Herbrand-Confluence for Cut Elimination in Classical First Order Logic

Stefan Hetzl, Lutz Straßburger

To cite this version:

Stefan Hetzl, Lutz Straßburger. Herbrand-Confluence for Cut Elimination in Classical First Order Logic. CSL 2012, Sep 2012, Fontainebleau, France. 10.4230/LIPIcs.CSL.2012.320 . hal-00759228

HAL Id: hal-00759228

https://hal.inria.fr/hal-00759228

Submitted on 3 Dec 2012

Herbrand-Confluence for Cut Elimination in Classical First Order Logic

Stefan Hetzl[1] and Lutz Straßburger[2]

1 **Institute of Discrete Mathematics and Geometry**
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
`hetzl@logic.at`

2 **INRIA Saclay – Île-de-France**
Ecole Polytechnique, LIX
Rue de Saclay, 91128 Palaiseau Cedex, France
`lutz@lix.polytechnique.fr`

Abstract

We consider cut-elimination in the sequent calculus for classical first-order logic. It is well known that this system, in its most general form, is neither confluent nor strongly normalizing. In this work we take a coarser (and mathematically more realistic) look at cut-free proofs. We analyze which witnesses they choose for which quantifiers, or in other words: we only consider the Herbrand-disjunction of a cut-free proof. Our main theorem is a confluence result for a natural class of proofs: all (possibly infinitely many) normal forms of the non-erasing reduction lead to the same Herbrand-disjunction.

1998 ACM Subject Classification F.4.1. Mathematical Logic, F.4.2. Grammars and Other Rewriting Systems, F.1.1. Models of Computation

Keywords and phrases proof theory, first-order logic, tree languages, term rewriting, semantics of proofs

Digital Object Identifier 10.4230/LIPIcs.CSL.2012.320

1 Introduction

The constructive content of proofs has always been a central topic of proof theory and it is also one of the most important influences that logic has on computer science. Classical logic is widely used and presents interesting challenges when it comes to understanding the constructive content of its proofs. These challenges have therefore attracted considerable attention, see, for example, [24, 11, 10], [6], [26, 27], [8], [21], or [5], for different investigations in this direction.

A well-known, but not yet well-understood, phenomenon is that a single classical proof usually allows several different constructive readings. From the point of view of applications this means that we have a choice among different programs that can be extracted. In [25] the authors show that two different extraction methods applied to the same proof produce two programs, one of polynomial and one of exponential average-case complexity. This phenomenon is further exemplified by case studies in [26, 3, 4] as well as the asymptotic results [2, 15]. The reason for this behavior is that classical "proofs often leave algorithmic detail underspecified" [1].

On the level of cut-elimination in the sequent calculus this phenomenon is reflected by the fact that the standard proof reduction without imposing any strategy is not confluent. In this paper we consider cut-elimination in classical first-order logic and treat the question which cut-free proofs one can obtain (by the strategy-free rewriting system) from a single proof with cuts. As our aim is to compare cut-free proofs we need a notion of equivalence of proofs: clearly the syntactic equality makes more differences than those which are mathematically interesting. Being in a system with quantifiers, a natural and more realistic choice is to consider two cut-free proofs equivalent if they choose the same terms for the same quantifiers, in other words: if they have the same Herbrand-disjunction.

A cut-reduction relation will then be called *Herbrand-confluent* if all its normal forms have the same Herbrand-disjunction. The main result of this paper is that, for a natural class of proofs, the standard reduction without erasing of subproofs is Herbrand-confluent. This result is surprising as this reduction is neither confluent nor strongly normalizing and may produce normal forms of arbitrary size (which—as our result shows—arise only from repetitions of the same instances).

As a central proof technique we use rigid tree languages which have been introduced in [19] with applications in verification (e.g. of cryptographic protocols as in [20]) as their primary purpose. To a proof we will associate a rigid tree grammar whose language is invariant under non-erasing cut-elimination and hence equal to the only obtainable Herbrand-disjunction. This property suggests the new notion of *Herbrand-content* of a proof, which is defined as the language of the grammar of the proof, and which is a strong invariant. A side effect of this proof technique is a combinatorial description of how the structure of a cut-free proof is related to that of a proof with cut. Such descriptions are important theoretical results which underlie applications such as algorithmic cut-introduction as in [18].

In Section 2 we briefly review the sequent calculus and cut-elimination for classical first-order logic. In Section 3 we describe regular and rigid tree grammars which we relate to proofs in Section 4. Section 5 is devoted to proving the invariance of the Herbrand-content under duplication of subproofs, and finally, in Section 6, we collect all results together.

2 Sequent Calculus and Cut-Elimination

For the sake of simplicity, we consider only a one-sided sequent calculus and formulas in negation normal form, but the results can be proved for a two-sided sequent calculus in the same way.

▶ **Definition 1.** A proof is a tree of multisets of formulas. Axioms are of the form $A, \overline{A}$ for A atomic (where $\overline{A}$ denotes the De Morgan-dual of A). The inference rules are:

$$\frac{\Gamma, A[x\backslash\alpha]}{\Gamma, \forall x\, A}\,\forall \qquad \frac{\Gamma, A[x\backslash t]}{\Gamma, \exists x\, A}\,\exists \qquad \frac{\Gamma, A, A}{\Gamma, A}\,\mathsf{c} \qquad \frac{\Gamma}{\Gamma, A}\,\mathsf{w} \qquad \frac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \wedge B}\,\wedge \qquad \frac{\Gamma, A, B}{\Gamma, A \vee B}\,\vee \qquad \frac{\Gamma, A \qquad \overline{A}, \Delta}{\Gamma, \Delta}\,\mathsf{cut}$$

where α is called *eigenvariable* and does not appear in $\Gamma, \forall x\, A$ and t does not contain a bound variable. We use the notation $[x\backslash\alpha]$ for the substitution that replaces x by the eigenvariable α. Similarly, $[x\backslash t]$ is the substitution that replaces x with t.

The explicitly mentioned formula in a conclusion of an inference rule, like $A \vee B$ for $\vee$ is called *main formula*. Analogously, the explicitly mentioned formulas in the premises of an inference rule, like A and B for $\vee$, are called *auxiliary formulas*. In the context of a concrete derivation we speak about *main* and *auxiliary occurrences* of inferences.

▶ **Definition 2.** A proof is called *regular* if different $\forall$-inferences have different eigenvariables.

We use the following convention: We use lowercase Greek letters $\alpha, \beta, \gamma, \delta, \ldots$ for *eigenvariables* in proofs, and $\pi, \psi, \ldots$ for proofs. For a proof π we write $\mathrm{EV}(\pi)$ for the set of

