

Theorem 4.1 (Combinatorial Soundness and Completeness). *A combinatorial proposition is true if and only if it has a combinatorial proof.*

5. Proof of Theorem 4.1

The diagram below shows the dependency between the Lemmas (4.1–5.8) and Theorems (T$i.j$) in this paper.

$$
\begin{tikzcd}
5.1 \arrow[r] \arrow[d] & 5.2 \arrow[dr] & 5.5 \arrow[d] & 5.7 \arrow[dl] \\
5.3 \arrow[r] & 5.4 \arrow[r] & \mathrm{T}5.1 \arrow[dl] \arrow[d] & 5.8 \arrow[l] \arrow[dl] \\
\mathrm{T}3.1 \arrow[r, leftrightarrow, "4.1"] & \mathrm{T}4.1 & \mathrm{T}5.2 \arrow[l] & 5.6 \arrow[l]
\end{tikzcd}
$$

Given a graph homomorphism $h : G \to G'$, an edge $v\widehat{w} \in E(G)$ is a ***skew lifting of*** $h(v)w \in E(G')$ ***at*** v if $h(\widehat{w})w \notin E(G')$. Thus h is a skew fibration if and only if every edge $h(v)w \in E(G')$ has a skew lifting at v.

A graph G is a ***subgraph*** of G', denoted $G \subseteq G'$, if $V(G) \subseteq V(G')$ and $E(G) \subseteq E(G')$. The subgraph $G[W]$ of G ***induced by*** $W \subseteq V(G)$ is $(W, \{ vw \in E(G) : v,w \in W \})$. Let $h : G \to H$ be a graph homomorphism and let G' and H' be induced subgraphs of G and H, respectively. Write $h(G')$ for the induced subgraph $H[h(V(G'))]$ and $h^{-1}(H')$ for the induced subgraph $G[h^{-1}(V(H'))]$. Define the ***restriction*** $h_{\restriction H'} : h^{-1}(H') \to H'$ by $h_{\restriction H'}(v) = h(v)$.

Lemma 5.1. *Let $\diamond \in \{\wedge, \vee\}$. If $h : G \to H_1 \diamond H_2$ is a skew fibration then both restrictions $h_{\restriction H_i}$ are skew fibrations.*

Proof. We prove that if $v\widehat{w}$ is a skew lifting of $h_{\restriction H_i}(v)w = h(v)w \in E(H_i)$ at v with respect to h, then $h(\widehat{w}) \in H_i$; hence $v\widehat{w}$ is a well-defined skew lifting with respect to $h_{\restriction H_i}$. Suppose $h(\widehat{w}) \in H_j$ and $j \neq i$. If $\diamond = \vee$, since h is a homomorphism, $h(v)h(\widehat{w})$ is an edge between H_1 and H_2 in $H_1 \vee H_2$, a contradiction; if $\diamond = \wedge$, since $H_1 \wedge H_2$ has all edges between H_1 and H_2, $h(\widehat{w})w$ is an edge, contradicting $v\widehat{w}$ being a skew lifting with respect to h. □

Lemma 5.2. *Let $h : (G_1 \wedge G_2) \vee (H_1 \vee H_2) \to (K_1 \wedge K_2) \vee L$ be a skew fibration with $h(G_i) \subseteq K_i$ and $h(H_i) \subseteq L$. Then $h_i : G_i \vee H_i \to K_i \vee L$ defined by $h_i(v) = h(v)$ is a skew fibration.*

Proof. Since a graph union $X_1 \vee X_2$ has no edges between X_1 and X_2, (a) if $k : X_1 \vee X_2 \to Y$ is a skew fibration, so also is $k^{\restriction X_i} : X_i \to Y$ defined by $k^{\restriction X_i}(x) = k(x)$, and (b) if $k_i : Z_i \to X_i$ is a skew fibration for $i = 1, 2$, so also is $k_1 \vee k_2 : Z_1 \vee Z_2 \to X_1 \vee X_2$ defined by $(k_1 \vee k_2)(z) = k_i(z)$ if and only if $z \in V(Z_i)$. Since $h_i = h_{\restriction K_i} \vee (h_{\restriction L})^{\restriction H_i}$, it is a skew fibration by (a), (b) and Lemma 5.1. □

