Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Sharing in Natural-Deduction Proof Graphs: Canonical Folds and the Complexity of Assumption Renaming

Khanh Pham

khanhvietdm@gmail.com

Vietnam Academy of Science and Technology

Research Article

Keywords: natural deduction, proof graphs, proof compression, assumption discharge, alpha-equivalence, NP-completeness

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11759516/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Sharing in Natural-Deduction Proof Graphs:

Canonical Folds and the Complexity of Assumption Renaming

Pham Van Khanh

Institute of Information Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam

khanhvietdm@gmail.com; ORCID 0000-0002-0227-4254

Abstract

A natural-deduction derivation can be stored as a directed acyclic graph in which repeated subderivations are shared, but sharing interacts with the discharge of assumptions. We study this interaction for proof graphs whose vertices are judgments with identifier-labelled contexts and whose hyperedges are rule instances with a discharge set for each premise, so that well-formedness is a local condition. For fixed identifiers sharing is unproblematic: a quotient of a proof graph is a proof graph exactly when the underlying equivalence is a justification congruence, and the coarsest such congruence yields the unique smallest graph with the same unfolding, computable in expected linear time. The identifiers of discharged assumptions, however, are bound names, and choosing them to maximise sharing is a global optimisation problem. Under the convention that bound identifiers differ from free ones, an optimal choice is computable in expected linear time for derivations of discharge depth one. For discharge depth two, deciding whether some renaming yields a graph with at most k vertices is NP-complete, and computing an optimal renaming admits no polynomial-time approximation scheme unless $\mathrm{P} = \mathrm{NP}$. Hardness already holds for purely implicational minimal logic, under single-identifier as well as set-valued discharge. Contexts summarise locally the open-thread condition of dag-like natural deduction, and, relative to a fixed derivation, level-wise merging of formulas can use exponentially fewer vertices than any proof graph with the same unfolding. The main constructions were checked computationally on small instances.

Keywords: natural deduction; proof graphs; proof compression; assumption discharge; alpha-equivalence; NP-completeness.

1 Introduction

Proofs produced by automated reasoners are increasingly treated as objects in their own right: they are exchanged between systems, checked independently, replayed in proof assistants and analysed for their structure [4, 31, 36], and they can be very large: proofs of unsatisfiability produced by distributed SAT solvers, for instance, can exceed two hundred gigabytes [26]. In these uses a proof is commonly stored as a directed acyclic graph (dag) rather than as a tree: a subproof that is needed at several places is written once and referred to several times. That dag-like derivations can be much shorter than tree-like ones has been known since Statman's work on the structural complexity of proofs [34], and sharing is routine in proof compression for resolution-style calculi [9]. For proof structures without discharge of assumptions, such as

condensed-detachment proofs, the number of inner nodes of the minimal dag, the *compacted size*, has recently been advocated in this journal as the adequate measure of proof size [36]; it is obtained by identifying equal subterms.

Natural deduction plays a role in several of these settings. Some human-oriented provers, such as AProS, search for proofs in natural deduction [32]; machine-found refutations are translated into natural-deduction-style proofs for replay or for presentation in proof assistants [1, 7]; and, through the Curry-Howard correspondence, natural deduction underlies the proof terms that some proof assistants record and check, as in the metalogic of Isabelle [31]. Here sharing is more delicate, because a formula occurrence is not a self-contained object. Whether an occurrence of A may be reused depends on the assumptions that are open above it, and an inference such as $\Rightarrow$-introduction or $\vee$-elimination changes this set by discharging assumptions along a particular premise. Two occurrences of the same formula, even two occurrences of the same *assumption* formula, may therefore play different roles, and identifying them may yield an object that is not a derivation, or a derivation of a different judgment. Proof formats for automated provers meet the same phenomenon when inference steps depend on a context of bound variables [4].

There are two established ways to cope with this. In box-based proof graphs, such as the deduction graphs of Geuvers and Loeb [20] or N-graphs [27], the scope of a discharged assumption is a region of the graph, and correctness is a condition on the arrangement of regions. In the dag-like deductions of Gordeev and Haeusler [22, 23], vertices are plain formulas, nodes with the same formula on the same level are merged, and correctness is a global condition on the *threads* (paths from assumption leaves to the root) of the dag. A third option, closer to sequent-style presentations of natural deduction [19, 29, 35], is to let every vertex carry the context on which it depends. A vertex is then a judgment $\Gamma \vdash \varphi$, where Γ maps assumption identifiers to formulas, and each premise of an inference carries the set of identifiers discharged along it. Correctness becomes local: every inference can be checked from the labels of its premises and its conclusion alone. We call such an object a *typed contextual inference hypergraph* (TCIH); tree-shaped TCIHs are exactly sequent-style natural-deduction derivations with named assumptions.

Locality has a price, and the purpose of this paper is to determine it. The central distinction is between sharing when the assumption identifiers are *fixed* and sharing when the identifiers of discharged assumptions, which are bound names, may be *chosen*. In the first case sharing is as simple as for first-order terms: the logical side conditions impose no constraint beyond equality of subderivations, and the smallest graph is canonical. In the second case one naming has to be chosen for the whole derivation before subderivations are compared. We show that choosing it optimally is, in the worst case, intractable already for derivations of discharge depth two, in which at most two discharging inferences lie on any path from an assumption to the conclusion, although the smallest graph for each individual naming is computed in expected linear time.

Contributions.

1. *The complexity of assumption renaming* (Section 4). A renaming of the discharged identifiers preserves the shape of a derivation, its formulas, the inference at which each assumption leaf is discharged, and the end judgment; it changes which subderivations coincide syntactically. In contrast to term graphs with scoped binders, a TCIH may share a vertex with an open identifier between two different discharging inferences (Example 4.1), so the choice of names matters. Under the usual convention that bound identifiers

differ from free ones, we show the following. For derivations of *discharge depth* one, and more generally for interference-free derivations, a canonical naming is optimal and computable in expected linear time (Proposition 4.4). For discharge depth two, deciding whether some renaming yields a graph with at most k vertices is NP-complete (Theorem 4.9, by a reduction from maximum cut), already for purely implicational minimal logic and regardless of whether an inference discharges one identifier or a set of identifiers per premise; and computing an optimal renaming admits no polynomial-time approximation scheme unless $\mathrm{P} = \mathrm{NP}$ (Corollary 4.11). A second reduction, from 3-terminal multiway cut, shows that under single-identifier discharge the decision problem is NP-complete even when only three discharging inferences interfere (Theorem 4.5).

2. *Canonical folds for fixed identifiers* (Section 3). An equivalence relation on the vertices of a TCIH has a TCIH as its quotient if and only if it is a *justification congruence*: equivalent vertices are concluded by the same rule instance, with the same discharge sets, from pairwise equivalent premises (Theorem 3.3). Equality of labels on equivalence classes and acyclicity of the quotient need not be assumed; both follow. Justification congruences form a lattice whose top element identifies vertices with equal unfoldings; the corresponding quotient, the *maximal fold*, is the unique smallest TCIH with the given unfolding (Theorem 3.8) and is computed by hash-consing on the skeleton, without inspecting contexts (Proposition 3.9). We do not claim a new algorithm here. The point is that contexts, discharge coherence and acyclicity reduce exactly to the classical common-subexpression situation, which makes the optimisation problem of Section 4 well defined and its cost exactly computable. A complementary operation, pruning (Proposition 3.10), preserves only the end judgment.

3. *Judgment-level versus formula-level sharing* (Section 5). The context of a vertex consists exactly of the identifiers that have an open thread to it (Lemma 2.7), so that a TCIH is a deterministic dag-like deduction together with a locally checkable certificate for the thread condition of [22]. Neither discipline dominates the other in general (Example 5.2). On a family Π_n of derivations with $\Theta(n2^n)$ vertices, the level-wise formula quotient has $6n + 1$ classes, whereas every TCIH with the same unfolding has at least 2^n vertices (Proposition 5.3). This is a separation in the number of vertices relative to a fixed derivation, not a comparison of complete proof certificates.

Table 1 summarises the three transformations that occur in the paper and what each of them preserves.

Table 1 The three transformations studied and what they preserve

preserved	transformation	outcome
the unfolding, including all identifiers	admissible folding (Section 3)	canonical smallest graph, the maximal fold; expected linear time
shape, formulas, the binder of every assumption leaf, and the end judgment; the identifiers of discharged assumptions may change	admissible naming followed by the maximal fold (Section 4)	optimum computable in expected linear time at discharge depth one; NP-hard for discharge depth two

preserved	transformation	outcome
the end judgment only	pruning (Proposition 3.10)	not unique; the derivation may change

Scope. All results concern propositional natural deduction (minimal, intuitionistic, or classical, depending on the rule library). The theorems of Sections 3 and 5 use nothing about the rules beyond their format and hold for any library of rule instances in the sense of Section 2; the bounds on running times are stated together with the size measures and the assumptions on which they depend (Section 2.3). Sizes of proof graphs are measured in vertices unless stated otherwise. We do not address proof search, the length of shortest proofs, or first-order eigenvariable conditions; open problems are collected in Section 7.

Related work. Sharing of repeated subterms in a dag is the common-subexpression problem [17]; on acyclic structures it amounts to bisimulation minimisation [16] and is implemented by hash-consing, a technique whose use in verified tools has been studied in this journal [10]. For proof structures without discharge, the resulting minimal dag underlies the compacted size of Wernhard and Bibel [36]. Theorems 3.3 and 3.8 show that, once vertices are judgments and discharge sets are attached to incidences, sharing in natural deduction with fixed identifiers reduces to this classical situation.

For terms with binders, three lines of work are close to ours and differ from it in the relation that is computed. Grabmayer and Rochel [24] obtain maximal sharing for the λ-calculus with letrec by a bisimulation collapse of term graphs that preserves the unfolding; this is the counterpart of our maximal fold. Condoluci, Accattoli and Sacerdoti Coen [14] decide in linear time whether two λ-terms represented with sharing are α-equal after unsharing. Blaauwbroek, Olšák and Geuvers [6] define context-sensitive α-equivalence of subterms as bisimilarity in the graph in which a variable points to its binder: two variable occurrences are identified only if their binders are, the resulting maximal sharing is canonical, and it is computable in time $O(n \log n)$. They explicitly reject the coarser relation under which two subterms are identified whenever the bound variables can be renamed so that the subterms become syntactically equal, because it identifies variables of unrelated binders. In all three cases a variable belongs to the scope of one binder, the relation to be computed is fixed in advance, and the answer is canonical; the same holds for term graphs [5] and for hashing modulo α-equivalence [25]. A different response to the difficulty of choosing names is to dispense with them: in nameless and locally nameless representations, which are widely used in proof assistants [11], a bound variable refers to its binder by position. This is not a solution to the problem studied here, since a position is not a global identifier: whether two subterms become syntactically equal then depends on the number of binders between each of them and its binder, which is what the context-sensitive notion of [6] is designed to abstract from. In a TCIH the coarser identification is sound, because discharge is attached to incidences and not to scopes. It cannot, however, in general be realised for all pairs of subderivations by a single naming, so the problem studied in Section 4 is not the computation of a fixed equivalence but a global optimisation over namings. The reductions used there are related to the hardness proof for aggressive coalescing in register allocation [8], which is likewise obtained from multiway cut [15].

Proof graphs for natural deduction with explicit scoping devices include deduction graphs [20], N-graphs [27] and the proof-graphs for minimal implicational logic of Quispe-Cruz, Haeusler

and Gordeev [30]; proof nets [21] play the corresponding role for linear logic. Gordeev and Haeusler [22, 23] introduced horizontal compression of tree-like deductions in minimal implicational logic into dag-like ones. The complexity-theoretic conclusions they draw from it have been disputed [12] and play no role here; we use only the compression operation itself as a point of comparison (Section 5). Translations of resolution refutations into natural deduction that compress the proof on the way [1] and the fine-grained, context-dependent proof steps of SMT solvers [4] are further settings in which the size of shared natural-deduction-style proof objects matters in practice.

2 Typed contextual inference hypergraphs

2.1 Formulas, contexts, rule instances

Formulas are built from atoms and $\bot$ by $\wedge$, $\vee$ and $\Rightarrow$; $\neg A$ abbreviates $A \Rightarrow \bot$. Let Id be a countably infinite set of *assumption identifiers*. A *context* is a finite partial map $\Gamma\colon \mathsf{Id} \rightharpoonup \mathsf{Form}$, written as a set of bindings $\{a_1\colon A_1, \ldots, a_n\colon A_n\}$ with pairwise distinct a_i. For $D \subseteq \mathsf{Id}$ we write $\Gamma\backslash D$ for the restriction of Γ to $\mathrm{dom}(\Gamma)\backslash D$. Contexts $\Gamma_1, \ldots, \Gamma_k$ are *compatible* if they agree on common identifiers, and then $\biguplus_i \Gamma_i$ denotes their union; otherwise the union is undefined. A *judgment* is a pair (Γ, φ), written $\Gamma \vdash \varphi$.

A *rule instance* ρ has an arity $\mathrm{ar}(\rho) = k \geq 0$, premise formulas $\mathrm{prem}_1(\rho), \ldots, \mathrm{prem}_k(\rho)$, a conclusion $\mathrm{concl}(\rho)$ and, for each premise position i, a *discharge formula* $\mathrm{dis}_i(\rho) \in \mathsf{Form} \cup \{*\}$, where $*$ means that nothing may be discharged along premise i. A *rule library* $\mathcal{R}$ is a set of rule instances. Table 2 lists the library $\mathcal{R}_{\mathrm{NJ}}$ of intuitionistic propositional natural deduction; adding RAA gives the classical library $\mathcal{R}_{\mathrm{NK}}$, and removing $\bot\,\mathsf{E}$ the minimal one. The nullary instance $\mathsf{Ass}_a(A)$ opens the assumption $a\colon A$; it exists for every $a \in \mathsf{Id}$ and every formula A.

