A Finitary Subsystem of Polymorphic λ-calculus

Thorsten Altenkirch and Thierry Coquand

Preliminary version at October 31, 2000

Introduction

We give a finitary normalisation proof for the restriction of system F where we quantify only over first-order type. As an application, the functions representable in this fragment are exactly the one provably total in Peano Arithmetic. This is inspired by the reduction of Π^1_1-comprehension to inductive definitions presented in [Buch] and this complements a result of [Leiv]. The argument uses a finitary model of a fragment of the system AF_2 considered in [Kriv, Leiv].

1 Polymorphic λ-calculus

We let D be the set of all untyped, maybe open, λ-terms, with β-conversion as equality. We let c_n be the lambda term $\lambda x \lambda f\ f^n\ x$. We consider the following types

$$T \quad ::= \quad \alpha \mid T \to T \mid (\Pi\alpha)T$$

where in the quantification, T has to be built using only α and $\to$.

We use the notation $T_1 \to T_2 \to T_3$ for $T_1 \to (T_2 \to T_3)$ and similarly $T_1 \to T_2 \to \ldots \to T_n$ for $T_1 \to (T_2 \to (\ldots \to T_n))$.

Let us give some examples to illustrate the restriction on quantification. We can have $T = (\Pi\alpha)[\alpha \to \alpha]$ or $(\Pi\alpha)[\alpha \to (\alpha \to \alpha) \to \alpha]$ or even $(\Pi\alpha)[((\alpha \to \alpha) \to \alpha) \to \alpha]$ but a type such as $(\Pi\alpha)[[(\Pi\beta)[\alpha \to \beta]] \to \alpha]$ is not allowed.

We have the following typing rules

$$\frac{}{\Gamma \vdash x : T} \quad x : T \in \Gamma$$

$$\frac{\Gamma, x : T \vdash t : U}{\Gamma \vdash \lambda x\ t : T \to U} \qquad \frac{\Gamma \vdash u : V \to T \quad \Gamma \vdash v : V}{\Gamma \vdash u\ v : T}$$

$$\frac{\Gamma \vdash t : (\Pi\alpha)T}{\Gamma \vdash t : T[U]} \qquad \frac{\Gamma \vdash t : T}{\Gamma \vdash t : (\Pi\alpha)T}$$

where Γ is a finite set of type declaration $x : T$, and in the last rule, α does not appear free in any type of Γ.

We let N be the type $(\Pi\alpha)[\alpha \to (\alpha \to \alpha) \to \alpha]$. We have $\vdash c_n : N$ for each n. The goal of this note is to provide a finitary proof of the following result.

Theorem 1.1 *If $\vdash t : N \to N$ then for each n there exists m such that $t\ c_n\ x\ f = f^m\ x$ for x, f variables.*

This result can be seen as a special case of the normalisation property. We concentrate on this simplified case to illustrate the principle of our argument. From a proof theoretical view point this special case is as hard as the normalisation property.

This result follows from [Leiv] if in the formation of $(\Pi\alpha)T$ we restrict T to be of rank ≤ 2. We extend this to cover types such as

$$(\Pi\alpha)[((\alpha \to \alpha) \to \alpha) \to \alpha]$$

One non finitary proof of this result is the following. Each type is interpreted by a subset of D. We define

$$[\![T \to U]\!]_\rho = \{v \in D \mid t \in [\![T]\!]_\rho \ \to \ v\ t \in [\![U]\!]_\rho\}$$

and

$$[\![(\Pi\alpha)T]\!] = \bigcap_{X \subseteq D} [\![T]\!]_{\alpha = X}$$

We prove then, by induction on derivations

Lemma 1.2 *If $x_1 : T_1, \ldots, x_n : T_n \vdash t : T$ and $u_i \in [\![T_i]\!]$ then $t[u_1, \ldots, u_n] \in [\![T]\!]$*

Corollary 1.3 *If $\vdash t : N$ then $t \in [\![N]\!]$*

Lemma 1.4 *If $u \in [\![N]\!]$ then there exists m such that $u\ x\ f = f^m\ x$ for x, f variables.*

Proof. Consider the subset $S = \{t \mid (\exists m)\ t = f^m\ x\}$. We have $x \in S$ and $f\ t \in S$ if $t \in S$. Hence the result. $\square$

We can now prove the theorem. If $\vdash t : N \to N$ we have then $t\ c_n : N$ because $c_n : N$. But this implies, by the two lemmas that there exist m such that $t\ c_n\ x\ f = f^m\ x$. Let us write $m = \phi(n)$. We say then that the function ϕ is *represented by* the term t.

