Mathematisches
Forschungsinstitut
Oberwolfach

Mitglied der Leibniz Gemeinschaft

Oberwolfach Preprints

OWP 2018 - 01

BAHAREH AFSHARI, STEFAN HETZL AND
GRAHAM E. LEIGH

Herbrand’s Theorem as Higher Order Recursion

Mathematisches Forschungsinstitut Oberwolfach gGmbH
Oberwolfach Preprints (OWP)   ISSN 1864-7596

Oberwolfach Preprints (OWP)

Imprint:

Mathematisches Forschungsinstitut Oberwolfach gGmbH (MFO)
Schwarzwaldstrasse 9-11
77709 Oberwolfach-Walke
Germany

Tel +49 7834 979 50
Fax +49 7834 979 55
Email admin@mfo.de
URL www.mfo.de

The Oberwolfach Preprints (OWP, ISSN 1864-7596) are published by the MFO.

DOI 10.14760/OWP-2018-01

Herbrand's Theorem as Higher Order Recursion

Bahareh Afshari
University of Gothenburg

Stefan Hetzl
TU Wien

Graham E. Leigh
University of Gothenburg

We provide a means to compute Herbrand disjunctions directly from sequent calculus proofs with cuts. Our approach associates to a first-order classical proof $\pi \vdash \exists v F$, where F is quantifier free, an acyclic higher order recursion scheme $\mathscr{H}$ whose language is finite and yields a Herbrand disjunction for $\exists v F$. More generally, we show that the language of $\mathscr{H}$ contains the Herbrand disjunction implicit in any cut-free proof obtained from π via a sequence of Gentzen-style cut reductions that always reduce the weak side of a cut before the strong side.

This research was supported through the programme "Research in Pairs" by the Mathematisches Forschungsinstitut Oberwolfach in 2016.

Contents

1 Introduction

The property of being a valid first-order formula is intimately tied to the consideration of the ground, i.e., variable-free, instances of that formula. This connection is apparent in most, if not all, proofs of the completeness theorem which, in one way or another, rely on the construction of a term model. It is plainly visible in Herbrand's theorem which states that 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). This feature of classical first-order logic is in contrast to both classical second-order logic, whose standard semantics goes beyond the ground instances of a countable language, and intuitionistic first-order logic, which exhibits a more complicated interaction between quantifiers and propositional connectives.

Proof-theoretically, the use of instances of a formula naturally leads to analytic, cut-free, proofs. Gentzen's mid-sequent theorem makes the close connection between Herbrand expansions and cut-free proofs apparent. Taking this perspective on the cut-elimination theorem, and thereby keeping the well-known complexity bounds in mind, shows that, in essence, cut-elimination consists of the computation of a Herbrand expansion. One may ask, however, whether given a proof with cut it is possible to compute a Herbrand expansion in a more direct way, circumventing the cumbersome process of cut-elimination. There is a number of formalisms that do just that, the historically first being Hilbert's ε-calculus [31] (see [35] for a contemporary exposition of the ε-theorems in English). In [20], Gerhardy and Kohlenbach adapt Shoenfield's variant [41] of Gödel's Dialectica interpretation [21, 5] to a system of pure predicate logic. Recent work, related to proof nets, is that of Heijltjes [22] and McKinley [33], and a similar approach, in the formalism of expansion trees [34], can be found in [30]. A different method with similar aims is cut-elimination by resolution [8].

The present work is motivated by the follow-up question: what is a minimal amount of information required for computing a Herbrand expansion from a proof with cuts? An approach which has been partially successful in answering the question is the representation of proofs as tree grammars, introduced in [24] for proofs with Π_1-cuts and extended to Π_2-cuts in [1, 2]. This emphasis on minimality plays a crucial role for several applications, such as cut-introduction [26, 25, 32], inductive theorem proving [15] and the confluence behaviour of cut-elimination [28, 29, 2]. For instance, in the case of cut-introduction, which is an attempt to automatically compress proofs via introduction of cuts (i.e. lemmas), the algorithm proceeds in two steps: 1. a smallest grammar which represents a given Herbrand expansion is computed and 2. this grammar is translated to a proof with cuts. The minimality of the grammar formalism makes the first step feasible and the second step total. Algorithms for cut-introduction and inductive theorem proving are currently being implemented in the GAPT-system [16], see e.g. [27, 17]. A further theoretical application of proof grammars is in the area of proof complexity, where lower bounds on the length of proofs with cuts (which are notoriously difficult to control) are obtained by transferring lower bounds on the size of the corresponding grammar [14, 13]. Proving these lower bounds on the size of grammars is considerably simplified by them containing only a minimal amount of information.

Continuing this research effort, we demonstrate how Herbrand expansions can be represented as languages of higher order recursion schemes derived directly from first-order proofs with cut. Higher order recursion schemes (see e.g. [37]) are a generalisation of regular tree grammars (which correspond to order-0 recursion schemes) to finite types. The representation we outline involves interpreting inference rules of proofs as non-terminals whose production rules follow the local instantiation structure of quantifiers. The type of a non-terminal is determined entirely by the quantifier complexity of the formulæ occurring in the corresponding inference, with an inference deriving $\Sigma_n \cup \Pi_n$ formulæ being represented by a non-terminal of order n. Cut corresponds to composition of non-terminals, and instances of contraction give rise to non-deterministic production rules. The language of the recursion scheme induces a Herbrand expansion for the end-sequent of the proof. At the level of Π_2-cuts, the schemes closely resemble the grammars introduced in [4]. The generic case of the representation, which permits capturing cuts of arbitrary quantifier complexity, turns out to be at the level of Π_3 where both sides of a cut feature $\exists\forall$ quantifier alternations.

As far as the authors are aware, the present work marks the first method of Herbrand extraction that operates directly on sequent calculus proofs. The main result can be summarised as follows and was announced in [3].

Theorem 1.1. *Let F be a quantifier-free formula and π a first-order proof of $\exists\vec{v}F$ in which cut-formulæ are prenex Π_n or Σ_n. There exists an acyclic order n recursion scheme $\mathscr{H}$ with language $L(\mathscr{H})$ such that: i) $\bigvee_{\vec{t}\in L(\mathscr{H})} F(\vec{t})$ is valid; ii) $|L(\mathscr{H})| \leq 2^{4|\pi|^3}_{n+2}$ where $|\pi|$ is the number of inference rules in π; iii) $L(\mathscr{H})$ contains the Herbrand set extracted from any cut-free proof that can be obtained from π via a sequence of Gentzen-style cut reductions that always reduces to the weak (quantifier) side of a cut before the strong side.*

2 Sequent Calculus for Classical First-order Logic

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, v, w, etc. Upper-case Roman letters, A, B, etc. denote formulæ and upper-case Greek letters Γ, Δ, etc. range over *sequents*, namely finite sequences of formulæ. We abbreviate by Γ, Δ the concatenation of Γ and Δ; and Γ, A is shorthand for $\Gamma, \{A\}$. The *length* of a

$$\begin{array}{llll}
\text{Axioms:} & A, \bar{A} & \text{for } A \text{ quantifier-free} & \\[1em]
\text{Inference rules:} & \vee\,\dfrac{\Gamma, A, B}{\Gamma, A \vee B} & \wedge\,\dfrac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, A \wedge B} & \\[1.5em]
 & \forall_{\vec{\alpha}}\,\dfrac{\Gamma, A(\vec{\alpha}/\vec{v})}{\Gamma, \forall\vec{v}A} & \exists_{\vec{r}}\,\dfrac{\Gamma, A(\vec{r}/\vec{v})}{\Gamma, \exists\vec{v}A} & \mathsf{cut}\,\dfrac{\Gamma, A \quad \Delta, \bar{A}}{\Gamma, \Delta} \\[1.5em]
 & \mathsf{w}\,\dfrac{\Gamma}{\Gamma, A} & \mathsf{c}\,\dfrac{\Gamma, A, A}{\Gamma, A} & \mathsf{p}\,\dfrac{\Gamma, B, A, \Delta}{\Gamma, A, B, \Delta}
\end{array}$$

Figure 1: Axioms and rules of sequent calculus

sequent Γ is denoted $|\Gamma|$. As the order of the formulæ in a sequent is often (though not always) unimportant, we will frequently identify sequents with (finite) multisets. We write $\bar{A}$ to denote the dual of the formula A obtained by de Morgan laws. Given a sequence of variable symbols $\vec{v} = (v_0, \ldots, v_{k-1})$ of length k, we write $\forall\vec{v}A$ and $\exists\vec{v}A$ as shorthand for $\forall v_0 \cdots \forall v_{k-1}A$ (resp. $\exists v_0 \cdots \exists v_{k-1}A$). If $\vec{t} = (t_0, \ldots, t_{k-1})$ is a sequence of terms of the same length, $A(\vec{t}/\vec{v})$ is the formula obtained from A by replacing each v_i by the corresponding term t_i, where bound variables in A are renamed as necessary to avoid variable capture.

The following abbreviations will be used in later sections. For a formula A, we write A_{qf} to indicate that A is quantifier-free, and $u(A)$ (resp. $e(A)$) for the number of consecutive universal (existential) quantifiers in A before encountering an existential (universal) quantifier:

$$\begin{aligned} u(\forall vA) &= u(A) + 1 & \qquad e(\exists vA) &= e(A) + 1 \\ u(\exists vA) &= u(A_{qf}) = 0 & \qquad e(\forall vA) &= e(A_{qf}) = 0 \end{aligned}$$

For notational simplicity, we work in one-sided sequent calculus with explicit structural rules for weakening (w), contraction (c) and permutation (p), though the results presented apply equally to two-sided (so-called Gentzen-style) sequent calculi and either form of calculus without explicit structural rules. The axioms and rules of the calculus are laid out in Figure 1. The quantifier introduction rules $\forall_{\vec{\alpha}}$ and $\exists_{\vec{r}}$ introduce a sequence of quantifiers in one application. Applications of $\forall_{\vec{\alpha}}$ are subject to an eigenvariable condition that if $\vec{\alpha} = (\alpha_0, \ldots, \alpha_{k-1})$ then α_i does not occur in the sequent Γ, A for any $i < k$. In each inference rule, the formulæ which are explicitly mentioned in the premise(s) (usually the right-most formula in the sequent) are said to be *active* in the rules applied. For example, A and B are active in $\wedge$ rule, both copies of A are active in contraction, and there are no active formulæ in the weakening rule. Active formulæ of cut are refereed to as *cut formulæ*. We often leave the applications of the permutation rule implicit, writing, for instance,

$$\forall_{\vec{\alpha}}\,\frac{\Gamma, A(\vec{\alpha}/\vec{v}), \Delta}{\Gamma, \forall\vec{v}A, \Delta} \qquad\qquad \mathsf{cut}\,\frac{\Gamma, A, \Gamma' \quad \Delta, \bar{A}, \Delta'}{\Gamma, \Gamma', \Delta, \Delta'}$$

to abbreviate derivations

$$\mathsf{p}^*\,\cfrac{\forall_{\vec{\alpha}}\,\cfrac{\mathsf{p}^*\,\cfrac{\Gamma, A(\vec{\alpha}/\vec{v}), \Delta}{\Gamma, \Delta, A(\vec{\alpha}/\vec{v})}}{\Gamma, \Delta, \forall\vec{v}A}}{\Gamma, \forall\vec{v}A, \Delta} \qquad\qquad \mathsf{cut}\,\cfrac{\mathsf{p}^*\,\cfrac{\Gamma, A, \Gamma'}{\Gamma, \Gamma', A} \quad \mathsf{p}^*\,\cfrac{\Delta, \bar{A}, \Delta'}{\Delta, \Delta', \bar{A}}}{\Gamma, \Gamma', \Delta, \Delta'}$$

where in each case p^* denotes a sequence of permutation inferences p, a notation we also extend to the other structural rules.

A *proof* is a finite tree labelled by sequents obtained from the axioms and rules of the calculus with the restriction that cuts apply to prenex formulæ only. Without loss of generality, we assume all proofs are *regular*, by which we mean that:

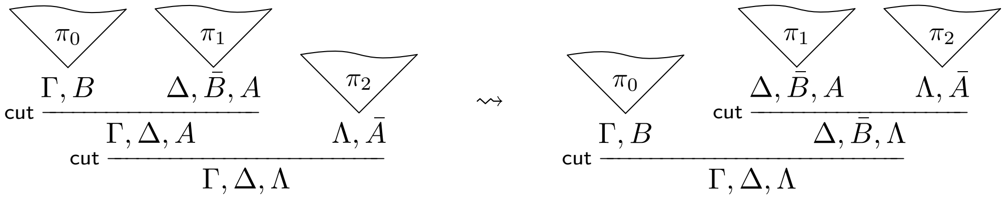
1. each eigenvariable in the proof appears in exactly one $\forall_{\vec{\alpha}}$ inference in the proof and does not occur in any sequent outside the sub-proof of this inference,

2. if A appears as the active formula of a quantifier inference $\forall$ ($\exists$) then $u(A) = 0$ (resp. $e(A) = 0$).

We write $\pi \vdash \Gamma$ to express that π is a regular proof with Γ being the sequent appearing at the root of π. $\mathrm{EV}(\pi)$ denotes the set of eigenvariables in a proof π, and for sequences $\vec{\alpha} = (\alpha_0, \ldots, \alpha_{k-1})$ and $\vec{t} = (t_0, \ldots, t_{k-1})$ of variable symbols and terms, $\pi^{(\vec{t}/\vec{\alpha})}$ is the result of replacing throughout the proof π each occurrence of the variable symbol α_i by the term t_i.

2.1 Cut Reduction and Normal Forms

The standard cut reduction and cut permutation steps are given in Figures 2 and 3. For the sake of a concise presentation, the axioms and rules are stated with implicit permutation in place. We assume all the proofs drawn in Figures 2 and 3 are regular. Hence, in the case of contraction reduction where the sub-proof π_1 is duplicated it is assumed that the eigenvariables are renamed in the copy, which is emphasised by annotating the sub-proof with an asterisk i.e. π_1^*. In the two reductions of Figure 3, r represents an arbitrary unary or binary inference rule. An example of the binary inference permutation rule for r = cut is



For proofs π and π' we write $\pi \rightsquigarrow \pi'$ to express that π' is obtained from π by application of a reduction or permutation rule to a sub-proof of π, and let $\rightsquigarrow^*$ denote the reflexive transitive closure of $\rightsquigarrow$. If $\pi \rightsquigarrow \pi'$ then the reduced cut either no longer exists, is replaced by cuts on formulæ with either lower logical complexity or fewer applied contractions, or permuted to a subproof. In any given proof there may, however, be many cuts and eliminating one can (through duplicating a sub-proof) result in introducing several copies of other cuts. To obtain a cut-free proof, it is necessary to provide a (terminating) *cut elimination strategy* i.e. a procedure that given any proof $\pi \vdash \Gamma$ induces a sequence of cut reduction and permutation steps $\pi \rightsquigarrow^* \pi'$ such that $\pi' \vdash \Gamma$ and the rule cut is not used in π'.

Theorem 2.1 (Gentzen's Hauptsatz). *There is a cut elimination strategy that transforms any proof in first-order logic to a cut-free proof.*

There are many cut elimination strategies such as top-most reduction strategy or the elimination of the cut with highest logical complexity. Different strategies provide different cut-free proofs, commonly also referred to as *normal forms.* In fact, there exist proofs with infinitely many normal forms (see e.g. [43, Example 2.1.3]). We now turn to the relationship between cut-elimination and Herbrand disjunctions in first-order logic. In the remainder of this article a *quasi cut-free proof* refers to a proof in which the only cuts are on quantifier-free formulae.

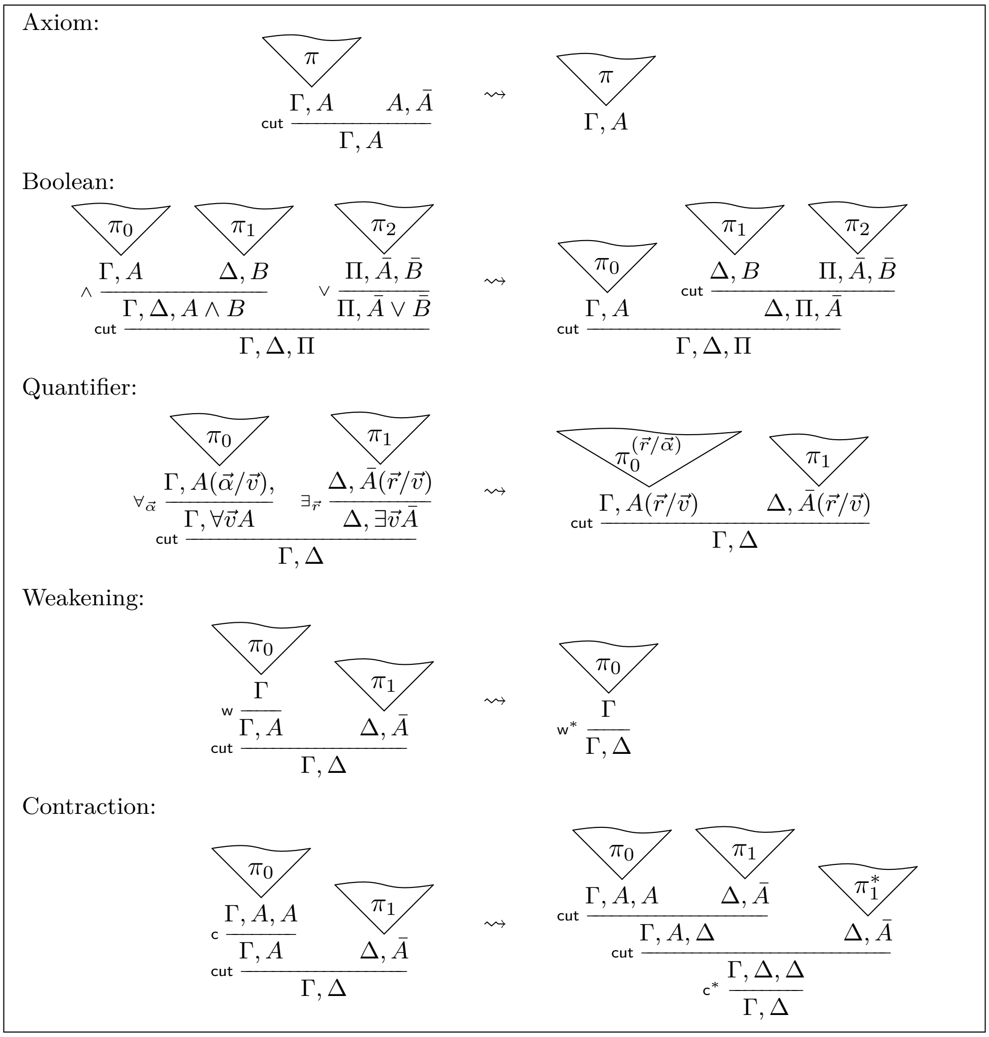


Figure 2: One-step cut reduction rules.

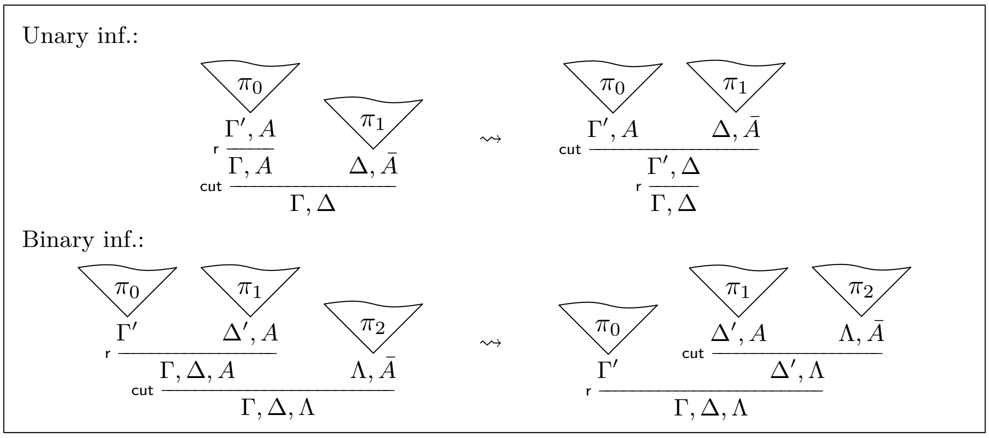


Figure 3: One-step cut permutation rules.

2.2 Herbrand's Theorem and Cut Elimination

Herbrand's theorem is considered a classic result in proof theory. It can be thought of as reducing validity in first-order logic to validity in propositional logic. From the modern perspective it can also be seen as extracting computational content to first-order proofs. A simple case of the theorem is the following.

Theorem 2.2 (Herbrand's theorem)**.** *A formula $\exists\vec{v}A_{qf}$ is valid if and only if there exists a finite set of sequences of terms $\{\vec{t_0},\ \vec{t_1},\ldots,\vec{t_k}\}$ such that $\bigvee_{i=0}^{k} A(\vec{t_i}/\vec{v})$ is valid.*

If a formula $\exists\vec{v}A_{qf}$ is valid then any set of terms $\{\vec{t_0},\ \vec{t_1},\ldots,\vec{t_k}\}$ that validate the disjunction $\bigvee_{i=0}^{k} A(\vec{t_i}/\vec{v})$ is called a *Herbrand set*, and the disjunction itself a *Herbrand disjunction* for the formula.

Herbrand's theorem pre-dates Gentzen's Hauptsatz but the latter readily provides an instructive proof of the theorem: Suppose $\exists\vec{v}A_{qf}$ is valid and fix a quasi cut-free proof $\pi \vdash \exists\vec{v}A$. It is possible to permute the rules applied in π so that no quantifier inference occurs above a purely propositional rule (Gentzen's mid-sequent theorem [18]). Once the proof is partitioned into a *propositional part* and a *quantifier part*, the terms that validate the formula can be directly read off from the *mid-sequent*, the sequent separating the two parts.

Herbrand's original statement is much more general than that stated above and applies to any formula of first-order logic thanks to *Herbrandisation*, the dual notion of Skolemization. Given an arbitrary formula A, by introducing suitable constant and function symbols it is possible to remove universal quantifiers in A and obtain a Σ_1 prenex-formula which is equi-valid to A. Herbrandisation can also be applied to a proof of a sequent Γ transforming it to a proof of the Herbandisation of Γ.

If a Herbrand set (disjunction) is obtained via cut elimination it is customary to refer to it as a Herbrand set (disjunction) of the proof. Note that these are not unique: different reduction strategies can lead to non-elementary many pairwise distinct Herbrand disjunctions [6]. For both computing and representing Herbrand disjunctions it is therefore desirable to bypass cut elimination. There has been a number of successful approaches such as via Herbrand nets [33], proof forests [22], expansion trees with cut [30] and functional interpretation [20]. In the next section we introduce a fresh approach using higher order recursion schemes which allows the extraction of Herbrand disjunctions directly from proofs in sequent calculus and represents them in a standard formalism from formal language theory.

3 Recursion Schemes for First-order Proofs

In this section we associate to each sequent calculus proof π with Σ_1 end-sequent a non-deterministic higher order recursion scheme $\mathscr{H}_\pi$. We begin with definition of the type system and terms that will be used throughout the paper. In sections 3.2 and 3.3, higher order recursion schemes over the type system are introduced and upper bounds on the size of languages of acyclic schemes are established. The definition of $\mathscr{H}_\pi$ is given in section 3.4.

3.1 Types and Terms

The type system we utilise extends the hierarchy of simple types (over a type of individuals ι) by pair types and two additional type constants. These are the unit type, denoted ϵ, and a type ς of (stacks of) substitutions, elements of which are finite sequences of pairs (α, r) where α and r are elements of some (and the same) type. We are interested specifically in the case that α is a constant symbol (of simple type) from a particular ranked alphabet Σ, and refer to the type ς as the type of *substitution stacks (over Σ)*, or simply Σ-*substitutions*.

The informal reading behind the type ς is that of an accumulator for a sequence of substitutions that are generated by reading a particular thread through a formal proof: when a witness to an existential quantifier is encountered along such a thread, the witness is outputted accompanied by the current stack of substitutions. The substitutions are not evaluated at the formal level but recorded as an element of ς.

We begin with a formal definition of the types and conventions for their representation, followed by ranked alphabets and the recursive definition of (typed) terms including the precise form of inhabitants of the type of substitution stacks.

Definition 3.1. The *types* are defined in the following way.

- ι is a type, called the type of *individuals*.
- ϵ is a type, called the *unit type*.
- ς is a type, called the type of *substitution stacks*.
- *Function types*: if ρ, σ are types then $\rho \to \sigma$ is a type.
- *Pair types*: if ρ, σ are types then $\rho \times \sigma$ is a type.

A type formed without reference to ς is called *basic*, and one formed only out of ι and $\to$ is *simple*. The types ι and ϵ are referred to collectively as *ground types* and any type that is not a function type is called *prime*. The *sequence types* are the types of the form ι^n for any n, where $\iota^0 = \epsilon$ and $\iota^{n+1} = \iota \times \iota^n$. The set of all types is denoted Type.