Table 2 Rule instances of propositional natural deduction. Discharge is per premise: $\vee\,\mathsf{E}$ may discharge an assumption of A along its second premise and an assumption of B along its third, and nothing along its first.

rule instance	premises	conclusion	discharge formulas
$\mathsf{Ass}_a(A)$	—	A	—
$\wedge\,\mathsf{I}(A,B)$	$A,\ B$	$A \wedge B$	$*, *$
$\wedge\,\mathsf{E}_1(A,B)$, $\wedge\,\mathsf{E}_2(A,B)$	$A \wedge B$	A, resp. B	$*$
$\Rightarrow\,\mathsf{I}(A,B)$	B	$A \Rightarrow B$	A
$\Rightarrow\,\mathsf{E}(A,B)$	$A \Rightarrow B,\ A$	B	$*, *$
$\vee\,\mathsf{I}_1(A,B)$, $\vee\,\mathsf{I}_2(A,B)$	A, resp. B	$A \vee B$	$*$
$\vee\,\mathsf{E}(A,B,C)$	$A \vee B,\ C,\ C$	C	$*,\ A,\ B$
$\bot\,\mathsf{E}(C)$	$\bot$	C	$*$
$\mathsf{RAA}(A)$ (classical)	$\bot$	A	$\neg A$

2.2 Event structures, skeletons, and labels

Definition 2.1 (event structure, skeleton). An *event structure* over $\mathcal{R}$ is a pair $G = (V, E)$ of a finite set V of vertices and a finite set E of *events*

$$e = (\rho;\ (s_1, D_1), \dots, (s_k, D_k);\ t), \qquad \rho \in \mathcal{R},\ k = \mathrm{ar}(\rho),\ s_i, t \in V,$$

where each *discharge set* $D_i \subseteq \mathsf{Id}$ is finite and is empty if $\mathrm{dis}_i(\rho) = *$. The vertices s_i are the *sources* and t is the *target* of e; the pair (e, i) is an *incidence*. G is a *skeleton* if

(U)
every vertex is the target of exactly one event (*unique justification*), and

(A)
the *dependency graph*, which has an edge $s \to t$ whenever s is a source of an event with target t, is acyclic.

In a skeleton we write $e_v = \big(\rho_v; \big(s_1(v), D_1(v)\big), \dots; v\big)$ for the event with target v, and $\mathrm{h}(v)$ for the *height* of v: $\mathrm{h}(v) = 0$ if $\mathrm{ar}(\rho_v) = 0$ and $\mathrm{h}(v) = 1 + \max_i \mathrm{h}\big(s_i(v)\big)$ otherwise. Vertices of height 0 are *leaves*. A vertex u *lies below* v if there is a path from u to v in the dependency graph (possibly of length 0).

Two conventions for discharge are in use, and both are covered by this definition. Under the *single-identifier discipline* every discharge set has at most one element; this is the convention of sequent-style natural deduction and of the λ-calculus, where an identifier names an assumption *class* and any number of leaves may carry the same identifier. Under *set-valued discharge* an inference may discharge several identifiers along one premise, as in presentations in which the occurrences of a discharged assumption carry individual labels. In both cases discharge of several leaves at once is available, as is vacuous discharge ($D_i = \emptyset$). A skeleton all of whose discharge sets have at most one element is a *single-identifier* skeleton. The definitions and results of Sections 2, 3 and 5 hold verbatim for arbitrary finite discharge sets; the distinction matters only in Section 4.

Definition 2.2 (coherent labelling, TCIH). A *labelling* of a skeleton G assigns to each vertex v a judgment $\lambda(v) = (\Gamma_v, \varphi_v)$. It is *coherent* if for every vertex v, with $e_v = (\rho; (s_1, D_1), \dots, (s_k, D_k); v)$:

(W1)
$\varphi_{s_i} = \mathrm{prem}_i(\rho)$ for all i, and $\varphi_v = \mathrm{concl}(\rho)$;

(W2)
for all i and all $a \in D_i$: $a \in \mathrm{dom}\big(\Gamma_{s_i}\big)$ and $\Gamma_{s_i}(a) = \mathrm{dis}_i(\rho)$;

(W3)
if $\rho = \mathsf{Ass}_a(A)$ then $\Gamma_v = \{a{:}\,A\}$; otherwise the contexts $\Gamma_{s_i} \backslash D_i$ are compatible and $\Gamma_v = \biguplus_i \big(\Gamma_{s_i} \backslash D_i\big)$.

A *typed contextual inference hypergraph* (TCIH) is a skeleton that admits a coherent labelling.

Thus an assumption is open at v exactly when its identifier belongs to $\mathrm{dom}(\Gamma_v)$; there is no separate record of which assumptions have been discharged. Condition (W3) is *branch-local*: an identifier discharged along premise i is removed from the contribution of premise i only. For $\vee\,\mathsf{E}$ this forces the two minor premises to be derived separately. If $w\colon\{a\colon A, b\colon B\} \vdash A \wedge B$ were used as both minor premises of $\vee\,\mathsf{E}(A, B, A \wedge B)$ with $D_2 = \{a\}$ and $D_3 = \{b\}$, the target context would retain both $b\colon B$ (from the second premise) and $a\colon A$ (from the third), so the unsound judgment $\{g\colon A \vee B\} \vdash A \wedge B$ is not obtained.

Lemma 2.3 (rigidity).

A skeleton has at most one coherent labelling.

Proof. By induction on $\mathrm{h}(v)$, the label of v is determined: $\varphi_v = \mathrm{concl}(\rho_v)$ by (W1), and Γ_v is determined by ρ_v, the sets $D_i(v)$ and the labels of the sources by (W3). ▫

We may therefore speak of *the* label $\Gamma_v \vdash \varphi_v$ of a vertex of a TCIH. Labels are derived data: a TCIH may be stored as a bare skeleton, and Section 3 shows that sharing can be computed on the skeleton alone.

Proposition 2.4 (soundness).

Let G be a TCIH over $\mathcal{R}_{\mathrm{NJ}}$ (resp. $\mathcal{R}_{\mathrm{NK}}$). For every vertex v, the formula φ_v is intuitionistically (resp. classically) derivable from $\mathrm{ran}(\Gamma_v)$.

Proof. Induction on $\mathrm{h}(v)$. For a leaf, $\varphi_v \in \mathrm{ran}(\Gamma_v)$. For an inner vertex, the induction hypothesis gives $\mathrm{ran}\big(\Gamma_{s_i}\big) \vdash \mathrm{prem}_i(\rho_v)$ for each i. By (W2), every identifier in D_i is bound in Γ_{s_i} to the formula that ρ_v may discharge along premise i, so $\mathrm{ran}\big(\Gamma_{s_i} \backslash D_i\big)$ together with $\mathrm{dis}_i(\rho_v)$, if $D_i \neq \emptyset$, derives the i-th premise. The rule of natural deduction underlying ρ_v then yields $\mathrm{concl}(\rho_v)$ from $\bigcup_i \mathrm{ran}\big(\Gamma_{s_i} \backslash D_i\big) = \mathrm{ran}(\Gamma_v)$. ▫

No semantic side condition is needed beyond the soundness of the rules of the library: for a fixed sound library, well-formedness is correctness.

2.3 Size measures and checking

Three measures of a proof graph have to be kept apart. The *number of vertices* $|V(G)|$ is the measure that is minimised in Sections 3 and 4; in a skeleton every vertex is the target of exactly one event, so it is then also the number of inferences. The *encoding size* is defined for an arbitrary event structure $G = (V, E)$, before any of (U), (A) and (W1)-(W3) is known to hold, by a sum over the events:

$$\| G \| = |V| + \sum_{e \in E} \left(1 + \mathrm{ar}(\rho_e) + \sum_i |D_i(e)| + |\rho_e| \right),$$

where ρ_e and $D_i(e)$ are the rule instance and the discharge sets of the event e, and $|\rho|$ is the size of the encoding of the rule instance ρ, essentially the total size of its formulas. If formulas are interned, so that a rule instance refers to its formulas by pointers into a shared formula dag, then $|\rho| = O\big(1 + \mathrm{ar}(\rho)\big)$ and the size of the formula dag is counted once. The term $|V|$ accounts for

vertices that are the target of no event. In a skeleton each vertex is the target of exactly one event, so $|E| = |V|$ and the sum may equivalently be taken over the vertices, with ρ_v and $D_i(v)$ those of the event e_v; this is the form used from Section 3 on. Finally, for a TCIH the *labelled size* $\| G \|^* = \| G \| + \sum_v |\Gamma_v|$ also accounts for the contexts. The three measures can differ by more than constant factors: $\| G \|^*$ may be quadratic in $\| G \|$, for instance on a chain that accumulates n distinct assumptions, and, if formulas are written out, $\| G \|$ may be quadratic in $|V(G)|$, for instance on a chain of $\wedge$-introductions.

Running times refer to a unit-cost random-access machine. Identifiers and formulas are interned: equal formulas are represented by the same pointer into a shared formula dag, so that two identifiers or two formulas are compared in constant time. If the input writes formulas out, interning them is the hash-consing of the formula trees, which takes expected time linear in their total size and thus in $\| G \|$. A rule instance over interned formulas is a record of $O\big(1 + \mathrm{ar}(\rho)\big)$ pointers, and two such records are compared in time $O\big(1 + \mathrm{ar}(\rho)\big)$. Alternatively, the records themselves can be interned in a preprocessing pass of expected time $O(\| G \|)$, after which rule instances are compared in constant time. All bounds below hold under either convention, and they include the preprocessing. Contexts and discharge sets are stored as hash maps and hash sets, with exact comparison on collision. Bounds that rely on hashing are bounds on the expected running time and are always stated as such; for the results of this paper, "linear time" without qualification refers to a deterministic bound. Replacing hashing by balanced search trees, or by lists sorted by identifier, gives deterministic bounds with an additional logarithmic factor.

Proposition 2.5 (checking).

There is an algorithm that decides whether an event structure G is a TCIH and, if it is, computes its labelling. In a first phase it checks (U) and (A) in time $O(\| G \|)$, without computing any context, and rejects G if one of them fails. Otherwise it runs in expected time

$$O\left(\| G \| + \sum_{v \in V'} \sum_{i \le \mathrm{ar}(\rho_v)} |\Gamma_{s_i(v)}| \right),$$

where $V' = V$ if G is a TCIH, and otherwise V' consists of the vertices processed until a violation of (W1)-(W3) is detected, including the vertex at which it is detected; the contexts occurring in the sum are contexts of sources of vertices in V', all of which have been computed. If the arities of the rule instances of the library are bounded by a constant d, as for $\mathcal{R}_{\mathrm{NJ}}$ and $\mathcal{R}_{\mathrm{NK}}$ with $d = 3$, the bound for a TCIH is $O(d \| G \|^*)$.

Proof. In the first phase, (U) is checked by counting, in one pass over the events, the events with a given target, and (A) by a depth-first search of the dependency graph; both take time $O\big(|V| + |E| + \sum_{e \in E} \mathrm{ar}(\rho_e)\big) \subseteq O(\| G \|)$, a bound in which no vertex-indexed notation occurs, since ρ_v is defined only once (U) is known to hold. In the second phase the vertices are processed in a topological order of the dependency graph, which the depth-first search provides; when a vertex v is processed, the contexts of its sources have been computed. Condition (W1) is checked in time $O\big(1 + \mathrm{ar}(\rho_v)\big)$. Conditions (W2) and (W3) are checked by scanning, for each i, the context $\Gamma_{s_i(v)}$, testing membership in $D_i(v)$, and inserting the remaining bindings into the hash map for Γ_v while testing compatibility; this takes expected time $O\big(|\Gamma_{s_i(v)}| + |D_i(v)|\big)$ per

incidence, and the algorithm stops at the first violation. The context Γ_v is built from the bindings read, so its construction is covered by the same bound. For the second claim, $\Gamma_{s_i(v)} \backslash D_i(v) \subseteq \Gamma_v$ gives $\left|\Gamma_{s_i(v)}\right| \leq |\Gamma_v| + |D_i(v)|$, whence $\sum_i \left|\Gamma_{s_i(v)}\right| \leq d\,|\Gamma_v| + \sum_i |D_i(v)|$. ▫

For libraries of unbounded arity the first bound is not $O(\| G \|^*)$ in general: a vertex whose k premises all coincide with one vertex of context size c costs kc steps but contributes only $k + c$ to $\| G \|^*$.

2.4 Threads

Definition 2.6 (thread). A *thread* from w_0 to w_m in a skeleton is a sequence

$$w_0, i_1, w_1, \ldots, i_m, w_m \qquad (m \geq 0)$$

with $w_{j-1} = s_{i_j}(w_j)$ for $1 \leq j \leq m$. It is *a-open*, for $a \in \mathsf{Id}$, if $a \notin D_{i_j}(w_j)$ for all j.

Lemma 2.7 (contexts are open threads).

Let v be a vertex of a TCIH. Then $a \in \mathrm{dom}(\Gamma_v)$ with $\Gamma_v(a) = A$ if and only if there is a leaf ℓ with $\rho_\ell = \mathsf{Ass}_a(A)$ and an a-open thread from ℓ to v.

Proof. Induction on $\mathrm{h}(v)$. If v is a leaf, Γ_v is $\{a\colon A\}$ when $\rho_v = \mathsf{Ass}_a(A)$ (and is empty for other nullary instances), and the only thread to v is the trivial one. Otherwise, by (W3), $a \in \mathrm{dom}(\Gamma_v)$ iff $a \in \mathrm{dom}\left(\Gamma_{s_i(v)}\right)$ and $a \notin D_i(v)$ for some i; by the induction hypothesis this holds iff some a-open thread from a leaf $\mathsf{Ass}_a(A')$ to $s_i(v)$ extends through the incidence i to an a-open thread to v. Compatibility in (W3) gives $A' = A$. ▫

2.5 Trees, unfoldings, morphisms

A skeleton is *tree-shaped* if it has a vertex r (the root) that is a source of no incidence and every other vertex is a source of exactly one incidence. Tree-shaped TCIHs over $\mathcal{R}_{\mathrm{NJ}}$ are, by (W1)-(W3), exactly the derivations of sequent-style natural deduction with named assumptions [35]; under the Curry-Howard reading of the implicational fragment [33], $\mathsf{Ass}_a(A)$ is the variable a, an $\Rightarrow\mathsf{I}$ event with $D_1 = \{a\}$ is λa, and $\Rightarrow\mathsf{E}$ is application.

