Basic Research in Computer Science

BRICS RS-03-32 Gerhardy & Kohlenbach: Extracting Herbrand Disjunctions by Functional Interpretation

Extracting Herbrand Disjunctions by Functional Interpretation

Philipp Gerhardy
Ulrich Kohlenbach

BRICS Report Series **RS-03-32**

ISSN 0909-0878 **October 2003**

**See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:**

**BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk**

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/03/32/`

Extracting Herbrand Disjunctions by Functional Interpretation

Philipp Gerhardy Ulrich Kohlenbach*

October 20, 2003

Abstract

Carrying out a suggestion by Kreisel, we adapt Gödel's functional interpretation to ordinary first-order predicate logic(PL) and thus devise an algorithm to extract Herbrand terms from PL-proofs. The extraction is carried out in an extension of PL to higher types. The algorithm consists of two main steps: first we extract a functional realizer, next we compute the β-normal-form of the realizer from which the Herbrand terms can be read off. Even though the extraction is carried out in the extended language, the terms are ordinary PL-terms. In contrast to approaches to Herbrand's theorem based on cut elimination or ε-elimination this extraction technique is, except for the normalization step, of low polynomial complexity, fully modular and furthermore allows an analysis of the structure of the Herbrand terms, in the spirit of Kreisel ([13]), already prior to the normalization step. It is expected that the implementation of functional interpretation in Schwichtenberg's MINLOG system can be adapted to yield an efficient Herbrand-term extraction tool.

*Ulrich Kohlenbach partially supported by the Danish Natural Science Research Council, Grant no. 21-02-0474.

1 Introduction

Herbrand's theorem states that for every proof in pure first-order logic without equality of a sentence $\exists x A_{qf}(x)$ (A_{qf} always denotes a quantifier-free formula), there is a collection of closed terms $t_1, \ldots, t_n$ witnessing that proof, so that $\bigvee_{i=1}^{n} A_{qf}(t_i)$ is a tautology. Such a disjunction is called a Herbrand disjunction of A and the terms $t_1, \ldots, t_n$ are called Herbrand terms. Herbrand's theorem easily generalizes to tuples of existential quantifiers $\exists \underline{x} A_{qf}(\underline{x})$, where $\underline{x} = x_1, \ldots x_k$,[1] and via the Herbrand normal form A^H to arbitrary formulas A in prenex normal form. Moreover, it extends to open first order theories T (i.e. theories whose axioms are purely universal sentences), where then the disjunction is verifiable in T, i.e. $T \vdash \bigvee_{i=0}^{n} A^H(\underline{t}_i)$ (and even is a tautological consequence of a conjunction of finitely many closed instances of the non-logical axioms of T). First order logic with equality can be treated as the special case, where T is an open axiomatization of equality. For first order logic (with or without equality) the Herbrand terms are built up out of A-material (resp. A^H-material) only with possible help of some distinguished constant symbol c in case A (resp. A^H) does not contain any constant. For open first order theories T they may in addition contain some of the constants and function symbols occurring in the non-logical T-axioms used in the proof. For more details see e.g. [20, 3, 6].

There are both model-theoretic and proof-theoretic proofs of Herbrand's theorem. But whereas the former proofs are ineffective the latter provide a procedure for extracting Herbrand terms t_i from a given proof of A. The actual construction of Herbrand terms out of a given proof is of importance in the area of computational logic and has also been used in significant applications to mathematics (see [13, 14]).

The existing proof-theoretic approaches to Herbrand's theorem are based on cut elimination or related techniques like ε-elimination which involve global transformations of the given proof. In his review [12] of [20], G. Kreisel suggested the possibility of using Gödel's functional ('dialectica') interpretation FI ([8, 23]) to prove Herbrand's theorem. To our knowledge this suggestion

[1] For notational simplicity we avoid below to write tuples.

has never been taken up in the literature and the present note aims at filling this lacuna: We give an extraction algorithm of Herbrand terms via functional interpretation in the variant developed in [20] which we from now on also call FI. The verifiability of the extracted disjunction as a tautology or T-provable disjunction is achieved by a simple model theoretic argument. As the case for open theories T immediately reduces (via the deduction theorem) to that of first order logic without equality PL, we only treat the latter.

¿From a given PL-proof of a sentence $\exists x A_{qf}(x)$, FI extracts a closed term t in an extension of typed λ-calculus by decision-by-case constants χ_A for each quantifier-free formula A of $\mathcal{L}(\mathrm{PL})$. After computing the β-normal form $nf(t)$ of t, the Herbrand terms can be read off. The length of the resulting Herbrand disjunction is bounded by $2^{\#_\chi(nf(t))}$, where $\#_\chi(nf(t))$ is the total number of χ-occurrences in $nf(t)$.

The significance of this FI-based approach to the extraction of Herbrand terms is due to the following points:

1. FI has recently been successfully implemented by M.-D. Hernest ([9]) in H. Schwichtenberg's MINLOG system which also contains an efficient normalization tool ('normalization by evaluation', see [2]). We expect that this implementation can be adapted to yield a useful Herbrand-term extraction tool.

