Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh

Herbrand Confluence for First-Order Proofs with Π_2-Cuts

Abstract: To each first-order proof π with cuts of at most complexity Π_2, we assign a formal grammar in which the number of production rules is bounded by the size of π and the set of generated terms is finite and forms a Herbrand expansion for the end-sequent of π. Using these grammars we prove that all (possibly infinitely many) normal forms of π obtained by non-erasing cut reductions have the same Herbrand expansion.

Keywords: Cut elimination, First-order logic, Herbrand's theorem, Formal language theory

Mathematics Subject Classification 2010: 03F05, 68Q42

1 Introduction

In classical first-order logic a proof can be considered as being composed of two layers: on the one hand the terms by which quantifiers are instantiated, and on the other hand, the propositional structure. This separation is most clearly illustrated by Herbrand's theorem [Herbrand, 1930, Buss, 1995]: a formula is valid if and only if there is a finite expansion (of existential quantifiers to disjunctions and universal quantifiers to conjunctions of instances) which is a propositional tautology. Such Herbrand expansions can be transformed to and obtained from cut-free sequent calculus proofs in a quite straightforward way.

Standard cut reduction is, however, not confluent, i.e. it permits the computation of many essentially different cut-free proofs. It was shown in [Baaz and Hetzl, 2011] (for pure first-order logic) and in [Hetzl, 2012b] (for arithmetical theories) that the number of different Herbrand expansions obtainable from a single proof with cut grows at least as fast as the size of the cut-free proofs. Still, it is not clear whether these results can be strengthened to obtain even more normal forms. In particular, it is an open question whether in general cut-elimination can produce infinitely many different Herbrand expansions.

In [Hetzl and Straßburger, 2012, 2013] an upper bound for the obtainable normal forms has been provided for proofs with Π_1-cuts in the following strong sense:

Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh: Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Vienna, Austria,
email: bahareh.afshari@tuwien.ac.at, stefan.hetzl@tuwien.ac.at, graham.leigh@tuwien.ac.at

a proof π with Π_1-cuts induces a finite set $[\![\pi]\!]$ such that every cut-free proof π' obtained from π via standard cut elimination has a Herbrand expansion, $\mathcal{H}(\pi')$, which is contained in $[\![\pi]\!]$. Moreover, if π' is obtained from π by *non-erasing* reductions (reductions that do not eliminate sub-proofs) then we even have $\mathcal{H}(\pi') = [\![\pi]\!]$. Consequently all normal forms of the non-erasing reduction (of which there infinitely many) have the same Herbrand expansion. This property of classical logic has been called *Herbrand-confluence* in [Hetzl and Straßburger, 2012, 2013] and provides a general way of defining the computational content of a classical proof.

The present paper extends Herbrand-confluence to proofs with Π_2-cuts. To a simple Π_2-proof[1] π we associate a recursively defined tree grammar $\mathcal{G}_\pi$ whose set of production rules is bounded by the size of π and generates a finite language $\mathcal{L}(\mathcal{G}_\pi)$ satisfying the following confluence result.

Theorem 1.1. *Let $\pi_0, \pi_1, \ldots, \pi_k$ be a sequence of simple Π_2-proofs such that π_{i+1} is obtained from π_i by a standard reduction rule (see Figures 1 and 2) other than weakening reduction. Then $\mathcal{L}(\mathcal{G}_{\pi_0}) = Ł(\mathcal{G}_{\pi_k})$. In particular, if the proof π_k contains only quantifier-free cuts $\mathcal{L}(\mathcal{G}_{\pi_0}) = \mathcal{H}(\pi_k)$.*

Theorem 1.1 can be seen as a refinement of [Afshari et al., 2015, Theorem 2]. Therein each simple Π_2-proof π is associated an acyclic context-free tree grammar $\mathcal{F}_\pi$ such that for $\pi_0, \pi_1, \ldots, \pi_k$ being a reduction sequence (possibly allowing reduction of weakening), $\mathcal{L}(\mathcal{F}_{\pi_0}) \supseteq Ł(\mathcal{F}_{\pi_k})$. For simple proofs, the grammars defined here and in [Afshari et al., 2015] can be shown to have the same language. There are, however, a number of technical differences between the two grammars motivated by the combinatorial nature of proving Herbrand confluence. Most notably, $\mathcal{G}_\pi$ may be cyclic (but permit only 'well-founded' derivations).

The grammar $\mathcal{G}_\pi$ can be considered as a directed graph whose nodes are quantifier occurrences and whose edges describe the information flow between them. In this sense it is also similar to the graphical formalisms of [Heijltjes, 2010, McKinley, 2013]. Other related structures are proof nets, which capture information flow on the propositional level and have been extensively studied starting with [Girard, 1987], as well as the logical flow graphs used by Buss [Buss, 1991] in the solution of the k-provability problem and further investigated by Carbone (see e.g. [Carbone and Semmes, 2000]).

1 See Definition 2.1.

2 First-order logic

We work with a Tait-style sequent calculus for first-order logic with explicit weakening and contraction rules. Terms and formulæ of first-order logic are defined as usual using the connectives $\wedge$, $\vee$ and quantifiers $\forall$, $\exists$, as well as a selection of predicate and function symbols. We assume two sets of variable symbols, *free* variables, denoted α, β, etc., and *bound* variables, x, y, z, with a formula only able to contain the latter sort in bound contexts.

Upper-case Roman letters, A, B, etc. denote formulæ and upper-case Greek letters Γ, Δ, etc. will range over *sequents*, finite unordered collections of formulæ with possible repetition. We abbreviate by Γ, Δ the disjoint union of Γ and Δ; and Γ, A is shorthand for $\Gamma, \{A\}$. We write $\bar{A}$ to denote the dual of the formula A obtained by de Morgan laws, and $A[x/t]$ for the formula obtained from A by replacing x with the term t if this will not induce any variable capture, and A otherwise.

Table 1. Axioms and rules of sequent-calculus

$$A, \bar{A} \quad \text{(for } A \text{ an atomic formula)}$$

$$\frac{\Gamma, A, B}{\Gamma, A \vee B}\vee \qquad \frac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, A \wedge B}\wedge \qquad \frac{\Gamma, A[x/\alpha]}{\Gamma, \forall x A}\forall \qquad \frac{\Gamma, A[y/s]}{\Gamma, \exists y A}\exists$$

$$\frac{\Gamma}{\Gamma, \Delta}\,\text{w} \qquad \frac{\Gamma, \Delta, \Delta^*}{\Gamma, \Delta}\,\text{c} \qquad \frac{\Gamma, A \quad \Delta, \bar{A}}{\Gamma, \Delta}\,\text{cut}$$

A *proof* is a finite binary tree of sequents obtained from the axioms and rules laid out in Table 1. In the ($\forall$) rule, α is called the *eigenvariable* and must not appear in $\Gamma, \forall x A$. In the ($\exists$) rule the term s is assumed to be free for y. In the contraction rule (c), Δ^* denotes a distinct copy of Δ. In each inference rule, those formulæ which are explicitly mentioned in the premise are said to be *principal* in the rules applied, for example A and B are principal in ($\wedge$) rule, every formula from Δ^* is principal in (c), and there are no principal formulæ in the weakening rule (w).

We assume all proofs are *regular*, namely all quantifiers' eigenvariables are distinct and different from any free variables. $\mathrm{EV}(\pi)$ denotes the set of eigenvariables in π and $\pi[\alpha/t]$ is the result of replacing throughout the proof π each occurrence of the variable symbol α by the term t. We write $\pi \vdash \Gamma$ to express that π is a proof with Γ being the sequent appearing at the root of π. A *position* in a proof π is a finite binary sequence pointing to a node in the proof-tree π. $\mathrm{Pos}(\pi)$ denotes the set of all positions in π. For $p \in \mathrm{Pos}(\pi)$, $\pi|p$ denotes the subproof of π at position

p with the convention that $\pi|\langle\rangle = \pi$ and $\pi|ip = \pi'|p$, where π' is the immediate left (or only) subproof of π if $i = 0$ and the immediate right subproof otherwise. The *size* of π, denoted $|\pi|$, is the total number of inference rules and axioms in π.

In this paper we primarily consider the following class of first-order proofs.

Definition 2.1 (Simple formulæ and simple Π_2-proofs). *We call a formula* simple *if it is a prenex Π_2 or prenex Σ_2 formula with at most one universal and one existential quantifier. A* simple Π_2-proof *is a proof in which each sequent is a finite multiset of simple formulæ and every universally quantified formula appearing above a cut is principal in the inference directly after its introduction (which is either a cut or existential introduction).*

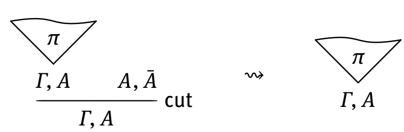
Lemma 2.2. *If $\pi \vdash \Gamma$ in which all cut formulæ in π and all formulæ in Γ are simple then there is a simple Π_2-proof $\pi' \vdash \Gamma$ such that $|\pi'| \leq |\pi|$.*

Proof: Apply inversion to all principal occurrences of universally quantified formulæ in π that appear above some cut to 'shift' the quantifier introduction rule $(\forall)$ downwards in the proof resulting in a simple Π_2-proof $\pi' \vdash \Gamma$. This operation will not introduce any new inference rules to π so $|\pi'| \leq |\pi|$. □

2.1 Cut reduction

The standard cut reduction rules are given in Figures 1 and 2.

Axiom:



Boolean:

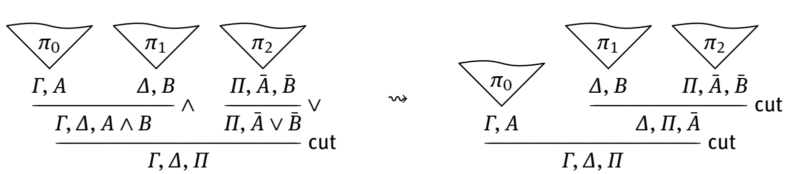


Fig. 1. One-step cut reduction and permutation rules I.

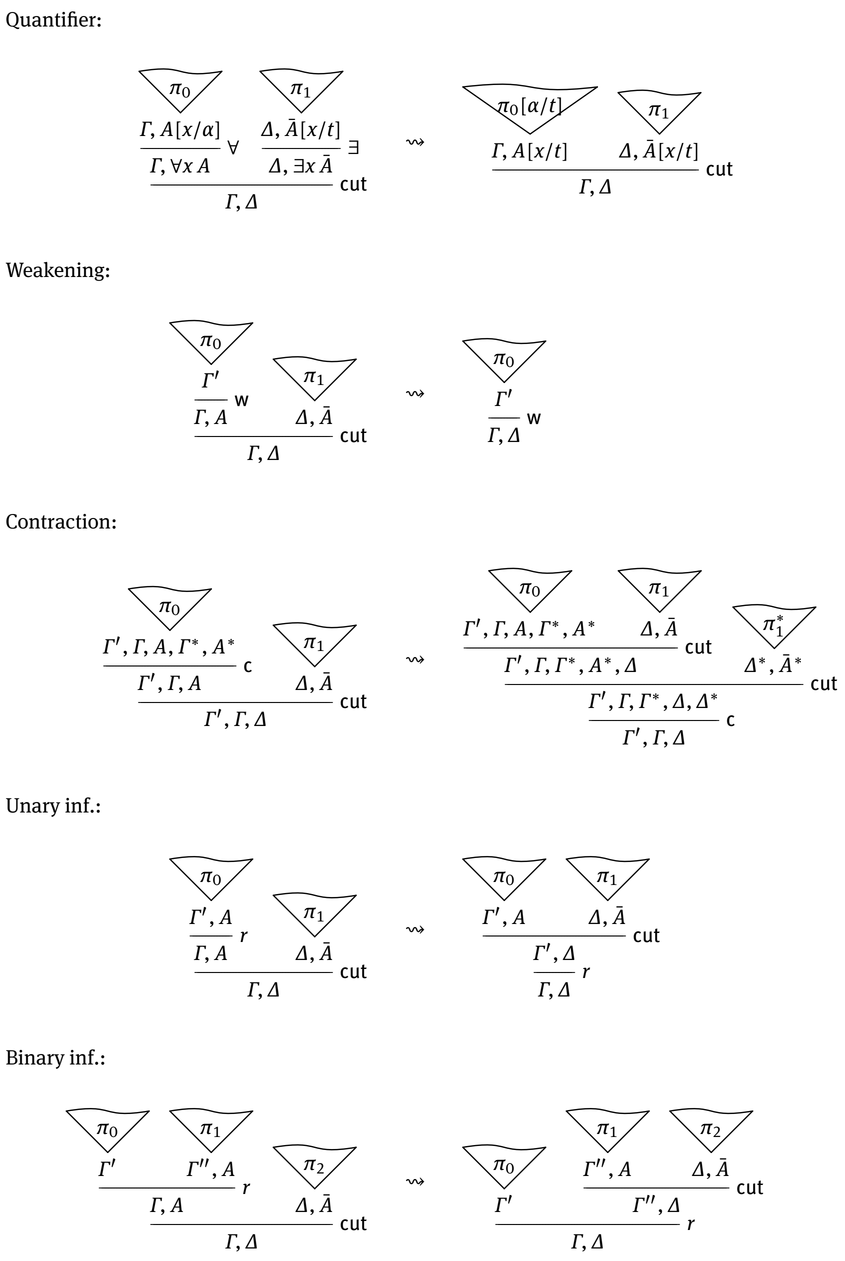


Fig. 2. One-step cut reduction and permutation rules II.

There is one reduction step that may not preserve simple Π_2-proofs, namely the particular instance of binary rule permutation in which (r) is (cut):

$$\dfrac{\overset{\pi_0}{\Gamma, A} \qquad \dfrac{\overset{\pi_1}{\Delta, \forall x B, \bar{A}} \qquad \overset{\pi_2}{\Lambda, \overline{\forall x B}}}{\Delta, \Lambda, \bar{A}}\,\text{cut}}{\pi \vdash \Gamma, \Delta, \Lambda}\,\text{cut}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\overset{\pi_0}{\Gamma, A} \qquad \overset{\pi_1}{\Delta, \forall x B, \bar{A}}}{\Gamma, \Delta, \forall x B}\,\text{cut} \qquad \overset{\pi_2}{\Lambda, \forall x B}}{\pi' \vdash \Gamma, \Delta, \Lambda}\,\text{cut}$$

In the right-hand proof, $\forall x B$ is principal in the cut but is not immediately preceded by the rule ($\forall$) introducing it, so is not a simple proof. If the left-hand proof is a simple Π_2-proof then it follows that π_1 has the form

$$\dfrac{\overset{\pi_1'}{\Delta, B[x/\beta], \bar{A}}}{\Delta, \forall x B, \bar{A}}\,\forall$$

whence we see that applying unary rule permutation to the upper cut in π' we may obtain the derivation

$$\dfrac{\dfrac{\dfrac{\overset{\pi_0}{\Gamma, A} \qquad \overset{\pi_1'}{\Delta, B[x/\beta], \bar{A}}}{\Gamma, \Delta, B[x/\beta]}\,\text{cut}}{\Gamma, \Delta, \forall x B}\,\forall \qquad \overset{\pi_2}{\Lambda, \forall x B}}{\pi'' \vdash \Gamma, \Delta, \Lambda}\,\text{cut}$$

We have $\pi \rightsquigarrow \pi' \rightsquigarrow \pi''$ and π'' is a simple Π_2-proof.