Definition 2.8 (unfolding). The *unfolding* of a vertex v of a skeleton is the finite term

$$\mathfrak{u}(v) = \rho_v\langle D_1(v).\,\mathfrak{u}\big(s_1(v)\big), \ldots, D_k(v).\,\mathfrak{u}\big(s_k(v)\big)\rangle.$$

A *rooted* skeleton (G, r) is *trim* if every vertex lies below r; its unfolding is $\mathfrak{u}(r)$.

Terms of this form are in bijection with tree-shaped skeletons, and we identify the two; in particular $\mathfrak{u}(v)$ is a tree-shaped skeleton whose vertices are the subterm occurrences of $\mathfrak{u}(v)$, and it may be exponentially larger than G.

Definition 2.9 (morphism). Let G, G' be skeletons. A map $f\colon V \to V'$ is a *morphism* if for every $v \in V$, $e_{f(v)} = \Big(\rho_v; \big(f\big(s_1(v)\big), D_1(v)\big), \ldots; f(v)\Big)$.

Lemma 2.10.

Let $f\colon G \to G'$ be a morphism of skeletons. Then $\mathrm{u}\big(f(v)\big) = \mathrm{u}(v)$ and $\mathrm{h}\big(f(v)\big) = \mathrm{h}(v)$ for all v. If G' is a TCIH then so is G, and $\lambda_G(v) = \lambda_{G'}\big(f(v)\big)$; if f is surjective and G is a TCIH, then so is G'.

Proof. The first two claims follow by induction on $\mathrm{h}(v)$. Conditions (W1)-(W3) for e_v mention only ρ_v, the sets $D_i(v)$ and the labels of the sources and of v; hence $\lambda_{G'} \circ f$ is a coherent labelling of G whenever $\lambda_{G'}$ is one of G'. For the last claim, note that the label of a vertex of a TCIH is a function of its unfolding (apply the first part to the canonical morphism from the tree $\mathrm{u}(v)$ to G that sends a subterm occurrence to the vertex it unfolds). Hence λ_G is constant on the fibres of f and descends to a labelling of G', which is coherent because every event of G' is the image of an event of G. ▫

3 Admissible foldings

3.1 The characterisation

Let G be a skeleton and $\sim$ an equivalence relation on V, with classes $[v]$. The *quotient* $G/\!\sim$ is the event structure with vertex set $V/\!\sim$ and events

$$[e_v] = \big(\rho_v; \big([s_1(v)], D_1(v)\big), \ldots, \big([s_k(v)], D_k(v)\big); [v]\big), \qquad v \in V.$$

Events that become equal are identified. In general $G/\!\sim$ violates (U) or (A), or has no coherent labelling. We call $\sim$ an *admissible folding* of a TCIH G if $G/\!\sim$ is a TCIH.

Definition 3.1 (justification congruence). An equivalence $\sim$ on the vertices of a skeleton is a *justification congruence* if $u \sim v$ implies $\rho_u = \rho_v$ and, for every $i \le \mathrm{ar}(\rho_u)$, $D_i(u) = D_i(v)$ and $s_i(u) \sim s_i(v)$.

Lemma 3.2.

If $\sim$ is a justification congruence and $u \sim v$, then $\mathrm{h}(u) = \mathrm{h}(v)$ and $\mathrm{u}(u) = \mathrm{u}(v)$.

Proof. By induction on $\min\big(\mathrm{h}(u), \mathrm{h}(v)\big)$. If, say, $\mathrm{h}(u) = 0$, then $\mathrm{ar}(\rho_v) = \mathrm{ar}(\rho_u) = 0$, so v is a leaf with the same rule instance. Otherwise the sources are pairwise equivalent, each pair has smaller minimum height, and the claim follows from the induction hypothesis and the definitions of h and u. ▫

Theorem 3.3 (characterisation of admissible foldings).

Let G be a TCIH and $\sim$ an equivalence relation on its vertices. The following are equivalent:

1. *$G/\!\sim$ is a TCIH;*
2. *$G/\!\sim$ satisfies unique justification (U);*
3. *$\sim$ is a justification congruence.*

If these hold, the projection $\pi\colon v \mapsto [v]$ is a surjective morphism, labels, heights and unfoldings are constant on the classes of $\sim$, and $[v] \mapsto \mathrm{h}(v)$ strictly increases along every edge of the dependency graph of $G/\!\sim$.

Proof. (a)⇒(b) is trivial. (b)⇒(c): if $u \sim v$, the events $[e_u]$ and $[e_v]$ have the same target $[u] = [v]$ and are therefore equal by (U); equality of the two tuples is the statement that $\rho_u = \rho_v$, $D_i(u) = D_i(v)$ and $[s_i(u)] = [s_i(v)]$ for all i.

(c)⇒(a): Every event of $G/\!\sim$ with target $[v]$ is of the form $[e_u]$ with $u \sim v$, and all of these coincide by congruence; this is (U). An edge $[s] \to [t]$ of the dependency graph of $G/\!\sim$ arises from some $u \in [t]$ and some i with $s_i(u) \in [s]$, so $\mathrm{h}\big(s_i(u)\big) < \mathrm{h}(u)$. By Lemma 3.2 the height is constant on classes, hence it strictly increases along every edge of the quotient, and the quotient is acyclic; this is (A). Thus $G/\!\sim$ is a skeleton and π is a morphism, which is surjective. By Lemma 2.10, $G/\!\sim$ is a TCIH and π preserves labels and unfoldings. ▫

The theorem isolates what has to be checked and, equally, what does not. We spell out the consequences that are most often relevant.

Corollary 3.4 (necessary conditions).

If $\sim$ is an admissible folding and $u \sim v$, then $\Gamma_u = \Gamma_v$, $\varphi_u = \varphi_v$ and u, v have the same justification up to $\sim$. In particular no admissible folding identifies two vertices with the same formula and different contexts.

Example 3.5 (content-only merging). Consider the derivation of $\{b\colon A\} \vdash (A \Rightarrow A) \wedge A$ with vertices $v_1\colon \{a\colon A\} \vdash A$ (by $\mathsf{Ass}_a(A)$), $v_2\colon \emptyset \vdash A \Rightarrow A$ (by $\Rightarrow \mathsf{I}$ from v_1 with $D_1 = \{a\}$), $v_3\colon \{b\colon A\} \vdash A$ (by $\mathsf{Ass}_b(A)$) and v_4 (by $\wedge\,\mathsf{I}$ from v_2, v_3). The equivalence merging v_1 and v_3, which carry the same formula, is not a congruence, since $\rho_{v_1} \neq \rho_{v_3}$, and its quotient violates (U): the merged vertex w has two justifications. Discarding one of them is no longer the quotient, and it does not preserve the labelled derivation either. If $\mathsf{Ass}_b(A)$ is kept, the event of v_2, which discharges a, violates (W2). If $\mathsf{Ass}_a(A)$ is kept, the result is a TCIH: by (W3) the identifier a is discharged along the incidence from w to v_2 and stays open along the incidence from w to v_4, so that v_4 is labelled $\{a\colon A\} \vdash (A \Rightarrow A) \wedge A$. This is a correct derivation, but of a different judgment: the root context has changed from $\{b\colon A\}$ to $\{a\colon A\}$. The two leaves can be shared without changing the end judgment only after renaming the discharged identifier a to b, that is, by letting a bound identifier coincide with a free one. Renamings are the subject of Section 4, and this particular coincidence is discussed in Remark 4.15(3).

Example 3.6 (equal judgments, different justifications). Let $\Gamma = \{a\colon A, b\colon B\}$, and let $u\colon \Gamma \vdash A$ be obtained by $\wedge\,\mathsf{E}_1$ from $\Gamma \vdash A \wedge B$ and $v\colon \Gamma \vdash A$ by $\wedge\,\mathsf{E}_2$ from $\Gamma \vdash B \wedge A$. The labels of u and v coincide, yet merging them is not admissible. Equality of labels is necessary but not sufficient; what can be done with such pairs is the subject of Proposition 3.10.

Remark 3.7 (acyclicity). For arbitrary equivalences on a dag, acyclicity of the quotient is a genuine extra condition, and it is not implied by the classes being antichains with respect to reachability: on the dag with edges $a_1 \to b_1$ and $b_2 \to a_2$, the classes $\{a_1, a_2\}$ and $\{b_1, b_2\}$ are antichains, but the quotient is a 2-cycle. Theorem 3.3 shows that for justification congruences no such test is needed: the height function of G itself is a rank certificate for $G/\!\sim$.

3.2 The lattice of foldings and the maximal fold

Let $\mathrm{Con}(G)$ be the set of justification congruences of a skeleton G, ordered by inclusion.

Theorem 3.8 (maximal fold).

Let G be a TCIH.

1. $\mathrm{Con}(G)$ *is a lattice with least element the identity; the meet of two congruences is their intersection and the join is the transitive closure of their union.*
2. *The greatest element $\equiv_G$ of $\mathrm{Con}(G)$ is given by $u \equiv_G v \Leftrightarrow \mathfrak{u}(u) = \mathfrak{u}(v)$.*
3. *Write $G^\flat = G/{\equiv_G}$. If (G, r) and (G', r') are trim rooted TCIHs with $\mathfrak{u}(r) = \mathfrak{u}(r')$, then there is a unique morphism $g\colon G' \to G^\flat$, it is surjective, and $g(r') = [r]$. Consequently $\left|V\left(G^\flat\right)\right| \leq |V(G')|$, with equality if and only if g is an isomorphism, and $\left|V\left(G^\flat\right)\right|$ is the number of distinct subterms of $\mathfrak{u}(r)$.*

Proof. (a) The intersection of two congruences is a congruence. Let $\approx$ be the transitive closure of $\sim_1 \cup \sim_2$ and $u \approx v$ via $u = w_0, w_1, \ldots, w_n = v$ with consecutive vertices related by $\sim_1$ or $\sim_2$. Consecutive vertices have the same rule instance and discharge sets, and their i-th sources are related by $\sim_1$ or $\sim_2$; hence $\rho_u = \rho_v$, $D_i(u) = D_i(v)$ and $s_i(u) \approx s_i(v)$.

(b) The relation $\mathfrak{u}(u) = \mathfrak{u}(v)$ is a congruence, because equality of two terms means equality of the root symbols ρ, of the sets D_i and of the immediate subterms. It contains every congruence by Lemma 3.2.

(c) Since G' is trim, each $\mathfrak{u}(v')$, $v' \in V'$, is a subterm of $\mathfrak{u}(r') = \mathfrak{u}(r)$, and every subterm of $\mathfrak{u}(r)$ is the unfolding of a vertex below r. By (b), the vertices of $G^\flat$ correspond bijectively to the distinct subterms of $\mathfrak{u}(r)$. Let $g(v')$ be the class of the vertices of G with unfolding $\mathfrak{u}(v')$. Then g is a morphism, since the event of that class has rule instance and discharge sets read off the root of $\mathfrak{u}(v')$ and has as sources the classes of the immediate subterms. It is surjective because G' realises every subterm of $\mathfrak{u}(r')$. It is unique because morphisms preserve unfoldings (Lemma 2.10) and distinct vertices of $G^\flat$ have distinct unfoldings. If $\left|V\left(G^\flat\right)\right| = |V(G')|$, then g is a bijective morphism, and its inverse is a morphism since events are determined by their targets. ▫

Hence every derivation has a canonical maximally shared representation, the admissible foldings of a TCIH are exactly the intermediate stages between it and this representation, and any two admissible foldings have a common admissible coarsening.

Proposition 3.9 (computing the maximal fold).

Let G be a skeleton whose rule instances are given over interned formulas and whose discharge sets are given as lists sorted by identifier. Then $\equiv_G$ and the quotient skeleton $G/{\equiv_G}$ can be computed in expected time $O(\| G \|)$, or deterministically in time $O(\| G \| \log |V(G)|)$, without computing any context. If G is a TCIH, the result is its maximal fold $G^\flat$.

Proof. Process the vertices in a topological order of the dependency graph, maintaining a dictionary from keys $\left(\rho_v, D_1(v), \ldots, D_k(v), c\left(s_1(v)\right), \ldots, c\left(s_k(v)\right)\right)$ to class numbers, where $c(s)$

is the class number already assigned to s; a vertex receives the class number of its key, a fresh one if the key is new. By induction along the order, $c(u) = c(v)$ iff $\mathfrak{u}(u) = \mathfrak{u}(v)$. A key has length $O(1 + \mathrm{ar}(\rho_v) + \sum_i |D_i(v)|)$ and, the discharge sets being sorted, two keys denote the same justification iff they are equal as sequences. With a hash table in which colliding keys are compared exactly, each vertex costs expected time proportional to the length of its key; with a balanced search tree over lexicographically ordered keys, each of the $O(\log|V(G)|)$ comparisons per vertex costs time proportional to the key length. The last claim follows from Theorems 3.3 and 3.8(b). This is the standard treatment of the acyclic common-subexpression problem [17, 16]. ▫

The algorithm takes a skeleton as input and does not verify that it is a TCIH; verification is the task of Proposition 2.5 and does depend on the contexts. What Theorem 3.3 contributes is that, for an input known to be a TCIH, maximal sharing needs no access to the labels: although an admissible folding must respect labels (Corollary 3.4), its cost is governed by $\| G \|$ and not by the possibly quadratic $\| G \|^{*}$.

3.3 Pruning: an identity-changing reduction

Admissible foldings preserve the unfolding, and thus the derivation that a TCIH denotes. If only the end judgment is to be preserved, vertices with equal labels and different justifications (Example 3.6) can be exploited as well; with judgment labels this is immediate.

