Classical Proof Forestry

Willem Heijltjes

LFCS
School of Informatics
University of Edinburgh
Informatics Forum
10 Crichton Street
Edinburgh EH8 9AB
United Kingdom

w.b.heijltjes@sms.ed.ac.uk

Abstract

Classical proof forests are a proof formalism for first-order classical logic based on Herbrand's Theorem and backtracking games in the style of Coquand. First described by Miller in a cut-free setting as an economical representation of first-order and higher-order classical proof, defining features of the forests are a strict focus on witnessing terms for quantifiers and the absence of inessential structure, or 'bureaucracy'.

This paper presents classical proof forests as a graphical proof formalism and investigates the possibility of composing forests by cut-elimination. Cut-reduction steps take the form of a local rewrite relation that arises from the structure of the forests in a natural way. Yet reductions, which are significantly different from those of the sequent calculus, are combinatorially intricate and do not exclude the possibility of infinite reduction traces, of which an example is given.

Cut-elimination, in the form of a weak normalisation theorem, is obtained using a modified version of the rewrite relation inspired by the game-theoretic interpretation of the forests. It is conjectured that the modified reduction relation is, in fact, strongly normalising.

To be published in Annals of Pure and Applied Logic *August 2010*

1. Introduction

An interesting problem in proof theory that has received much attention over the recent past is to find a satisfactory description of classical proof. Central to this problem are the related concepts of a canonical proof system, identity of proofs, and confluent normalisation (or associative composition).

In the context of intuitionistic and linear logic these concepts are inextricably linked. In the presence of confluent normalisation the normal forms of proofs provide a natural basis for proof identity. Generally, the proof system which best embodies the identifications suggested by such an account of identity is considered canonical. For classical logic, however, this scenario seems unavailable. The symmetry of classical proof, in the form of an involutive negation, is a desirable feature for an account of classical proof identity, but experience seems to suggest it rules out confluent normalisation.

In response to this, other possible foundations for proof identity have been sought, among others in category theory [2, 6, 7] and in invariants of existing proof formalisms [11, 13]. Another such direction is via a weaker notion of canonical proof. Abandoning the requirement of confluent reduction, a proof system may be considered canonical if it is free of *bureaucracy*: proof structure that is inconsequential, such as the actual order of freely permutable inferences. The idea is perhaps best illustrated by proof nets for MLL, which are a clean abstraction over the bureaucracy of sequent systems. Proof nets over the classical sequent calculus have been developed by Girard and Robinson [9, 18], and Lamarche and Straßburger have explored a minimalist notion of proof related to proof nets [14]. Intriguing in the context of canonical proof are Hughes' syntax-free 'combinatorial proofs' [12].

In a different direction, an emerging alternative to sequent calculi is 'deep inference' [3], a formalism resembling term-rewriting, with interesting reduction properties. Although it is not yet clear whether the 'atomic flows' [10] developed for normalisation form a sensible invariant, deep inference provides a contrasting view on both normalisation and bureaucracy.

Illustrative for the difficulty of the problem of proof identity, and hence finding canonical proof, is that equally well-motivated approaches, based on different vantage points, have provided different answers.

A common feature of the above approaches is an emphasis on the propositional fragment of classical proof. Inspired by Herbrand's Theorem and game semantics, this paper presents a different approach to classical proof. It investigates cut-elimination, as a means of composition, for a strictly first-order proof system that is canonical, in the sense that it is bureaucracy-free.

An important lesson of Herbrand's Theorem is that the essential content of a first-order classical proof is provided by the witnesses to quantified variables, while propositional content, being decidable, does not require explicit treatment. Several similar proof systems based on this idea have been proposed: Miller has considered 'Expansion Tree Proofs' [17], as an economical representation of

classical proof, and closely related are Buss' 'Herbrand Proofs' [4], an adaptation of Herbrand's original formulation.

Interest in such proof systems has been expressed sporadically by several people, among others by Martin Hyland, and recently ideas from other areas have brought renewed attention to the subject. McKinley, working on the extension of classical categories [7] to first-order logic [15], and the author, inspired by Coquand's backtracking games [5], have independently started investigations into normalisation for such a proof system, called *classical proof forests* here. The canonical nature of these forests, along with an elegant game-theoretic interpretation in which normalisation is viewed as the composition of strategies, makes them an interesting target for such investigations.

The reduction algorithm consists of rewrite rules that arise naturally from the available proof structure and was arrived at independently by both authors. Although reductions are combinatorially intricate, it is not hard to expose examples of non-confluence, one of which will be given. More interestingly, reductions differ significantly from those in the sequent calculus, mainly due to the way they interact with the forests' natural ability of sharing substructure.

The first main contribution of this paper is to show that this algorithm is not strongly normalising. An example exhibiting an infinite reduction path, found by the author in 2008, will be presented. Expecting reductions to be well behaved because of the elegant structure of the proof system, this example came as a surprise to both the author and McKinley.

The second contribution is to propose a modification of the algorithm, drawing on the game-theoretic interpretation of the forests and ideas from concurrency theory. The modification consists of adding a subtle garbage-collection mechanism, based on a carefully formulated correctness criterion. The modified reduction relation is shown to be weakly normalising. In addition it is conjectured to be strongly normalising.

In this approach the game-theoretic interpretation and the interesting divergence with sequent calculus, both in reduction behaviour and the ability of sharing, are preserved. An alternative direction is to use classical proof forests or a similar proof system as a vehicle for representing reductions in the sequent calculus. This is being pursued by McKinley in [16].

This paper is built up as follows. In Section 2, classical proof forests will be introduced and defined. Section 3 will present an intuitive cut-reduction algorithm for proof forests, and Section 4 will demonstrate how non-confluence and non-termination arise in this algorithm. Section 5 will present a subtle modification to the reduction algorithm and show that in its modified form it is weakly normalising. Finally, Section 6 will give a more technical account of the differences with the closely related proof systems of Miller and McKinley. The appendices hold the more technical proofs, as well as a brief explanation of the reduction mechanism from the perspective of the sequent calculus.

2. Classical Proof Forests

In this section classical proof forests will be introduced, from a view as strategies for backtracking games. Proof forests will first be treated informally, in a cut-free setting. After cuts have been introduced and different views on them have been explored, the forests will be rigourously defined.

2.1. Backtracking games

Backtracking games were identified as a means of classical proof by Coquand [5] in the early 1990s. Where regular games are generally used to show the truth of a formula in a certain model, backtracking games are a tool for establishing validity. This is made possible by using backtracking and by defining strategies over just a signature, irrespective of any particular structure.

Backtracking games can be defined in several ways: for instance, some games allow backtracking for both players; others, like the ones used here, for just one of the two. Since not much hinges on the precise choice of definition, the games will only be informally sketched.

A game is played by two players, '∀belard' (falsifier) and '∃loise' (verifier), on a chosen structure. The players take turns assigning witnesses, elements from the domain of the structure, to the quantifiers in a sequent of prenex formulae. Positions in the game are (partially) instantiated subformulae. ∃loise can revert to any previous position where it was her turn and change the chosen witness; her current position is recorded and can be a target for later backtracking. She wins the game if it reaches a quantifier-free position that is true in the structure.

A proof is a winning strategy for ∃loise. Traditionally strategies are functions that, given the history of a game, provide the next move. Proof forests deviate from this by abstracting over the order of moves in a game: moves in the strategy are only partially ordered, and given the history of a game the strategy suggests a range of possible moves.

2.2. Cut-free proof forests

Classical proof forests represent strategies for sequents of first-order formulae in prenex-normal form. A forest contains a tree for each formula in the sequent and is defined as a graph, with edges representing moves and nodes corresponding to positions. The order in which moves are played is only partially defined, by means of a partial order on nodes and edges called the *dependency*.

As an example, consider the proof of the drinker's formula[1] in Figure 1.

[1] This typical example of a classically valid formula with no constructive proof is so named after the interpretation: 'there is a man in a bar, and if he drinks, everyone drinks.' This is also the example used by Miller [17].

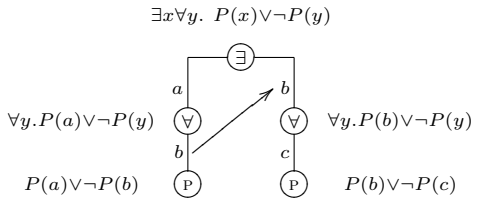


Figure 1: A forest proof of the drinker's formula

The root node at the top is the starting position: in the illustrations, edges point downwards. The strategy opens on the left branch, where ∃loise assigns an arbitrary value from the domain (represented by the variable a) to the existential quantifier. Next, ∀belard instantiates the universal quantifier with a certain value, recorded as b. If the position bottom left is true for these values, ∃loise wins. Otherwise, she backtracks to the root of the tree, this time taking the right branch and assigning the value b to the existential quantifier. Then, whichever value c ∀belard chooses, at the bottom right position $P(b) \vee \neg P(c)$ must be true, since previously in the game $P(a) \vee \neg P(b)$ was false.

The arrow in the diagram indicates where ∃loise's choice of witness relies on earlier witness assignments by ∀belard. Together with the ordering of the nodes and edges in a tree—which reflects that before the subformulae of a position can be reached it must be reached itself—this forms the *dependency ordering*. Backtracking is represented by branching at existential positions, where the strategy does not necessarily define which branch to take first.

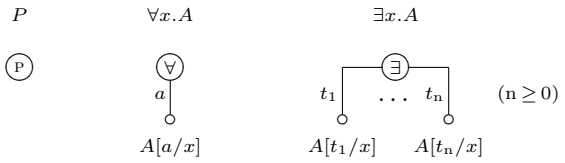


Figure 2: Forest components

As an informal, but precise characterisation, a cut-free classical proof forest is a forest of trees built from the components in Figure 2, plus a dependency ordering over the combined nodes and edges. In the diagram, P and A are propositional and prenex formulae, respectively, and the smaller circles represent arbitrary nodes. From left to right are displayed a propositional position, a move by ∀belard, and several moves from the same position by ∃loise.

A dependency ordering is a relation on nodes and edges subject to three conditions: 1) an edge is larger than its source node and smaller than its target, 2) an edge carrying ∀belard's choice a is smaller than an edge indicating ∃loise's choice t if a occurs free in t, and 3) it is a partial order. The smallest such ordering is called the *minimal* dependency. Later, a forest will be allowed to

carry a non-minimal dependency, but for now the minimal one will be used.

A forest is a proof—of the disjunction over its root nodes—if it represents a winning strategy for ∃loise, regardless of the actual structure on which any particular game is played. This is precisely the case when the disjunction over all propositional nodes in the forest forms a tautology. A cut-free forest with this property is called *correct*.

The dependency, central to classical proof forests, already appears in Miller's expansion tree proofs [17], of which cut-free proof forests are the (prenex) first-order fragment. Soundness and correctness are firmly established in that paper, and also follow from translations with the sequent calculus, described below.

$$\frac{}{\vdash A_1, \ldots, A_n}\,\mathrm{Taut}^* \qquad \frac{\vdash \Gamma, A[a/x]}{\vdash \Gamma, \forall x.A}\,\forall\mathrm{R}^{**} \qquad \frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x.A}\,\exists\mathrm{R}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, \exists x.A}\,\mathrm{W}\exists \qquad \frac{\vdash \Gamma, \exists x.A, \exists x.A}{\vdash \Gamma, \exists x.A}\,\mathrm{C}\exists$$

* $\bigvee_{i=1}^{n} A_i$ is a propositional tautology $\qquad$ ** $a \notin FV(\Gamma)$

Figure 3: A restricted version of sequent calculus

Figure 3 displays a restricted form of sequent calculus for prenex formulae, where contractions and weakenings are only allowed on existential formulae and propositional content is evaluated by a tautology check. Due to the absence of cuts and conjunctions, proofs in this system do not branch out. Soundness and correctness follow from Gentzen's sharpened Hauptsatz (also known as the midsequent theorem) [8] and a simple argument, carried out by Buss in [4], that contractions can be pushed away from universal formulae.

Cut-free proof forests and sequent proofs in this system can be translated back and forth straightforwardly, using the following interpretation. Edges in a forest correspond to ∀R-inferences and ∃R-inferences, branching on existential nodes to contraction, and an existential position without branches corresponds to a weakening. The dependency witnesses a non-permutable ordering of inferences, either because one inference's conclusion is another's premise, or due to the *eigenvariable condition*, the side-condition on ∀R-inferences that the eigenvariable may not occur free in the context. Informally, the dependants of a move in a forest correspond to the smallest possible subproof of a sequent inference, given all the possible permutations of the sequent proof. To translate a forest to a sequent proof involves making contractions explicit and strengthening the dependency to a linear order; the other direction involves the reverse.