2. Suppose that in a PL-proof of (1) $\exists x\, A_{qf}(x)$ classical logic is only used to infer (1) from (2) $\forall x A_{qf}(x) \to \bot$, where (2) is proved intuitionistically. Then already the original direct Gödel functional interpretation (i.e. without negative translation as a preprocessing step and also without Shoenfield's modification) can be used to extract a Herbrand disjunction for (1) which will in general (though not always[2]) be simpler than the detour through full classical logic. This is because the type levels will be lower resulting in a more efficient normalization and hence a shorter Herbrand disjunction.

3. When combined with known estimates ([1]) on the size of $nf(t)$ we

[2]In the Statman example discussed below the original functional interpretation already creates as high types as the Shoenfield variant does. This is unavoidable here since the Statman example has the worst possible Herbrand complexity despite the fact that its form (2) is provable in intuitionistic logic.

immediately obtain bounds on Herbrand's theorem which match the most advanced estimates based on cut-elimination ([6, 7, 25]).

4. In [13] Kreisel discusses how to derive new results in mathematics by analysing the structure of Herbrand terms, e.g. growth conditions, extracted from a given proof. This has been carried out in connection with Roth's theorem by Luckhardt in [14]. Often it will be possible to read off some structural properties of the Herbrand terms already from the FI-extracted E-PL$^\omega$ term t *prior to normalization*, e.g. by analysing which constant and function symbols occur in the extracted term, thereby establishing bounds on the complexity or independence from parameters for the Herbrand terms prior to their actual construction via $nf(t)$.

2 An FI-based approach to Herbrand's Theorem

FI is usually applied to (appropriate formulations of) intuitionistic arithmetic (Heyting arithmetic) in all finite types. Already for the logical axioms and rules the proof of the soundness of FI relies on some minimal amount of arithmetic. Combined with negative translation FI extends to (higher type extensions of) Peano arithmetic (PA). In the following we will use Shoenfield's variant which achieves this in one step and denote this form by FI as well.

To apply FI to first-order predicate logic(PL), we will adapt the soundness proof from Shoenfield [20]. Shoenfield gives a soundness proof of FI for PA which for logical axioms and rules only uses properties of arithmetic to ensure the existence of decision-by-case terms for quantifier-free formulas. By explicitly adding decision-by-case constants χ_A for all quantifier-free formulas A in $\mathcal{L}$(PL) to the language of PL$^\omega$, we can re-use Shoenfield's proof for the soundness of FI of PL in E-PL$^\omega$:=PL extended to all finite types (based on extensionally defined equality).

We then can, for proofs of sentences $\exists x A_{qf}(x)$ in the language $\mathcal{L}$(PL), extract realizing terms t in the extended language E-PL$^\omega$. After normalizing the E-PL$^\omega$-term t one can read off from the normal form $nf(t)$ a collection of

terms $t_1, \ldots, t_n$ for a Herbrand disjunction over A, where the t_i again are ordinary closed terms of PL without any higher type constructs and without the decision-by-case constants.

Remark. *At a first look one might think that the so-called Diller-Nahm version ([5, 4]) of Shoenfield's variant might be more suitable in connection with Herbrand's theorem: it avoids definitions by cases which depend on the prime formulas in favour of definition of case-functionals which do not depend on A_{qf} but only on cases $x =_0 0$ versus $x \neq 0$. However, our technique of eliminating all definitions by cases by explicitly writing out all cases as different terms does not distinguish between these two kinds of case-definitions. In addition to not being beneficial, the Diller-Nahm variant actually relies on a modest amount of arithmetic which is not available in our context of pure logic.*

We now describe the system of first-order predicate logic PL and its extension E-PL$^\omega$ to all finite types, in which our proof will be carried out.

First-order predicate logic PL

I. **The language $\mathcal{L}$(PL) of PL:**

As logical constants we use $\neg, \vee, \forall$. $\mathcal{L}$(PL) contains variables $x, y, z, \ldots$ which can be free or bound, and constants $c, d, \ldots$. Furthermore we have, for every arity n, (possibly empty) sets of function symbols $f, g, \ldots$ and predicate symbols $P, Q, \ldots$. Formulas and terms are defined in the usual way.

Abbreviations:

$A \to B :\equiv \neg A \vee B, A \wedge B :\equiv \neg(\neg A \vee \neg B), \exists x A(x) :\equiv \neg\forall x \neg A(x)$.

II. **Axioms of PL**

(i) $\neg A \vee A$

(ii) $\forall x A(x) \to A[t/x]$ (t free for x in $\forall x A(x)$)

III. **Rules of PL**

(i) $A \vdash B \vee A$ (expansion)

(ii) $A \vee A \vdash A$ (contraction)

(iii) $(A \vee B) \vee C \vdash A \vee (B \vee C)$ (associativity)

(iv) $A \vee B, \neg A \vee C \vdash B \vee C$ (cut)

(v) $A \vee B \vdash \forall x A(x) \vee B$ ($\forall$ -introduction), where x is not free in B.