Proposition 3.10 (pruning).

Let $u \neq v$ be vertices of a TCIH G with $\lambda(u) = \lambda(v)$ such that v does not lie below u. Let $G[v \mapsto u]$ be obtained by replacing v with u in every source position. Then $G[v \mapsto u]$ is a TCIH in which every vertex has the same label as in G. Consequently, from every trim rooted TCIH (G, r) one obtains in polynomial time a trim rooted TCIH with the same root label and at most as many vertices in which distinct vertices have distinct labels.

Proof. Conditions (W1)-(W3) for an event depend on its sources only through their labels, so λ remains coherent, and (U) is untouched. A new dependency edge has the form $u \to t$ where v was a source of e_t, and all other edges are edges of G. Since every new edge leaves u, a cycle in $G[v \mapsto u]$ would contain a new edge $u \to t$ followed by a path from t back to u that consists of edges of G only; together with the edge $v \to t$ of G this places v below u in G, contrary to the hypothesis. For the second claim, as long as two vertices $u \neq v$ of the trim structure have equal labels, at most one lies below the other by (A); say v does not lie below u. If $v = r$, replace (G, r) by the substructure of the vertices below u, rooted at u; otherwise pass to $G[v \mapsto u]$ and remove the vertices that no longer lie below r, among them v. Each step strictly decreases the number of vertices and preserves the root label. ▫

The result of pruning is not unique and in general denotes a different derivation. Minimising the number of vertices over all TCIHs that can be obtained from a given one by pruning is a combinatorial problem of its own, the search for a smallest derivation of the end judgment that uses only inferences occurring in the given graph; we do not address it. It should not be confused with finding a shortest proof of the end judgment in the logic, where inferences that do not occur in the given graph are available as well. Size-reducing transformations that preserve the conclusion of a derivation but not the derivation itself have been studied for Fitch-style natural

deduction [2], and similar compression steps are applied to natural-deduction-style proofs obtained from automated provers [7].

4 Sharing up to renaming of discharged identifiers

4.1 Alpha-variants and the optimisation problem

The maximal fold is canonical for a derivation with fixed identifiers. The identifiers of discharged assumptions, however, are bound names, and the choice of names affects which subderivations are syntactically equal.

Example 4.1 (sharing across discharging events). Let $\sigma(a)$ denote the term $\wedge\, \mathsf{I}(p,q)\langle \mathsf{Ass}_a(p), \mathsf{Ass}_c(q)\rangle$, which has the label $\{a\colon p, c\colon q\} \vdash p \wedge q$, and consider a derivation that contains the two subderivations

$$\Rightarrow \mathsf{I}\langle \{a_1\}.\, \sigma(a_1)\rangle \qquad \text{and} \qquad \Rightarrow \mathsf{I}\langle \{a_2\}.\wedge\, \mathsf{I}\langle \sigma(a_2), \mathsf{Ass}_d(r)\rangle\rangle.$$

If $a_1 \neq a_2$, the two copies of σ are different terms and cannot be folded. After renaming a_2 to a_1 they coincide, and the maximal fold contains a single vertex $\{a_1\colon p, c\colon q\} \vdash p \wedge q$ whose identifier a_1 is discharged by two different events along two different threads (Fig. 1). This is a TCIH, since (W1)-(W3) are local. It is not expressible if the scope of a discharge is a region with a single binder, where a subterm with a free variable cannot belong to two scopes.

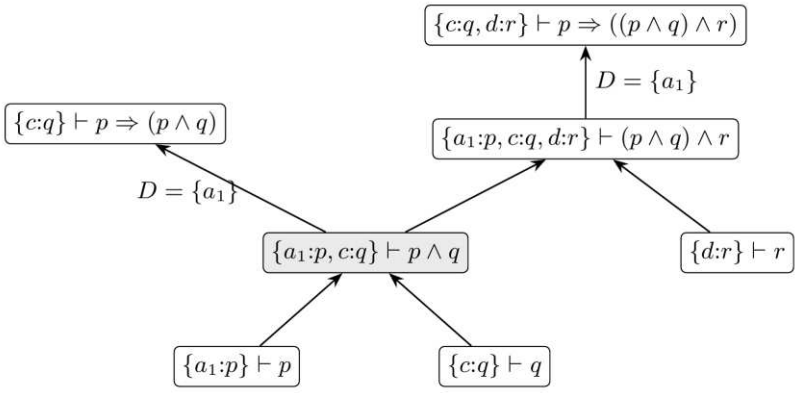

Fig. 1 The shaded vertex is shared between two discharging events (Example 4.1)

Let T be a tree-shaped TCIH. For a leaf ℓ with $\rho_\ell = \mathsf{Ass}_a(A)$, let $\ell = w_0, i_1, w_1, \ldots, i_m, w_m = r$ be the unique thread from ℓ to the root. The *binder* $\beta_T(\ell)$ of ℓ is the incidence $\left(e_{w_j}, i_j\right)$ with the least j such that $a \in D_{i_j}\left(w_j\right)$; if there is none, ℓ is *free* and a is a *free identifier* of T. An incidence with non-empty discharge set is a *discharging incidence*; by (W2) and Lemma 2.7 it is the binder of at least one leaf. An inference whose discharge is vacuous thus has no discharging incidence.

Definition 4.2 (naming, α-variant). A *naming* ν of T assigns an identifier to each non-free leaf. T^ν is obtained from T by replacing $\mathsf{Ass}_a(A)$ with $\mathsf{Ass}_{\nu(\ell)}(A)$ at each non-free leaf ℓ and the discharge set of each discharging incidence (e,i) with $\{\nu(\ell) \mid \beta_T(\ell) = (e,i)\}$. The naming is

admissible if (i) T^{ν} is a TCIH, (ii) $\beta_{T^{\nu}} = \beta_T$, and (iii) no value of ν is a free identifier of T. The trees T^{ν} for admissible ν are the *α-variants* of T. A naming is a *single-identifier naming* if $\nu(\ell) = \nu(\ell')$ whenever $\beta_T(\ell) = \beta_T(\ell')$; such a naming is determined by, and will be identified with, a map from the discharging incidences of T to identifiers, and T^{ν} is then a single-identifier tree.

What a naming preserves. For an admissible naming ν, the tree T^{ν} has the same shape as T, the same rule names and formulas, the same free leaves with the same identifiers, the same binder map by (ii), and hence, by Lemma 2.7, the same root judgment. What a naming changes is the partition of the non-free leaves into classes that carry the same identifier, and with it the discharge sets. A single-identifier naming assigns one identifier to each discharging incidence; this is α-conversion as in the λ-calculus, where a binder binds one variable. A general naming is more liberal: the leaves bound at one incidence may receive several identifiers, all of which are then discharged there, so that the identifier classes of T may be split as well as merged. This corresponds to presentations of natural deduction in which the occurrences of a discharged assumption carry individual labels. We use the term α-variant in this wider sense only in connection with set-valued discharge.

Hygiene. Condition (ii) excludes capture. Condition (iii) is the usual convention that bound names differ from free ones; call a tree *hygienic* if no identifier is carried both by a free and by a non-free leaf. Every α-variant is hygienic, and T is one of its own α-variants if and only if it is hygienic. For instance, $T = \wedge\, \mathsf{I}\langle \Rightarrow \mathsf{I}\langle \{a\}.\, \mathsf{Ass}_a(A)\rangle, \mathsf{Ass}_a(A)\rangle$ is a TCIH in which a is bound in the first premise and free in the second; its α-variants are the trees $\wedge\, \mathsf{I}\langle \Rightarrow \mathsf{I}\langle \{c\}.\, \mathsf{Ass}_c(A)\rangle, \mathsf{Ass}_a(A)\rangle$ with $c \neq a$. Since T^{ν} depends on T only through its shape, its formulas, its binder map and its free identifiers, every tree has α-variants (name the discharging incidences by pairwise distinct fresh identifiers) and may be replaced by a hygienic one without changing the set of its α-variants. The α-variants of T are thus the hygienic representatives of its class under renaming of discharged identifiers, and the optimisation below ranges over these. Condition (iii) is a genuine restriction of the search space and not a technicality: Example 3.5 exhibits a sharing that it excludes, and Remark 4.15(3) shows that the complexity of the problem changes if it is dropped. All results of this section are relative to Definition 4.2, including (iii).

Problem 4.3 (Min-α-Fold and Min-α-Fold^1). Min-α-Fold: given a tree-shaped TCIH T and an integer k, is there an admissible naming ν of T such that $(T^{\nu})^{\flat}$ has at most k vertices? Min-α-Fold^1: the same question for a single-identifier tree T and admissible single-identifier namings.

Both are decision problems. By the corresponding *optimisation problems* we mean the task of computing an admissible naming, respectively an admissible single-identifier naming, ν that minimises $\left|V\left((T^{\nu})^{\flat}\right)\right|$. Min-α-Fold is the problem for set-valued discharge and Min-α-Fold^1 the problem for the single-identifier discipline. Every single-identifier naming is a naming, so on a single-identifier tree the optimum of Min-α-Fold is at most that of Min-α-Fold^1; Example 4.8 shows that it can be strictly smaller, so the two problems are different.

The global character of the problem. In both problems one naming is chosen for the whole tree, and only then is the maximal fold computed. This distinguishes them from the problems for terms with binders discussed in the introduction. Deciding whether two given terms are α-equivalent [14], or collapsing a term graph by a bisimulation that is fixed in advance [24, 6], has a canonical answer that is computed in linear or nearly linear time. Here, by contrast, each of two

coincidences between subtrees may be achievable by some admissible naming while no admissible naming achieves both; the reductions of Sections 4.3 and 4.4 are built from such conflicts. The relation "can be made equal by some admissible naming" is therefore not an equivalence that one naming realises, and the question is which of the possible coincidences to realise simultaneously. For each fixed ν the number $\left|V\left((T^{\nu})^{\flat}\right)\right|$ is computed in expected linear time by Proposition 3.9; the difficulty lies entirely in the choice of ν.

4.2 A tractable case

A leaf ℓ with $\rho_{\ell} = \mathsf{Ass}_a(A)$ and binder $\left(e_{w_j}, i_j\right)$ *suffers interference* if its thread passes, strictly before its binder, through a discharging incidence $\left(e_{w_{j'}}, i_{j'}\right)$, $j' < j$, with the same discharge formula, $\mathrm{dis}_{i_{j'}}\left(\rho_{w_{j'}}\right) = A$. T is *interference-free* if no leaf suffers interference. For example, the derivation of $A \Rightarrow \left(A \Rightarrow (A \wedge A)\right)$ corresponding to $\lambda x.\, \lambda y.\, \langle x, y\rangle$, in which both assumptions are used, is not interference-free: the leaf x meets the incidence that discharges y before its own binder. The same holds, within the implicational fragment, for the derivation $\lambda x.\, \lambda y.\, (f\ x)\ y$ of $A \Rightarrow (A \Rightarrow B)$ from a free assumption $f\colon A \Rightarrow (A \Rightarrow B)$. The derivation of $A \Rightarrow \left(B \Rightarrow (A \wedge B)\right)$ with $A \neq B$ is interference-free. So is the derivation of $A \Rightarrow (A \Rightarrow A)$ corresponding to $\lambda x.\, \lambda y.\, x$: as there is no leaf y, its inner $\Rightarrow$-introduction discharges vacuously and has no discharging incidence, so that the leaf x meets only its own binder. A discharging incidence is *involved in interference* if some leaf that suffers interference is bound at it or meets it before its binder. The *discharge depth* of T is the maximum number of discharging incidences on a thread from a leaf to the root. A tree of discharge depth at most one is interference-free, because a non-free leaf then meets no discharging incidence other than its binder.

Proposition 4.4 (canonical naming).

Choose for each formula A an identifier k_A that is not free in T, with $k_A \neq k_B$ for $A \neq B$, and let κ name every non-free leaf of formula A by k_A, so that every discharging incidence with discharge formula A receives the discharge set $\{k_A\}$. If T is interference-free, then κ is an admissible single-identifier naming and $\left|V\left((T^{\kappa})^{\flat}\right)\right| \leq \left|V\left((T^{\nu})^{\flat}\right)\right|$ for every admissible naming ν. Hence Min-α-Fold and Min-α-Fold^{l} restricted to interference-free trees, as well as the corresponding optimisation problems, are solvable in expected linear time.

Proof. Admissibility. (iii) holds by the choice of the k_A. For (ii), let ℓ be a leaf of formula A with binder (e, i) in T; in T^{κ} it carries k_A. The discharging incidences on its thread before (e, i) have discharge formulas different from A, by interference-freeness, hence discharge sets $\{k_{A'}\}$ with $k_{A'} \neq k_A$ in T^{κ}, and (e, i) itself has the discharge set $\{k_A\}$; so $\beta_{T^{\kappa}}(\ell) = (e, i)$. A free leaf keeps its identifier, which is not of the form k_B, and remains free. For (i): in T^{κ} every identifier is attached to a single formula (the k_A by construction, the free identifiers because all of them occur in the root context of T), so all unions in (W3) are compatible; (W1) is unaffected by renaming; and (W2) holds at each discharging incidence (e, i) because, by (ii), a leaf bound there has a k_A-open thread to the i-th source of e, so that k_A is in the context of that source by Lemma 2.7, whose proof uses only (W3).

Optimality. Let ν be admissible, and let x, y be vertices of T whose subtrees are equal as terms in T^{ν}. Corresponding vertices of the two subtrees carry the same rule name and the same formulas, and corresponding incidences are both discharging or both not. By (iii), corresponding leaves are both free in T, in which case they carry the same identifier in T and in T^{κ}, or both non-free, in which case they have the same formula A and both carry k_A in T^{κ}. Corresponding discharging incidences have the same discharge formula and hence the same discharge set under κ. Thus the two subtrees are also equal in T^{κ}: the partition of the vertices into classes of equal subterms under κ is coarser than under ν, and the claim follows from Theorem 3.8(c). ▫

4.3 NP-completeness with three interfering discharges

Theorem 4.5.