The comparison with the sequent calculus underlines how proof forests may be considered bureaucracy-free, and in that way canonical for classical proof. The treatment of contraction as branching on existential nodes is natural from the game-theoretic perspective. And allowing the dependency to be a partial order,

where the main structure of most proof systems is a tree or linear order, means that inessential permutations such as those in the sequent calculus are factored out.

2.3. Cut

Forests for sequents Γ, A and $A^\perp, \Gamma'$ (where $A^\perp$ denotes the DeMorgan dual of A) can be composed using a cut, which links the two dual trees from both forests. Figure 4 gives a schematic impression, where triangles and trapezoids abbreviate trees and forests respectively, and the cut is labelled with the *cut-formula*. The result is a forest for the sequent Γ, Γ', whose formulae are represented by the remaining root nodes.

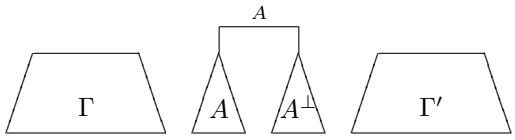


Figure 4: Composing forests for Γ, A and $A^\perp, \Gamma'$ with a cut

A common interpretation of a composition of strategies, among others found in [5, 1], is to let two strategies play against each other. Moves by ∃loise (or more generally 'the player') in one game are interpreted as moves by ∀belard (or 'the opponent') in the other game, and vice versa. The cut then indicates which moves are matched.

This interpretation, external to the games themselves, works well with strategies as functions indicating the next move, but not so well in the present setting of backtracking and partially ordered moves. For this reason the formal definition of a cut will be guided by a different, complementary interpretation in terms of moves in a game.

This requires the introduction of two additional types of move, plus a special treatment of the generic contradiction $\perp$. Firstly, $\perp$ is considered to be part of every sequent, available as a position for ∃loise at all times. As a move by ∃loise, $\perp$ is instantiated with a particular formula $A \wedge A^\perp$, which can be seen as ∃loise choosing the cut-formula A. The conjunction $A \wedge A^\perp$ is then treated in standard fashion, as a choice for either side by ∀belard. The combined construction is displayed in Figure 5; the simple bar on the right will be used as an abbreviation.

In principle, a cut-formula might contain variables assigned by ∀belard. This will not be the case with cuts that result from composition, but it will arise in reductions and when translating a sequent proof with cuts. To account for this the edge that introduces a cut-formula, the top edge in the left configuration in Figure 5, will be part of the dependency in the same way that ∃loise's other moves are. The special position $\perp$ will be unique in a forest, and will be used as the source for all cuts.

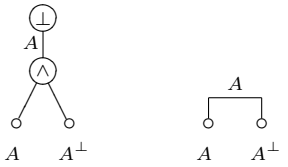


Figure 5: Cuts

The two interpretations of the cut, as strategies playing each other and as moves in a game, will both prove a valuable source of inspiration: the former will provide a natural foundation for reductions, while the latter will give the basis for a delicate correctness condition for forests with cuts.

The view of a cut as two consecutive moves hints at a more general treatment of conjunction. In the present setting the move by ∀belard is considered independent of moves in other trees, and ∃loise's choices there are not influenced by this move. To allow this would be straightforward, by extending the dependency to cover such cases. Likewise, in the design of a disjunctive position there are many choices to be made: whether to allow non-minimal dependencies on such moves, and whether to allow contraction and weakening. Since there is presently no scope to explore and motivate such choices, the current setting will remain confined to prenex formulae with cut.

$$\frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Gamma'}{\vdash \Gamma, \Gamma'}\,\mathrm{Cut}$$

Figure 6: Cut in the sequent calculus

Finally, the cut in proof forests is similar to the (multiplicative) cut of the sequent calculus (Figure 6), as both are a means of composition. Consequently, it is straightforward to translate a proof in the system of Figure 3 plus cut to a forest: a cut on proofs of Γ, A and $A^{\perp}, \Gamma'$ is translated as the composition of the translations of both proofs. However, two remarks are in order.

Firstly, the cut-formula of an inner cut (one not at the root) in a sequent proof may contain occurrences of eigenvariables of $\forall R$-inferences below it. When translated to a forest, these cuts will then be dependent on moves by ∀belard. Note that otherwise there is nothing to distinguish them from the translation of a top-level cut. This is only natural: in a sequent proof a cut-formula has no ancestors in the conclusion, and since cuts may often be permuted, which cut is actually at the root is not always significant.

Secondly, for the reverse translation, from forests with cuts to sequent proofs, a notion of *decomposition* is needed: a way of separating a forest into two subforests, while removing a cut and assigning its branches each to a different subforest. However, after composing two forests it is impossible to determine, for every tree but the two joined by the cut, to which of the original forests

it belonged. What is more, to be sure to translate to a correct sequent proof, decomposition should preserve the correctness of the forest proofs involved. How to decompose forests thus depends on the correctness criterion for forests with cuts; both will be described in Section 5.1.

2.4. Definitions

Formal definitions will closely follow the diagrams. In particular, the arrows drawn to relate dependent moves will be implemented as an explicit relation ($\rightarrow$) on edges, from which the dependency will then be generated. This will provide a better basis for reduction steps than directly defining the dependency as a partial order.

Formulae, predicates and terms are taken from a fixed, but arbitrary first-order language. The signature may include function symbols. A *sequent* is a multiset of formulae.

Definition 1 (Pre-proof forests)**.** *A pre-proof forest* F *is a tuple*

$$(\mathrm{V}, \bot, \mathrm{E}, \rightarrow, \mathit{lab}, \mathit{wit})$$

consisting of the following:

V	*a finite set of vertices,*
$\bot$	*a distinguished element of* V*,*
$\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$	*a set of edges,*
$(\rightarrow) \subseteq \mathrm{E} \times \mathrm{E}$	*a relation on* E,
lab	*a* labelling *function on* V*, assigning first-order formulae, and*
wit	*a* witnessing *function on* E*, assigning first-order formulae, terms, or one of the special symbols* L *and* R*,*

and forming a forest of trees:

$$(\mathrm{v}_1, \mathrm{w}), (\mathrm{v}_2, \mathrm{w}) \in \mathrm{E} \;\Rightarrow\; \mathrm{v}_1 = \mathrm{v}_2 \qquad \textit{(edges do not converge)}$$
$$(\mathrm{v}, \mathrm{v}_1), (\mathrm{v}_1, \mathrm{v}_2), \ldots, (\mathrm{v}_\mathrm{n}, \mathrm{w}) \in \mathrm{E} \;\Rightarrow\; \mathrm{v} \neq \mathrm{w} \qquad \textit{(acyclicity).}$$

Standard notions used are as follows: *root* nodes, those not the target of any edge; the *edges* of a node, those of which it is the source; the *children* of a node, the targets of its edges; and *leaves*, nodes without children. The variable letters $\mathrm{u}, \mathrm{v}, \ldots, \mathrm{z}$ range over nodes, while e is used for edges. A triple $(\mathrm{v}, X, \mathrm{w})$ indicates an edge (v, w) carrying the witness X. Figure 7 defines five types of node, given as subsets of V: $\mathrm{V}^\forall$, $\mathrm{V}^\exists$, V^P, $\mathrm{V}^\wedge$ and $\mathrm{V}^\bot$. Nodes in these subsets are said to be in a *legal configuration.*

The node $\bot$ is the special node, also labelled $\bot$, included for technical convenience in dealing with cuts. Membership of one of four types of edge $\mathrm{E}^\forall$, $\mathrm{E}^\exists$, $\mathrm{E}^\wedge$ and $\mathrm{E}^\bot$ reflects the type of an edge's source node. In formal definitions the name *cut* will be reserved for $\mathrm{E}^\bot$-edges; elsewhere, it will be used loosely.

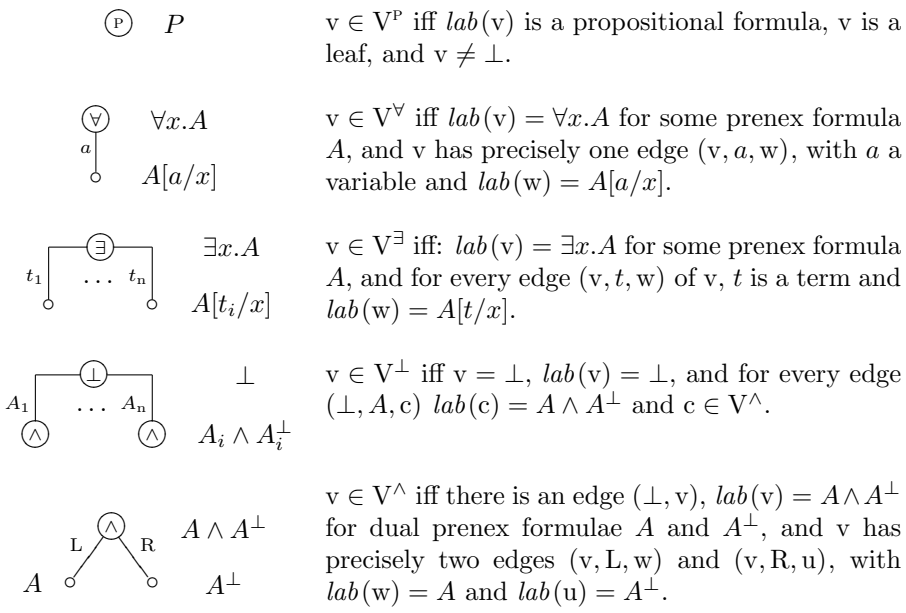


Figure 7: Legal configurations

The *dependency* $\leq$ is the smallest preorder on nodes and edges $(\mathrm{V} \cup \mathrm{E})$ s.t.

$$\begin{array}{l} \forall \mathrm{e}_1, \mathrm{e}_2 \in \mathrm{E}.\ \mathrm{e}_1 \to \mathrm{e}_2 \ \Rightarrow\ \mathrm{e}_1 \leq \mathrm{e}_2 \\ \forall (\mathrm{v}, \mathrm{w}) \in \mathrm{E}.\ \mathrm{v} \leq (\mathrm{v}, \mathrm{w}) \leq \mathrm{w}\ . \end{array}$$

For convenience the dependency ranges over both edges and vertices. Finally, the general notion of domain restriction is extended to forests. Let $f|_{\mathrm{X}}$ denote the restriction of the function $f : \mathrm{Y} \to \mathrm{Z}$ to the subdomain $\mathrm{X} \subseteq \mathrm{Y}$, and let $R|_{\mathrm{X}}$ be the relation $R \subseteq \mathrm{Y} \times \mathrm{Y}$ confined to $\mathrm{X} \times \mathrm{X}$ (where $\mathrm{X} \subseteq \mathrm{Y}$). Define:

$$\mathrm{F}|_{\mathrm{X}} = (\mathrm{X},\ \bot,\ \mathrm{E}|_{\mathrm{X}},\ \to|_{(\mathrm{E}|_{\mathrm{X}})},\ lab|_{\mathrm{X}},\ wit|_{(\mathrm{E}|_{\mathrm{X}})})$$

(it is assumed that $\bot \in \mathrm{X}$). In the above characterisation of domain restriction the forest $\mathrm{F}|_{\mathrm{X}}$ is the subgraph of F over the domain $\mathrm{X} \subseteq \mathrm{V}$. Clearly, the axioms of pre-proof forests are preserved under such restrictions, and it is in this sense that the term *subforest* will be used.

Definition 2 (Proof forests)**.** *A pre-proof forest*

$$\mathrm{F} = (\mathrm{V}, \bot, \mathrm{E}, \to, lab, wit)$$

is a proof forest for a sequent Γ *of prenex, first-order formulae if*

1. Γ *is equal to the multiset of the labels of root nodes in* $\mathrm{V} - \{\bot\}$*,*
2. *all nodes in* V *are in legal configurations (see Figure 7),*
3. *for an edge* $\mathrm{e}_1 \in \mathrm{E}^{\forall}$ *with* $wit(\mathrm{e}_1) = a$ *the following conditions hold:*

- *a is not free in any formula in* Γ,
- $a \neq wit(\mathrm{e}_2)$ *for any other edge* $\mathrm{e}_2 \in \mathrm{E}^{\forall}$,
- *if* $\mathrm{e}_2 \in \mathrm{E}^{\exists} \cup \mathrm{E}^{\bot}$ *and* $a \in FV(wit(\mathrm{e}_2))$ *then* $\mathrm{e}_1 \to \mathrm{e}_2$*;*

4. *if* $\mathrm{e}_1 \to \mathrm{e}_2$ *then* $\mathrm{e}_1 \in \mathrm{E}^{\forall}$ *and* $\mathrm{e}_2 \in \mathrm{E}^{\exists} \cup \mathrm{E}^{\bot}$,
5. *the dependency* $(\leq)$ *is a partial order (it is antisymmetric).*

In the above definition, condition 3 governs the *eigenvariables* representing ∀belard's choices. They are required to be unique and may occur free only at dependent moves and positions. The use of the explicit relation $\to$ is a generalisation, designed to make the reduction steps uniform, over using just the occurrences of eigenvariables to establish the dependency of a forest. A dependency of the latter kind, denoted $\leq_{\mathrm{M}}$, can be imposed on a forest by replacing $\to$ with $\to_{\mathrm{M}}$, defined as follows:

$$\mathrm{e}_1 \to_{\mathrm{M}} \mathrm{e}_2 \;\Leftrightarrow\; \mathrm{e}_1 \in \mathrm{E}^{\forall} \;\wedge\; \mathrm{e}_2 \in \mathrm{E}^{\exists} \cup \mathrm{E}^{\bot} \;\wedge\; wit(\mathrm{e}_1) \in FV(wit(\mathrm{e}_2)) \;.$$

Proposition 3. *The dependency imposed by variable occurrences is minimal, i.e. given a forest* F *with an arbitrary relation* $\to$*, for all* $\mathrm{v}, \mathrm{w} \in \mathrm{V}$

$$\mathrm{v} \leq_{\mathrm{M}} \mathrm{w} \quad \Rightarrow \quad \mathrm{v} \leq \mathrm{w} \;.$$

The dependency $\leq_{\mathrm{M}}$ will be called the *minimal* dependency. Condition 4 of Definition 2 requires that non-minimal dependencies respect the pattern that ∃loise's moves respond to ∀belard's.