In order to permit reduction strategies of this form it is convenient to consider the reduction from π to π'' as a single reduction step, so we add the following additional rule to the definition of $\rightsquigarrow$:

$$\dfrac{\overset{\pi_0}{\Gamma, A} \qquad \dfrac{\dfrac{\overset{\pi_1}{\Delta, B[x/\beta], \bar{A}}}{\Delta, \forall x B, \bar{A}}\,\forall \qquad \overset{\pi_2}{\Lambda, \overline{\forall x B}}}{\Delta, \Lambda, \bar{A}}\,\text{cut}}{\pi \vdash \Gamma, \Delta, \Lambda}\,\text{cut}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\overset{\pi_0}{\Gamma, A} \qquad \overset{\pi_1}{\Delta, B[x/\beta], \bar{A}}}{\Gamma, \Delta, B[x/\beta]}\,\text{cut}}{\Gamma, \Delta, \forall x B}\,\forall \qquad \overset{\pi_2}{\Lambda, \forall x B}}{\pi'' \vdash \Gamma, \Delta, \Lambda}\,\text{cut} \tag{1}$$

Definition 2.3. *For proofs π, π' we write $\pi \rightsquigarrow \pi'$ if π' is the result of applying one of the rules of Figures 1 or 2, or* (1) *above to (a sub-proof of) π. Notice that in contraction reduction the left sub-proof is duplicated and care is taken to rename the eigenvariables (expressed by annotating the proof/sequent/formula in question by an asterisk) to maintain regularity.*

2.2 Herbrand's theorem

Suppose Γ is a set of simple formulæ and $\pi \vdash \Gamma$ is a proof in which all cuts are on non-quantified formulæ. For each $F \in \Gamma$ let the *Herbrand set* for F, denoted $\mathcal{H}(\pi, F)$, be the set of terms that occur in π as witnesses to the existential quantifier in F (if there is one). The *Herbrand set for π* is the set $\mathcal{H}(\pi) = \{(F, t) \mid F \in \Gamma \wedge t \in \mathcal{H}(\pi, F)\}$.

Given a set X of terms and simple formula F, let F^X denote the prenex Π_1 formula given by

$$(\forall x F)^X = \forall x F^X \qquad\qquad (\exists x F)^X = \bigvee_{t \in X} F[x/t]$$

and $F^X = F$ if F is quantifier-free.

Theorem 2.4 (Herbrand's theorem for simple formulæ). *Let Γ be a finite set of simple Σ_1 formulæ. $\bigvee \Gamma$ is valid iff there exist finite sets $\{X_F \subseteq \text{Terms} \mid F \in \Gamma\}$ such that the formula $\bigvee_{F \in \Gamma} F^{X_F}$ is valid.*

Proof: The right-to-left direction is immediate. For the left-to-right direction, suppose $\bigvee \Gamma$ is valid formula of first-order logic. Fix a cut-free proof $\pi \vdash \Gamma$ and for each $F \in \Gamma$, set $X_F = \mathcal{H}(\pi, F)$. By Gentzen's mid-sequent theorem π induces a proof of $\bigvee_{F\in\Gamma} F^{X_F}$ and we are done. □

3 Proof grammars

It this section we define a class of grammars suitable for the analysis of Π_2-proofs. Our definition of a grammar somehow deviates from a standard one as it allows certain (controlled) bounded non-terminals to be re-written in the derivations. It is, nevertheless, possible to provide an equivalent definition of these grammars in terms of (standard) context-free tree grammars [Afshari et al., 2015]. The presentation given in this paper has the advantage that the types of non-terminals are fixed from the outset allowing the verification of (proof-specific) language properties such as reductions proved in Section 5 to be clearer.

3.1 Terms, positions and substitution

Fix a ranked alphabet Σ and let $\mathcal{V}$ be a fixed set of variable symbols distinct from Σ. Let Term(Σ) denote the set of terms in the simply-typed λ-calculus built from $\Sigma \cup \mathcal{V}$. The set of *positions* of a term $T \in \mathrm{Term}(\Sigma)$, denoted $\mathrm{Pos}(T)$, are the nodes of the underlying tree, i.e.

$$\mathrm{Pos}(T) = \{\langle\rangle\} \cup \begin{cases} \emptyset, & \text{if } T \in \Sigma \cup \mathcal{V}, \\ \{ip \mid i \in \{0,1\} \wedge p \in \mathrm{Pos}(T_i)\}, & \text{if } T = T_0 \cdot T_1, \\ \{0p \mid p \in \mathrm{Pos}(T_0)\}, & \text{if } T = (\lambda\alpha.T_0). \end{cases}$$

For a term T and $p \in \mathrm{Pos}(T)$, we write $T|p$ for the subterm of T at position p.

Given $T \in \mathrm{Term}(\Sigma)$ and $\alpha \in \mathcal{V}$, let $\mathrm{Free}(T, \alpha) \subseteq \mathrm{Pos}(T)$ be the collection of positions at which α appears free in T:

$$\begin{aligned} \mathrm{Free}(\alpha, \alpha) &= \{\langle\rangle\} \\ \mathrm{Free}(\gamma, \alpha) &= \emptyset \quad \text{if } \alpha \neq \gamma \in \mathcal{V} \\ \mathrm{Free}(a, \alpha) &= \emptyset \quad \text{if } a \in \Sigma \\ \mathrm{Free}(T_1 \cdot T_2, \alpha) &= \{ip \mid i \in \{0,1\} \text{ and } p \in \mathrm{Free}(T_i, \alpha)\} \\ \mathrm{Free}(\lambda\gamma.T, \alpha) &= \begin{cases} \emptyset, & \text{if } \gamma = \alpha \\ \{0p \mid p \in \mathrm{Free}(T, \alpha)\}, & \text{if } \gamma \neq \alpha \end{cases} \end{aligned}$$

We introduce two forms of term substitution. Let $S, T \in \mathrm{Term}(\Sigma)$.

- $T[a/S]$, where $a \in \Sigma \cup \mathcal{V}$, denotes the *safe* substitution of S for a in T. So in particular, $(\lambda\gamma.T)[\gamma/S] = \lambda\gamma.T$ and for a distinct from γ, $(\lambda\gamma.T)[a/S] = \lambda\delta.(T[\gamma/\delta][a/S])$ where δ is a variable symbol not occurring free in S.
- $T[p/S]$, where $p \in \mathrm{Pos}(T)$, is defined by recursion on p:

$$T[\langle\rangle/S] = S \qquad a[p/S] = S, \quad a \in \Sigma \cup \mathcal{V}$$

$$(T_0 \cdot T_1)[0p/S] = T_0[p/S] \cdot T_1 \qquad (\lambda\alpha.T)[0p/S] = \lambda\alpha.T[p/S]$$

$$(T_0 \cdot T_1)[1p/S] = T_0 \cdot (T_1[p/S])$$

  Note that this may involve an *unsafe* substitution.

3.2 Syntax and semantics of proof grammars

A *proof grammar* is a tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{S}, \mathrm{Pr}\rangle$ where $\mathcal{N}$ is a set of typed *non-terminals* of order at most 1, $\mathcal{S} \subseteq \mathcal{N}$ is a set of *starting symbols* (of base type), Σ is a ranked alphabet, called *terminals*, disjoint from $\mathcal{N}$, and Pr consists of pairs $(a, T) \in \mathcal{N} \times \mathrm{Term}(\Sigma \cup \mathcal{N})$ (called *production rules* and written $a \to T$) such that a and T have the same type. Given a proof grammar $\mathcal{G}$ we assume $\mathcal{G} = \langle \mathcal{N}_\mathcal{G}, \Sigma_\mathcal{G}, \mathcal{S}_\mathcal{G}, \mathrm{Pr}_\mathcal{G}\rangle$.

Let d be a sequence $\langle \rho_i, p_i\rangle_{i<k}$ of pairs of production rules of a proof grammar $\mathcal{G}$ and positions, and S and T terms. We call d a *derivation from S to T*, written $d\colon S \to T$, if there exist terms $(N_i)_{i\leq k}$ such that $N_0 = S$, $N_k = T$, and for each $0 \leq i < k$,

1. ρ_i is a production rule of $\mathcal{G}$ and $p_i \in \mathrm{Pos}(N_i)$,
2. For $\rho_i = (a \to S)$, we have $N_i|p_i = a$ and $N_{i+1} = N_i[p_i/S]$.

The sequence of terms $(N_i)_{i\leq k}$ is uniquely determined by d and S, whence we may write $d(i)$ for N_i. The *length* of d, $\mathrm{lh}(d)$, is k. We write $\mathrm{Der}(\mathcal{G})$ for the set of derivations in $\mathcal{G}$, and say *T is derivable from S* if there exists a derivation $d\colon S \to T$. A derivation d *writes* a non-terminal a if there is a production rule of the form $a \to S$ for some S occurring in d.

The *language* of a proof grammar $\mathcal{G}$, $\mathcal{L}(\mathcal{G})$, is the set of terms not containing free occurrences of non-terminal symbols that are derivable from the starting symbols of the grammar:

$$Ł(\mathcal{G}) = \{T \mid \exists d\colon \sigma \to T \text{ s.t. } \sigma \in \mathcal{S}_\mathcal{G} \text{ and } T \text{ contains no free occurrence of a non-terminal}\}.$$

When comparing languages of proof grammars it is convenient to work modulo β-convertibility. For λ-terms S and T, we write $S \to_\beta T$ to abbreviate T is obtainable from S by one step β-reduction, and $\to_\beta^*$ for the reflexive transitive closure of this relation. Thus for grammars $\mathcal{G}_1$, $\mathcal{G}_2$, we write $\mathcal{L}(\mathcal{G}_1) \subseteq Ł(\mathcal{G}_2)$ to express that for every $S \in Ł(\mathcal{G}_1)$ there exists a $T \in Ł(\mathcal{G}_2)$ such that $T \to_\beta^* S$.

3.3 Rigidity

Rigid grammars were studied in [Hetzl, 2010, 2012a] for the construction of terms appearing in cut-elimination for first-order logic with Π_1-cuts. In this section we extend the notion of rigidity for the analysis of proofs with Π_2-cuts.

Let $\mathcal{G}$ be a proof grammar and suppose $\lhd$ is a transitive binary relation on $\mathcal{N}_\mathcal{G}$ and $\mathcal{R} \subseteq \mathcal{N}_\mathcal{G}$ is a designated set of non-terminals. A derivation $d = \langle (a_i \to S_i), q_i \rangle_{i<\mathrm{lh}(d)} : S \to T$ induces an equivalence relation on $\{i \mid i < \mathrm{lh}(d)\}$ corresponding to connectedness in parse trees: for $j_0, j_1 < \mathrm{lh}(d)$, set $j_0 \approx_d j_1$ iff there exist $i_0 \leq j_0, j_1$ such that

1. $q_{i_0} \leq q_{j_0}, q_{j_1}$,
2. for every $k \in \{0, 1\}$ and $i_0 < i < j_k < \mathrm{lh}(d)$, if $q_i \leq q_{j_k}$ and $a_i \in \mathcal{R}$ then $a_{j_k} \ntriangleleft a_i$.

In other words, two non-terminals occurring in (the natural tree representation of) the derivation d are considered connected if there is no non-terminal of higher priority between them and their closest common ancestor. We write $j_0 \sim_d j_1$ if $j_0 \approx_d j_1$ and in addition $a_{j_0} = a_{j_1} \in \mathcal{R}$. Notice that $\sim_d$ may not be an equivalence relation on $\{i \mid i < \mathrm{lh}(d)\}$. For instance, if $a_j \notin \mathcal{R}$ then $j \nsim_d j$.

We consider derivations that *respect* the relation $\sim_d$ and we permit unsafe substitutions that are *controlled* by priority ordering $\lhd$:

Definition 3.1 (Rigid derivations). *Let $\mathcal{G}$ be a proof grammar and suppose $\lhd$ is a transitive binary relation on $\mathcal{N}_\mathcal{G}$ and $\mathcal{R} \subseteq \mathcal{N}_\mathcal{G}$. A derivation $d = \langle a_i \to T_i, q_i \rangle_{i<k} : S \to T$ in $\mathcal{G}$ is* rigid with respect to $(\lhd, \mathcal{R})$ *if*

1. *for every $i, j < k$, $i \sim_d j$ implies $T|q_i = T|q_j$, and*
2. *for every $i < k$ with $a_i \in \mathcal{R}$, if $d(i)|q = \lambda a_i.S_0$ for some position $q < q_i$ and term S_0, then there exist position $q < q' < q_i$, term S_1 and variable b such that $d(i)|q' = \lambda b.S_1$ and $a_i \lhd b$.*

A *rigid* proof grammar is a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{R}, \lhd, \Sigma, \mathcal{S}, \mathrm{Pr} \rangle$ such that $\langle \mathcal{N}, \Sigma, \mathcal{S}, \mathrm{Pr} \rangle$ is a proof grammar, $\mathcal{R} \subseteq \mathcal{N}$ and $\lhd$ is a transitive relation on $\mathcal{N}$. $\mathcal{R} = \mathcal{R}_\mathcal{G}$ is the set of *rigid* non-terminals of $\mathcal{G}$ and $\lhd = \lhd_\mathcal{G}$ is the *priority ordering of* $\mathcal{G}$. $\mathcal{G}$ is *acyclic* if the ordering $\lhd_\mathcal{G}$ is acyclic, that is for all $a \in \mathcal{N}_\mathcal{G}$ it is not the case that $a \lhd_\mathcal{G} a$, and is *totally rigid* if $\mathcal{R}_\mathcal{G} = \mathcal{N}_\mathcal{G}$.

In a rigid proof grammar $\mathcal{G}$, a *derivation* is simply a derivation in the underlying proof grammar and a *rigid derivation* is a derivation which is rigid with respect to $(\lhd_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}})$.

A rigid derivation $d\colon S \to T$ (in $\mathcal{G}$) is *full* if there is no extension of d that is a rigid derivation in $\mathcal{G}$. The *language* of a rigid proof grammar $\mathcal{G}$, $\mathcal{L}(\mathcal{G})$, is the collection of terms derivable from full rigid derivations starting from $\mathcal{S}_{\mathcal{G}}$:

$$Ł(\mathcal{G}) = \{T \mid \exists d\colon \sigma \to T \text{ s.t. } \sigma \in \mathcal{S}_{\mathcal{G}} \text{ and } d \text{ is full and } (\lhd_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}})\text{-rigid}\}.$$

Note Definition 3.1 allows certain variables that are bounded (by an abstraction) to be re-written. Therefore in rigid derivations, arbitrary β-reductions are not permissible until the derivation is *fully* written in the sense given above.