Min-α-Fold$^{|}$ is NP-complete. It remains NP-hard for single-identifier trees over the rule instances $\mathsf{Ass}_a(A)$, $\wedge\mathsf{I}$ *and* $\Rightarrow\mathsf{I}$ *in which all discharged formulas are one and the same atom, the discharge depth is three, and only three discharging incidences are involved in interference.*

Membership. Let L be the number of non-free leaves of T, and fix identifiers $x_1, \dots, x_L$ that are not free in T. Conditions (W1)-(W3), the binder map and equality of subterms depend on identifiers only through equality. Hence, if ν is admissible and π maps the at most L values of ν injectively into $\{x_1, \dots, x_L\}$, then $\pi \circ \nu$ is admissible, is a single-identifier naming if ν is, and gives a maximal fold of the same size, because π, extended by the identity on the free identifiers of T, is injective by condition (iii). For Min-α-Fold as well as for Min-α-Fold$^{|}$ a certificate is therefore a map from the non-free leaves to $\{1, \dots, L\}$, of $O(L\log L)$ bits. Given it, conditions (i)-(iii) are checked in polynomial time, for instance by Proposition 2.5 and a computation of the binders, and $(T^{\nu})^{\flat}$ is computed by Proposition 3.9; the deterministic variants of these two algorithms are used, so that the verification is deterministic polynomial time.

The reduction. We reduce from 3-TERMINAL MULTIWAY CUT with unit weights: given a graph H with three distinguished vertices t_1, t_2, t_3 and an integer c, is there a map $\chi\colon V(H) \to \{1,2,3\}$ with $\chi(t_i) = i$ that leaves at most c edges bichromatic? This problem is NP-hard [15]. We may assume that no edge joins two terminals, since such an edge is bichromatic under every χ. Write $U = V(H)\backslash\{t_1, t_2, t_3\}$ and $F = E(H)$, and let $\mathrm{mc}(H)$ be the minimum number of bichromatic edges.

Fix atoms p, q_e ($e \in F$), r_u ($u \in U$) and $r_{\top}$, and pairwise distinct identifiers c_e, d_u, $d_{\top}$, a_1, a_2, a_3 and a_u ($u \in U$). We use term notation: a stands for $\mathsf{Ass}_a(p)$, and $c_e, d_u, d_{\top}$ for $\mathsf{Ass}_{c_e}(q_e)$, $\mathsf{Ass}_{d_u}(r_u)$, $\mathsf{Ass}_{d_{\top}}(r_{\top})$; $\langle\tau, \tau'\rangle$ is $\wedge\mathsf{I}$ applied to τ, τ'; and $\lambda a.\,\tau$ is $\Rightarrow\mathsf{I}(p,\cdot)$ applied to τ with discharge set $\{a\}$. For an edge e and an identifier a let

$$S_e(a) = \langle a, c_e\rangle, \qquad \text{with label } \{a\colon p,\ c_e\colon q_e\} \vdash p \wedge q_e.$$

For $u \in U$ with incident edges $e_1, \dots, e_m$ (in a fixed order) and for the terminals, with $f_1, \dots, f_l$ the edges incident to t_1, then those incident to t_2, then those incident to t_3, and $\iota(j) \in \{1,2,3\}$ the index of the terminal of f_j, let

$$\begin{aligned} B_u(a) &= \langle \cdots \langle\langle a, d_u\rangle, S_{e_1}(a)\rangle \cdots, S_{e_m}(a)\rangle, & R_u &= \lambda a_u.\ B_u(a_u), \\ B_{\top} &= \langle \cdots \langle\langle\langle a_1, d_{\top}\rangle, a_2\rangle, a_3\rangle, S_{f_1}(a_{\iota(1)})\rangle \cdots, S_{f_l}(a_{\iota(l)})\rangle, & R_{\top} &= \lambda a_1.\lambda a_2.\lambda a_3.\ B_{\top}. \end{aligned}$$

Finally, with $U = \{u_1, \dots, u_n\}$,

$$T_H = \langle \cdots \langle \langle R_\top, R_{u_1} \rangle, R_{u_2} \rangle \cdots, R_{u_n} \rangle.$$

T_H is a tree-shaped TCIH with $O(|F| + |U|)$ vertices whose root context consists of the bindings $c_e : q_e$, $d_u : r_u$ and $d_\top : r_\top$. Each of its formulas has size $O(|F| + |U|)$, so its encoding size $\| T_H \|$ is $O((|F| + |U|)^2)$, and $O(|F| + |U|)$ if formulas are interned; in either case T_H is computed from H in polynomial time. Each edge $e = xy$ contributes two copies of the gadget S_e, one in the body of the binder of x and one in that of y. The discharging incidences of T_H are the three incidences of $R_\top$, which we identify with t_1, t_2, t_3, and one for each $u \in U$; a single-identifier naming is thus a map $\nu : V(H) \to \mathsf{Id}$.

Lemma 4.6.

A single-identifier naming ν of T_H whose values are not free identifiers of T_H is admissible if and only if $\nu(t_1), \nu(t_2), \nu(t_3)$ are pairwise distinct.

Proof. The thread of a leaf a_u passes through no discharging incidence before its binder, and the leaves a_1, a_2, a_3 of $B_\top$ pass through the incidences of λa_3, λa_2, λa_1 in this order. If the three names are pairwise distinct, no leaf is captured, every identifier is attached to a single formula, and (W2) holds as in the proof of Proposition 4.4. If $\nu(t_i) = \nu(t_j)$ with $i < j$, then the leaf a_i of $B_\top$ meets the incidence of λa_j before its own binder and is bound there in T_H^ν, violating (ii). ▫

Lemma 4.7 (cost).

For every admissible single-identifier naming ν of T_H,

$$|V((T_H^\nu)^\flat)| = K_H + m(\nu) + \mathrm{cut}(\nu), \qquad K_H = 4|F| + 4|U| + 7,$$

where $m(\nu)$ is the number of distinct values of ν and $\mathrm{cut}(\nu)$ is the number of edges $xy \in F$ with $\nu(x) \neq \nu(y)$.

Proof. By Theorem 3.8(c) we count the distinct subterms of T_H^ν. Subterms with different conclusion formulas are different. (1) The leaves of formula p yield one subterm per value of ν: $m(\nu)$ in total. (2) The leaves c_e, d_u, $d_\top$ yield $|F| + |U| + 1$ subterms. (3) The only subterms with conclusion $p \wedge q_e$ are the two copies of S_e, namely $\langle \nu(x), c_e \rangle$ and $\langle \nu(y), c_e \rangle$ for $e = xy$; they coincide iff $\nu(x) = \nu(y)$. This gives $|F| + \mathrm{cut}(\nu)$ subterms. (4) Every other subterm contains an occurrence of some d_u or of $d_\top$, hence its conclusion mentions r_u or $r_\top$, and these subterms have pairwise distinct conclusions: the $1 + \deg(u)$ initial segments of B_u and the term R_u mention r_u and no other atom of the form $r_{u'}$ or $r_\top$, and differ in size; likewise for the $3 + \sum_i \deg(t_i)$ initial segments of $B_\top$ of size greater than one and the three abstractions leading to $R_\top$; and the $|U|$ nodes of the top-level chain mention $r_\top$ and at least one r_u and differ in size. Their number is $\sum_{u \in U} (2 + \deg(u)) + 6 + \sum_i \deg(t_i) + |U| = 3|U| + 2|F| + 6$. Summing (2)-(4) without the term $\mathrm{cut}(\nu)$ gives K_H. ▫

Proof of Theorem 4.5, hardness. We claim that $\min_\nu |V((T_H^\nu)^\flat)| = K_H + 3 + \mathrm{mc}(H)$, the minimum being over admissible single-identifier namings; then $(H, c) \mapsto (T_H, K_H + 3 + c)$ is the required reduction. By Lemmas 4.6 and 4.7 we have to minimise $m(\nu) + \mathrm{cut}(\nu)$ subject to $\nu(t_1), \nu(t_2), \nu(t_3)$ being pairwise distinct.

"≤": from an optimal χ let $\nu(x) = a_{\chi(x)}$; then $m(\nu) = 3$ and $\mathrm{cut}(\nu) = \mathrm{mc}(H)$.

"≥": let ν be an admissible single-identifier naming, and let ν' agree with ν except that every $u \in U$ with $\nu(u) \notin \{\nu(t_1), \nu(t_2), \nu(t_3)\}$ is renamed to $\nu(t_1)$. An edge with exactly one renamed endpoint was counted in $\mathrm{cut}(\nu)$, because its other endpoint carries a terminal name; an edge with two renamed endpoints is not counted in $\mathrm{cut}(\nu')$; all other edges are unaffected. Hence $\mathrm{cut}(\nu') \leq \mathrm{cut}(\nu)$, and $m(\nu') = 3 \leq m(\nu)$. The naming ν' is admissible and induces a map χ with $\chi(t_i) = i$ and exactly $\mathrm{cut}(\nu')$ bichromatic edges, so $m(\nu) + \mathrm{cut}(\nu) \geq 3 + \mathrm{mc}(H)$.

The restrictions stated in the theorem are satisfied by T_H: its rule instances are $\mathsf{Ass}_a(A)$, $\wedge\,\mathsf{I}$ and $\Rightarrow \mathsf{I}$, every discharged formula is p, a thread passes through at most three discharging incidences, and the only leaves that suffer interference are the leaves a_1, a_2 of $B_\top$, so that exactly the three incidences of $R_\top$ are involved in interference. ▫

Example 4.8 (the two disciplines differ). Let H be the path $t_1 - u - t_2$, with t_3 isolated. Then $\mathrm{mc}(H) = 1$ and $K_H = 19$, so the optimum of Min-α-Fold[1] on T_H is $K_H + 3 + 1 = 23$. Under set-valued discharge, the leaf of u in the gadget of the edge t_1u may be named a_1 and the leaf of u in the gadget of ut_2 may be named a_2 (and the remaining leaf of u by either). This naming is admissible, since no leaf of B_u meets a discharging incidence other than its binder, and it lets both gadgets of u coincide with their copies in $B_\top$. The counting in the proof of Lemma 4.7, applied to the individual leaves, then gives $K_H + 3 = 22$ vertices; this is the optimum of Min-α-Fold on T_H, because every admissible naming of T_H uses at least three identifiers.

4.4 Discharge depth two, set-valued discharge, and the implicational fragment

Theorem 4.5 uses conjunction, three nested discharges and, by Example 4.8, the single-identifier discipline. None of the three is essential.

Theorem 4.9.

Min-α-Fold and Min-α-Fold[1] are NP-complete. Both remain NP-hard for single-identifier trees of discharge depth two over the rule instances $\mathsf{Ass}_a(A)$, $\Rightarrow \mathsf{I}$ *and* $\Rightarrow \mathsf{E}$ *of purely implicational minimal logic in which all discharged formulas are one and the same atom.*

Corollary 4.10 (dichotomy in the discharge depth).

With admissibility as in Definition 4.2, in particular under the hygiene condition (iii): for trees of discharge depth at most one, Min-α-Fold and Min-α-Fold[1] and the corresponding optimisation problems are solvable in expected linear time; for trees of discharge depth two, Min-α-Fold and Min-α-Fold[1] are NP-complete and the optimisation problems are NP-hard.

Proof. A tree of discharge depth at most one is interference-free, so Proposition 4.4 applies. The second statement is Theorem 4.9; an algorithm computing an optimal naming would decide the problem. ▫

Corollary 4.11 (no approximation scheme).

Unless $\mathsf{P} = \mathsf{NP}$, *there is a constant* $\varepsilon_0 > 0$ *such that no polynomial-time algorithm computes, for every single-identifier tree* T *of discharge depth two over the rule instances* $\mathsf{Ass}_a(A)$, $\Rightarrow \mathsf{I}$ *and* $\Rightarrow \mathsf{E}$, *an admissible naming* ν *for which* $(T^\nu)^\flat$ *has at most* $(1 + \varepsilon_0)$ *times the minimum number of*

vertices over all admissible namings of T. The same holds with admissible single-identifier namings in place of admissible namings. In particular, neither optimisation problem admits a polynomial-time approximation scheme, already on this class of trees.

The rest of this subsection contains the proofs. Call a tree *p-uniform*, for an atom p, if all its non-free leaves have formula p; by (W2), all its discharge formulas are then p.

Lemma 4.12 (admissibility in uniform trees).

Let T be p-uniform and let ν be a naming of T none of whose values is free in T. Then ν is admissible if and only if, for every non-free leaf ℓ and every discharging incidence b that the thread of ℓ meets strictly before $\beta_T(\ell)$, we have $\nu(\ell) \neq \nu(\ell')$ for all leaves ℓ' with $\beta_T(\ell') = b$.

Proof. Write $D^\nu(b) = \{\nu(\ell') \mid \beta_T(\ell') = b\}$ for the discharge set of b in T^ν. If $\nu(\ell) \in D^\nu(b)$ for an incidence b met before $\beta_T(\ell)$, then the binder of ℓ in T^ν is b or an incidence met even earlier, so (ii) fails. Conversely, assume the condition. A non-free leaf ℓ meets no incidence whose new discharge set contains $\nu(\ell)$ before $\beta_T(\ell)$, and $\nu(\ell) \in D^\nu\big(\beta_T(\ell)\big)$; a free leaf carries an identifier that is not a value of ν and hence belongs to no discharge set of T^ν. This is (ii). For (i), every identifier of T^ν is attached to a single formula: the values of ν to p, and the free identifiers to the formulas they have in T. Hence all unions in (W3) are compatible, and (W1) is not affected by renaming. For (W2), let $a \in D^\nu(b)$, say $a = \nu(\ell')$ with $\beta_T(\ell') = b$. By (ii), ℓ' has an a-open thread in T^ν to the source of b, so a belongs to the context of that source by Lemma 2.7, where it is bound to p, the discharge formula of b. ▫

The reduction. We reduce from SIMPLE MAX CUT: given a graph Q and an integer k, is there a map $\sigma\colon V(Q) \to \{0,1\}$ under which at least k edges are *cut*, that is, have endpoints with different values? This problem is NP-complete [18]. Let $n = |V(Q)|$ and $m = |E(Q)|$; we may assume that $m \geq 1$ and that Q has no isolated vertices, so that $n \leq 2m$. Write $\mathrm{maxcut}(Q)$ for the maximum number of cut edges and $\mathrm{mono}(\sigma)$ for the number of edges not cut by σ, so that $\min_\sigma \mathrm{mono}(\sigma) = m - \mathrm{maxcut}(Q)$. Fix an orientation of every edge and write $e = (u, v)$ for an edge with *tail* u and *head* v. Put $M = m$.