We follow the convention that the two type forming operations × and → associate to the right, and that → binds more strongly than ×, so for $\rho_0, \dots, \rho_k$ types we have

$$\begin{aligned} \rho_0 \times \rho_1 \times \cdots \times \rho_k &= \rho_0 \times (\rho_1 \times \cdots \times \rho_k) \\ \rho_0 \to \rho_1 \to \cdots \to \rho_k &= \rho_0 \to (\rho_1 \to \cdots \to \rho_k) \\ \rho_0 \times \cdots \times \rho_i \to \rho_{i+1} \times \cdots \times \rho_k &= (\rho_0 \times \cdots \times \rho_i) \to (\rho_{i+1} \times \cdots \times \rho_k) \end{aligned}$$

Every type ρ has a unique decomposition $\rho = \rho_1 \to \rho_2 \to \cdots \to \rho_k \to co(\rho)$ where $co(\rho)$ is a prime type. Given such a decomposition of ρ we refer to $co(\rho)$ as the *co-domain* of ρ, to k as the *arity* of ρ, and to ρ_i $(1 \leq i \leq k)$ as the *i-th domain* of ρ.

We now define the *order* of a type which extends the usual definition of order for the simple types. Motivated by later technicalities, it is convenient to assign order -1 to the type of substitutions, and order 0 to any function type with co-domain ϵ.

Definition 3.2 (Order)**.** The *order* of a type ρ, $ord(\rho)$, is defined as follows.

$$ord(\iota) = ord(\epsilon) = 0 \qquad ord(\varsigma) = -1 \qquad ord(\rho \times \sigma) = \max\{ord(\rho), ord(\tau)\}$$

$$ord(\rho \to \sigma) = \begin{cases} 0, & \text{if } co(\sigma) = \epsilon, \\ \max\{ord(\rho) + 1, ord(\sigma)\}, & \text{otherwise.} \end{cases}$$

Definition 3.3 (Ranked alphabet)**.** A (ranked) *alphabet* is a pair $\mathcal{A} = \langle S, \lambda \rangle$ where S is a set, called the *carrier* of $\mathcal{A}$, and $\lambda \colon S \to \text{Type}$ is a type assignment for elements of S. If $\lambda(\mathsf{S})$ is a simple (basic) type for every $\mathsf{S} \in S$ we call $\mathcal{A}$ *simple* (resp. *basic*). Two ranked alphabets are *disjoint* just in case their carriers are disjoint sets.

Given an alphabet $\mathcal{A} = \langle S, \lambda \rangle$, we write $\alpha^\rho \in \mathcal{A}$ if $\alpha \in S$ and $\lambda(\alpha) = \rho$, and hence frequently identify $\mathcal{A}$ with the set $\{\alpha^{\lambda(\alpha)} \mid \alpha \in S\}$ of symbols with type annotations. For alphabets $\mathcal{A} = \langle S, \lambda \rangle$ and $\mathcal{B} = \langle S', \lambda' \rangle$, we write $\mathcal{A} \subset \mathcal{B}$ if $S \subseteq S'$ and $\lambda = \lambda' \restriction S$. In case $\mathcal{A}$ and $\mathcal{B}$ are disjoint, $\mathcal{A} \cup \mathcal{B}$ denotes the alphabet formed by the union of $\mathcal{A}$ and $\mathcal{B}$, namely $\langle S \cup S', \lambda \cup \lambda' \rangle$. The empty alphabet is denoted $\emptyset$.

Definition 3.4 (Terms and substitutions)**.** Fix alphabets $\Sigma \subset \mathcal{A}$ where Σ is simple. The $\mathcal{A}$*-terms over* Σ (henceforth $\mathcal{A}$*-terms*) and the types they inhabit are defined inductively as follows, where $r : \rho$ expresses that r is an $\mathcal{A}$-term of type ρ.

1. $\langle\rangle$ is an $\mathcal{A}$-term of type ϵ.

2. If $\alpha^\rho \in \mathcal{A}$ then α is an $\mathcal{A}$-term of type ρ.

3. If $r : \rho$ and $s : \sigma$ then $\langle r, s \rangle$ is a $\mathcal{A}$-term of type $\rho \times \sigma$.

4. If $r : \sigma \to \tau$ and $s : \sigma$ then rs is a $\mathcal{A}$-term of type τ.

5. $\perp$ is an $\mathcal{A}$-term of type ς.

6. If $a : \varsigma$ and $r : \rho$, and $\alpha^\rho \in \Sigma$ then $[\alpha \leftarrowtail r]a$ is an $\mathcal{A}$-term of type ς.

7. If $r : \rho$ and $a : \varsigma$ and $\rho \neq \varsigma$ then $r \cdot a$ is an $\mathcal{A}$-term of type ρ.

Note that λ-abstraction is not present in the term calculus, so the existence of terms of function type ρ depends on the presence of $\mathcal{A}$-symbols with type $\rho_1 \to \cdots \to \rho_k \to \rho$.

In addition to the abbreviation $r : \rho$ used above, we occasionally write r^ρ to express that r is an $\mathcal{A}$-term of type ρ. We often drop mention of Σ and $\mathcal{A}$ if they can be inferred from the context or are not important to the given setting, in which case $\mathcal{A}$-terms are referred to simply as *terms*. Terms arising from 1, 2 and 5 are called *constants*; terms arising from cases 3 and 4 are called *pairs* and *applications* respectively; terms of type ς are called *substitution stacks*; and terms of the form in 7 are called *explicit substitutions* (or simply *substitutions* if there is no cause for confusion). A *basic term* is any term constructed via the rules 1 to 4 only, i.e. a $\mathcal{B}$-term over $\emptyset$ for some basic alphabet $\mathcal{B}$. A term of sequence type is called a *sequence*. Application is assumed to associate to the left, and pairing and the formation rule for substitution stacks both associate to the right.

The sub-term relation is defined as usual over the basic terms, and is extended to terms containing substitutions by defining the sub-terms of $\perp$ to be $\{\perp\}$, the sub-terms of $a = [\alpha \mapsfrom r]b$ to be a and any sub-term of r or b, and the sub-terms of $r = s \cdot a$ to be r and any sub-term of s or a. Thus the basic terms are precisely those terms that do not have a substitution stack as a sub-term.

Given a finite sequence of terms $(r_i : \rho_i)_{i \leq k}$, let $\langle r_0, r_1, \dots, r_k \rangle$ be the term r_0 if $k = 0$ and, otherwise, the pair $\langle r_0, \langle r_1, \dots \langle r_{k-1}, r_k \rangle \cdots \rangle \rangle$ of type $\rho_0 \times \rho_1 \times \cdots \times \rho_k$. The *order* of a term is the order of its type.

Proposition 3.5. *If $\Sigma \subset \Sigma'$ are simple alphabets and $\mathcal{A}$ is an alphabet extending Σ', then every $\mathcal{A}$-term over Σ is an $\mathcal{A}$-term over Σ'.*

In addition to the term-level explicit substitutions, there is of course the usual operation of substituting given symbols by terms of corresponding type which we refer to as *implicit substitution.* Explicit substitutions can be interpreted as implicit substitutions by reading terms $r \cdot a$ as the image of r under the (implicit) substitution described by a, a process we call *evaluation.* The following definitions explicate these two operations. Fix an alphabet $\mathcal{A}$.

Definition 3.6 (Implicit substitution)**.** For $\mathcal{A}$-terms $r : \rho$, $t_0 : \tau_0$, $\dots$, $t_k : \tau_k$ and distinct symbols $\alpha_0^{\tau_0}, \dots, \alpha_k^{\tau_k} \in \mathcal{A}$, the term $r(\vec{t}/\vec{\alpha})$ is the $\mathcal{A}$-term given by simultaneously replacing every occurrence of α_i (for $i \leq k$) in r by t_i, defined recursively by:

$$\beta(\vec{t}/\vec{\alpha}) = \begin{cases} \beta, & \text{if } \beta \in \mathcal{A} \text{ and } \beta \notin \{\alpha_i \mid i \leq k\}, \\ t_i, & \text{if } \beta = \alpha_i, \end{cases}$$

$$\begin{aligned} \langle\rangle(\vec{t}/\vec{\alpha}) &= \langle\rangle & (rs)(\vec{t}/\vec{\alpha}) &= r(\vec{t}/\vec{\alpha})s(\vec{t}/\vec{\alpha}) \\ \bot(\vec{t}/\vec{\alpha}) &= \bot & \langle r, s\rangle(\vec{t}/\vec{\alpha}) &= \langle r(\vec{t}/\vec{\alpha}), s(\vec{t}/\vec{\alpha})\rangle \\ ([\beta \mapsfrom s]a)(\vec{t}/\vec{\alpha}) &= [\beta \mapsfrom s(\vec{t}/\vec{\alpha})](a(\vec{t}/\vec{\alpha})) & (r \cdot a)(\vec{t}/\vec{\alpha}) &= r(\vec{t}/\vec{\alpha}) \cdot (a(\vec{t}/\vec{\alpha})) \end{aligned}$$

If the choice of $\vec{\alpha}$ can be inferred from context, we write $r(\vec{t})$ in place of $r(\vec{t}/\vec{\alpha})$.

Definition 3.7 (Evaluating substitutions)**.** Given an $\mathcal{A}$-term r and a substitution stack $a = [\alpha_1 \mapsfrom s_1] \cdots [\alpha_k \mapsfrom s_k]\bot : \varsigma$ over some simple alphabet $\Sigma \subset \mathcal{A}$, the *evaluation of r relative to a* is the $\mathcal{A}$-term over Σ given by

$$r^a := r(s_1/\alpha_1) \cdots (s_k/\alpha_k).$$

The *evaluation* of r is the term r° given by recursively evaluating relative to each substitution in r, namely evaluation leaves basic terms unchanged, commutes with application and pairing, is defined by $\bot^\circ = \bot$ and $([\alpha \mapsfrom r]a)^\circ = [\alpha \mapsfrom r^\circ]a^\circ$ on substitution stacks, and by $(r \cdot a)^\circ = (r^\circ)^{a^\circ}$ for explicit substitutions.

Note that the evaluation of a substitution stack on a term is well-defined due to the typing constraints on their formation.

An alphabet generally specifies a set of symbols which are associated certain re-write rules in a recursion scheme. In this context, an explicit substitution acts as a delayed substitution which is not evaluated until no further re-writes to sub-terms are possible. For instance, over the alphabet $\{\mathsf{F}^{\iota\to\iota}, \mathsf{G}^{\iota}, \mathsf{e}^{\iota}, \circ^{\iota\to\iota\to\iota}, \alpha^{\iota}\}$ with associated re-write rules

$\mathsf{F}x \to (\mathsf{G} \circ x) \cdot [\alpha \hookleftarrow \mathsf{e}]\bot$ (for any instantiation of x) and $\mathsf{G} \to \alpha$, a derivation starting from the term $t = \mathsf{F}\alpha$ is

$$\begin{aligned} t &\to (\mathsf{G} \circ \alpha) \cdot ([\alpha \hookleftarrow \mathsf{e}]\bot) \\ &\to (\alpha \circ \alpha) \cdot ([\alpha \hookleftarrow \mathsf{e}]\bot). \end{aligned}$$

The final term evaluates to $\mathsf{e} \circ \mathsf{e}$. Attempting to read the explicit substitution implicitly leads also to the derivation

$$\begin{aligned} t &\to (\mathsf{G} \circ \alpha) \cdot ([\alpha \hookleftarrow \mathsf{e}]\bot) \\ &= \mathsf{G} \circ \mathsf{e} \\ &\to \alpha \circ \mathsf{e}. \end{aligned}$$

Lemma 3.8. *If $r : \rho$ is a Σ-term for some simple alphabet Σ and ρ is a basic type then r° is a basic Σ-term of type ρ.*

Proof. All substitution stacks that may occur in a term of basic type built from an alphabet of simply-typed symbols must be within the context of an explicit substitution. As evaluation replaces every explicit substitution by an implicit one, the result is a basic term of the same type. □

Lemma 3.9. *If r and $a = [\alpha \hookleftarrow s]b : \varsigma$ are $\mathcal{A}$-terms such that α does not occur in r, then $r^a = r^b$.*

Definition 3.10 (Σ-length)**.** Given alphabets $\Sigma \subset \mathcal{A}$ and an $\mathcal{A}$-term s, the *Σ-length* of s, written $|s|_\Sigma$, is the number of occurrences of symbols in s that are not Σ-terms, formally: $|\alpha|_\Sigma = 0$ if $\alpha \in \Sigma \cup \{\langle\rangle\}$; $|\alpha|_\Sigma = 1$ if $\alpha \in \mathcal{A}$ and $\alpha \notin \Sigma$; and $|r|_\Sigma = |s|_\Sigma + |t|_\Sigma$ if $r \in \{\langle s, t\rangle, st, s \cdot t, [\alpha \hookleftarrow s]t\}$.

In particular, the Σ-length of a Σ-term is 0 and if Σ is the empty alphabet then Σ-length of any term is the number of leaves in the tree representation of the term not labelled by $\langle\rangle$.

Remark 3.11 (Notational conventions)**.** Symbols ρ, σ and τ (also with indices) range over types. We commonly notate alphabets by upper-case Roman symbols in calligraphic typeface: $\mathcal{A}$, $\mathcal{B}$, etc, though Greek symbols Σ and Σ' will be used for simple alphabets. Sans-serif typeface (f, F, s, S, etc.) and lowercase Greek symbols α, β, etc. range over elements of ranked alphabets, with the latter particularly used for constants of simple type. Italicised letters r, s, t, R, S, etc. range over terms and a, b over substitution stacks, i.e. terms of type ς.

3.2 Higher Order Recursion Schemes

Higher order recursion schemes provide a handy framework for extending the correspondence between formal grammars and Herbrand sets established in [1, 24]. Their advantage over formal grammars can be appreciated by the fact that they allow us to dispense with the rigidity conditions (equality constraints) that were necessary in previous approaches.

Definition 3.12 (Higher order recursion scheme)**.** A *(non-deterministic) higher order recursion scheme*, or simply *recursion scheme*, is a tuple $\mathscr{R} = \langle \Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P} \rangle$ where Σ is a simple alphabet, $\mathcal{N}$ is a alphabet of *non-terminals* disjoint from Σ, $\mathcal{S} \subseteq \mathcal{N}$ is a designated finite set of *starting symbols* of sequence type, $\mathcal{P}$ is a set of pairs (F^ρ, t), called *production rules*, such that $\mathsf{F}^\rho \in \mathcal{N}$ and $t : co(\rho)$ is a $(\Sigma \cup \mathcal{N} \cup \{x_1^{\rho_1}, \dots, x_k^{\rho_k}\})$-term over Σ where x_i is a fresh symbol not in $\mathcal{N}$ and ρ_i is the i-th domain of ρ. A production rule (F^ρ, t) where the arity of ρ is k is written as

$$\mathsf{F}x_1 \cdots x_k \to_{\mathscr{R}} t,$$

or $\mathsf{F}\vec{x} \to_{\mathscr{R}} t$. Notice that by definition the term $\mathsf{F}x_1 \cdots x_k$ is of type $co(\rho)$.

A non-terminal $\mathsf{F} \in \mathcal{N}$ of $\mathscr{R}$ is *determined* if there is a unique production rule (F^ρ, t) in $\mathcal{P}$. By an *$\mathscr{R}$-term* we mean a $(\Sigma \cup \mathcal{N})$-term over Σ. The *order* of $\mathscr{R}$ is the supremum over orders of the types of non-terminals of $\mathscr{R}$.

Notice that we do not require that $\mathscr{R}$ contains only finitely many non-terminals, nor that the set of start symbols is non-empty. This is for technical convenience as it allows us to consider the recursion schemes of the next section as finitely generated 'sub-schemes' of a single infinite recursion scheme. Moreover, higher order recursion schemes are traditionally presented in the context of simple types, wherein start symbols are all of type ι (and indeed a single start symbol suffices) and production rules have the form $\mathsf{F}\vec{x} \to t$ with $t : \iota$. We consider the above definition to be the natural extension of recursion schemes to accommodate non-trivial prime types.

A given non-terminal may be assigned multiple production rules, leading to non-determinism. To simplify presentation of production rules in this case we adopt the convention of writing

$$\mathsf{F}\vec{x} \to_{\mathscr{R}} t_0 \mid \cdots \mid t_k$$

to express that $\mathscr{R}$ contains exactly the production rule $\mathsf{F}\vec{x} \to t_i$ for each $i \leq k$, i.e. $\mathsf{F}\vec{x} \to_{\mathscr{R}} t_i$ for each $i \leq k$ and if $\mathsf{F}\vec{x} \to_{\mathscr{R}} t$ then $t = t_i$ for some $i \leq k$.

Definition 3.13 (Derivations and language)**.** Let $\mathscr{R} = \langle \Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P} \rangle$ be a higher order recursion scheme. We extend the relation $\to_{\mathscr{R}}$ to a relation on $\mathscr{R}$-terms defined by setting $r \to_{\mathscr{R}} s$ if either

- $r = \mathsf{F}r_1 \cdots r_k$ for some $\mathsf{F}^\rho \in \mathcal{N}$ with arity k and there exists a production rule $\mathsf{F}\vec{x} \to_{\mathscr{R}} t$ such that $s = t(\vec{r}/\vec{x})$;
- $r = t(r_0/x)$, $s = t(s_0/x)$ and $r_0 \to_{\mathscr{R}} s_0$.

A *derivation* of s from r is a sequence $r = r_0 \to_{\mathscr{R}} \cdots \to_{\mathscr{R}} r_k = s$, the *length* of which is k. We say s is *derivable from* r in $\mathscr{R}$, in symbols $r \to_{\mathscr{R}}^* s$ (or $r \to^* s$ if $\mathscr{R}$ is clear from the context), if there exists a derivation of s from r, and s is *derivable in* $\mathscr{R}$ if s is derivable from some $\mathsf{S} \in \mathcal{S}$. The *language* of $\mathscr{R}$, written $L(\mathscr{R})$, is the set of pairs (S, t) such that $\mathsf{S} \in \mathcal{S}$, t is a basic Σ-term and $\mathsf{S} \to_{\mathscr{R}}^* t$.

Definition 3.14. Let $\mathscr{R} = \langle \Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P} \rangle$ be a higher order recursion scheme. $\mathscr{R}$ is *finite* if $\mathcal{N}$ and $\mathcal{P}$ are both finite sets, and is *acyclic* if there exists a transitive, irreflexive relation $<$ on $\mathcal{N}$ such that for every production rule $\mathsf{F}\vec{x} \to_{\mathscr{R}} t$ and every non-terminal G occurring in t, $\mathsf{G} < \mathsf{F}$.

Lemma 3.15. *A finite acyclic recursion scheme induces a finite language.*

An upper bound on the size of the language of acyclic recursion schemes can be obtained by reducing the problem to the length of reduction sequences for the simply-typed λ-calculus. Bounds on normalisation in the simply-typed λ-calculus have been given by Schwichtenberg [39] and improved to exact bounds by Beckmann [10]. In the following we use Beckmann's result to obtain concrete bounds for acyclic recursion schemes. Let $2^n_0 = n$ and $2^n_{k+1} = 2^{2^n_k}$ and extend the length function of the previous section to include λ-abstractions by setting $|\lambda xs|_\Sigma = |s|_\Sigma + 1$.

Theorem 3.16 (Beckmann [10])**.** *Let t be a term in the simply-typed λ-calculus over a simple alphabet Σ. The length of any β-reduction sequence starting from t is bounded by $2^{|t|_\Sigma}_{d(t)}$ where $d(t)$ denotes the maximum among orders of sub-terms of t.*

Beckmann's bound still applies if t is an arbitrary λ-term over the calculus of Σ-terms given in Definition 3.4 subject to the restriction that $\langle\rangle$ is the only sub-term of t of type ϵ (a necessary restriction due to our non-standard definition of order). Non-deterministic reductions can also be incorporated via a fresh operator $|$ and permitting β-reductions of the form $(\lambda x.\, t_0|\cdots|t_k)s \to_\beta t_i(s/x)$ for each $i \leq k$. In this case the length and the function d is given by $|s|t|_\Sigma = \max\{|s|_\Sigma, |t|_\Sigma\}$ and $d(s|t) = \max\{d(s), d(t)\}$. Finally, we wish to allow for so-called η-long reductions, i.e., reductions $(\lambda x_0 \cdots x_k.\, s)t_0 \cdots t_k \to_\beta s(\vec{t}/\vec{x})$ where s is not an abstraction. Provided that only η-long reductions are permitted and each counts as one step in a β-reduction sequence, Beckmann's bound holds with the analogous change to the length function: $|\lambda\vec{x}s|_\Sigma = |s|_\Sigma + 1$ if s not an abstraction.

From these observations we may deduce the following result. We restrict ourselves to recursion schemes built without the substitution stacks as this will suffice for our later use.

Theorem 3.17. *Let $\mathscr{R} = \langle\Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P}\rangle$ be a finite acyclic order n recursion scheme such that every non-terminal has basic type, and for every production rule $\mathsf{F}\vec{x} \to t$ in $\mathscr{R}$, $|t|_\Sigma < k$. The length of every derivation in $\mathscr{R}$ is bounded by $2^{|\mathcal{N}|(k+1)}_{n+1}$.*

Proof. Let $\mathscr{R} = \langle\Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P}\rangle$ be an order n recursion scheme fulfilling the requirements in the statement. Without loss of generality we may assume that $\mathcal{S}$ is a singleton, that every non-terminal is associated at least one production rule, and that $\mathcal{X}$ is the alphabet of variable symbols disjoint from both Σ and $\mathcal{N}$ such that every term occurring in a production rule in $\mathscr{R}$ is an $(\Sigma \cup \mathcal{N} \cup \mathcal{X})$-term.

Fix an enumeration $\mathsf{F}^{\rho_N}_N < \cdots < \mathsf{F}^{\rho_2}_2 < \mathsf{F}^{\rho_1}_1$ of the non-terminals of $\mathscr{R}$ according to a total ordering $(<)$ witnessing acyclicity of $\mathscr{R}$. We may assume $\mathcal{S} = \{\mathsf{F}_1\}$, so ρ_1 is prime. Let $\mathcal{Y} = \{y^{\rho_1}_1, y^{\rho_2}_2, \ldots, y^{\rho_N}_N\}$ be a set of fresh (and pairwise distinct) variable symbols of marked type. We define by recursion a sequence $s_1, \ldots, s_N$ of well-typed λ-terms all of type ρ_1 such that s_i contains only the variables $y_{i+1}, \ldots, y_N$ free and the length of every derivation from F_1 which only re-writes non-terminals F_j for $j \leq i$ is bounded by the length of the longest β-reduction sequence starting from s_i. Suppose $\{\mathsf{F}_i\vec{x} \to T_j : j \leq m\}$ is the set of production rules associated to F_i in $\mathscr{R}$. For each $j \leq m$, let $t_j = T_j(y_{i+1}, \ldots, y_N/\mathsf{F}_{i+1}, \ldots, \mathsf{F}_N)$ be the $(\Sigma \cup \mathcal{X} \cup \mathcal{Y})$-term resulting from T_j by substituting the non-terminals $\mathsf{F}_{i+1}, \ldots, \mathsf{F}_N$ by variables $y_{i+1}, \ldots, y_N$ respectively. It follows that $|t_j|_\Sigma \leq |T_j|_\Sigma < k$. Finally, define $s_i = \lambda\vec{x}.\, t_0|\cdots|t_m$ if $i = 1$, and

$s_i = (\lambda y_i.\, s_{i-1})(\lambda \vec{x}.\, t_0| \cdots |t_m)$ otherwise, and notice that $|s_N|_\Sigma \leq N(k+1)$ and the maximal order among sub-terms of s_N is no greater than $n+1$. Every $\mathscr{R}$-derivation from F_1 can be replicated as a sequence of one-step η-long β-reductions starting from s_N, the length of which, by Beckmann's bound, is no greater than $2^{N(k+1)}_{n+1}$. □

As a corollary we obtain we obtain bounds on the size of languages.

Corollary 3.18. *Let $\mathscr{R}$ and k be as in the previous theorem and suppose every non-terminal in $\mathscr{R}$ is associated at most two production rules. Then the size of $L(\mathscr{R})$ is bounded by $2^{|\mathcal{N}|(k+1)}_{n+2}$.*

Proof. Given a recursion scheme $\mathscr{R}$ all terms in $L(\mathscr{R})$ can be derived via the leftmost reduction strategy. By the previous theorem, the length of these derivations is bounded by $2^{|\mathcal{N}|(k+1)}_{n+1}$, leading to a bound of $2^{|\mathcal{N}|(k+1)}_{n+2}$ on the size of $L(\mathscr{R})$. □

The bound given in Corollary 3.18 is optimal in the parameter n as the next lemma demonstrates.