Note. *As will be seen later, the degree of the terms extracted by FI depends on the $\neg$-depth of formulas. We treat only Shoenfield's calculus, but when translating other calculi for PL into Shoenfield's calculus, we extend Shoenfield's quantifier axioms and rules and the translation $\exists x A(x) :\equiv \neg\forall x \neg A(x)$ to blocks of quantifiers, i.e.$\exists \underline{x} A(\underline{x}) :\equiv \neg\forall \underline{x} \neg A(\underline{x})$, to avoid an artificial blow-up of the degrees when treating blocks of existential quantifiers.*

Note. *We assume w.l.o.g. that there exists at least one constant symbol, c, in our language, as Herbrand's theorem would fail otherwise.*

Extensional predicate logic in all finite types

The set $\mathbf{T}$ of all finite types is defined inductively:

$$(i)\ 0 \in \mathbf{T},\ (ii)\ \rho, \tau \in \mathbf{T} => \rho \to \tau \in \mathbf{T}$$

For convenience we write $0^n \to 0$ for $\overbrace{0 \to (0 \to (\ldots (0}^{n} \to 0) \ldots)$.

The language of E-PL$^\omega$

The language E-PL$^\omega$ is based on a many-sorted version PL$^\omega$ of PL which contains variables $x^\rho, y^\rho, z^\rho, \ldots$ and quantifiers $\forall^\rho, \exists^\rho$ for all types ρ. As constants E-PL$^\omega$ contains the constants $c, d, \ldots$ (at least one: c) of PL as constants of type 0, and the function symbols $f, g, \ldots$ of PL as constants of type $0^n \to 0$ for functions of arity n. Furthermore E-PL$^\omega$ contains decision-by-case constants χ_A of type $0^n \to 0 \to 0 \to 0$ for all quantifier-free formulas A in the original language $\mathcal{L}$(PL), where n is the number of free variables in A. E-PL$^\omega$, moreover, contains a λ-abstraction operator. The predicate symbols of E-PL$^\omega$ are the predicate symbols of PL and equality of type 0 (denoted by $=_0$).

Higher type equality in E-PL$^\omega$ is defined extensionally over type 0 equality:

$$s =_\rho t :\equiv \forall x_1^{\rho_1}, \ldots, x_n^{\rho_n}(s\underline{x} =_0 t\underline{x}),$$

where $\rho = \rho_1 \to \ldots \to \rho_n \to 0$.

Formulas are defined in the usual way starting from prime formulas $s =_0 t$ and $P(t_1, \ldots, t_n)$.

Remark. *Below we often refer implicitly to the obvious embedding of* PL *into* E-PL$^\omega$*, where constants and variables of* PL *represented by their type* 0 *counterparts in* E-PL$^\omega$ *and (n-ary) function symbols of* PL *as constants of type* $0^n \to 0$*, in particular* PL *terms* $f(t_1, \ldots, t_n)$ *are represented by* $((\ldots(ft_1)\ldots)t_n)$*. Recall that the predicate symbols of* E-PL$^\omega$ *are those of* PL *plus* $=_0$*.*

Terms of E-PL$^\omega$

(i) constants c^ρ and variables x^ρ are terms of type ρ (in particular the constants $c, d, \ldots$ of PL are terms of type 0),

(ii) if x^ρ is a variable of type ρ and t^τ a term of type τ, then $\lambda x^\rho.t^\tau$ is a term of type $\rho \to \tau$,

(iii) if t is a term of type $\rho \to \tau$ and s is a term of type ρ, then (ts) is a term of type τ. In particular, if $t_1, \ldots, t_n$ are terms of type 0 and f is an n-ary function symbols of PL, then $((\ldots(ft_1)\ldots)t_n)$ is a term of type 0 which we usually will write as $f(t_1, \ldots, t_n)$.

Axioms and Rules of E-PL$^\omega$

(i) axioms and rules of PL extended to all sorts of E-PL$^\omega$,

(ii) axioms for β-normalization in the typed λ-calculus: $(\lambda x.t)s =_\rho t[s/x]$ for appropriately typed x, t and s,

(iii) equality axioms for $=_0$,

(iv) higher type extensionality:

$$E_\rho : \forall z^\rho, x^{\rho_1}, y^{\rho_1}, \ldots, x^{\rho_k}, y^{\rho_k}(\bigwedge_{i=1}^{k}(x_i =_{\rho_i} y_i) \to z\underline{x} =_0 z\underline{y}),$$

where $\rho = \rho_1 \to (\rho_2 \to (\ldots \to \rho_k) \to 0)\ldots)$,

(v) axioms for the constants $\chi_{A_{qf}}$: $A_{qf}(\underline{x}) \to \chi_{A_{qf}}\underline{x}yz =_0 y$ and $\neg A_{qf}(\underline{x}) \to \chi_{A_{qf}}\underline{x}yz =_0 z$, where $\underline{x}$ are the free variables of the quantifier-free formula A_{qf} of $\mathcal{L}$(PL).