Fix atoms $p = \omega_0, \omega_1, \dots, \omega_M$ and q_e ($e \in E(Q)$), and free identifiers g_j and c_e with

$$g_j\colon\ \omega_{j-1} \Rightarrow \omega_j \quad (1 \leq j \leq M), \qquad c_e\colon\ \omega_M \Rightarrow q_e \quad \big(e \in E(Q)\big).$$

We use term notation: an identifier stands for the leaf that opens it; $\tau \cdot \tau'$ is $\Rightarrow \mathsf{E}$ applied to τ, with conclusion $A \Rightarrow B$, and τ', with conclusion A; and $\lambda a.\,\tau$ is $\Rightarrow \mathsf{I}(p,\cdot)$ applied to τ with discharge set $\{a\}$. For an identifier a of formula p let

$$W^0(a) = a, \qquad W^j(a) = g_j \cdot W^{j-1}(a) \quad (1 \leq j \leq M), \qquad W = W^M, \qquad S_e(a) = c_e \cdot W(a),$$

so that $W^j(a)$ has conclusion ω_j and $S_e(a)$ has conclusion q_e. Several derivations are combined into one by *aggregation*: for terms $\tau_1, \dots, \tau_s$ with conclusions $\varphi_1, \dots, \varphi_s$ and a tag x, take fresh atoms $r_1^x, \dots, r_s^x$ and fresh free identifiers $d_1^x\colon \varphi_1 \Rightarrow r_1^x$ and $d_k^x\colon r_{k-1}^x \Rightarrow (\varphi_k \Rightarrow r_k^x)$ for $2 \leq k \leq s$, and let

$$\Sigma^x(\tau_1) = d_1^x \cdot \tau_1, \qquad \Sigma^x(\tau_1, \dots, \tau_k) = \big(d_k^x \cdot \Sigma^x(\tau_1, \dots, \tau_{k-1})\big) \cdot \tau_k.$$

The term $\Sigma^x(\tau_1, \dots, \tau_s)$ has conclusion r_s^x and consists, apart from the τ_k and the s leaves d_k^x, of $2s-1$ vertices, whose conclusions r_k^x and $\varphi_k \Rightarrow r_k^x$ occur nowhere else.

For a vertex v of Q with outgoing edges $e_1, \dots, e_a$ (those with tail v) and incoming edges $f_1, \dots, f_b$ (those with head v), fix two identifiers a_v, b_v and let

$$\begin{aligned} Y_v &= \Sigma^v\left(W(a_v),\, W(b_v),\, S_{e_1}(a_v), \dots, S_{e_a}(a_v),\, S_{f_1}(b_v), \dots, S_{f_b}(b_v)\right), \\ R_v &= \lambda a_v.\, \lambda b_v.\, Y_v. \end{aligned}$$

Finally, with $V(Q) = \{v_1, \dots, v_n\}$, let

$$T_Q = \Sigma^{\top}\left(R_{v_1}, \dots, R_{v_n}\right).$$

Figure 2 shows T_Q for a graph with a single edge. T_Q is a single-identifier tree-shaped TCIH with $O\big(m(n+m)\big)$ vertices. All formulas occurring in it have constant size, so its encoding size $\| T_Q \|$ is of the same order, and its labelled size is polynomial as well. Its rule instances are of the forms $\mathsf{Ass}_a(A)$, $\Rightarrow \mathsf{I}$ and $\Rightarrow \mathsf{E}$, and all its formulas are implicational. It is p-uniform. Its free identifiers are the g_j, the c_e and the aggregation identifiers d_k^x; its non-free leaves are the occurrences of the a_v and b_v. Each vertex v of Q contributes two discharging incidences: the *inner* one, of λb_v, at which the $1+b$ leaves b_v of Y_v are bound, and the *outer* one, of λa_v, at which the $1+a$ leaves a_v are bound. We call these leaves the *inner* and the *outer leaves* of v. An outer leaf of v meets the inner incidence of v before its binder, and no other non-free leaf meets a discharging incidence before its binder; in particular the discharge depth of T_Q is two. Every edge $e = (u, v)$ contributes two copies of the gadget S_e: one in Y_u, on an outer leaf of u, and one in Y_v, on an inner leaf of v.

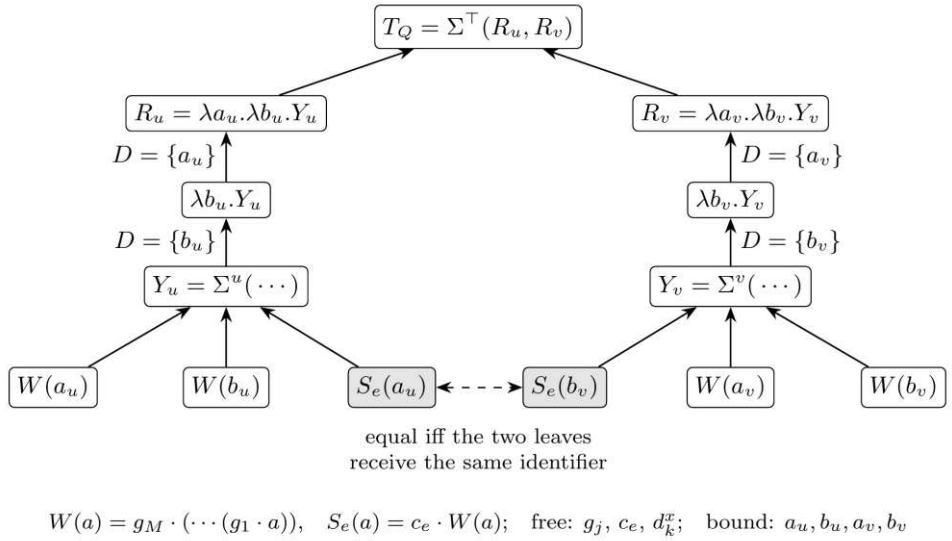

Fig. 2 The tree T_Q for a graph Q with a single edge $e = (u, v)$, drawn schematically with premises below conclusions; the aggregation vertices and the leaves d_k^x are not shown. The outer leaves of a vertex (identifier a) pass through its inner discharging incidence and must be named differently from its inner leaves (identifier b). The two shaded copies of the gadget S_e coincide

in the maximal fold if and only if the outer leaf of u and the inner leaf of v occurring in them receive the same identifier

For a naming ν of T_Q let $O_\nu(\nu)$ and $I_\nu(\nu)$ be the sets of names of the outer and of the inner leaves of v, let $m(\nu)$ be the number of distinct values of ν, and let $\mathrm{unc}(\nu)$ be the number of edges e for which the leaves of the two copies of S_e receive different names.

Lemma 4.13.

A naming ν of T_Q none of whose values is free in T_Q is admissible if and only if $O_\nu(\nu) \cap I_\nu(\nu) = \emptyset$ for every vertex v of Q.

Proof. By Lemma 4.12, since the only discharging incidences met by non-free leaves before their binders are the inner incidences, each met by the outer leaves of the same vertex. ▫

Lemma 4.14 (cost).

For every admissible naming ν of T_Q,

$$|V\left(\left(T_Q^\nu\right)^\flat\right)| = K_Q + (M+1)\,m(\nu) + \mathrm{unc}(\nu), \qquad K_Q = M + 8m + 10n - 1.$$

Proof. By Theorem 3.8(c) we count the distinct subterms of T_Q^ν; subterms with different conclusions are different. (1) The free leaves: M leaves g_j, m leaves c_e, and $\sum_v (\deg(v)+2) + n = 2m + 3n$ leaves d_k^x; in total $M + 3m + 3n$ subterms. (2) For $0 \le j \le M$, the subterms with conclusion ω_j are the terms $W^j(x)$, where x ranges over the values of ν, because every non-free leaf occurs inside a copy of W; this gives $(M+1)\,m(\nu)$ subterms. (3) The subterms with conclusion q_e are the two copies $c_e \cdot W(x)$ and $c_e \cdot W(y)$ of S_e, where x, y are the names of their leaves; they coincide if and only if $x = y$. This gives $m + \mathrm{unc}(\nu)$ subterms. (4) The remaining positions of the tree are the $\sum_v (2\deg(v)+3) = 4m + 3n$ aggregation vertices of the terms Y_v, the $2n$ abstractions, with conclusions $p \Rightarrow r_s^v$ and $p \Rightarrow (p \Rightarrow r_s^v)$ for the last atom r_s^v of Y_v, and the $2n-1$ aggregation vertices of $\Sigma^\top$. Their conclusions contain aggregation atoms, are pairwise distinct (note that $\varphi_k \neq p$ in every aggregation, as $M \ge 1$), and each occurs at exactly one position, so these positions contribute $4m + 7n - 1$ subterms whatever ν is. The sum of (1), (4) and the term m of (3) is K_Q; Table 3 summarises the count. ▫

Table 3 Numbers of distinct subterms in the four groups of the proof of Lemma 4.14, with $K_Q = M + 8m + 10n - 1$

subterms of T_Q^ν	conclusion	number
(1) free leaves g_j, c_e, d_k^x	their formulas	$M + 3m + 3n$
(2) $W^j(x)$, $0 \le j \le M$, x a value of ν	ω_j	$(M+1)\,m(\nu)$
(3) copies of the gadgets S_e	q_e	$m + \mathrm{unc}(\nu)$
(4) aggregation vertices, abstractions	with an aggregation atom	$4m + 7n - 1$

subterms of T_Q^ν	conclusion	number
all subterms		$K_Q + (M+1)\,m(\nu) + \mathrm{unc}(\nu)$

Proof of Theorem 4.9. Membership in NP was shown in Section 4.3. For hardness we claim that, over the admissible namings of T_Q as well as over the admissible single-identifier namings,

$$\min_\nu |V\left(\left(T_Q^\nu\right)^\flat\right)| = K_Q + 2(M+1) + m - \mathrm{maxcut}(Q).$$

Then $(Q,k) \mapsto \left(T_Q,\ K_Q + 2(M+1) + m - k\right)$ is a polynomial-time reduction to both problems, and T_Q has the properties stated in the theorem.

Let ν be admissible. By Lemma 4.13, $O_\nu(\nu)$ and $I_\nu(\nu)$ are disjoint, and they are non-empty, so $m(\nu) \geq 2$. If $m(\nu) \geq 3$, then by Lemma 4.14 the size is at least $K_Q + 3(M+1) > K_Q + 2(M+1) + m$, because $M + 1 > m$. If $m(\nu) = 2$, say with values x and y, then for every vertex v the sets $O_\nu(\nu)$ and $I_\nu(\nu)$ are complementary singletons; let $\sigma(v) = 0$ if $O_\nu(\nu) = \{x\}$ and $\sigma(v) = 1$ otherwise. For an edge $e = (u,v)$, the leaf of S_e in Y_u is an outer leaf of u and the leaf of S_e in Y_v is an inner leaf of v; their names differ if and only if $O_u(\nu) \neq I_\nu(\nu)$, that is, if and only if $\sigma(u) = \sigma(v)$. Hence $\mathrm{unc}(\nu) = \mathrm{mono}(\sigma) \geq m - \mathrm{maxcut}(Q)$, which proves "≥". For "≤", let σ cut $\mathrm{maxcut}(Q)$ edges, choose two identifiers x, y that are not free in T_Q, and name the outer leaves of v by x and the inner leaves by y if $\sigma(v) = 0$, and conversely if $\sigma(v) = 1$. This single-identifier naming is admissible by Lemma 4.13, has $m(\nu) = 2$ and $\mathrm{unc}(\nu) = \mathrm{mono}(\sigma) = m - \mathrm{maxcut}(Q)$. ▫

Proof of Corollary 4.11. Let $B = K_Q + 2(M+1) = 11m + 10n + 1$, so that $B \leq 32m$, and let $\mu = m - \mathrm{maxcut}(Q)$. By the claim in the preceding proof, the optimum on T_Q is $B + \mu \leq 33m$ for both problems. Suppose that an algorithm returns an admissible naming ν of T_Q with $s = \left|V\left(\left(T_Q^\nu\right)^\flat\right)\right| \leq (1+\varepsilon)(B+\mu)$. If $m(\nu) = 2$, let σ be the map obtained from ν as in the preceding proof; then $\mathrm{mono}(\sigma) = s - B$. Otherwise let σ be arbitrary; then $s - B \geq M + 1 > m \geq \mathrm{mono}(\sigma)$. In both cases

$$\mathrm{mono}(\sigma) \leq s - B \leq \mu + \varepsilon(B+\mu) \leq \mu + 33\,\varepsilon m,$$

so σ cuts at least $\mathrm{maxcut}(Q) - 33\,\varepsilon m \geq (1 - 66\,\varepsilon)\mathrm{maxcut}(Q)$ edges, because $\mathrm{maxcut}(Q) \geq m/2$. A polynomial-time approximation scheme for Min-α-Fold or for Min-α-Fold[1] would therefore yield one for MAX CUT on simple graphs. As MAX CUT is MAX SNP-hard [28], this is impossible unless P = NP [3]. ▫