One thing that has not yet been defined is a correctness criterion: a way to determine whether a forest represents a valid proof. As mentioned before, for a cut-free forest this is the case when the disjunction over its propositional nodes is a tautology. The primary requirement when extending this criterion to forests with cuts is that both composition with cut and cut-reduction steps should preserve correctness. The design of a correctness criterion is thus dependent on the combinatorics of the reduction algorithm. Since reduction steps will be based on a natural notion of composition of strategies, regardless of whether they are winning, they will be presented and discussed first.

3. Reductions

The view of composition as two strategies being played against each other provides a natural angle on cut-reduction. In this interpretation ∀belard's choice on one side of the cut mirrors ∃loise's move on the other side. For *logical* cuts, with precisely one existential branch, this means that the witnesses on both sides can be identified. A reduction step on such a cut makes this identification at a syntactic level, by substituting all occurrences of ∀belard's eigenvariable with ∃loise's witnessing term.

Figure 8 gives a schematic of a reduction step on a logical cut. The cut is reduced, occurrences of the eigenvariable a are replaced with the term t throughout the forest, and the dependency is adjusted accordingly. The contexts represented by Γ and Δ are, respectively, the predecessors and successors in the relation $\rightarrow$ of the existential and universal edges of the cut. As the result of the reduction step reveals, where Γ has become related to the cut itself, the full definition will need to account for dependencies towards a cut as well.

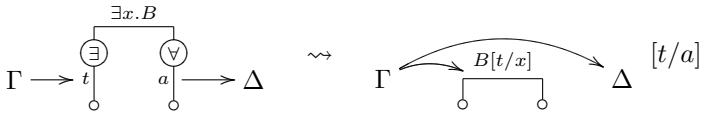


Figure 8: A logical reduction step

Considering this reduction step reveals an important problem. If the left and right edges of a cut are related in the dependency, the configurations in Figure 9 may arise. On the left, a cycle is created in the dependency. On the right the eigenvariable a occurs free in the term $t(a)$. Semantically, this would require $\forall$belard's witness a and $\exists$loise's witness $t(a)$ to be identified. Resolving this cut with a substitution $[t(a)/a]$, which removes the eigenvariable a but not its occurrences in the substituted terms $t(a)$, is clearly undesirable.

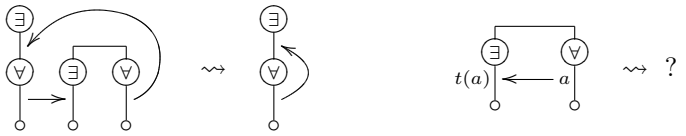


Figure 9: A problem with logical reduction steps

The general situation, of which the illustrations in Figure 9 are instances, is when the existential edge of a logical cut depends on the universal edges. A dependency between the edges of a cut, in this way, is not a natural situation, and does not arise from composition. The configuration will be called a *bridge*. The problem will be addressed in a more general setting in Section 5; for now, a cut with a bridge will be left unreduced.

Definition 4 (Bridges)**.** *A cut* $(\bot, \mathrm{c}) \in \mathrm{E}^{\bot}$ *in a forest* F *is said to have a* bridge *if it is a logical cut where the existential edge depends on the universal edge, i.e. it consists of edges* (c, u)*,* (u, v)*,* (c, x) *and* (x, y) *such that* $\mathrm{u} \in \mathrm{V}^{\forall}$*,* $\mathrm{x} \in \mathrm{V}^{\exists}$ *and* $(\mathrm{u}, \mathrm{v}) \leq (\mathrm{x}, \mathrm{y})$*.*

A second problem is how to address backtracking, represented by arbitrary branching at existential positions, which means there may be no unique move for $\forall$belard to copy. However, the strategy describes how $\exists$loise should react to any move he picks; moreover, moves made in response to $\forall$belard's choice are precisely those indicated by the dependency. This gives a natural way of

treating such a cut: ∀belard's move and all its dependants are duplicated, and matched against one of the existential branches of the cut.

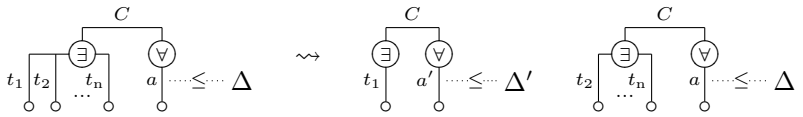
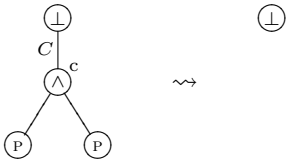


Figure 10: A structural reduction step

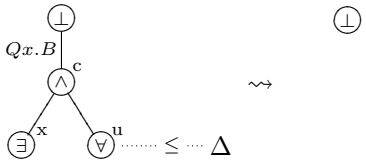
A reduction step of this kind is called a *structural* step, displayed in Figure 10. The context Δ, which represents the dependants of the universal side of the cut, is duplicated, with the copy indicated as Δ'. Then a new cut pairs Δ' with one of the existential branches, here the one carrying the witness t_1.

In addition to logical and structural reduction steps, two more types of reduction step are needed: a *propositional* step for propositional cuts, and a *disposal* step for first-order cuts with no existential branches. Before giving a formal definition, the four types of reduction step will first be illustrated in more detail. Another instructive angle on the reduction steps is provided in Appendix B, where proof forest reductions are compared to those in the sequent calculus.

Let $(\bot, \mathrm{c})$ be the cut to be reduced (the *primary cut* of the reduction step). If $lab(\mathrm{c})$ is propositional, $(\bot, \mathrm{c})$ is reduced by a *propositional reduction step*: the cut and all its dependants, which are just the edges and children of c, are removed, as illustrated in the diagram below. The other children of the node $\bot$ remain intact.

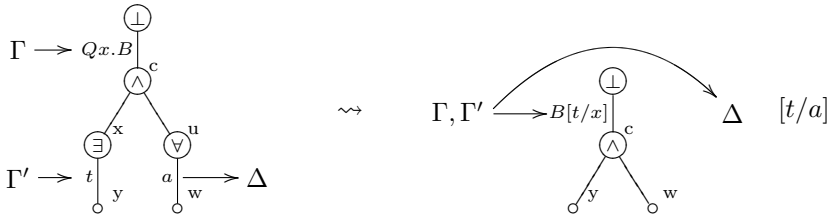


The remaining three steps reduce first-order cuts with 0, 1, or more existential edges. The first of these is a *disposal step*, where the existential node x is a leaf, depicted below. The context Δ represents the dependants of u. The reduction step removes the cut and all its dependants: the nodes c, x, u, and the nodes and edges represented by Δ. Nodes that are a source but not a target of an edge that is removed, such as $\bot$, remain in place.



For a logical cut, where the existential side of the cut has a single branch, the reduction step consists of local rearrangement of edges and dependencies, depicted below, plus a global substitution of the witnessing term on the existential

side for the eigenvariable on the universal side, here $[t/a]$.

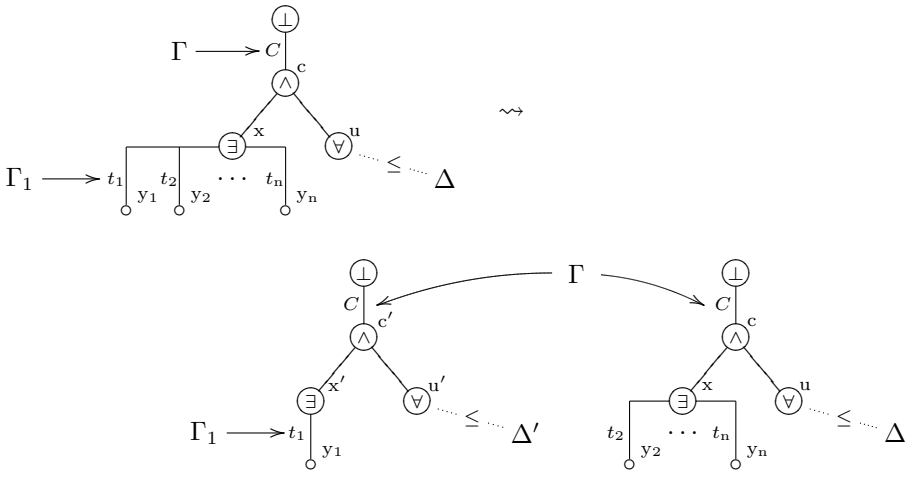


The contexts Γ, Γ' and Δ portray the following sets of edges:

$$\Gamma = \{\mathrm{e} \mid \mathrm{e} \to (\bot, \mathrm{c})\} \qquad \Gamma' = \{\mathrm{e} \mid \mathrm{e} \to (\mathrm{x}, \mathrm{y})\} \qquad \Delta = \{\mathrm{e} \mid (\mathrm{u}, \mathrm{w}) \to \mathrm{e}\}\ .$$

The removal of the edges (x, y) and (u, w), and the global substitution $[t/a]$, implement ∃loise mirroring ∀belard's strategy. The rewiring of the arrows is designed to preserve the dependency relation for all other edges.

A structural reduction step on a cut $(\bot, \mathrm{c})$ is best described as consisting of two stages. First, the cut and its dependants on the universal side are duplicated. Then from the edges on the existential side of the original cut a *primary edge* is selected, here $(\mathrm{x}, \mathrm{y}_1)$, and attached to the duplicate cut. The reduction is shown pictorially below; in the result, for aesthetic reasons the node $\bot$ is represented twice, but it is not duplicated.



Here, the contexts Γ, Γ_1 and Δ are:

$$\Gamma = \{\mathrm{e} \mid \mathrm{e} \to (\bot, \mathrm{c})\} \qquad \Gamma_1 = \{\mathrm{e} \mid \mathrm{e} \to (\mathrm{x}, \mathrm{y}_1)\} \qquad \Delta = \{\mathrm{v} \mid \mathrm{u} \leq \mathrm{v}\}\ .$$

The duplication of parts of a forest is made complicated by the need to preserve dependencies. The way duplication will be implemented in Definition 5 is illustrated in Figure 11. To duplicate the dependants of the node v in the forest F,

first the forest F' is created, by renaming just the nodes depending on v. Then F and F' are combined: $\mathrm{F} \cup \mathrm{F}'$ denotes the pointwise union of the components of both forests, assuming both share the same special $\bot$-node. In the definition below, eigenvariables will also be duplicated and renamed; for simplicity, that is omitted in Figure 11.