Definition 2.1. *We define the type level $lv(t)$ of a term t inductively over the type of t as follows: $lv(0) := 0$ and $lv(\rho \to \tau) := max(lv(\tau), lv(\rho) + 1)$. The degree $dg(t)$ of a term t is then the maximum over the type levels of all subterms of t.*

Definition 2.2. *Let $\mathcal{M} = \{M, \mathcal{F}\}$ be a model for $\mathcal{L}$(PL). Then $\mathcal{M}^\omega = \{M^\omega, \mathcal{F}^\omega\}$ is the full set-theoretic type structure over M, i.e. $M^0 :\equiv M$, $M^{\rho\to\tau} :\equiv {}^{M^\rho}{}_{M^\tau}$ and $M^\omega :\equiv \bigcup_{\rho\in T} M^\rho$. Constants, functions and predicates of $\mathcal{M}$ retain their interpretation under $\mathcal{F}$ in $\mathcal{F}^\omega$. λ-terms are interpreted in the obvious way. Furthermore, $\mathcal{F}^\omega$ defines the following interpretation of χ_A:*

For $\underline{a}, b, c \in M$ we define $[\chi_A]_{\mathcal{M}^\omega}\underline{a}bc := \begin{cases} b & \text{if } \mathcal{M} \models A_{qf}(\underline{a})^3 \\ c & \text{otherwise.} \end{cases}$

Proposition 2.3. *$\mathcal{M}^\omega$ is a model of* E-PL$^\omega$. *If A is a sentence of $\mathcal{L}$(PL) and $\mathcal{M}^\omega \models A$, then $\mathcal{M} \models A$.*

Proof. Obvious from the construction of $\mathcal{M}^\omega$. □

In the following $\exists x A_{qf}(x)$ will denote a closed formula. For open formulas one can replace each free variable with new distinct constants, carry out the extraction procedure and then reintroduce the variables to get a corresponding Herbrand disjunction for the open case.

[3]More precisely, $\mathcal{M} \models A_{qf}(\underline{a})$ means that $A_{qf}(\underline{x})$ holds in $\mathcal{M}$ provided the free variables x_i get assigned the element $a_i \in M$.

Lemma 2.4. *If* PL $\vdash \exists x A_{qf}(x)$ *then* FI *extracts a closed term* t^0 *of* E-PL$^\omega$ *s.t.* E-PL$^\omega \vdash A_{qf}(t)$.
The proof of $A_{qf}(t)$ *can actually be already carried out in the quantifier-free fragment* qf-WE-PL$^\omega$ *(in the sense of* [23]*) of* WE-PL$^\omega$, *where the latter is the fragment of* E-PL$^\omega$ *which results by replacing the extensionality axioms by the quantifier-free weak rule of extensionality due to* [21] *(see also* [11]*).*

Proof. This is essentially Shoenfield's proof in [20]. The only two cases to note are the expansion rule and the contraction rule.

If $B \vee C$ has been inferred from B by the expansion rule we need an arbitrary closed term of suitable type to realize C. Since we assumed there exists at least one constant c of type 0, we can, using lambda abstraction, construct closed terms $\lambda \underline{x}.c^0$ of suitable type to realize C.

For the contraction rule the argument is somewhat more involved: Let $A(\underline{a})$ be an arbitrary formula with $\underline{a}$ denoting the free variables of A. To each formula A Shoenfield assigns a formula $A^* \equiv \forall \underline{x} \exists \underline{y} A'(\underline{x}, \underline{y}, \underline{a})$, where A' is quantifier-free. The quantifier-free skeleton A_{qfs} of $A \in \mathcal{L}(\overline{\text{PL}})$ is the formula A with all quantifiers removed and distinct new variables substituted for the quantified variables of A, i.e. $A_{qfs}(\underline{b}, \underline{a})$, where $\underline{b}$ are the new variables and $\underline{a}$ are the original free variables of A. The formula A' is a substitution instance $A_{qfs}([\underline{x}, \underline{y}], \underline{a})$ of $A_{qfs}(\underline{b}, \underline{a})$, where $[\underline{x}, \underline{y}]$ denotes some tuple of terms which do not contain any constants but are built up exclusively out of $\underline{x}, \underline{y}$. These terms have been substituted for $\underline{b}$. For simplicity we will in the following consider only single variables x, y and a single parameter a, as the argument easily generalizes to tuples of variables.

To interpret the contraction rule $A \vee A \vdash A$ we have to produce a realizer for the conclusion

$$\forall x_3 \exists y_3 A'(x_3, y_3, a)$$

from realizers of the premise

$$\forall x_1, x_2 \exists y_1, y_2 (A'(x_1, y_1, a) \vee A'(x_2, y_2, a)),$$

where in general x_i, y_i will be of arbitrary type. However, the terms composed of x_i, y_i instantiating A_{qfs} to yield A' are of type 0, since A^* interprets the

first order formula $A \in \mathcal{L}(\mathrm{PL})$. The functional interpretation of the premise yields closed terms t_1, t_2 s.t.

$$\forall x_1, x_2, a\big(A'(x_1, t_1 x_1 x_2 a, a) \vee A'(x_2, t_2 x_1 x_2 a, a)\big).$$

Substituting x_1 for x_2 gives

$$\forall x_1, a\big(A'(x_1, t_1' x_1 a, a) \vee A'(x_1, t_2' x_1 a, a)\big),$$

where $t_1' x_1 a := t_1 x_1 x_1 a$ and $t_2' x_1 a := t_2 x_1 x_1 a$.