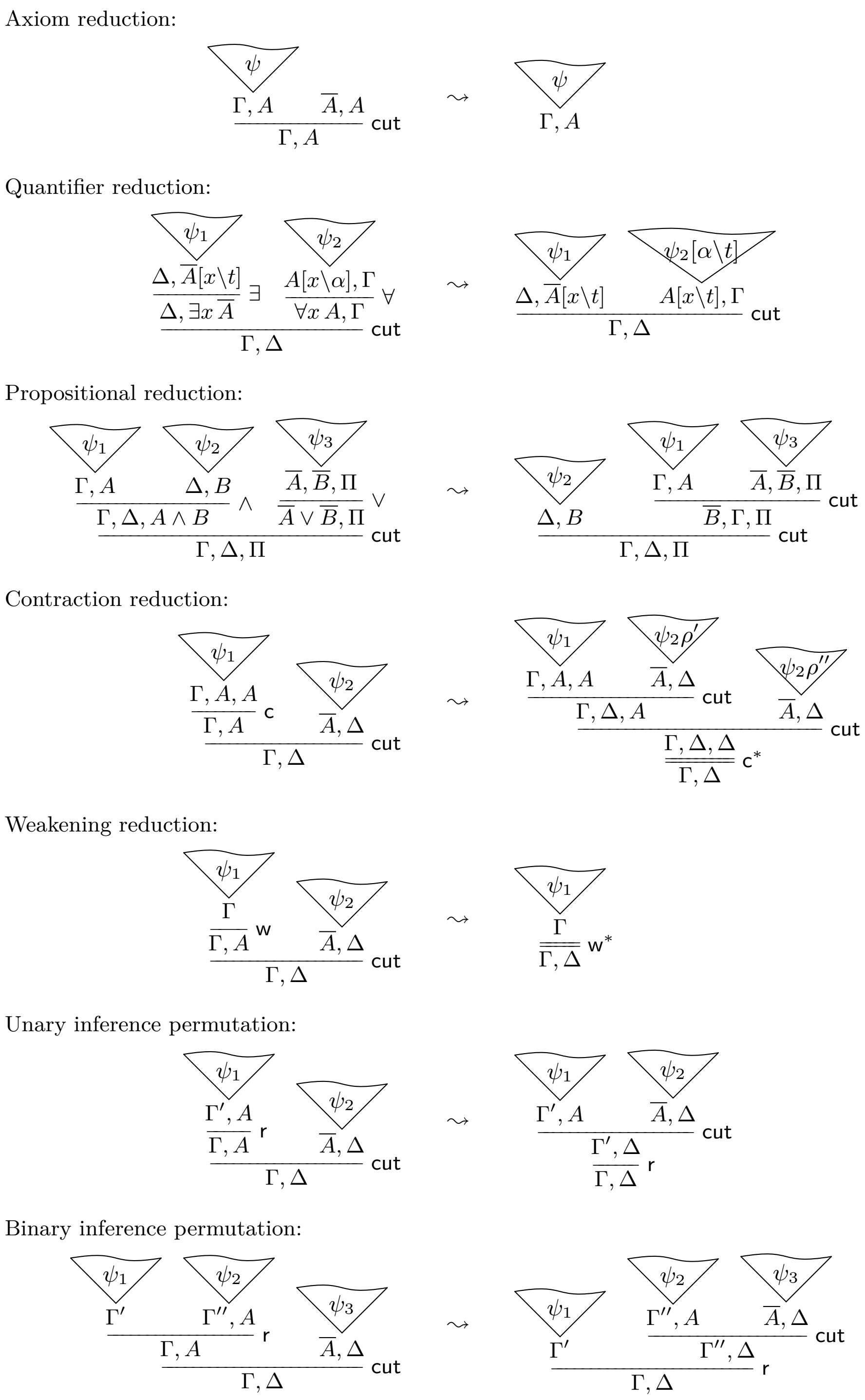


Figure 1 Cut-reduction steps.

eigenvariables of ∀-inferences of π. Furthermore, we write $|\pi|$ for the number of inferences in π. Our results do not depend on technical differences in the definition of the calculus (which in classical logic are inessential) such as the choice between multiplicative and additive rules and the differences in the cut-reduction induced by these choices. However, for the sake of precision, let us formally define the cut-reduction we use in this paper.

▶ **Definition 3.** Cut-reduction is defined on regular proofs and consists of the proof rewrite steps shown in Figure 1 (as well as all their symmetric variants), where in the contraction reduction step $\rho' = [\alpha \backslash \alpha']_{\alpha \in \mathrm{EV}(\psi_2)}$ and $\rho'' = [\alpha \backslash \alpha'']_{\alpha \in \mathrm{EV}(\psi_2)}$ are substitutions replacing each eigenvariable occurrence α in ψ_2 by fresh copies, i.e., α' and α'' are fresh for the whole proof. We write $\rightsquigarrow$ for the compatible (w.r.t. the inference rules), reflexive and transitive closure of $\rightsquigarrow$.

The above system for cut-reduction consists of purely local, minimal steps and therefore allows the simulation of many other reduction relations. We chose to work in this system in order to obtain invariance results of maximal strength. Among the systems that can be simulated literally are for example all color annotations of [11] in the multiplicative version of LK defined there. The real strength of the results in this paper lies however in the general applicability of the used proof techniques: the extraction of a grammar from a proof (that is described in the next sections) is possible in all versions of sequent calculus for classical logic and in principle also in other systems like natural deduction.

3 Regular and Rigid Tree Grammars

Formal language theory constitutes one of the main areas of theoretical computer science. Traditionally, a formal language is defined to be a set of strings but this notion can be generalized in a straightforward way to considering a language to be a set of first-order terms. Such tree languages possess a rich theory and many applications, see e.g. [13], [9]. In this section we introduce notions and results from the theory of tree languages that we will use for our proof-theoretic purposes.

A *ranked alphabet* Σ is a finite set of symbols which have an associated arity (their *rank*). We write $\mathscr{T}_\Sigma$ to denote the set of all finite trees (or terms) over Σ, and we write $\mathscr{T}_\Sigma(X)$ to denote the set of all trees over Σ and a set X of variables (seen as symbols of arity 0). We also use the notion of *position* in a tree, which is a list of natural numbers. We write ε for the empty list (the root position), and we write $p.q$ for the concatenation of lists p and q. we write $p \leq q$ if p is a prefix of q and $p < q$ if p is a proper prefix of q. Clearly, $\leq$ is a partial order and $<$ is its strict part. We write $\mathrm{Pos}(t)$ to denote the set of all position in a term $t \in \mathscr{T}_\Sigma(X)$.

▶ **Definition 4.** A *regular tree grammar* is a tuple $G = \langle N, \Sigma, \theta, P \rangle$, where N is a finite set of *non-terminal symbols*, and Σ is a ranked alphabet, such that $N \cap \Sigma = \emptyset$, θ is the *start symbol* with $\theta \in N$, and P is a finite set of production rules of the form $\beta \rightarrow t$ with $\beta \in N$ and $t \in \mathscr{T}_\Sigma(N)$.

The derivation relation $\rightarrow_G$ of a regular tree grammar $G = \langle N, \Sigma, \theta, P \rangle$ is defined as follows. We have $s \rightarrow_G r$ if there is a production rule $\beta \rightarrow t$ in P and a position $p \in \mathrm{Pos}(s)$, such that $s|_p = \beta$ and r is obtained from s by replacing β at p by t. The *language* of G is then defined as $L(G) = \{t \in \mathscr{T}_\Sigma \mid \theta \rightarrow^*_G t\}$, where $\rightarrow^*_G$ is the transitive, reflexive closure of $\rightarrow_G$. A *derivation* $\mathscr{D}$ of a term $t \in L(G)$ is a sequence $t_0 \rightarrow_G t_1 \rightarrow_G \ldots \rightarrow_G t_n$ with $t_0 = \theta$ and $t_n = t$. Note that a term t might have different derivations in G.

In [19] the class of rigid tree languages has been introduced with applications in verification (e.g. of cryptographic protocols as in [20]) as primary motivation. It will turn out that this class is appropriate for describing cut-elimination in classical first-order logic. In contrast to [19] we do not use automata but grammars—their equivalence is shown in [17].

▶ **Definition 5.** A *rigid tree grammar* is a tuple $\langle N, N_R, \Sigma, \theta, P \rangle$, where $\langle N, \Sigma, \theta, P \rangle$, is a regular tree grammar and $N_R \subseteq N$ is the set of *rigid non-terminals*. We speak of a *totally rigid tree grammar* if $N_R = N$. In this case we will just write $\langle N_R, \Sigma, \theta, P\rangle$.

A derivation $\theta = t_0 \rightarrow_G t_1 \rightarrow_G \ldots \rightarrow_G t_n = t$ of a rigid tree grammar $G = \langle N, N_R, \Sigma, \theta, P\rangle$ is a derivation in the underlying regular tree grammar satisfying the additional *rigidity condition*: If there are $i, j < n$, a non-terminal $\beta \in N_R$, and positions p and q such that $t_i|_p = \beta$ and $t_j|_q = \beta$ then $t|_p = t|_q$. The language $L(G)$ of the rigid tree grammar G is the set of all terms $t \in \mathscr{T}_\Sigma$ which can be derived under the rigidity condition. For a given derivation $\mathscr{D}\colon \theta = t_0 \rightarrow_G t_1 \rightarrow_G \ldots \rightarrow_G t_n = t$ and a non-terminal β we say that $p \in \mathrm{Pos}(t)$ is a *β-position in $\mathscr{D}$* if there is an $i \leq n$ with $t_i|_p = \beta$, i.e., either a production rule $\beta \rightarrow s$ has been applied at p in $\mathscr{D}$, or β occurs at position p in t. In the context of a given grammar G, we sometimes write $\mathscr{D}\colon \alpha \rightarrow^*_G t$ to specify that $\mathscr{D}$ is a derivation starting with α and ending with the term t.