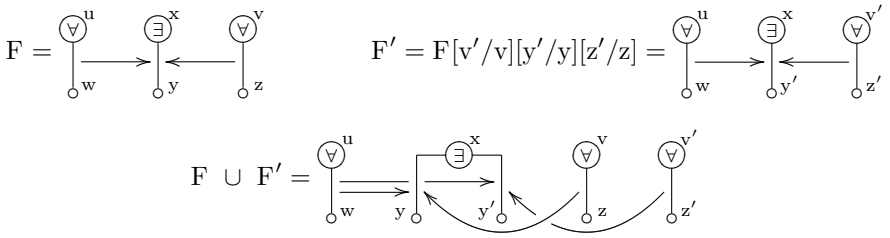


Figure 11: Duplication (of the node v and its dependants)

Definition 5 (Reduction)**.** *A reduction step* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ *reduces a forest* $\mathrm{F_A}$ *to a forest* $\mathrm{F_B}$*, where the superscript* c *indicates a* primary cut $(\bot, C, \mathrm{c}) \in \mathrm{E_A}$ *that does not have a bridge, in one of four ways described below.*

I. *A propositional step. If the cut-formula* C *is propositional, then define* $\mathrm{F_B} = \mathrm{F_A}|_{\mathrm{X}}$*, where*

$$\mathrm{X} = \{\mathrm{v} \in \mathrm{V} \mid \mathrm{c} \not\leq_{\mathrm{A}} \mathrm{v}\} \ .$$

In the remaining three cases, the cut-formula C *is of the form* $Qx.B$*, with* Q *one of* $\forall, \exists$*. Let the edges of the node* c *be* $(\mathrm{c}, U, \mathrm{u})$ *and* $(\mathrm{c}, X, \mathrm{x})$*, where* $\mathrm{u} \in \mathrm{V_A^\forall}$*,* $\mathrm{x} \in \mathrm{V_A^\exists}$ *and* $\{U, X\} = \{\mathrm{L}, \mathrm{R}\}$*.*

II. *A disposal step. If the existential node* x *has no edges, then define* $\mathrm{F_B} = \mathrm{F_A}|_{\mathrm{X}}$*, where*

$$\mathrm{X} = \{\mathrm{v} \in \mathrm{V} \mid \mathrm{c} \not\leq_{\mathrm{A}} \mathrm{v}\} \ .$$

III. *A logical step. If the node* x *has precisely one edge* $(\mathrm{x}, t, \mathrm{y})$*, let* $(\mathrm{u}, a, \mathrm{w})$ *be the edge of* u*, and define* $\mathrm{F_B}$ *as follows:*

- $\mathrm{V_B} = \mathrm{V_A} - \{\mathrm{u}, \mathrm{x}\}$.
- $\mathrm{E_B}$ *and* wit_{B} *are obtained from* $\mathrm{E_A}$ *and* wit_{A} *by replacing*
  $(\bot, Qx.B, \mathrm{c})$ *with* $(\bot, B[t/x], \mathrm{c})$,
  $(\mathrm{c}, U, \mathrm{u})$ *and* $(\mathrm{u}, a, \mathrm{w})$ *with* $(\mathrm{c}, U, \mathrm{w})$,
  $(\mathrm{c}, X, \mathrm{x})$ *and* $(\mathrm{x}, t, \mathrm{y})$ *with* $(\mathrm{c}, X, \mathrm{y})$*, and*
  any other $(\mathrm{v}_1, Y, \mathrm{v}_2)$ *with* $(\mathrm{v}_1, Y[t/a], \mathrm{v}_2)$.
- *The relation* $(\rightarrow_{\mathrm{B}})$ *is the smallest relation on* $\mathrm{E_B}$ *such that*
  $\mathrm{e}_1 \rightarrow_{\mathrm{B}} \mathrm{e}_2$ *if* $\mathrm{e}_1 \rightarrow_{\mathrm{A}} \mathrm{e}_2$,
  $\mathrm{e}_1 \rightarrow_{\mathrm{B}} \mathrm{e}_2$ *if* $\mathrm{e}_1 \rightarrow_{\mathrm{A}} (\bot, \mathrm{c})$ *and* $(\mathrm{u}, \mathrm{w}) \rightarrow_{\mathrm{A}} \mathrm{e}_2$,
  $\mathrm{e}_1 \rightarrow_{\mathrm{B}} \mathrm{e}_2$ *if* $\mathrm{e}_1 \rightarrow_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$ *and* $(\mathrm{u}, \mathrm{w}) \rightarrow_{\mathrm{A}} \mathrm{e}_2$*, and*
  $\mathrm{e}_1 \rightarrow_{\mathrm{B}} (\bot, \mathrm{c})$ *if* $\mathrm{e}_1 \rightarrow_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$.

– $lab_{\mathrm{B}}(\mathrm{c}) = B \wedge B^{\perp}[t/x]$; *otherwise* $lab_{\mathrm{B}}(\mathrm{v}) = lab_{\mathrm{A}}(\mathrm{v})[t/a]$.

IV. *A structural step. If the node* x *has two or more edges* $(\mathrm{x},\mathrm{y}_1),\ldots,(\mathrm{x},\mathrm{y}_\mathrm{n})$, *one* $(\mathrm{x},\mathrm{y}_\mathrm{i})$ *is selected as the* primary edge. *Let* ρ, σ *and* τ *be the following substitution maps on nodes, (eigen)variables and edges, respectively:*

$$\rho = \{\mathrm{v} \mapsto \mathrm{v}' \mid \mathrm{v} = \mathrm{c} \ \vee \ \mathrm{u} \leq_{\mathrm{A}} \mathrm{v}\}$$

$$\sigma = \{a \mapsto a' \mid (\mathrm{v}, a, \mathrm{w}) \in \mathrm{E}_{\mathrm{A}}^{\forall} \ \wedge \ \mathrm{u} \leq_{\mathrm{A}} (\mathrm{v},\mathrm{w})\}$$

$$\tau = \{(\mathrm{c}',\mathrm{x}) \mapsto (\mathrm{c}',\mathrm{x}'),\ (\mathrm{x},\mathrm{y}_\mathrm{i}) \mapsto (\mathrm{x}',\mathrm{y}_\mathrm{i})\}$$

where all v' *and* a' *are fresh for* F_{A}. *Then define* F_{B} *as follows:*

$$\mathrm{F}_{\mathrm{B}} = \big(\ \mathrm{F}_{\mathrm{A}} \ \cup \ \mathrm{F}_{\mathrm{A}}[\rho][\sigma]\ \big)\ [\tau]\ .$$

Technically, a structural step proceeds as follows. The substitution ρ duplicates the graph over the cut-node c plus the dependants of its universal child u. The substitution σ provides fresh eigenvariables for the duplicated subgraph. At this point, the existential side is shared between the old cut and the new cut, as there are edges (c,x) and (c',x). This is corrected by the substitution τ, which transfers the edges (c',x) and $(\mathrm{x},\mathrm{y}_\mathrm{i})$ from x to a new node x'.

Whereas propositional and disposal steps only remove nodes and edges, logical and structural reduction steps involve adding and restructuring them. The following two lemmas describe how the changes affect the dependency ($\leq$).

Lemma 6. *In a logical reduction step* $\mathrm{F}_{\mathrm{A}} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F}_{\mathrm{B}}$, *where* $\mathrm{c},\mathrm{u},\mathrm{w},\mathrm{x},\mathrm{y}$ *are as in Definition 5 part* III, *for all* $\mathrm{v}_1,\mathrm{v}_2 \in \mathrm{V}_{\mathrm{B}}$ *s.t.* $\mathrm{v}_1 \notin \{\perp,\mathrm{c}\}$,

$$\mathrm{v}_1 \leq_{\mathrm{B}} \mathrm{v}_2 \quad \Leftrightarrow \quad \mathrm{v}_1 \leq_{\mathrm{A}} \mathrm{v}_2 \ \vee \ (\mathrm{v}_1 \leq_{\mathrm{A}} (\mathrm{x},\mathrm{y}) \ \wedge \ (\mathrm{u},\mathrm{w}) \leq_{\mathrm{A}} \mathrm{v}_2)$$

Proof. By inspection of the rewrite rule. □

Lemma 7. *In a structural reduction step* $\mathrm{F}_{\mathrm{A}} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F}_{\mathrm{B}}$, *where the nodes* $\mathrm{c},\mathrm{u},\mathrm{x},\mathrm{y}_\mathrm{i}$ *and the renaming convention* $\mathrm{v} \mapsto \mathrm{v}'$ *are as in Definition 5 part* IV, *for all* $\mathrm{v},\mathrm{w} \in \mathrm{V}_{\mathrm{A}}$,

$$\begin{array}{lcl}
\mathrm{v} \leq_{\mathrm{B}} \mathrm{w} & \Rightarrow & \mathrm{v} \leq_{\mathrm{A}} \mathrm{w} \\
\mathrm{v} \leq_{\mathrm{B}} \mathrm{w}' & \Rightarrow & \mathrm{v} \leq_{\mathrm{A}} \mathrm{w} \ \wedge \ \mathrm{u} \not\leq_{\mathrm{A}} \mathrm{v} \\
\mathrm{v}' \leq_{\mathrm{B}} \mathrm{w} & \Rightarrow & \mathrm{v} \leq_{\mathrm{A}} \mathrm{w} \ \wedge \ \mathrm{v} \in \{\mathrm{c},\mathrm{x}\} \ \wedge \ \mathrm{y}_\mathrm{i} \leq_{\mathrm{A}} \mathrm{w} \\
\mathrm{v}' \leq_{\mathrm{B}} \mathrm{w}' & \Rightarrow & \mathrm{v} \leq_{\mathrm{A}} \mathrm{w}
\end{array}$$

Proof. By inspection of the rewrite rule. □

As the above lemma shows, a dependency between a duplicated node and an original one only occurs in limited circumstances. The dependants of u and their duplicates, the dependants of u', are entirely separate in this respect. The following basic proposition establishes that reduction steps are well defined.

Proposition 8. *If* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ *then* $\mathrm{F_B}$ *is a proof forest.*

Proof. See appendix. □

4. Non-confluence and non-termination

Cut-reduction in classical proof forests is based on a natural notion of composition of strategies. As presented in Figure 8, a logical reduction step is centred on the substitution, from which the restructuring of the dependency follows. Structural reduction steps, in Figure 10, use the dependency to duplicate only the necessary minimum. Although the technical implementation is quite involved, the design of the reduction steps follows from the structure of the forests in a natural way.

Since the forests themselves are a natural representation of classical proof, this raised the hope that reductions would be well behaved. The fact of the matter is, however, that reductions display both non-confluence and non-termination. In this section, both will be demonstrated by examples.

4.1. Non-confluence

Non-confluence is certainly no stranger to symmetric classical proof. Nonetheless, in this particular case the appearance of non-confluence is interesting because of two characteristics of proof forests. Firstly, the forests are strictly focussed on witness assignment to quantifiers, and cuts aside no propositional connectives are evaluated. Secondly, forests are *polarised* in the sense that contraction and weakening are only applied to existential formulae. Thus two notorious cases in the standard sequent calculus, a cut on two weakenings and a cut on two contractions, are avoided (see e.g.[9, 14]).

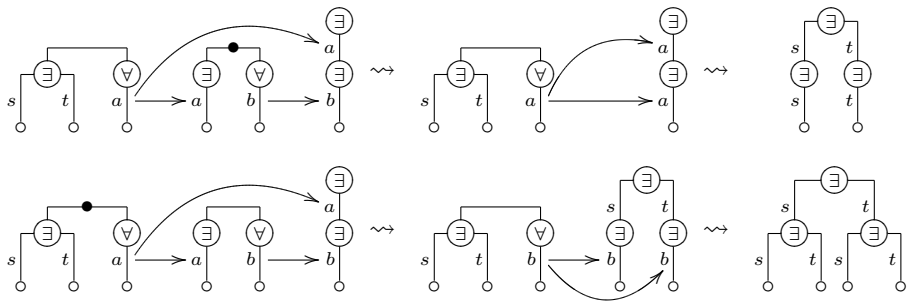


Figure 12: An example of non-confluence

Many examples of non-confluence in proof forests have been found, centred around a common principle. The particular example in Figure 12, due to

Richard McKinley, is probably the simplest one (in the diagram the black tokens indicate the primary cut of a reduction step).

The general principle illustrated by Figure 12 is that different free variables (a and b) may be identified by reductions, as happens in the upper reduction, or may be instantiated without being identified, as in the lower reduction.

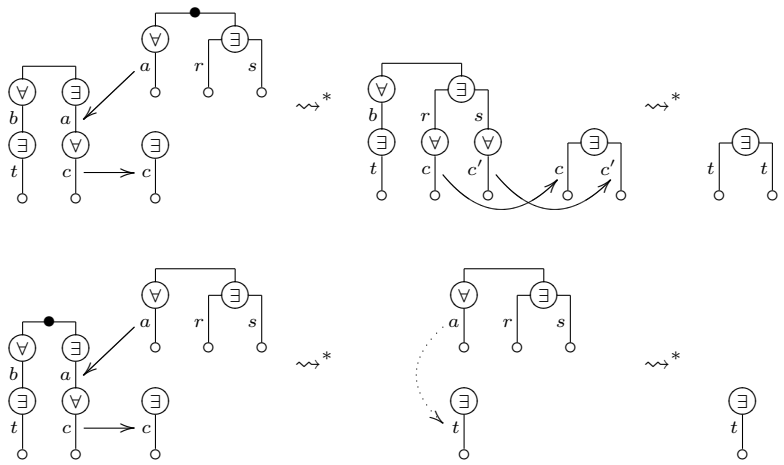


Figure 13: Non-confluence with minimal dependencies (where $b \notin FV(t)$)

An additional form of non-confluence is introduced when forests are only allowed to have minimal dependencies, enforced by replacing the dependency of a forest with the minimal one after each reduction step. In Figure 13 this is applied in the lower reduction path, where the dotted arrow results from the reduction, but is not part of the minimal dependency. The reduction shows that minimality of the dependency is not preserved in reductions, and that reverting to a minimal dependency after each reduction step results in an additional form of non-confluence.