Hence, after renaming x_3 in the conclusion into x_1, a term t_3 realizing y_3 (when applied to x_1, a) must satisfy:

$$t_3 x_1 a = \begin{cases} t_1' x_1 a & \text{if } A'(x_1, t_1' x_1 a, a) \\ t_2' x_1 a & \text{otherwise,} \end{cases}$$

i.e.

$$t_3 x_1 a = \begin{cases} t_1' x_1 a & \text{if } A_{qfs}([x_1, y](y/t_1' x_1 a), a) \\ t_2' x_1 a & \text{otherwise.} \end{cases}$$

This term t_3 can be defined via our decision-by-case constants for the quantifier-free skeleton A_{qfs} of A as follows:

$$t_3 := \lambda x_1, a, \underline{v}.\chi_{A_{qfs}}([x_1, y](y/t_1' x_1 a), a, t_1' x_1 a \underline{v}, t_2' x_1 a \underline{v}),$$

where $\underline{v}$ is a tuple of fresh variables of appropriate types such that $t_1' x_1 a \underline{v}$ is of type 0.

Hence it is sufficient to have decision-by-case constants χ_A for each quantifier-free formula A of $\mathcal{L}(\mathrm{PL})$. These have been explicitly added to the language of E-PL$^\omega$. □

Example. *As an example, consider the formula* $A \equiv \exists x \forall y (P(x) \vee \neg P(y))$. *The Shoenfield translation* A^* *of* A *is* $A^* \equiv \forall f \exists x \neg\neg (P(x) \vee \neg P(f(x)))$, *which is classically equivalent to* $\forall f \exists x (P(x) \vee \neg P(f(x)))$. *The matrix* $A' \equiv (P(x) \vee \neg P(f(x)))$ *is an instance of* $A_{qfs}(b_1, b_2) \equiv P(b_1) \vee \neg P(b_2)$, *namely* $A_{qfs}(x, f(x))$.

Functional interpretation will extract from a proof of A, *which necessarily must use the contraction rule at least once, a functional* Φ *realizing* x *in* f.

The term will also use some constant c, since A itself contains no constants. An obvious Φ *is the following:*

$$\Phi(f) := \begin{cases} c & \text{if } P(c) \vee \neg P(f(c)) \\ f(c) & \text{otherwise.} \end{cases}$$

Lemma 2.5. *If* E-PL$^\omega \vdash A_{qf}(t)$ *and* $nf(t)$ *is the* β*-normal form of* t*, then* E-PL$^\omega \vdash A_{qf}(nf(t))$.

Proof. Since t reduces to $nf(t)$, we have E-PL$^\omega \vdash t =_\rho nf(t)$. □

Lemma 2.6. *If* t *is of type* 0*, closed and in* β*-normal form, then there exist closed terms* $t_1, \ldots, t_n \in \mathcal{L}(\mathrm{PL})$*, s.t.* $\mathcal{M}^\omega \models t = t_1 \vee \ldots \vee t = t_n$*. Moreover,* $n \leq 2^{\#_\chi(nf(t))}$*, where* $\#_\chi(nf(t))$ *is the total number of all* χ*-occurrences in* $nf(t)$.

Proof. Since t is of type 0, closed and in β-normal form and has only constants of degree ≤ 1 it contains no more λ-expressions: Assume there still is a λ-expression $\lambda x.r$ left and assume w.l.o.g. that it is not contained in any other λ-expression. Then if $\lambda x.r$ occurs with an argument $(\lambda x.r)s$ it could be further reduced, which contradicts that t is in normal form. If $\lambda x.r$ occurs without an argument it must be at least of type 1, and then since t is closed either $\lambda x.r$ must occur in another λ-expression, since the function symbols of PL only take arguments of type 0, or $t \equiv \lambda x.r$. But this contradicts that $\lambda x.r$ was not contained in any other term and that t was of type 0. Similarly, one infers that the function symbols f always occur with a full stock of arguments in t.

To read off the terms t_i by consider a tree constructed from t by "evaluating" the χ's : choose any outermost χ and build the left (resp. right) subtree by replacing the occurance of the corresponding term $\chi(\underline{s}, t_1, t_2)$ in t with t_1 (resp. t_2). Continue recursively on the left and right subtrees until all χ's have been evaluated. Every path in the tree from the root to a leaf then represents a list of choices on the χ's and thus every leaf is a term $t_i \in \mathcal{L}(\mathrm{PL})$.