Lemma 3.19. *Let Σ be the ranked alphabet $\{\mathsf{a}^\iota, \mathsf{b}^\iota, \mathsf{d}^{\iota\to\iota\to\iota\to\iota}\}$. There exists a sequence of acyclic higher order recursion schemes $\mathscr{R}_n = \langle \Sigma, \mathcal{N}_n, \mathcal{S}_n, \mathcal{P}_n \rangle$ such that*

1. *the order of $\mathscr{R}_n$ is n,*
2. *$|\mathcal{N}_n|, |\mathcal{P}_n| = O(n)$,*
3. *$\max\{|t|_\Sigma : \mathsf{F}\vec{x} \to_{\mathscr{R}_n} t\} = O(n)$,*
4. *$|L(\mathscr{R}_n)| \geq 2^1_{n+2}$.*

Proof. It suffices to translate Beckmann's lower bounds from [10] to the context of recursion schemes. Define $\tau_0 = \iota$ and $\tau_{i+1} = \tau_i \to \tau_i$ for each $i < \omega$. So τ_i has order and arity i for each i. Fix $n > 0$. The recursion scheme $\mathscr{R}_n$ comprises a single start symbol $\mathsf{S}_n : \iota$ and a non-terminal $\mathsf{F}_i : \tau_i$ for each $i \leq n$. The production rules are

$$\begin{array}{cc}
\mathsf{F}_0 \to \mathsf{a} \mid \mathsf{b} & \mathsf{S}_n \to \mathsf{F}_n(\mathsf{F}_n\mathsf{F}_{n-1})\mathsf{F}_{n-2}\ldots\mathsf{F}_1\mathsf{F}_0 \\
\mathsf{F}_1 x_0 \to \mathsf{d}\mathsf{F}_0 x_0 x_0 & \mathsf{F}_{i+2} x_0 x_1 \cdots x_{i+1} \to x_0(x_0 x_1) x_2 \cdots x_{i+1}
\end{array}$$

Requirements 1–3 are clearly satisfied. To deduce 4, observe that applying deterministic production rules only, $\mathsf{S}_n \to^* \mathsf{F}_1^{(2^1_n)}\mathsf{F}_0$, where $\mathsf{X}^{(k)}$ denotes the k-fold iteration of X. Thus we see that $L(\mathscr{R}_n)$ is the set of complete binary trees of height $2^1_n + 1$ with each leaf and inner node labelled by either a or b, i.e. $|L(\mathscr{R}_n)| \geq 2^1_{n+2}$. □

3.3 Recursion Schemes with Pattern-Matching

To control the space of derivations we will utilise recursion schemes equipped with pattern-matching, introduced in [36]. In their full generality pattern-matching recursion schemes form a Turing complete model of computation, though we will only employ a subclass in which pattern-matching is restricted to decomposing sequences. The following definition presents the particular schemes we utilise.

Definition 3.20 (Pattern-matching recursion schemes)**.** A *pattern-matching recursion scheme* is a tuple $\mathscr{R} = \langle \Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P} \rangle$ where Σ, $\mathcal{N}$ and $\mathcal{S}$ are as in Definition 3.12 and $\mathcal{P}$ may include type-preserving production rules of the form

$$\mathsf{F} x_0 \cdots x_{k-1} \langle x_k, \dots, x_{k+l} \rangle \to_{\mathscr{R}} t$$

where t is a $\Sigma \cup \mathcal{N} \cup \{x_i \mid i \leq k+l\}$-term over Σ of prime type.

The associated reduction relation $r \to_{\mathscr{R}} s$ is defined by the two conditions in Definition 3.13 and an additional clause:

- $r = \mathsf{F} r_0 \cdots r_{k-1} \langle r_k, \dots, r_{k+l} \rangle$ for some $\mathsf{F} \in \mathcal{N}$ of arity k and terms $\vec{r} = (r_i)_{i \leq k+l}$, and there exists a production rule $\mathsf{F} x_0 \cdots x_{k-1} \langle x_k, \dots, x_{k+l} \rangle \to_{\mathscr{R}} t$ such that $s = t(\vec{r}/\vec{x})$.

The definition of a derivation and language for pattern-matching recursion schemes are analogous.

Remark 3.21. Pattern-matching recursion schemes can be simulated by higher order recursion schemes using constants representing projection functions for pairs in place of pattern-matching. In particular, the upper-bounds given by Theorem 3.17 and Corollary 3.18 apply to pattern-matching recursion schemes without change. There is, however, a subtle difference between the two in the presence of non-determinism and this will be exploited heavily in the next section. In the remainder of this paper *recursion scheme* refers to pattern-matching recursion schemes unless otherwise stated.

3.4 Herbrand Schemes

The recursion scheme associated to a proof π, which we call the *Herbrand scheme for* π and denote as $\mathscr{H}_\pi$, contains a non-terminal $\mathsf{N}^i_{\pi'}$ for each sub-proof $\pi' \vdash B_0, \dots, B_k$ of π and each $i \leq k$. The interpretation of such a non-terminal is a function which returns a witness (possibly containing explicit substitutions) for each weak quantifier in B_i given input for each strong quantifier in the sequent. The arity of $\mathsf{N}^i_{\pi'}$ is $k+2$, namely one greater than the length of the sequent: the first argument is a substitution stack and the $(j+1)$-th argument is the 'input' for the formula B_j. The type of $\mathsf{N}^i_{\pi'}$ depends only on the quantifier structure of the formulæ in the sequent. In particular, the types of $\mathsf{N}^i_{\pi'}$ and $\mathsf{N}^j_{\pi'}$ differ only in their co-domain. Furthermore, the reduction rules governing $\mathsf{N}^i_{\pi'}$ are determined by the final inference in π' and choice of i, and re-write $\mathsf{N}^i_{\pi'}$ to a term containing non-terminals for the immediate sub-proofs of π', so are independent of the particular starting proof. This property implies that the typing and re-write rules for a non-terminal $\mathsf{N}^i_{\pi'}$ are invariant across all Herbrand schemes $\mathscr{H}_\pi$ for which π' is a sub-proof of π, whence we may consider two Herbrand schemes as comprising identical sets of non-terminals and production rules and differing only in the selection of start symbols.

We begin by introducing the types that occur most prominently in Herbrand schemes. To each prenex formula F we assign two types, the *output* type, τ_F, and the *input* type, τ^*_F, representing the 'existential' and 'universal' structure of F respectively. These types are determined by the quantifier structure of F and are defined as follows. If F is quantifier-free, $\tau_F = \tau^*_F = \epsilon$; otherwise,

$$\tau_{\forall v F} = \tau_F, \quad \tau_{\exists v F} = \begin{cases} \iota \times \tau_F, & \text{if } u(F) = 0, \\ \iota \times (\tau_{\bar{F}} \to \tau_F), & \text{if } u(F) > 0, \end{cases} \quad \tau^*_F = \begin{cases} \tau_{\bar{F}}, & \text{if } e(F) = 0, \\ \tau_F \to \tau_{\bar{F}}, & \text{if } e(F) > 0. \end{cases}$$

Lemma 3.22. *Let F be a prenex formula and $\vec{v} = (v_i)_{i<k}$. Then*

1. τ_F *is a non-simple basic prime type.*
2. $\tau_F = \tau_{F(\vec{r}/\vec{\alpha})}$ *and* $\tau^*_F = \tau^*_{F(\vec{r}/\vec{\alpha})}$.
3. *If* $e(F) > 0$ *then* $\tau^*_F = \tau^*_{\bar{F}} \to \tau_{\bar{F}}$.
4. $\tau_{\exists\vec{v}F} = \underbrace{\iota \times \cdots \times \iota}_{k} \times \tau^*_{\bar{F}}$ *and* $\tau^*_{\forall\vec{v}F} = \underbrace{\iota \times \cdots \times \iota}_{k} \times \tau^*_F$.

Proof. By definition and (for 4) induction. □

Example 3.23. We compute the input and output types for prenex Π_2 and Σ_2 formulæ. Let $B = \exists\vec{w}A_{qf}$ and $C = \forall\vec{v}B$ where $\vec{w}$ and $\vec{v}$ have non-zero length m and n respectively.

$$\begin{array}{llll}
\tau_B = \iota^m & \tau_{\bar{B}} = \epsilon & \tau^*_B = \iota^m \to \epsilon & \tau^*_{\bar{B}} = \iota^m \\
\tau_C = \iota^m & \tau_{\bar{C}} = \underbrace{\iota \times \cdots \times \iota}_{n} \times (\iota^m \to \epsilon) & \tau^*_C = \tau_{\bar{C}} & \tau^*_{\bar{C}} = \tau_{\bar{C}} \to \tau_C
\end{array}$$

Definition 3.24 (Herbrand scheme)**.** Fix a proof $\pi \vdash A_0, \ldots, A_k$ with Σ_1 end-sequent and let Σ_π be the simple alphabet consisting of a constant symbol c of type ι and the function symbols and eigenvariables occurring in π (typed accordingly). The *Herbrand scheme for* π is the higher order recursion scheme $\mathscr{H}_\pi = \langle \Sigma_\pi, \mathcal{N}_\pi, \mathcal{S}_\pi, \mathcal{P}_\pi \rangle$ with the following non-terminals and production rules.

1. A non-terminal $\mathsf{c}_\rho : \rho$ for each basic type $\rho \notin \{\iota, \epsilon\}$ that occurs as a sub-type of a type τ_B or τ^*_B for a formula B occurring in π, with production rules
$$\begin{array}{ll}
\mathsf{c}_\rho \to \langle \mathsf{c}_{\tau_0}, \mathsf{c}_{\tau_1} \rangle & \text{if } \rho = \tau_0 \times \tau_1, \\
\mathsf{c}_\rho x_0^{\rho_0} \cdots x_k^{\rho_k} \to \mathsf{c}_{co(\rho)} & \text{if } \rho = \rho_0 \to \cdots \to \rho_k \to co(\rho),
\end{array}$$
with c_ι and c_ϵ defined to be the constants c and $\langle\rangle$ respectively.

2. A non-terminal $\mathsf{N}^i_{\pi'}$ for each sub-proof $\pi' \vdash B_0, \ldots, B_l$ of π and for each $i \leq l$, with type
$$\mathsf{N}^i_{\pi'} : \varsigma \to \tau^*_{B_0} \to \cdots \to \tau^*_{B_l} \to \tau_{B_i}$$
and production rule(s) as given in Table 1, determined in each case by the final inference of π'.

3. A start symbol $\mathsf{S}_{\pi,i} : \tau_{A_i}$ for each $i \leq k$ with associated production rules
$$\mathsf{S}_{\pi,i} \to \mathsf{N}^i_\pi \bot \mathsf{c}_{\tau^*_{A_0}} \cdots \mathsf{c}_{\tau^*_{A_k}}$$

The *language of* π is the set $L(\pi) = \{(i, r^\circ) \mid i \leq k \text{ and } (\mathsf{S}_{\pi,i}, r) \in L(\mathscr{H}_\pi)\}$.

Inference deriving π	Corresponding production rule(s)
$\mathsf{ax}\colon \pi \vdash A, \bar{A}$	$\mathsf{N}^i_\pi a x_0 x_1 \to \langle\rangle$
$\vee\,\dfrac{\pi_0 \vdash \Gamma, A, B}{\pi \vdash \Gamma, A \vee B}$	$\mathsf{N}^i_\pi a\vec{x}z \to \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}zz, & \text{if } i < m,\\ \langle\rangle, & \text{otherwise.}\end{cases}$
$\wedge\,\dfrac{\pi_0 \vdash \Gamma, A \quad \pi_1 \vdash \Delta, B}{\pi \vdash \Gamma, \Delta, A \wedge B}$	$\mathsf{N}^i_\pi a\vec{x}\vec{y}z \to \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}z, & \text{if } i < m,\\ \mathsf{N}^{i-m}_{\pi_1} a\vec{y}z, & m \le i < m+n,\\ \langle\rangle, & \text{otherwise.}\end{cases}$
$\forall_{\vec{\alpha}}\,\dfrac{\pi_0 \vdash \Gamma, A(\vec{\alpha}/\vec{v})}{\pi \vdash \Gamma, \forall\vec{v}A}$	$\mathsf{N}^i_\pi a\vec{x}\langle z_0, \dots, z_{p+1}\rangle \to \mathsf{N}^i_{\pi_0}([\alpha_p \mapsfrom z_p]\cdots[\alpha_0 \mapsfrom z_0]a)\vec{x}z_{p+1}$
$\exists_{\vec{r}}\,\dfrac{\pi_0 \vdash \Gamma, A(\vec{r}/\vec{v})}{\pi \vdash \Gamma, \exists\vec{v}A}$	$\mathsf{N}^i_\pi a\vec{x}z \to \begin{cases} \vec{r}\cdot a \star (\mathsf{N}^m_{\pi_0} a\vec{x}), & \text{if } i = m \text{ and } u(A) > 0,\\ \vec{r}\cdot a \star \langle\rangle, & \text{if } i = m \text{ and } u(A) = 0,\\ \mathsf{N}^i_{\pi_0} a\vec{x}(z(\mathsf{N}^m_\pi a\vec{x}z)) & \text{if } i \neq m,\end{cases}$
$\mathsf{cut}\,\dfrac{\pi_0 \vdash \Gamma, A \quad \pi_1 \vdash \Delta, \bar{A}}{\pi \vdash \Gamma, \Delta}$	$\mathsf{N}^i_\pi a\vec{x}\vec{y} \to \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}((\mathsf{N}^n_{\pi_1} a\vec{y}) \circ_A (\mathsf{N}^m_{\pi_0} a\vec{x})), & \text{if } i < m,\\ \mathsf{N}^{i-m}_{\pi_1} a\vec{y}((\mathsf{N}^m_{\pi_0} a\vec{x}) \circ_{\bar{A}} (\mathsf{N}^n_{\pi_1} a\vec{y})), & \text{if } m \le i.\end{cases}$
$\mathsf{w}\,\dfrac{\pi_0 \vdash \Gamma}{\pi \vdash \Gamma, A}$	$\mathsf{N}^i_\pi a\vec{x}z \to \begin{cases} \mathsf{c}_{\tau_A}, & \text{if } i = m,\\ \mathsf{N}^i_{\pi_0} a\vec{x}, & \text{otherwise.}\end{cases}$
$\mathsf{c}\,\dfrac{\pi_0 \vdash \Gamma, A, A}{\pi \vdash \Gamma, A}$	$\mathsf{N}^i_\pi a\vec{x}z \to \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}zz, & \text{if } i < m,\\ \mathsf{N}^i_{\pi_0} a\vec{x}zz \mid \mathsf{N}^{i+1}_{\pi_0} a\vec{x}zz, & \text{if } i = m.\end{cases}$
$\mathsf{p}\,\dfrac{\pi_0 \vdash \Gamma, B, A, \Delta}{\pi \vdash \Gamma, A, B, \Delta}$	$\mathsf{N}^i_\pi a\vec{x}z_0z_1\vec{y} \to \begin{cases} \mathsf{N}^{i+1}_{\pi_0} a\vec{x}z_1z_0\vec{y}, & \text{if } i = m,\\ \mathsf{N}^{i-1}_{\pi_0} a\vec{x}z_1z_0\vec{y}, & \text{if } i = m+1,\\ \mathsf{N}^i_{\pi_0} a\vec{x}z_1z_0\vec{y}, & \text{otherwise.}\end{cases}$

$$\begin{aligned} \vec{\alpha} &= (\alpha_0, \dots, \alpha_p)\\ \vec{r} &= (r_0, \dots, r_p)\\ \vec{r}\cdot a &= (r_0 \cdot a, \dots, r_p \cdot a)\\ (u_j)_{j \le q} \star t &= \langle u_0, \dots, u_q, t\rangle \end{aligned} \qquad r \circ_A s = \begin{cases} r, & \text{if } e(A) > 0,\\ rs, & \text{if } u(A) > 0,\\ \langle\rangle, & \text{otherwise.}\end{cases}$$

Table 1: Production rules of $\mathscr{H}_\pi$. $\vec{x}$ and $\vec{y}$ are sequences of distinct variable symbols of length $m := |\Gamma|$ and $n := |\Delta|$ respectively.

It remains to check that the production rules of Herbrand schemes are well-typed. This task will be taken up later in Lemma 3.32. For now we take for granted the fact that Herbrand schemes are well-defined and continue with some basic properties of them (Lemmas 3.26 to 3.29) followed by the intended interpretation of the schemes as generating Herbrand disjunctions (Definition 3.30) and the observation that this interpretation coincides with the Herbrand set for quasi cut-free proofs (Lemma 3.31). We start, however, with a brief explanation of some of the production rules from Table 1.

Remark 3.25. We comment on some of the rules from Table 1.

- Axiom. We are restricting axioms to quantifier-free formulæ only, which motivates the simple production rule given in the table. One may wish to permit axioms $\pi \vdash A_0, A_1$ where $A_1 = \bar{A}_0$ has arbitrary (prenex) complexity. These can be accommodated by the production rules

$$\mathsf{N}^i_\pi a x_0 x_1 \to \begin{cases} x_{1-i}, & \text{if } u(A_i) = 0, \\ x_{1-i} x_i, & \text{if } u(A_i) > 0, \end{cases}$$

  which the interested reader can check are well-typed. This definition mimics the behaviour of the Herbrand scheme for the natural proof of A_0, A_1 that uses only quantifier-free instances of axioms and alternate applications of $\exists$ and $\forall$ inferences. Our reason for favouring quantifier-free axioms is that, as a consequence, production rules never return their arguments as output, a fact that simplifies some technical aspects of the later analysis (specifically Lemma 5.9).

- $\wedge$ and $\vee$. As proofs involve prenex formulæ only, conjunctions and disjunctions are necessarily quantifier-free with associated type ϵ, and therefore possess no computational content relevant to the construction of a Herbrand disjunction. When focusing on such formulæ, the production rule in each case returns the empty sequence.

- $\exists_{\vec{r}}$. The production rule in this case depends on both i and the quantifier form of the active formula. Consider the instance of $\exists_{\vec{r}}$ given in Table 1. As π is assumed regular, the active formula (A in the table) is either quantifier-free or universally quantified. If i marks the active formula (i.e. $i = m$) then the production rule for N^i_π directly outputs the witness terms provided by the proof and the current substitution (the sequence $(r_0 \cdot a, \ldots, r_p \cdot a)$) as the first $p + 1$ components of a nested pair. The final component is either trivial (in case A is quantifier-free) or, if A is universally quantified, the continuation of the trace to the immediate sub-proof in the form of a function. If $i \neq m$, the production rule instead passes the above term to the corresponding argument.

- $\forall_{\vec{\alpha}}$. This is the only case that involves pattern matching in Herbrand schemes. Although it can be simulated by a recursion scheme without pattern-matching using projection functions for pair types, doing so introduces a duplication of arguments that is avoided in the chosen formulation. For instance, the production rule for $\forall_{\vec{\alpha}}$ where $\vec{\alpha}$ consists of the single eigenvariable α and the sequent Γ is empty yields the production rule

$$\mathsf{N}^0_\pi a \langle z_0, z_1 \rangle \to \mathsf{N}^0_{\pi_0}([\alpha \mapsfrom z_0]a) z_1$$

  which may be simulated by the rule

$$\mathsf{N}^0_\pi a z \to \mathsf{N}^0_{\pi_0}([\alpha \mapsfrom \mathsf{p}_0 z]a)(\mathsf{p}_1 z) \tag{1}$$

where p_0 and p_1 are constants representing the two projection functions for pair types. If s is a term such that $s \to^*_{\mathscr{H}} \langle r_0, s_0\rangle \mid \langle r_1, s_1\rangle$ and the four sub-terms are pairwise distinct then the reduction in (1) permits the derivation $\mathsf{N}^0_\pi \bot s \to^* \mathsf{N}^0_{\pi_0}([\alpha \mapsfrom \mathsf{p}_0\langle r_0, s_0\rangle]\bot)(\mathsf{p}_1\langle r_1, s_1\rangle)$, essentially the term $\mathsf{N}^0_{\pi_0}([\alpha \mapsfrom r_0]\bot)s_1$, which is forbidden in the Herbrand scheme due to pattern-matching. In this sense pattern matching plays a role analogous to the rigidity conditions utilised in [24, 1, 2] for representing first-order proofs with Π_1/Π_2 cut complexity.

- cut. For each choice of i, the rule provides exactly one reduction for the non-terminal N^i_π: for $i < m$ this is

$$\mathsf{N}^i_\pi a\vec{x}\vec{y} \to \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x})), & \text{if } u(A) > 0, \\ \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}), & \text{if } e(A) > 0, \\ \mathsf{N}^i_{\pi_0} a\vec{x}\langle\rangle, & \text{if } A \text{ is q.f.} \end{cases}$$

Note that, in the case $e(A) = 0$ the type τ^*_A (which marks the final argument to $\mathsf{N}^i_{\pi_0}$) is prime, and is otherwise the function type $\tau^*_{\bar{A}} \to \tau_{\bar{A}}$. Moreover, the case distinction above is independent of i. For instance, if $A = \forall v B$ exactly the following production rules arise from the cut.

$$\mathsf{N}^j_\pi a\vec{x}\vec{y} \to \begin{cases} \mathsf{N}^j_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x})), & \text{if } j < m, \\ \mathsf{N}^{m-j}_{\pi_1} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x}), & \text{if } m \le j < m + n. \end{cases}$$

In the following let $\mathscr{H} = \langle \Sigma, \mathcal{N}, \mathcal{S}, \mathcal{P}\rangle$ be the Herbrand scheme for a regular proof π with prenex Σ_1 end-sequent.

Lemma 3.26. *$\mathscr{H}$ is an acyclic recursion scheme. Hence, $L(\pi)$ is finite.*

Proof. Let $<$ be the transitive relation on non-terminals in $\mathscr{H}$ generated by the equations: $\mathsf{c}_\rho < \mathsf{c}_\sigma$ if ρ is a proper sub-type of σ; $\mathsf{c}_\rho < \mathsf{N}^i_{\pi_0}$ for every ρ, sub-proof π_0 of π and i; $\mathsf{N}^i_{\pi_0} < \mathsf{N}^j_{\pi_1}$ if either π_0 is a proper sub-proof of π_1 or $\pi_0 = \pi_1$ and $j < i$; and $\mathsf{N}^i_\pi < \mathsf{S}_{\pi,i}$ for any i. Clearly $<$ is acyclic and irreflexive. Moreover, for every production rule $\mathsf{F}\vec{x} \to_{\mathscr{H}} t$ and any non-terminal G occurring in t we have $\mathsf{G} < \mathsf{F}$. □

Lemma 3.27. *Every $\mathscr{H}$-term of simple type is a Σ-term, and every $\mathscr{H}$-term of substitution stack type has the form either $\bot$ or $[\alpha \mapsfrom s]b$ for some $\alpha \in \Sigma$, Σ-term s and $b : \varsigma$.*

Proof. The non-terminals of $\mathscr{H}$ all have type one of three forms: ϵ, pair type, or function type with non-simple co-domain. It therefore follows that the only $\mathscr{H}$-terms of simple type are the Σ-terms. Likewise, $\bot$ and $[\alpha \mapsfrom s]b$ are the only kind of $\mathscr{H}$-terms of type ς. Given a substitution stack $[\alpha \mapsfrom s]b$ however, as $\alpha \in \Sigma$ the first part of the lemma implies that s is a Σ-term. □

Lemma 3.28. *If $r : \sigma \to \tau$ is a $\mathscr{H}$-term then τ is a basic type and σ is either basic or the type of substitution stacks. In the latter case, $r = \mathsf{N}^i_\pi$ or $\hat{\mathsf{N}}^i_\pi$ for some π and i.*

Proof. By inspection of the types of non-terminals and terms. □

Lemma 3.29. *Suppose r is an $\mathscr{H}$-term of type ϵ containing no explicit substitutions (i.e. having no sub-term of the form $t \cdot a$). If $r \to^*_{\mathscr{H}} s$ for some basic term s then $s = \langle\rangle$.*

Proof. By induction on the proof generating $\mathscr{H}$, on the composition of r and the length of the derivation $r \to^* s$. □

We now describe how Herbrand schemes can be interpreted as ascribing existential content to first-order proofs.

Definition 3.30 (Herbrand expansion)**.** Let $\pi \vdash \Gamma$ be a proof with $\Gamma = \exists\vec{v}_0 A_0, \dots, \exists\vec{v}_k A_k$ where A_i is quantifier-free for each $i \leq k$. Let k_i be the length of $\vec{v}_i$. The *Herbrand expansion of* π is the quantifier free sequent Γ^π given by

$$\Gamma^\pi := \{A_i(\vec{r}_i/\vec{v}_i) \mid \vec{r}_i = (r_j)_{j<k_i} \text{ and } (i, \langle r_0, \dots, r_{k_i-1}, \langle\rangle\rangle) \in L(\pi)\}.$$

Lemma 3.31. *If $\pi \vdash \Gamma$ is a quasi cut-free proof of a Σ_1 end-sequent then the Herbrand expansion of π is a valid sequent and $\bigvee \Gamma^\pi$ is a Herbrand disjunction in the sense of Theorem 2.2.*

Proof. Observe that in every production rule associated to a quantifier-free cut, the term $r \circ_A s$ becomes $\langle\rangle$. Derivations in π are therefore in 1-1 correspondence with traces following the breakdown of formulæ in the end-sequent. As a result we observe that $L(\pi)$ simply outputs all literal witnesses to the existential quantifiers in the end-sequent. □

The idea behind Herbrand schemes is to provide a generalisation of the above lemma to proofs containing quantified cuts. The analysis necessary for the result is carried out in Section 5. In the remainder of this section we prove the production rules of Herbrand schemes are well-typed and derive upper bounds on the size of Herbrand expansions.

Lemma 3.32. *The production rules of Herbrand schemes are type preserving.*

Proof. Fix a proof π with prenex end-sequent $A_0, \dots, A_m$ and $i \leq m$. We establish type-preservation of the production rules for the non-terminals $\mathsf{N}^0_\pi, \dots, \mathsf{N}^m_\pi$ via a case distinction on the final inference rule in π.