Remark 4.15 (variants and limits of the results). (1) The sharing that makes the problem hard is the cross-scope sharing of Example 4.1: in both reductions the two copies of a gadget S_e lie under different binders. If sharing were restricted to subderivations without open bound identifiers, the choice of names would be irrelevant up to α-equivalence of closed subterms, which can be decided by hashing modulo α [25]; if it were restricted to subderivations whose open bound identifiers are discharged by the same, or by bisimilar, inferences, it would amount to context-sensitive α-equivalence, which is likewise decidable by hashing [6]. (2) Proposition 4.4 and Theorems 4.5 and 4.9 delimit the problem from several sides. Without interference a

canonical naming is optimal. With interference, hardness sets in immediately in two respects: at discharge depth two, for both disciplines (Theorem 4.9), and, under the single-identifier discipline, when only three discharging incidences are involved in interference (Theorem 4.5). In particular, Min-α-Fold[1] is NP-hard for a fixed value of each of the two parameters, the discharge depth and the number of incidences involved in interference; unless $\mathrm{P} = \mathrm{NP}$ it is therefore not fixed-parameter tractable, and not even solvable in polynomial time for each fixed parameter value, with respect to either of them. For Min-α-Fold the same holds with respect to the discharge depth. (3) Condition (iii) of Definition 4.2 matters. If it is dropped, so that a discharged assumption may reuse the identifier of a free assumption occurring elsewhere, the sharing excluded in Example 3.5 becomes available, and the hardness of Min-α-Fold[1] already holds for trees without nested discharges, that is, at discharge depth one: in the construction of Theorem 4.5, three free assumptions g_1, g_2, g_3 of p take the role of the terminals, the gadgets $S_e(g_i)$ are placed at top level, and the same counting gives the optimum $K'_H + \mathrm{mc}(H)$ for a constant K'_H depending only on H. The dichotomy of Corollary 4.10 therefore depends on (iii). (4) Three questions remain open. We do not know whether Min-α-Fold remains NP-hard when the number of incidences involved in interference is bounded; on the trees T_H it is trivial, since every gadget can be made to coincide with its copy as in Example 4.8, and the trees T_Q contain $2n$ such incidences. We do not know whether Min-α-Fold[1] is solvable in polynomial time when exactly two incidences are involved in interference. Finally, Corollary 4.11 excludes an approximation scheme, but we know of no polynomial-time algorithm with a constant approximation ratio for either problem.

5 Judgment-level and formula-level sharing

5.1 Contexts as certificates for the thread condition

In the dag-like natural deductions of Gordeev and Haeusler [22], vertices carry formulas only, and a dag proves its root formula if it is locally correct and every maximal thread is closed, that is, passes through an inference discharging the assumption it starts from. This condition quantifies over threads, of which a dag may have exponentially many: in the skeleton with a leaf v_0 and v_j obtained by $\wedge\,\mathsf{I}$ from $\left(v_{j-1}, v_{j-1}\right)$, there are 2^n threads from v_0 to v_n. Lemma 2.7 relates the two formalisms.

Corollary 5.1.

Let G be an event structure over $\mathcal{R}$ satisfying (U) and (A). Whether G is a TCIH is decided within the time bound of Proposition 2.5 by computing the labels bottom-up and checking (W1)-(W3). In a TCIH, a vertex r has empty context if and only if no leaf $\mathsf{Ass}_a(A)$ has an a-open thread to r.

In other words, a TCIH is a deterministic dag-like deduction (one justification per vertex, discharge by identifiers) in which every vertex is annotated with the set of assumptions having an open thread to it; the annotation is unique, locally checkable, and replaces the quantification over threads. Two differences from the setting of [22] should be kept in mind. First, discharge there is by formula ($\Rightarrow\,\mathsf{I}$ with conclusion $A \Rightarrow B$ closes all threads starting from A above it), which corresponds to using one identifier per formula. Secondly, and more importantly, the result of horizontal compression is in general not deterministic: merging all nodes with the same

formula on the same level produces nodes with several alternative premises, and the compressed object has to be accompanied by a coherent selection of threads. Theorem 3.3 explains why this cannot be avoided: an equivalence that merges vertices with equal formulas and different contexts or justifications is never a justification congruence, so its quotient is never a TCIH.

5.2 A separation in the number of vertices, relative to a fixed derivation

As means of compressing a given derivation, level-wise merging of formulas and admissible folding are incomparable. The *level* of a vertex of a tree is its distance from the root.

Example 5.2 (neither discipline dominates). Let $\tau = \wedge\,\mathsf{I}\langle \mathsf{Ass}_a(p), \wedge\,\mathsf{I}\langle \mathsf{Ass}_a(p), \mathsf{Ass}_a(p)\rangle\rangle$. The maximal fold of τ has three vertices, with formulas p, $p \wedge p$ and $p \wedge (p \wedge p)$, because folding shares equal subderivations wherever they occur. The quotient of τ by equality of level and formula has four classes, since the formula p occurs on levels 1 and 2. Conversely, on the trees Π_n defined below the level-wise formula quotient is exponentially smaller than the maximal fold.

Gordeev and Haeusler remark that merging equal sequents instead of equal formulas would not suffice for their purposes [23]; the following proposition makes a corresponding statement precise for TCIHs, in terms of the number of vertices.

Let $p, q_1, \dots, q_n$ be atoms and $h, k_1, \dots, k_n$ identifiers. For terms τ, τ' with conclusions A, B let $\tau * \tau' = \wedge\,\mathsf{E}_1(A,B)\langle \wedge\,\mathsf{I}(A,B)\langle \tau, \tau'\rangle\rangle$, a derivation of A whose context is the union of those of τ and τ'. For $S \subseteq \{1, \dots, n\}$ define $X_0^S = \mathsf{Ass}_h(p)$ and

$$X_i^S = \begin{cases} X_{i-1}^S * \mathsf{Ass}_{k_i}(q_i) & \text{if } i \in S, \\ X_{i-1}^S * \mathsf{Ass}_h(p) & \text{if } i \notin S, \end{cases} \qquad 1 \le i \le n,$$

so that X_n^S has label $\{h\colon p\} \cup \{k_i\colon q_i \mid i \in S\} \vdash p$. Let Π_n be obtained by combining the 2^n terms X_n^S in a complete binary tree of depth n whose inner nodes are applications of $*$. To fix the tree, the terms X_n^S are arranged from left to right in increasing order of $\sum_{i \in S} 2^{i-1}$, and an inner node applies $*$ to its left and its right subtree, in this order.

Proposition 5.3 (separation).

For every $n \ge 1$:

1. *Π_n is a tree-shaped TCIH without discharge, with $(3n+1)2^n + 2^{n+1} - 2$ vertices and root label $\{h\colon p, k_1\colon q_1, \dots, k_n\colon q_n\} \vdash p$;*
2. *the vertices of Π_n fall into exactly $6n+1$ classes of equal level and equal formula;*
3. *every TCIH (G, r) with $\mathfrak{u}(r) = \Pi_n$ has at least 2^n vertices, and Π_n has no α-variant other than itself.*

Proof. (a) Each X_n^S has $3n+1$ vertices, and the combining tree adds two vertices for each of its $2^n - 1$ inner nodes. (b) The combining tree occupies levels $0, \dots, 2n$, with formula p on even and $p \wedge p$ on odd levels: $2n+1$ classes. Within each X_n^S, for $1 \le i \le n$, the $\wedge\,\mathsf{I}$ vertex of step i lies on level $4n - 2i + 1$ and has formula $p \wedge q_i$ or $p \wedge p$, and its two sources lie on level $4n - 2i + 2$ and have formula p or q_i: four classes for each i, all on levels greater than $2n$. (c) The 2^n

subterms X_n^S have pairwise distinct contexts, hence are pairwise distinct, and by Theorem 3.8(c) every TCIH with unfolding Π_n has at least as many vertices as Π_n has distinct subterms. There is no discharging incidence, so there is nothing to rename. ▫

Three qualifications delimit what the proposition says. First, the separation is exponential in n and with respect to the size of the formula quotient, not with respect to the size of the input tree: Π_n has $\Theta(n2^n)$ vertices and its maximal fold has $\Theta(2^n)$ (Table 4), a gain of a factor of order n, whereas the formula quotient has $6n + 1 = O(\log|V(\Pi_n)|)$ classes. Secondly, the comparison is relative to the fixed derivation Π_n. The end judgment itself has the short derivation $X_n^{\{1,\ldots,n\}}$ with $3n + 1$ vertices, to which Π_n can be reduced by pruning (Proposition 3.10) but not by folding; the proposition gives no lower bound on the size of proofs of a judgment. Thirdly, the proposition compares numbers of vertices and not complete proof certificates. The level-wise formula quotient is not a TCIH, some of its nodes having two alternative justifications, and to serve as a proof in the sense of [22] it has to be accompanied by a coherent selection of threads, whose size is not accounted for here. A comparison of certificate sizes would require a format, a correctness condition and an encoding to be fixed on both sides, for instance within a general framework for proof certificates [13].

With these qualifications, the proposition quantifies the trade-off between the two disciplines: judgment labels make correctness local and deterministic, at the cost of separating occurrences of a formula that depend on different assumptions; formula-level merging identifies such occurrences, at the cost of a global correctness condition and of nondeterministic justifications.

Remark 5.4 (an implicational variant). The trees Π_n use conjunction, whereas the deductions of [22] are purely implicational. The separation does not depend on this. Replace $\tau * \tau'$ by $(\kappa_B \cdot \tau) \cdot \tau'$, where B is the conclusion of τ', $\cdot$ denotes $\Rightarrow \mathsf{E}$, and κ_B is a free assumption of the formula $p \Rightarrow (B \Rightarrow p)$. The resulting trees ${\Pi'}_n$ use the rule instances $\mathsf{Ass}_a(A)$ and $\Rightarrow \mathsf{E}$ only. The 2^n subterms that correspond to the X_n^S still have pairwise distinct contexts, so every TCIH with unfolding ${\Pi'}_n$ has at least 2^n vertices. On the other hand ${\Pi'}_n$ has height at most $4n$, and only the $3n + 3$ formulas p, $p \Rightarrow p$, $p \Rightarrow (p \Rightarrow p)$, q_i, $q_i \Rightarrow p$ and $p \Rightarrow (q_i \Rightarrow p)$, $1 \leq i \leq n$, occur in it, so that its vertices fall into at most $(4n + 1)(3n + 3)$ classes of equal level and formula. The bound is quadratic rather than linear because the levels are no longer aligned as in Π_n. The separation is therefore exponential in n relative to a quotient whose size is polynomial in n; measured in the number q of classes, the lower bound 2^n is $2^{\Omega(\sqrt{q})}$ for ${\Pi'}_n$, whereas it is $2^{\Omega(q)}$ for Π_n, where $q = 6n + 1$.

5.3 Proof graphs with boxes

In the deduction graphs of Geuvers and Loeb [20] the scope of a discharged assumption is delimited by a box. Edges point from a node to the nodes it is inferred from, a box is a set of nodes with an associated *box node*, and the boxes are subject to the following conditions:

(B1)
two boxes are disjoint or one is contained in the other;

(B2)
the only edge leaving a box node points into its box;

(B3)
apart from this edge, no edge points into a box.

Moreover, (B0) every box has its own box node, which does not belong to the box (the latter follows from the box-topological ordering required of deduction graphs), and (B4) the hypotheses discharged at a box node, in an implication introduction, are nodes of its box. The following observation uses only these five properties; should a variant of the formalism not satisfy one of them, it does not apply.

Proposition 5.5.

In an acyclic graph with boxes satisfying (B0)-(B4), let a hypothesis node h be discharged at two box nodes $b \neq b'$, and let c and c' be the nodes to which b and b' point. Then h is not reachable both from c and from c' by paths that contain no box node.

Proof. We first note that if a node n reaches a node of a box B by a path that does not contain the box node of B, then $n \in B$: otherwise the path contains an edge from a node outside B to a node inside B, which by (B3) starts at the box node of B. Let B and B' be the boxes of b and b'. By (B4), $h \in B \cap B'$, and $B \neq B'$ by (B0), so by (B1) one box is properly contained in the other; by symmetry let $B \subsetneq B'$. Suppose h is reachable from c' by a path without box nodes. Then $c' \in B$ by the first remark. The edge from b' to c' thus points into B, and $b' \neq b$, so $b' \in B$ by the first remark again, whence $b' \in B'$, contradicting (B0). ▫

The sharing of Example 4.1 is of the excluded kind. There the assumption leaf a_1 is discharged by two implication introductions, with premises $\sigma(a_1)$ and $\wedge\mathsf{I}\langle\sigma(a_1), \mathsf{Ass}_d(r)\rangle$, and it is reachable from both premises through the shared vertex $\sigma(a_1)$ without passing through a discharging inference. Consider a graph with boxes that represents this derivation inference by inference, with a box node for each of the two implication introductions and with no further box nodes on the paths from their premises to the hypotheses. By Proposition 5.5 it cannot contain a single node for $\sigma(a_1)$ from which one hypothesis node discharged at both box nodes is reached; since that node has a single edge to a hypothesis of p, the node for $\sigma(a_1)$ has to be duplicated, once for each discharge. A TCIH, by contrast, may share it after renaming, and Theorems 4.5 and 4.9 show that exploiting this additional sharing optimally is intractable. Proposition 5.5 excludes this particular form of sharing under (B0)-(B4). It is not a translation between the two formalisms, and it says nothing about the cost of other ways of representing TCIHs by graphs with boxes; a precise translation, and in particular a characterisation of the TCIHs that arise from box structures, remains to be given.

6 Computational validation

The statements above are elementary enough to be tested mechanically, and we did so as a safeguard against errors in the definitions and in the counting arguments. The script tcih_validate.py (Online Resource 1) depends only on the Python standard library and uses a fixed random seed. It implements skeletons, the labelling (W1)-(W3), quotients, hash-consing, threads and namings. Well-formedness is always decided directly from (W1)-(W3), and admissibility of a naming directly from Definition 4.2; the criteria derived in this paper (justification congruence, Lemma 4.13, the cost formulas) are implemented separately and compared with these direct

computations. The following checks are performed; all of them pass, and the expected output is part of Online Resource 1.