It follows trivially that $\mathcal{M}^\omega \models t = t_1 \vee \ldots \vee t = t_n$. As a simple estimate on the length n we get $n \leq 2^{\#_\chi(nf(t))}$. □

Theorem 2.7. *Assume that* PL $\vdash \exists x A_{qf}(x)$. *Then there is a collection of closed terms* $t_1, t_2, \ldots, t_n$ *in* $\mathcal{L}$(PL) *which can be obtained by normalizing a* FI *extracted realizer* t *of* $\exists x$ *s.t.* $\bigvee_{i=1}^{n} A_{qf}(t_i)$ *is a tautology. The terms* t_i *are built up out of the* A_{qf}*-material (possibly with the help of the distinguished constant* c *in case* A_{qf} *does not contain any constant). Moreover,* $n \leq 2^{\#_\chi(nf(t))}$.
The theorem also extends to tuples $\exists \underline{x}$ *of quantifiers.*

Proof. The theorem follows from the above propositions and lemmas. By the soundness of FI we can extract a closed term t in E-PL$^\omega$ realizing '$\exists x$'. We can assume that t consists exclusively of constants and function symbols for $\mathcal{L}$(PL) and some decision-by-case constants χ_B, restricted to quantifier-free formulas B built up from **predicates occurring in** A by means of propositional connectives. This restriction can be verified by a simple model-theoretic argument: give all predicates not occurring in A a trivial interpretation, e.g. interpret them as "always true", and replace decision-by-case expressions over such predicates by appropriate constants. In decision-by-case constants over combinations of predicates occurring and predicates not occurring in A, those not occurring in A can be absorbed.

We then normalize t to $nf(t)$ and read off the terms $t_1, \ldots, t_n$ from $nf(t)$ as in lemma 2.6. Let $\mathcal{M}$ be an arbitrary model of $\mathcal{L}$(PL), then $\mathcal{M}^\omega \models \bigvee_{i=1}^{n} A_{qf}(t_i)$. As the t_i are already closed terms of $\mathcal{L}$(PL), also $\mathcal{M} \models \bigvee_{i=1}^{n} A_{qf}(t_i)$. Since $\mathcal{M}$ was an arbitrary model, the completeness theorem for PL yields that also PL $\vdash \bigvee_{i=1}^{n} A_{qf}(t_i)$. Since $\bigvee_{i=1}^{n} A_{qf}(t_i)$ is quantifier-free it follows that it is a tautology (note that PL is predicate logic **without** equality).

The FI-extracted term t consists of A_{qf}-material, decision-by-case constants and λ-abstractions. The normal form $nf(t)$ contains no more λ, the extracted t_i no more decision-by-case constants, so the result follows. □

Corollary 2.8. *Let* $\mathcal{T}^\omega :=$ WE-PL$^\omega + \Gamma$, *where all additional axioms of the set* Γ *have a functional interpretation in by closed terms of* WE-PL$^\omega$ *(provably in* WE-PL$^\omega + \Gamma$*). If* $\mathcal{T}^\omega \vdash \exists x^0 A_{qf}(x)$, *then there is a collection of terms* $t_1, \ldots, t_n$ *in* $\mathcal{L}$(PL), *extractable via* FI, *s.t.* $\mathcal{T}^\omega \vdash \bigvee_{i=1}^{n} A_{qf}(t_i)$. *The*

terms t_i are built up out of the constant and function symbols of $\mathcal{L}(\mathrm{PL})$ which occur (modulo the embedding of PL *into* WE-PL$^\omega$*) in A_{qf} and Γ.*

Proof. It is sufficient to note that extending E-PL$^\omega$ with the axioms Γ adds no new constants to the language. The corollary then follows by the same arguments as in the proof of Theorem 2.7, except that $\bigvee_{i=1}^{n} A_{qf}(t_i)$ is no longer a tautology, but provable in $\mathcal{T}^\omega$. □

Example (continued). *For $A \equiv \exists x \forall y (P(x) \vee \neg P(y))$ the functional Φ realizing x in f can be defined in* E-PL$^\omega$ *as $\Phi :\equiv \lambda f.\chi_{A_{qfs}}(c, f(c), c, f(c))$. This new decision-by-case term is then applied to f, so that after normalization and unfolding of the χ_A the Herbrand disjunction will be:*

$$(P(c) \vee \neg P(f(c))) \vee (P(f(c)) \vee \neg P(f(f(c))))$$

In order to give an estimate on the number of extracted PL-terms, we need an estimate on the degree $dg(t)$ of the FI-extracted E-PL$^\omega$-term t.