Suppose $\pi \vdash A_0, \dots, A_{m-1}, \exists\vec{v}A$ is obtained from proof π_0 by $\exists_{\vec{r}}$. Thus $\pi_0 \vdash \Gamma, A(\vec{r}/\vec{v})$ for some sequence $\vec{r} = (r_j)_{j \leq k}$ of simple Σ-terms of type ι. By regularity, $e(A) = 0$, i.e. either A is quantifier-free or $u(A) > 0$. Let $\Gamma = A_0, \dots, A_{m-1}$ and fix a term $z : \tau^*_{\exists\vec{v}A}$ and a sequence of terms $\vec{x}$ of length m such that $\mathsf{N}^i_\pi \vec{x} z$ is well-typed. By definition $\mathsf{N}^i_{\pi_0}$ has type

$$\mathsf{N}^i_{\pi_0} : \varsigma \to \tau^*_{A_0} \to \cdots \to \tau^*_{A_{m-1}} \to \tau^*_A \to \begin{cases} \tau_A, & \text{if } i = m, \\ \tau_{A_i}, & \text{otherwise.} \end{cases}$$

To check type preservation there are two cases to consider:

1. $i = m$. If $u(A) = 0$ then A is quantifier-free and $\tau^*_{\bar{A}} = \epsilon$. If $u(A) > 0$ then $\tau^*_{\bar{A}} = \tau^*_A \to \tau_A$ by Lemma 3.22(3), so the type of $\mathsf{N}^i_{\pi_0} a\vec{x}$ is $\tau^*_{\bar{A}}$. Since also $\tau_{\exists\vec{v}A} = \underbrace{\iota \times \cdots \times \iota}_{k} \times \tau^*_{\bar{A}}$ by Lemma 3.22(4), we are done.

A	Σ_0	$\Sigma_n \setminus \Pi_n$	$\Pi_n \setminus \Sigma_n$
$ord(\tau_A)$	0	$n \dot{-} 2$	$n \dot{-} 3$
$ord(\tau_A^*)$	0	$n \dot{-} 1$	$n \dot{-} 2$

Table 2: Order of types τ_A and τ_A^*.

2. $i \neq m$. In this case it is necessary to check that $\tau^*_{\exists\vec{v}A} = \tau_{\exists\vec{v}A} \to \tau^*_A$. But this follows directly from the definition and the fact that $\tau_{\forall\vec{v}\bar{A}} = \tau_{\bar{A}} = \tau^*_A$ as $e(A) = 0$.

Suppose π is derived from π_0 via the inference $\forall_{\vec{\alpha}}$ and $A_m = \forall\vec{v}A$ with $u(A) = 0$ and $\vec{\alpha} = (\alpha_j)_{j<k}$. Let $i \leq m$ and fix terms $\vec{x}$, $\vec{z} = (z_0, \dots, z_k)$ such that $\mathsf{N}^i_\pi a\vec{x}\langle z_0, \dots, z_k\rangle$ is well-typed. Lemma 3.22 implies that $z_j : \iota$ for each $j < k$, and $z_k : \tau^*_A$. Thus $\mathsf{N}^i_{\pi_0} b\vec{x}z_k$ is well-typed and has type $\tau_A = \tau_{A_m}$.

Suppose π is derived via cut from sub-proofs $\pi_0 \vdash \Gamma, A$ and $\pi_1 \vdash \Delta, \bar{A}$. Let $m = |\Gamma|$ and $n = |\Delta|$ and fix $\vec{x}$ and $\vec{y}$ suitably typed. Without loss of generality we may assume $i < m$, in which case we require to show $(\mathsf{N}^n_{\pi_1} a\vec{y}) \circ_A (\mathsf{N}^m_{\pi_0} a\vec{x}) : \tau^*_A$ which reduces (via Remark 3.25) to proving

$$\begin{aligned} &e(A) > 0 \text{ implies } \tau^*_A = \tau^*_{\bar{A}} \to \tau_{\bar{A}}, \\ &u(A) > 0 \text{ implies } \tau^*_A = \tau_{\bar{A}} \text{ and } \tau^*_{\bar{A}} = \tau^*_A \to \tau_A, \end{aligned}$$

both of which follow directly from Lemma 3.22.

The remaining cases are straightforward and omitted. □

Lemma 3.33. *Fix a prenex formula A. The order of τ_A, τ^*_A are as presented in Table 2 where $\dot{-}$ denotes subtraction truncated at 0, i.e. $n \dot{-} m = \max\{n - m, 0\}$.*

Proof. By induction on complexity of A. If A is quantifier free then $\tau_A = \epsilon = \tau^*_A$ so $ord(\tau_A) = ord(\tau^*_A) = 0$. Moreover, by Example 3.23, the lemma holds for $A \in (\Sigma_1 \setminus \Pi_1) \cup (\Pi_1 \setminus \Sigma_1)$. Suppose $n > 1$. For $A = \exists\vec{v}B$ where $B \in \Pi_{n-1} \setminus \Sigma_{n-1}$,

$$\begin{aligned} ord(\tau_A) &= ord(\tau^*_{\bar{B}}) && \text{(Lemma 3.22(4))} \\ &= n - 2 && \text{(induction hypothesis)} \\ ord(\tau^*_A) &= \max\{ord(\tau_A) + 1, ord(\tau_{\bar{B}})\} && \text{(definition)} \\ &= n - 1 && \text{(induction hypothesis)} \end{aligned}$$

For $A = \forall\vec{v}B$ where $B \in \Sigma_{n-1} \setminus \Pi_{n-1}$,

$$\begin{aligned} ord(\tau_A) &= ord(\tau_B) && \text{(definition)} \\ &= n \dot{-} 3 && \text{(induction hypothesis)} \\ ord(\tau^*_A) &= ord(\tau_{\bar{A}}) && \text{(definition)} \\ &= n - 2 && \square \end{aligned}$$

Corollary 3.34. *For a proof $\pi \vdash A_0, \dots, A_k$ and $i < k$, the order of the non-terminal N^i_π is equal to the smallest n such that $\{A_j : j \leq k\} \subset \Pi_{n+1}$, unless A_i is Π_1, in which case the order of N^i_π is zero.*

Proof. If A_i is Π_1 then $\tau_{A_i} = \epsilon$ and the order of N^i_π is 0 by definition. Otherwise, the order of N^i_π is one greater than the maximum among the orders of $\tau^*_{A_j}$ for $j \leq k$. □

It is now possible to strengthen Lemma 3.26 to a concrete bound on the number of terms derivable from a Herbrand scheme. The idea is to eliminate occurrences of pattern-matching in a Herbrand scheme $\mathscr{H}$ in a way that does not decrease the length of derivations so that Theorem 3.17 and Corollary 3.18 can be applied.

Theorem 3.35. *If $\pi \vdash \Gamma$ is a proof of a single prenex Σ_1 formula in which all cut formulæ are contained in $\Pi_n \cup \Sigma_n$ then the size of the Herbrand expansion Γ^π is no greater than $2^{4|\pi|^3}_{n+2}$ where $|\pi|$ is the number of inference rules in π.*

Proof. The case $n = 0$ is covered by Lemma 3.31 so suppose $n > 0$. Let $\mathscr{H}$ be the Herbrand scheme of π. Since the cut rank of π is bounded by n, Corollary 3.34 implies that the order of $\mathscr{H}$ is no greater than n. To obtain the desired bounds we apply Theorem 3.17. However, this requires first eliminating the explicit substitutions introduced by the $\forall$ inferences. Let $\mathscr{H}'$ denote the higher order recursion scheme with non-terminals of basic type obtained from $\mathscr{H}$ by removing all substitutions terms and types from non-terminals and production rules. In particular, the productions originating from $\forall_{\vec{\alpha}}$ and $\exists_{\vec{r}}$ inferences are replaced by following in $\mathscr{H}'$:

$$\forall_{\vec{\alpha}}\colon\quad \mathsf{N}^i_{\pi'}\vec{x}\langle z_0, \dots, z_{p+1}\rangle \to \mathsf{N}^i_{\pi'_0}\vec{x}z_p$$

$$\exists_{\vec{r}}\colon\qquad \mathsf{N}^i_\pi\vec{x}z \to \begin{cases} \mathsf{N}^i_{\pi_0}\vec{x}(z(\mathsf{N}^m_\pi\vec{x})), & i \neq m, \\ \langle \mathsf{c}, \dots, \mathsf{c}, \mathsf{N}^m_{\pi_0}\vec{x}\rangle, & i = m \text{ and } u(A) > 0, \\ \langle \mathsf{c}, \dots, \mathsf{c}, \langle\rangle\rangle, & i = m \text{ and } u(A) = 0. \end{cases}$$

The second part of Lemma 3.27 implies that derivations in $\mathscr{H}'$ from the start symbol are in 1-1 correspondence with derivations in $\mathscr{H}$. Repeating the argument of Corollary 3.18, the size of $L(\pi)$ is therefore bounded by 2^K where K is the length of the longest derivation in $\mathscr{H}'$ from the single start symbol. The order of $\mathscr{H}'$ is no greater than n, the number of non-terminals is bounded by $|\pi|^2$, and for each production rule $\mathsf{F}\vec{x} \to t$ in $\mathscr{H}'$, $|t|_\Sigma < 3 \times |\pi|$ where Σ is the ranked alphabet of function symbols and constants occurring in π. Theorem 3.17 then implies $K \leq 2^{4|\pi|^3}_{n+1}$. □

4 Example: a Herbrand Disjunction for the Pigeonhole Principle

We consider a formal proof of the pigeonhole principle for two boxes via the infinite pigeonhole principle. The question of the computational content of this proof is attributed to G. Stolzenberg in [12]. A variety of analytic methods have since been applied to this proof [23, 9, 43, 7, 1] and its generalisations [40, 38]. The version we present here is a formal proof with a single Π_3 cut based on the proof with two Π_2 cuts given in [1, 43].

Let $f\colon \mathbb{N} \to \{0, 1\}$ be a total Boolean function, let I_i (for $i = 0, 1$) express that there are infinitely many $m \in \mathbb{N}$ for which $f(m) = i$ and T express that there exists $m < n$ such that $f(m) = f(n)$. A consequence of the law of excluded middle is $\exists w I_w$. Moreover,

I_i implies T for each $i \in \{0,1\}$: assuming I_i there exists $m \geq 0$ and $n \geq m+1$ for which $f(m) = f(n) = i$. Combining these observations we conclude T.

The following formalises the above argument into a proof with a single Π_3 cut. The formal language, Σ, comprises two unary function symbols f, s, one binary function symbol m, a constant symbol 0 and a binary relation $\leq$. We make the following definitions and abbreviations:

- $T = \exists u \exists v (u < v \wedge \mathsf{f}u = \mathsf{f}v)$,
- $I = \exists w I_w$ where $I_r = \forall u \exists v (u \leq v \wedge \mathsf{f}v = r)$,
- $\Gamma = \{\forall u \forall v (u \leq \mathsf{m}uv \wedge v \leq \mathsf{m}uv), \forall u(\mathsf{f}u = 0 \vee \mathsf{f}u = \mathsf{s}0)\}$,
- $\Delta = \{\forall u \forall v \forall w (u = v \wedge w = v \to u = w), \forall u \forall v (\mathsf{s}u \leq v \to u < v)\}$,
- I_r^s and $I_r^{s,t}$ denote, respectively, $\exists v(s \leq v \wedge \mathsf{f}v = r)$ and $(s \leq t \wedge \mathsf{f}t = r)$,
- $T_{s,t}$ denotes $(s < t \wedge \mathsf{f}s = \mathsf{f}t)$.

The intended interpretation of the symbols is: f represents the (arbitrary) function f, s the successor function on $\mathbb{N}$, $\leq$ the standard ordering and m the binary max function.

$$
\mathsf{cut}\ \dfrac{
\mathsf{c}\,\dfrac{\exists\,\dfrac{\exists\,\dfrac{\forall_{\hat\alpha}\,\dfrac{\forall_{\alpha}\,\dfrac{\exists\,\dfrac{\exists\,\dfrac{h : \Gamma \vdash I_0^{\alpha,\mathsf{m}\alpha\hat\alpha}, I_1^{\hat\alpha,\mathsf{m}\alpha\hat\alpha}}{g : \Gamma \vdash I_0^{\alpha,\mathsf{m}\alpha\hat\alpha}, I_1^{\hat\alpha}}}{f : \Gamma \vdash I_0^{\alpha}, I_1^{\hat\alpha}}}{e : \Gamma \vdash I_0, I_1^{\hat\alpha}}}{d : \Gamma \vdash I_0, I_1}}{c : \Gamma \vdash I, I_1}}{b : \Gamma \vdash I, I}}{a : \Gamma \vdash I}
\qquad
\forall_{\gamma}\,\dfrac{\mathsf{c}\,\dfrac{\exists\,\dfrac{\forall_{\beta}\,\dfrac{\exists\,\dfrac{\forall_{\hat\beta}\,\dfrac{\exists\,\dfrac{7 : \Delta \vdash \bar{I}_\gamma^{0,\beta}, \bar{I}_\gamma^{\mathsf{s}\beta,\hat\beta}, T_{\beta,\hat\beta}}{6 : \Delta \vdash \bar{I}_\gamma^{0,\beta}, \bar{I}_\gamma^{\mathsf{s}\beta,\hat\beta}, T}}{5 : \Delta \vdash \bar{I}_\gamma^{0,\beta}, \bar{I}_\gamma^{\mathsf{s}\beta}, T}}{4 : \Delta \vdash \bar{I}_\gamma^{0,\beta}, \bar{I}_\gamma, T}}{3 : \Delta \vdash \bar{I}_\gamma^{0}, \bar{I}_\gamma, T}}{2 : \Delta \vdash \bar{I}_\gamma, \bar{I}_\gamma, T}}{1 : \Delta \vdash \bar{I}_\gamma, T}}{0 : \Delta \vdash \bar{I}, T}
}{\Gamma, \Delta \vdash T}
$$

Figure 4: Proof π_∞ of pigeonhole principle.

A formal proof of the pigeonhole principle (namely $\Gamma, \Delta \vdash T$) is given in Figure 4 which we name π_∞. The proof is displayed in two-sided sequent calculus as this simplifies the presentation and following discussion. The intended interpretation of the two-sided sequent $A_1, \ldots, A_k \vdash B_1, \ldots, B_l$ is the sequent $\bar{A}_1, \ldots, \bar{A}_k, B_1, \ldots, B_l$. For brevity, only eigenvariables and witnesses of the quantifiers and instances of the existential formula T are displayed in π_∞. The proof fully fleshed out uses about 50 application of the axioms and rules of the calculus but the only cut in π_∞ is the one displayed in the figure. Two normal forms of the proof of size ∼200 have been computed in a case study [43] from which one can read off the Herbrand sets for the formula T (also for formulæ in $\Gamma \cup \Delta$ but these are less interesting). Up to interpretation of the logical symbols by their intended semantics, the two Herbrand sets combined provide the witnesses $\{\langle 0,1\rangle, \langle 1,2\rangle, \langle 2,3\rangle, \langle 0,2\rangle, \langle 1,3\rangle\}$ to the existential quantifiers in T.[1] The Herbrand scheme $\mathscr{H}_{\pi_\infty}$ associated to the proof π_∞ computes the same Herbrand set, a fact we demonstrate in the following.

Types and terms The Herbrand scheme for π_∞ comprises a non-terminal for each sub-proof of π_∞ and each formula in the end-sequent of that sub-proof. Recall, for each sub-proof $p : \Pi \vdash \Lambda$ of π_∞ and each $i < |\Pi| + |\Lambda|$ there is a non-terminal N_p^i in $\mathscr{H}_{\pi_\infty}$ representing the existential content of the i-th formula in the sequent at position p. In the following, in place of N_p^i we will write N_p^A where A is the i-th formula in the sequent assuming this is unique. In case A occurs more than once in the sequent $\Pi \vdash \Lambda$ (such as at positions b and 2) the non-terminal N_p^A refers to the first occurrence of A and we use the notation N_p^{A+} for the second occurrence. Concerning the type of N_p^A, we recall the types τ_F and τ_F^* for each formula F in π_∞. Let $\hat\iota = \iota \times (\iota^1 \to \epsilon)$ and $\hat\epsilon = \iota^2 \to \epsilon$.

- For $F \in \Gamma \cup \Delta$ we have $\tau_{\bar F} \in \{\iota^1, \iota^2, \iota^3\}$, and $\tau_{\bar F}^* = \tau_{\bar F} \to \epsilon$.
- T: $\tau_T = \iota^2$, $\tau_T^* = \hat\epsilon$.
- I_r^s: $\tau_{I_r^s} = \iota^1 = \tau^*_{\bar I_r^s}$, $\tau^*_{I_r^s} = \iota^1 \to \epsilon$ and $\tau_{\bar I_r^s} = \epsilon$.
- I_r: $\tau_{I_r} = \iota^1$, $\tau_{\bar I_r} = \tau^*_{I_r} = \hat\iota$ and $\tau^*_{\bar I_r} = \hat\iota \to \iota^1$.
- I: $\tau_I = \iota \times (\hat\iota \to \iota^1) = \tau^*_{\bar I}$, $\tau_{\bar I} = \hat\iota$ and $\tau_I^* = \tau_I \to \tau_{\bar I}$.
- The remaining formulæ that occur in π_∞ are quantifier-free and are assigned type ϵ in all cases.

According to the definition, the type of $\mathsf{N}^T_{\pi_\infty}$ is $\varsigma \to \tau^*_{\bar F} \to \tau^*_{\bar G} \to \tau^*_{\bar C} \to \tau^*_{\bar D} \to \tau^*_T \to \tau_T$ where ς is the type of substitution stacks, F and G are the two formulæ in Γ and C and D are the formulæ in Δ. As the formulæ in $\Gamma \cup \Delta$ are Σ_1, their input type carries no computational content (cf. Lemma 5.17), and we can ignore these formulæ and identify the type above with $\varsigma \to \tau_T^* \to \tau_T$, and the term $\mathsf{N}^T_{\pi_\infty} a \mathsf{c}_{\tau^*_{\bar F}} \mathsf{c}_{\tau^*_{\bar G}} \mathsf{c}_{\tau^*_{\bar C}} \mathsf{c}_{\tau^*_{\bar D}}$ with $\mathsf{N}^T_{\pi_\infty} a$. Likewise, the type of $\mathsf{N}_c^{I_1}$ is assumed to be $\varsigma \to \tau_I^* \to \tau^*_{I_1} \to \tau_{I_1}$ and the type of $\mathsf{N}_2^{\bar I^+_\gamma}$ is $\varsigma \to (\hat\iota \to \iota^1) \to (\hat\iota \to \iota^1) \to \hat\epsilon \to \hat\iota$.

Other abbreviations and simplifications we utilise are:

- $\langle r\rangle$ for either the sequence $\langle r, \langle\rangle\rangle$ or $\langle r, \mathsf{c}_{\iota^1 \to \epsilon}\rangle$, depending on type, and $\langle r, s\rangle$ as a term of type ι^2 represents $\langle r, s, \langle\rangle\rangle$.
- $\hat 0 = \mathsf{m}00$, $1 = \mathsf{s}\hat 0$, $\hat 1 = \mathsf{m}01$, $2 = \mathsf{s}(\mathsf{m}10)$ and $\hat 2 = \mathsf{m}02$.
- For each non-terminal N_p^A where A is the i-th formula at position p, an additional non-terminal $\hat{\mathsf{N}}_p^A$ with the same arity as N_p^A and associated production rule
$$\hat{\mathsf{N}}_p^A a x_0 \cdots x_k \to \mathsf{N}_p^A a x_0 \cdots x_{i-1} x_k x_i \cdots x_{k-1}$$
is included in the Herbrand scheme $\mathscr{H}_{\pi_\infty}$. These non-terminals ease the computation in derivation steps involving permutation.

[1] In [43] π_∞ is formalised as a proof with two Π_2-cuts but as far as computing Herbrand sets, the two proofs are essentially identical.

- The Herbrand scheme also includes explicit non-terminals for non-determinism at each type, which are represented via set notation: for terms $s_0, \dots, s_k : \rho$ of the same type, the set $S = \{s_i \mid i \leq k\}$ is a term of type ρ with reduction $S \to s_i$ for each $i \leq k$.

- An equivalence relation $\asymp$ on terms of identical type defined as inducing the same language within all contexts. Formally, we set $r \asymp s$ iff $r \prec s \prec r$ where $r \prec s$ holds just if $r, s : \rho$ and for every $\mathscr{H}_{\pi_\infty} \cup \{x^\rho\}$-term t of basic type (where x is a fresh symbol of type ρ), whenever $t(r/x) \to^* u$ for a Σ-term u, then $t(s/x) \to^* v$ for some Σ-term v such that $u^\circ = v^\circ$.

For instance, if $r \to s$ via an application of a deterministic production rule then $r \asymp s$, and if $S = S'$ are two representations of the same set of terms then $S \asymp S'$. In general, $r(S/x) \not\asymp \{r(s/x) \mid s \in S\}$ as shown by considering $r = \mathsf{F}x$ with reduction $\mathsf{F}x \to^* \mathsf{m}xx$.[2] However, suppose $r = \mathsf{F}t_1 \cdots t_k x$, S is a set of pairs, F is deterministic and $\mathsf{F}t_1 \cdots t_k \langle x, x' \rangle \to t$. Then $r(S/x) \asymp \{r(s/x) \mid s \in S\} \asymp \{t((u,v)/(x,x')) \mid \langle u, v \rangle \in S\}$.

Finally, we remark that, generalising Lemma 3.29, for every type ρ with co-domain ϵ and every term $r : \rho$, we have $r \asymp \mathsf{c}_\rho$.

Language of π_∞ We now compute the language of $\mathscr{H}_{\pi_\infty}$ focussing on the formula T, i.e. set of terms (after evaluation) derivable from the term $\mathsf{N}^T_{\pi_\infty} \bot \mathsf{c}_{\hat\epsilon}$. The first, and only, production rule applicable to this term is given by the cut rule at the root of the proof:

$$\mathsf{N}^T_{\pi_\infty} \bot \mathsf{c}_{\hat\epsilon} \to \mathsf{N}^T_0 \bot (\mathsf{N}^I_a \bot (\hat{\mathsf{N}}^{\bar I}_0 \bot \mathsf{c}_{\hat\epsilon})) \mathsf{c}_{\hat\epsilon}. \tag{2}$$

Analysing derivations directly from this term is complicated. As the right sub-proof at 0 culminates in a $\forall_\gamma$ inference, the external non-terminal $\mathsf{N}^{\bar I}_0$ cannot be reduced until its second argument (the term $\mathsf{N}^I_a \bot (\hat{\mathsf{N}}^{\bar I}_0 \bot \mathsf{c}_{\hat\epsilon})$) is reduced to an explicit pair. But the inference at a in the left sub-proof is a contraction, so this immediately introduces non-determinism and duplication of arguments. After resolving the non-determinism and reducing the two continuations of N^I_a to pairs (say in terms of $\mathsf{N}^{I_0}_d / \mathsf{N}^{I_1}_d$), the external non-terminal can be reduced. The argument $\hat{\mathsf{N}}^{\bar I}_0 \bot \mathsf{c}_{\hat\epsilon}$ comes into play at this point: the productions for $\mathsf{N}^{I^+}_b$ and N^I_c increase the nesting of non-terminals which must also be evaluated as pairs in order to proceed beyond $\mathsf{N}^{I_0}_d / \mathsf{N}^{I_1}_d$.

In the following, we compute the language via a top-down approach, analysing derivations starting from relatively simple terms, and building these together to compute the language of more complex interactions between non-terminals. We begin with the most simple derivations available. Recall that $\tau^*_{I_0} = \tau^*_{I_1} = \iota^1$. Concerning non-terminals from the left sub-proof we have the following derivation starting from $\mathsf{N}^{I_0}_d / \mathsf{N}^{I_1}_d$.

$$\begin{aligned} \mathsf{N}^{I_i}_d a \langle r \rangle \langle s \rangle &\to \mathsf{N}^{2+i}_e ([\hat\alpha \leftarrowtail s] a) \langle r \rangle \langle\rangle \\ &\to \mathsf{N}^{2+i}_f ([\alpha \leftarrowtail r][\hat\alpha \leftarrowtail s] a) \langle\rangle \langle\rangle \\ &\to^* \langle \mathsf{m} \alpha \hat\alpha \cdot [\alpha \leftarrowtail r][\hat\alpha \leftarrowtail s] a \rangle \end{aligned}$$

[2] Formally, we require an $\mathscr{H}_{\pi_\infty}$ analogue of F but this is not difficult to find.