▶ **Lemma 6.** *Let $G = \langle N, N_R, \Sigma, \theta, P\rangle$ be a rigid tree grammar and let $t \in L(G)$. Then there is a derivation $\theta \rightarrow_G \ldots \rightarrow_G t$ which uses at most one β-production for each $\beta \in N_R$.*

Proof. Given any derivation of t, suppose both $\beta \rightarrow s_1$ and $\beta \rightarrow s_2$ are used at positions p_1 and p_2 respectively. Then by the rigidity condition $t|_{p_1} = t|_{p_2}$ and we can replace the derivation at p_2 by that at p_1 (or the other way round). This transformation does not violate the rigidity condition because it only copies existing parts of the derivation. ◀

▶ **Lemma 7.** *Let $G = \langle N_R, \Sigma, \theta, P\rangle$ be a totally rigid tree grammar and $\theta \neq \beta \in N_R$, such that there is exactly one t with $\beta \rightarrow t$ in P. If $G' = \langle N_R \setminus \{\beta\}, \Sigma, \theta, (P \setminus \{\beta \rightarrow t\})[\beta\backslash t]\rangle$ then $L(G) = L(G')$.*

Proof. If a G-derivation of a term s uses β, it must replace β by t hence s is derivable using the productions of G' as well. The rigidity condition is preserved as the equality constraints of the G'-derivation are a subset of those of the G-derivation. Conversely, given a G'-derivation of a term s we obtain a derivation of s from the productions of G by replacing applications of $\delta \rightarrow r[\beta\backslash t]$ by $\delta \rightarrow r$ followed by a copy of $\beta \rightarrow t$ for each occurrence of β in r. Let $\gamma_1, \ldots, \gamma_n$ be the non-terminals that appear in t. By the rigidity condition for $i \in \{1, \ldots, n\}$ there is a unique term at all γ_i-positions in the derivation. Hence β fulfills the rigidity condition as well, and we have obtained a G-derivation of s. ◀

▶ **Lemma 8.** *If a rigid tree grammar G' is obtained from another rigid tree grammar G by deletion of production rules, then $L(G') \subseteq L(G)$.*

Proof. Every G'-derivation is a G-derivation. ◀

▶ **Notation 9.** For a given non-terminal β and a term t, we will write $\beta \in t$ or $t \ni \beta$ for denoting that β occurs in t.

▶ **Definition 10.** Let G be a tree grammar. A *path* of G is a list $\mathscr{P}$ of productions $\alpha_1 \rightarrow t_1$, $\ldots$, $\alpha_n \rightarrow t_n$ with $n \geq 1$ and $\alpha_{i+1} \in t_i$ for all $i \in \{1, \ldots, n-1\}$. The *length* of a path is $|\mathscr{P}| = n$. We will also write $\mathscr{P}\colon \alpha_1 \rightarrow t_1 \ni \alpha_2 \rightarrow \ldots \ni \alpha_n \rightarrow t_n$ to denote a path.

For a given path $\mathscr{P}\colon \alpha_1 \rightarrow t_1 \ni \alpha_2 \rightarrow \ldots \ni \alpha_n \rightarrow t_n$ we say that $\alpha_1, \ldots, \alpha_n$ are *on the path* $\mathscr{P}$ and write $\alpha_i \in \mathscr{P}$ for that. We also write $\mathscr{P}\colon \alpha_1 \dashrightarrow t_n$ and $\mathscr{P}\colon \alpha_1 \dashrightarrow \alpha_n$, if we do not want to explicitly mention the intermediate steps. For a fixed grammar G, we write $\alpha \dashrightarrow \beta$ to denote that there is a path $\mathscr{P}$ in G with $\mathscr{P}\colon \alpha \dashrightarrow \beta$.

For a set P of production rules, we write $\alpha \prec_P \beta$ (or simply $\alpha \prec \beta$, when P is clear from context) if there is a production $\alpha \rightarrow t$ in P with $\beta \in t$. We write $\prec^+$ for the transitive closure of $\prec$, and $\prec^*$ for its reflexive, transitive closure. Note that $\alpha \dashrightarrow \beta$ implies $\alpha \prec^+ \beta$, but not the other way around, since β could be a non-terminal with no production $\beta \to s$ in P.

▶ **Definition 11.** A tree grammar $\langle N, \Sigma, \theta, P\rangle$ is called *cyclic* if $\alpha \prec_P^+ \alpha$ for some $\alpha \in N$, and *acyclic* otherwise.

▶ **Lemma 12.** *If G is totally rigid and acyclic, then up to renaming of the non-terminals $G = \langle \{\alpha_1, \ldots, \alpha_n\}, \Sigma, \alpha_1, P\rangle$ with $L(G) = \{\alpha_1[\alpha_1\backslash t_1]\cdots[\alpha_n\backslash t_n] \mid \alpha_i \to t_i \in P\}$.*

Proof. Acyclicity permits a renaming of non-terminals, such that $\alpha_i \prec_P^+ \alpha_j$ implies $i < j$. Then $L(G) \supseteq \{\alpha_1[\alpha_1\backslash t_1]\cdots[\alpha_n\backslash t_n] \mid \alpha_i \to t_i \in P\}$ is obvious. For the left-to-right inclusion, let $\mathscr{D}\colon \alpha_0 = s_0 \to_G \ldots \to_G s_n = s \in \mathscr{T}_\Sigma$ be a derivation in G. By Lemma 6 we can assume that for each j at most one production whose left-hand side is α_j is applied, say $\alpha_j \to t_j$. By acyclicity we can rearrange the derivation so that $\alpha_j \to t_j$ is only applied after $\alpha_i \to t_i$ for all $i < j$. For those α_j which do not appear in the derivation we can insert any substitution without changing the final term so we obtain $s = \alpha_0[\alpha_0\backslash t_0]\cdots[\alpha_n\backslash t_n]$. ◀

This lemma entails that $|L(G)| \leq \prod_{i=1}^n |\{t \mid \alpha_i \to t \in P\}|$, in particular we are dealing with a finite language. The central questions in this context are (in contrast to the standard setting in formal language theory) not concerned with *representability* but with the *size of a representation.*

4 Proofs as Grammars

We will now restrict our attention to a certain class of proofs, called *simple proofs* below.

▶ **Definition 13.** A proof π is called *simple* if it is regular, the end-sequent is of the form $\exists x_1 \cdots \exists x_n\, A$ with A quantifier-free, and every cut in π whose cut-formula contains a quantifier is of the following form, where B is quantifier-free:

$$\dfrac{\Gamma, \exists x\, B \qquad \dfrac{\overline{B}[x\backslash\alpha], \Delta}{\forall x\, \overline{B}, \Delta}\,\forall}{\Gamma, \Delta}\,\mathsf{cut} \tag{1}$$

The above definition requires regularity which is a necessary assumption in the context of cut-elimination. The restriction of the end-sequent is done for expository purposes only, and can be extended to arbitrary sequents. The requirement of the $\forall$-rule being applied directly above the cut is natural as the rule is invertible. Moreover, any proof which does not fulfill this requirement can be pruned to obtain one that does, by simply permuting $\forall$-inferences down and identifying their eigenvariables when needed. The only significant restriction is that of disallowing quantifier alternations in the cut formulas. We conjecture that the central results extend to the general case. However, this will require the development of an adequate class of grammars.

▶ **Observation 14.** Simple proofs have the technically convenient property of exhibiting a 1-1 relationship between eigenvariables and cuts. For an eigenvariable α we will therefore write $\forall_\alpha$ for the inference introducing α and cut_α for the corresponding cut.

▶ **Definition 15.** Let π be a proof of $\exists x_1 \cdots \exists x_n\, A$ and let ψ be a subproof of π. The *Herbrand-set* $\mathrm{H}(\psi, \pi)$ *of* ψ *with respect to* π is defined as follows. If ψ is an axiom, then $\mathrm{H}(\psi, \pi) = \emptyset$. If ψ is of the form

$$\cfrac{\begin{array}{c}\psi' \\ \Gamma, A[x_n\backslash t]\end{array}}{\Gamma, \exists x_n\, A}\ \exists$$

then $\mathrm{H}(\psi, \pi) = \mathrm{H}(\psi', \pi) \cup \{A[x\backslash t]\}$. If ψ ends with any other unary inference and ψ' is its immediate subproof then $\mathrm{H}(\psi, \pi) = \mathrm{H}(\psi', \pi)$. If ψ ends with a binary rule and ψ' and ψ'' are its immediate subproofs, then $\mathrm{H}(\psi, \pi) = \mathrm{H}(\psi', \pi) \cup \mathrm{H}(\psi'', \pi)$. We write $\mathrm{H}(\pi)$ for $\mathrm{H}(\pi, \pi)$.

