Arithmetic as a Theory Modulo

Gilles Dowek and Benjamin Werner

Projet LogiCal
Pôle Commun de Recherche en Informatique du Plateau de Saclay
École polytechnique, INRIA, CNRS and Université de Paris-Sud
LIX, École polytechnique, 91128 Palaiseau Cedex, France
{Gilles.Dowek,Benjamin.Werner}@polytechnique.fr

Abstract. We present constructive arithmetic in Deduction modulo with rewrite rules only.

In natural deduction and in sequent calculus, the cut elimination theorem and the analysis of the structure of cut free proofs is the key to many results about predicate logic with no axioms: analyticity and non-provability results, completeness results for proof search algorithms, decidability results for fragments, constructivity results for the intuitionistic case...

Unfortunately, the properties of cut free proofs do not extend in the presence of axioms and the cut elimination theorem is not as powerful in this case as it is in pure logic. This motivates the extension of the notion of cut for various axiomatic theories such as arithmetic, Church's simple type theory, set theory and others. In general, we can say that a new axiom will necessitate a specific extension of the notion of cut: there still is no notion of cut general enough to be applied to any axiomatic theory. Deduction modulo [2, 3] is one attempt, among others, towards this aim.

In deduction modulo, a theory is not a set of axioms but a set of axioms combined with a set of rewrite rules. For instance, the axiom $\forall x\; x + 0 = x$ can be replaced by the rewrite rule $x + 0 \longrightarrow x$. The point is that replacing the axiom by the rewrite rule introduces short-cuts in the corresponding proofs, which avoid axiomatic cuts. When the set of rewrite rules is empty, one is simply back to regular predicate logic. On the other hand, when the set of axioms is empty we have theories expressed by rewrite rules only. For such theories, cut free proofs are similar to cut free proofs in pure logic, in particular they end with an introduction rule. Thus, when a theory can be expressed in deduction modulo with rewrite rules only and, in addition, cuts can be eliminated modulo these rewrite rules, the theory has most of the properties of pure logic. This leads to the question of which theories can be expressed with rewrite rules only in such a way that cut-elimination holds.

It is known that several theories can be expressed in such a setting, for instance all equational theories, type theory, set theory, etc... But arithmetic was an important example of a theory that lacked such a presentation. The goal of this paper is to show that arithmetic can indeed be presented in deduction modulo without axioms in such a way that cut elimination holds. The cut elimination result is built using the generic tools introduced in [3].

When considering arithmetic, it is customary to keep the cut-elimination argument predicative. We show that these generic tools also make it possible to build a predicative proof.

It should be noticed that second-order arithmetic can be embedded in simple type theory with the axiom of infinity and thus that it can be expressed in deduction modulo. Our presentation of first-order arithmetic in deduction modulo uses many ideas coming from second-order arithmetic. However, our presentation of arithmetic has exactly the power of first-order arithmetic.

1 Deduction Modulo

1.1 Identifying Propositions

In deduction modulo, the notions of language, term and proposition are those of predicate logic. But, a theory is formed with a set of axioms Γ *and a congruence* $\equiv$ defined on propositions. Such a congruence may be defined by a rewrite system on terms and on propositions (as propositions contain binders — quantifiers — these rewrite systems are in fact *combinatory reduction systems* [10]). Then, the deduction rules take this congruence into account. For instance, the *modus ponens* is not stated as usual

$$\frac{A \Rightarrow B \quad A}{B}$$

as the first premise need not be exactly $A \Rightarrow B$ but may be only congruent to this proposition, hence it is stated

$$\frac{C \quad A}{B} \text{ if } C \equiv A \Rightarrow B$$

All the rules of intuitionistic natural deduction may be stated in a similar way (see Figure 1).