(Note, as $|\Gamma| = 2$ the $(2+i)$-th formula at positions e and f is the ancestor of I_i from d.) If r happens to be such that $r \cdot [\hat{\alpha} \mapsfrom s]a \asymp r \cdot a$, then since the derivation above follows deterministic reductions only, we deduce

$$\mathsf{N}_d^{I_i} a\langle r\rangle\langle s\rangle \asymp \langle \mathsf{m} rs \cdot a\rangle. \tag{3}$$

Examining the non-terminals from lower in the left sub-proof affords us

$$\begin{aligned}
\mathsf{N}_c^{I} ars &\to \langle 0, \hat{\mathsf{N}}_d^{I_0} as\rangle & \qquad \mathsf{N}_c^{I_1} ars &\to^* \mathsf{N}_d^{I_1} a(r\langle 0, \hat{\mathsf{N}}_d^{I_0} as\rangle)s \\
\mathsf{N}_b^{I^+} ars &\to \langle 1, \mathsf{N}_c^{I_1} ar\rangle & \mathsf{N}_b^{I} ars &\to \mathsf{N}_c^{I} ar(s(\mathsf{N}_b^{I^+} ars)) \\
& & &\to^* \langle 0, \hat{\mathsf{N}}_d^{I_0} a(s\langle 1, \mathsf{N}_c^{I_1} ar\rangle)\rangle
\end{aligned}$$

The derivation from $\mathsf{N}_c^{I_1} ars$ can be continued provided that the two arguments of $\mathsf{N}_d^{I_1}$, namely $r\langle 0, \hat{\mathsf{N}}_d^{I_0} as\rangle$ and s, are reducible to pairs. Thus if $r\langle 0, \hat{\mathsf{N}}_d^{I_0} a\langle s\rangle\rangle \to^* \langle r_0, r_0'\rangle$ and $r_0 \cdot [\hat{\alpha} \mapsfrom s]a \asymp r_0 \cdot a$ then

$$\begin{aligned}
\mathsf{N}_c^{I_1} ar\langle s\rangle &\to^* \mathsf{N}_d^{I_1} a\langle r_0\rangle\langle s\rangle \\
&\asymp \langle \mathsf{m} r_0 s \cdot a\rangle.
\end{aligned}$$

Because the reductions governing $\mathsf{N}_c^{I_1}$ are all deterministic, when phrased in terms of equivalences, this becomes

$$\left.\begin{aligned}
r\langle 0, \hat{\mathsf{N}}_d^{I_0} a\langle s\rangle\rangle &\asymp \{\langle r_i\rangle \mid i \le k\} \\
r_i \cdot [\hat{\alpha} \mapsfrom s]a &\asymp r_i \cdot a \text{ each } i \le k
\end{aligned}\right\} \text{ implies } \mathsf{N}_c^{I_1} ar\langle s\rangle \asymp \{\langle \mathsf{m} r_i s \cdot a\rangle \mid i \le k\} \tag{4}$$

Property (4) will be useful later.

Returning briefly to the derivations from non-terminals N_b^{I} and $\mathsf{N}_b^{I^+}$ started earlier, each of these derivations is also deterministic, so therefore

$$\begin{aligned}
\mathsf{N}_a^{I} ar &\asymp \{\mathsf{N}_b^{I} arr, \mathsf{N}_b^{I^+} arr\} \\
&\asymp \{\langle 0, \hat{\mathsf{N}}_d^{I_0} a(r\langle 1, \mathsf{N}_c^{I_1} ar\rangle)\rangle, \langle 1, \mathsf{N}_c^{I_1} ar\rangle\}
\end{aligned} \tag{5}$$

which provides the first step in the continuation of the derivation from $\mathsf{N}_{\pi_\infty}^{T}$. Before extending (2) however we consider some simple derivations arising from the right sub-proof. On this side, the alternation of universal and existential inference rules means that few non-terminals can be adequately analysed in isolation as we did above. Most straightforward are non-terminals $\mathsf{N}_4^{\bar{I}_\gamma}$ and $\mathsf{N}_2^{\bar{I}_\gamma}$, for which we have

$$\mathsf{N}_4^{\bar{I}_\gamma} arst \asymp \langle \mathsf{s}\beta \cdot a\rangle \qquad \mathsf{N}_2^{\bar{I}_\gamma} arst \asymp \langle 0 \cdot a\rangle \asymp \langle 0\rangle$$

This gives rise to, for example,

$$\mathsf{N}_3^{\bar{I}_\gamma} a\langle r_0\rangle st \asymp \mathsf{N}_4^{\bar{I}_\gamma}([\beta \mapsfrom r_0]a)\langle\rangle st \asymp \langle \mathsf{s} r_0 \cdot a\rangle$$

and hence if $r : (\iota \times (\iota^1 \to \epsilon)) \to \iota^1$ is a term such that $r\langle 0\rangle \asymp \{\langle r_i\rangle \mid i \le k\}$ then also

$$\begin{aligned}
\mathsf{N}_2^{\bar{I}_\gamma^+} arst &\asymp \mathsf{N}_3^{\bar{I}_\gamma} a(r(\mathsf{N}_2^{\bar{I}_\gamma} arst))st \\
&\asymp \mathsf{N}_3^{\bar{I}_\gamma} a\{\langle r_i\rangle \mid i \le k\}st \\
&\asymp \{\langle \mathsf{s} r_i \cdot a\rangle \mid i \le k\}
\end{aligned}$$

The equivalences for $\mathsf{N}_2^{\bar{I}_\gamma}$ and $\mathsf{N}_2^{\bar{I}_\gamma^+}$ combine to yield, given the same r,

$$\mathsf{N}_1^{\bar{I}_\gamma} art \asymp \{\mathsf{N}_2^{\bar{I}_\gamma} arrt, \mathsf{N}_2^{\bar{I}_\gamma^+} arrt\} \asymp \{\langle 0\rangle\} \cup \{\langle \mathsf{s}r_i \cdot a\rangle \mid i \leq k\}. \tag{6}$$

In particular, choosing $r = \hat{\mathsf{N}}_d^{I_0} \bot \langle s\rangle$, this implies

$$\mathsf{N}_1^{\bar{I}_\gamma} a(\hat{\mathsf{N}}_d^{I_0} \bot \langle s\rangle)t \asymp \{\langle 0\rangle, \langle \mathsf{s}(\mathsf{m}0s) \cdot a\rangle\} \tag{7}$$

which will be needed later. In addition to (7), it is necessary to analyse the complex term $\mathsf{N}_1^{\bar{I}_\gamma} a(\mathsf{N}_c^{I_1} \bot (\hat{\mathsf{N}}_0^{\bar{I}} \bot \mathsf{c}_{\hat{\epsilon}}))\mathsf{c}_{\hat{\epsilon}}$. However, here we can use (6) again. If $\delta : \iota$ is a fresh symbol then, applying (7) and (4) (using $r = \mathsf{N}_0^{\bar{I}} \bot \mathsf{c}_{\hat{\epsilon}}$), we get

$$\begin{aligned} \mathsf{N}_0^{\bar{I}} \bot \langle 0, \hat{\mathsf{N}}_d^{I_0} \bot \langle \delta\rangle\rangle \mathsf{c}_{\hat{\epsilon}} &\asymp \mathsf{N}_1^{\bar{I}_\gamma}([\gamma \mapsfrom 0]\bot)(\hat{\mathsf{N}}_d^{I_0} \bot \langle \delta\rangle)\mathsf{c}_{\hat{\epsilon}} \\ &\asymp \{\langle 0\rangle, \langle \mathsf{s}(\mathsf{m}0\delta)\rangle\} && (8) \\ \mathsf{N}_c^{I_1} \bot (\hat{\mathsf{N}}_0^{\bar{I}} \bot \mathsf{c}_{\hat{\epsilon}})\langle \delta\rangle &\asymp \{\langle \mathsf{m}0\delta\rangle, \langle \mathsf{m}(\mathsf{s}(\mathsf{m}0\delta))\delta\rangle\} && (9) \end{aligned}$$

whence (6) implies

$$\mathsf{N}_1^{\bar{I}_\gamma} a(\mathsf{N}_c^{I_1} \bot (\hat{\mathsf{N}}_0^{\bar{I}} \bot \mathsf{c}_{\hat{\epsilon}}))\mathsf{c}_{\hat{\epsilon}} \asymp \{\langle 0\rangle, \langle 1\rangle, \langle 2\rangle\}. \tag{10}$$

We have still not examined derivations starting from the non-terminals N_0^T, N_1^T, and N_i^T for $i \geq 2$, which will arise in the computation of $L(\pi_\infty)$. The first three non-terminals behave according to

$$\begin{aligned} \mathsf{N}_0^T a\langle r, s\rangle t &\asymp \mathsf{N}_1^T([\gamma \mapsfrom r]a)st \\ &\asymp \mathsf{N}_2^T([\gamma \mapsfrom r]a)sst \\ &\asymp \mathsf{N}_3^T([\gamma \mapsfrom r]a)(s\langle 0\rangle)st \end{aligned}$$

The remaining behave similarly to the N_i^A non-terminals analysed earlier, except that it is N_6^T that provides the only 'outputs' in the derivation. In particular,

$$\mathsf{N}_6^T ars \asymp \langle \beta, \hat{\beta}\rangle \cdot a \qquad\qquad \mathsf{N}_5^T ar\langle s_0\rangle t \asymp \langle \beta, s_0\rangle \cdot a$$
$$\mathsf{N}_3^T a\langle r_0\rangle st \asymp \mathsf{N}_5^T([\beta \mapsfrom r_0]a)\langle\rangle(s\langle \mathsf{s}r_0 \cdot a\rangle)t$$

Let $\delta : \iota$ be a fresh symbol. Combining the two sets of equations above, if $s : \hat{\iota} \to \iota^1$ is such that $s\langle \delta\rangle \asymp \{\langle s_i\rangle \mid i \leq k\}$ and s_i contains neither β or γ for each i, it follows that

$$\mathsf{N}_0^T \bot \langle r, s\rangle t \asymp \{\langle s_i \cdot [\delta \mapsfrom 0]\bot, s_j \cdot [\delta \mapsfrom \mathsf{s}s_i][\delta \mapsfrom 0]\bot\rangle \mid i, j \leq k\}\,. \tag{11}$$

We can now proceed with calculating the language of $\mathsf{N}_{\pi_\infty}^T \bot \mathsf{c}_{\hat{\epsilon}}$. Let $w = \hat{\mathsf{N}}_0^{\bar{I}} \bot \mathsf{c}_{\hat{\epsilon}}$. Following on from (2) and (5) we have

$$\begin{aligned} \mathsf{N}_{\pi_\infty}^T \bot \mathsf{c}_{\hat{\epsilon}} &\asymp \mathsf{N}_0^T \bot (\mathsf{N}_a^I \bot w)\mathsf{c}_{\hat{\epsilon}} \\ &\asymp \Big\{\mathsf{N}_0^T \bot \langle 0, \hat{\mathsf{N}}_d^{I_0} \bot (w\langle 1, \mathsf{N}_c^{I_1} \bot w\rangle)\rangle \mathsf{c}_{\hat{\epsilon}}, \mathsf{N}_0^T \bot \langle 1, \mathsf{N}_c^{I_1} \bot w\rangle \mathsf{c}_{\hat{\epsilon}}\Big\} && (12) \end{aligned}$$

Thus, we need only compute

$$\mathsf{N}_d^{I_0}\bot\langle\delta\rangle(w\langle 1, \mathsf{N}_c^{I_1}\bot w\rangle) \qquad\qquad \mathsf{N}_c^{I_1}\bot(\hat{\mathsf{N}}_0^{\bar{I}}\bot\mathsf{c}_{\hat{\epsilon}})\langle\delta\rangle$$

and apply (11) (assuming that the terms obtained will be free of β and γ). The latter was already established in (9):

$$\mathsf{N}_c^{I_1}\bot(\hat{\mathsf{N}}_0^{\bar{I}}\bot\mathsf{c}_{\hat{\epsilon}})\langle\delta\rangle \asymp \{\langle\mathsf{m}0\delta\rangle, \langle\mathsf{m}(\mathsf{s}(\mathsf{m}0\delta))\delta\rangle\}$$

For the former, we have $w\langle 1, \mathsf{N}_c^{I_1}\bot w\rangle \asymp \mathsf{N}_1^{\bar{I}_\gamma}([\gamma \mapsfrom 1]\bot)(\mathsf{N}_c^{I_1}\bot w)\mathsf{c}_{\hat{\epsilon}}$, whence (10) implies

$$\begin{aligned}\mathsf{N}_d^{I_0}\bot\langle\delta\rangle(w\langle 1, \mathsf{N}_c^{I_1}\bot w\rangle) &\asymp \mathsf{N}_d^{I_0}\bot\langle\delta\rangle(\mathsf{N}_1^{\bar{I}_\gamma}([\gamma \mapsfrom 1]\bot)(\mathsf{N}_c^{I_1}\bot w)\mathsf{c}_{\hat{\epsilon}}) \\ &\asymp \left\{\mathsf{N}_d^{I_0}\bot\langle\delta\rangle\langle s\rangle \mid s \in \{0,1,2\}\right\} \\ &\asymp \{\langle\mathsf{m}\delta 0\rangle, \langle\mathsf{m}\delta 1\rangle, \langle\mathsf{m}\delta 2\rangle\}\end{aligned}$$

Hence, by (11) and (12), we deduce

$$\begin{aligned}\mathsf{N}_{\pi_\infty}^{T}\bot\mathsf{c}_{\hat{\epsilon}} \asymp &\left\{\langle r, s^{[\delta \mapsfrom \mathsf{s}r]}\rangle \mid r \in \{\hat{0},\hat{1},\hat{2}\}, s \in \{\mathsf{m}\delta 0, \mathsf{m}\delta 1, \mathsf{m}\delta 2\}\right\} \\ &\cup \left\{\langle r, s^{[\delta \mapsfrom \mathsf{s}r]}\rangle \mid r \in \{\hat{0}, \mathsf{m}10\}, s \in \{\mathsf{m}0\delta, \mathsf{m}(\mathsf{s}(\mathsf{m}0\delta))\delta\}\right\}\end{aligned}$$

Under the standard interpretation of the symbols 0, s and m (as zero, successor and binary 'max') $L(\pi_\infty)$ ascribes to T the set

$$\{(0,1),(0,2),(1,2),(2,3),(1,3)\}\,.$$

5 Language Preservation

Recall the relation $\pi \rightsquigarrow \pi'$ which expresses that π' is obtained from π by the application of a reduction rule in Figures 2 and 3 to a sub-proof of π. In the present section we determine in which cases $\rightsquigarrow$ supports: (i) *language inclusion*: $\pi \rightsquigarrow \pi'$ implies $L(\pi') \subseteq L(\pi)$; and (ii) *language equality*: $\pi \rightsquigarrow \pi'$ implies $L(\pi') = L(\pi)$. Establishing language inclusion for cut reduction steps will suffice to derive the main theorem; language equality allows a more fine-grained study of the Herbrand content of proofs as if π_0 and π_1 are two proofs that can be connected by a sequence of forward and backward language preserving reductions then $L(\pi_0) = L(\pi_1)$.

The first task is to define a single unified Herbrand scheme in which we can reason about derivations concerning arbitrary regular proofs. At the same time, we expand the grammar by certain non-terminals that will facilitate the analysis. These additional non-terminals were introduced informally in the example of the previous section.

Definition 5.1 (Universal Herbrand Scheme)**.** Let Σ be the signature of first-order logic. We let $\mathscr{H}$ denote the infinite recursion scheme comprising:

1. a non-deterministic non-terminal $\mathsf{D}_\rho : \rho \to \rho \to \rho$ for each basic type ρ with production rules $\mathsf{D}_\rho rs \to r$ and $\mathsf{D}_\rho rs \to s$,

2. all non-terminals N_π^i, $\mathsf{S}_{\pi,i}$ and c_ρ from Definition 3.24 with their associated production rules formulated deterministically in terms of the D_ρ non-terminals above,

3. for each non-terminal $\mathsf{N}^i_\pi : \varsigma \to \tau_0 \to \cdots \tau_m \to \tau$ from the above with τ prime, a non-terminal $\hat{\mathsf{N}}^i_\pi$ with type and associated production rule

$$\hat{\mathsf{N}}^i_\pi : \varsigma \to \tau_0 \to \cdots \to \tau_{i-1} \to \tau_{i+1} \to \cdots \tau_m \to \tau_i \to \tau$$
$$\hat{\mathsf{N}}^i_\pi a x_0 \cdots x_m \to \mathsf{N}^i_\pi a x_0 \cdots x_{i-1} x_m x_i \cdots x_{m-1}$$

We refer to $\mathscr{H}$ as the *universal Herbrand scheme.*

Henceforth, a *term* is an $\mathscr{H}$-term and we write $\to$ in place of $\to_{\mathscr{H}}$. Finite sets of $\mathscr{H}$-terms will represent applications of the non-deterministic non-terminals D_ρ. Specifically, the set $\{s^\rho_0, \dots, s^\rho_k\}$ represents any term formed by combining all the terms s_0, …, s_k (possibly with repetitions) via the non-terminal D_ρ. If S is a finite set of terms of the same type, it follows that $S \to^* s$ for each $s \in S$.

Notice that there are no start symbols in $\mathscr{H}$. In this regard we may consider the individual Herbrand scheme $\mathscr{H}_\pi$ as obtained from $\mathscr{H}$ by specifying an appropriate set of start symbols. The new 'hat' non-terminals do not play a role in viewing $\mathscr{H}$ as a universal Herbrand scheme. Rather, they become useful in 'transferring' non-terminals lacking their final argument through applications of permutation. For example, the following partial proof (where we assume $u(A) > 0$) gives rise to the production rules on the right:

$$\exists_r\,\dfrac{\mathsf{p}\,\dfrac{\pi_1 \vdash A(r/v), B}{\pi_0 \vdash B, A(r/v)}}{\pi \vdash B, \exists v A} \qquad\qquad \begin{aligned} \mathsf{N}^i_{\pi_0} axz &\to \mathsf{N}^{1-i}_{\pi_1} azx \\ \mathsf{N}^1_\pi axz &\to \langle r \cdot a, \mathsf{N}^1_{\pi_0} x\rangle \\ \mathsf{N}^0_\pi axz &\to \mathsf{N}^0_{\pi_0} ax(z(\mathsf{N}^1_\pi axz)) \end{aligned}$$

yielding the derivation $\mathsf{N}^0_\pi axz \to^* \mathsf{N}^1_{\pi_1} a(z\langle r \cdot a, \mathsf{N}^1_{\pi_0} ax\rangle)x$. The derivation cannot be extended as it stands because $\mathsf{N}^1_{\pi_0}$ lacks an argument, meaning that it is not formally possible to express the term $\mathsf{N}^0_\pi axz$ by reference to the proof π_1 only without instantiating x and z by concrete terms. However, $\mathsf{N}^1_{\pi_0} ax$ is extensionally equal to the term $\hat{\mathsf{N}}^0_{\pi_1} ax$, allowing us to equate $\mathsf{N}^0_\pi axz$ with the term $\mathsf{N}^1_{\pi_1} a(z\langle r \cdot a, \hat{\mathsf{N}}^0_{\pi_1} ax\rangle)x$ for any choice of a, x and z. Equations such as these are useful in the close examination of the cut elimination process carried out in the sections below.

In the previous section a natural subsumption and equivalence relation on terms was introduced given by equating terms that induce the same language in all contexts. In the context of the universal Herbrand scheme $\mathscr{H}$, this subsumption is given by $r \prec s$ which holds just if $r, s : \rho$ for some ρ and, for every $\mathscr{H} \cup \{x^\rho\}$-term t of basic type, whenever $t(r/x) \to^* r_0$ for a Σ-term r_0, then $t(s/x) \to^* s_0$ for some Σ-term s_0 such that $r^\circ_0 = s^\circ_0$. The corresponding equivalence relation $\asymp$ is defined by $r \asymp s$ iff $r \prec s \prec r$. The following properties of the relations $\prec$ and $\asymp$ were remarked in the last section.

Lemma 5.2. *Let $r, s : \rho$ be $\mathscr{H}$-terms of the same type and S a finite set of terms of pair type $\sigma = \sigma_0 \times \cdots \times \sigma_l$.*

1. *If $r \to s$ then $s \prec r$. If, in addition, the reduction follows from a production rule for a deterministic non-terminal then $r \asymp s$.*

2. *If r and s are representations of the same finite set of $\mathscr{H}$-terms then $r \asymp s$.*

3. *If* $r = \mathsf{F}s_0 \cdots s_{k-1}x^\sigma$ *for a non-terminal* F *with production rule*

$$\mathsf{F}x_0 \cdots x_{k-1}\langle x_k, \dots, x_{k+l}\rangle \to t$$

then

$$r(S/x) \asymp \{r(s/x) \mid s \in S\} \asymp \{t((s_0 \dots, s_{k+l})/(x_0, \dots, x_{k+l})) \mid \langle s_k, \dots, s_{k+l}\rangle \in S\}.$$

4. *If the co-domain of* ρ *is* ϵ *then* $r \asymp c_\rho$*.*

Proof. Properties 1–3 are straight-forward, though for 3 we note that only deterministic non-terminals have production rules that invoke pattern-matching. 4 generalises Lemma 3.29 and is proved by induction on ρ and r, noting that $\langle\rangle \cdot a \asymp \langle\rangle$ for any substitution a. □

Both relations can be extended to proofs by setting $\pi' \prec \pi$ if π and π' have the same end-sequent and $\mathsf{N}^i_{\pi'} \prec \mathsf{N}^i_\pi$ for each i. For many cases of $\pi \rightsquigarrow \pi'$ indeed $\pi' \prec \pi$ (or even $\pi' \asymp \pi$), from which we may immediately deduce $L(\pi') \subseteq L(\pi)$ (resp. $L(\pi') = L(\pi)$). However, there exist reductions $\pi \rightsquigarrow \pi'$ for which $L(\pi') \subseteq L(\pi)$ but $\pi' \not\prec \pi$. These scenarios all concern reductions that interact with quantifiers and alter the contexts in which explicit substitutions appear in derivations. To establish language preservation also in these cases we introduce a coarser relation $\prec$, denoted $\sqsubset$, based on quantifying over contexts of a particular syntactic shape that is preserved by $\mathscr{H}$-derivations. Such terms, which we call *normal* $\mathscr{H}$*-terms*, are defined below.

5.1 Normal Terms and Subsumption

In order to focus on the impact of substitutions in $\mathscr{H}$-terms it is necessary to introduce a notion of free and bound occurrences of Σ-symbols in these terms where, recall, Σ is the signature of first-order logic. The *free* symbols of a basic Σ-term are simply the Σ-symbols that occur in the term; Σ-terms have no *bound* symbols. For a basic $\mathscr{H}$-term t, the *free* symbols of t are the Σ-symbols occurring in t combined with the Σ-symbols occurring in any proof π for which a non-terminal N^i_π or $\hat{\mathsf{N}}^i_\pi$ appears in t; the *bound* symbols of t are the eigenvariables of the proofs which occur leftmost in t. For non-basic terms, substitutions and substitution stacks are interpreted as contributing to the set of bound symbols, and limiting the set of free symbols in the natural way. Explicitly, for a substitution stack $a : \varsigma$ and $\mathscr{H}$-term $r : \rho$ we define

$$Bd(r) = \begin{cases} \emptyset, & \text{if } r \text{ is a } \Sigma\text{-term or } r = \mathsf{c}_\rho, \\ EV(\pi), & \text{if } r = \mathsf{N}^i_\pi \text{ or } r = \hat{\mathsf{N}}^i_\pi \text{ for some } i, \\ Bd(s) \cup Bd(t), & \text{if } r = \langle s, t\rangle, \\ Bd(s) \cup Bd(t), & \text{if } r = \mathsf{D}_\rho st, \\ Bd(s) \cup Bd(a), & \text{if } r = sa \text{ or } r = s \cdot a \text{ for } a : \varsigma, \\ Bd(s), & \text{if } r = st \text{ and } t : \tau \text{ where } \tau \neq \varsigma, \end{cases}$$

$$Bd(a) = \begin{cases} \emptyset, & \text{if } a = \bot, \\ \{\alpha\} \cup Bd(b), & \text{if } a = [\alpha \mapsfrom s]b, \end{cases}$$

$$Fr(r) = \begin{cases} \emptyset, & \text{if } r = \mathsf{c}_\rho \text{ or } r = \mathsf{D}_{\rho'}, \\ \{r\}, & \text{if } r \in \Sigma, \\ Fr(\pi), & \text{if } r = \mathsf{N}^i_\pi \text{ or } r = \hat{\mathsf{N}}^i_\pi \text{ for some } i, \\ Fr(s) \cup Fr(t), & \text{if } r = \langle s, t \rangle, \\ (Fr(s) \setminus Bd(a)) \cup Fr(a), & \text{if } r = sa \text{ or } r = s \cdot a \text{ for } a : \varsigma, \\ Fr(s) \cup Fr(t), & \text{if } r = st \text{ and } t : \tau \text{ where } \tau \neq \varsigma, \end{cases}$$

$$Fr(a) = \begin{cases} \emptyset, & \text{if } a = \bot, \\ (Fr(s) \setminus Bd(b)) \cup Fr(b), & \text{if } a = [\alpha \hookleftarrow s]b, \end{cases}$$

$EV(\pi)$ denotes the set of eigenvariables in the proof π, and $Fr(\pi)$ the set of all non-eigenvariable Σ-symbols occurring in the π. Notice that $Bd(\mathsf{N}^i_\pi)$ and $Fr(\mathsf{N}^j_\pi)$ are disjoint sets by definition.

Definition 5.3 (Normal terms)**.** A *normal term* is an $\mathscr{H}$-term r satisfying:

1. if a is substitution stack which is a sub-term of r then $Bd(a) \cap Fr(a) = \emptyset$,

2. if st is an application which is a sub-term of r then $Bd(s) \cap Fr(t) = \emptyset$,

3. if $s \cdot a$ is a substitution occurring as a sub-term of r then s is of simple type.

As mentioned at the beginning of this section, the aim of the above definition is to provide a class of terms for which we can examine a more refined subsumption relation on $\mathscr{H}$-terms that captures both language inclusion and equality for a wide range of cut reduction rules. The subsumption relation that achieves this is essentially the restriction of $\prec$ that only quantifies over normal contexts.

Definition 5.4 (Subsumption)**.** Given normal $\mathscr{H}$-terms $r, s : \rho$ of the same type, s *subsumes* r, in symbols $r \sqsubset s$, just if, for every $\mathscr{H} \cup \{x^\rho\}$-term t of basic type such that $t(r/x)$ and $t(s/x)$ are both normal, whenever $t(r/x) \to^* u$ for a Σ-term u then $t(s/x) \to^* v$ for some Σ-term v satisfying $u^\circ = v^\circ$. Define $r \sim s$ if $r \sqsubset s$ and $s \sqsubset r$.