Definition 2.9. *Let A be a formula, then we define the degree $dg(A)$ to be the $\neg$-depth of A. Let ϕ be a proof, then $dg(\phi)$ is the maximum degree of cut formulas occurring in ϕ and the end-formula of ϕ. The end-formula always is purely existential, hence $dg(\phi) = \max\{1, dg(A_1), \ldots, dg(A_n)\}$ for cut formulas A_i in ϕ.*

In Shoenfield's variant of FI only negation increases the type of the functional realizers. Since none of the derivation rules further increase the types, $dg(\phi)$ correctly estimates degree of the FI-extracted E-PL$^\omega$-term t. Refining a result by Schwichtenberg [18, 19], Beckmann [1] proves the following bound on normalization in the typed λ-calculus (which applies to our 'applied' λ-calculus by treating our constant symbols as free variables):

Theorem 2.10. (Beckmann,[1]) *Let t be a term in typed λ-calculus, then the length of any reduction sequence is bounded by $2^{\|t\|}_{dg(t)}$*

Corollary 2.11. *The number of terms extracted in Theorem 2.7 from a proof ϕ can be bounded by $2^{3\|t\|}_{dg(\phi)+1}$.*

Proof. To give a bound on $\#_\chi(nf(t))$ we use the following trick : from t construct a term t' by replacing every occurrance of χ by a term $((\lambda x^0.\chi)c^0)$. Then $\|t'\| \leq 3 \cdot \|t\|$ and t, t' have the same normal form. For t' consider a normalization sequence of the following kind : first perform all possible reduction steps except those on the terms substituted for the χ, then perform the reductions on the $((\lambda x^0.\chi)c^0)$ terms. The length of such a reduction sequence trivially is an upper bound on $\#_\chi(nf(t')) = \#_\chi(nf(t))$.

By Definition 2.9 and Theorem 2.10 we can bound the length of any reduction sequence of t' and hence $\#_\chi(nf(t))$ by $2^{3\cdot\|t\|}_{dg(\phi)}$. The result then follows from Theorem 2.7. □

Remark. *The dependence of the size of the Herbrand disjunction extracted by FI on the $\neg$-depth of cut formulas directly corresponds to the dependence of the complexity of cut elimination (and hence the length of Herbrand disjunctions extracted by cut elimination) on the quantifier alternations in the cut formulas.*

As mentioned above, the extraction of realizing terms generalizes to tuples, i.e. to formulas $\exists \underline{x} A_{qf}(\underline{x})$. For arbitrary prenex formulas we first construct the Herbrand normal form which then is a purely existential statement.

3 Discussion of bounds on Herbrand's Theorem

By an analysis of the E-PL$^\omega$ terms extracted by FI and using Beckmann's bounds on normalisation in the typed λ-calculus, we can extract bounds on the size of a Herbrand disjunction (i.e. the number of disjuncts), which match the best known bounds obtained via the cut elimination theorem [6, 7].

In [24, 25], Zhang gives a very technical proof that the hyperexponential complexity of cut elimination and the length of Herbrand disjunctions depend primarily on the quantifier alternations in the cut formulas, while quantifier blocks and propositional connectives do not contribute to the height of the tower of exponentials. These results on the length of the Herbrand disjunction follow easily from the extraction of Herbrand terms via FI, the bound

on the degree of extracted terms and Beckmann's bounds on normalization.

In [22], Statman shows a hyperexponential lower bound on Herbrand's theorem, by describing formulas S_n for which there exist short proofs, but every Herbrand disjunction must have size at least 2_n. Later presentations of Statman's theorem are due to Orevkov and Pudlak [15, 16, 17]. The short proofs given by Pudlak are of size polynomial in n, yielding FI-extracted terms of size exponential in n (by [10]). The formulas occurring in the proof can be shown to have $\neg$-depth at most n, but by careful analysis of the extracted FI terms one can bound their degree by $n - 1$. Together with Corollary 2.11 this yields a match between an upper bound on the size of a Herbrand disjunction for S_n and Statman's lower bound as good as those obtained via cut-elimination.

References

[1] A. Beckmann. Exact bounds for lengths of reductions in typed λ-calculus. *Journal of Symbolic Logic*, 66:1277–1285, 2001.

[2] U. Berger and H. Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In R. Vemuri, editor, *Proceedings of the 6. Annual IEEE Symposium on Logic in Computer Sciences(LICS)*, pages 203–211. IEEE Press, Los Alamitos, 1991.

[3] S. R. Buss. An Introduction to Proof Theory. In S. R. Buss, editor, *Handbook of Proof Theory*, pages 2–78. Elsevier Science B.V., 1998.

[4] J. Diller. Logical problems of functional interpretations. *Annals of Pure and Applied Logic*, 114:27–42, 2002.

[5] J. Diller and W. Nahm. Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik Endlicher Typen. *Arch. Math. Logik*, 16:49–66, 1974.

[6] P. Gerhardy. Improved Complexity Analysis of Cut Elimination and Herbrand's Theorem. Master's thesis, Aarhus, 2003.

[7] P. Gerhardy. Refined Complexity Analysis of Cut Elimination. In M. Baaz and J. Makovsky, editors, *Proceedings of the 17th International*

Workshop CSL 2003, volume 2803 of *LNCS*, pages 212–225. Springer-Verlag, Berlin, 2003.

[8] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

[9] M.-D. Hernest. A comparison between two techniques of program extraction from classical proofs. Preprint 14pp., 2003.