Observation 3.2. *The language of a rigid proof grammar is a set of closed (well-typed) ground-type λ-terms.*

Observation 3.3 (Permutation). *Let d and d' be derivations in a rigid grammar $\mathcal{G}$ and suppose d' is a permutation of d. d' is rigid iff d is rigid.*

Example 3.4. *Let $\mathcal{G}$ be the rigid proof grammar with start symbol σ, the non-terminals σ, α and γ, rigid non-terminals $\mathcal{R}$, ordering $\lhd$, all other symbols terminals of appropriate arity, and production rules*

$$\sigma \to f(\alpha, \gamma, \gamma) \qquad \gamma \to g(\gamma) \mid a \qquad \alpha \to \gamma$$

If $\gamma \notin \mathcal{R}$ we have, unsurprisingly, $\mathcal{L}(\mathcal{G}) = \{f(g^m(a), g^n(a), g^o(a)) \mid m, n, o \geq 0\}$. For $\gamma \in \mathcal{R}$,

1. *If $\gamma \lhd \gamma \lhd \alpha$ and $\alpha \in \mathcal{R}$, $\mathcal{L}(\mathcal{G}) = \{f(g^m(a), g^n(a), g^n(a)) \mid m, n \geq 0\}$;*
2. *If $\gamma \lhd \gamma \ntriangleleft \alpha$, $\mathcal{L}(\mathcal{G}) = \{f(g^m(a), g^m(a), g^m(a)) \mid m \geq 0\}$;*
3. *If $\gamma \ntriangleleft \gamma \ntriangleleft \alpha$, $\mathcal{L}(\mathcal{G}) = \{f(a, a, a)\}$.*

Example 3.5. *Let $\mathcal{G}$ be a proof grammar and suppose $\mathcal{R} \subseteq \mathcal{N}_{\mathcal{G}}$. Further, let $d = \langle \rho_i, q_i \rangle_{i<k}\colon M \to N$ be a derivation in $\mathcal{G}$. Two simple choices of $\lhd$ are*

1. Global rigidity. *Set $\lhd_G = \emptyset$. Then d is $(\lhd_G, \mathcal{R})$-rigid iff for all $i, j < k$, $d(i)|q_i = d(j)|q_j \in \mathcal{R}$ implies $N|q_i = N|q_j$.*
2. Local rigidity. *$\lhd_L = \mathcal{N}_{\mathcal{G}} \times \mathcal{N}_{\mathcal{G}}$. In this form rigidity is treated only at the level of production rules: d is $(\lhd_L, \mathcal{R})$-rigid iff for every $i < k$, if $\rho_i = (a \to S)$ and $S|p = S|q \in \mathcal{R}$, then $N|q_i p = N|q_i q$.*

Notation 3.6. *Given a derivation $d = \langle \rho_i, q_i \rangle_{i<k}$ and a position p we define d^p to be the derivation $\langle \rho_i, pq_i \rangle_{i<k}$.*

Definition 3.7 (Subderivation). *Let $d = \langle \rho_i, q_i \rangle_{i<\mathrm{lh}(d)}$ be a derivation and suppose $k < \mathrm{lh}(d)$. The* subderivation of d from k *is the longest derivation $e = \langle \hat{\rho}_i, \hat{q}_i \rangle_{i<\mathrm{lh}(e)}$ such that $\hat{q}_0 = \langle\rangle$ and e^{q_k} is a subsequence of d.*

Definition 3.8 (Strong rigidity). *A derivation $d = \langle \rho_i, q_i \rangle_{i<\mathrm{lh}(d)}$ is* strongly rigid *iff for all $i_0, i_1 < \mathrm{lh}(d)$ if $i_0 \sim_d i_1$ then the subderivations of d from i_0 and i_1 are identical or a permutation of one another.*

Restricting to strongly-rigid derivations does not reduce the language of a rigid grammar:

Lemma 3.9. *Let $\mathcal{G}$ be a rigid grammar. For all M and N, if there is a rigid derivation from M to N in $\mathcal{G}$ then there exists a strongly rigid derivation from M to N in $\mathcal{G}$.*

Proof: By recursion through d: for each p and q such that $d(i)|p = d(j)|q \in \mathcal{N}_\mathcal{G}$ and $N|p = N|q$ replace the sub-derivation of d at q by a copy of the sub-derivation of d at p. The result will be strongly rigid iff d is rigid. □

3.4 Bounds on rigid proof grammars

Lemma 3.10 (Bounding). *Suppose $\mathcal{G}$ is a rigid proof grammar satisfying the following condition:*

$$\begin{gathered}\textit{for every rigid derivation } d = \langle \rho_i, p_i \rangle_{i \le k} \textit{ in } \mathcal{G} \textit{ and every } i < j \le k,\\ \textit{if } p_i \le p_j \textit{ and } d(i)|p_i = d(j)|p_j \textit{ then there exists } l \in (i, j) \textit{ such that}\\ p_i \le p_l \le p_j \textit{ and } d(j)|p_j \lhd_\mathcal{G} d(l)|p_l.\end{gathered} \tag{2}$$

If $\mathcal{G}$ is acyclic then $\mathcal{L}(\mathcal{G})$ is finite; if $\mathcal{G}$ is acyclic and totally rigid, $|Ł(\mathcal{G})| \le |\mathrm{Pr}_\mathcal{G}|^{2^{|\mathcal{N}_\mathcal{G}|-1}}$.

Proof: Assume $\mathcal{G}$ is a rigid proof grammar satisfying (2) and $\lhd_\mathcal{G}$ is acyclic. Fix a rigid derivation $d = \langle \rho_i, p_i \rangle_{i \le k}$ in $\mathcal{G}$ from an starting symbol. Call a derivation $\hat{d} = \langle \hat{\rho}_i, \hat{p}_i \rangle_{i \le \hat{k}}$ a *path through d* if $\hat{d}$ is a subsequence of d, $(\hat{\rho}_0, \hat{p}_0) = (\rho_0, p_0)$, and for every $i < j < \hat{k}$, $\hat{p}_i \le \hat{p}_j$.

Let $\hat{d}$ be a path through d and let $\mathcal{N}_d$ denote the set of non-terminals written by $\hat{d}$, i.e. $\mathcal{N}_d = \{a \in \mathcal{N}_\mathcal{G} \mid \exists i < \hat{k} \exists S\, \hat{\rho}_i = (a \to S)\}$. Suppose $\mathcal{N}_d = \{a_0, a_1, \dots, a_{|\mathcal{N}_d|-1}\}$ and for all $i < j < |\mathcal{N}_d|$, $a_i \ntriangleleft a_j$ which is possible as $\lhd_\mathcal{G}$ is acyclic. By the stated requirement there can be at most one $i \le \hat{k}$ such that $\hat{\rho}_i$ writes a_0, whence there are at most two production rules in $\hat{d}$ that write a_1, four production rules in $\hat{d}$ writing a_2, and in general no more than 2^i production rules writing a_i. So $\hat{k} < 2^{|\mathcal{N}_d|}$. The set of paths through d forms a tree which has branching degree bounded by some constant K independent of d and depth bounded by $2^{|\mathcal{N}_\mathcal{G}|}$, and it follows that $k \le K^{2^{|\mathcal{N}_\mathcal{G}|}}$. As d is arbitrary there are only finitely many rigid derivations and the first result follows.

Suppose $\mathcal{G}$ is acyclic, totally rigid and satisfies (2). Let $m = |\mathrm{Pr}_\mathcal{G}|$ and $n = |\mathcal{N}_\mathcal{G}|$. We argue, by induction on the number of non-terminals in $\mathcal{G}$, that the set of terms

$T \in \mathrm{Terms}(\Sigma_{\mathcal{G}})$ rigidly derivable from a starting symbol in $\mathcal{G}$ has size bounded by $m^{2^{n-1}}$.

The base case is $n = 1$. By the main assumption of the lemma every derivation $d \colon \sigma \to T$ with $\sigma \in \mathcal{S}_{\mathcal{G}}$ and $\in \mathrm{Terms}(\Sigma_{\mathcal{G}})$ has length 1, of which there are no more than m. For the induction step, suppose $n = n_0 + 1$. Let $\mathcal{N} = \mathcal{N}_{\mathcal{G}} \setminus \{a\}$ where $a \neq \sigma$ is chosen such that $a \not< b$ for all $b \in \mathcal{N}_{\mathcal{G}} \setminus \{\sigma\}$. Suppose $d \colon \sigma \to T \in \mathrm{Terms}(\Sigma_{\mathcal{G}})$ is a rigid derivation in $\mathcal{G}$. By the main assumption d can be re-ordered to have the form $d_0 d_1$ where $d_0 \colon \sigma \to S$ and $d_1 \colon S \to T = S[a/S']$ for appropriate terms S and S', such that the non-terminal a is not written in d_0 and not introduced by a production rule in d_1. The induction hypothesis implies there is no more than $m^{2^{n_0-1}}$ possibilities for each of S and S', whence there are $\leq m^{2^{n-1}}$ possibilities for T. □

3.5 Grammars for Π_2-proofs

In this section we associate to each simple Π_2-proof π a proof grammar $\mathcal{G}_\pi$ which will be used in the subsequent sections to prove the confluence result. We begin with a motivating example:

Example 3.11. *Let $\pi \vdash \exists x F$ be the proof given below in which there is a Π_2-cut on $\forall x \exists y A$, a Π_1-cut on $\forall x B$ and we assume these are the only cuts on quantified formulæ in π; α and γ the respectively eigenvariables of these cuts; t_i and u_i witness terms of the existential quantifier in, respectively, $\exists x \forall y \bar{A}$ and $\exists x \bar{B}$; and β_i the eigenvariable of the universal quantifier in $\forall y \bar{A}(t_i, y)$. For a formula C, we use C^* to distinguish between the two copies of C that may appear in the same sequent in the proof.*

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{B(\gamma), A(\alpha, s(\alpha,\gamma)), F(g(\gamma,\alpha))}{B(\gamma), A(\alpha, s(\alpha,\gamma)), \exists x F}\ \exists
}{\forall x B, \exists y A(\alpha, y), \exists x F}\ \exists
\qquad
\dfrac{
\dfrac{
\dfrac{\bar{B}(u_1), \bar{B}^*(u_2(\alpha))}{\bar{B}(u_1), \exists x \bar{B}^*}\ \exists
}{\exists x \bar{B}, \exists x \bar{B}^*}\ \exists
}{\exists x \bar{B}}\ c
}{\exists y A(\alpha, y), \exists x F}\ cut
}{\forall x \exists y A(x, y), \exists x F}\ \forall
\qquad
\pi' \vdash \exists x \forall y \bar{A}(x, y)
}{\pi \vdash \exists x F}\ cut
$$

where π' is given by

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\bar{A}(t_1,\beta_1), \bar{A}^*(t_2(\beta_1),\beta_2)}{\bar{A}(t_1,\beta_1), \forall y\bar{A}^*(t_2(\beta_1),y)}\,\forall}{\bar{A}(t_1,\beta_1), \exists x\forall y\bar{A}^*(x,y)}\,\exists}{\forall y\bar{A}(t_1,y), \exists x\forall y\bar{A}^*(x,y)}\,\forall}{\exists x\forall y\bar{A}(x,y), \exists x\forall y\bar{A}^*(x,y)}\,\exists}{\pi' \vdash \exists x\forall y\bar{A}(x,y)}\,c$$

Having the cut reduction process in mind, it is natural to consider the following grammar. First we introduce a starting symbol $\sigma_{\exists xF}$ which will write to the (literal) witness of $\exists xF$ in π, i.e. the term $g(\gamma,\alpha)$. This production rule initiates the search for the witnesses to $\exists xF$. Next we add production rules that write the eigenvariables α and γ of the two cuts to the associated terms t_i and u_i respectively, mirroring the substitutions performed in Gentzen-style cut elimination. To capture the correct rules for eigenvaribles of the universal quantifier in $\exists x\forall y\bar{A}$, i.e. β_0 and β_1, we introduce a functional non-terminal σ which represents the existential quantifier in $\forall x\exists yA(x,y)$ and allows β_i to write (modulo β-reduction) to the term $s(t_i,\gamma)$.[2] More precisely, $\mathcal{G} = \langle \mathcal{N}, \mathcal{R}, \lhd, \Sigma, \mathcal{S}, \mathrm{Pr}\rangle$ with $\mathcal{S} = \{\sigma_{\exists xF}\}$, $\mathcal{R} = \{\alpha,\gamma,\beta_1,\beta_2\}$, $\mathcal{N} = \mathcal{S}\cup\mathcal{R}\cup\{\sigma\}$ and Pr consisting of rules:

$$\begin{array}{lll} \sigma_{\exists xF} \to g(\gamma,\alpha) & \gamma \to u_1 \mid u_2 & \\ \alpha \to t_1 \mid t_2 & \beta_i \to \sigma\cdot t_i \quad \text{for } i = 1,2 & \sigma \to \lambda\alpha.s \end{array}$$

The priority ordering is given by the order in which the eigenvariables are eliminated in the proof, increasing in priority from top down, and left to right through a Π_2-cut (which has the universal formula on the left subproof):

$$\gamma \lhd \alpha \lhd \beta_2 \lhd \beta_1$$

We now calculate $\mathcal{L}(\mathcal{G})$. There are two possible starting derivations for $\sigma_{\exists xF}$:

$$\sigma_{\exists xF} \to g(\gamma,\alpha) \to g(u_1,\alpha) \tag{3}$$

$$\sigma_{\exists xF} \to g(\gamma,\alpha) \to g(u_2(\alpha),\alpha) \tag{4}$$

Extending (3) *we obtain the closed term $g(u_1,t_1) \in \text{Ł}(\mathcal{G})$ as well as*

$$\sigma_{\exists xF} \to^* g(u_1,\alpha) \to g(u_1,t_2(\beta_1)) \to g(u_1,t_2(\sigma\cdot t_1)) \to g(u_1,t_2(\lambda\alpha.s(\alpha,\gamma)\cdot t_1)).$$

2 Further motivation for the use of non-terminals of function type can be found in [Afshari et al., 2015].

Since $\gamma \lhd \beta_1$, the γ appearing in the last term above is not connected to its earlier occurrence in (3) *and may freely write to either u_1 or u_2, yielding terms $g(u_1, t_2(s(t_1, u_1))$ and $g(u_1, t_2(s(t_1, u_2(t_1)))$ in $\mathcal{L}(\mathcal{G})$.*

To extend (4)*, first note that the (possible) two occurrences of α in $g(u_2(\alpha), \alpha)$ must be written to the same result, of which the following are allowed.*

$$g(u_2(t_1), t_1)$$
$$g(u_2(t_2(s(t_1, u_1))), t_2(s(t_1, u_1)))$$
$$g(u_2(t_2(s(t_1, u_2(t_1)))), t_2(s(t_1, u_2(t_1))))$$

Indeed, it is not hard to check that any cut-free proof π' obtainable from π via the reduction steps in Figures 1 and 2, $\mathcal{H}(\pi', \exists x F)$ consists of exactly the six closed terms derived above.

We now proceed with the definition of the grammar. Let $\pi \vdash \Gamma$ be a simple Π_2-proof of a set of prenex Π_2 and Σ_2 formulæ. The *proof grammar for π* is the grammar $\mathcal{G}_\pi = \langle \mathcal{N}_\pi, \mathcal{R}_\pi, \lhd_\pi, \Sigma_\pi, \mathcal{S}_\pi, \mathrm{Pr}_\pi \rangle$ where the components are defined as follows.

Symbols and their types

We will use symbols of the form τ_F^p or σ_F^p where p is a position and F is a formula occurring in π.[3] Each such symbol is assigned a *type*, either 0 (ground type) or $0 \to 0$ (function type):

- $\mathrm{type}(\tau_F^p) = \mathrm{type}(\sigma_{\bar{F}}^p) = 0$ if $F \in \Pi_2$.
- $\mathrm{type}(\tau_F^p) = \mathrm{type}(\sigma_{\bar{F}}^p) = (0 \to 0)$ otherwise (i.e. $F \in \Sigma_2 \setminus \Pi_2$).
- All eigenvariables in π and first-order terms are ground-type terms.