Clearly, for normal terms r and s, $r \prec s$ implies $r \sqsubset s$, and $r \asymp s$ implies $r \sim s$. Hence, if π, π' are two regular proofs of a Σ_1 sequent Γ and $\mathsf{S}_{\pi',i} \sqsubset \mathsf{S}_{\pi,i}$ for every $i < |\Gamma|$ then $L(\pi') \subseteq L(\pi)$. However, what we require is the more general property that if for every $\mathscr{H}$-derivation $\mathsf{S}_{\pi',i} \to^* u$ of a Σ-term there exists $\mathscr{H}$-terms r, s and t such that $\mathsf{S}_{\pi',i} \to^* t(r/x) \to^* u$, $\mathsf{S}_{\pi,i} \to^* t(s/x)$ and $r \sqsubset s$, then we may conclude $L(\pi') \subseteq L(\pi)$. This result holds trivially for $\prec$ in place of $\sqsubset$. For it to work for subsumption, the terms r, s and t must all be normal, i.e., we require

Lemma 5.5. *If* $\mathsf{S}_{\pi,i} \to^* s$ *then* s *is a normal term.*

The proof of Lemma 5.5 is not difficult but requires some technical observations concerning the preservation of free and bound symbols through $\mathscr{H}$-derivations, so will be postponed to the next section (Lemma 5.10). Another key lemma concerning derivations is the following which, when combined with Lemma 5.5, implies that all derivations from start symbols lead to Σ-terms.

Lemma 5.6. *If r is a normal term of prime type and not a Σ-term then $r \to s$ for some term s.*

There are two scenarios in which a term r that is not a Σ-term cannot be reduced. First, the reduction of r may require pattern-matching on an argument that cannot be reduced to a pair, such as $\langle s, t\rangle \cdot a$. Second, r may contain a sub-term $\mathsf{F} r_1 \cdots r_k \cdot a$ for which the arity of F is greater than k. Normality prevents either scenario from occurring (Lemma 5.11). As a consequence, terms of pair type always reduce to pairs (Lemma 5.12) and pattern-matching will not block derivations. Thus,

Lemma 5.7 (Finite basis lemma)**.** *For every normal $\mathscr{H}$-term r of pair type there exists terms $\langle s_0, t_0\rangle$, ..., $\langle s_k, t_k\rangle$ such that $r \sim \{\langle s_i, t_i\rangle \mid i \leq l\}$.*

We now turn to proving these three lemmas, and establishing some further properties of subsumption. The reader may wish to proceed directly to Section 5.3 at this point and refer back to Section 5.2 as needed.

5.2 Technical Lemmas

In order to establish Lemma 5.5, we require two observations on free and bound symbols in normal terms. The effect is to reduce the work to checking that production rules locally preserve normality (in contrast to within arbitrary contexts).

Lemma 5.8. *If $r(s/x)$ is a normal term, t is a normal term of the same type as s, $Fr(t) \subseteq Fr(s)$ and $Bd(t) \subseteq Bd(s)$ then $r(t/x)$ is normal.*

Proof. By definition. □

Lemma 5.9. *If $\mathsf{F} x_0 \cdots x_{k-1}\langle x_k, \ldots, x_{k+l}\rangle \to t$ is a production rule of $\mathscr{H}$, and r_0, ..., r_{k+l} are such that $s = \mathsf{F} r_0 \cdots r_{k-1}\langle r_k, \ldots, r_{k+l}\rangle$ is normal, then $Fr(t(\vec{r}/\vec{x})) \subseteq Fr(s)$ and $Bd(t(\vec{r}/\vec{x})) \subseteq Bd(s)$.*

Proof. We examine two particular cases, namely the quantifier rules, and leave the remaining for the reader to check. Consider an instance of the production rule for $\forall_\alpha$ for a single eigenvariable:

$$\mathsf{N}^i_\pi a r_0 \cdots r_{k-1}\langle s, r_k\rangle \to \mathsf{N}^i_{\pi_0}([\alpha \hookleftarrow s]a) r_0 \cdots r_k$$

where r_0, ..., r_k and s are terms of suitable type and number, and $a : \varsigma$ is a substitution stack. Let m and n abbreviate the left- and righthand term in the above rule respectively. Assume m is a normal term.

$$\begin{aligned} Bd(m) &= Bd(\mathsf{N}^i_\pi a) \\ &= EV(\pi) \cup Bd(a) \\ &= EV(\pi_0) \cup \{\alpha\} \cup Bd(a) \\ &= Bd(n) \end{aligned}$$

Concerning free symbols, we have

$$\begin{aligned} Fr(m) &= (Fr(\pi) \setminus Bd(a)) \cup Fr(a) \cup Fr(r_0, \ldots, r_k, s) \\ Fr(n) &= (Fr(\pi_0) \setminus Bd([\alpha \hookleftarrow s]a)) \cup Fr([\alpha \hookleftarrow s]a) \cup Fr(r_0, \ldots, r_k) \\ &= (Fr(\pi_0) \setminus (\{\alpha\} \cup Bd(a))) \cup (Fr(s) \setminus Bd(a)) \cup Fr(a) \cup Fr(r_0, \ldots, r_k) \end{aligned}$$

where $Fr(u_0, \dots, u_l) = \bigcup_{i \leq l} Fr(u_i)$. By normality of m, $Fr(s) \setminus Bd(a) = Fr(s)$ and as $Fr(\pi) = Fr(\pi_0) \setminus \{\alpha\}$, so $Fr(n) \subseteq Fr(m)$.

For production rules resulting from the inference rule $\exists_s$, suppose

$$\mathsf{N}^i_\pi a r_0 \cdots r_k \to \mathsf{N}^i_{\pi_0} a r_0 \cdots r_{k-1}(r_k \langle s \cdot a, \mathsf{N}^k_{\pi_0} a r_0 \cdots r_{k-1} \rangle) \tag{13}$$

for suitable terms r_0, ..., r_k and a. Recall that s is the Σ-term instantiating the existential quantifier in the active formula of π_0. By our regularity condition on proofs, $Fr(s) \subseteq Fr(\pi)$, so, letting m and n denote the left and right side of the reduction in (13), we have

$$\begin{aligned} Fr(n) &= Fr(\mathsf{N}^i_{\pi_0} a) \cup Fr(r_0, \dots, r_k) \cup Fr(s \cdot a) \cup Fr(\mathsf{N}^k_{\pi_0} a) \\ &\subseteq Fr(\mathsf{N}^i_\pi a) \cup Fr(r_0, \dots, r_k) \\ &= Fr(m). \end{aligned}$$

□

The following lemma implies Lemma 5.5.

Lemma 5.10. *If $r \to s$ and r is normal then s is normal.*

Proof. By the previous two lemmas, it suffices to show that every production rule of $\mathscr{H}$ locally preserves normality. As in the previous proof, we offer the argument for some particular cases of the two quantifier rules. Let the derivation

$$\mathsf{N}^i_\pi a r_1 \cdots r_k \langle s, r_{k+1} \rangle \to \mathsf{N}^i_{\pi_0}([\alpha \hookleftarrow s]a) r_1 \cdots r_k r_{k+1}$$

arise from an inference $\forall_\alpha$, with m and n denoting the left and righthand term. Assume m is normal. In particular,

$$(EV(\pi) \cup Bd(a)) \cap (Fr(a) \cup Fr(s)) = \emptyset \tag{14}$$

We first show that for every application $s't'$ occurring in n, $Bd(s') \cap Fr(t') = \emptyset$. This is evident if $s't'$ is a sub-term of a, r_1, ..., r_k, s or t. Moreover, it holds for the case $s' = \mathsf{N}^i_{\pi_0}$ and $t' = [\alpha \hookleftarrow s]a$ because

$$\begin{aligned} Bd(s') \cap Fr(t') &= EV(\pi_0) \cap ((Fr(s) \setminus Bd(a)) \cup Fr(a)) \\ &\subseteq EV(\pi) \cap (Fr(s) \cup Fr(a)) \\ &= \emptyset \end{aligned}$$

and for $s' = \mathsf{N}^i_{\pi_0}([\alpha \hookleftarrow s]a) r_1 \cdots r_j$, $t' = r_{j+1}$ $(j \leq k)$ because $Bd(\mathsf{N}^i_{\pi_0}([\alpha \hookleftarrow s]a)) \subseteq Bd(\mathsf{N}^i_\pi a)$. The other requirement to check for normality is that the sets $Bd([\alpha \hookleftarrow s]a)$ and $Fr([\alpha \hookleftarrow s]a)$ are disjoint, but this follows from (14), given that $Fr([\alpha \hookleftarrow s]a) \subseteq Fr(s) \cup Fr(a)$ and $Bd([\alpha \hookleftarrow s]a) \subseteq EV(\pi) \cup Bd(a)$. Hence n is normal.

The second case we consider is the one-step derivation

$$\mathsf{N}^i_\pi a r_0 \cdots r_k \to \mathsf{N}^i_{\pi_0} a r_0 \cdots r_{k-1}(r_k \langle s \cdot a, \mathsf{N}^k_{\pi_0} a r_0 \cdots r_{k-1} \rangle)$$

due to the inference $\exists_s$. Suppose $\mathsf{N}^i_\pi a r_0 \cdots r_k$ is a normal term so, in particular, $Bd(\mathsf{N}^i_\pi a)$ is disjoint from $Fr(r_i)$ for each $i \leq k$. In this case it only needs establishing that

$$Bd(\mathsf{N}^i_{\pi_0} a r_0 \cdots r_{k-1}) \cap Fr(r_k \langle s \cdot a, \mathsf{N}^k_{\pi_0} a r_0 \cdots r_{k-1} \rangle) = \emptyset$$

i.e., that

$$Bd(\mathsf{N}^i_{\pi_0}a) \cap \Big(\bigcup_{i\leq k} Fr(r_i) \cup Fr(s\cdot a) \cup Fr(\mathsf{N}^k_{\pi_0}a)\Big) = \emptyset,$$

as all other cases follow immediately from normality of $\mathsf{N}^i_\pi a r_0 \cdots r_k$. But by regularity of π, $Fr(s) \subseteq Fr(\pi)$, so we have

$$\begin{aligned} Bd(\mathsf{N}^i_{\pi_0}a) &= EV(\pi) \cup Bd(a) \\ Fr(s\cdot a) &\subseteq (Fr(\pi) \setminus Bd(a)) \cup Fr(a) \\ Fr(\mathsf{N}^k_{\pi_0}a) &= (Fr(\pi) \setminus Bd(a)) \cup Fr(a) \end{aligned}$$

and, as $Bd(a)$ is disjoint from $Fr(a)$ and $EV(\pi)$ is disjoint from $Fr(\pi) \cup Fr(a)$, we are done. □

We now turn to the task of proving Lemma 5.6 which, as explained earlier, follows from the next two lemmas. The first lemma characterises the form normal $\mathscr{H}$-terms may take, and will be useful in the subsequent analysis of derivations in Herbrand schemes.

Lemma 5.11. *If $r : \rho$ is a normal $\mathscr{H}$-term and ρ is a basic type whose co-domain is a pair $\sigma \times \tau$ in which σ is simple and τ is not simple, then either $r = \langle s, t\rangle$ for a Σ-term s and $\mathscr{H}$-term t, or $r = \mathsf{F}r_1 \cdots r_k$ for some non-terminal F and terms $r_1, \ldots, r_k$.*

Proof. By induction on r. Let $r : \rho = \rho_1 \to \cdots \to \rho_l \to \sigma \times \tau$ be a normal $\mathscr{H}$-term satisfying the hypothesis of the lemma. Since ρ is not a simple type, r is not a Σ-symbol nor of the form $s \cdot a$ for a substitution stack a (by definition of normal terms). This leaves three cases: i) $l = 0$ and $r = \langle s, t\rangle$ for $s : \sigma$ and $t : \tau$; ii) $r = st$ for $s : \tau' \to \rho$ and $t : \tau'$; or iii) r is a non-terminal of $\mathscr{H}$. If (i), as σ is simple, s is a Σ-term by Lemma 3.27 and we are done. In case (ii), suppose $r = st$ is an application and $s : \sigma' = \tau' \to \rho$ and $t : \tau'$. If σ' is not basic then Lemma 3.28 implies $s = \mathsf{F}$ for some π and i, whence $r = \mathsf{F}t$. On the other hand, if σ' is basic the induction hypothesis applies and $s = \mathsf{F}r_1 \cdots r_k$ for terms $r_1, \ldots, r_k$, and so similarly for r. So we are done. □

Lemma 5.12. *If $r : \iota \times \rho$ is a normal $\mathscr{H}$-term of pair type but not a pair then $r \to s$ for some $\mathscr{H}$-term s.*

Proof. Assume to the contrary that $r : \iota \times \rho$ is an $\mathscr{H}$-term which is not a pair and that there is no s such that $r \to s$. Without loss of generality assume r is minimal in length. By Lemma 5.11, $r = \mathsf{F}r_1, \ldots, r_k$ for some non-terminal F and terms $r_1, \ldots, r_k$. It follows that $\mathsf{F} \neq \mathsf{c}_\sigma$ for any σ as otherwise $r \to \langle \mathsf{c}, \mathsf{c}_\rho\rangle$. Also $\mathsf{F} \neq \mathsf{D}_{\iota\times\rho}$ (as then $r \to r_1$) and $\mathsf{F} \neq \hat{\mathsf{N}}^i_\pi$ for any π and i. So $\mathsf{F} = \mathsf{N}^i_\pi$ for some π and i. The fact that r is not reducible means that the production rule for N^i_π requires pattern-matching on the final argument. But then $r_k : \iota \times \sigma$ for some σ, is not a pair and is not reducible, contradicting minimality of r. □

In the remainder of this section we present some basic properties of the subsumption relation which will be needed in the next section, starting with a proof of the Finite Basis Lemma.

Proof of Lemma 5.7. Let $r : \rho$ be a $\mathscr{H}$-term where $\rho = \sigma \times \tau$. Without loss of generality, we may assume $r \neq \mathsf{D}_\rho r_0 r_1$ for any r_0 and r_1. If r has the form $\langle s, t\rangle$ then trivially $r \sim \{\langle s, t\rangle\}$ and if $r = \mathsf{c}_\rho$ then $r \sim \{\langle \mathsf{c}_\sigma, \mathsf{c}_\tau\rangle\}$. Otherwise, Lemma 5.11 implies that $r \sim \mathsf{N}^i_\pi a r_0 \cdots r_k$ for some π, i, a, r_0, …, r_k. An induction on π determines terms $s_0, \ldots, s_l$ and $t_0, \ldots, t_l$ such that $r \sim \{\langle s_j, t_j\rangle \mid j \leq l\}$. Note that there is no issue with pattern-matching stopping derivations from fully writing out, as Lemma 5.12 demonstrates. □

Lemma 5.13. *If $t(r/x)$ and $t(s/x)$ are normal terms and $r \sqsubset s$ then $t(r/x) \sqsubset t(s/x)$.*

Proof. Direct consequence of the definition. □

Lemma 5.14. *If $ru \sqsubset su$ for every term u then $r \sqsubset s$.*

Proof. For every derivation $t(r/x) \to^* r_0$ of a Σ-term, there are terms t', u_0, …, u_k such that $t(r/x) \to^* t'((ru_i)_{i\leq k}/\vec{x}) \to^* r_0$ and $t(s/x) \to^* t'((su_i)_{i\leq k}/\vec{x})$. Since normality is preserved through derivations, we are done. □

Lemma 5.15. *Let r be a basic $\mathscr{H}$-term and a be a substitution stack over Σ. Then,*

i) $Fr(r(s/\alpha) \cdot a) \subseteq Fr(r \cdot ([\alpha \leftarrowtail s]a))$,

ii) $Bd(r(s/\alpha) \cdot a) \subseteq Bd(r \cdot ([\alpha \leftarrowtail s]a))$ *provided s is basic,*

iii) if $Fr(r) \cap Bd(a) = \emptyset$ *then* $r^a = r$,

iv) if $\alpha \notin Bd(a) \cup Fr(a)$ *then* $r(s/\alpha)^a = r^a(s^a/\alpha)$ *provided s is basic.*

Proof. By induction on r and a. □

Lemma 5.16. *Let $r : \rho$ and $a : \varsigma$ be Σ-terms, $s : \sigma$ a basic Σ-term, $\alpha^\rho \in \Sigma$ and π a regular proof. Under the assumption that $\mathsf{N}^i_\pi([\alpha \leftarrowtail r]a)$ is normal the following hold.*

1. $s \cdot ([\alpha \leftarrowtail r]a) \sim s(r/\alpha) \cdot a$,

2. $r \sim r^\circ$,

3. $[\alpha \leftarrowtail r]a \sim [\alpha \leftarrowtail r \cdot a]a$,

4. *If* $EV(\pi) \cap Fr(r) = \emptyset$ *then* $\mathsf{N}^i_\pi([\alpha \leftarrowtail r]a) \sim \mathsf{N}^i_{\pi(r^\circ/\alpha)} a$,

5. *If* $\alpha \notin Fr(\pi)$ *then* $\mathsf{N}^i_\pi a \sim \mathsf{N}^i_\pi([\alpha \leftarrowtail r]a)$.

Proof. 1 is proved via induction on the basic term s. That r and a are Σ-terms is necessary for showing $s(r/\alpha) \cdot a \sqsubset s \cdot ([\alpha \leftarrowtail r]a)$. 2 follows from 1 by induction on r. Regarding 3, Lemma 5.15(i,ii) imply $Fr(a^\circ) \subseteq Fr(a)$ and $Bd(a^\circ) \subseteq Bd(a)$, so $\alpha \notin Fr(a^\circ)$ by normality. Hence, if t is a basic term then

$$\begin{aligned} t \cdot [\alpha \leftarrowtail r]a \sim t(r^\circ/\alpha)^{a^\circ} &\sim t^{a^\circ}((r^\circ)^{a^\circ}/\alpha) \\ &\sim t^{a^\circ}(((r^\circ)^{a^\circ})^{a^\circ}/\alpha) \\ &\sim t((r^\circ)^{a^\circ}/\alpha)^{a^\circ} \\ &\sim t \cdot [\alpha \leftarrowtail r \cdot a]a. \end{aligned}$$

The first and last equivalence are applications of 2; the second and fourth equivalence are consequences of Lemma 5.15(iv); and the third equivalence uses Lemma 5.15(iii) and the fact that $Fr(a^\circ) \cap Bd(a^\circ) = \emptyset$. Via 2 the above holds for t an arbitrary Σ-term, and from there generalises to deduce $[\alpha \mapsfrom r]a \sim [\alpha \mapsfrom r \cdot a]a$.

4 is derived by induction on π. By 2 we may assume r is a basic term, i.e., $r = r^\circ$. In the base case, where π is an axiom, the equivalence is trivial as $\mathsf{N}^i_\pi([\alpha \mapsfrom r]a)r_1r_2 \sim \mathsf{N}^i_{\pi(r/\alpha)}ar_1r_2$ for any choice of r_1 and r_2 of appropriate type. The induction step is straightforward except in the case of quantifier rules. If π ends in the inference

$$\exists_{\vec{s}}\,\frac{\pi_0 \vdash \Gamma, A(\vec{s}/\vec{v})}{\pi \vdash \Gamma, \exists\vec{v}A}$$

where $\vec{s} = (s_j)_{j\le k}$ then we have, if $i = |\Gamma|$, $b = [\alpha \mapsfrom r]a$ and $r_1, \ldots, r_{|\Gamma|}$ and t are suitable normal terms,

$$\begin{aligned}\mathsf{N}^{|\Gamma|}_\pi br_1 \cdots r_{|\Gamma|}t &\sim \langle s_0 \cdot b, \ldots, s_k \cdot b, \mathsf{N}^{|\Gamma|}_{\pi_0} br_1 \cdots r_{|\Gamma|}\rangle\\ &\sim \langle s_0(r/\alpha) \cdot a, \ldots, s_k(r/\alpha) \cdot a, \mathsf{N}^{|\Gamma|}_{\pi_0^{(r/\alpha)}} ar_1 \cdots r_{|\Gamma|}\rangle\\ &\sim \mathsf{N}^{|\Gamma|}_{\pi(r/\alpha)} ar_1 \cdots r_{|\Gamma|}t\end{aligned}$$

where the second equivalence due to the induction hypothesis for $\mathsf{N}^{|\Gamma|}_{\pi_0}([\alpha \mapsfrom r]a)$. The case $i < |\Gamma|$ is similar. For applications of the $\forall$ inferences, we consider the inference

$$\forall_{\vec{\beta}}\,\frac{\pi_0 \vdash \Gamma, A(\vec{\beta}/\vec{v})}{\pi \vdash \Gamma, \forall\vec{v}A}$$

for $\vec{\beta} = (\beta^\iota_j)_{j\le k}$. By normality of $\mathsf{N}^i_\pi([\alpha \mapsfrom r]a)$ and the assumption that $Fr(r) \cap EV(\pi) = \emptyset$, it follows that $\beta_j \notin Fr(a) \cup Fr(r)$ for each j. Let $r_1, \ldots, r_{|\Gamma|}, s_0, \ldots, s_k$ and t be such that the term $n := \mathsf{N}^i_\pi([\alpha \mapsfrom r]a)r_1 \cdots r_{|\Gamma|}\langle s_0, \ldots, s_k, t\rangle$ is well-typed and normal. In particular, $\alpha, \beta_0, \ldots, \beta_k \notin Fr(\langle s_0, \ldots, s_k, t\rangle)$. Moreover, as s_j has simple type for each $j \le k$, Lemma 3.27 implies that $\vec{s}$ is a sequence of Σ-terms. Then assuming $\alpha \notin \{\beta_j \mid j \le k\}$, and writing $[\vec{\beta} \mapsfrom \vec{s}]b$ in place of $[\beta_0 \mapsfrom s_0] \cdots [\beta_k \mapsfrom s_k]b$, we have

$$\begin{aligned}n &\sim \mathsf{N}^i_{\pi_0}([\vec{\beta} \mapsfrom \vec{s}][\alpha \mapsfrom r]a)r_1 \cdots r_{|\Gamma|}t\\ &\sim \mathsf{N}^i_{\pi_0^{(\vec{s}/\vec{\beta})(r/\alpha)}} ar_1 \cdots r_{|\Gamma|}t\\ &\sim \mathsf{N}^i_{\pi_0^{(r/\alpha)(\vec{s}/\vec{\beta})}} ar_1 \cdots r_{|\Gamma|}t\\ &\sim \mathsf{N}^i_{\pi_0^{(r/\alpha)}}([\vec{\beta} \mapsfrom \vec{s}]a)r_1 \cdots r_{|\Gamma|}t\\ &\sim \mathsf{N}^i_{\pi(r/\alpha)} ar_1 \cdots r_{|\Gamma|}\langle s_0, \ldots, s_k, t\rangle.\end{aligned}$$

The third equivalence holds since $\alpha \notin Fr(\langle s_0, \ldots, s_k, t\rangle)$ and $\beta_j \notin Fr(r)$ for any j. If

$\alpha = \beta_j$ then $\pi^{(r/\alpha)} = \pi$ and, using again that $\alpha \notin Fr(\langle s_0, \ldots, s_k \rangle)$, we have

$$\begin{aligned} n &\sim \mathsf{N}^i_{\pi_0}([\vec{\beta} \mapsfrom \vec{s}][\alpha \mapsfrom r]a)r_1 \cdots r_{|\Gamma|}t \\ &\sim \mathsf{N}^i_{\pi_0^{(\vec{s}/\vec{\beta})(r/\alpha)}} a r_1 \cdots r_{|\Gamma|}t \\ &\sim \mathsf{N}^i_{\pi_0^{(\vec{s}/\vec{\beta})}} a r_1 \cdots r_{|\Gamma|}t \\ &\sim \mathsf{N}^i_{\pi_0}([\vec{\beta} \mapsfrom \vec{s}]a)r_1 \cdots r_{|\Gamma|}t \\ &\sim \mathsf{N}^i_{\pi} a r_1 \cdots r_{|\Gamma|}\langle s_0, \ldots, s_k, t \rangle \\ &\sim \mathsf{N}^i_{\pi^{(r/\alpha)}} a r_1 \cdots r_{|\Gamma|}\langle s_0, \ldots, s_k, t \rangle. \end{aligned}$$

5 is a special case of 4. □

Let π be a proof with end-sequent $A_1, \ldots, A_m, B, C_1, \ldots, C_n$ and suppose $a : \varsigma$, $r_i : \tau^*_{A_i}$, $s : \tau^*_B$ and $t_j : \tau^*_{C_j}$ are terms for each $1 \leq i \leq m$ and $1 \leq j \leq n$ such that $\mathsf{N}^m_\pi a\vec{r}s\vec{t}$ is normal. Let $\rho = \tau^*_B$.

Lemma 5.17. *If B is prenex Σ_1 then*

$$\mathsf{N}^m_\pi a\vec{r}s\vec{t} \sim \mathsf{N}^m_\pi a\vec{r}\mathsf{c}_\rho\vec{t}.$$

Proof. Since B is Σ_1, $\tau^*_B = \tau_B \to \epsilon$. Lemma 5.2 completes the proof. □

Lemma 5.18. *If $e(B) > 0$ and there are no applications of contraction to B in π then*

$$\mathsf{N}^m_\pi a\vec{r}s\vec{t} \sim \mathsf{N}^m_\pi a\vec{r}\mathsf{c}_\rho\vec{t}.$$

Lemma 5.19. *If the final inference in π is an application of p with sub-proof $\pi' \vdash A_1, \ldots, A_m, C_1, B, C_2, \ldots, C_n$ then for each $j \in [0, m) \cup [m+2, m+n]$,*

$$\mathsf{N}^j_\pi a r_1 \cdots r_m s t_1 \cdots t_n \sim \mathsf{N}^j_{\pi'} a r_1 \cdots r_m t_1 s t_2 \cdots t_n$$

The proofs of the final two lemmas proceed by induction on π.