2 Second-Order Functional Arithmetic

The proof above is not finitary, because of the use of intersection over all subsets, which requires, a priori, Π^1_1-comprehension.

In some cases however, we can replace Π^1_1-comprehension by arithmetical comprehension. For instance, if T is $\alpha \to (\alpha \to \alpha) \to \alpha$ then we have

$$\bigcap_{X \subseteq D} [\![T]\!]_{\alpha = X} = \{t \in D \mid (\exists n)(\forall u, v \in D)\ t\ u\ v = v^n\ u\ \}$$

Indeed if t belongs to all $[\![T]\!]_{\alpha = X}$ then we can take x, y variables not free in t and take X to be the set of all terms of the form $y^n\ x$. We have then $t\ x\ y \in X$ and hence $t\ x\ y = y^n\ x$ for some n. Since x, y does not occur free in t this implies $t\ u\ v = v^n\ u$ for all $u, v \in D$. Conversely if we have $t\ u\ v = v^n\ u$ for all $u, v \in D$ then it is direct to check that we have $t \in [\![T]\!]_{\alpha = X}$ for all $X \subseteq D$.

More generally if T is of rank ≤ 2 then we can directly define $[\![(\Pi\alpha)T]\!]$ by using arithmetical comprehension only. But this does not seem to extend simply to the general case.

To analyse the proof in general, we first translate it in the language of second-order logic over D. We introduce the following logic AF_2: we have two sorts, subsets and terms. We use $X, Y, \ldots$ for variables over subsets and $x, y, \ldots$ for variables over terms. The terms are elements of D. The formulae are

$$A \ ::= \ t \in X \mid A \to A \mid (\forall x)A \mid (\forall X)A$$

In forming $(\forall X)A$, the formula A should be a first-order formula having at most X as a subset variable.

A *model* of AF_2 consists in an implicative algebra $(H, \to, \wedge, 1)$[1] and a valuation function $[\![A]\!]_{\rho,\nu} \in H$ where ρ assigns an element of D to each term variable and ν assigns a function $D \to H$ to each predicate variable. We write as usual $(\rho, x = d)$ and $(\nu, X = f)$ for the *update* of respectively ρ and ν. The valuation function should be such that $[\![(\forall x)A]\!]_{\rho,\nu}$ is the greatest lower bound of all $[\![A]\!]_{(\rho,x=d),\nu}$ for $d \in D$ and $[\![(\forall X)A]\!]_{\rho,\nu}$ is the greatest lower bound of all $[\![A]\!]_{\rho,(\nu,X=f)}$ for $f \in D \to H$. Notice that we don't require H to be complete. Furthermore, we should have

$$[\![A_1 \to A_2]\!]_{\rho,\nu} \in H = [\![A_1]\!]_{\rho,\nu} \to [\![A_2]\!]_{\rho,\nu}$$

and

$$[\![t \in X]\!]_{\rho,\nu} \in H = \nu(X)([\![t]\!]_\rho).$$

To each second-order type T we can associate a formula $C_T(x)$ with one term variable x, by taking $C_\alpha = x \in X_\alpha$, $C_{T \to U} = (\forall y)[C_T(y) \to C_T(x\ y)]$ and $C_{(\Pi\alpha)T} = (\forall X_\alpha)C_T(x)$. To each context $\Gamma = x_1 : T_1, \ldots, x_n : T_n$ we associate the set of formulae $C_\Gamma = C_{T_1}(x_1), \ldots, C_{T_n}(x_n)$ and we have

Lemma 2.1 *If $\Gamma \vdash t : T$ then $\wedge_{A \in C_\Gamma}[\![A]\!] \leq [\![C_T(t)]\!]$ in any model of AF_2. In particular if $\vdash t : T$ we have $[\![C_T(t)]\!] = 1$.*

Next we are going to build a model of AF_2 in a finitary way.