For example, we can define a congruence with the following rewrite system

$$\begin{array}{ll} 0 + y \longrightarrow y & S(x) + y \longrightarrow S(x+y) \\ 0 \times y \longrightarrow 0 & S(x) \times y \longrightarrow x \times y + y \end{array}$$

In the theory formed with a set of axioms Γ containing the axiom $\forall x\ x = x$ and this congruence, we can prove, in natural deduction modulo, that the number 4 is even

$$\cfrac{\cfrac{\cfrac{}{\Gamma \vdash_\equiv \forall x\ x = x}\text{ axiom}}{\Gamma \vdash_\equiv 2 \times 2 = 4}\ \langle x, x = x, 4\rangle\ \forall\text{-elim}}{\Gamma \vdash_\equiv \exists x\ 2 \times x = 4}\ \langle x, 2 \times x = 4, 2\rangle\ \exists\text{-intro}$$

Substituting the variable x by the term 2 in the proposition $2 \times x = 4$ yields the proposition $2 \times 2 = 4$, that is congruent to $4 = 4$. The transformation of one proposition into the other, that requires several proof steps in usual formulations of natural deduction, is dropped from the proof in deduction modulo.

$$\frac{}{\Gamma \vdash_{\equiv} B} \text{ axiom if } A \in \Gamma \text{ and } A \equiv B$$

$$\frac{\Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C} \Rightarrow\text{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma \vdash_{\equiv} A}{\Gamma \vdash_{<} \equiv B} \Rightarrow\text{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} A \quad \Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C} \wedge\text{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} A} \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B} \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} C} \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C} \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} D \quad \Gamma, A \vdash_{\equiv} C \quad \Gamma, B \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} C} \vee\text{-elim if } D \equiv (A \vee B)$$

$$\frac{}{\Gamma \vdash_{\equiv} A} \top\text{-intro if } A \equiv \top$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} A} \bot\text{-elim if } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B} \langle x, A \rangle \ \forall\text{-intro if } B \equiv (\forall x \ A) \text{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C} \langle x, A, t \rangle \ \forall\text{-elim if } B \equiv (\forall x \ A) \text{ and } C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B} \langle x, A, t \rangle \ \exists\text{-intro if } B \equiv (\exists x \ A) \text{ and } C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} B} \langle x, A \rangle \ \exists\text{-elim if } C \equiv (\exists x \ A) \text{ and } x \notin FV(\Gamma B)$$

Fig. 1. Natural deduction modulo.

In this example, the rewrite rules apply to terms only. Deduction modulo permits also to consider rules rewriting atomic propositions to arbitrary ones. For instance, in the theory of integral domains, we can take the rule

$$x \times y = 0 \longrightarrow x = 0 \vee y = 0$$

that rewrites an atomic proposition to a disjunction.

Notice that, in the proof above, we do not need the axioms of addition and multiplication. Indeed, these axioms are now redundant: since the terms $0 + y$ and y are congruent, the axiom $\forall y \ 0+y = y$ is congruent to the axiom of equality $\forall y \ y = y$. Hence, it can be dropped. Thus, rewrite rules replace axioms.

This equivalence between rewrite rules and axioms is expressed by the the *equivalence lemma* that for every congruence $\equiv$, we can find a theory $\mathcal{T}$ such that $\Gamma \vdash_{\equiv} A$ is provable in deduction modulo if and only if $\mathcal{T}, \Gamma \vdash A$ is provable in ordinary predicate logic [2]. Hence, deduction modulo is not a true extension of predicate logic, but rather an alternative formulation of predicate logic. Of course, the provable propositions are the same in both cases, but the proofs are very different.

1.2 Model of a Theory Modulo

A *model* of a congruence $\equiv$ is a model such that if $A \equiv B$ then for all assignments, A and B have the same denotation. A *model* of a theory modulo $\Gamma, \equiv$ is a model of the theory Γ and of the congruence $\equiv$. Unsurprisingly, the completeness theorem extends to classical deduction modulo [6] and a proposition is provable in the theory $\Gamma, \equiv$ if and only if it is valid in all the models of $\Gamma, \equiv$.

1.3 Normalization in Deduction Modulo

Replacing axioms by rewrite rules in a theory changes the structure of proofs and in particular some theories may have the normalization property when expressed with axioms and not when expressed with rewrite rules. For instance, from the normalization theorem for predicate logic, we get that any proposition that is provable with the axiom $A \Leftrightarrow (B \wedge (A \Rightarrow \bot))$ has a normal proof. But if we transform this axiom into the rule $A \rightarrow B \wedge (A \Rightarrow \bot)$ (Crabbé's rule [1]) the proposition $B \Rightarrow \bot$ has a proof, but no normal proof.