▶ **Definition 16.** Let Q be an occurrence of a formula $\exists x\, A$ in a proof. We define the set $\mathrm{tm}(Q)$ of *terms associated with* Q as follows: if Q is introduced as the main formula of a weakening, then $\mathrm{tm}(Q) = \emptyset$. If Q is introduced by a quantifier rule $\dfrac{\Gamma, A[x\backslash t]}{\Gamma, \exists x\, A}\ \exists$ then $\mathrm{tm}(Q) = \{t\}$. If Q is the main formula in the conclusion of a contraction, and Q_1 and Q_2 are the two occurrences of the same formula in the premise that are contracted, then $\mathrm{tm}(Q) = \mathrm{tm}(Q_1) \cup \mathrm{tm}(Q_2)$. In all other cases, an inference with the occurrence Q in the conclusion has a corresponding occurrence Q' of the same formula in one of its premises, and we let $\mathrm{tm}(Q) = \mathrm{tm}(Q')$.

▶ **Definition 17.** Let π be a simple proof, let $\alpha \in \mathrm{EV}(\pi)$, and let Q be the occurrence of the existentially quantified cut-formula in the premise of cut_α. Then we write $\mathrm{B}(\alpha)$ for the set $\{\ [\alpha\backslash t] \mid t \in \mathrm{tm}(Q)\ \}$ of substitutions and we define $\mathrm{B}(\pi) = \bigcup_{\alpha \in \mathrm{EV}(\pi)} \mathrm{B}(\alpha)$.

Structures similar to the above $\mathrm{B}(\pi)$ have been investigated also in [14] and [22] where they form the basis of proof net like formalisms using local reductions for quantifiers in classical first-order logic. Our aim in this work is however quite different: we use these structures for a global analysis of the sequent calculus.

▶ **Definition 18.** The *grammar of a simple proof* π is defined to be the totally rigid grammar $\mathrm{G}(\pi) = \langle N_R, \Sigma, \theta, P\rangle$ with

$$\begin{aligned} N_R &= \mathrm{EV}(\pi) \cup \{\theta\} \\ \Sigma &= \Sigma(\pi) \cup \{\wedge, \vee\} \\ P &= \{\theta \to A \mid A \in \mathrm{H}(\pi)\} \cup \{\alpha \to t \mid [\alpha\backslash t] \in \mathrm{B}(\pi)\} \end{aligned}$$

where $\Sigma(\pi)$ is the signature of π, the rank of $\wedge$ and $\vee$ is 2, and θ does not occur in π.

▶ **Lemma 19.** *If* π *is a simple proof, then* $\mathrm{G}(\pi)$ *is acyclic.*

Proof. By induction on the number of cuts in π. The grammar of a cut-free proof is trivially acyclic. For the induction step, let r be the lowest binary inference with subproofs π_1 and π_2 s.t. either (i) r is a cut or (ii) r is not a cut but both π_1 and π_2 contain at least one cut. Let P, P_1, and P_2 be the set of productions induced by the cuts in π, π_1, π_2, respectively. In case (ii), $\prec_P = \prec_{P_1} \cup \prec_{P_2}$, which is acyclic by induction hypothesis (since $\mathrm{EV}(\pi_1) \cap \mathrm{EV}(\pi_2) = \emptyset$). In case (i), let P_{r} be the productions induced by the cut r, then $\prec_P = \prec_{P_1} \cup \prec_{P_2} \cup \prec_{P_{\mathsf{r}}}$. By induction hypothesis, $\prec_{P_1}$ and $\prec_{P_2}$ are acyclic and as the cut-formula in r contains at most one quantifier, also $\prec_{P_{\mathsf{r}}}$ is acyclic. Therefore, a cycle in $\prec_P^+$ must be of the form $\alpha_1 \prec_{P_1}^* \beta_1 \prec_{P_{\mathsf{r}}} \alpha_2 \prec_{P_2}^+ \beta_2 \prec_{P_{\mathsf{r}}} \alpha_1$ where $\alpha_1, \beta_1 \in \mathrm{EV}(\pi_1)$ and $\alpha_2, \beta_2 \in \mathrm{EV}(\pi_2)$. However, r contains only one quantifier and depending on its polarity all productions in P_{r} lead from π_1 to π_2 or from π_2 to π_1 but not both, so $\prec_P$ is acyclic. ◀

We now come to a central definition of this paper.

▶ **Definition 20.** For a simple proof π, we define its *Herbrand-content* as $[\![\pi]\!] = L(\mathrm{G}(\pi))$.

Lemma 19 together with Lemma 12 implies that the Herbrand-content of a simple proof π with n cuts can be written as

$$[\![\pi]\!] = \{A[\alpha_1 \backslash t_1] \cdots [\alpha_n \backslash t_n] \mid A \in \mathrm{H}(\pi), [\alpha_i \backslash t_i] \in \mathrm{B}(\alpha_i)\}.$$

Note that for cut-free π we have $[\![\pi]\!] = \mathrm{H}(\pi)$, i.e. the Herbrand-content is nothing else but the Herbrand-disjunction induced by the proof. Furthermore, the Herbrand-content is a strong invariant: it is not changed by axiom reduction, propositional reduction and inference permutations as those transformations do not change the grammar. Furthermore, Lemma 7 shows that $[\![\pi]\!]$ is not changed by quantifier reduction and Lemma 8 shows that if $\pi \rightsquigarrow \pi'$ is a step of weakening reduction then $[\![\pi']\!] \subseteq [\![\pi]\!]$. A more difficult result is that the Herbrand-content is even invariant under the reduction of a contraction; the following section is devoted to proving this.

5 Invariance under Duplication

For simplifying the presentation, we assume in the following (without loss of generality) that the $\forall$-side is on the right of a cut and the $\exists$-side on the left. Then, a production $\beta \to t$ in $\mathrm{G}(\pi)$ corresponds to three inferences in π: a cut, an instance of the $\forall$-rule, and an instance of the $\exists$-rule, that we denote by cut_β, $\forall_\beta$, and $\exists_t$, respectively, and that are, in general, arranged in π as shown below.

$$\cfrac{\begin{array}{c}\cfrac{\Gamma', A[x\backslash t]}{\Gamma', \exists x\, A}\,\exists_t \\ \vdots \\ \Gamma, \exists x\, A\end{array} \qquad \begin{array}{c}\cfrac{\overline{A}[x\backslash\beta], \Delta'}{\forall x\, \overline{A}, \Delta'}\,\forall_\beta \\ \vdots \\ \forall x\, \overline{A}, \Delta\end{array}}{\Gamma, \Delta}\,\mathsf{cut}_\beta \tag{2}$$

The additional condition that $\forall_\beta$ is directly above cut_β, as indicated in (1) is only needed because in the following we make extensive use of Observation 14: there is a one-to-one correspondence between the cuts and the eigenvariables in π, and thus, the notation cut_β makes sense.

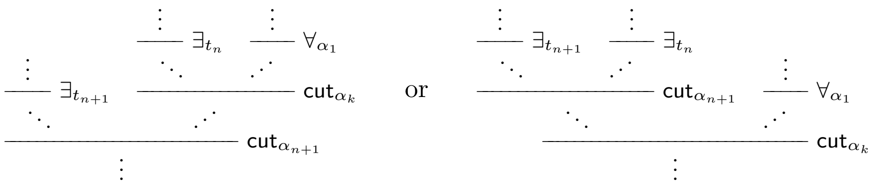
Furthermore, we say that the instances cut_β, $\forall_\beta$, and $\exists_t$ *are on a path* $\mathscr{P}$ *in* $\mathrm{G}(\pi)$ if the production $\beta \to t$ is in $\mathscr{P}$.