Lemma 5.3. *If $h : G \to K$ is a skew fibration into a cograph K, then every clause of K contains a clause of $h(G)$.*

Proof. By induction on the number of vertices in K. The base case with K a single vertex is immediate. Otherwise $K = K_1 \diamond K_2$ for $\diamond \in \{\wedge, \vee\}$ and cographs K_i. Let $G_i = h^{-1}(K_i)$ and $h_i = h_{\restriction K_i} : G_i \to K_i$, a skew fibration by Lemma 5.1. Let C be a clause of K. If $\diamond = \wedge$ then C is a clause of K_j for $j = 1$ or 2; by induction C contains a clause C' of $h_j(G_j)$, also a clause of $h_1(G_1) \wedge h_2(G_2) = h(G)$. If $\diamond = \vee$ then $C = C_1 \cup C_2$ for clauses C_i of K_i; by induction C_i contains a clause C_i' of $h_i(G_i)$, so C contains the clause $C_1' \cup C_2'$ of $h_1(G_1) \vee h_2(G_2) = h(G)$. □

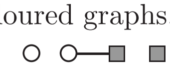
Lemma 5.4. *Let $h : G \to P$ be a skew fibration into a combinatorial proposition P. If $h(G)$ is true then P is true.*

Proof. Lemma 5.3 and the definition of *true*. □

The ***empty*** graph is the graph with no vertices. A graph is ***disconnected*** if it is a union of non-empty graphs, and ***connected*** otherwise. A ***component*** is a maximal non-empty connected subgraph. A graph homomorphism $h : G \to H$ is ***shallow*** if $h^{-1}(K)$ has at most one component for every component K of H.

Lemma 5.5. *For any combinatorial proof $h : G \to P$ there exists a shallow combinatorial proof $h' : G \to P'$ such that P is true if and only if P' is true.*

Proof. Let $G_1, \ldots, G_n$ be the components of G, and let P' be the union of n copies of P defined by $V(P') = V(P) \times \{1, \ldots, n\}$ and $\langle v, i\rangle\langle w, j\rangle \in E(P')$ if and only if $vw \in E(P)$ and $i = j$, and the label of $\langle v, i\rangle$ in P' is equal to the label of v in P. Define $h' : G \to P'$ on $v \in V(G_i)$ by $h'(v) = \langle h(v), i\rangle$. Since P' is a union of copies of P, it is true if and only if P is true (every clause of P' contains a clause of P; conversely the union of n copies of a clause of P is a clause of P'), and h' is a combinatorial proof (skew liftings copied from h). □

A subgraph G' of G is a ***portion*** of G if $G = G' \vee G''$ for some G''. A ***fusion*** of graphs G and H is any graph obtained from $G \vee H$ by selecting portions G' of G and H' of H and adding edges between every vertex of G' and every vertex of H'. Union and join are extremal cases of fusion: union with G', H' empty; join with $G'\!=\!G$, $H'\!=\!H$. On coloured graphs, fusion does not reduce to union and join: the coloured cograph  is a fusion of  and , but is not a union or a join of coloured graphs (since we defined a colouring as an equivalence relation). Henceforth abbreviate *nicely coloured* to *nice*.