We have proved a *normalization theorem*: proofs normalize in a theory modulo if this theory bears a *pre-model* [3]. A pre-model is a many-valued model whose truth values are reducibility candidates, i.e. sets of proof-terms. Hence we first define proof-terms, then reducibility candidates and finally pre-models.

Definition 1 (Proof-term). *We write $t, u \ldots$ for terms of the language.* Proof-terms *denoted by $\pi, \sigma \ldots$ and are inductively defined as follows.*

$$\begin{array}{lll} \pi ::= & \alpha & \mid I \\ & \mid \lambda\alpha\ \pi \mid (\pi\ \pi') & \mid \delta_\bot(\pi) \\ & \mid \langle \pi, \pi' \rangle \mid fst(\pi) \mid snd(\pi) & \mid \lambda x\ \pi \mid (\pi\ t) \\ & \mid i(\pi) \mid j(\pi) \mid \delta(\pi_1, \alpha\pi_2, \beta\pi_3) & \mid \langle t, \pi \rangle \mid \delta_\exists(\pi, x\alpha\pi') \end{array}$$

Each proof-term construction corresponds to an intuitionistic natural deduction rule: terms of the form α express proofs built with the axiom rule, terms of the form $\lambda\alpha\ \pi$ and $(\pi\ \pi')$ express proofs built with the introduction and elimination rules of the implication, terms of the form $\langle \pi, \pi' \rangle$ and $fst(\pi)$, $snd(\pi)$ express proofs built with the introduction and elimination rules of the conjunction, terms of the form $i(\pi), j(\pi)$ and $\delta(\pi_1, \alpha\pi_2, \beta\pi_3)$ express proofs built with the introduction and elimination rules of the disjunction, the term I expresses the proof built with the introduction rule of the truth, terms of the form $\delta_\bot(\pi)$ express proofs built with the elimination rule of the contradiction, terms of the form $\lambda x\ \pi$ and $(\pi\ t)$ express proofs built with the introduction and elimination rules of the universal quantifier and terms of the form $\langle t, \pi \rangle$ and $\delta_\exists(\pi, x\alpha\pi')$ express proofs built with the introduction and elimination rules of the existential quantifier.

Definition 2 (Reduction). Reduction *on proof-terms is defined as the contextual closure of the following rules that eliminate cuts step by step.*

$$\begin{array}{ll}
(\lambda\alpha\ \pi_1\ \pi_2) \rhd (\pi_2/\alpha)\pi_1 & (\lambda x\ \pi\ t) \rhd (t/x)\pi \\
fst(\langle \pi_1, \pi_2\rangle) \rhd \pi_1 & snd(\langle \pi_1, \pi_2\rangle) \rhd \pi_2 \\
\delta(i(\pi_1), \alpha\pi_2, \beta\pi_3) \rhd (\pi_1/\alpha)\pi_2 & \delta(j(\pi_1), \alpha\pi_2, \beta\pi_3) \rhd (\pi_1/\beta)\pi_3 \\
\delta_\exists(\langle t, \pi_1\rangle, \alpha x \pi_2) \rhd (t/x, \pi_1/\alpha)\pi_2 &
\end{array}$$

We write $\rhd^*$ for the reflexive-transitive closure of the relation $\rhd$.

In the following, the techniques are usual for normalization proofs by reducibility. The setting, however, is different.

Definition 3 (Reducibility candidates). *A proof-term is said to be* neutral *if it is a proof variable or an elimination (i.e. of the form $(\pi\ \pi')$, $fst(\pi)$, $snd(\pi)$, $\delta(\pi_1, \alpha\pi_2, \beta\pi_3)$, $\delta_\perp(\pi)$, $(\pi\ t)$, $\delta_\exists(\pi, x\alpha\pi')$), but not an introduction. A set R of proof-terms is a* reducibility candidate *if*

- *whenever $\pi \in R$, then π is strongly normalizable,*
- *whenever $\pi \in R$ and $\pi \rhd^* \pi'$ then $\pi' \in R$,*
- *whenever π is neutral and if for every π' such that $\pi \rhd^1 \pi'$, $\pi' \in R$ then $\pi \in R$.*

We write $\mathcal{CR}$ for the set of all reducibility candidates.