4.2. Non-termination

A second problem is the existence of infinite reduction paths. Figure 14 shows a configuration of two cuts that exhibits such behaviour.

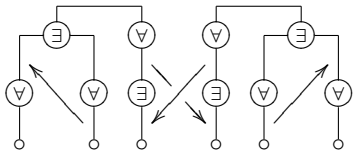


Figure 14: The counterexample to strong normalisation

The example in Figure 14 consists only of cuts and the trees below them, and omits labels and witnesses. A forest with this configuration can be constructed as the seemingly innocent composition of three cut-free forests: two proofs similar to that of the drinker's formula, as in Figure 1, and one as in Figure 15.

The counterexample in Figure 14 is reduced in Figure 17 on page 20, until a single cut remains. Inessential dependencies have been drawn in dotted arrows, and in places several reduction steps have been taken at once; such multi-steps are indicated as $(\rightsquigarrow^*)$.

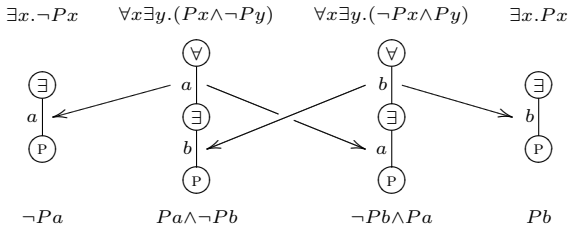


Figure 15: A proof that instantiates part of the counterexample

The last remaining cut in Figure 17 may reduce to a similar cut, plus an additional one. Clearly such a reduction step, shown in Figure 16, gives rise to an infinite reduction path. The cut itself is similar to a bridge, and equally unnatural; yet this cut arises by reduction from a composition of cut-free forests. The following proposition summarises the above findings.

Proposition 9. *The reduction relation* $(\rightsquigarrow)$ *is not strongly normalising. This holds even for the class of forests that arise from cut-free forests by composition with cut.*

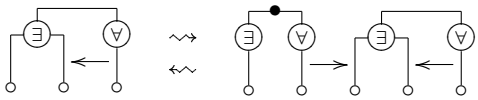


Figure 16: A reduction cycle

5. The Modified Reduction Relation

The main obstacle to obtaining normalisation is the cyclic reduction shown in Figure 16. Below, a correctness condition for forests with cuts will be presented that will provide a natural way of solving this problem. A remaining problem will be identified and addressed; after the reduction relation has been modified to incorporate these solutions, it will be shown to be weakly normalising.

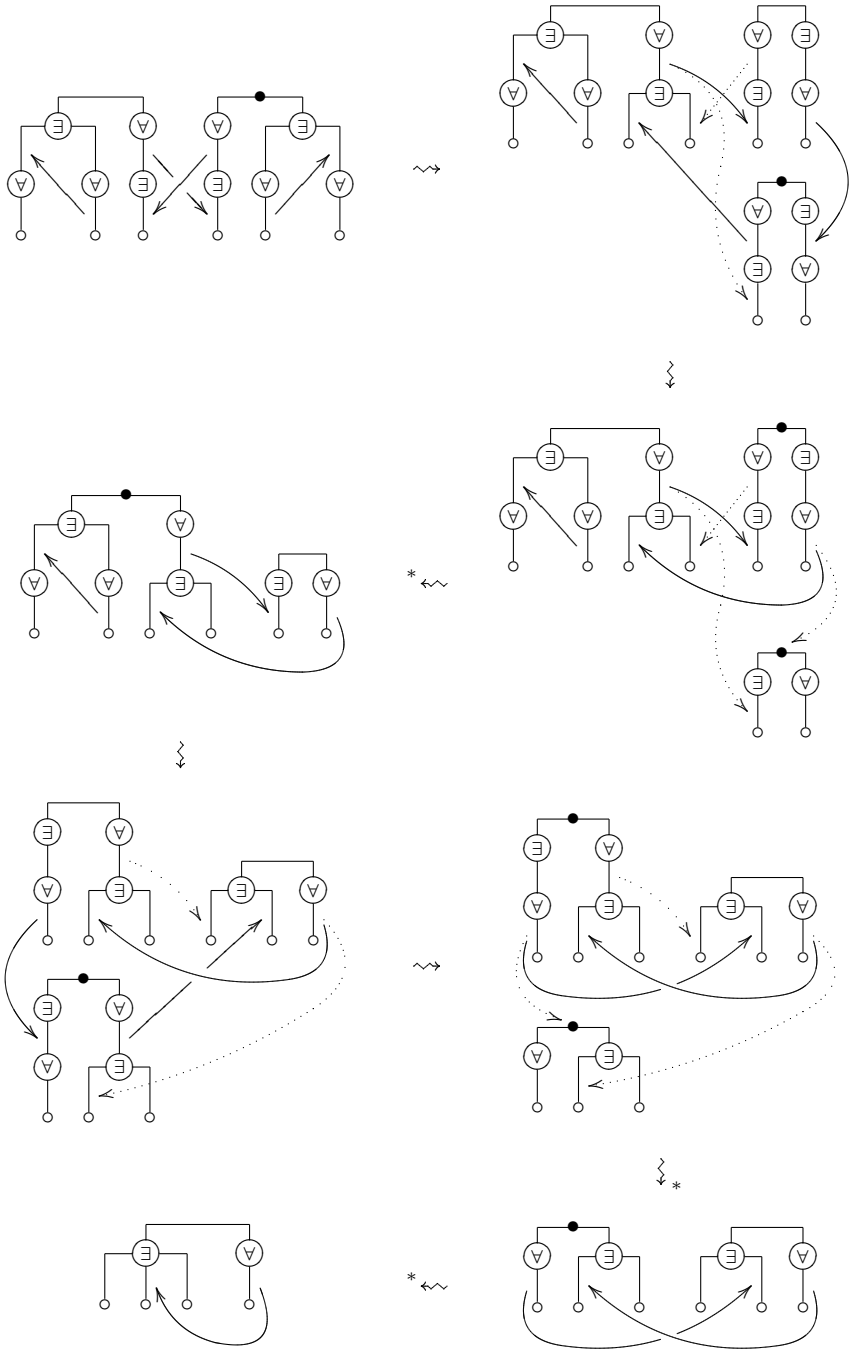


Figure 17: Reducing the counterexample to strong normalisation

5.1. Correctness

As mentioned before, the idea is to construct a correctness condition such that a correct forest represents a winning strategy for ∃loise. In a game played according to a cut-free strategy, she can visit every propositional position in turn using backtracking; hence, a cut-free forest is correct if the disjunction over all propositional nodes is a tautology.

The view of a cut as encompassing a conjunction, where it is ∀belard's turn to move, provides a way of dealing with cuts in terms of games and strategies. In a game, a choice by ∀belard of either the left or right branch of a conjunction has as a consequence that the *other* subformula is *not* a position in the game. In addition, if a position in a strategy is not reached in a certain game, then any position depending on it will not be reached either.

A way to define correctness is then as follows: given any combination of choices ∀belard makes at conjunctions, the propositional positions still available to ∃loise must form a tautology. A strategy for ∀belard on conjunctions is called a *switching*; after unreachable positions have been removed a *switched forest* remains, which is naturally cut-free.

Definition 10 (Switching)**.** *A* switching $s : \mathrm{V}^{\wedge} \to \{\mathrm{L}, \mathrm{R}\}$ *is a choice of* L, R *for each conjunction node in a forest* F*, indicating a set* $\mathrm{E}^s \subseteq \mathrm{E}^{\wedge}$ *that contains one branch of each conjunction:*

$$\mathrm{E}^s = \{(\mathrm{v}, \mathrm{w}) \in \mathrm{E}^{\wedge} \mid s(\mathrm{v}) \neq \mathit{wit}(\mathrm{v}, \mathrm{w})\} \ .$$

A switched forest F^s *is the subforest* $\mathrm{F}|_{\mathrm{X}}$*, where*

$$\mathrm{X} = \{\mathrm{v} \in \mathrm{V} \mid \mathrm{v} \notin \mathrm{V}^{\wedge} \ \wedge \ \forall \mathrm{e} \in \mathrm{E}^{\mathrm{S}}.\ \mathrm{e} \not\leq \mathrm{v}\} \ .$$

The edges E^s are the branches ∀belard does *not* choose; their dependent positions become unreachable in the game, and are ignored in the switched forest.

Definition 11 (Correctness)**.** *Let the* value *of a cut-free proof forest* F *be the disjunction over all propositional positions:*

$$\mathit{val}(\mathrm{F}) = \bigvee \{\mathit{lab}(\mathrm{v}) \mid \mathrm{v} \in \mathrm{V}^{\mathrm{P}}\} \ .$$

A proof forest F *is* correct *if for any switching* s *the value* $\mathit{val}(\mathrm{F}^s)$ *is a tautology.*

Remark. The present definition of correctness is different from the one used by Miller in [17], which looks at the tree-structure of the forest, not at the dependency as a whole. The difference, together with the consequences of using the current correctness criterion, is explored in detail in Section 6.

As to the requirements mentioned at the end of Section 2, a cut-free forest only has the empty switching, and it is easy to see that the composition of two correct forests with a cut is correct itself. That correctness is also preserved under reduction steps is expressed by the following proposition.

Proposition 12. *If* $\mathrm{F_A} \overset{c}{\rightsquigarrow} \mathrm{F_B}$ *and* $\mathrm{F_A}$ *is correct, then so is* $\mathrm{F_B}$.

Proof. See appendix. □

The switching condition makes it directly insightful how forests can be *decomposed*. Consider a forest with a cut that is not dependent on any other edge. For either branch of the cut, removing all dependants will leave a correct forest, since this is precisely what a switching does. This characterisation is sufficient to interpret a proof forest with cuts as a sequent proof, but decomposing in this way is clearly not inverse to composition with cut. In sequent calculus terms, composition treats a cut as multiplicative, while decomposition treats it as additive. This is a natural consequence of the game-semantic treatment. One important aim of McKinley's project is to modify proof forests to make an inverse to composition possible, and thus obtain a closer correspondence to sequent calculus; see Section 6 for technical details.

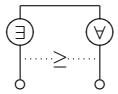


Figure 18: A bridge

The treatment of conjunctions as moves by ∀belard also sheds new light on bridges. Recall that a bridge is a dependency from the universal edge to the existential edge of a logical cut, as illustrated in Figure 18. The existential edge then depends on both the left and right positions in the cut, of which ∀belard only chooses one; consequently, such a move or position can never be reached in a game.

A natural way of characterising this is via a notion of *conflict*, as encountered in concurrency theory (see e.g. [19]). There, it is a relation between events that are mutually exclusive. Applied to ∀belard's moves at conjunctions, where a choice for one branch prevents positions dependent on the other branch from being reached, this gives the following definition.

Definition 13 (Conflict)**.** *The conflict relation* (#) *holds between nodes that depend on different branches of the same conjunction:*

$$\mathrm{v_1 \# v_2} \;\Leftrightarrow\; \exists (\mathrm{c,u}), (\mathrm{c,w}) \in \mathrm{E^{\wedge}}.\ \mathrm{u} \neq \mathrm{w} \;\wedge\; \mathrm{u} \leq \mathrm{v_1} \;\wedge\; \mathrm{w} \leq \mathrm{v_2}\ .$$

A position in conflict with itself will never be reached. A forest containing no such positions will be called *safe*.

Definition 14 (Safeness)**.** *A proof forest is* safe *if* (#) *is irreflexive.*

Since the correctness condition is based on which positions ∃loise can reach in any particular game, safeness can be trivially enforced by removing self-conflicting positions; this will be called *pruning*.

Definition 15 (Pruning)**.** *The* pruning *function removes all self-conflicting nodes from a forest:* $\mathit{prune}(\mathrm{F}) = \mathrm{F}|_{\mathrm{X}}$*, where* $\mathrm{X} = \{\mathrm{v} \in \mathrm{V} \mid \neg \mathrm{v} \# \mathrm{v}\}$.

A pruned forest is by definition safe, and the restriction that $\rightarrow$ points from $\mathrm{E}^{\forall}$-edges to $\mathrm{E}^{\exists}$-edges or $\mathrm{E}^{\bot}$-edges ensures that no branching conditions are violated. Correctness is preserved, because a self-conflicting node is not a member of any switched forest.

Proposition 16. *Pruning preserves correctness.*

Proof. Given a forest F, a node $\mathrm{v} \in \mathrm{V}$ such that $\mathrm{v} \# \mathrm{v}$, and any switching s, then $\mathrm{v} \notin \mathrm{F}^s$. The switchings of F are those of $\mathit{prune}(\mathrm{F})$ plus an insignificant choice at self-conflicting conjunction nodes. The switched forests of $\mathit{prune}(\mathrm{F})$ are precisely those of F. □

The above is also immediate from an alternative formulation of the notion of a switched forest, using the conflict relation: a switched forest is a subforest over a *largest conflict-free subset* of the nodes of the original forest.