3 A Finitary Model

We consider now only first-order formulae

$$A \ ::= \ t \in X \mid A \to A \mid (\forall x)A$$

We define a first-order logic AF_1 on these formulae. We let $L, M, \ldots$ denote finite sets of formulae, and we write L, M for $L \cup M$ and L, A for $L \cup \{A\}$. We have the rules

$$\frac{}{L \vdash A} \ (A \in L)$$

$$\frac{L \vdash A_1 \quad L, A_2 \vdash A}{L \vdash A} \ (A_1 \to A_2 \in L) \qquad \frac{L, A_1 \vdash A_2}{L \vdash A_1 \to A_2}$$

$$\frac{L, A_1(t) \vdash A}{L \vdash A} \ ((\forall x)A_1 \in L) \qquad \frac{L \vdash A}{L \vdash (\forall x)A}$$

In the last rule, x should not occur free in L.

We write $L \leq M$ iff $L \vdash A$ for all A in M. It can be proved in a finitary way that this defines a poset, that we call S_0. We use this poset to give a finitary Kripke model of AF_2.

[1] That is $(H, \wedge, 1)$ is a meet semilattice and we have $x \wedge y \leq z$ iff $x \leq y \to z$.

The subsets are interpreted as functions $D \to Down(S_0)$ where $Down(S_0)$ is the set of downward closed subsets of S_0. If A is a first-order formula we let $[A]$ be $\{L \in S_0 \mid L \vdash A\}$, and if L is a finite set of formulae $A_1, \ldots, A_n$ we let $[L]$ be $[A_1] \cap \ldots \cap [A_n]$. If X is a variable we let $F_X : D \to Down(S_0)$ be the function $F_X(t) = [X\ t]$. An assignment ν associates functions $D \to Down(S_0)$ to subsets variables. We define

$$[\![X\ t]\!]_\nu = \nu(X)(t) \qquad [\![(\forall x)A]\!] = \cap_{u \in D}[\![A(u)]\!]$$

$$[\![A_1 \to A_2]\!] = \{L \in S_0 \mid M \in [\![A_1]\!] \to L, M \in [\![A_2]\!]\}$$

We let $[\![A_1, \ldots, A_n]\!]$ to be $[\![A_1]\!] \cap \ldots \cap [\![A_n]\!]$.

Lemma 3.1 *if $L \vdash A$ in AF_1 we have $[\![L]\!]_\nu \subseteq [\![A]\!]_\nu$.*

Proof. Since AF_1 is intuitionistic, its derivations are valid in any Kripke model. □

Lemma 3.2 *If A is a first-order formula then $[\![A]\!]_\nu = [A]$ if $\nu(X) = F_X$ for X free in A. Also, $[\![L]\!]_\nu = [L]$ if $\nu(X) = F_X$ for X free in L.*

Proof. By induction on A. This follows from the equalities

$$[A_1 \to A_2] = \{L \in S_0 \mid M \in [A_1] \to L, M \in [A_2]\}$$

and $[(\forall x)A] = \cap_{u \in D}[A(u)]$. □

Lemma 3.3 *If A is a first-order formula with at most X as a subset variable then*

$$\bigcap_{F \in D \to Down(S_0)} [\![A]\!]_{X=F} \in Down(S_0)$$

can be finitary described as the set of all $L \in S_0$ such that $L \vdash A(Z)$ for Z not free in L.

Proof. If $L \vdash A(Z)$ for Z not free in L then we have $[\![L]\!]_\nu \subseteq [\![A(Z)]\!]_\nu$ for any interpretation by lemma 3.1. If we take $\nu(Z) = F$ and $\nu(X) = F_X$ for $X \neq Z$ we get $[\![L]\!] = [L]$ and hence $[L] \subseteq [\![A(Z)]\!]_{Z=F}$ so that $L \in [\![A]\!]_{X=F}$ for all F.

Conversely if $L \in [\![A]\!]_{X=F}$ for all F, then in particular $L \in [\![A]\!]_{X=F_Z}$ and so $L \in [A(Z)]$ that is $L \vdash A(Z)$ for Z not free in L, since $[\![A]\!]_{X=F_Z} = [\![A(Z)]\!]_{Z=F_Z} = [A(Z)]$ by lemma 3.2. □

By finitary, we mean here that the functions we consider in $D \to Down(S_0)$ if looked as relations on $D \times S_0$, are only formed by using arithmetical comprehension.