▶ **Definition 21.** Let π be a proof containing the configuration $\cfrac{\begin{array}{c}\cfrac{\vdots}{\ddots}\,\mathsf{r}_1\end{array} \quad \begin{array}{c}\cfrac{\vdots}{\cdot^{\cdot^{\cdot}}}\,\mathsf{r}_2\end{array}}{\vdots}\,\mathsf{r}_3$, where r_1, r_2, and r_3 are arbitrary rule instances, and r_3 is a branching rule, and r_1 and r_2 might or might not be branching. Then we say that r_1 is *on the left above* r_3, denoted by $\mathsf{r}_1 \Lsh \mathsf{r}_3$, and r_2 is *on the right above* r_3, denoted by $\mathsf{r}_3 \Rsh \mathsf{r}_2$, and r_1 and r_2 are *in parallel*, denoted by $\mathsf{r}_1 \Lsh\!\Rsh \mathsf{r}_2$.

▶ **Lemma 22.** *Let π be a simple proof and $\mathscr{P}\colon \alpha_1 \to t_1 \ni \alpha_2 \ldots \to t_n$ be a path in $\mathrm{G}(\pi)$. Then there is a $k \in \{1, \ldots, n\}$ s.t. cut_{α_k} is lowermost among all inferences on $\mathscr{P}$. Furthermore, $\forall_{\alpha_1}$ is on the right above cut_{α_k} and $\exists_{t_n}$ is on the left above cut_{α_k}.*

Proof. We proceed by induction on n. If $n = 1$, then $n = k = 1$. For the induction step consider a path $\alpha_1 \to t_1 \ni \ldots \ni \alpha_n \to t_n \ni \alpha_{n+1} \to t_{n+1}$. As $\alpha_{n+1} \in t_n$ we know that $\exists_{t_n}$

must be on the right above $\mathsf{cut}_{\alpha_{n+1}}$. By induction hypothesis there is a $k \in \{1, \dots, n\}$ such that we are in one of the following two situations



In the first case we let $l = n + 1$ and in the second we let $l = k$. In both cases cut_{α_l} has the desired properties. ◀

▶ **Lemma 23.** *Let π be a simple proof,* $\mathrm{G}(\pi) = \langle N_R, \Sigma, \theta, P\rangle$*, and* $\beta, \alpha \in \mathrm{EV}(\pi)$*. If* $\beta \dashrightarrow \alpha$ *then either* $\mathsf{cut}_\alpha \Lsh \mathsf{cut}_\beta$ *or* $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_\beta$ *or* $\mathsf{cut}_\alpha \Lsh\!\Rsh \mathsf{cut}_\beta$*.*

Proof. Since $\beta \dashrightarrow \alpha$, we have a path $\beta \to \dots \ni \alpha \to t$ for some t. By Lemma 22 there is a γ, such that $\exists_t \Lsh \mathsf{cut}_\gamma$ and $\mathsf{cut}_\gamma \Rsh \forall_\beta$, and such that cut_α and cut_β are not below cut_γ. Furthermore, cut_α must be below $\exists_t$, and cut_β below $\forall_\beta$. If $\gamma = \beta$, then $\mathsf{cut}_\alpha \Lsh \mathsf{cut}_\beta$. If $\gamma = \alpha$, then $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_\beta$. And if $\gamma \neq \beta$ and $\gamma \neq \alpha$, then $\mathsf{cut}_\alpha \Lsh\!\Rsh \mathsf{cut}_\beta$. ◀

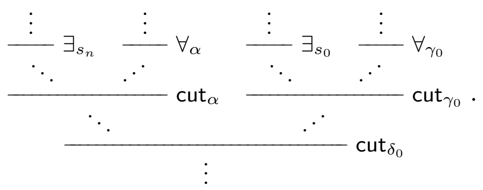
▶ **Lemma 24.** *Let* $\mathrm{G}(\pi) = \langle N_R, \Sigma, \theta, P\rangle$ *be the grammar of a simple proof π, such that there are two paths*

$$\beta \to t \ni \gamma_0 \to s_0 \ni \gamma_1 \to s_1 \ni \dots \to s_{n-1} \ni \gamma_n = \alpha \to s_n$$
$$\beta \to t \ni \delta_0 \to r_0 \ni \delta_1 \to r_1 \ni \dots \to r_{m-1} \ni \delta_m = \alpha \to r_m$$

such that γ_0 and δ_0 occur at two different positions in t. Then we have one of the following two cases:

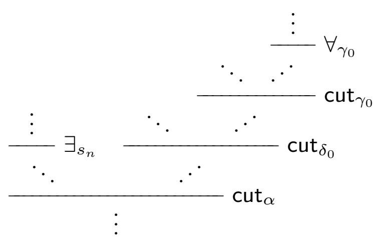
1. *we have $\gamma_i = \delta_j$ for some $0 \le i < n$ and $0 \le j < m$, or*
2. *for all $0 \le i < n$ and $0 \le j < m$ we have* $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\gamma_i}$ *and* $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_j}$*.*

Proof. Note that because of acyclicity of $\mathrm{G}(\pi)$, we have that $\beta \neq \gamma_i$ for all $i \le n$ and $\beta \neq \delta_j$ for all $j \le m$, in particular $\beta \neq \alpha$. Assume, for the moment, that $m, n > 0$; the case of one of them being zero will be treated at the very end of the proof. Then $\gamma_0 \neq \alpha$ and $\delta_0 \neq \alpha$. If $\gamma_0 = \delta_0$, we have case 1. So, assume also $\gamma_0 \neq \delta_0$. As $\beta \to t$ is a production in $\mathrm{G}(\pi)$, the proof π contains a formula which contains both γ_0 and δ_0 hence $\forall_{\gamma_0}$ and $\forall_{\delta_0}$ are not parallel. Since we have $\mathsf{cut}_{\gamma_0} \Rsh \forall_{\gamma_0}$ and $\mathsf{cut}_{\delta_0} \Rsh \forall_{\delta_0}$, we also have that cut_{γ_0} and cut_{δ_0} are not parallel. Without loss of generality, assume that cut_{δ_0} is below cut_{γ_0}. Then $\mathsf{cut}_{\delta_0} \Rsh \mathsf{cut}_{\gamma_0}$ (since $\mathsf{cut}_{\gamma_0} \Lsh \mathsf{cut}_{\delta_0}$ would entail $\forall_{\gamma_0} \Lsh\!\Rsh \forall_{\delta_0}$). Since we have $\delta_0 \dashrightarrow \alpha$, we can apply Lemma 23, giving us three possibilities:

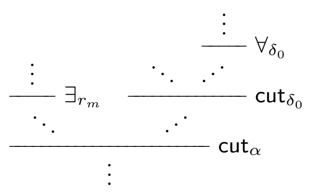
- If $\mathsf{cut}_\alpha \Lsh \mathsf{cut}_{\delta_0}$ then we have the situation  .

  By Lemma 22 applied to the path $\gamma_0 \dashrightarrow s_n$ we have that cut_{δ_0} must coincide with cut_{γ_i} for some $0 \le i < n$ (since π is a tree), so $\delta_0 = \gamma_i$ (by Observation 14), and we are in case 1.

- If $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_0}$ then we are in *both* of the following two situations:

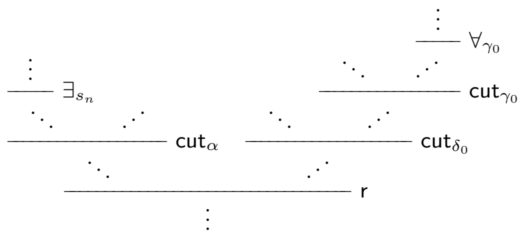


and

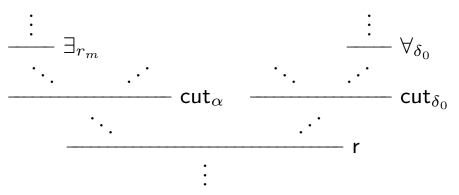


Thus, by Lemma 22 applied to the paths $\gamma_0 \dashrightarrow s_n$ and $\delta_0 \dashrightarrow r_m$ we know that $\mathsf{cut}_\alpha = \mathsf{cut}_{\gamma_k} = \mathsf{cut}_{\delta_l}$ for some $0 \le k \le n$ and $0 \le l \le m$ hence $\gamma_k = \alpha = \delta_l$. Furthermore $k = n$ and $l = m$ by acyclicity of $\mathrm{G}(\pi)$. Now consider any γ_i with $0 \le i < n$. Since $\gamma_i \dashrightarrow \alpha$, we can apply Lemma 23 and get either $\mathsf{cut}_\alpha \Lsh \mathsf{cut}_{\gamma_i}$ or $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\gamma_i}$ or $\mathsf{cut}_\alpha \Lsh\!\Rsh \mathsf{cut}_{\gamma_i}$. Since by Lemma 22 cut_{γ_i} must be above cut_α, we conclude $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\gamma_i}$. With the same reasoning we can conclude that $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_j}$ for all $0 \le j < m$. We are therefore in case 2.