Before it is shown how pruning is used to obtain weak normalisation, the following proposition shows that allowing non-minimal dependencies is a conservative generalisation, in the sense that the dependency in a forest can always be replaced with the minimal one.

Proposition 17. *A proof forest remains safe and correct if the dependency is replaced with the minimal one.*

Proof. Let F be a forest, and let F_{M} be the same forest with the dependency replaced by the minimal one ($\leq_{\mathrm{M}}$). Given that $\mathrm{v} \leq_{\mathrm{M}} \mathrm{w}$ implies that $\mathrm{v} \leq \mathrm{w}$, it is clear from Definition 13 that $\mathrm{v} \#_{\mathrm{M}} \mathrm{w}$ implies that $\mathrm{v} \# \mathrm{w}$, by which safeness is preserved.

Next, any switching s for F is also a switching for F_{M}, and vice versa. Since the dependency in F_{M} is strictly smaller, fewer nodes are removed in a switched forest, and $\mathrm{F}_{\mathrm{M}}^s$ has all the nodes of F^s. Hence, $\mathit{val}(\mathrm{F}^s) \Rightarrow \mathit{val}(\mathrm{F}_{\mathrm{M}}^s)$, and if F is correct, so is F_{M}. □

5.2. Compound reduction steps

Pruning solves the infinite reduction of Figure 16 by removing the branch that causes the problem, as shown in Figure 19.

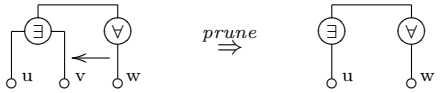


Figure 19: Pruning at work

However, pruning alone is not sufficient to force the counterexample in Figure 14 to normalise. In a configuration of two cuts with mutually dependent branches, as in the bottom right diagram in Figure 17, each cut can duplicate the other's existential branches.[2] Figure 20 illustrates how this gives rise to a cyclic reduction path. Note that the reduction returns only to the same *configuration*; the labels and witnesses, omitted from the diagram, will have changed over a cycle.

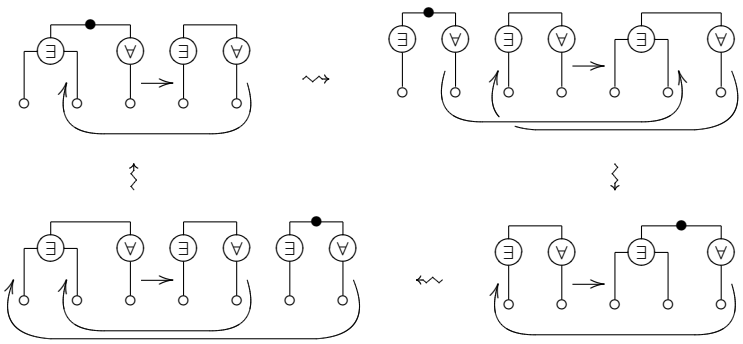


Figure 20: Another cyclic reduction path

The second problem, infinite reductions of the kind in Figure 20, is addressed by grouping reduction steps together: isolating a single ∃-edge from a cut will require isolating the other ∃-edges as well, and immediately reducing the isolated logical cuts. A duplication initiated by a cut can then no longer incur a duplication on the existential edges of that cut, because it will have been reduced in its entirety.

Definition 18 (Compound reduction steps)**.** *A* compound reduction step $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathrm{F_B}$ *on a safe forest* $\mathrm{F_A}$ *is a series of reduction steps and pruning steps, inductively defined as follows (roman numerals refer to Definition 5).*

- *If* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ *according to* I *or* II*, then* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathrm{F_B}$*.*
- *If* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ *according to* III *, then* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathit{prune}(\mathrm{F_B})$*.*
- *If* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ *according to* IV*,* $\mathrm{F_B} \overset{\mathrm{c'}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathrm{F_C}$ *and* $\mathrm{F_C} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathrm{F_D}$*, then* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\Rightarrow} \mathrm{F_D}$ *(where* $\mathrm{c'}$ *is the duplicate of* c *as in Definition 5).*

A compound reduction step on a first-order cut isolates the existential branches one by one, while immediately reducing each newly formed cut. Pruning after each logical step enforces safeness. Two additional properties will be established: firstly, crucially, that a compound step is finite, and secondly, that existential branches can be isolated in any order, with the same result. Both follow from the next lemma.

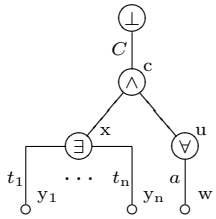
[2]This was pointed out by Richard McKinley in private communication.

Lemma 19. *Given successive structural and logical reduction steps*

$$\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B} \overset{\mathrm{c'}}{\rightsquigarrow} \mathrm{F_C}$$

where $\mathrm{F_A}$ *is safe and* $\mathrm{c'}$ *is as in Definition 5, the subgraph dependent on the universal side of the cut* $(\bot, \mathrm{c})$ *is identical in* $\mathrm{F_A}$ *and in* $\mathrm{F_D} = \mathit{prune}(\mathrm{F_C})$*:* $\mathrm{F_A}|_\mathrm{X} = \mathrm{F_D}|_\mathrm{Y}$*, where* $\mathrm{X} = \{\mathrm{v} \in \mathrm{V_A} \mid \mathrm{u} \leq_\mathrm{A} \mathrm{v}\}$ *and* $\mathrm{Y} = \{\mathrm{v} \in \mathrm{V_D} \mid \mathrm{u} \leq_\mathrm{D} \mathrm{v}\}$.

Proof. Let the cut $(\bot, \mathrm{c})$ in $\mathrm{F_A}$ be configured as below:



Let $(\mathrm{x}, \mathrm{y_i})$ be the primary edge in $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$. The nodes in $\mathrm{F_B}$ and $\mathrm{F_C}$ are

$$\begin{aligned}\mathrm{V_B} &= \mathrm{V_A} \cup \{\mathrm{v'} \mid \mathrm{v} = \mathrm{c} \ \vee \ \mathrm{v} = \mathrm{x} \ \vee \ \mathrm{u} \leq_\mathrm{A} \mathrm{v}\} \\ \mathrm{V_C} &= \mathrm{V_B} - \{\mathrm{x'}, \mathrm{u'}\} = \mathrm{V_A} \cup \{\mathrm{v'} \mid \mathrm{v} = \mathrm{c} \ \vee \ \mathrm{u} <_\mathrm{A} \mathrm{v}\} \ .\end{aligned}$$

Clearly, $\mathrm{V_A} \subseteq \mathrm{V_C}$; likewise, all edges in $\mathrm{E_A}$ are present in $\mathrm{E_C}$ with the exception of the primary edge $(\mathrm{x}, \mathrm{y_i})$. With the same exception, $\rightarrow_\mathrm{A}$ is a subrelation of $\rightarrow_\mathrm{C}$, and $\mathrm{F_A}|_\mathrm{X}$ is a subforest of $\mathrm{F_C}|_\mathrm{Z}$, where $\mathrm{Z} = \{\mathrm{v} \in \mathrm{V_C} \mid \mathrm{u} \leq_\mathrm{C} \mathrm{v}\}$.

To show that $\mathrm{u} \leq_\mathrm{C} \mathrm{v} \Rightarrow \mathrm{u} \leq_\mathrm{A} \mathrm{v}$, by Lemma 7 $\mathrm{u} \leq_\mathrm{B} \mathrm{v}$ implies that $\mathrm{u} \leq_\mathrm{A} \mathrm{v}$ for any $\mathrm{v} \in \mathrm{V_A}$. In the second step, by Lemma 6 a new dependency $\mathrm{v_1} \leq_\mathrm{C} \mathrm{v_2}$ is introduced only when $\mathrm{v_1} \leq_\mathrm{B} (\mathrm{x'}, \mathrm{y_i})$ and $(\mathrm{u'}, \mathrm{w'}) \leq_\mathrm{B} \mathrm{v_2}$. But $\mathrm{u} \leq_\mathrm{B} (\mathrm{x'}, \mathrm{y_i})$ would imply that $\mathrm{u} \leq_\mathrm{A} (\mathrm{x}, \mathrm{y_i})$, which violates the safeness of $\mathrm{F_A}$. Hence, $\mathrm{u} \leq_\mathrm{C} \mathrm{v}$ only if $\mathrm{u} \leq_\mathrm{A} \mathrm{v}$. Finally, pruning $\mathrm{F_C}$ does not remove dependants of u, since $\mathrm{u} \leq_\mathrm{B} \mathrm{v}$ implies that $\mathrm{u'} \not\leq_\mathrm{B} \mathrm{v}$. □

The above lemma shows that the universal side of a cut remains unchanged during the individual reduction steps of a compound step. As a consequence of this (and of safeness), none of the existential edges of the original cut is duplicated during reduction. Removing them one by one is thus entirely predictable.

Proposition 20. *A compound reduction step* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\rightarrow} \mathrm{F_B}$ *on a cut with* $n \geq 1$ *existential branches consists of* $2n - 1$ *reduction steps. If also* $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow\!\!\!\!\!\rightarrow} \mathrm{F_C}$ *then* $\mathrm{F_B} = \mathrm{F_C}$ *(modulo the names of vertices and eigenvariables).*

Proof. The first statement is immediate from the previous lemma and the safeness of $\mathrm{F_A}$. The second statement follows if also the local nature of the logical step is taken into account. □

The second statement of the above proposition is not essential to normalisation; rather, it serves to illustrate that compound reduction steps are a natural modification of the reduction relation. Since the branches of a node are unordered except by the dependency, reducing a cut all at once, rather than isolating an arbitrary branch, is more true to the idea of eliminating bureaucracy.

Figure 21 gives a schematic view of a compound reduction step. In the diagram, labels and witnesses, and also the substitutions, are omitted; the context Δ is duplicated in the right-hand side, while Γ merely has several representations.

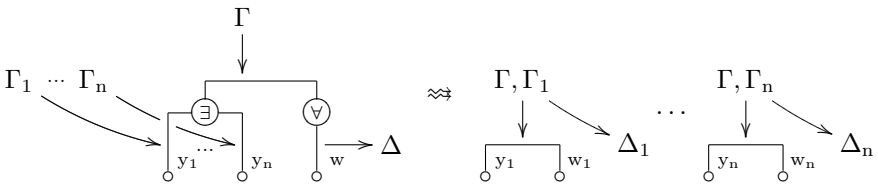


Figure 21: A compound reduction step

A compound step replaces a cut with cut-formula $C = \forall x.B$ by a number of cuts each with a cut-formula $B[t/x]$ for some term t. This strict reduction in formula complexity allows a remarkably easy proof of weak normalisation.

Theorem 21 (Weak normalisation)**.** *For any safe proof forest* $\mathrm{F_A}$ *there is a finite reduction path* $\mathrm{F_A} \rightsquigarrow^* \mathrm{F_B}$ *such that* $\mathrm{F_B}$ *is cut-free.*

Proof. Let the *complexity* $comp(\bot, \mathrm{c})$ of a cut $(\bot, C, \mathrm{c})$ be the number of quantifiers in C, and let the *measure* of a forest be the multiset of the complexities of all its cuts.

Given a forest $\mathrm{F_A}$ that is not cut-free, select a cut $(\bot, \mathrm{c}) \in \mathrm{E_A^\bot}$ that has no cut with equal or higher complexity amongst its dependants:

$$\forall \mathrm{e} \in \mathrm{E_A^\bot}.\ (\bot, \mathrm{c}) < \mathrm{e} \ \Rightarrow\ comp(\bot, \mathrm{c}) > comp(\mathrm{e})\ ;$$

by acyclicity of the dependency such a cut exists. Then if $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ the measure of $\mathrm{F_B}$ is strictly smaller, in the usual multiset ordering, than that of $\mathrm{F_A}$: the primary cut $(\bot, \mathrm{c})$ is replaced by several cuts of smaller complexity, and no cut of same or higher complexity is duplicated. The smallest measure, $\varnothing$, applies to forests that are cut-free. □

The reduction strategy employed in the above proof is simple and general: any cut whose dependants include only cuts of lower complexity may be reduced. Moreover, using the modified reduction algorithm the original counterexample in Figure 14 now strongly normalises, and no infinite reduction paths in this algorithm have yet been found. For these reasons the following conjecture is put forward.

Conjecture 22. *The relation* $(\rightsquigarrow)$ *is strongly normalising.*

6. Related Formalisms

The closest relatives of classical proof forests are Miller's expansion tree proofs, and the Herbrand nets under current investigation by McKinley. With the technical treatment of proof forests complete, a closer look will be taken at the differences with these other formalisms.