Alphabet and non-terminals

Σ_π consists of the function symbols, constants and variables occurring in π, and symbols $\tau_F^{\langle\rangle}$ for every $F \in \Gamma$. The set of non-terminals, $\mathcal{N}_\pi$, consists of

- rigid non-terminals $\mathcal{R}_\pi = \mathrm{EV}(\pi)$;
- starting symbols $\mathcal{S}_\pi = \{\sigma_F^{\langle\rangle} \mid F \in \Gamma\}$;
- symbols τ_A^p where $p \in \mathrm{Pos}(\pi) \setminus \{\langle\rangle\}$ and A is a formula occurring in the end-sequent of $\pi|p$;

3 For brevity we write τ_F for $\tau_F^{\langle\rangle}$.

– symbols σ_A^p where $p \in \mathrm{Pos}(\pi)$ and A is a formula occurring in the end-sequent of $\pi|p$.

Table 2. Production rules in $\mathcal{G}_\pi$

$\pi\|p \vdash \Gamma$	Production rules in $\mathcal{G}_\pi$ for position p, $\mathrm{Pr}_{\pi,p}$
$\pi\|p \vdash A, \bar{A}$	$\{\sigma_A^p \to \tau_{\bar{A}}^p, \sigma_{\bar{A}}^p \to \tau_A^p\}$
$\dfrac{\Delta, A_0, A_1}{\pi\|p \vdash \Delta, A_0 \vee A_1} \vee$	$\mathrm{prod}(p, \Delta, \emptyset)$
$\dfrac{\Pi, A_0 \quad \Delta, A_1}{\pi\|p \vdash \Pi, \Delta, A_0 \wedge A_1} \wedge$	$\mathrm{prod}(p, \Pi, \Delta)$
$\dfrac{\Delta, A[x/\alpha]}{\pi\|p \vdash \Delta, \forall xA} \forall$	$\mathrm{prod}(p, \Delta, \emptyset) \cup \{\alpha \to \tau_{\forall xA}^p, \sigma_{\forall xA}^p \to \lambda\alpha.\sigma_{A[x/\alpha]}^{p0}\}$
$\dfrac{\Delta, A[y/s]}{\pi\|p \vdash \Delta, \exists yA} \exists$	$\mathrm{prod}(p, \Delta, \emptyset) \cup \{\sigma_{\exists yA}^p \to s,\ \tau_{A[y/s]}^{p0} \to \tau_{\exists yA}^p \cdot s\}$
$\dfrac{\Delta, A \quad \Pi, \bar{A}}{\pi\|p \vdash \Delta, \Pi} \mathrm{cut}$	$\mathrm{prod}(p, \Delta, \Pi) \cup \{\tau_A^{p0} \to \sigma_{\bar{A}}^{p1}, \tau_{\bar{A}}^{p1} \to \sigma_A^{p0}\}$
$\dfrac{\Delta}{\pi\|p \vdash \Delta, \Pi} \mathrm{w}$	$\mathrm{prod}(p, \Delta, \emptyset)$
$\dfrac{\Delta, \Pi, \Pi^*}{\pi\|p \vdash \Delta, \Pi} \mathrm{c}$	$\mathrm{prod}(p, \Pi \cup \Delta, \emptyset) \cup \{\tau_{F^*}^{p0} \to \tau_F^p, \sigma_F^p \to \sigma_{F^*}^{p0} \mid F \in \Pi\}$
	$\mathrm{prod}(p, \Gamma, \Delta) = \mathrm{prod}_\tau(p, \Gamma, \Delta) \cup \mathrm{prod}_\sigma(p, \Gamma, \Delta)$
	$\mathrm{prod}_\tau(p, \Gamma_0, \Gamma_1) = \{\tau_F^{pj} \to \tau_F^p \mid j \in \{0, 1\} \wedge F \in \Gamma_j\}$
	$\mathrm{prod}_\sigma(p, \Gamma_0, \Gamma_1) = \{\sigma_F^p \to \sigma_F^{pj} \mid j \in \{0, 1\} \wedge F \in \Gamma_j\}$

Production rules and priority ordering

As the set of production rules in $\mathcal{G}_\pi$ we choose $\mathrm{Pr}_\pi := \bigcup_{p \in \mathrm{Pos}(\pi)} \mathrm{Pr}_{\pi,p}$, where the sets $\mathrm{Pr}_{\pi,p}$ (defined in Table 2) are determined by the rule of inference occurring at position p in π. Informally, non-terminals of the form σ_F^p represent the existential quantifier in F (if there is one) with their production rules 'searching' for witnesses within the sub-proof $\pi|p$. Dually, τ^p_F represents the universal quantifier in F (at p) and link eigenvariables on one side of a cut to the existential witnesses on the other side.

Like the production rules, the rigidity ordering, $\lhd_\pi$, is determined by the local structure of π and is the smallest transitive relation satisfying the following five conditions.

– Concerning production rules for axioms and side formulæ to all inference rules:
  1. For each production rule of the form $a \to \tau^q_B$ or $a \to \tau^q_B \cdot s$ where $a \in \mathcal{N}_\pi \setminus \mathrm{EV}(\pi)$ we have $a \lhd_\pi \tau^q_B$;
  2. For each rule of the form $\sigma^p_A \to \sigma^{pj}_B$ or $\sigma^p_A \to \lambda\alpha.\sigma^{pj}_B$ ($j \in \{0, 1\}$) we have $\sigma^p_A \lhd_\pi \sigma^{pj}_B$;
– For production rules introducing or eliminating eigenvariables:
  3. For each production rule $\alpha \to \tau^q_{\forall xA}$ where $\alpha \in \mathrm{EV}(\pi)$ we have $\alpha \lhd_\pi \tau^q_{\forall xA}$, and $\xi \lhd_\pi \alpha$ for every $\xi \in \mathrm{EV}(\pi|q0)$;
  4. For each rule of the form $\sigma^p_A \to s$ and each variable α appearing in s we set $\sigma^p_A \lhd_\pi \alpha$;
– And for production rules of cut formulæ:
  5. For a rule $\tau^{pi}_A \to \sigma^{pj}_{\bar{A}}$ with $i = 1 - j \in \{0, 1\}$, if $A = \forall x A_0$ for some Σ_1 formula A_0 then $\tau^{pi}_A \lhd_\pi \sigma^{pj}_{\bar{A}}$.

The first four conditions increase the priority of non-terminals as one follows either 'τ' production rules 'down' the proof towards the conclusion or principal cuts, or 'σ' productions 'upwards' towards witnesses of existential quantifiers. The final condition mediates the passage between the two paths over (one direction of) a Π_2-cut. The additional cases added by 3. capture, through the rigidity ordering, the duplication of eigenvariables that may occur when reducing a quantified cut.

Example 3.12. *Consider the proof π in Example 3.11. The grammar $\mathcal{G}_\pi$ is the tuple $\langle \mathcal{N}_\pi, \mathcal{R}_\pi, \lhd_\pi, \Sigma_\pi, \mathcal{S}_\pi, \mathrm{Pr}_\pi \rangle$ where $\mathcal{S}_\pi = \{\sigma^{\langle\rangle}_{\exists xF}\}$, $\mathcal{R}_\pi = \{\alpha, \gamma, \beta_1, \beta_2\}$, we have in particular $\gamma \lhd_\pi \alpha \lhd_\pi \tau^0_{\forall x \exists y A} \lhd_\pi \sigma^1_{\exists x \forall y \bar{A}} \lhd_\pi \beta_2 \lhd_\pi \beta_1$ and the production rules are:*

1. *For the starting symbol we have:*

$$\sigma^{\langle\rangle}_{\exists xF} \to \sigma^{0}_{\exists xF} \to \sigma^{00}_{\exists xF} \to \sigma^{000}_{\exists xF} \to \sigma^{0000}_{\exists xF} \to g(\gamma, \alpha)$$

2. *For the eigenvariable γ:*

$$\gamma \to \tau^{000}_{\forall xB} \to \sigma^{001}_{\exists x\bar{B}} \to \sigma^{0010}_{\exists x\bar{B}} \mid \sigma^{0010}_{\exists x\bar{B}^*}$$
$$\sigma^{0010}_{\exists x\bar{B}} \to u_1$$
$$\sigma^{0010}_{\exists x\bar{B}^*} \to \sigma^{00100}_{\exists x\bar{B}^*} \to u_2$$

3. *For eigenvariable α:*

$$\alpha \to \tau^{0}_{\forall x \exists y A} \to \sigma^{1}_{\exists x \forall y \bar{A}} \to \sigma^{10}_{\exists x \forall y \bar{A}} \mid \sigma^{10}_{\exists x \forall y \bar{A}^*}$$
$$\sigma^{10}_{\exists x \forall y \bar{A}} \to t_1$$
$$\sigma^{10}_{\exists x \forall y \bar{A}^*} \to \sigma^{100}_{\exists x \forall y \bar{A}^*} \to \sigma^{1000}_{\exists x \forall y \bar{A}^*} \to t_2$$

4. *For eigenvariables β_1 and β_2:*

$$\beta_1 \to \tau^{100}_{\forall y \bar{A}(t_1)} \to \tau^{10}_{\exists x \forall y \bar{A}} \cdot t_1$$
$$\tau^{10}_{\exists x \forall y \bar{A}} \to \tau^{1}_{\exists x \forall y \bar{A}}$$
$$\beta_2 \to \tau^{10000}_{\forall y \bar{A}^*(t_2)} \to \tau^{1000}_{\exists x \forall y \bar{A}^*} \cdot t_2$$
$$\tau^{1000}_{\exists x \forall y \bar{A}^*} \to \tau^{100}_{\exists x \forall y \bar{A}^*} \to \tau^{10}_{\exists x \forall y \bar{A}^*} \to \tau^{1}_{\exists x \forall y \bar{A}}$$
$$\tau^{1}_{\exists x \forall y \bar{A}} \to \sigma^{0}_{\forall x \exists y A} \to \lambda\alpha.\sigma^{00}_{\exists y A}$$
$$\sigma^{00}_{\exists y A} \to \sigma^{000}_{\exists y A} \to s$$

Notice that the function type non-terminal $\sigma^{0}_{\forall x \exists y A}$ plays the same role as σ in grammar $\mathcal{G}$ of Example 3.11. Indeed, $\mathcal{L}(\mathcal{G}) = \text{Ł}(\mathcal{G}_\pi)$.

Shifting symbols, terms and rules

Following the definition above, it is clear that if π is a sub-proof of π', say $\pi = \pi'|p$, then $\mathcal{G}_\pi$ can be viewed as a sub-grammar of $\mathcal{G}_{\pi'}$ by 'shifting' the annotation of non-terminals in $\mathcal{G}_\pi$ by the position p. This action of 'shifting' a grammar relative to a position turns out to be a useful operation on grammars. We let $\mathcal{G}^p$ denote the result of shifting $\mathcal{G}$ relative to p, that is, $\mathcal{G}^p = \langle \mathcal{N}^p_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}}, \lhd^p_\pi, \Sigma_{\mathcal{G}}, \mathcal{S}^p_{\mathcal{G}}, \mathrm{Pr}^p_{\mathcal{G}} \rangle$, where

$$\mathcal{N}^p_{\mathcal{G}} = \{a^p \mid a \in \mathcal{N}_{\mathcal{G}}\} \qquad \mathcal{S}^p_{\mathcal{G}} = \{a^p \mid a \in \mathcal{S}_{\mathcal{G}}\}$$
$$\mathrm{Pr}^p_{\mathcal{G}} = \{a^p \to T^p \mid (a \to T) \in \mathrm{Pr}_{\mathcal{G}}\} \qquad \lhd^p_\pi = \{(a^p, b^p) \mid a \lhd_\pi b\}$$

given by

$$(\tau^q_F)^p = \tau^{pq}_F \qquad (\sigma^q_F)^p = \sigma^{pq}_F \qquad \gamma^p = \gamma \quad \text{for } \gamma \in \mathcal{R}_{\mathcal{G}}$$
$$(T_1 \cdot T_2)^p = T^p_1 \cdot T^p_2 \qquad (\lambda\alpha.T)^p = \lambda\alpha.(T^p)$$

So in particular $\mathcal{G}^p_{\pi|p}$ is a sub-grammar of $\mathcal{G}_\pi$ whenever $p \in \mathrm{Pos}(\pi)$.

Lemma 3.13. *If π is a simple Π_2-proof then $\lhd_\pi$ is acyclic.*

Proof: By induction on the proof $\pi \vdash \Gamma$ noticing that no production rule in $\mathcal{G}_\pi$ writes $\tau^{\langle\rangle}_F$ or introduces $\sigma^{\langle\rangle}_F$ (for $F \in \Gamma$), and that if π ends in a cut on a simple

Π_2 formula $A = \forall x \exists y B$ (say $\pi|0 \vdash \Gamma, A$ and $\pi|1 \vdash \Gamma, \bar{A}$) then for all $a \in \mathcal{N}^0_{\pi|0}$ and $b \in \mathcal{N}^1_{\pi|1}$ we have $a \ntriangleleft_\pi b$. □

Theorem 3.14. *Let π be a simple Π_2-proof. Then $|Ł(\mathcal{G}_\pi)| < 2^{2^{2|\pi|}}$.*

Proof: First, note that requirement (2) of Lemma 3.10 is satisfied by all proof grammars arising from simple Π_2-proofs. We define a new grammar $\mathcal{G}'$ by making the following changes to $\mathcal{G}_\pi$: for every non-terminal $v^p_F \notin \mathcal{R}_\pi$ with $v \in \{\tau, \sigma\}$ and production rule $v^p_F \to S$, if there is a production rule $a \to v^p_F$ in $\mathcal{G}_\pi$ then remove $v^p_F \to S$ and instead add $a \to S$. Observe this process is well-defined as the term S cannot contain any occurrences of v^p_F. Once all production rules writing v^p_F are replaced, remove v^p_F from the set of non-terminals. Let $\mathcal{G}_0$ be the resulting grammar, and $\mathcal{G}'$ the subsequent grammar obtained by indicating any remaining non-rigid non-terminals in $\mathcal{G}_0$ to be rigid.

$\mathcal{G}'$ is a totally rigid grammar and $|\mathcal{N}_{\mathcal{G}'}| + |\mathrm{Pr}_{\mathcal{G}'}| \leq 2|\pi|$. Moreover, $\mathcal{L}(\mathcal{G}_\pi) = Ł(\mathcal{G}')$: it is easy to see that the 'compression' preserves rigid derivations and so $\mathcal{L}(\mathcal{G}_0) = Ł(\mathcal{G}_\pi)$. In $\mathcal{G}_0$ the only non-rigid non-terminals remaining are of the form $\tau^p_{\exists y A}$ or $\sigma^p_{\forall y \bar{A}}$ for some p and A. Observe that i) there is only one choice for writing such non-terminals, and ii) $\tau^p_{\exists y A}, \sigma^p_{\forall y \bar{A}} \triangleleft_\pi \alpha$ where α is the eigenvariable of $\forall y \bar{A}$. Thus any two non-terminals that become disconnected through $\tau^p_{\exists y A}$ or $\sigma^p_{\forall y \bar{A}}$ in $\mathcal{G}'$ were already disconnected in $\mathcal{G}_\pi$ through α. Indeed $\mathcal{G}'$ and $\mathcal{G}_\pi$ have the same language, and the comparability between their rigid derivations means $\mathcal{G}'$ also satisfies requirement (2).