Using lemma 3.3 we can build in a finitary way a model of AF_2 by taking $H = Down(S_0)$ and

$$1 = S_0, \quad X \wedge Y = X \cap Y, \quad X \to Y = \{L \in S_0 \mid M \in X \to L, M \in Y\}$$

By lemma 2.1 we have that, if $\vdash u : U$ then $1 = [\![C_U(u)]\!]$. In particular, if $\vdash t : N \to N$ then $1 = [\![C_N(t\ c_n)]\!]$, and so $1 = F(t\ c_n\ x\ f)$ if we have $1 = F(x)$ and $F(u) \leq F(f\ u)$ for all u. In particular we can take

$$F(u) = \bigcup_{m \in N} \{L \in S_0 \mid u = f^m\ x\}$$

and we have $1 = F(t\ c_n\ x\ f)$ which implies $t\ c_n\ x\ f = f^m\ x$ for some m.

4 Application

We work now in SAS_0 second order arithmetic with arithmetical comprehension [Leiv]. It is known that this system is conservative over Peano Arithmetic [Troe]. It is possible to represent D and the poset S_0 in SAS_0. The argument above however cannot be formalised as it is in SAS_0 because of the lemma 3.1 which requires the definition of semantics of formulae.

We consider a fixed derivation of a typing judgement of the form $\vdash t : N \rightarrow N$. In this derivation occurs only a finite set of quantified types $T_1, \ldots, T_n$ and we consider the set SF of subformulae of the formulae of the form $C_{T_1}(t_1), \ldots, C_{T_n}(t_n)$. We let then S_1 be the subposet of S_0 which consists only of finite set of such subformulae.

Given any poset defined in SAS_0 we can define $[\![A]\!]_\nu$ for $A \in SF$ in SAS_0 [Troe].

Lemma 4.1 *If $M \vdash A$ with $A \in SF$, $M \subseteq SF$ then $[\![M]\!]_\nu \subseteq [\![A]\!]_\nu$ and this is provable in SAS_0.*

We consider then the model $H = Down(S_1)$ and

$$1 = S_1, \quad X \wedge Y = X \cap Y, \quad X \rightarrow Y = \{L \in S_1 \mid M \in X \rightarrow L, M \in Y\}$$

for all $X, Y \subseteq D$.

Corollary 4.2 *If A is a first-order formula in SF with at most X as a subset variable then*

$$\bigcap_{F \in D \rightarrow H} [\![A]\!]_{X=F} \in H$$

can be finitary described as the set of all $L \in S_1$ such that $L \vdash A(Z)$ for Z not free in L, and this is provable in SAS_0.

Theorem 4.3 *If ϕ is represented by a term then ϕ is provably total in Peano Arithmetic.*

Proof. Suppose that ϕ is represented by a term t. We have a derivation of $\vdash t : N \rightarrow N$. The previous results allow us to transform this derivation to a proof in SAS_0 that $[\![C_N]\!](t\ c_n)$ holds for all n. $\square$

It follows from [Glad] that we can represent all functions provably total in Peano Arithmetic, using only the quantified type $N = (\Pi\alpha)[\alpha \rightarrow (\alpha \rightarrow \alpha) \rightarrow \alpha]$. Indeed, [Glad] shows that we can program the predecessor function, and indeed all primitive recurive function, using only iteration. If follows from this that all functions of Gödel system T can be programmed using only iteration.

Theorem 4.4 *The set of representable functions is exactly the set of functions provably total in Peano Arithmetic.*

References

[Buch] W. Buchholz and K. Schütte. *Proof theory of impredicative subsystems of analysis.* Studies in Proof Theory. Monographs, 2. Bibliopolis, Naples, 1988.

[Glad] M.D. Gladstone. A reduction of the recursion scheme. J. Symbolic Logic 32 1967 505–508.

[Kriv] J.L. Krivine. *Lambda-calcul. Types et modèles.* Masson, Paris, 1990.

[Leiv] D. Leivant. Peano's Lambda Calculus: The Functional Abstraction Implicit in Arithmetic

[Troe] A. Troesltra. *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis.* Lecture Notes in Mathematics 344, 1973.