6.1. Expansion tree proofs

Two distinctions between proof forests and expansion tree proofs need no further explanation: expansion tree proofs allow higher-order and non-prenex formulae, but have not been given a normalisation procedure. The third important difference is in the correctness criteria employed.

Apart from the use of a dependency instead of a prenexification, expansion tree proofs are similar to Herbrand proofs: in particular the correctness condition, which corresponds to the expansion and substitution of the latter. The criterion used by Miller in [17] is differently formulated, but equivalent to the following: for every possibility of deleting one branch, ignoring the dependency, from each conjunction node, the disjunction over the remaining propositional positions should form a tautology.

In contrast, the actual correctness condition on proof forests uses the dependency, and not just the tree structure, to determine what is removed by a switching. The main consequence of the difference is that the present condition is preserved by the pruning operation, whereas the one used by Miller would not be. Given that pruning is essential to the modified reduction relation and that reductions need to preserve correctness, the current correctness condition is a crucial component of the weak normalisation result presented in this paper.

6.2. Herbrand nets

Like proof forests, the Herbrand nets currently under development by McKinley [16] are aimed at providing a canonical representation of first-order classical proof by removing bureaucracy, and share the same basic forest structure. Different from proof forests, in Herbrand nets the sequent calculus is taken as primary, and in particular the axiom rule—or in the first-order case, the tautology rule—is considered to contribute essential proof content. By adding *tautology links* corresponding to the tautology rule of Figure 3, Herbrand nets provide a notion of proof net for a first-order sequent calculus similar to the one in Figure 3 with cuts.

The technical distinctions between the two formalisms can mostly be ascribed to two properties, required of Herbrand nets in order to be a reasonable notion of proof net: one, translation from nets to sequent proofs should be invertible up to permutations; two, this translation should commute with reductions in either formalism. To achieve invertible translation the main ingredient, and the main distinction with proof forests, is invertible composition.

The tautology links of Herbrand nets are reminiscent of the axiom links found in other forms of proof net. Tautology links connect several propositional nodes, and a node can be connected by multiple links. Figure 22 illustrates a forest with two tautology links, drawn as indices to which the propositional nodes connect.

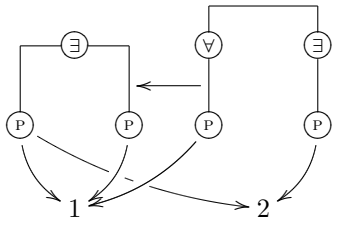


Figure 22: Tautology links on a proof forest

Another distinction is that the correctness criterion of Herbrand nets is a Danos–Regnier-style switching condition, familiar from MLL nets, on the dependency graph of a forest. Universal and existential nodes are switched, and each switching must give rise to an acyclic, connected subgraph. Based on the correctness criterion and the tautology links, Herbrand nets have a notion of *subnet* which allows the constituents of a composition to remain distinguishable, as the subnets on either side of a cut. This is sufficient to provide an inverse to composition.

Finally, reductions in Herbrand nets use the tautology links and subnets to determine what is duplicated in reductions, and in doing so reductions stay within the image of sequent calculus.

7. Conclusions

Classical proof forests, as presented here, are a natural system of proofs, based on a conceptually clear game-theoretic interpretation. Reduction steps are well motivated from this interpretation and, although complex, interact with the main structure of the forests, the dependency, in a natural way.

From this perspective, it is perhaps surprising that reductions are actually not that well behaved. Obtaining weak normalisation has required modifications to the reduction algorithm that are quite reasonable, but only in retrospect, and given the problems that arise. It is striking that reductions require explicit pruning in order to preserve correctness. To many, the failure of confluence will be less of a surprise; after all, classical proof is notorious in this respect.

On the other hand, the weak normalisation result is again elegant, and quite general. The measure involved is simple, while the dependency makes the only restriction to the reduction strategy, the prohibition to duplicate cuts of same or

greater complexity, straightforward. This raises a natural suspicion that strong normalisation is 'just around the corner'; for now, however, that is an open question.

One question that has not yet been addressed is whether the original reduction relation, while not strongly normalising, might be weakly normalising. This question was answered in the positive after this paper had been submitted. The proof will appear in my forthcoming thesis.

For future investigations, two important questions remain: one is that of strong normalisation, and the other is whether proof forests can be modified to obtain confluence. While the mechanics of non-confluence in proof forests are different from those in sequent calculus in an interesting way, their investigation has not given much hope of a confluent modification to the reduction relation.

Acknowledgements

I am greatly indebted to my supervisor Alex Simpson, for providing the foundations for this project as well as lending his continuing support and encouragement. Thanks also go to Richard McKinley for a fruitful exchange of both general ideas and concrete examples; and to Julian Gutierrez for introducing me to concurrency theory. Finally, thanks go to the anonymous referees for their constructive, helpful comments. The author was supported by a PH.D. studentship on EPSRC research grant EP/F042043/1

References

[1] S. Abramsky, Sequentiality vs. concurrency in games and logic, Mathematical Structures in Computer Science 13 (4) (2003) 531–565.

[2] G. Bellin, M. Hyland, E. Robinson, C. Urban, Categorical proof theory of classical propositional logic, Theoretical Computer Science 364 (2) (2006) 146–165.

[3] K. Brünnler, Locality for classical logic, Notre Dame Journal of Formal Logic 47 (4) (2006) 557–580.

[4] S. R. Buss, On herbrand's theorem, LNCS 960 (1995) 195–209.

[5] T. Coquand, A semantics of evidence for classical arithmetic, JSL 60 (1) (1995) 325–337.

[6] K. Došen, Identity of proofs based on normalisation and generality, The Bulletin of Symbolic Logic 9 (2003) 447–503.

[7] C. Führmann, D. Pym, Order-enriched categorical models of the classical sequent calculus, Journal of Pure and Applied Algebra 204 (2006) 21–78.

[8] G. Gentzen, Untersuchungen über das logische Schliessen I, II (1935), in M. E. Szabo (ed.), The Collected Papers of Gerhard Gentzen, pp. 68–131 (North-Holland, 1969).

[9] J.-Y. Girard, A new constructive logic: Classical logic, Mathematical Structures in Computer Science 1 (3) (1991) 255–296.

[10] A. Guglielmi, T. Gundersen, Normalisation control in deep inference via atomic flows, Logical Methods in Computer Science 4 (1:9) (2008) 1–36.

[11] S. Hetzl, A. Leitsch, Proof transformations and structural invariance, LNCS 4460 (2007) 201–230.

[12] D. J. D. Hughes, Proofs without syntax, Annals of Mathematics 164 (3) (2006) 1065–1076.

[13] R. Kuznets, Proof identity for classical logic: Generalizing to normality, in: Logical Foundations of Computer Science, 2007, pp. 332–348.

[14] F. Lamarche, L. Straßburger, Naming proofs in classical propositional logic, LNCS 3461 (2005) 246–261.

[15] R. McKinley, Categorical models of first-order proofs, Ph.D. thesis, University of Bath (2006).

[16] R. McKinley, Proof nets for Herbrand's Theorem, arXiv:1005.3986v1 (2010).

[17] D. A. Miller, A compact representation of proofs, Studia Logica 46 (4) (1987) 347–370.

[18] E. P. Robinson, Proof nets for classical logic, Journal of Logic and Computation 13 (5) (2003) 777–797.

[19] G. Winskel, Events in computation, Ph.D. thesis, University of Edinburgh (1980).

Appendix A: Postponed proofs

Proof of Proposition 8. If $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ then $\mathrm{F_B}$ is a proof forest.

$\mathrm{F_B}$ is straightforwardly seen to obey most conditions of Definition 2. The following details are treated explicitly.

2. All nodes in V are in legal configurations (see Figure 7).

The requirements in Figure 7 concerning labels and witnesses are easily verified. The other requirements fix the arity (the number of edges) of universal nodes ($\mathrm{V}^\forall$) and cut-nodes ($\mathrm{V}^\wedge$). Removal and duplication in disposal and structural steps (II and IV) affects only the arity of existential positions and $\bot$, since by condition 4 only edges in $\mathrm{E}^\exists$ or $\mathrm{E}^\bot$ are targets in ($\to$); other edges are removed or duplicated only along with their source nodes.

3. For an edge $\mathrm{e}_1 \in \mathrm{E}^\forall$ with $wit(\mathrm{e}_1) = a$ the following conditions hold:
   - a is not free in any formula in Γ,
   - $a \neq wit(\mathrm{e}_2)$ for any other edge $\mathrm{e}_2 \in \mathrm{E}^\forall$,
   - if $\mathrm{e}_2 \in \mathrm{E}^\exists \cup \mathrm{E}^\bot$ and $a \in FV(wit(\mathrm{e}_2))$ then $\mathrm{e}_1 \to \mathrm{e}_2$.

In a logical step the reorganisation of the dependency traces precisely the substitution $[t/a]$. Let $(\mathrm{x}, t, \mathrm{y})$ and $(\mathrm{u}, a, \mathrm{w})$ be as in Definition 5. If the eigenvariable of an edge $\mathrm{e}_1 \in \mathrm{E}^\forall_\mathrm{A}$ is free in t then $\mathrm{e}_1 \leq_\mathrm{A} (\mathrm{x}, \mathrm{y})$; for an edge e_2 where t is to be substituted either $\mathrm{e}_2 = (\bot, \mathrm{c})$ or $(\mathrm{u}, \mathrm{w}) \leq_\mathrm{A} \mathrm{e}_2$, and after reduction $\mathrm{e}_1 \leq_\mathrm{B} \mathrm{e}_2$.

5. The dependency ($\leq$) is a partial order (it is antisymmetric).

In a logical step, if $\mathrm{v} \leq_\mathrm{B} \mathrm{v}$ then by Lemma 6 either $\mathrm{v} \leq_\mathrm{A} \mathrm{v}$ or both $\mathrm{v} \leq_\mathrm{A} (\mathrm{x}, \mathrm{y})$ and $(\mathrm{u}, \mathrm{w}) \leq_\mathrm{A} \mathrm{v}$. The latter case would mean that $(\bot, \mathrm{c})$ has a bridge in $\mathrm{F_A}$. □

Proof of Proposition 12. If $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ and $\mathrm{F_A}$ is correct, then so is $\mathrm{F_B}$.

Let $(\bot, C, \mathrm{c})$ be the primary cut and let s be a switching for $\mathrm{F_B}$. Each of the four cases of Definition 5 will be addressed in turn.

I. If $\mathrm{F_A} \overset{\mathrm{c}}{\rightsquigarrow} \mathrm{F_B}$ is a propositional step ,there are two switchings for $\mathrm{F_A}$, with the following values of the switched forests:

$$\begin{array}{rclcrcl} s' & = & s \cup \{\mathrm{c} \mapsto \mathrm{L}\}; & \qquad & val(\mathrm{F}^{s'}_\mathrm{A}) & = & val(\mathrm{F}^s_\mathrm{B}) \vee C \\ s'' & = & s \cup \{\mathrm{c} \mapsto \mathrm{R}\}; & & val(\mathrm{F}^{s''}_\mathrm{A}) & = & val(\mathrm{F}^s_\mathrm{B}) \vee C^\bot \end{array}$$

If both values are tautologies, so is $val(\mathrm{F}^s_\mathrm{B})$.

For each of the three first-order reduction steps a switching s' for $\mathrm{F_A}$ will be given such that $val(\mathrm{F}^{s'}_\mathrm{A}) \Rightarrow val(\mathrm{F}^s_\mathrm{B})$. Let the branches of c be $(\mathrm{c}, U, \mathrm{u})$ and $(\mathrm{c}, X, \mathrm{x})$, as in Definition 5.

II. Choose $s' = s \cup \{\mathrm{c} \mapsto X\}$. Since x has no dependants and is not propositional, $val(\mathrm{F}_{\mathrm{A}}^{s'}) = val(\mathrm{F}_{\mathrm{B}}^{s})$.

III. Let edges $(\mathrm{v}, a, \mathrm{w})$ and $(\mathrm{x}, t, \mathrm{y})$ be as in Definition 5. There are two cases to consider.

1. The existential branch (x, y) is not switched off by another cut, i.e. there is no $\mathrm{e}_0 \in \mathrm{E}_{\mathrm{A}}^{s}$ such that $\mathrm{e}_0 <_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$.

Let $\mathrm{v} \in \mathrm{V}_{\mathrm{A}}^{\mathrm{p}}$ be an arbitrary propositional node that occurs in $\mathrm{F}_{\mathrm{A}}^{s}$. It will be shown that v also occurs in $\mathrm{F}_{\mathrm{B}}^{s}$. Assuming the contrary, suppose that $\mathrm{e} \leq_{\mathrm{B}} \mathrm{v}$ for some edge $\mathrm{e} \in \mathrm{E}_{\mathrm{B}}^{s}$. By Lemma 6, there are two possibilities.