5.3 Language Preservation for Gentzen-style Cut Elimination

Let π and π' be regular proofs of some sequent Γ. We say that π *subsumes* π', in symbols $\pi' \sqsubset \pi$, if $\mathsf{N}^i_{\pi'} \sqsubset \mathsf{N}^i_\pi$ for every $i < |\Gamma|$. If π and π' each subsumes the other then π and π' are *equivalent*, in symbols $\pi \sim \pi'$. As an immediate consequence of the definition we have

Lemma 5.20. *Suppose π and π' are proofs of the same Σ_1 sequent. If $\pi' \sqsubset \pi$ then $L(\pi') \subseteq L(\pi)$.*

Herbrand schemes have the property that their languages are invariant under many basic proof transformations. The first example we give concerns the operation of substitution in proofs:

Lemma 5.21. *Suppose π and π' are proofs with the same end-sequent such that π' is the result of replacing a sub-proof π_0 of π by π'_0. If $\pi_0 \sqsubset \pi'_0$ then $\pi \sqsubset \pi'$.*

Proof. Let π, π_0, π' and π'_0 be as in the statement. We assume π and π' have the same end-sequent, say Γ. Given a subproof $\hat{\pi}$ of π which is not a proper subproof of π_0, let $\hat{\pi}'$ denote the corresponding subproof of π'. Observe that if $\hat{\pi}$ is a subproof of π but not a proper subproof of π_0 then the non-terminals $\mathsf{N}^j_{\hat{\pi}}$ and $\mathsf{N}^j_{\hat{\pi}'}$ are of the same type for each j. Fix $i < |\Gamma|$ and a normal term $t_0 = t(\mathsf{N}^i_\pi/x)$. Suppose $t_0 \to t_1 \to \cdots \to t_k = r$ is a derivation in $\mathscr{H}$ of a Σ-term r. By Lemma 5.10, t_i is normal for every $i \leq k$ and, without loss of generality, we may assume t does not feature any non-terminals labelled by proofs with π_0 as a sub-proof. Throughout this derivation, recursively replace each occurrence of a non-terminal $\mathsf{N}^j_{\hat{\pi}}$ for which $\hat{\pi}$ is not a proper subproof of π_0 by the non-terminal $\mathsf{N}^j_{\hat{\pi}'}$. Arguing by induction on k, using $\pi_0 \sqsubset \pi'_0$, we deduce $t(\mathsf{N}^i_{\pi'}/x) \to^* s$ for some Σ-term s with $s^\circ = r^\circ$. □

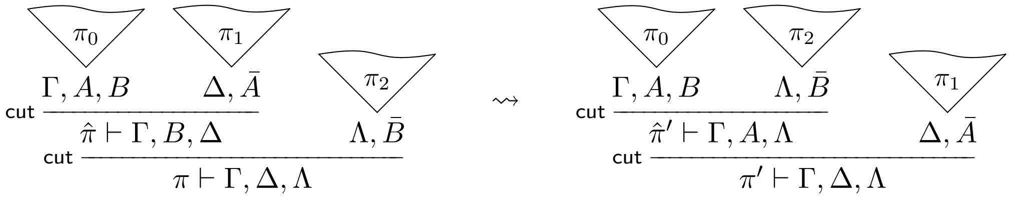
We now turn our attention to the analysis of the subsumption relation with respect to the cut reduction and permutation steps of Figures 2 and 3. Only the most interesting cases will be covered in detail: the cut and quantifier permutation, and contraction and quantifier reduction. As before, we leave instances of the permutation inference implicit and make use of Lemma 5.19 without reference. Recall the characterisation of the cut inference from Remark 3.25:

$$\mathsf{cut}\ \frac{\pi_0 \vdash \Gamma, A \quad \pi_1 \vdash \Delta, \bar{A}}{\pi \vdash \Gamma, \Delta} \qquad \mathsf{N}^i_\pi a\vec{x}\vec{y} \sim \begin{cases} \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x})), & \text{if } u(A) > 0, \\ \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}), & \text{if } e(A) > 0, \\ \mathsf{N}^i_{\pi_0} a\vec{x}\langle\rangle, & \text{if } A \text{ is q.f.,} \end{cases}$$

where $i < |\Gamma|$ and $\vec{x}$, $\vec{y}$ and a are terms of suitable type.

5.3.1 Cut Permutation

Suppose $\pi \rightsquigarrow \pi'$ are the two proofs



Due to the asymmetry in the production rules for cut, it is necessary to split the analysis of this reduction into two cases, depending on whether or not A and B are both universally quantified. Provided at least one of the two formulæ is existentially quantified or quantifier free, the two proofs above are equivalent and their languages are equal. This is proved in Lemma 5.22. If both A and B are universally quantified we do not expect equivalence to hold in general. However, if there are no contractions to the formula $\bar{A}$ in π_1 or the formula $\bar{B}$ in π_2, the proofs π and π' are equivalent. This is relevant to the cut reduction strategies employed in Theorem 1.1 and is treated in Lemma 5.25.

Lemma 5.22. *For $\pi \rightsquigarrow \pi'$ as above, if at least one of $u(A)$ and $u(B)$ is zero then $\pi \sim \pi'$.*

Proof. If one of A or B is quantifier-free the argument is straightforward following the production rules for cut. This leaves the following three cases to consider: $u(\bar{A}), u(B) >$

0, $u(A), u(\bar{B}) > 0$ and $u(\bar{A}), u(\bar{B}) > 0$. We consider only the first case as the second is symmetric and the third follows a simpler argument. Thus assume $e(A), u(B) > 0$.

Let $\vec{r}$, $\vec{s}$, $\vec{t}$ be sequences of terms of length $m = |\Gamma|$, $n = |\Delta|$ and $o = |\Lambda|$ respectively, and let $a : \varsigma$ be an arbitrary substitution stack. By the production rules for cut we have, for each $i \leq m$, $j < n$ and $k < o$, and each term w, w' of suitable type,

$$\mathsf{N}^{i}_{\hat{\pi}} a\vec{r}w\vec{s} \sim \begin{cases} \mathsf{N}^{i}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})w, & \text{if } i < m, \\ \mathsf{N}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})w, & \text{if } i = m, \end{cases} \qquad \mathsf{N}^{m+1+j}_{\hat{\pi}} a\vec{r}w\vec{s} \sim \mathsf{N}^{j}_{\pi_1} a\vec{s}(\mathsf{N}^{m}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})w)$$

$$\mathsf{N}^{i}_{\hat{\pi}'} a\vec{r}w'\vec{t} \sim \mathsf{N}^{i}_{\pi_0} a\vec{r}w'(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\mathsf{N}^{m+1}_{\pi_0} a\vec{r}w')) \qquad \mathsf{N}^{m+1+k}_{\hat{\pi}'} a\vec{r}w'\vec{t} \sim \mathsf{N}^{k}_{\pi_2} a\vec{t}(\mathsf{N}^{m+1}_{\pi_0} a\vec{r}w')$$

In particular,

$$\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s} \sim \mathsf{N}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s}) \tag{15}$$

and so, for $i \leq m$,

$$\mathsf{N}^{i}_{\hat{\pi}'} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})\vec{t} \sim \mathsf{N}^{i}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s})). \tag{16}$$

We prove $\mathsf{N}^{i}_{\pi} a\vec{r}\vec{s}\vec{t} \sim \mathsf{N}^{i}_{\pi'} a\vec{r}\vec{s}\vec{t}$ for every $i < m + n + o$, from which Lemma 5.14 implies $\mathsf{N}^{i}_{\pi} \sim \mathsf{N}^{i}_{\pi'}$. For $i < m$ we have

$$\begin{aligned} \mathsf{N}^{i}_{\pi} a\vec{r}\vec{s}\vec{t} \sim \mathsf{N}^{i}_{\hat{\pi}} a\vec{r}(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s}))\vec{s} &\sim \mathsf{N}^{i}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s})) \\ &\sim \mathsf{N}^{i}_{\hat{\pi}'} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})\vec{t} \\ &\sim \mathsf{N}^{i}_{\pi'} a\vec{r}\vec{s}\vec{t} \end{aligned}$$

by applying (16). For $j < n$,

$$\begin{aligned} \mathsf{N}^{m+j}_{\pi} a\vec{r}\vec{s}\vec{t} \sim \mathsf{N}^{m+1+j}_{\hat{\pi}} a\vec{r}(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s}))\vec{s} &\sim \mathsf{N}^{j}_{\pi_1} a\vec{s}(\mathsf{N}^{m}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})(\mathsf{N}^{o}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s}))) \\ &\sim \mathsf{N}^{j}_{\pi_1} a\vec{s}(\mathsf{N}^{m}_{\hat{\pi}'} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})\vec{t}) \\ &\sim \mathsf{N}^{m+j}_{\pi'} a\vec{r}\vec{s}\vec{t} \end{aligned}$$

again applying (16). For $k < o$, using (15):

$$\begin{aligned} \mathsf{N}^{m+n+k}_{\pi} a\vec{r}\vec{s}\vec{t} \sim \mathsf{N}^{k}_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s}) &\sim \mathsf{N}^{k}_{\pi_2} a\vec{t}(\mathsf{N}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})) \\ &\sim \mathsf{N}^{m+1+k}_{\hat{\pi}'} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s})\vec{t} \\ &\sim \mathsf{N}^{m+n+k}_{\pi'} a\vec{r}\vec{s}\vec{t}. \end{aligned}$$ □

As noted above, in the case $u(A)$ and $u(B)$ are both positive, language equality holds only in particular circumstances. A sufficient condition for this is given by the next lemma.

Lemma 5.23. *Let $\pi \rightsquigarrow \pi'$ be as above and assume $u(A), u(B) > 0$. Let $\rho = \tau^{*}_{\bar{A}}$, $\sigma = \tau^{*}_{\bar{B}}$, $R = \hat{\mathsf{N}}^{m}_{\hat{\pi}'} a\vec{r}\vec{t}$ and $S = \hat{\mathsf{N}}^{m}_{\hat{\pi}} a\vec{r}\vec{s}$. If*

$$R \sim \hat{\mathsf{N}}^{m}_{\pi_0} a\vec{r}(\mathsf{N}^{o}_{\pi_2} a\vec{t}S) \qquad \text{and} \qquad S \sim \hat{\mathsf{N}}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^{n}_{\pi_1} a\vec{s}R)$$

then $\pi \sim \pi'$.

Proof. Recall that R and S have type ρ and σ respectively. Then for $i < m$,

$$\begin{aligned}
\mathsf{N}^i_{\pi} a\vec{r}\vec{s}\vec{t} &\sim \mathsf{N}^i_{\hat{\pi}} a\vec{r}(\mathsf{N}^o_{\pi_2} a\vec{t}S)\vec{s} \\
&\sim \mathsf{N}^i_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}(\hat{\mathsf{N}}^m_{\pi_0} a\vec{r}(\mathsf{N}^o_{\pi_2} a\vec{t}S)))(\mathsf{N}^o_{\pi_2} a\vec{t}S) \\
&\sim \mathsf{N}^i_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}R)(\mathsf{N}^o_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m+1}_{\pi_0} a\vec{t}(\mathsf{N}^n_{\pi_1} a\vec{s}R))) \\
&\sim \mathsf{N}^i_{\hat{\pi}'} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}R)\vec{t} \\
&\sim \mathsf{N}^i_{\pi'} a\vec{r}\vec{s}\vec{t}
\end{aligned}$$

The other cases, namely $m \leq i < n + o$ follow similar reasoning. □

Lemma 5.24. *If $A, B \in \Pi_1 \cup \Sigma_1$ then $\pi \sim \pi'$.*

Proof. Assume $\bar{A}$ and $\bar{B}$ are both Σ_1 formulæ (if not, apply Lemma 5.22). Let R and S be as in Lemma 5.23. We have, by Lemma 5.17,

$$\begin{aligned}
Rw &\sim \mathsf{N}^m_{\pi_0} a\vec{r}w(\mathsf{N}^n_{\pi_2} a\vec{t}(\hat{\mathsf{N}}^{m+1}_{\pi_0} a\vec{r}w)) & \qquad Sw' &\sim \mathsf{N}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}(\hat{\mathsf{N}}^m_{\pi_0} a\vec{r}w'))w' \\
&\sim \mathsf{N}^m_{\pi_0} a\vec{r}w(\mathsf{N}^n_{\pi_2} a\vec{t}S) & &\sim \mathsf{N}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}R)w'
\end{aligned}$$

and hence

$$R \sim \hat{\mathsf{N}}^m_{\pi_0} a\vec{r}(\mathsf{N}^o_{\pi_2} a\vec{t}S) \qquad\qquad S \sim \hat{\mathsf{N}}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}R).$$

The previous lemma then implies $\pi \sim \pi'$. □

Lemma 5.25. *For the same π and π', if there are no contractions to either the formula $\bar{A}$ in the sub-proof π_1 or the formula $\bar{B}$ in the sub-proof π_2 then $\pi \sim \pi'$.*

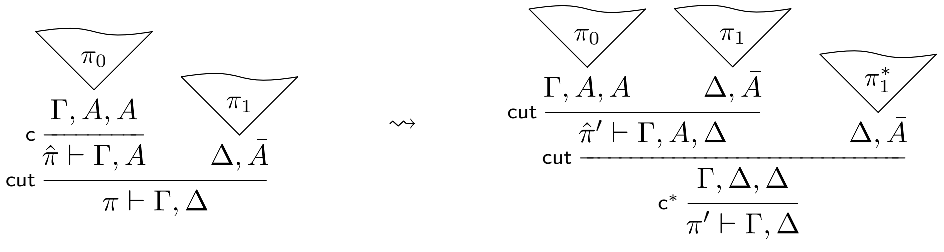
Proof. Suppose there are no contractions to $\bar{B}$ in π_2 and let R and S be as above. By Lemma 5.18, $\mathsf{N}^k_{\pi_2} a\vec{t}u \sim \mathsf{N}^k_{\pi_2} a\vec{t}v$ for any two terms $u, v : \tau^*_{\bar{B}}$. Hence, in particular,

$$R \sim \hat{\mathsf{N}}^m_{\pi_0} a\vec{r}(\mathsf{N}^o_{\pi_2} a\vec{t}S) \qquad\qquad \mathsf{N}^o_{\pi_2} a\vec{t}S \sim \hat{\mathsf{N}}^{m+1}_{\pi_0} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{s}R)$$

which suffice, by the proof of Lemma 5.23, to show $\pi \sim \pi'$. □

5.3.2 Contraction Reduction

Consider the two proofs



where π_1^* denotes a copy of π_1 with fresh eigenvariables. Observe that $\pi_1 \sim \pi_1^*$.

Although the reduction above does not in general induce language inclusion, for the two scenarios required in Theorem 1.1, namely either $u(A) = 0$ or there are no applications of contraction are applied to the formula $\bar{A}$ in the sub-proof π_1, we have $\pi' \sqsubset \pi$. The following two lemmas deal with these two cases.

Lemma 5.26. *If $u(A) = 0$ then $\pi' \sqsubset \pi$.*

Proof. If A is quantifier-free then $\pi \sim \pi'$ is easily established by following the reduction rules for cut. So assume $u(A) = 0 < u(\bar{A})$. Let $m = |\Gamma|$ and $n = |\Delta|$, and fix $i < m$ and $j < n$. Let $r = \mathsf{N}^n_{\pi_1} a\vec{y}$ and $r_* = \mathsf{N}^n_{\pi_1^*} a\vec{y}$. Unravelling the production rules for the two proofs yield

$$\begin{aligned} \mathsf{N}^i_{\pi} a\vec{x}\vec{y} &\sim \mathsf{N}^i_{\pi_0} a\vec{x}rr & \mathsf{N}^{m+j}_{\pi} a\vec{x}\vec{y} &\sqsupset \{\mathsf{N}^j_{\pi_1} a\vec{y}(\mathsf{N}^{m+1}_{\pi_0} a\vec{x}rr), \mathsf{N}^j_{\pi_1} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x}rr)\} \\ \mathsf{N}^i_{\pi'} a\vec{x}\vec{y} &\sim \mathsf{N}^i_{\pi_0} a\vec{x}r_*r & \mathsf{N}^{m+j}_{\pi'} a\vec{x}\vec{y} &\sim \{\mathsf{N}^j_{\pi_1} a\vec{y}(\mathsf{N}^{m+1}_{\pi_0} a\vec{x}r_*r), \mathsf{N}^j_{\pi_1^*} a\vec{y}(\mathsf{N}^m_{\pi_0} a\vec{x}r_*r)\} \end{aligned}$$

Since $\pi_1 \sim \pi_1^*$, Lemma 5.13 implies $\pi' \sqsubset \pi$. □

Lemma 5.27. *If $u(A) > 0$ and there are no contractions on the formula $\bar{A}$ in π_1, then $\pi' \sim \pi$.*

Proof. Suppose $u(A) > 0$. Let $\tau = \tau^*_{\bar{A}}$. Lemma 5.18 implies $\mathsf{N}^j_{\pi_1} a\vec{y}s \sim \mathsf{N}^j_{\pi_1} a\vec{y}\mathsf{c}_\tau$ for every $s : \tau$. Concerning derivations from π, this yields the following equivalences for $i < |\Gamma|$ and $j < |\Delta|$.

$$\begin{aligned} \mathsf{N}^i_{\pi} a\vec{x}\vec{y} &\sim \mathsf{N}^i_{\hat{\pi}} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}(\hat{\mathsf{N}}^m_{\hat{\pi}} a\vec{x})) & \mathsf{N}^{m+j}_{\pi} a\vec{x}\vec{y} &\sim \mathsf{N}^j_{\pi_1} a\vec{y}(\hat{\mathsf{N}}^m_{\hat{\pi}} a\vec{x}) \\ &\sim \mathsf{N}^i_{\hat{\pi}} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}\mathsf{c}_\tau) & &\sim \mathsf{N}^j_{\pi_1} a\vec{y}\mathsf{c}_\tau \\ &\sim \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1} a\vec{y}\mathsf{c}_\tau)(\mathsf{N}^n_{\pi_1} a\vec{y}\mathsf{c}_\tau) \end{aligned}$$

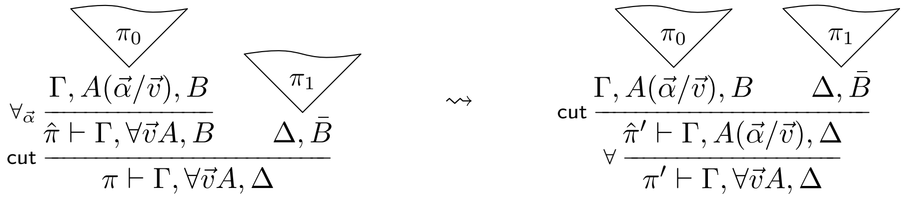
Starting from π' we obtain

$$\begin{aligned} \mathsf{N}^i_{\pi'} a\vec{x}\vec{y} &\sim \mathsf{N}^i_{\hat{\pi}'} a\vec{x}(\mathsf{N}^n_{\pi_1^*} a\vec{y}\left(\hat{\mathsf{N}}^m_{\hat{\pi}'} a\vec{x}\vec{y}\right))\vec{y} & \mathsf{N}^{m+j}_{\pi'} a\vec{x}\vec{y} &\sim \{\mathsf{N}^{m+1+j}_{\hat{\pi}'} a\vec{x}(\mathsf{N}^n_{\pi_1^*} a\vec{y}\mathsf{c}_\tau)\vec{y}, \mathsf{N}^j_{\pi_1^*} a\vec{y}\mathsf{c}_\tau\} \\ &\sim \mathsf{N}^i_{\hat{\pi}'} a\vec{x}(\mathsf{N}^n_{\pi_1^*} a\vec{y}\mathsf{c}_\tau)\vec{y} & &\sim \{\mathsf{N}^j_{\pi_1} a\vec{y}\mathsf{c}_\tau, \mathsf{N}^j_{\pi_1^*} a\vec{y}\mathsf{c}_\tau\} \\ &\sim \mathsf{N}^i_{\pi_0} a\vec{x}(\mathsf{N}^n_{\pi_1^*} a\vec{y}\mathsf{c}_\tau)(\mathsf{N}^n_{\pi_1} a\vec{y}\mathsf{c}_\tau) \end{aligned}$$

So $\pi' \sim \pi$. □

5.3.3 Quantifier Permutation

Concerning permuting quantifier rules with cut, consider the following two proofs.


π_0
π_1
$\Gamma, A(\vec{\alpha}/\vec{v}), B$
$\forall_{\vec{\alpha}}$
$\hat{\pi} \vdash \Gamma, \forall\vec{v}A, B$
$\Delta, \bar{B}$
cut
$\pi \vdash \Gamma, \forall\vec{v}A, \Delta$
$\rightsquigarrow$
π_0
π_1
$\Gamma, A(\vec{\alpha}/\vec{v}), B$
$\Delta, \bar{B}$
cut
$\hat{\pi}' \vdash \Gamma, A(\vec{\alpha}/\vec{v}), \Delta$
$\forall$
$\pi' \vdash \Gamma, \forall\vec{v}A, \Delta$

(17)

Let $\vec{\alpha} = (\alpha_j)_{j \le p}$ and $\vec{v} = (v_j)_{j \le p}$. Regularity ensures that $u(A) = 0$. In the following, if $\vec{u} = (u_j)_{j \le p}$ is a sequence of terms of type ι and $u_{p+1} : \tau^*_A$, we write $\vec{u}\star u_{p+1}$ to abbreviate the sequence term $\langle u_0, \ldots, u_{p+1} \rangle : \tau^*_{\forall\vec{v}A}$.

Like with the case of permuting cuts, an application of the quantifier permutation reduction does not preserve equivalence of proofs in all cases. For the main theorem it suffice to prove only $\pi' \sqsubset \pi$. This is taken up in Lemma 5.29 below. First, however, we show that if B is not universally quantified then indeed $\pi \sim \pi'$.

Lemma 5.28. *For π and π' above, if $u(B) = 0$ then $\pi' \sim \pi$.*

Proof. Suppose $u(B) = 0$ and B is not quantifier-free. The other cases involve much similar arguments. Fix $\vec{r}$ and $\vec{s}$ sequences of normal terms of length $m = |\Gamma|$ and $n = |\Delta|$ respectively, and normal terms t and $\vec{u} \star u'$ of type $\tau^*_{\forall \vec{v} A}$. By regularity of π and Lemma 5.16,

$$\mathsf{N}^j_{\pi_1}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{s} \sim \mathsf{N}^j_{\pi_1} a\vec{s}$$

for each $j \leq n$. Concerning π the following equivalences therefore appear for $i \leq m$, $j < n$ and $k \leq m+1$,

$$\begin{aligned} \mathsf{N}^i_{\pi} a\vec{r}t\vec{s} &\sim \mathsf{N}^i_{\hat{\pi}} a\vec{r}t(\mathsf{N}^n_{\pi_1} a\vec{s}) & \qquad \mathsf{N}^k_{\hat{\pi}} a\vec{r}(\vec{u} \star u') &\sim \mathsf{N}^k_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{r}u' \\ \mathsf{N}^{m+1+j}_{\pi} a\vec{r}t\vec{s} &\sim \mathsf{N}^j_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\hat{\pi}} a\vec{r}t(\mathsf{N}^n_{\pi_1} a\vec{s})) \end{aligned}$$

So, if t is a normal term and $t \sim \{\vec{u}_0 \star u'_0, \ldots, \vec{u}_l \star u'_l\}$ is given by Lemma 5.7 then for $i \leq m$ and $j < n$,

$$\begin{aligned} \mathsf{N}^i_{\pi} a\vec{r}t\vec{s} &\sim \{\mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k(\mathsf{N}^n_{\pi_1} a\vec{s}) \mid k \leq l\} \\ \mathsf{N}^{m+1+j}_{\pi} a\vec{r}t\vec{s} &\sim \{\mathsf{N}^j_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k(\mathsf{N}^n_{\pi_1} a\vec{s})) \mid k \leq l\} \end{aligned}$$

Examining π', we observe

$$\begin{aligned} \mathsf{N}^i_{\pi'} a\vec{r}t\vec{s} &\sim \{\mathsf{N}^i_{\hat{\pi}'}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k\vec{s} \mid k \leq l\} \\ &\sim \{\mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k(\mathsf{N}^n_{\pi_1} a\vec{s}) \mid k \leq l\} \\ \mathsf{N}^{n+j}_{\pi'} a\vec{r}t\vec{s} &\sim \{\mathsf{N}^j_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k(\mathsf{N}^n_{\pi_1} a\vec{s})) \mid k \leq l\} \end{aligned}$$

Hence $\mathsf{N}^i_{\pi'} \sim \mathsf{N}^i_{\pi}$ for every $i \leq m+n$ and so $\pi' \sim \pi$. □

Lemma 5.29. *For π and π' as in* (17)*, $\pi' \sqsubset \pi$.*

Proof. Fix $\vec{r}$ and $\vec{s}$ sequences of terms of length $m = |\Gamma|$ and $n = |\Delta|$ respectively. Let $\vec{u} \star u' : \tau^*_{\forall \vec{v} A}$. Suppose $u(B) > 0$ and $i \leq m$. The other cases have been considered earlier or involve similar but simpler arguments. As was observed earlier,

$$\mathsf{N}^j_{\pi_1}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{s} \sim \mathsf{N}^j_{\pi_1} a\vec{s}$$

for each $j \leq n$. With respect to π' the following equivalences therefore appear.

$$\begin{aligned} \mathsf{N}^i_{\pi'} a\vec{r}(\vec{u} \star u')\vec{s} &\sim \mathsf{N}^i_{\hat{\pi}'}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{r}u'\vec{s} \\ &\sim \mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{r}u'(\mathsf{N}^n_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{r}u')) \end{aligned}$$

whereas the rules for π yield, for arbitrary $t : \tau^*_{\forall \vec{v} A}$,

$$\mathsf{N}^i_{\pi} a\vec{r}t\vec{s} \sim \mathsf{N}^i_{\hat{\pi}} a\vec{r}t(\mathsf{N}^n_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\hat{\pi}} a\vec{r}t)) \qquad \mathsf{N}^i_{\hat{\pi}} a\vec{r}(\vec{u} \star u') \sim \mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}]a)\vec{r}u'$$

If t is a normal term and $t \sim \{\vec{u}_0 \star u'_0, \ldots, \vec{u}_l \star u'_l\}$ is given by Lemma 5.7 then for each $i \leq m$,

$$\begin{aligned} \mathsf{N}^i_{\hat{\pi}} a\vec{r}t &\sim \{\mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k \mid k \leq l\} \\ \mathsf{N}^i_{\pi} a\vec{r}t\vec{s} &\sim \{\mathsf{N}^i_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_k]a)\vec{r}u'_k(\mathsf{N}^n_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\pi_0}([\vec{\alpha} \mapsfrom \vec{u}_j]a)\vec{r}u'_j)) \mid k, j \leq l\} \end{aligned} \qquad (18)$$

whereas, due to pattern-matching in the production rule for π',

$$\mathsf{N}^i_{\pi'} a\vec{r}\vec{t}\vec{s} \sim \{\mathsf{N}^i_{\pi_0}([\vec{\alpha} \leftarrow\!\shortmid\, \vec{u}_k]a)\vec{r}\vec{u}'_k(\mathsf{N}^n_{\pi_1} a\vec{s}(\mathsf{N}^{m+1}_{\pi_0}([\vec{\alpha} \leftarrow\!\shortmid\, \vec{u}_k]a)\vec{r}\vec{u}'_k)) \mid k \leq l\} \tag{19}$$

Hence $\mathsf{N}^i_{\pi'} \sqsubset \mathsf{N}^i_{\pi}$ and $\pi' \sqsubset \pi$. □

The contrast between equations (18) and (19) demonstrates why $\pi \sqsubset \pi'$ need not hold in general.