Now Lemma 3.10 implies $|Ł(\mathcal{G}')| \leq |\mathrm{Pr}_{\mathcal{G}'}|^{2^{|\mathcal{N}_{\mathcal{G}'}|-1}}$. We therefore have

$$|Ł(\mathcal{G}_\pi)| = |Ł(\mathcal{G}')| \leq 2^{2^{|\mathcal{N}_{\mathcal{G}'}|+|\mathrm{Pr}_{\mathcal{G}'}|-1}} < 2^{2^{2|\pi|}}.$$

This concludes the proof. □

4 Technical lemmas

In Section 5 we will prove that for regular simple Π_2-proofs π and π' if $\pi \rightsquigarrow \pi'$ then $\mathcal{L}(\mathcal{G}_{\pi'}) \subseteq Ł(\mathcal{G}_\pi)$. Before considering the theorem, however, we require additional results concerning the fine structure of proof grammars. In Section 4.1 we introduce the notion of homomorphism between proof grammars as a means to test language containment. Section 4.2 highlights a number of properties relating to derivations in proof grammars and Section 4.3 prepares the ground-work for replacing sub-proofs by ones with comparable grammars.

4.1 Comparing grammars

An easy way to compare the languages induced by grammars is by providing a function mapping non-terminals of one grammar into another preserving rigid derivability. In the simplest form (which is all that is required here) this is given by a *homomorphism* as described below. First we need the following definitions.

Controlled β-reduction

We generalise the relation of β-reduction to obtain a finer relation between terms. Given two λ-terms S and T and a (possibly empty) set of variables $X \subseteq \mathcal{V}$, we write $S \rightarrowtail_X T$ if one of the following conditions hold.

- $S = T$;
- $S = S_0 \cdot S_1$, $T = T_0 \cdot T_1$ and $S_i \rightarrowtail_X T_i$;
- $S = \lambda\alpha.S_0$, $T = \lambda\alpha.T_0$ and $S_0 \rightarrowtail_X T_0$;
- $S = (\lambda\alpha.S_0) \cdot S_1$, $\alpha \in X$ and $T = S_0[\alpha/S_1]$.

Notice that $S \rightarrowtail_{\mathcal{V}} T$ is the usual 1-step parallel β-reduction. We write $S \rightarrowtail T$ iff $S \rightarrowtail_{\mathcal{V}} T$. This is not to be confused with $S \rightarrowtail_{\emptyset} T$ which holds iff $S = T$. Also let $S \rightarrowtail_X^* T$ denote the (reflexive) transitive closure of $S \rightarrowtail_X T$.

Following the definition of $S \rightarrowtail T$ we define a canonical function $r_{T,S}$ mapping positions in T to their corresponding position in S:

$$\begin{aligned} r_{T,S}(\langle\rangle) &= \langle\rangle \\ r_{(T_0\cdot T_1),(S_0\cdot S_1)}(ip) &= i\, r_{T_i,S_i}(p), \quad \text{for } i \in \{0,1\} \\ r_{(\lambda\alpha.T_0),(\lambda\alpha.S_0)}(0p) &= 0\, r_{T_0,S_0}(p) \\ r_{S_0[\alpha/S_1],(\lambda\alpha.S_0)\cdot S_1}(p) &= \begin{cases} 1p', & \text{if } p = qp',\ q \in \mathrm{Free}(\alpha, S_0) \text{ and } p' \in \mathrm{Pos}(S_1) \\ 00p, & \text{otherwise} \end{cases} \end{aligned}$$

where $\mathrm{Free}(\alpha, S_0)$ denotes the positions in S_0 marking free occurrences of α.

Homomorphism function

Let $\mathcal{F}$ and $\mathcal{G}$ be rigid proof grammars and $f\colon \mathcal{N}_{\mathcal{F}} \to \mathcal{N}_{\mathcal{G}}$. If $\Sigma_{\mathcal{F}} \subseteq \Sigma_{\mathcal{G}}$ then f naturally extends to a function $(.)^f\colon \mathrm{Terms}(\Sigma_{\mathcal{F}} \cup \mathcal{N}_{\mathcal{F}}) \to \mathrm{Terms}(\Sigma_{\mathcal{G}} \cup \mathcal{N}_{\mathcal{G}})$ given by

$$\begin{aligned} \alpha^f &= f(\alpha) \quad \text{if } \alpha \in \mathcal{N}_{\mathcal{F}}, & (\lambda\alpha.S)^f &= \lambda\alpha^f.S^f \\ a^f &= a \quad \text{if } a \in \Sigma_{\mathcal{F}}, & (S\cdot T)^f &= S^f \cdot T^f \end{aligned}$$

Definition 4.1 (Homomorphism). *Let $\mathcal{F}, \mathcal{G}$ be rigid grammars and suppose $\Sigma_\mathcal{F} \subseteq \Sigma_\mathcal{G}$ and $\mathcal{S}_\mathcal{F} \subseteq \mathcal{S}_\mathcal{G}$. A* homomorphism *from $\mathcal{F}$ into $\mathcal{G}$ is a function $f\colon \mathcal{N}_\mathcal{F} \to \mathcal{N}_\mathcal{G}$ such that*

1. *$f(\sigma) = \sigma$ for every $\sigma \in \mathcal{S}_\mathcal{F}$;*
2. *$f(a)$ has the same type as a for every $a \in \mathcal{N}_\mathcal{F}$;*
3. *$f(\alpha) \in \mathcal{R}_\mathcal{G}$ iff $\alpha \in \mathcal{R}_\mathcal{F}$;*
4. *$a \lhd_\mathcal{F} b$ implies $f(a) \lhd_\mathcal{G} f(b)$;*
5. *for each $\rho = (a \to S) \in \mathcal{F}$, there is a derivation $\rho^f = \langle \rho_i, p_i\rangle_{i<k}\colon f(a) \to S^f$ in $\mathcal{G}$ such that $\rho^f(i)|p_i \in \mathcal{R}_\mathcal{G}$ only if $i = 0$;*
6. *For every rigid derivation $d = \langle (a_i \to S_i), p_i\rangle_{i<l}\colon \sigma \to S$ in $\mathcal{F}$ with $\sigma \in \mathcal{S}_\mathcal{F}$ and every $j_0, j_1 < l$, if $f(a_{j_0}) = f(a_{j_1})$ and $j_0 \approx_d j_1$ then $a_{j_0} = a_{j_1}$.*

We write $f\colon \mathcal{F} \to \mathcal{G}$ to stipulate that $\Sigma_\mathcal{F} \subseteq \Sigma_\mathcal{G}$, $\mathcal{S}_\mathcal{F} \subseteq \mathcal{S}_\mathcal{G}$ and f is a homomorphism from $\mathcal{F}$ to $\mathcal{G}$.

Notice that condition 6 is trivially satisfied if f is injective, thus its only role is for in the case of Contraction Reduction.

A homomorphism $f\colon \mathcal{F} \to \mathcal{G}$ extends to a function $(.)^f\colon \mathrm{Der}(\mathcal{F}) \to \mathrm{Der}(\mathcal{G})$ in the obvious way: given a derivation $d = \langle \rho, p\rangle e\colon S \to T$ in $\mathcal{F}$, suppose $e^f\colon d(1)^f \to T^f$ is already defined and set $d^f := \langle \rho^f, p\rangle e^f\colon S^f \to T^f$ in $\mathcal{G}$.

Lemma 4.2. *Suppose $f\colon \mathcal{F} \to \mathcal{G}$ is a homomorphism. If $d\colon S \to T$ is a rigid derivation in $\mathcal{F}$ then $d^f\colon S^f \to T^f$ is a rigid derivation in $\mathcal{G}$.*

Proof: Let $\mathcal{F}$ and $\mathcal{G}$ be as above, $d = \langle (a_i \to S_i), p_i\rangle_{i<k}\colon S \to T$ a rigid derivation in $\mathcal{F}$ and $d^f = \langle (b_j \to M_j), q_j\rangle_{j<\mathrm{lh}(d^f)}\colon S^f \to T^f$ the derivation in $\mathcal{G}$ induced by f.

First suppose $j_0, j_1 < \mathrm{lh}(d^f)$ are such that $j_0 \approx_{d^f} j_1$ and $b_{j_0} = b_{j_1} \in \mathcal{R}_\mathcal{G}$. Condition 4 (combined with the definition of $d \mapsto d^f$) implies that there are $i_0, i_1 < \mathrm{lh}(d)$ such that $b_{j_l} = f(a_{i_l})$ and $q_{j_l} = p_{i_l}$ for $l = 0, 1$. Let $j^* < j_0, j_1$ be the index witnessing $j_0 \approx_{d^f} j_1$. Set $j \leq j^*$ be the greatest index such that $b_j = f(a_{i^*})$ for some $i^* < \mathrm{lh}(d)$. We now verify that i^* witnesses $i_0 \approx_d i_1$: To obtain a contradiction, suppose for $l = 0$ or $l = 1$ there is a non-terminal a_i with $i^* < i < i_l$ such that $p_i \leq p_l$ and $a_{i_l} \lhd_\mathcal{F} a_i$. By condition 4 we have $b_{j_l} = f(a_{i_l}) \lhd_\mathcal{G} f(a_i)$. Let $f(a_i) = b_{j'}$. By the definition of j we know for every $j^* \geq j' > j$, $b_{j'} \notin f(\{a_i \mid i \leq \mathrm{lh}(d)\})$. Therefore, $j' > j^*$, and since $b_{j_l} \lhd_\mathcal{G} b_{j'}$ we see that $b_{j'}$ violates $j_0 \approx_{d^f} j_1$, hence we are done. Since $i_0 \approx_d i_1$, from condition 6 we deduce $a_{i_0} = a_{i_1}$ so, since d is rigid, $T|p_{i_0} = T|p_{i_1}$. But then $T^f|q_{j_0} = T^f|q_{j_1}$ as required.

To check the second condition of rigidity for d^f, suppose $d^f(j)|x = \lambda\alpha.S_0^f$ for $x < q_j$ and $b_j = \alpha \in \mathcal{R}_\mathcal{G}$. Note that for any term T and position y, $T^f|y = (T|y)^f$. In particular, since α is also rigid there is $i < \mathrm{lh}(d)$ such that $d^f(j) = (d(i))^f$, $f(a_i) = b_j$ with $a_i \in \mathcal{R}_\mathcal{F}$, and $p_i = q_j$. Then we have $d(i)|x = \lambda a_i.S_0$ and so by rigidity of d there exists $x < y < p_i$ such that $d(i)|y = \lambda b.S_1$ and $a_i \lhd_\mathcal{F} b$. Let $\beta = f(b)$. Then $d^f(j)|y = \lambda\beta.S_1^f$ and by condition 4 $\alpha \lhd_\mathcal{G} \beta$. □

Since homomorphisms fix terms from $\Sigma_{\mathcal{F}}$ we conclude

Corollary 4.3. *Suppose $\mathcal{F}$ and $\mathcal{G}$ are rigid proof grammars and there is a homomorphism $f\colon \mathcal{F} \to \mathcal{G}$. Then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{G})$.*

4.2 Partitioning

Let π be a proof ending in a cut. The grammar $\mathcal{G}_\pi$ can be viewed as the union of the two disjoint subgrammars $\mathcal{G}^0_{\pi|0}$ and $\mathcal{G}^1_{\pi|1}$ and 'connecting' production rules (i.e. rules introduced by the cut). A full derivation d in $\mathcal{G}_\pi$ can therefore be permuted to the form $d_1 d_2 \cdots d_k$ where each d_i is a full derivation in either $\mathcal{G}^0_{\pi|0}$ or $\mathcal{G}^1_{\pi|1}$, plus one direction of the connecting rules. The next lemma demonstrates that in the case of cuts on (simple) Π_2-formulæ, there can be at most three such 'alternations' between the two subgrammars. This observation will be particularly relevant in the analysis of the Quantifier Reduction rule (Lemma 5.6).

Lemma 4.4 (Cut Partition). *Suppose π is the simple Π_2-proof given below where A is assumed to be a Π_2 formula.*

$$\frac{\pi_0 \vdash \Gamma, A \qquad \pi_1 \vdash \Delta, \bar{A}}{\pi \vdash \Gamma, \Delta}\,\mathit{cut}$$

For $F \in \Gamma$ and $N \in \mathrm{Term}(\Sigma_\pi)$, if $d\colon \sigma^0_F \to N$ is a rigid derivation in $\mathcal{G}_\pi$ then there is a permutation $d' = d_1 d_2 d_3 d_4\colon \sigma^0_F \to N$ of d such that
- *$d_1\colon \sigma^0_F \to M_1$ is a full derivation in $\mathcal{G}^0_{\pi_0} \cup \{\tau^0_A \to \sigma^1_{\bar{A}}\}$;*
- *$d_2\colon M_1 \to M_2$ is a rigid derivation $\mathcal{G}^1_{\pi_1} \cup \{\tau^1_{\bar{A}} \to \sigma^0_A\}$;*
- *$d_3\colon M_2 \to M_3$ is a derivation in $\mathcal{G}^0_{\pi_0}$;*
- *$d_4\colon M_3 \to N$ is a derivation using only the rules $\{\tau^0_F \to \tau^{\langle\rangle}_F \mid F \in \Gamma\} \cup \{\tau^1_F \to \tau^{\langle\rangle}_F \mid F \in \Delta\}$.*

Proof: The derivation d' is obtained through permuting the production rules so that we fully re-write the non-terminals of a subgrammar $\mathcal{G}^i_{\pi_i}$ before passing (through the cut on A) to those of $\mathcal{G}^{1-i}_{\pi_{1-i}}$. In this way it is clear how to obtain full derivations d_1 and d_2 satisfying the requirements such that $d_1 d_2$ is a subsequence of d. Note, as d_1 and d_2 are full, M_1 and M_2 are, respectively, terms in $\Sigma_\pi \cup \{\sigma^1_{\bar{A}}\} \cup \{\tau^0_F \mid F \in \Gamma\}$ and $\Sigma_\pi \cup \{\sigma^0_A\} \cup \{\tau^0_F \mid F \in \Gamma\} \cup \{\tau^1_F \mid F \in \Delta\}$. Finally, to obtain d_3 we rewrite (according to d) all occurrences of σ^0_A in M_2. Observe that in doing so the non-terminal τ^0_A will not be created: d_3 writes each occurrence of σ^0_A in M_2 to a term of the form $\lambda\alpha.S$ where α is the *unique* eigenvariable for the external universal quantifier in A; the definition of $\lhd_\pi$ and the restriction on rigid derivations implies d_3 may not utilise a production rule of the form $\alpha \to \tau^q_A$, so τ_A^0 cannot appear in M_3. Therefore, the only non-terminals from $\mathcal{G}_\pi$ in M_3 are $\{\tau_F^0 \mid F \in \Gamma\} \cup \{\tau_F^1 \mid F \in \Delta\}$ which will be written with the remaining of production rules from $\mathrm{prod}_\tau(\langle\rangle, \Gamma, \Delta)$. □

Theorem 4.5 (Partition). *Let π be a simple Π_2-proof and $p \in \mathrm{Pos}(\pi)$. Denote by $\mathcal{G}_p$ the sub-grammar of $\mathcal{G}$ comprising only those production rules writing non-terminals in $\{\sigma_F^q, \tau_F^q \mid p \leq q\} \cup \mathrm{EV}(\pi|p)$ with start symbols $\{\sigma_F^p \mid F$ in the end sequent of $\pi|p\}$. If $N \in \text{Ł}(\mathcal{G}_\pi)$, then there exists a rigid derivation $d\colon \sigma \to N$ in $\mathcal{G}_\pi$ of the form $d = d_0 d_1 \cdots d_L$ where for all $i \leq L$,*

- *$d_i\colon M_i \to M_{i+1}$ is a full derivation in $\mathcal{G}_\pi \setminus \mathcal{G}_p$ if i is even;*
- *$d_i\colon M_i \to M_{i+1}$ is a full derivation in $\mathcal{G}_p$ if i is odd.*

Proof: The proof is by induction on p. The base case $p = \langle\rangle$ holds trivially, choosing $d_0 = d$. Suppose $p = qj$ where $j \in \{0, 1\}$. Let L' and $\langle d_i' \rangle_{i \leq L'}$ be given by the induction hypothesis for $q \in \mathrm{Pos}(\pi)$. For each $i \leq L'$ therefore $d_i'\colon M_i \to M_{i+1}$ is a full derivation in $\mathcal{G}_\pi \setminus \mathcal{G}_q$, if i is even, and in $\mathcal{G}_q$ otherwise.