- $\mathrm{e} = (\mathrm{c}, \mathrm{y})$ or $\mathrm{e} = (\mathrm{c}, \mathrm{w})$. Then $(\mathrm{c}, \mathrm{x}) \leq_{\mathrm{A}} \mathrm{v}$ or $(\mathrm{c}, \mathrm{u}) \leq_{\mathrm{A}} \mathrm{v}$.
- Otherwise e is unmodified in the reduction step and $\mathrm{e} \in \mathrm{E}_{\mathrm{A}}^{s}$. Since $\mathrm{e} \not\leq_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$ by 1, above, $\mathrm{e} \leq_{\mathrm{A}} \mathrm{v}$.

The second case is as follows.

2. There is some branch $\mathrm{e}_0 \in \mathrm{E}_{\mathrm{A}}^{s}$ such that $\mathrm{e}_0 <_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$. Then modify the switching so that it agrees with s except on c, where it indicates the existential branch:

$$s' = \{\mathrm{v} \mapsto s(\mathrm{v}) \mid \mathrm{v} \in \mathrm{V}_{\mathrm{A}}^{\wedge} \wedge \mathrm{v} \neq \mathrm{c}\} \cup \{\mathrm{c} \mapsto X\}$$

In this case all dependants of c in F_{A} are switched off, in both F_{A} and F_{B}. The details are as follows. Let $\mathrm{v} \in \mathrm{V}_{\mathrm{A}}^{\mathrm{p}}$ again be an arbitrary propositional node that occurs in $\mathrm{F}_{\mathrm{A}}^{s'}$ and suppose that $\mathrm{e} \leq_{\mathrm{B}} \mathrm{v}$ for some edge $\mathrm{e} \in \mathrm{E}_{\mathrm{B}}^{s}$. Lemma 6 gives the following options.

- $\mathrm{e} = (\mathrm{c}, \mathrm{y})$ and $\mathrm{y} \leq_{\mathrm{B}} \mathrm{v}$. By 2, $\mathrm{e}_0 \leq_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$, and by Lemma 6, $\mathrm{y} \leq_{\mathrm{A}} \mathrm{v}$.
- $\mathrm{e} = (\mathrm{c}, \mathrm{w})$ and $\mathrm{w} \leq_{\mathrm{B}} \mathrm{v}$. By 2, $(\mathrm{c}, \mathrm{u}) \in \mathrm{E}_{\mathrm{A}}^{s'}$, while $(\mathrm{u}, \mathrm{w}) \in \mathrm{E}_{\mathrm{A}}$ and $\mathrm{w} \leq_{\mathrm{A}} \mathrm{v}$.
- $\mathrm{e} \leq_{\mathrm{A}} (\mathrm{x}, \mathrm{y})$ and $(\mathrm{u}, \mathrm{w}) \leq_{\mathrm{A}} \mathrm{v}$. As above, $(\mathrm{c}, \mathrm{u}) \in \mathrm{E}_{\mathrm{A}}^{s'}$.
- Otherwise $\mathrm{e} \in \mathrm{E}_{\mathrm{A}}^{s'}$ and by Lemma 6, $\mathrm{e} \leq_{\mathrm{A}} \mathrm{v}$.

To summarise, if v is switched off by s in F_{B} it is switched off by s' in F_{A} as well. Since $\mathrm{V}_{\mathrm{A}}^{\mathrm{p}} = \mathrm{V}_{\mathrm{B}}^{\mathrm{p}}$ and the substitution $[t/a]$ is uniform, $val(\mathrm{F}_{\mathrm{A}}^{s'}) \Rightarrow val(\mathrm{F}_{\mathrm{B}}^{s})$.

IV. Let $(\mathrm{x}, \mathrm{y}_1), \ldots, (\mathrm{x}, \mathrm{y}_{\mathrm{n}})$ be as in Definition 5, as well as the substitution maps ρ, σ and τ. Three cases are distinguished.

1. The switching s for F_{B} selects the existential branch of both c and c′: $s(\mathrm{c}) = s(\mathrm{c}') = X$. Let s' on F_{A} agree with s: $s' = \{\mathrm{v} \mapsto s(\mathrm{v}) \mid \mathrm{v} \in \mathrm{V}_{\mathrm{A}}^{\wedge}\}$.
2. The switching s selects the universal branch at c: $s(\mathrm{c}) = X$. As above, $s' = \{\mathrm{v} \mapsto s(\mathrm{v}) \mid \mathrm{v} \in \mathrm{V}_{\mathrm{A}}^{\wedge}\}$.

3. The switching s selects the universal branch at c': $s(\mathrm{c}') = X$. Let $s' : \mathrm{V}_\mathrm{A}^\wedge \to \{\mathrm{L}, \mathrm{R}\}$ be as follows:

$$s'(\mathrm{v}) = \begin{cases} s(\mathrm{v}') & \text{if } \mathrm{v}' \in \mathrm{V}_\mathrm{B} \\ s(\mathrm{v}) & \text{otherwise.} \end{cases}$$

Let $\mathrm{v} \in \mathrm{V}_\mathrm{A}^\mathrm{p}$ be a propositional node in $\mathrm{F}_\mathrm{A}^{s'}$. First it will be shown for cases 1 and 2 that $\mathrm{v} \in \mathrm{F}_\mathrm{B}^s$. Since $\mathrm{V}_\mathrm{A}^\mathrm{p} \subseteq \mathrm{V}_\mathrm{B}^\mathrm{p}$ this requires only the following:

$$\exists \mathrm{e}_1 \in \mathrm{E}_\mathrm{B}^s.\ \mathrm{e}_1 \leq_\mathrm{B} \mathrm{v} \quad \Rightarrow \quad \exists \mathrm{e}_2 \in \mathrm{E}_\mathrm{A}^{s'}.\ \mathrm{e}_2 \leq_\mathrm{A} \mathrm{v}\ .$$

The edge e_1 is either an original one or a duplicated one. If it is original, then $\mathrm{e}_1 \in \mathrm{E}_\mathrm{A}^{s'}$ and $\mathrm{e}_1 \leq_\mathrm{A} \mathrm{v}$. If it is a duplicate, then by Lemma 7 it can only be $(\mathrm{c}', \mathrm{x}')$, since v is an original node. In case 1, $(\mathrm{c}, \mathrm{x}) \in \mathrm{E}_\mathrm{A}^s$ and $(\mathrm{c}, \mathrm{x}) \leq_\mathrm{A} \mathrm{v}$; in case 2, $(\mathrm{c}', \mathrm{x}') \notin \mathrm{E}_\mathrm{B}^s$.

For case 3, let v again be a propositional node in $\mathrm{F}_\mathrm{A}^{s'}$. It will be shown that if v is duplicated then v' is in F_B^s, and otherwise v will be.

If v is not duplicated, the argument is the same as above. Otherwise, $\mathrm{u} \leq_\mathrm{A} \mathrm{v}$; assuming that $\mathrm{e}_1 \leq_\mathrm{B} \mathrm{v}'$ for some $\mathrm{e}_1 \in \mathrm{E}_\mathrm{B}^s$, these are the possibilities.

- $\mathrm{e}_1 \in \mathrm{E}_\mathrm{A}$. Then $\mathrm{e}_1 \leq_\mathrm{A} \mathrm{v}$, and, by Lemma 7, $\mathrm{u} \not\leq \mathrm{e}_1$. Thus, there is no duplicate of the source of e_1, and s' agrees with s in such cases, which gives $\mathrm{e}_1 \in \mathrm{E}_\mathrm{A}^{s'}$.
- $\mathrm{e}_1 = \mathrm{e}'$ for some $\mathrm{e} \in \mathrm{E}_\mathrm{A}$. Then $\mathrm{e} \in \mathrm{E}_\mathrm{A}^{s'}$ and $\mathrm{e} \leq_\mathrm{A} \mathrm{v}$.

Since the substitution map σ is uniformly applied to all duplicate nodes, in all three cases $val(\mathrm{F}_\mathrm{A}^{s'}) \Rightarrow val(\mathrm{F}_\mathrm{B}^s)$. □

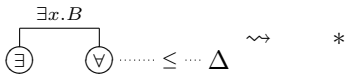
Appendix B: Reductions and Sequent Calculus

In this appendix the reduction steps of proof forests will be compared to structurally similar manipulations in the sequent calculus. For clarity, general contraction and weakening are used rather than those of Figure 3, which are restricted to existential formulae.

Above, a propositional step removes a propositional cut. The context of the forest remains intact; the asterisk indicates that nothing remains of the cut itself. Below, a corresponding rewrite in a sequent system is pictured.

$$\dfrac{\dfrac{}{\vdash \Gamma, C}\,\mathrm{Taut} \qquad \dfrac{}{\vdash C^{\perp}, \Gamma'}\,\mathrm{Taut}}{\vdash \Gamma, \Gamma'}\,\mathrm{Cut} \quad \Rightarrow \quad \dfrac{}{\vdash \Gamma, \Gamma'}\,\mathrm{Taut}$$

Secondly, a disposal step removes a cut on an existential leaf, and all dependants on the universal side, depicted as Δ, along with it.

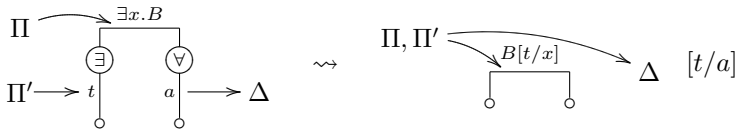


The corresponding step in sequent calculus is as follows:

$$\dfrac{\dfrac{\vdash \Gamma}{\vdash \Gamma, A}\,\mathrm{WR} \qquad \begin{matrix}\Delta\\ \vdash A^{\perp}, \Gamma'\end{matrix}}{\vdash \Gamma, \Gamma'}\,\mathrm{Cut} \quad \Rightarrow \quad \dfrac{\vdash \Gamma}{\vdash \Gamma, \Gamma'}\,\mathrm{WR}$$

The mechanics are similar: a cut on a weakening is reduced by removing the subproof on the opposite side. The other formulae in the removed subproof, depicted by Γ below, are introduced by weakenings in the result. A similar situation in proof forests arises when the last branch of an existential node is removed.

Thirdly, a logical reduction step is as follows:

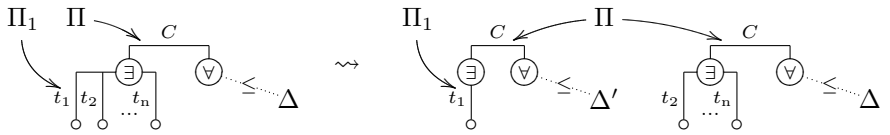


Its counterpart in sequent calculus is given below:

$$\dfrac{\dfrac{\begin{matrix}\Theta\\ \vdash B[t/x]\end{matrix}}{\vdash \Gamma, \exists x.B}\,\exists\mathrm{R} \qquad \dfrac{\begin{matrix}\Delta\\ \vdash B^{\perp}[a/x]\end{matrix}}{\vdash \Gamma', \forall x.B^{\perp}}\,\forall\mathrm{R}}{\vdash \Gamma, \Gamma'}\,\mathrm{Cut} \quad \Rightarrow \quad \dfrac{\begin{matrix}\Theta\\ \vdash B[t/x]\end{matrix} \qquad \begin{matrix}\Delta[t/a]\\ \vdash B^{\perp}[t/x]\end{matrix}}{\vdash \Gamma, \Gamma'}\,\mathrm{Cut}$$

The reduction of the cut itself and the substitution $[t/a]$ proceeds similarly in both systems. The changes in the dependency of the forest can be interpreted as changes in which inferences can be permuted in the sequent proof. The contexts represented by Δ in both illustrations almost, but not quite correspond; in the forest it indicates not all dependants, but only the targets in the relation $\to$.

Finally, the structural step in proof forests is pictured below:



In the forest proof the dependants of the universal node, indicated by Δ, are duplicated. The illustration below shows a similar transformation in the sequent calculus. A cut on a contracted formula (C on the left) is replaced by two cuts, while the subproof on the other side, Δ, is duplicated.

$$\dfrac{\dfrac{\overset{\Theta}{\vdash \Gamma, C, C}}{\vdash \Gamma, C}\mathrm{CR} \qquad \overset{\Delta}{\vdash C^{\perp}, \Gamma'}}{\vdash \Gamma, \Gamma'}\mathrm{Cut} \quad \Rightarrow \quad \dfrac{\dfrac{\overset{\Theta}{\vdash \Gamma, C, C} \qquad \overset{\Delta'}{\vdash C^{\perp}, \Gamma'}}{\vdash \Gamma, \Gamma', C}\mathrm{Cut} \qquad \overset{\Delta}{\vdash C^{\perp}, \Gamma'}}{\dfrac{\vdash \Gamma, \Gamma', \Gamma'}{\vdash \Gamma, \Gamma'}\mathrm{CR}}\mathrm{Cut}$$