5.3.4 Quantifier Reduction

Consider the reduction

$$\mathsf{cut}\,\dfrac{\forall_{\vec{\alpha}}\,\dfrac{\overset{\pi_0}{\Gamma, A(\vec{\alpha}/\vec{v})}}{\hat{\pi}_0 \vdash \Gamma, \forall\vec{v}A} \qquad \exists_{\vec{s}}\,\dfrac{\overset{\pi_1}{\Delta, \bar{A}(\vec{s}/\vec{v})}}{\hat{\pi}_1 \vdash \Delta, \exists\vec{v}\bar{A}}}{\pi \vdash \Gamma, \Delta} \quad \rightsquigarrow \quad \mathsf{cut}\,\dfrac{\overset{\pi_0^{(\vec{s}/\vec{\alpha})}}{\Gamma, A(\vec{s}/\vec{v})} \qquad \overset{\pi_1}{\Delta, \bar{A}(\vec{s}/\vec{v})}}{\pi' \vdash \Gamma, \Delta} \tag{20}$$

Lemma 5.30. *If $\pi \rightsquigarrow \pi'$ is the reduction above then $\pi \sim \pi'$.*

Proof. Let $m = |\Gamma|$, $n = |\Delta|$, $\vec{\alpha} = (\alpha_i)_{i \leq p}$ and $\vec{s} = (s_i)_{i \leq p}$. Recall that $\vec{s} \cdot a = (s_i \cdot a)_{i \leq p}$. Note that regularity of π implies $u(A) = 0$. This leaves two cases to consider: A is quantifier-free or $e(A) > 0$. Suppose the latter, so the cut in π' remains a quantified cut (the case A is q.f. follows an analogous argument). The following equivalences arise, where $i < m$ and $j < n$.

$$\begin{aligned}
\mathsf{N}^i_{\pi} a\vec{r}\vec{t} &\sim \mathsf{N}^i_{\hat{\pi}_0} a\vec{r}(\mathsf{N}^n_{\hat{\pi}_1} a\vec{t}(\mathsf{N}^m_{\hat{\pi}_0} a\vec{r})) \\
&\sim \mathsf{N}^i_{\hat{\pi}_0} a\vec{r}(\vec{s} \cdot a \star \mathsf{N}^n_{\hat{\pi}_1} a\vec{t}) \\
&\sim \mathsf{N}^i_{\pi_0}([\vec{\alpha} \leftarrow\!\shortmid\, \vec{s} \cdot a]a)\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{t}) \\
&\sim \mathsf{N}^i_{\pi_0^{(\vec{s}/\vec{\alpha})}} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{t}) \\
&\sim \mathsf{N}^i_{\pi'} a\vec{r}\vec{t}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{N}^{m+j}_{\pi} a\vec{r}\vec{t} &\sim \mathsf{N}^j_{\hat{\pi}_1} a\vec{t}(\mathsf{N}^m_{\hat{\pi}_0} a\vec{r}) \\
&\sim \mathsf{N}^j_{\pi_1} a\vec{t}(\mathsf{N}^m_{\hat{\pi}_0} a\vec{r}(\vec{s} \cdot a \star \mathsf{N}^n_{\pi_1} a\vec{t})) \\
&\sim \mathsf{N}^j_{\pi_1} a\vec{t}(\mathsf{N}^m_{\pi_0}([\vec{\alpha} \leftarrow\!\shortmid\, \vec{s} \cdot a]a)\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{t})) \\
&\sim \mathsf{N}^j_{\pi_1} a\vec{t}(\mathsf{N}^m_{\pi_0^{(\vec{s}/\vec{\alpha})}} a\vec{r}(\mathsf{N}^n_{\pi_1} a\vec{t})) \\
&\sim \mathsf{N}^{m+j}_{\pi'} a\vec{r}\vec{t}.
\end{aligned}$$

The penultimate equivalence in each column is given by Lemma 5.16. □

5.3.5 Remaining Reductions

The remaining rules are all straightforward to analyse and all induce language equality with the exception of weakening reduction for which only language inclusion holds in general.

5.4 Proof of Main Theorem

We can now prove Theorem 1.1. Let $\pi \vdash \exists\vec{v}F_{qf}$ be a regular proof and $\pi = \pi_0 \rightsquigarrow \pi_1 \rightsquigarrow \cdots \rightsquigarrow \pi_n$ be a reduction of π to a quasi cut-free proof π_n such that for each $i < n$, the reduction $\pi_i \rightsquigarrow \pi_{i+1}$ applies a cut reduction or permutation rule from Figures 2 or 3 to a sub-proof of π_i with the restriction that a rule reducing the strong quantifier side of a cut is applied only if no other reduction of this cut is possible. By Lemma 5.21 and the analysis in the previous section, $L(\pi_{i+1}) \subseteq L(\pi_i)$ for each $i < n$. This together with Lemma 3.31 establishes part (iii) of the theorem. The existence of a reduction of the form above is well-known: see, e.g. [42], hence (i). Acyclicity of $\mathscr{H}_\pi$ is shown in Lemma 3.26, the bound on the order of $\mathscr{H}_\pi$ is given by Corollary 3.34, and the language bound in (ii) follows from Theorem 3.35.

6 Discussion

This work contributes to the structural analysis of first-order proofs with respect to their Herbrand content. To a first-order classical proof $\pi \vdash F$ of a Σ_1 formula we associate a recursion scheme $\mathscr{H}$ with a finite language that constitutes a Herbrand set for F. More generally, the language of $\mathscr{H}$ covers the Herbrand set implicit in any quasi cut-free proof obtained from π by a sequence of reductions fulfilling the following two restrictions.

1. A contraction on a universally quantified formula is reduced only when no other reduction rule is applicable to this cut;
2. If two cuts are permuted in the following form then either there are no contractions on the formula $\bar{B}$ in the relevant subproof, or one of A and B is not universally quantified.

$$\mathsf{cut}\,\dfrac{\mathsf{cut}\,\dfrac{\Gamma, A, B \quad \Delta, \bar{A}}{\Gamma, \Delta, B} \qquad \Lambda, \bar{B}}{\Gamma, \Delta, \Lambda} \quad \rightsquigarrow \quad \mathsf{cut}\,\dfrac{\mathsf{cut}\,\dfrac{\Gamma, A, B \quad \Lambda, \bar{B}}{\Gamma, \Lambda, A} \qquad \Delta, \bar{A}}{\Gamma, \Delta, \Lambda}$$

The size of the Herbrand set is bounded by $2^{4|\pi|^3}_{n+2}$ where $|\pi|$ is the number of inferences in π and n is the maximal quantifier rank of a cut in π. Comparing with related work, the bound on the cardinality of the Herbrand expansion obtained by Gerhardy and Kohlenbach [20, Corollary 15] is $2^{3\|t\|}_{\mathrm{dg}(\phi)+1}$ where ϕ is a proof in Shoenfield's calculus [41], t is the realiser extracted from ϕ, and $\|t\|$ is the number of symbols in t. The degree $\mathrm{dg}(\phi)$ is the maximal $\neg$-depth of a cut formula in ϕ. The $\neg$-depth of a formula is defined precisely in the discussion on pp. 17–25 of [19] as the maximal number of nested negations over quantifier-free sub-formulæ (that may contain an arbitrary number of negations). This is sufficient for describing the height of the tower of exponentials since, in Shoenfield's system, $\exists x$ is considered an abbreviation of $\neg\forall x \neg$. Thus (the translation of) a $\Pi_n \cup \Sigma_n$ formula has $\neg$-depth at most n. Presumably it is possible to give a polynomial translation from the sequent calculus into Shoenfield's system which preserves the maximal $\neg$-depth of cut formulæ (but, to the knowledge of the authors, this has not been done in the literature) and, moreover, to bound $\|t\|$ polynomially in terms of the number of inferences of ϕ. Under these assumptions, the bound of Gerhardy and Kohlenbach would yield the upper bound $2^{p(|\pi|)}_{n+1}$ on the cardinality of a Herbrand expansion for some polynomial p and any sequent calculus proof π with $\Pi_n \cup \Sigma_n$-cuts. This would be one exponent less than our own.

Closely related is a bound obtained by Buss in [11]. The proof of Theorem 9 of [11] shows that, given a proof π where all cut formulæ are contained in $\Pi_n \cup \Sigma_n$, there is a cut-free proof whose number of inferences is no greater than $2^{|\pi|}_{n+2}$. As an immediate corollary this also yields the upper bound of $2^{|\pi|}_{n+2}$ on the cardinality of the Herbrand expansion. If one is interested in the cardinality of the Herbrand expansion, Buss's bound and our Theorem 1.1 give the same number of iterations of the exponential function, but Gerhardy and Kohlenbach's would give one less. If one is interested in the number of inferences in the cut-free proof, Buss's bound is one exponential better than ours but has the same number of exponentials as the one that could be obtained from Gerhardy and Kohlenbach's since the number of inferences is at most exponential in the cardinality of the Herbrand expansion (considering the symbolic complexity of the end-sequent is constant). That being said, the bounds we obtain apply to any cut-free proof (and Herbrand disjunction) that can be reached by the class of reductions pertaining to 1 and 2 above. In particular, it places no restriction on which cut is to be reduced at any given step, and therefore accommodates a variety of strategies, including top-most and maximal cut-complexity. Whether this freedom of strategies necessitates the larger bound is not entirely clear, and requires further investigation.

Our approach provides a framework that appears well-suited for extensions. Below we highlight finer features of our representation of Herbrand's theorem and some potential applications.

Sequent-based versus trace-based grammars In this paper, the grammar associated to a proof is 'sequent'-based in the following sense. Consider an inference of the form

$$\mathsf{r}\,\frac{G_0, \ldots, G_n}{F_0, \ldots, F_m}$$

The production rules corresponding to r can be seen as transforming a sequence of inputs $(x_0, \ldots, x_m)$ for the formulæ $F_0, \ldots, F_m$ to a sequence of terms $(t_0, \ldots, t_n)$ which are used as inputs for $G_0, \ldots, G_n$ in the inference rule immediately above r. The production rules effect the whole sequence of inputs regardless of which formula is active. This is in contrast with the 'trace'-based grammars of, e.g., [24, 1, 2] where an inference of the form

$$\mathsf{r}\,\frac{\Gamma, G}{\Gamma, F}$$

is associated a production rule that updates an input for F to an input for G, entirely ignoring presence of formulæ in Γ. In the latter type of grammars the derivations can be viewed as traces that climb up and also down the proof tree essentially mimicking the traces revealed through Gentzen-style cut-elimination. These grammars are generally cyclic and it is necessary to place equality constraints (the 'rigidity' conditions of [24, 1]) on derivations to ensure finite languages. For proofs that contain cuts with complexity greater than Π_2/Σ_2 the trace-based analysis quickly becomes infeasible. In contrast, the sequent-based approach generates an acyclic term grammar that not only ensures a finite language but allows one to obtain upper bounds on language size by standard language-theoretic arguments.

Providing a minimal grammar As mentioned in the introduction, part of the motivation behind this study is to ultimately invert the cut-elimination procedure and find an algorithmic method for introducing cuts into cut-free proofs. The idea has been successfully carried out for Π_1/Σ_1-cut introduction and more recently for the introduction of a single Π_2/Σ_2-cut (see [26, 25, 27, 32]). The general method proceeds as follows. Given a cut-free proof π, one first computes a concise representation of π as a term grammar (such as a regular tree grammar whose language contains the Herbrand set induced by π). This grammar is then viewed as a proof with cut, in which the cut-formulæ are yet to be determined. Finding the cut-formulæ involves solving a unification problem induced by the grammar. Key to successfully carrying out this procedure is identifying natural classes of formal grammars that describe the instantiation structure of a proof with cut. Higher order recursion schemes provide a promising candidate to lift the method of cut-introduction above the Π_2 level.

First-order logic in finite types A natural extension to consider is first-order logic in finite (simple) types, namely many-sorted predicate logic with a sort of individuals for each simple type and well-typed application as a term forming operation. On the sequent calculus side, we add new quantifier inferences for each type:

$$\forall^{\sigma}_{\alpha}\,\frac{\Gamma, A(\alpha^{\sigma}/v^{\sigma})}{\Gamma, \forall v^{\sigma} A} \qquad\qquad \exists^{\sigma}_{r}\,\frac{\Gamma, A(r^{\sigma}/v^{\sigma})}{\Gamma, \exists v^{\sigma} A}$$

Here one can use the type hierarchy underpinning higher order recursion schemes. The formula types τ_A and τ^*_A are extended to incorporate higher-type quantification (for example $\tau_{\exists v^{\sigma} F} = \sigma \times \tau_F$ if $u(F) = 0$). The corresponding production rules for the new quantifier inferences will be identical to the rules for the ground type, though the move to higher-type means that substitution stacks can contain substitutions of higher-type symbols. We expect the analogous language preservation lemmas to hold. Moreover, Herbrand schemes for higher-type logic may provide a direct way to study the relation between the present work and the approach via functional interpretation in [20].

Lifting the prenex restriction Our representation of first-order proofs as recursion schemes forces an asymmetric interpretation of formulæ to types that does not easily generalise to non-prenex cuts. Specifically, the type of an existentially quantified formula is (except in the case of Σ_1) an order higher than the dual universally quantified formula. This disparity is due to the production rules for cut which treat the cut formula from one premise as a function which receives as its input 'witnesses' for the dual (cut) formula in the other premise. If the same representation is applied to non-prenex cuts then we must have that the type associated to a disjunction is an order higher than that assigned to the dual conjunction. Writing sound production rules for the disjunction and conjunction introduction inferences then becomes non-trivial.

One possible remedy is to switch to a two-sided sequent calculus. The order distinction forced by the production rules for cut becomes a distinction between the two sides of the sequent arrow thus permitting a more uniform association of formulæ to types that may permit a generalisation to cuts of any form. The hurdle here, however, will be in a satisfactory interpretation of negation which requires further investigation.

7 References

[1] Bahareh Afshari, Stefan Hetzl and Graham E. Leigh. ‘Herbrand disjunctions, cut elimination and context-free tree grammars’. In: *13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015).* Ed. by Thorsten Altenkirch. Vol. 38. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 1–16.

[2] Bahareh Afshari, Stefan Hetzl and Graham E. Leigh. ‘Herbrand confluence for first-order proofs with Π_2-cuts’. In: *Proof: Concepts of Proof in Mathematics, Philosophy, and Computer Science.* Vol. 6. Ontos Mathematical Logic. De Gruyter, 2016, pp. 4–50.

[3] Bahareh Afshari, Stefan Hetzl and Graham E. Leigh. ‘Herbrand’s theorem revisited’. In: *Proceedings of Applied Mathematics and Mechanics* 16.1 (2016), pp. 905–906.

[4] Bahareh Afshari, Stefan Hetzl and Graham E. Leigh. ‘On the Herbrand content of LK’. In: *Proceedings of Sixth International Workshop on Classical Logic and Computation (CL&C 2016).* Ed. by Ulrich Kohlenbach, Steffen van Bakel and Stefano Berardi. Vol. 213. EPTCS. 2016, pp. 1–10.

[5] Jeremy Avigad and Solomon Feferman. ‘Gödel’s functional (“Dialectica”) interpretation’. In: *The Handbook of Proof Theory.* Ed. by Sam Buss. North-Holland, 1999, pp. 337–405.

[6] Matthias Baaz and Stefan Hetzl. ‘On the non-confluence of cut-elimination’. In: *Journal of Symbolic Logic* 76.1 (2011), pp. 313–340.

[7] Matthias Baaz, Stefan Hetzl et al. ‘Cut-elimination: Experiments with CERES’. In: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004).* Ed. by Franz Baader and Andrei Voronkov. Vol. 3452. Lecture Notes in Computer Science. Springer, 2005, pp. 481–495.

[8] Matthias Baaz and Alexander Leitsch. ‘Cut-elimination and redundancy-elimination by resolution’. In: *Journal of Symbolic Computation* 29.2 (2000), pp. 149–176.

[9] Franco Barbanera, Stefano Berardi and Massimo Schivalocchi. ‘“Classical” programming-with-proofs in λ_{PA}^{Sym}: An analysis of non-confluence’. In: *Theoretical Aspects of Computer Software.* Ed. by Martín Abadi and Takayasu Ito. Vol. 1281. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, pp. 365–390.

[10] Arnold Beckmann. ‘Exact bounds for lengths of reductions in typed lambda-calculus.’ In: *Journal of Symbolic Logic* 66.3 (2001), pp. 1277–1285.

[11] Sam Buss. ‘Cut Elimination in Situ’. In: *Gentzen’s Centenary: The Quest for Consistency.* Ed. by R. Kahle and M. Rathjen. Springer, 2015, pp. 245–277.

[12] Thierry Coquand. ‘A semantics of evidence for classical arithmetic’. In: *Journal of Symbolic Logic* 60.1 (1995), pp. 325–337.

[13] Sebastian Eberhard and Stefan Hetzl. ‘On the compressibility of finite languages and formal proofs’. To appear in *Information and Computation.*

[14] Sebastian Eberhard and Stefan Hetzl. 'Compressibility of finite languages by grammars'. In: *Descriptional Complexity of Formal Systems (DCFS 2015).* Ed. by Jeffrey Shallit and Alexander Okhotin. Vol. 9118. Lecture Notes in Computer Science. Springer, 2015, pp. 93–104.

[15] Sebastian Eberhard and Stefan Hetzl. 'Inductive theorem proving based on tree grammars'. In: *Annals of Pure and Applied Logic* 166.6 (2015), pp. 665–700.

[16] Gabriel Ebner, Stefan Hetzl et al. 'System Description: GAPT 2.0'. In: *8th International Joint Conference on Automated Reasoning (IJCAR 2016).* Ed. by Nicola Olivetti and Ashish Tiwari. Vol. 9706. Lecture Notes in Computer Science. Springer, 2016, pp. 293–301.

[17] Gabriel Ebner, Stefan Hetzl et al. 'On the generation of quantified lemmas'. Submitted. 2018.

[18] Gerhard Gentzen. 'Untersuchungen über das logische Schließen II'. In: *Mathematische Zeitschrift* 39.3 (1935), pp. 405–431.

[19] Philipp Gerhardy. 'Applications of Proof Interpretations'. PhD thesis. University of Aarhus, 2006.

[20] Philipp Gerhardy and Ulrich Kohlenbach. 'Extracting Herbrand disjunctions by functional interpretation'. In: *Archive for Mathematical Logic* 44 (2005), pp. 633–644.

[21] Kurt Gödel. 'Über eine noch nicht benützte Erweiterung des finiten Standpunktes'. In: *Dialectica* 12 (1958), pp. 280–287.

[22] Willem Heijltjes. 'Classical proof forestry'. In: *Annals of Pure and Applied Logic* 161.11 (2010), pp. 1346–1366.

[23] Hugo Herbelin. 'Séquents qu'on calcule'. PhD thesis. Université Paris 7, 1995.

[24] Stefan Hetzl. 'Applying tree languages in proof theory'. In: *Language and Automata Theory and Applications (LATA 2012).* Ed. by Adrian-Horia Dediu and Carlos Martín-Vide. Vol. 7183. Lecture Notes in Computer Science. Springer, 2012, pp. 301–312.

[25] Stefan Hetzl, Alexander Leitsch, Giselle Reis and Daniel Weller. 'Algorithmic introduction of quantified cuts'. In: *Theoretical Computer Science* 549 (2014), pp. 1–16.

[26] Stefan Hetzl, Alexander Leitsch and Daniel Weller. 'Towards algorithmic cut-introduction'. In: *Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2012).* Ed. by Nikolaj Bjørner and Andrei Voronkov. Vol. 7180. Lecture Notes in Computer Science. Springer, 2012, pp. 228–242.

[27] Stefan Hetzl, Alexander Leitsch et al. 'Introducing quantified cuts in logic with equality'. In: *7th International Joint Conference on Automated Reasoning (IJCAR 2014).* Ed. by Stéphane Demri, Deepak Kapur and Christoph Weidenbach. Vol. 8562. Lecture Notes in Computer Science. Springer, 2014, pp. 240–254.

[28] Stefan Hetzl and Lutz Straßburger. 'Herbrand-confluence for cut-elimination in classical first-order logic'. In: *Computer Science Logic (CSL 2012).* Ed. by Patrick Cégielski and Arnaud Durand. Vol. 16. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 320–334.

[29] Stefan Hetzl and Lutz Straßburger. 'Herbrand-Confluence'. In: *Logical Methods in Computer Science* 9.4 (2013).

[30] Stefan Hetzl and Daniel Weller. 'Expansion trees with cut'. Preprint available at `http://arxiv.org/abs/1308.0428`. 2013.

[31] David Hilbert and Paul Bernays. *Grundlagen der Mathematik II.* Springer, 1939.

[32] Alexander Leitsch and Michael Peter Lettmann. 'The problem of Π_2-cut-introduction'. In: *Theoretical Computer Science* 706 (2018), pp. 83–116.

[33] Richard McKinley. 'Proof nets for Herbrand's theorem'. In: *ACM Transactions on Computational Logic* 14.1 (2013), 5:1–5:31.

[34] Dale Miller. 'A compact representation of proofs'. In: *Studia Logica* 46.4 (1987), pp. 347–370.

[35] Georg Moser and Richard Zach. 'The epsilon calculus and Herbrand complexity'. In: *Studia Logica* 82.1 (2006), pp. 133–155.

[36] C.-H. Luke Ong and Steven J. Ramsay. 'Verifying higher-order functional programs with pattern-matching algebraic data types'. In: *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2011).* ACM, 2011, pp. 587–598.

[37] Luke Ong. 'Recursion schemes, collapsible pushdown automata and higher-order model checking'. In: *Language and Automata Theory and Applications (LATA 2013).* Ed. by Adrian-Horia Dediu, Carlos Martín-Vide and Bianca Truthe. Springer, 2013, pp. 13–41.

[38] Diana Ratiu and Trifon Trifonov. 'Exploring the computational content of the infinite pigeonhole principle'. In: *Journal of Logic and Computation* 22.2 (2012), pp. 329–350.

[39] Helmut Schwichtenberg. 'Complexity of normalization in the pure typed lambda-calculus'. In: *The L.E.J.Brouwer Centenary Symposium.* Ed. by Anne S. Troelstra and D. van Dalen. North-Holland, 1982, pp. 453–457.

[40] Monika Seisenberger. 'On the constructive content of proofs'. PhD thesis. Ludwig-Maximilians-Universität München, 2003.

[41] Joseph R. Shoenfield. *Mathematical Logic.* 2nd edition. AK Peters, 2001.

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

[43] Christian Urban. 'Classical logic and computation'. PhD thesis. University of Cambridge, 2000.