First suppose the final rule of $\pi|q$ is a cut. By Lemma 4.4 for each odd i, d_i' can be replaced by a derivation of the form $e_{i,0}e_{i,1}e_{i,2}\colon M_i \to M_{i+1}$ where $e_{i,0}$ and $e_{i,2}$ are full derivations in $\mathcal{G}_{q0} = \mathcal{G}_p$ and $e_{i,1}$ is a full derivation in $\mathcal{G}_{q1}$.

Suppose $j = 1$. Then we pick $L = L'$ and set

$$d_i = \begin{cases} e_{i-1,2} d_i' e_{i+1,0} & i \leq L \text{ even,} \\ e_{i,1}, & i \leq L \text{ odd,} \end{cases}$$

where, if necessary, $e_{L+1,0} = e_{-1,2} = \langle\rangle$. If instead $j = 0$ we set $L = 2L'$ and for each $2i \leq L'$ we define

$$d_{4i} = d_{2i}' \qquad d_{4i+2} = e_{2i+1,1}$$
$$d_{4i+1} = e_{2i+1,0} \qquad d_{4i+3} = e_{2i+1,2}.$$

Now suppose the rule at position q in π is not a cut. Notice that if $e\colon a \to S$ is a derivation in $\mathcal{G}_q$ with $a \in \mathcal{N}_{\mathcal{G}_q}$ then e is either a derivation wholly in $\mathcal{G}_{q0}$ or a derivation wholly in $\mathcal{G}_{q1}$. Thus for each odd $i \leq L'$, d_i' can be re-ordered as a derivation $e_{i,0}e_{i,1}\colon M_i \to M_{i+1}$ where $e_{i,0}$ is full in $\mathcal{G}_{q0}$ and $e_{i,1}$ is full in $\mathcal{G}_{q1}$. Moreover, observe that the two derivations are independent of one another, so $e_{i,1}e_{i,0}\colon M_i \to M_{i+1}$. Using this fact it is straightforward to alter the arrangement of derivations to obtain a satisfying sequence $\langle d_i \rangle_{i \leq L}$ with $L \leq L'$. □

4.3 Lifting

Another ingredient required in the analysis of the Quantifier Reduction rule is commuting derivations with β-reduction. More precisely, given a derivation

$d\colon T \to U$ and β-reduction $R \rightarrowtail_X T$, whether it is possible to 'lift' d to a derivation $e\colon R \to S$ such that $S \rightarrowtail_X U$ as in the figure below. This is not in general possible as demonstrated by the example in Figure 3. Lemma 4.8 pins down sufficient conditions for completing the square. Before this we define the reverse process which can be visualised as 'pushing' a derivation through β-reduction.

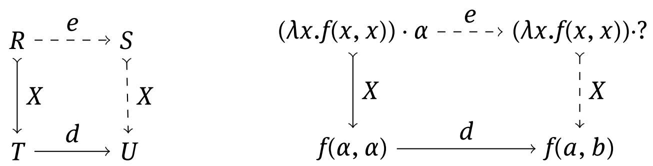


Fig. 3. Lifting

Definition 4.6. *Let $e = \langle \rho_i, p_i \rangle_{i<\mathrm{lh}(e)}\colon R \to S$ be a derivation in grammar $\mathcal{G}$ and suppose $R \rightarrowtail_X T$. The derivation e^T is defined by recursion on $\mathrm{lh}(e)$.*

- *For $\mathrm{lh}(e) = 0$, $e^T := \langle\rangle$;*
- *For $\mathrm{lh}(e) = n + 1$, $e^T := e_0^T \langle \rho_n, q_i \rangle_{i \le k}$ where $e_0 = \langle \rho_i, p_i \rangle_{i<n}$ and $\{q_i \mid i \le k\}$ is an enumeration of $\{q \mid r_{T,R}(q) = p_n\}$.*

It is easily seen that

Lemma 4.7. *If $e\colon R \to S$, $R \rightarrowtail_X T$ and $e^T\colon T \to U$ then $S \rightarrowtail_X U$, and if d is strongly rigid, so is e^T.*

Lemma 4.8 (Lifting). *Let $d = \langle a_i \to S_i, p_i \rangle_{i<k}\colon T \to U$ be a strongly rigid derivation in grammar $\mathcal{G}$ and $R \rightarrowtail_X T$. Suppose*

1. *for all positions q_0, q_1, if $r_{T,R}(q_0) = r_{T,R}(q_1)$ then the subderivations of d from q_0 and q_1 are permutations of one another;*
2. *if $a_i \in X$ then there exists $p < p_i$ and non-terminal $\alpha \notin X$ such that $a_i \lhd \alpha$ and $U|p = \lambda\alpha.(U|p0)$.*

Then there exists a strongly rigid derivation $e\colon R \to S$ such that e^T is a permutation of d and $S \rightarrowtail_X U$.

Proof: Let $r_{T,R}$ be the canonical function mapping positions in T to their corresponding position in R. To each $p \in \mathrm{Pos}(S)$ we can associate a particular position in T corresponding to p. Thus let $r_{T,S}^{-1}$ be a function such that

$$r_{T,S}^{-1}(p) \in \{q \mid r_{T,S}(q) = p\}$$

for every position p. We now define $e = f(d)$ recursively:

- For $k = 0$, $f(d) = \langle\rangle$;
- For $k = l + 1$, if $p_l = r^{-1}_{T,R}(p)q$ for some maximal $p \in \mathrm{Pos}(R)$ and some q, set $f(d) = f(d_0)\langle a_l \to S_l, pq\rangle$ where $d_0 = \langle a_i \to S_i, p_i\rangle_{i<l}$, otherwise set $f(d) = f(d_0)$.

$f(d)$ is well-defined since condition 1. ensures that $f(d)$ is invariant under the choice of the inverse r^{-1}, and condition 2. guarantees that every applicable production rule in d will be also freely applicable in $f(d)$. □

5 Reductions

We begin with showing that the simplest reductions yield homomorphisms.

Lemma 5.1. *Suppose $\pi \rightsquigarrow \pi'$ is an instance of a reduction rule in Figure 1 or 2 except Contraction or Quantifier Reduction. Then $\mathcal{L}(\mathcal{G}_{\pi'}) \subseteq$ Ł$(\mathcal{G}_\pi)$. If the reduction is not a case of Weakening Reduction then in fact $\mathcal{L}(\mathcal{G}_{\pi'}) =$ Ł$(\mathcal{G}_\pi)$.*

Proof: Suppose $\pi \rightsquigarrow \pi'$ is and instance of Axiom Reduction and

$$\dfrac{\begin{matrix}\pi_0\\ \vdots\\ \Gamma, A\end{matrix} \qquad A, \bar{A}}{\pi|p \vdash \Gamma, A}\,\text{cut} \qquad\qquad \begin{matrix}\pi_0\\ \vdots\\ \pi'|p \vdash \Gamma, A\end{matrix}$$

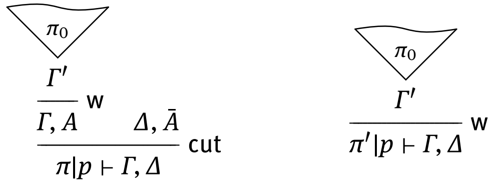
A homomorphism $f\colon \mathcal{G}_{\pi'} \to \mathcal{G}_\pi$ can be easily given by a function which maps a non-terminal v^{pq}_F for $v \in \{\sigma, \tau\}$ and $pq \in \mathrm{Pos}(\pi')$ with $q \neq \langle\rangle$ to the corresponding ones in π at position $p0q$ and is identity on all other non-terminals. Consider a rule $\sigma^p_F \to S$ in π' then $\sigma^{0p} \to f(S)$ is a production rule in π. If $F \in \Gamma$, $\sigma^p_F \to \sigma^{p0}_F$ is also a production rules in $\mathcal{G}_\pi$; otherwise, $F = A$ and the sequence

$$\sigma^p_A \to \sigma^{p1}_A \to \tau^{p1}_{\bar{A}} \to \sigma^{p0}_A$$

is a derivation in π. In either case, we have a derivation $f(\sigma^p_F) \to f(S)$ in π as required. Conversely, one can define a homomorphism $f\colon \mathcal{G}_\pi \to \mathcal{G}_{\pi'}$ by the function below, hence $\mathcal{L}(\mathcal{G}_\pi) =$ Ł$(\mathcal{G}_{\pi'})$.

$$f(v^p_F) = v^p_F \qquad\qquad f(v^{p0q}_F) = v^{pq}_F$$
$$f(\sigma^{p1}_A) = f(\tau^{p1}_{\bar{A}}) = \sigma^p_A \qquad\qquad f(\tau^{p1}_A) = f(\sigma^{p1}_{\bar{A}}) = \tau^p_A$$

Now consider an instance of Weakening Reduction. Suppose

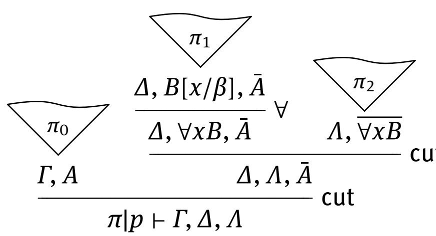


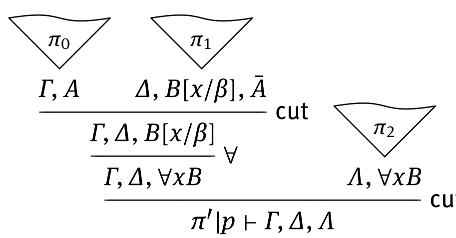
where $\Gamma' \subseteq \Gamma$. A homomorphism from $\mathcal{G}_{\pi'}$ to $\mathcal{G}_\pi$ is defined as follows. The function $f\colon \mathcal{N}_{\pi'} \to \mathcal{N}_\pi$ is determined by

$$f(v_F^{pq}) = v_F^{p0q} \qquad \text{for } v \in \{\sigma, \tau\} \text{ and every } q \neq \langle\rangle \text{ with } pq \in \mathrm{Pos}(\pi'|p);$$
$$f(\gamma) = \gamma \qquad \text{for all other non-terminals.}$$

Production rules from π' are readily mapped to derivation in π and the remaining conditions of d are trivially met.

Suppose instead that $\pi \rightsquigarrow \pi'$ is a case of rule permutation. Of particular interest is if the reduction is the permutation of two cuts, for example $\pi|p$ and $\pi'|p$ are of the form





where A is a Π_2 formula and B is Σ_1.

We begin showing $\mathcal{L}(\mathcal{G}_{\pi'}) \subseteq Ł(\mathcal{G}_\pi)$. This is achieved by considering the homomorphism that maps, for $i = 0, 1, 2$, the copy of $\mathcal{G}_{\pi_i}$ in $\mathcal{G}_{\pi'}$ to the copy in $\mathcal{G}_\pi$. Explicitly, we define $f\colon \mathcal{N}_{\pi'} \to \mathcal{N}_\pi$ as follows. For each position q, formula F and $v \in \{\sigma, \tau\}$ set

$$f(v_F^{p000q}) = v_F^{p0q} \qquad f(v_F^{p001q}) = v_F^{p100q} \qquad f(v_F^{p1q}) = v_F^{p11q}.$$

The non-terminals arising from positions $p0$ and $p00$ in π' are then appropriately mapped to those in π so connections between the sub-grammars are preserved:

$$f(v_F^{p0}) = \begin{cases} v_F^{p0} & \text{if } F \in \Gamma \\ v_F^{p10} & \text{if } F \in \Delta \cup \{\forall x B\} \end{cases} \qquad f(v_F^{p00}) = \begin{cases} v_F^{p0} & \text{if } F \in \Gamma \\ v_F^{p100} & \text{if } F \in \Delta \cup \{B[x/\beta]\} \end{cases}$$

In all other cases (i.e. a is a non-terminal of $\mathcal{G}_{\pi'}$ but not of $\mathcal{G}^p_{\pi'|p}$) we set $f(a) = a$.

We now argue that if A is not quantifier free and the last rule in π_0 is not weakening then f is a homomorphism. Since π is a simple proof, A has the form $\forall x A_0$ for some Σ_1 formula A_0 and there is an eigenvariable $\alpha \in \mathrm{EV}(\pi_0)$ such that the last inference rule of π_0 is

$$\frac{\Gamma, A_0[x/\alpha]}{\Gamma, A}\ \forall$$

By the definition of $\lhd_\pi$ notice that $\xi \lhd_\pi \alpha \lhd_\pi \tau_A^{p0}$ for every $\xi \in \mathrm{EV}(\pi_0) \setminus \{\alpha\}$. It is clear that f respects the ordering $\lhd'_\pi$ when restricted to non-terminals from either π_0, π_1 or π_2. The only other cases we need show are

$$f(\tau_A^{p000}) \lhd_\pi f(\sigma_{\bar{A}}^{p001}) \qquad \xi \lhd_\pi \beta \lhd_\pi f(\tau_{\forall x B}^{p0}) \lhd_\pi f(\sigma_{\overline{\forall x B}}^{p1})$$

for every $\xi \in \mathrm{EV}(\pi_0)$. The first inequation is given by the definition of $\lhd_\pi$. The second is determined by observing

$$\xi \lhd_\pi \tau_A^{p0} \lhd_\pi \sigma_{\bar{A}}^{p1} \lhd_\pi \sigma_{\bar{A}}^{p100} \lhd_\pi \beta \lhd_\pi \tau_{\forall x B}^{p10} \lhd_\pi \sigma_{\overline{\forall x B}}^{p11}.$$

As f is injective on eigenvariables in π' condition 6 is also satisfied and by Corollary 4.3 we are done.

In the other cases, namely that A is quantifier free or A has no eigenvariable associated to it in π_0, f need not be a homomorphism because for an eigenvariable $\xi \in \mathrm{EV}(\pi_0)$ we will have $\xi \not\lhd_\pi \beta$ but $\xi \lhd'_\pi \beta$. Nevertheless, by the Cut Partition Lemma 4.4 it is simple to show that f maps rigid derivations in $\mathcal{G}_{\pi'}$ to rigid derivations in $\mathcal{G}_\pi$.