[10] M.-D. Hernest and U. Kohlenbach. A Complexity Analysis of Functional Interpretations. Technical report BRICS RS-03-12, DAIMI, Department of Computer Science, University of Aarhus, Aarhus, Denmark, Feb. 2003.

[11] U. Kohlenbach. A note on Spector's quantifier-free rule of extensionality. *Arch. Math. Logic*, 40:89–92, 2001.

[12] G. Kreisel. Review of [20]. in Mathematical Reviews. 37 #1224.

[13] G. Kreisel. Finiteness Theorems in Arithmetic : An Application of Herbrand's Theorem for σ_2-formulas. In J. Stern, editor, *Logic Colloquium '81*, pages 39–55. North-Holland Publishing Company, 1982.

[14] H. Luckhardt. Herbrand-Analysen zweier Beweise des Satzes von Roth: Polynomiale Anzahlsschranken. *Journal of Symbolic Logic*, 54:234–263, 1989.

[15] V. P. Orevkov. Lower bounds on the increase in complexity of deductions in cut elimination. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov*, 88:137–162, 1979. English transl., J. Soviet Math. **20**(1982), no.4.

[16] V. P. Orevkov. *Complexity of Proofs and Their Transformations in Axiomatic Theories*, volume 128 of *Translations of Mathematical Monographs*. AMS, Providence, R.I., 1993.

[17] P. Pudlak. The Length of Proofs. In S. R. Buss, editor, *Handbook of Proof Theory*, pages 548–637. Elsevier Science B.V., 1998.

[18] H. Schwichtenberg. Complexity of normalization in the pure typed lambda-calculus. In A. Troelstra and D. van Dalen, editors, *The L.E.J.*

Brouwer Centenary Symposium, pages 453–457. North-Holland Publishing Company, 1982.

[19] H. Schwichtenberg. An upper bound for reduction sequences in the typed λ-calculus. *Arch. Math. Logic*, 30:405–408, 1991.

[20] J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley, Reading, Mass., 1967.

[21] C. Spector. Provably recursive functionals of analysis : a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics. In J. Dekker, editor, *Proceedings of Symposia in Pure Mathematics*, volume 5, pages 1–27. AMS, Providence, R.I., 1962.

[22] R. Statman. Lower Bounds on Herbrand's Theorem. *Proc. of the Amer. Math. Soc.*, 75:104–107, 1979.

[23] A. S. Troelstra, editor. *Metamathematical investigation of intuitionistic arithmetic and analysis*, volume 344 of *Springer Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1973.

[24] W. Zhang. Cut elimination and automatic proof procedures. *Theoretical Computer Science*, 91:265–284, 1991.

[25] W. Zhang. Depth of proofs, depth of cut-formulas and complexity of cut formulas. *Theoretical Computer Science*, 129:193–206, 1994.

Recent BRICS Report Series Publications

RS-03-32 **Philipp Gerhardy and Ulrich Kohlenbach.** ***Extracting Herbrand Disjunctions by Functional Interpretation*****. October 2003. 17 pp.**

RS-03-31 **Stephen Lack and Paweł Sobociński.** ***Adhesive Categories*****. October 2003.**

RS-03-30 **Jesper Makholm Byskov, Bolette Ammitzbøll Madsen, and Bjarke Skjernaa.** ***New Algorithms for Exact Satisfiability*****. October 2003. 31 pp.**

RS-03-29 **Aske Simon Christensen, Christian Kirkegaard, and Anders Møller.** ***A Runtime System for XML Transformations in Java*****. October 2003. 15 pp.**

RS-03-28 **Zoltán Ésik and Kim G. Larsen.** ***Regular Languages Definable by Lindström Quantifiers*****. August 2003. 82 pp. This report supersedes the earlier BRICS report RS-02-20.**

RS-03-27 **Luca Aceto, Willem Jan Fokkink, Rob J. van Glabbeek, and Anna Ingólfsdóttir.** ***Nested Semantics over Finite Trees are Equationally Hard*****. August 2003. 31 pp.**

RS-03-26 **Olivier Danvy and Ulrik P. Schultz.** ***Lambda-Lifting in Quadratic Time*****. August 2003. 23 pp. Extended version of a paper appearing in Hu and Rodríguez-Artalejo, editors,** ***Sixth International Symposium on Functional and Logic Programming*****, FLOPS '02 Proceedings, LNCS 2441, 2002, pages 134–151. This report supersedes the earlier BRICS report RS-02-30.**

RS-03-25 **Biernacki Dariusz and Danvy Olivier.** ***From Interpreter to Logic Engine: A Functional Derivation*****. June 2003.**

RS-03-24 **Mads Sig Ager, Olivier Danvy, and Jan Midtgaard.** ***A Functional Correspondence between Call-by-Need Evaluators and Lazy Abstract Machines*****. June 2003. 13 pp.**

RS-03-23 **Korovin Margarita.** ***Recent Advances in Σ-Definability over Continuous Data Types*****. June 2003. 26 pp.**

RS-03-22 **Ivan B. Damgård and Mads J. Jurik.** ***Scalable Key-Escrow*****. May 2003. 15 pp.**