Lemma 5.6. *A fusion of nice cographs is a nice cograph.*

Proof. Let C be the fusion of nice cographs C_1 and C_2 obtained by joining portions C'_i of C_i. Suppose U is a union of two-vertex colour classes in C which induces a matching. Let $U_i = U \cap V(C_i)$ and $U'_i = U \cap V(C'_i)$. By definition of fusion, the only edges in C between U_1 and U_2 are between U'_1 and U'_2, and there are edges between all vertices of U'_1 and all vertices of U'_2; thus ($\star$) *there is at most one edge between* U_1 *and* U_2, or else two edges of C on U would intersect. Since U is a union of two-vertex colour classes, each either in U_1 or U_2, each U_i contains an even number of vertices. Therefore, since U induces a matching, (†) *there must be an even number of edges between* U_1 *and* U_2. Together ($\star$) and (†) imply there is no edge between U_1 and U_2; hence, for whichever U_i is non-empty (perhaps both), U_i is a union of two-vertex colour classes inducing a matching in C_i, contradicting C_i being nice. □

LEMMA 5.7. *Every nice cograph with more than one colour class is a fusion of nice cographs.*

Proof. Let C be a nice cograph. Since C is a cograph, its underlying (uncoloured) graph has the form $(C_1 \wedge C_2) \vee (C_3 \wedge C_4) \vee \ldots \vee (C_{n-1} \wedge C_n) \vee H$ for cographs C_i and H with no edges. Assume $n \neq 0$, otherwise the result is trivial. Let G be the graph whose vertices are the C_i, with $C_iC_j \in E(G)$ if and only if there is an edge or colour class $\{v, w\}$ in C with $v \in V(C_i)$ and $w \in V(C_j)$ (cf. the proof of Theorem 4 in [Ret03]). A *perfect matching* is a set of pairwise disjoint edges whose union contains all vertices. Since C is nice, $M = \{C_1C_2, C_3C_4, \ldots, C_{n-1}C_n\}$ is the only perfect matching of G. For if M' is another perfect matching, then $M' \backslash M$ determines a set of two-vertex colour classes in C whose union induces a matching in C: for each $C_iC_j \in M' \setminus M$ pick a colour class $\{v, w\}$ with $v \in V(C_i)$ and $w \in V(C_j)$. Since G has a unique perfect matching, some $C_kC_{k+1} \in M$ is a bridge [Kot59], [LP86], i.e., $(\, V(G),\, E(G) \setminus C_kC_{k+1}\,) = X \vee Y$ with $C_k \in V(X)$ and $C_{k+1} \in V(Y)$. Let W be the union of all colour classes of C coincident with any C_i in X, and let $W' = V(C) \setminus W$. Then $C[W]$ and $C[W']$ are nice (since W and W' are unions of colour classes), and C is the fusion of $C[W]$ and $C[W']$ joining portions C_k of $C[W]$ and C_{k+1} of $C[W']$. □

LEMMA 5.8. *Let* P_1 *and* P_2 *be combinatorial propositions and* Q *a combinatorial proposition or the empty graph. Then* $(P_1 \wedge P_2) \vee Q$ *is true if and only if* $P_1 \vee Q$ *and* $P_2 \vee Q$ *are true.*

Proof. A clause of $(P_1 \wedge P_2) \vee Q$ is a clause of $P_1 \vee Q$ or $P_2 \vee Q$, and vice versa. □

THEOREM 5.1 (Combinatorial Soundness). *If a combinatorial proposition has a combinatorial proof, it is true.*

Proof. Let $h : C \to P$ be a combinatorial proof. We show P is true by induction on the number of colour classes in C. In the base case, $V(C)$ is a colour class. If $v \in V(C)$ then $h(v)$ is in no edge of P (for if $h(v)w \in E(P)$ then a skew lifting at v is an edge in C, a contradiction), hence is in every clause K of P. Since $V(C)$ is axiomatic, K is true.