The converse direction, $\mathcal{L}(\mathcal{G}_\pi) \subseteq \text{Ł}(\mathcal{G}_{\pi'})$, is witnessed by the function $f\colon \mathcal{N}_\pi \to \mathcal{N}_{\pi'}$ given by

$$f(v_F^{p0q}) = v_F^{p000q} \qquad f(v_F^{p100q}) = v_F^{p001q} \qquad f(v_F^{p11q}) = v_F^{p1q}$$

$$f(v_F^{p10}) = \begin{cases} v_F^{p001} & \text{if } F = \overline{A} \\ v_F^{p0} & \text{otherwise} \end{cases} \qquad f(v_F^{p1}) = \begin{cases} v_F^{p0} & \text{if } F \in \Delta \\ v_F^{p001} & \text{if } F = \overline{A} \\ v_F^{p1} & \text{otherwise} \end{cases}$$

for each q, F and $v \in \{\sigma, \tau\}$, and setting $f(a) = a$ in all other cases which in all cases is a homomorphism. □

We now proceed with Contraction Reduction. In this case the obvious renaming of non-terminals is not injective and additional properties are required in order to verify the preservation of rigidity i.e. condition 6 of homomorphism.

For the remainder of the section we use symbols x and y (possibly with indices) as meta-variables for positions in terms to contrast with positions in proofs which will be denoted by p, q, etc.

Lemma 5.2 (Contraction Reduction). *If π, π' are simple Π_2-proofs and $\pi \rightsquigarrow \pi'$ is an instance of Contraction Reduction, $\mathcal{L}(\mathcal{G}_{\pi'}) = \mathcal{L}(\mathcal{G}_\pi)$.*

Proof: Suppose that the reduction $\pi \rightsquigarrow \pi'$ is an instance of Contraction Reduction to position p as follows:

$$\dfrac{\begin{array}{c}\pi_0\\ \Gamma, A\end{array} \qquad \dfrac{\begin{array}{c}\pi_1\\ \Delta, \bar{A}, \bar{A}^*\end{array}}{\Delta, \bar{A}}\,\text{c}}{\pi|p \vdash \Gamma, \Delta}\,\text{cut} \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\pi_0\\ \Gamma, A\end{array} \qquad \dfrac{\begin{array}{c}\pi_0^*\\ \Gamma^*, A^*\end{array} \quad \begin{array}{c}\pi_1\\ \Delta, \bar{A}, \bar{A}^*\end{array}}{\Gamma^*, \Delta, \bar{A}}\,\text{cut}}{\Gamma, \Gamma^*, \Delta}\,\text{cut}}{\pi'|p \vdash \Gamma, \Delta}\,\text{c}$$

Note that we have assumed the contraction is on a single formula $\bar{A}$; the case for contraction of multiple formulæ is analogous. Consider the function $f\colon \mathcal{N}_{\pi'} \to \mathcal{N}_\pi$ defined by

$$\begin{array}{rlrl}
f(\gamma) = f(\gamma^*) = \gamma & \text{for } \gamma \in \mathrm{EV}(\pi_0) & f(v_F^{p00q}) = f(v_{F^*}^{p010q}) = v_F^{p0q} & \\
f(\delta) = \delta & \text{for } \delta \in \mathrm{EV}(\pi_1) & f(v_F^{p011q}) = v_F^{p10q} & \\
f(v_F^{p01}) = v_F^{p10} & \text{for } F \in \Delta \cup \{\bar{A}\} & f(v_{F^*}^{p01}) = v_F^{p0} & \text{for } F \in \Gamma \\
f(v_F^{p0}) = v_F^{p10} & \text{for } F \in \Delta & f(v_{F/F^*}^{p0}) = v_F^{p0} & \text{for } F \in \Gamma \\
f(a) = a & & &
\end{array}$$

It is easy to verify that f transforms production rules in $\mathcal{G}_{\pi'}$ to derivations in $\mathcal{G}_\pi$. For instance, the production rules $\tau_A^{p00} \to \sigma_{\bar{A}}^{p01}$ and $\tau_{A^*}^{p010} \to \sigma_{\bar{A}^*}^{p011}$, from the cuts in π' on A and A^*, are transformed under f to $\tau_A^{p0} \to \sigma_{\bar{A}}^{p1} \to \sigma_{\bar{A}}^{p10}$ and $\tau_A^{p0} \to \sigma_{\bar{A}}^{p1} \to \sigma_{\bar{A}^*}^{p10}$ respectively.

Condition 6 is the only non-trivial case to verify. Suppose, in search of a contradiction, $d = \langle (a_i \to S_i), y_i \rangle_{i<\mathrm{lh}(d)}\colon a_0 \to S$ is a strongly rigid derivation in $\mathcal{G}_{\pi'}$ and $i_0, i_1 < \mathrm{lh}(d)$ are such that i) $i_0 \approx_d i_1$, ii) $f(a_{i_0}) = f(a_{i_1}) \in R_\pi$ and iii) $a_{i_0} \neq a_{i_1}$. We may assume d is the shortest rigid derivation satisfying these conditions, so in particular for every $i < \mathrm{lh}(d)$, $y_i \leq y_{i_0}$ or $y_i \leq y_{i_1}$, both hold only if $i = 0$, and $\mathrm{lh}(d) = \max\{i_0 + 1, i_1 + 1\}$. By the definition of f, (ii) and (iii) we may assume $a_{i_0} = \gamma \in \mathrm{EV}(\pi_0)$ and $a_{i_1} = \gamma^* \in \mathrm{EV}(\pi_0^*)$. The argument breaks into three cases:

1. d writes one of the two non-terminals $\sigma_{A^*}^{p010}$ or σ_A^{p00};
2. $a_0 \in \mathrm{EV}(\pi_0) \cup \mathrm{EV}(\pi_0^*)$ or $a_0 \in \{\sigma_F^{p00q}, \sigma_F^{p010q}\}$ for some $q \in \mathrm{Pos}(\pi_0)$ and simple formula F;
3. there exist distinct $j_0, j_1 < \mathrm{lh}(d)$ and formulæ $F_0, F_1 \in \Gamma$ such that for each $k \in \{0,1\}$, $0 < j_k < i_k$, $y_{j_k} \leq y_{i_k}$, $a_{j_0} = \sigma_{F_0}^{p0}$ and $a_{j_1} = \sigma_{F_1^*}^{p0}$.

Notice that the failure of 1 implies either 3 or that a_0 is a non-terminal from one of π_0 or π_0^*, a condition that, by the minimality of d, is equivalent to 2.

First suppose 1 holds. Due to the minimality of d the derivation must pass through (from right to left) at least one of the two marked cuts, so there is $i \geq j$ such that either $a_i = \tau_{\bar{A}}^{p01}$ and $S_i = \sigma_A^{p00}$, or $a_i = \tau_{\bar{A}^*}^{p011}$ and $S_i = \sigma_{A^*}^{p010}$. This is only possible if $\bar{A}$ contains a universal quantifier and so, by simplicity of π, $\bar{A}$ cannot be in the class Π_2, that is, $A = \forall x \exists y A_0$ for some quantifier-free A_0. But then we may find an eigenvariable of $\mathrm{EV}(\pi_1)$ that is written by d and contradicts assumption (i).

Suppose 2 holds; in particular that $a_0 \in \mathrm{EV}(\pi_0)$ or $a_0 = \sigma_F^{p00q}$ for some $q \in \mathrm{Pos}(\pi_0)$ and simple formula F. Since $a_{i_1} \in \mathrm{EV}(\pi_0^*)$ and $y_0 < y_{i_1}$ there exists $j_0 < i_1$ such that $y_0 < y_{j_0} \leq y_{i_1}$ and either $a_{j_0} = \tau_A^{p00}$ or $a_{j_0} \in \{\tau_F^q \mid q \leq p \text{ and } F \text{ is simple}\}$ (if the production rule in d giving rise to a_{j_0} writes an eigenvariable from $\mathrm{EV}(\pi') \setminus \mathrm{EV}(\pi_0)$ then in fact $a_{j_0} = \tau_F^q$ for some $q < p$ and some F). Assuming 1 does not hold the latter case will hold for some choice of j_0 whence there also exists $j_0 \leq j_1 < i_1$ such that $a_{j_1} \in \{\sigma_F^p \mid F \in \Gamma\}$. But then there will be $j_0 < i < j_1$ such that $y_i \leq y_{j_1} \leq y_{i_1}$, $a_i \in \mathrm{EV}(\pi') \setminus \mathrm{EV}(\pi'|p)$ and $\xi \lhd'_\pi a_i$ for every $\xi \in \mathrm{EV}(\pi'|p)$, contradicting (i).

If 3 holds then similar to the above argument there must exist $i < \mathrm{lh}(d)$ such that $a_i \in \mathrm{EV}(\pi') \setminus \mathrm{EV}(\pi'|p)$ and $\xi \lhd'_\pi a_i$ for every $\xi \in \mathrm{EV}(\pi')$.

By Lemma 4.2 we conclude $\mathcal{L}(\mathcal{G}_{\pi'}) \subseteq \text{Ł}(\mathcal{G}_\pi)$. The converse inclusion holds because f is, in a suitable sense, surjective. In particular, for every rigid derivation d in $\mathcal{G}_\pi$ there exists a derivation d' in $\mathcal{G}_{\pi'}$ such that d is a permutation of $f(d')$. d' is chosen by replacing each sub-derivation that resides wholly within π_0 by its counterpart in π_0^* if this sub-derivation is immediately preceded or succeeded by the production rules $\tau_{\bar{A}^*}^{p10} \to \tau_{\bar{A}}^{p1}$ or $\sigma_{\bar{A}}^{p1} \to \sigma_{\bar{A}^*}^{p10}$. This operation is guaranteed to yield a derivation in $\mathcal{G}_{\pi'}$ by the Partition Lemma for Cut which implies that there is no derivation in $\mathcal{G}_\pi$ that write σ_A^{p0} to a term containing τ_A^{p0} and involves only rules taken from π_0. □

Before we proceed with Quantifier Reduction we need the following lemmas.

Lemma 5.3. *If π is a simple Π_2-proof, $d\colon \tau_A^p \to S$ is a derivation in $\mathcal{G}_\pi$ and $S|x = \alpha \in \mathrm{EV}(\pi|q)$ for some $x \in \mathrm{Pos}(S)$ and $p < q \in \mathrm{Pos}(\pi)$ then there exist $y < x$ and $\beta \in \mathrm{EV}(\pi)$ such that $S|y = \lambda\beta.(S|y0)$ and $\alpha \lhd \beta$.*

Proof: By examination of the definition of $\mathcal{G}_\pi$. □

Lemma 5.4 (Substitution). *Let π be a regular proof, s a term whose free variables do not appear in* $\mathrm{EV}(\pi)$ *and* $\alpha \notin \mathrm{EV}(\pi)$. *Then for every $d\colon M \to N$ in $\mathcal{G}_{\pi[\alpha/s]}$ there exists a derivation $d'\colon M' \to N'$ in $\mathcal{G}_\pi$ such that $M = M'[\alpha/s]$ and $N = N'[\alpha/s]$ and d' is strongly rigid iff d is strongly rigid.*

Proof: Let f be the natural homomorphism from $\mathcal{G}_\pi$ to $\mathcal{G}_{\pi[\alpha/s]}$. Notice f is surjective on derivations up to permutation. □

Let $d = \langle (a_i \to S_i), y_i \rangle_{i<\mathrm{lh}(d)} \colon S \to T$ be a derivation. Recall the relation $\sim_d$ defined in Section 3.3. We could instead define $\sim_d$ on the set $\mathrm{Pos}(T)$. For $x_0, x_1 \in \mathrm{Pos}(T)$, $x_0 \sim_d x_1$ if and only if there exists $j_0, j_1 < \mathrm{lh}(d)$ such that

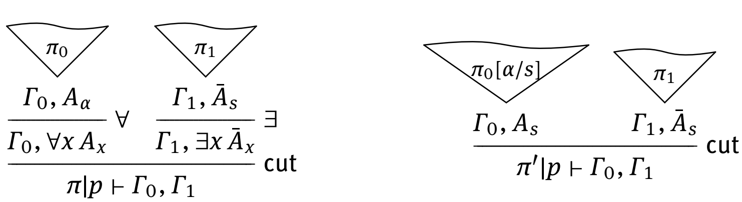
1. $x_0 = y_{j_0}$ and $x_1 = y_{j_1}$,
2. $a_{j_0} = a_{j_1} \in \mathcal{R}$,
3. for every $i_0 < i < \mathrm{lh}(d)$ and $k \in \{0, 1\}$, if $a_i \in R$ and $y_i \leq y_{j_k}$ then $a_{j_i} \ntriangleleft a_i$.

The relation $\sim_d$ on positions (as opposed to indices) is not in general transitive but it suffices to define rigidity and is a useful alternative. For example,

Lemma 5.5. *Let $d\colon S \to T$ be a derivation and suppose d' is a permutation of d. If d' is a derivation and $\sim$ is defined on positions then $\sim_d = \sim_{d'}$. In addition, $\sim_d$ is independent of the choice of S.*

Lemma 5.6 (Quantifier Reduction). *If π and π' are the simple Π_2-proofs and $\pi \rightsquigarrow \pi'$ is an instance of Quantifier Reduction $\mathcal{L}(\mathcal{G}_{\pi'}) = \text{Ł}(\mathcal{G}_\pi)$.*

Proof: Suppose that the reduction $\pi \rightsquigarrow \pi'$ is an instance of Quantifier Reduction to position p as follows:



where A_t abbreviates $A[\alpha/t]$. Let $\mathcal{G} = \mathcal{G}_\pi$ and $\mathcal{G}' = \mathcal{G}_{\pi'}$. We begin showing $\mathcal{L}(\mathcal{G}_{\pi'}) \subseteq \text{Ł}(\mathcal{G}_\pi)$.

Suppose $a \in N_{\mathcal{G}'}$ and let $d\colon a \to T$ be a full strongly rigid derivation in $\mathcal{G}'$. We may assume d is partitioned according to the Partition Theorem 4.5, whence $d = d_0 d_1 \cdots d_{2L}$ and for each $i \leq L$, $d_i\colon S_i \to S_{i+1}$ is a full derivation in $\mathcal{G}' \setminus \mathcal{G}'_p$, if i is even, and in $\mathcal{G}'_p$ otherwise. We determine a term $T' \rightarrowtail^* T$ and rigid derivation $d'\colon a \to T'$ in $\mathcal{G}$ by recursion on L.