- If $\mathsf{cut}_\alpha \Lsh\!\Rsh \mathsf{cut}_{\delta_0}$ then we are in *both* of the following two situations:

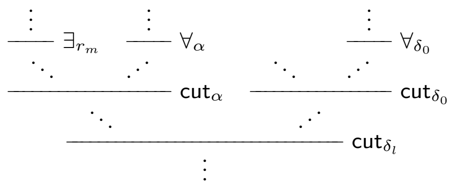


and



By Lemma 22 applied to the paths $\gamma_0 \to \ldots \to s_n$ and $\delta_0 \to \ldots \to r_m$, the rule r coincides with cut_{γ_i} and cut_{δ_j} for some $0 < i < n$ and $0 < j < m$, therefore $\gamma_i = \delta_j$ (by Observation 14), and we are in case 1.

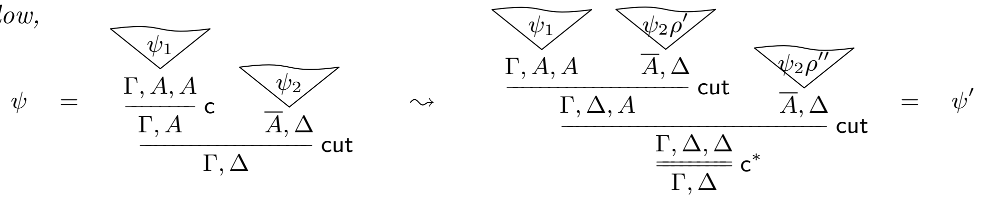
It remains to treat the case $n = 0$ or $m = 0$. If $m = n = 0$ then we are trivially in case 2 (there is no $0 \le i < n$ or $0 \le j < m$). If $n = 0$ and $m > 0$, we can apply Lemma 22 to the path $\delta_0 \to \ldots \to r_m$ and obtain an $l \in \{0, \ldots, m\}$ such that we are in the situation



But by the same argument as at the beginning of the proof, we also have that $\forall_\alpha$ and $\forall_{\delta_0}$ cannot be in parallel (α and δ_0 both appear in t), and therefore either $\mathsf{cut}_{\delta_0} \Rsh \mathsf{cut}_\alpha$ or $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_0}$. Since $\delta_0 \dashrightarrow \alpha$, the only possibility is $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_0}$, by Lemma 23. Thus $\mathsf{cut}_\alpha = \mathsf{cut}_{\delta_l}$, and therefore $l = m$ and we are in case 2. The case $m = 0$ and $n > 0$ is similar. ◀

The following is the main result of this section:

▶ **Proposition 25.** *Let π be a simple proof that contains a subproof ψ, shown on the left below,*



and let π' be the proof obtained from π from replacing ψ by ψ' shown on the right above, where $\rho' = [\alpha\backslash\alpha']_{\alpha\in\mathrm{EV}(\psi_2)}$ and $\rho'' = [\alpha\backslash\alpha'']_{\alpha\in\mathrm{EV}(\psi_2)}$ are substitutions that replace all eigenvariables in ψ_2 by fresh copies. Then $[\![\pi]\!] = [\![\pi']\!]$.

Proof. Let us first show $[\![\pi]\!] \subseteq [\![\pi']\!]$: write P for the productions of $\mathrm{G}(\pi)$ and P' for those of $\mathrm{G}(\pi')$. Let $F \in [\![\pi]\!] = L(\mathrm{G}(\pi))$ and $\mathscr{D}$ be its derivation. If the duplicated cut is quantifier-free, then $P' = P\rho' \cup P\rho''$ hence $\mathscr{D}\rho'$ (as well as $\mathscr{D}\rho''$) is a derivation of F in $\mathrm{G}(\pi')$. If the duplicated cut contains a quantifier, let α be its eigenvariable, let $t_1, \dots, t_k$ be its terms coming from the left copy of A and $t_{k+1}, \dots, t_n$ those from the right copy of A and let $Q = \{\alpha \to t_1, \dots, \alpha \to t_n\} \subseteq P$. We then have

$$P' = (P \setminus Q)\rho' \cup \{\alpha' \to t_1, \dots, \alpha' \to t_k\} \cup (P \setminus Q)\rho'' \cup \{\alpha'' \to t_{k+1}, \dots, \alpha'' \to t_n\} \quad .$$

If $\mathscr{D}$ does not contain α, then $\mathscr{D}\rho'$ (as well as $\mathscr{D}\rho''$) is a derivation of F in $\mathrm{G}(\pi')$. If $\mathscr{D}$ does contain α, then by Lemma 6 we can assume that it uses only one α-production, say $\alpha \to t_i$. If $1 \leq i \leq k$, then $\mathscr{D}\rho'$ is a derivation of F in $\mathrm{G}(\pi')$ and if $k < i \leq n$, then $\mathscr{D}\rho''$ is a derivation of F in $\mathrm{G}(\pi')$.

Let us now show $[\![\pi']\!] \subseteq [\![\pi]\!]$: let F be a formula in $[\![\pi']\!] = L(\mathrm{G}(\pi'))$, and let $\mathscr{D}'$ be a derivation of F in $\mathrm{G}(\pi')$. We construct $\mathscr{D} = \mathscr{D}'(\rho')^{-1}(\rho'')^{-1}$ by "undoing" the renaming of the variables in ψ_2. Then $\mathscr{D}$ is a derivation for F, using the production rules of $\mathrm{G}(\pi)$, but possibly violating the rigidity condition.

First, observe that only non-terminals $\alpha \in \mathrm{EV}(\psi_2)$ can violate the rigidity condition in $\mathscr{D}$: if $\beta \notin \mathrm{EV}(\psi_2)$ violates the rigidity condition then there are β-positions p_1, p_2 in $\mathscr{D}$ with $F|_{p_1} \neq F|_{p_2}$ and as $\beta\rho'\rho'' = \beta$ the positions p_1, p_2 are also β-positions in $\mathscr{D}'$ and they violate the rigidity condition in $\mathscr{D}'$ which is a contradiction to $\mathscr{D}'$ being a $\mathrm{G}(\pi')$-derivation.

Now define for each $\alpha \in \mathrm{EV}(\psi_2)$ the value $\mathbf{n}(\mathscr{D}, \alpha)$ to be the number of pairs $(p_1, p_2) \in \mathrm{Pos}(F) \times \mathrm{Pos}(F)$ where p_1 and p_2 are α-positions in $\mathscr{D}$ with $p_1 \neq p_2$ and $F|_{p_1} \neq F|_{p_2}$, and define $\mathbf{n}(\mathscr{D}) = \sum_{\alpha\in\mathrm{EV}(\psi_2)} \mathbf{n}(\mathscr{D}, \alpha)$. We proceed by induction on $\mathbf{n}(\mathscr{D})$ to show that $\mathscr{D}$ can be transformed into a derivation which does no longer violate rigidity. If $\mathbf{n}(\mathscr{D}) = 0$ then $\mathscr{D}$ obeys the rigidity condition, and we are done. Otherwise there is at least one $\alpha \in \mathrm{EV}(\psi_2)$ with $\mathbf{n}(\mathscr{D}, \alpha) > 0$. We now pick one such α which is minimal with respect to $\prec^*$ (which exists since $\mathrm{G}(\pi)$ is acyclic). Let p_1 and p_2 be α-positions in $\mathscr{D}$ with $p_1 \neq p_2$ and $F|_{p_1} \neq F|_{p_2}$, let p be the maximal common prefix of p_1 and p_2 and let q be the maximal prefix of p where a production rule has been applied in $\mathscr{D}$. Due to the tree structure of F, the position q is uniquely defined, and q is a β-position for some non-terminal β, and some production rule $\beta \to t$ has been applied at position q in $\mathscr{D}$, and we have two paths:

$$\beta \to t \ni \gamma_0 \to s_0 \ni \gamma_1 \to s_1 \ni \ldots \to s_{n-1} \ni \gamma_n = \alpha \to s_n$$
$$\beta \to t \ni \delta_0 \to r_0 \ni \delta_1 \to r_1 \ni \ldots \to r_{m-1} \ni \delta_m = \alpha \to r_m$$

where γ_0 and δ_0 occur at two different positions in t. Thus, we can apply Lemma 24, giving us the following two cases:

- We have $\gamma_i = \delta_j$ for some $0 \leq i < n$ and $0 \leq j < m$. Say $\eta = \gamma_i = \delta_j$, and let p_γ and p_δ be the positions of γ_i and δ_j (respectively) in $\mathscr{D}$. Since $\eta \prec^+ \alpha$ we know that η does not violate the rigidity condition (we chose α to be minimal), and therefore $F|_{p_\gamma} = F|_{p_\delta} = F'$. Let $\mathscr{D}_\gamma \colon \gamma_i \to^*_{\mathrm{G}(\pi)} F'$ and $\mathscr{D}_\delta \colon \delta_j \to^*_{\mathrm{G}(\pi)} F'$ be the two subderivations of $\mathscr{D}$ starting in positions p_γ and p_δ, respectively. Without loss of generality, we can assume that $\mathbf{n}(\mathscr{D}_\gamma) \leq \mathbf{n}(\mathscr{D}_\delta)$. Then let $\tilde{\mathscr{D}}$ be the derivation obtained from $\mathscr{D}$ by replacing $\mathscr{D}_\delta$ by $\mathscr{D}_\gamma$. Then $\tilde{\mathscr{D}}$ is still a derivation for F, but $\mathbf{n}(\tilde{\mathscr{D}}) < \mathbf{n}(\mathscr{D})$.
- For all $0 \leq i < n$ and $0 \leq j < m$ we have $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\gamma_i}$ and $\mathsf{cut}_\alpha \Rsh \mathsf{cut}_{\delta_j}$. So all inferences of the path $\gamma_0 \to \ldots \to s_{n-1}$ as well as all inferences of $\delta_0 \to \ldots \to r_{m-1}$ are in ψ_2. Therefore all variables of of these paths are in $\mathrm{EV}(\psi_2)$. As α violates the rigidity in $\mathscr{D}$

one of p_1, p_2 must be a α'-position and the other a α''-position in $\mathscr{D}'$ because $\mathscr{D}'$ does satisfy the rigidity condition. Without loss of generality we can assume that p_1 is the α'-position and p_2 the α''-position. As the paths are contained completely in ψ_2 we have $\gamma_0 \in \mathrm{EV}(\psi_2)\rho'$ and $\delta_0 \in \mathrm{EV}(\psi_2)\rho''$ which is a contradiction as no term can contain both a variable from $\mathrm{EV}(\psi_2)\rho'$ and one from $\mathrm{EV}(\psi_2)\rho''$. ◀

6 Herbrand-Confluence

We now turn to cut reduction sequences that start with a simple proof. All the reductions shown in Figure 1 preserve simplicity, except the following:

$$\dfrac{\dfrac{\cdots \quad \dfrac{\cdots}{\cdots}\,\forall_\alpha}{\cdots}\,\mathsf{cut}_\alpha \qquad \dfrac{\cdots}{\cdots}\,\forall_\beta}{\cdots}\,\mathsf{cut}_\beta \quad \rightsquigarrow \quad \dfrac{\cdots \qquad \dfrac{\dfrac{\cdots}{\cdots}\,\forall_\alpha \quad \dfrac{\cdots}{\cdots}\,\forall_\beta}{\cdots}\,\mathsf{cut}_\beta}{\cdots}\,\mathsf{cut}_\alpha$$

where cut_α is permuted down under cut_β (using the bottommost reduction in Fig. 1) and the cut formula of cut_β has its ancestor on the right side of cut_α. So in the following, when we speak about a *reduction sequence of simple proofs* we require that the above reduction is immediately followed by permuting $\forall_\alpha$ down as well, in order to arrive at

$$\dfrac{\cdots \qquad \dfrac{\dfrac{\cdots \quad \dfrac{\cdots}{\cdots}\,\forall_\beta}{\cdots}\,\mathsf{cut}_\beta}{\cdots}\,\forall_\alpha}{\cdots}\,\mathsf{cut}_\alpha$$

which is again simple. Recall that for our result this step is not strictly needed. We only add it here to simplify the presentation.

Collecting together the results proved in this paper we then obtain the following theorem.

▶ **Theorem 26.** *If $\pi \rightsquigarrow \pi'$ is a reduction sequence of simple proofs, then $[\![\pi]\!] \supseteq [\![\pi']\!]$.*

Proof. By induction on the length of the reduction $\pi \rightsquigarrow \pi'$ making a case distinction on the applied reduction step. If $\pi_i \rightsquigarrow \pi_{i+1}$ is a propositional reduction, an axiom reduction or a rule permutation, we even have $\mathrm{G}(\pi_i) = \mathrm{G}(\pi_{i+1})$. If it is a quantifier reduction, then $[\![\pi_i]\!] = [\![\pi_{i+1}]\!]$ by Lemma 7. If it is the reduction of a contraction, then $[\![\pi_i]\!] = [\![\pi_{i+1}]\!]$ by Proposition 25. If it is the reduction of a weakening, then $[\![\pi_i]\!] \supseteq [\![\pi_{i+1}]\!]$ by Lemma 8. ◀

▶ **Corollary 27.** *If $\pi \rightsquigarrow \pi'$ is a reduction sequence of simple proofs and π' is cut-free, then* $\mathrm{H}(\pi') \subseteq [\![\pi]\!]$.

This corollary shows that $[\![\pi]\!]$ is an upper bound (w.r.t. the subset relation) on the Herbrand-disjunctions obtainable by cut-elimination from π. Let us now compare this result with another upper bound that has previously been obtained in [16]. To that aim let $\mathrm{G}_0(\pi)$ denote the regular tree grammar underlying $\mathrm{G}(\pi)$ which can be obtained by setting all non-terminals to non-rigid. In this notation, a central result of [16], adapted to this paper's setting of proofs of non-prenex formulas, is

▶ **Theorem 28.** *If $\pi \rightsquigarrow \pi'$ and π' is cut-free, then* $\mathrm{H}(\pi') \subseteq L(\mathrm{G}_0(\pi))$.

While the above theorem 28 applies also to non-simple proofs, Corollary 27 is stronger in several respects:

First, the size of the Herbrand-content is by an exponential smaller than the size of the bound given by Theorem 28. Indeed, it is a straightforward consequence of Lemma 12 that the language of a totally rigid acyclic tree grammar with n production rules is bound by n^n. On the other hand, there are acyclic regular tree grammars G_n with $2n$ productions and $|L(G_n)| = n^{n^n}$ (by encoding in G_n the construction of a tree of depth n and branching degree n with an independent choice between n constant symbols at each leaf). These grammars can be obtained from appropriately constructed proofs.

Secondly, the class of totally rigid acyclic tree grammars can be shown to be in exact correspondence with the class of simple proofs in the following sense. Not only can we use a totally rigid acyclic tree grammar to simulate the process of cut-elimination, we can also—in the other direction—use cut-elimination to simulate the process of calculating the language of a grammar. It is shown in [17] how to transform an arbitrary acyclic totally rigid tree grammar G into a simple proof that has a $\rightsquigarrow$ normal form whose Herbrand-disjunction is essentially the language of G.

The third and—for the purposes of this paper—most important difference is that the bound of Corollary 27 is *tight* (in a sense that we are going to make precise now). This property of the Herbrand-content leads naturally to a confluence result for classical logic.