Induction step. By Lemmas 5.4 and 5.5, assume h is shallow and surjective. By Lemma 5.7, C is a fusion of nice cographs C_1 and C_2 obtained from $C_1 \vee C_2$ by joining portions C_i' of C_i. If $C = C_1 \vee C_2$ then $h' : C_1 \to P$ defined by $h'(v) = h(v)$ is a combinatorial proof, and P is true by the induction hypothesis. Otherwise each C_i' is non-empty. Let $P_i = h(C_i')$. Since $C_1' \wedge C_2'$ is a component of C and h is a shallow surjection, $P_1 \wedge P_2$ is a component of P, say $P = (P_1 \wedge P_2) \vee Q$. Define $h_i : C_i \to P_i \vee Q$ by $h_i(v) = h(v)$, a combinatorial proof: C_i is a nice cograph, the axiomatic colour class property is inherited from h, and h_i is a skew fibration by Lemma 5.2 (applied after forgetting colourings). By the induction hypothesis $P_i \vee Q$ is true; hence P is true by Lemma 5.8. □

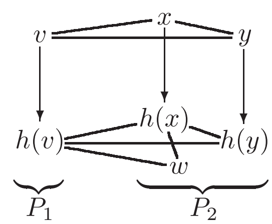
Theorem 5.2 (Combinatorial Completeness). *Every true combinatorial proposition has a combinatorial proof.*

Proof. Let P be a true combinatorial proposition. We construct a combinatorial proof of P by induction on the number of edges in P. In the base case, $V(P)$ is a true clause, so there exists $W \subseteq V(P)$ comprising a 1-labelled vertex or a pair of vertices labelled with dual literals. Inclusion $W \to P$ is a combinatorial proof (viewing W as a graph with no edge and a single colour class, and forgetting its labels).

Induction step. Since P is a cograph with an edge, $P = (P_1 \wedge P_2) \vee Q$ for combinatorial propositions P_i and Q a combinatorial proposition or the empty graph. Assume Q is empty or not true; otherwise by induction there is a combinatorial proof $C \to Q$ composable with inclusion $Q \to P$ for a combinatorial proof of P, and we are done. By Lemma 5.8, $P_i \vee Q$ is true, so by induction has a combinatorial proof $h_i : C_i \to P_i \vee Q$. Let C be the fusion of C_1 and C_2 obtained by joining the portions $h_i^{-1}(P_i)$ of C_i. By Lemma 5.6, C is nice. Define $h : C \to P$ by $h(v) = h_i(v)$ if and only if $v \in V(C_i)$. Then h is a graph homomorphism: let $vw \in E(C)$ with $v \in V(C_i)$ and $w \in V(C_j)$; if $i = j$ then $h(v)h(w) \in E(P)$ since h_i is a homomorphism; if $i \neq j$ then vw arose from fusion, so $h(v) \in P_i$ and $h(w) \in P_j$, hence $h(v)h(w) \in E(P)$ since $P_1 \wedge P_2 \subseteq P$ has all edges between P_1 and P_2.

The axiomatic colour class property for h is inherited from the h_i, so it remains to show that h is a skew fibration. Let $v \in V(C)$ and $h(v)w \in E(P)$. By symmetry, assume $v \in V(C_1)$. Assume $h(v) \in V(P_1)$ and $w \in V(P_2)$; otherwise we immediately obtain a skew lifting of $h(v)w$ since h_1 is a skew fibration. There is a vertex x in $h_2^{-1}(P_2)$: if Q is empty, this is immediate;

otherwise Q is not true and $h_{2\restriction Q} : C_2 \to Q$ would be a combinatorial proof, contradicting soundness. Since fusion joined the $h_i^{-1}(P_i)$, we have $vx \in E(C)$. If $h(x)w \notin E(P_2)$ we are done; otherwise since h_2 is a skew fibration and $h(x)w \in E(P_2)$ there exists $xy \in E(C_2)$ with $h(y)w \notin E(P_2)$. Since $vy \in E(C)$ (again by fusion), we have the desired skew lifting of $h(v)w$ at v. (See figure below. *Note*: $h(y) = w$ is possible.)



$\square$

Stanford University, Stanford, CA
E-mail address: dominic@theory.stanford.edu

References

[And81] P. B. Andrews, Theorem proving via general matings, *J. ACM* **28** (1981), 193–214.