If $L = 0$, $d = d_0$ is a derivation solely in $\mathcal{G}' \setminus \mathcal{G}'_p$ and hence also a derivation in $\mathcal{G}$. So suppose $L > 0$. The Cut Partition Lemma 4.4 determines a strongly rigid permutation of d_1 of the form $e_0 e_1 e_2 e_3 e_4$ where

- $e_0 \colon S_1 \to S_1^1$ is derivation using only rules in $\{\sigma_F^p \to \sigma_F^{pi} \mid i < 2 \wedge F \in \Gamma_i\}$;
- $e_1 \colon S_1^1 \to S_1^2$ is a full derivation in the grammar $\mathcal{G}'_{p0}$;
- $e_2 \colon S_1^2 \to S_1^3$ is a full derivation in $(\mathcal{G}'_{p1} \setminus \{\tau^{p1}_{\bar{A}_s}\})$;
- $e_3 = \langle(\tau^{p1}_{\bar{A}_s} \to \sigma^{p0}_{A_s}), x_i\rangle_{1 \le i \le k} \colon S_1^3 \to S_1^4 = S_1^3[\tau^{p1}_{\bar{A}_s}/\sigma^{p0}_{A_s}]$;
- $e_4 = e_{4,1}^{x_1} \cdots e_{4,k}^{x_k} \colon S_1^4 \to S_2$ where for each $1 \le i \le k$, $e_{4,i}$ is a derivation of $S_2|x_i$ from $\sigma^{p0}_{A_s}$ using only rules in $\mathcal{G}'_{p0}$, and $e_{4,i}^{x_i}$ is the induced derivation of $S_1^4[x_i/S_2|x_i]$ from S_1^4 obtained by adjusting $e_{4,i}$ to S_1^4.

The Substitution Lemma 5.4 provides translations of e_1 and e_4 to derivations $\hat{e}_1 \colon S_1^1 \to \hat{S}_1^2$ and $\hat{e}_4 = \hat{e}_{4,1}^{x_1} \cdots \hat{e}_{4,k}^{x_k} \colon \hat{S}_1^4 \to \hat{S}_2$ in $\mathcal{G}_{p0}$ such that

$$S_1^2 = \hat{S}_1^2[\tau^{p00}_{A_\alpha}/\tau^{p0}_{A_s}][\alpha/s] \qquad S_1^4 = \hat{S}_1^4[\sigma^{p0}_{A_s}/\sigma^{p00}_{A_\alpha}] \qquad S_2 = \hat{S}_2[\alpha/s].$$

Within $\mathcal{G}_{\pi|p}$ there is a simple derivation of s from α (i.e., $\alpha \to \tau^{p0}_{\forall x A_x} \to \sigma^{p1}_{\exists x \bar{A}_x} \to s$) which, when combined with $\hat{e}_1$ and the natural translation of e_2 into $\mathcal{G}_{p10}$, yields a derivation $e = e_0 \hat{e}_1 \hat{e}_2 \colon S_1 \to \hat{S}_1^3 = S_1^3[\tau^{p1}_{\bar{A}_s}/\tau^{p10}_{\bar{A}_s}]$.

The derivation e_3, when translated to $\mathcal{G}_{\pi|p}$,[4] becomes

$$\hat{e}_3 \colon \hat{S}_1^3 \to S_1^3[\tau^{p1}_{\bar{A}_s}/(\lambda\alpha.\, \sigma^{p00}_{A_\alpha}) \cdot s]$$

and may be connected with e and $\hat{e}_{4,1}^{x_1 00} \cdots \hat{e}_{4,k}^{x_k 00}$ to yield the derivation $\hat{e}$ given by

$$e_0 \hat{e}_1 \hat{e}_2 \hat{e}_3 \hat{e}_{4,1}^{x_1 00} \cdots \hat{e}_{4,k}^{x_k 00} \colon S_1 \to S_2' = S_2[x_1/(\lambda\alpha.\hat{S}_2|x_1) \cdot s] \cdots [x_k/(\lambda\alpha.\hat{S}_2|x_k) \cdot s].$$

Clearly $S_2' \rightarrowtail_\alpha S_2$. All that remains is to show that the derivation $d_2 \cdots d_{2L} \colon S_2 \to T$ 'lifts' to a derivation $\hat{d}_2 \colon S_2' \to \hat{T}$ such that $d_0 \hat{e} \hat{d}_2 \colon a \to \hat{T}$ is rigid and $\hat{T} \rightarrowtail T$ as then applying the induction hypothesis to $\hat{d}_2$ will determine a rigid derivation $d' \colon a \to T'$ in $\mathcal{G}$ such that $T' \rightarrowtail T$.

First we note the following. Suppose y_0 and y_1 are distinct positions in S_2 such that $S_2|y_0 = S_2|y_1 \in \mathcal{R}_{\mathcal{G}'}$ and $r_{S_2,S_2'}(y_0) \sim_{d_0\hat{e}} r_{S_2,S_2'}(y_1)$. If neither y_0 nor y_1 has the form $x_i 1 y$ for any $i \le k$ and position y then by the choice of $\hat{e}$ and strong rigidity of d in fact $y_0 \sim_{d_0 d_1} y_1$. If $y_0 \ge x_i 1$ however, then since $S_2|y_0$ is an eigenvariable outside $\pi|p$, also $y_0 \sim_{d_0 d_1} y_1$.

We can now verify that the lifting lemma can indeed be applied to $d_2 \cdots d_{2L}$ to obtain a term $\hat{T} \rightarrowtail T$ and derivation $\hat{d}_2 \colon S_2' \to \hat{T}$: The first condition for lifting

4 Each $(\tau^{p1}_{\bar{A}_s} \to \sigma^{p0}_{A_s})$ is readily translated to $\tau^{p10}_{\bar{A}_s} \to \tau^{p1}_{\exists x \bar{A}_x} \cdot s \to \sigma^{p0}_{\forall x A_x} \cdot s \to (\lambda\alpha.\sigma^{p00}_{A_\alpha}) \cdot s$.

is provided by the argument in the previous paragraph in the case $r_{S_2,S_2'}(y_0) = r_{S_2,S_2'}(y_1)$; the second condition follows from Lemma 5.3. The previous argument also suffices to deduce that $d_0\hat{e}\hat{d}_2$ is rigid. As applying the induction hypothesis to $\hat{d}_2$ does not affect rigidity, we are done.

The converse direction, namely $\mathcal{L}(\mathcal{G}_\pi) \subseteq Ł(\mathcal{G}_{\pi'})$, is more straightforward. Every derivation e starting from α with $\mathrm{lh}(e) \geq 3$ will have $e(1) = \tau^{p0}_{\forall x A_x}$, $e(2) = \sigma^{p1}_{\overline{\forall x A_x}}$ and $e(3) = s$. Similarly, for any derivation e' in $\mathcal{G}_\pi$, if $i < \mathrm{lh}(e')$ and $e'(i)|x$ has the form $(\lambda\alpha.S_0) \cdot S_1$ then $S_1 = s$. These observations allow a simple translation of derivations in $\mathcal{G}_\pi$ to derivations in $\mathcal{G}_{\pi'}$ preserving rigidity. Let $d\colon a \to S$ be a derivation in $\mathcal{G}_\pi$ where a and S contain neither $\tau^{p1}_{\overline{\forall x A}}$ nor $\sigma^{p0}_{\forall x A}$, and let $S \rightarrowtail^*_\alpha S'$ be the term in which all abstractions on α have been reduced. Then there exists a derivation $d'\colon f(a) \to f(S')$ where $f\colon \mathcal{N}_\pi \to \mathrm{Terms}(\Sigma_\pi \cup \mathcal{N}_{\pi'})$ is defined by

$$f(\alpha) = f(\tau^{p0}_{\forall x A_x}) = f(\sigma^{p1}_{\exists x \bar{A}_x}) = s \qquad f(v_F^{p00q_0}) = v^{p0q_0}_{F[\alpha/s]} \qquad f(v_F^{p10q_1}) = v_F^{p1q_1}$$

for $v \in \{\sigma, \tau\}$, $q_0 \in \mathrm{Pos}(\pi_0)$ and $q_1 \in \mathrm{Pos}(\pi_1)$, and $f(a) = a$ in all other cases. Finally we may deduce that d' is rigid whenever d is strongly rigid by an induction on $\mathrm{lh}(d)$ using Lemma 4.7. As f is the identity function on terms in $\mathrm{Terms}(\Sigma_\pi)$, we conclude $\mathcal{L}(\mathcal{G}_\pi) \subseteq Ł(\mathcal{G}_{\pi'})$. □

6 Conclusion

We have established the following theorems.

Theorem 6.1. *Let $\pi_0, \pi_1, \dots, \pi_k$ be a sequence of simple Π_2-proofs such that π_{i+1} is obtained from π_i by the reduction steps outlined in Section 2.1. Then $\mathcal{L}(\mathcal{G}_{\pi_k}) \subseteq Ł(\mathcal{G}_{\pi_0})$. If, moreover, no step in the reduction is an instance of weakening reduction then $\mathcal{L}(\mathcal{G}_{\pi_0}) = Ł(\mathcal{G}_{\pi_k})$.*

Theorem 6.2. *Let $\pi \vdash \Gamma$ be a simple Π_2-proof where Γ is a set of Σ_1 formulæ. There exists a grammar $\mathcal{G}$ such that*

1. $|\mathrm{Pr}_\mathcal{G}| \leq |\pi| + |\Gamma|$,
2. $|Ł(\mathcal{G})| \leq 2^{2^{2|\pi|}}$,
3. *The formula $\bigvee_{F\in\Gamma} F^{\{t|(F,t)\in Ł(\mathcal{G})\}}$ is valid,*
4. *If π' is obtained from π by a sequence of simple Π_2-proofs using the reduction steps of Section 2.1 and all cuts in π' are on quantifier-free formulæ then the Herbrand set for π', $\mathcal{H}(\pi')$, is contained in $\mathcal{L}(\mathcal{G})$; if no application of weakening reduction is used then $\mathcal{H}(\pi') = Ł(\mathcal{G})$.*

Proof: Let $\mathcal{G}$ be the modification of $\mathcal{G}_\pi$ which contains a single (fresh) start symbol σ and production rules $\sigma \to (F, \sigma_F)$ for each $F \in \Gamma \cap \Sigma_1$. 1. is immediate from the definition of $\mathcal{G}_\pi$, 2. is a consequence of Lemma 3.14 and 4. is a corollary of the previous theorem and the observation that if π contains only cuts on quantifier-free formulæ then, up to β-equivalence of terms, $\mathcal{H}(\pi) = Ł(\mathcal{G})$. Finally, 3. is deducible from the first part of 4. by observing that standard cut elimination strategies, such as in [Troelstra and Schwichtenberg, 1996], preserve simplicity of proofs. □

These results may be extended to proofs involving prenex Π_2 and Σ_2 formulæ. This can be achieved by endowing the simply typed λ-calculus with product types and pairing functions in the usual way. In addition to the typing definition for simple Π_1 and Σ_1 formulæ defined in the paper, for each position q and prenex Π_2 formula $A = \forall x B$ the two non-terminals τ^q_A and $\sigma^q_{\bar{A}}$ are assigned type type$(\tau^q_A) = o \times \text{type}(\tau^q_B)$, and σ^q_A and $\tau^q_{\bar{A}}$ have the dual type $o \to \text{type}(\sigma^q_B)$ where o denotes the type of first-order terms. In other words, if $A = \forall x_1 \cdots \forall x_m \exists y_1 \cdots \exists y_n B$ and B is quantifier free then

$$\text{type}(\tau^q_A) = \text{type}(\sigma^q_{\bar{A}}) = o^{m+1} = \underbrace{o \times (o \times \cdots \times o)}_{m+1}$$

$$\text{type}(\sigma^q_A) = \text{type}(\tau^q_{\bar{A}}) = \underbrace{o \to o \to \cdots \to o}_{m} \to o^{n+1}.$$

The production rules for existential and universal introduction in Table 2 then become respectively

$$\text{prod}(p, \Pi, \emptyset) \cup \{\sigma^p_{\exists y A} \to (s, \sigma^{p0}_{A[y/s]}),\ \tau^{p0}_{A[y/s]} \to \tau^p_{\exists y A} \cdot s\},$$

$$\text{prod}(p, \Pi, \emptyset) \cup \{\alpha \to (\tau^p_{\forall x A})_0,\ \tau^{p0}_{A[x/\alpha]} \to (\tau^p_{\forall x A})_1,\ \sigma^p_{\forall x A} \to \lambda\alpha.\sigma^{p0}_{A[x/\alpha]}\}$$

where $(.,.)$ is a (polymorphic) binary pairing function with first and second projections $(.)_0$ and $(.)_1$. Aside from a few technical considerations all the lemmas and proofs presented here generalise to proofs with cuts and end-sequents of prenex Π_2 and Σ_2 formulæ.

6.1 Future work

We have shown the language of grammars defined for simple Π_2-proofs is invariant under most cut reduction rules. It would be interesting to investigate under which further transformations language invariance is maintained. For example, as part of Lemma 5.1 we have shown that permuting two cuts does not change the language of the proof grammar. In other words, composition by cut is an associative operation which, as another interesting line of future work, would allow the consideration of a category of proofs in the spirit of [Hyland, 2002]. Note, the results and techniques presented in this paper are independent of the exact syntactic variant of the sequent calculus used and apply to two-sided, additive, etc. calculi as well.

Further generalisation of this work would involve larger classes of cut-formulæ of which there are two natural directions to proceed. The first is to generalise to cuts on a wider class of Π_2 and Σ_2-formulæ. The simplest extension, which can already be realised, is to simple proofs with cuts on arbitrary prenex Π_2 formulæ (this generalisation is described in detail above); more complex will be considering boolean combinations of prenex Π_2 and Σ_2 formulæ and non-prenex formulæ. A second generalisation would see proof grammars extended to prenex Π_n-formulæ. The relationship between proofs and grammars established thus far suggests how to proceed: with Π_1-cuts inducing non-terminals of type level 0 and Π_2-cuts inducing, in addition, non-terminals if type level 1, the conjecture is that Π_{n+1}-cuts will be amenable to analysis using non-terminals of level $\leq n$ and order n recursion schemes (a natural extension of context-free tree grammars to higher-order). This is reminiscent to the relationship between the number of quantifier alternations in an induction and the type level of the functional obtained from Gödel's Dialectica interpretation [Gödel, 1958], see e.g. [Avigad and Feferman, 1999].

Acknowledgment: The authors' research was supported by the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) project VRG12-04.

References

Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh. Herbrand Disjunctions, Cut Elimination and Context-Free Tree Grammars. In Thorsten Altenkirch, editor, *13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015)*, volume 38 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1–16, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Jeremy Avigad and Solomon Feferman. Gödel's functional ("dialectica") interpretation. In Sam Buss, editor, *The Handbook of Proof Theory*, pages 337–405. North-Holland, 1999.

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

Samuel R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53: 75–102, 1991.

Samuel R. Buss. On Herbrand's Theorem. In *Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 1995.

Alessandra Carbone and Stephen Semmes. *A Graphic Apology for Symmetry and Implicitness*. Oxford Mathematical Monographs. Oxford University Press, 2000.

Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

Kurt Gödel. Über eine noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

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

Jacques Herbrand. *Recherches sur la théorie de la démonstration*. PhD thesis, Université de Paris, 1930.

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

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*, pages 301–312. Springer, 2012a.

Stefan Hetzl. The Computational Content of Arithmetical Proofs. *Notre Dame Journal of Formal Logic*, 53(3):289–296, 2012b.

Stefan Hetzl and Lutz Straßburger. Herbrand-Confluence for Cut-Elimination in Classical First-Order Logic. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL) 2012*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 320–334. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

Stefan Hetzl and Lutz Straßburger. Herbrand-confluence. *Logical Methods in Computer Science*, 9(4), 2013.

J. Martin E. Hyland. Proof theory in the abstract. *Annals of Pure and Applied Logic*, 114:43–78, 2002.

Richard McKinley. Proof nets for Herbrand's Theorem. *ACM Transactions on Computational Logic*, 14(1):5:1–5:31, 2013.

Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge: Cambridge University Press, 1996.