- *Theorem 3.3 and Theorem 3.8.* For 60 proof objects with 4 to 9 vertices (three hand-built ones, including Example 3.5 and a ∨ E derivation, and 57 random ones over $\{\wedge, \Rightarrow\}$), *all* set partitions of the vertex set were enumerated. For each partition, the quotient was tested directly for being a TCIH (unique justification, acyclicity, existence of a coherent labelling) and, independently, the partition was tested for being a justification congruence. The two tests agreed on every partition; 167 admissible foldings were found in total, labels were constant on their classes, and the smallest admissible quotient always had the size computed by hash-consing.

- *Lemma 2.7.* On the same objects, the contexts computed by (W3) coincide with the sets of identifiers obtained by forward reachability along open threads.

- *Examples.* The claims of Example 3.5 (the quotient violates (U); keeping $\mathsf{Ass}_b(A)$ violates (W2); keeping $\mathsf{Ass}_a(A)$ gives a TCIH with root context $\{a\colon A\}$; renaming a to b permits the fold but violates (iii)), the hygiene example of Section 4, the derivations discussed at the beginning of Section 4.2 with their discharge depths, and Example 5.2 were checked individually.

- *Lemmas 4.6, 4.7 and Theorem 4.5.* For 41 random graphs with three terminals and one to four further vertices, the tree T_H was built, all single-identifier namings over a pool of three terminal names and $|U|$ further names were enumerated, the cost formula $K_H + m(v) + \mathrm{cut}(v)$ with $K_H = 4|F| + 4|U| + 7$ was confirmed for each, and the optimum was confirmed to equal $K_H + 3 + \mathrm{mc}(H)$, with $\mathrm{mc}(H)$ computed by brute force. Namings with coinciding terminal names were confirmed to be inadmissible. The variant of Remark 4.15(3) was checked in the same way.

- *Example 4.8.* For the path $t_1 - u - t_2$, all 4^8 namings of the eight non-free leaves of T_H with values among four identifiers were enumerated. The optimum is 23 over the single-identifier namings and 22 over all namings.

- *Lemmas 4.12, 4.13, 4.14 and Theorem 4.9.* For three graphs (a single edge, a path with two edges, and a triangle) the tree T_Q was built and checked to use only the rule instances $\mathsf{Ass}_a(A)$, ⇒ I, ⇒ E and to have discharge depth two. All namings of its non-free leaves over three identifiers (two for the triangle) were enumerated: 729, 59 049 and 4096 namings, 63 874 in total. For every one of them, admissibility was decided from Definition 4.2 and compared with the disjointness condition of Lemma 4.13; the two agree throughout, and 144, 2592 and 8 namings, 2744 in total, are admissible. For each admissible naming the cost formula $K_Q + (M+1)\,m(v) + \mathrm{unc}(v)$ with $K_Q = M + 8m + 10n - 1$ was confirmed, and the optimum, over all namings and over the single-identifier namings, was confirmed to equal $K_Q + 2(M+1) + m - \mathrm{maxcut}(Q)$, with $\mathrm{maxcut}(Q)$ computed by brute force; the optima are 32, 53 and 65 vertices. For seven random graphs with three to five vertices, 250 random namings each over four identifiers (1337 of them admissible) confirmed Lemma 4.13 and the cost formula.

- *Proposition 4.4.* On 40 random interference-free trees with two to five discharging incidences, the canonical naming was admissible, no admissible single-identifier naming over a pool of as many names as there are binders gave a smaller maximal fold, and neither did any admissible naming of the non-free leaves over three identifiers.

- *Proposition 5.3 and Remark 5.4.* Table 4 reports the measured sizes for $n \leq 6$. For the implicational variant Π'_n, the numbers of classes of equal level and formula are 12, 35, 62, 95, 134 and 179 for $n = 1, \ldots, 6$, within the bound $(4n + 1)(3n + 3)$, and the numbers of distinct judgments are at least 2^n.

Table 4 The separation family of Proposition 5.3: number of vertices of the tree and of its maximal fold, number of distinct judgments, and number of classes under level-wise formula merging $(6n + 1)$. The number of distinct judgments depends on the order in which the terms X_n^S are combined; the values are for the order fixed before Proposition 5.3

n	$\lvert V(\Pi_n)\rvert$	$\lvert V(\Pi_n^{\flat})\rvert$	distinct judgments	(level, formula) classes
1	10	8	6	7
2	34	21	12	13
3	94	46	24	19
4	238	95	47	25
5	574	192	92	31
6	1342	385	181	37

These computations are consistency checks of the examples, constructions and counting formulas on small instances. They do not verify the general statements, in particular not the hardness results, and are not a substitute for the proofs. A formalisation of Sections 2-4 in a proof assistant is left for future work.

7 Conclusion and open problems

For proof graphs whose vertices are judgments with identifier-labelled contexts, sharing with fixed identifiers is governed by a single condition, justification congruence. Label preservation and acyclicity are consequences, and the maximal fold is canonical and computable in expected linear time on the skeleton. The only source of computational hardness in identity-preserving sharing is the choice of names for discharged assumptions. Under the convention that bound identifiers differ from free ones, this choice is trivial in the absence of interference, in particular at discharge depth one. At discharge depth two the decision problem is NP-complete and the optimisation problem admits no approximation scheme unless $\mathrm{P} = \mathrm{NP}$, for single-identifier and for set-valued discharge alike, already in purely implicational minimal logic. Relative to formula-level merging, the judgment discipline exchanges succinctness for locality, and on a fixed derivation the difference in the number of vertices can be exponential.

Several questions remain open. (1) For the renaming problem: whether a constant approximation ratio is achievable; whether Min-α-Fold remains hard when boundedly many discharging incidences are involved in interference; and whether Min-α-Fold[1] is polynomial when exactly

two are (Remark 4.15(4)). (2) The complexity of finding a smallest TCIH that can be obtained from a given one by pruning (Proposition 3.10), a problem about the inferences present in the given graph rather than about shortest proofs in the logic, and the relation of pruning to proof simplification and compression procedures [2, 7, 9]. (3) An extension to first-order natural deduction, where eigenvariables are a second kind of bound name and the eigenvariable condition a second kind of scope; we expect Theorem 3.3 to carry over once vertices also record the free variables in force, and the renaming problem to become harder. (4) A common framework in which admissible foldings, nondeterministic formula-level quotients and box-based graphs can be compared by translations rather than by examples and by Propositions 5.3 and 5.5. (5) An evaluation of maximal folds, with and without renaming, on natural-deduction proof objects exported from proof assistants or produced by proof-reconstruction tools.

Statements and Declarations

Funding. This research was supported by the initial scientific and technological project of the Institute of Information Technology (IOIT), Vietnam Academy of Science and Technology (VAST), “Training a large language model integrating formal logical thinking for solving general mathematics problems,” grant number NVKN02.05/26-27.

Competing interests. The author has no competing interests to declare that are relevant to the content of this article.

Data availability. No external datasets were used. The test instances of Section 6 (proof objects, graphs and namings) are generated by the validation script from a fixed random seed, and the results reported there are reproduced by running it.

Code availability. The validation script, instructions for running it, and its expected output are provided as Online Resource 1. The script depends only on the Python standard library.

Use of large language models. We use ChatGPT Pro to translate from Vietnamese to English.

References

[1] Amjad, H.: Data compression for proof replay. J. Autom. Reason. **41**, 193-218 (2008). https://doi.org/10.1007/s10817-008-9109-2

[2] Arkoudas, K.: Simplifying proofs in Fitch-style natural deduction systems. J. Autom. Reason. **34**(3), 239-294 (2005). https://doi.org/10.1007/s10817-005-9000-3

[3] Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. J. ACM **45**(3), 501-555 (1998)

[4] Barbosa, H., Blanchette, J.C., Fleury, M., Fontaine, P.: Scalable fine-grained proofs for formula processing. J. Autom. Reason. **64**(3), 485-510 (2020). https://doi.org/10.1007/s10817-018-09502-y

[5] Barendregt, H.P., van Eekelen, M.C.J.D., Glauert, J.R.W., Kennaway, J.R., Plasmeijer, M.J., Sleep, M.R.: Term graph rewriting. In: PARLE, Parallel Architectures and Languages Europe, LNCS, vol. 259, pp. 141-158. Springer (1987)

[6] Blaauwbroek, L., Olšák, M., Geuvers, H.: Hashing modulo context-sensitive α-equivalence. Proc. ACM Program. Lang. **8**(PLDI), 229, 2027-2050 (2024). https://doi.org/10.1145/3656459

[7] Blanchette, J.C., Böhme, S., Fleury, M., Smolka, S.J., Steckermeier, A.: Semi-intelligible Isar proofs from machine-generated proofs. J. Autom. Reason. **56**(2), 155-200 (2016). https://doi.org/10.1007/s10817-015-9335-3

[8] Bouchez, F., Darte, A., Rastello, F.: On the complexity of register coalescing. In: International Symposium on Code Generation and Optimization (CGO), pp. 102-114. IEEE (2007)

[9] Boudou, J., Fellner, A., Woltzenlogel Paleo, B.: Skeptik: a proof compression system. In: IJCAR 2014, LNCS, vol. 8562, pp. 374-380. Springer (2014)

[10] Braibant, T., Jourdan, J.-H., Monniaux, D.: Implementing and reasoning about hash-consed data structures in Coq. J. Autom. Reason. **53**, 271-304 (2014). https://doi.org/10.1007/s10817-014-9306-0

[11] Charguéraud, A.: The locally nameless representation. J. Autom. Reason. **49**(3), 363-408 (2012). https://doi.org/10.1007/s10817-011-9225-2

[12] Chavrimootoo, M.C., Ferland, E., Gibson, E., Wilson, A.H.: A closer look at some recent proof compression-related claims. arXiv:2212.12150 (2022)

[13] Chihani, Z., Miller, D., Renaud, F.: A semantic framework for proof evidence. J. Autom. Reason. **59**(3), 287-330 (2017). https://doi.org/10.1007/s10817-016-9380-6

[14] Condoluci, A., Accattoli, B., Sacerdoti Coen, C.: Sharing equality is linear. In: PPDP 2019, pp. 9:1-9:14. ACM (2019). https://doi.org/10.1145/3354166.3354174

[15] Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**(4), 864-894 (1994)

[16] Dovier, A., Piazza, C., Policriti, A.: An efficient algorithm for computing bisimulation equivalence. Theor. Comput. Sci. **311**, 221-256 (2004)

[17] Downey, P.J., Sethi, R., Tarjan, R.E.: Variations on the common subexpression problem. J. ACM **27**(4), 758-771 (1980). https://doi.org/10.1145/322217.322228

[18] Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theor. Comput. Sci. **1**(3), 237-267 (1976)

[19] Gentzen, G.: Untersuchungen über das logische Schließen. I, II. Math. Z. **39**, 176-210, 405-431 (1935)

[20] Geuvers, H., Loeb, I.: Natural deduction via graphs: formal definition and computation rules. Math. Struct. Comput. Sci. **17**(3), 485-526 (2007). https://doi.org/10.1017/S0960129507006123

[21] Girard, J.-Y.: Linear logic. Theor. Comput. Sci. **50**(1), 1-101 (1987)

[22] Gordeev, L., Haeusler, E.H.: Proof compression and NP versus PSPACE. Stud. Log. **107**(1), 53-83 (2019)

[23] Gordeev, L., Haeusler, E.H.: Proof compression and NP versus PSPACE II. Bull. Sect. Log. **49**(3), 213-230 (2020)

[24] Grabmayer, C., Rochel, J.: Maximal sharing in the lambda calculus with letrec. In: ICFP 2014, pp. 67-80. ACM (2014). https://doi.org/10.1145/2628136.2628148

[25] Maziarz, K., Ellis, T., Lawrence, A., Fitzgibbon, A., Peyton Jones, S.: Hashing modulo alpha-equivalence. In: PLDI 2021, pp. 960-973. ACM (2021). https://doi.org/10.1145/3453483.3454088

[26] Michaelson, D., Schreiber, D., Heule, M.J.H., Kiesl-Reiter, B., Whalen, M.W.: Producing proofs of unsatisfiability with distributed clause-sharing SAT solvers. J. Autom. Reason. **69**(2), 12 (2025). https://doi.org/10.1007/s10817-025-09725-w

[27] de Oliveira, A.G., de Queiroz, R.J.G.B.: Geometry of deduction via graphs of proofs. In: de Queiroz, R.J.G.B. (ed.) Logic for Concurrency and Synchronisation, pp. 3-88. Kluwer (2003)

[28] Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**(3), 425-440 (1991)

[29] Prawitz, D.: Natural Deduction: A Proof-Theoretical Study. Almqvist & Wiksell, Stockholm (1965)

[30] Quispe-Cruz, M., Haeusler, E.H., Gordeev, L.: Proof-graphs for minimal implicational logic. In: DCM 2013, EPTCS, vol. 144, pp. 16-29 (2014)

[31] Roßkopf, S., Nipkow, T.: A formalization and proof checker for Isabelle's metalogic. J. Autom. Reason. **67**, 1 (2023). https://doi.org/10.1007/s10817-022-09648-w

[32] Sieg, W., Derakhshan, F.: Human-centered automated proof search. J. Autom. Reason. **65**, 1153-1190 (2021). https://doi.org/10.1007/s10817-021-09594-z

[33] Sørensen, M.H., Urzyczyn, P.: Lectures on the Curry-Howard Isomorphism. Elsevier (2006)

[34] Statman, R.: Structural Complexity of Proofs. PhD thesis, Stanford University (1974)

[35] Troelstra, A.S., Schwichtenberg, H.: Basic Proof Theory, 2nd edn. Cambridge University Press (2000)

[36] Wernhard, C., Bibel, W.: Investigations into proof structures. J. Autom. Reason. **68**, 24 (2024). https://doi.org/10.1007/s10817-024-09711-8

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- TCIH.zip