[Bib74] W. Bibel, An approach to a systematic theorem proving procedure in first-order logic, *Computing* **12** (1974), 43–55.

[BV02] P. Boldi and S. Vigna, Fibrations of graphs, *Discrete Math.* **243** (2002), 21–66.

[BLS99] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes*: *A Survey*, *SIAM Monographs on Discrete Math.* & *Applic.*, Soc. for Industrial and Applied Math., Philadelphia, 1999.

[CR79] S. A. Cook and R. A. Reckhow, The relative efficiency of propositional proof systems, *J. Symbolic Logic* **44** (1979), 36–50.

[CLS81] D. G. Corneil, H. Lerchs, and L. Stewart-Burlingham, Complement reducible graphs, *Discr. Appl. Math.* **3** (1981), 163–174.

[Dav71] G. V. Davydov, The synthesis of the resolution method and the inverse method, *Zapiski Nauchnykh Seminarov Lomi* **20** (1971), 24–35; Translation in *J. Sov. Math.* **1** (1973), 12–18.

[Fr1879] G. Frege, *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, 1879; Translation [vHe67, 5–82].

[Gen35] G. Gentzen, Untersuchungen über das logische Schließen I, II, *Math. Z.* **39** (1935), 176–210 405–431; Translation [Gen69, 68–131].

[Gen69] ———, *The Collected Papers of Gerhard Gentzen* (M. E. Szabo, ed.), North-Holland Publ. Co., New York, 1969.

[Gir87] J.-Y. Girard, Linear logic, *Theor. Comp. Sci.* **50** (1987), 1–102.

[Gir91] ———, A new constructive logic: Classical logic, *Math. Struc. Comp. Sci.* **1** (1991), 255–296.

[Gra66] J. W. Gray, *Fibred and cofibred categories*, in *Proc. Conf. on Categorical Algebra* (La Jolla 1965), 21–83, Springer-Verlag, New York, 1966.

[Gro59] A. Grothendieck, Technique de descente et théorèmes d'existence en géométrie algébrique. I: Généralités. Descente par morphismes fidèlement plats, *Séminaire Bourbaki* **190** (1959–1960).

[vHe67] J. van Heijenoort (ed.), *From Frege to Gödel. A Source Book in Mathematical Logic* 1879–1931, Harvard Univ. Press, Boston, 1967.

[Hil28] D. Hilbert, Die Grundlagen der Mathematik, *Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität* **6** (1928), 65–85; Translation [vHe67, 465–479].

[Joh87] P. T. Johnstone, *Notes on Logic and Set Theory*, Cambridge Univ. Press, Cambridge, 1987.

[KM71] G. M. Kelly and S. Mac Lane, Coherence in closed categories, *J. Pure Appl. Algebra* **1** (1971), 97–140; erratum, *ibid.* no. 2, 219.

[KW84] J. Ketonen and R. Weyhrauch, A decidable fragment of predicate calculus, *Theor. Comp. Sci.* **32** (1984), 297–307.

[Kot59] A. Kotzig, On the theory of finite graphs with a linear factor, *Mat.-Fyz. Casopis Slovensk. Akad. Vied* **9, 10**, 1959.

[LP86] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland Publ. Co., New York, 1986.

[deM68] A. de Morgan, Review of J. M. Wilson's 'Elementary Geometry', *The Athenaeum* **2** (1868), 71–73.

[Pei58] C. S. Peirce, *Collected Papers* (Hartshorne, Weiss, and Burks, eds.), Harvard Univ. Press, Boston, 1931–1935 and 1958.

[Ret03] C. Retoré, Handsome proof-nets: perfect matchings and cographs, *Theor. Comp. Sci.* **294** (2003), 473–488.

[Whi78] G. W. Whitehead, *Elements of Homotopy Theory*, Springer-Verlag, New York, 1978.

(Received 20 August, 2004)