For tightening this bound, a first obvious observation is that there is no mechanism for deletion in the grammar but there is one in cut-elimination: the reduction of weakening. So, any cut-elimination strategy that does exactly compute $[\![\pi]\!]$ must be non-erasing. Consequently we define the *non-erasing cut-reduction* $\overset{ne}{\rightsquigarrow}$ as $\rightsquigarrow$ without the reduction rule for weakening. Note that a $\overset{ne}{\rightsquigarrow}$-normal form π is an analytic proof as well, e.g. $\mathrm{H}(\pi)$ is a (tautological!) Herbrand-disjunction. In contrast to a $\rightsquigarrow$-normal form (which might contain implicit redundancy) a $\overset{ne}{\rightsquigarrow}$-normal form might also contain explicit redundancy in the form of cuts whose cut-formulas are introduced by weakening on one or on both sides. Non-erasing reduction is also of interest in the context of the λ-calculus where it is often considered in the form of the λI-calculus and gives rise to the conservation theorem (see Theorem 13.4.12 in [7]). Our situation here is however quite different: neither $\rightsquigarrow$ nor $\overset{ne}{\rightsquigarrow}$ is confluent and neither of them is strongly normalizing. Nevertheless we obtain:

▶ **Theorem 29.** *If $\pi \overset{ne}{\rightsquigarrow} \pi'$ is a reduction sequence of simple proofs, then $[\![\pi]\!] = [\![\pi']\!]$.*

Proof. Inspection of the proof of Theorem 26 shows that the reduction of weakening is the only step that does not preserve the Herbrand-content. ◀

▶ **Definition 30** (Herbrand-confluence)**.** A relation $\longrightarrow$ on a set of proofs is called *Herbrand-confluent* iff $\pi \longrightarrow \pi_1$ and $\pi \longrightarrow \pi_2$ with π_1 and π_2 being normal forms for $\longrightarrow$ implies that $\mathrm{H}(\pi_1) = \mathrm{H}(\pi_2)$.

▶ **Corollary 31.** *The relation $\overset{ne}{\rightsquigarrow}$ is Herbrand-confluent on the set of simple proofs.*

How do these results fit together with $\overset{ne}{\rightsquigarrow}$ being neither confluent nor strongly normalizing? In fact, note that it is possible to construct a simple proof which permits an infinite $\overset{ne}{\rightsquigarrow}$ reduction sequence from which one can obtain normal forms of arbitrary size by bailing out from time to time. This can be done by building on the propositional double-contraction example found e.g. in [11, 12, 26] and in a similar form in [28]. While these infinitely many normal forms do have pairwise different Herbrand-disjunctions when regarded as *multisets*, Corollary 31 shows that as *sets* they are all the same. This observation shows that the lack of strong normalization is taken care of by using sets instead of multisets as data structure. But what about the lack of confluence? Results like [2] and [15] show that the number of $\rightsquigarrow$ normal forms with different Herbrand-disjunctions can be enormous. On the other hand we have just seen that $\overset{ne}{\rightsquigarrow}$ induces only *a single* Herbrand-disjunction: $[\![\pi]\!]$. The relation between $[\![\pi]\!]$ and the many Herbrand-disjunctions induced by $\rightsquigarrow$ is explained by Corollary 27: $[\![\pi]\!]$ contains them all as subsets.

7 Conclusion

We have shown that non-erasing cut-elimination for the class of simple proofs is Herbrand-confluent. While there are different and possibly infinitely many normal forms, they all induce the same Herbrand-disjunction. This result motivates the definition of this unique Herbrand-disjunction as Herbrand-*content* of the proof with cut.

As future work, the authors plan to extend this result to arbitrary first-order proofs. The treatment of blocks of quantifiers is straightforward: the rigidity condition must be changed to apply to vectors of non-terminals. Treating quantifier alternations is more difficult: the current results suggest to use a *stack* of totally rigid tree grammars, each layer of which corresponds to one layer of quantifiers (and is hence acyclic). Concerning further generalizations, note that the method of describing a cut-free proof by a tree language is applicable to any proof system with quantifiers that has a Herbrand-like theorem, e.g., even full higher-order logic as in [23]. The difficulty consists in finding an appropriate type of grammars.

Given the wealth of different methods for the extraction of constructive content from classical proofs, what we learn from our work is this: the first-order structure possesses (in contrast to the propositional structure) a unique and canonical unfolding. The various extraction methods hence do not differ in the choice of how to unfold the first-order structure but only in choosing *which part* of it to unfold. We therefore see that the effect of the underspecification of algorithmic detail in classical logic is redundancy.

Acknowledgments

The authors would like to thank Paul-André Melliès for helpful comments on this work. The first author was supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme and by the projects I-603 N18 and P22028 of the Austrian Science Fund (FWF).

References

1 Jeremy Avigad. The computational content of classical arithmetic. In Solomon Feferman and Wilfried Sieg, editors, *Proofs, Categories, and Computations: Essays in Honor of Grigori Mints*, pages 15–30. College Publications, 2010.

2 Matthias Baaz and Stefan Hetzl. On the non-confluence of cut-elimination. *Journal of Symbolic Logic*, 76(1):313–340, 2011.

3 Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Cut-Elimination: Experiments with CERES. In Franz Baader and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR) 2004*, volume 3452 of *Lecture Notes in Computer Science*, pages 481–495. Springer, 2005.

4 Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. CERES: An Analysis of Fürstenberg's Proof of the Infinity of Primes. *Theoretical Computer Science*, 403(2–3):160–175, 2008.

5 Matthias Baaz and Alexander Leitsch. Cut-elimination and Redundancy-elimination by Resolution. *Journal of Symbolic Computation*, 29(2):149–176, 2000.

6 Franco Barbanera and Stefano Berardi. A Symmetric Lambda Calculus for Classical Program Extraction. *Information and Computation*, 125(2):103–117, 1996.

7 Hendrik Pieter Barendregt. *The Lambda Calculus*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1984.

8 Ulrich Berger, Wilfried Buchholz, and Helmut Schwichtenberg. Refined Program Extraction from Classical Proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.

9 H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata: Techniques and Applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.

10 Pierre-Louis Curien and Hugo Herbelin. The Duality of Computation. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00)*, pages 233–243. ACM, 2000.

11 Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A New Deconstructive Logic: Linear Logic. *Journal of Symbolic Logic*, 62(3):755–807, 1997.

12 Jean Gallier. Constructive Logics. Part I: A Tutorial on Proof Systems and Typed λ-Calculi. *Theoretical Computer Science*, 110(2):249–339, 1993.

13 Ferenc Gécseg and Magnus Steinby. Tree Languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages: Volume 3: Beyond Words*, pages 1–68. Springer, 1997.

14 Willem Heijltjes. Classical proof forestry. *Annals of Pure and Applied Logic*, 161(11):1346–1366, 2010.

15 Stefan Hetzl. The Computational Content of Arithmetical Proofs. to appear in the *Notre Dame Journal of Formal Logic.*

16 Stefan Hetzl. On the form of witness terms. *Archive for Mathematical Logic*, 49(5):529–554, 2010.

17 Stefan Hetzl. Applying Tree Languages in Proof Theory. In Adrian-Horia Dediu and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications (LATA) 2012*, volume 7183 of *Lecture Notes in Computer Science*. Springer, 2012.

18 Stefan Hetzl, Alexander Leitsch, and Daniel Weller. Towards Algorithmic Cut-Introduction. In *Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18)*, volume 7180 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2012.

19 Florent Jacquemard, Francis Klay, and Camille Vacher. Rigid tree automata. In Adrian Horia Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *Third International Conference on Language and Automata Theory and Applications (LATA) 2009*, volume 5457 of *Lecture Notes in Computer Science*, pages 446–457. Springer, 2009.

20 Florent Jacquemard, Francis Klay, and Camille Vacher. Rigid tree automata and applications. *Information and Computation*, 209:486–512, 2011.

21 Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics.* Springer, 2008.

22 Richard McKinley. Herbrand expansion proofs and proof identity. In *Classical Logic and Computation (CL&C) 2008, participant's proceedings*, 2008. available at `http://wwwhomes.doc.ic.ac.uk/~svb/CLaC08/programme.html`.

23 Dale Miller. A Compact Representation of Proofs. *Studia Logica*, 46(4):347–370, 1987.

24 Michel Parigot. $\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction. In Andrei Voronkov, editor, *Logic Programming and Automated Reasoning (LPAR) 1992*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

25 Diana Ratiu and Trifon Trifonov. Exploring the Computational Content of the Infinite Pigeonhole Principle. *Journal of Logic and Computation*, 22(2):329–350, 2012.

26 Christian Urban. *Classical Logic and Computation.* PhD thesis, University of Cambridge, October 2000.

27 Christian Urban and Gavin Bierman. Strong Normalization of Cut-Elimination in Classical Logic. *Fundamenta Informaticae*, 45:123–155, 2000.

28 J. Zucker. The Correspondence Between Cut-Elimination and Normalization. *Annals of Mathematical Logic*, 7:1–112, 1974.