Definition 4. *Let $\mathcal{SN}$ be the set of all strongly normalizable proof-terms and $\perp\!\!\!\perp$ be the set of all strongly normalizing proof-terms whose normal form is neutral.*

It is easy to check that both $\mathcal{SN}$ and $\perp\!\!\!\perp$ are reducibility candidates. Furthermore, they are respectively the maximal and minimal reducibility candidate with respect to inclusion.

Definition 5 (Pre-model). *A* pre-model $\mathcal{M}$ *for a many-sorted language $\mathcal{L}$ is given by:*

- *for every sort s a set M_s,*
- *for every function symbol f of rank $\langle s_1, \dots, s_n, s_{n+1}\rangle$ a mapping $\hat{f}$ from $M_{s_1} \times \dots \times M_{s_n}$ to $M_{s_{n+1}}$,*
- *for every predicate symbol P of rank $\langle s_1, \dots, s_n\rangle$ a mapping $\hat{P}$ from $M_{s_1} \times \dots \times M_{s_n}$ to $\mathcal{CR}$.*

Definition 6 (Denotation in a pre-model). *Let $\mathcal{M}$ be a pre-model, ϕ an assignment mapping any variable x of sort s to an element of M_s and let t be a term of sort s. We define the object $[\![t]\!]_\phi \in M_s$ by induction over the structure of t.*

- *$[\![x]\!]_\phi = \phi(x)$,*
- *$[\![f(t_1, \dots, t_n)]\!]_\phi = \hat{f}([\![t_1]\!]_\phi, \dots, [\![t_n]\!]_\phi)$.*

Let A be a proposition and ϕ a well-sorted assignment as above. We define the reducibility candidate $[\![A]\!]_\phi$ by induction over the structure of A.

If A is an atomic proposition $P(t_1, \ldots, t_n)$ then $[\![A]\!]_\phi = \hat{P}([\![t_1]\!]_\phi, \ldots, [\![t_n]\!]_\phi)$.

When A is a non-atomic proposition, its interpretation is defined by the following, usual, equations:

$$\begin{aligned}
[\![B \Rightarrow C]\!]_\phi &= \{\pi \in \mathcal{SN} | \pi \rhd^* \lambda\alpha\ \pi' \Rightarrow \forall\sigma \in [\![B]\!]_\phi\ (\sigma/\alpha)\pi' \in [\![C]\!]_\phi\}\\
[\![B \vee C]\!]_\phi &= \{\pi \in \mathcal{SN} \mid \pi \rhd^* i(\pi_1) \Rightarrow \pi_1 \in [\![B]\!]_\phi \wedge \pi \rhd^* j(\pi_2) \Rightarrow \pi_2 \in [\![C]\!]_\phi\}\\
[\![B \wedge C]\!]_\phi &= \{\pi \in \mathcal{SN} | \pi \rhd^* \langle \pi_1, \pi_2\rangle \Rightarrow (\pi_1 \in [\![B]\!]_\phi \wedge \pi_2 \in [\![C]\!]_\phi)\}\\
[\![\top]\!]_\phi &= \mathcal{SN}\\
[\![\bot]\!]_\phi &= \mathcal{SN}\\
[\![\exists x\ B]\!]_\phi &= \{\pi \in \mathcal{SN} | \pi \rhd^* \langle t, \pi'\rangle \Rightarrow \exists X \in M_s\ \pi' \in [\![B]\!]_{\phi, X/x}\}\\
[\![\forall x\ B]\!]_\phi &= \{\pi \in \mathcal{SN} | \pi \rhd^* \lambda x\ \pi' \Rightarrow \forall X \in M_s \forall t \in \mathcal{T}\ (t/x)\pi' \in [\![B]\!]_{\phi, X/x}\}
\end{aligned}$$

where $\mathcal{T}$ is th set of terms of the language.

Definition 7. *A pre-model is said to be a* pre-model of a congruence $\equiv$ *if when $A \equiv B$ then for every assignment ϕ, $[\![A]\!]_\phi = [\![B]\!]_\phi$.*

Theorem 1 (Normalization). *[3] If a congruence $\equiv$ has a pre-model all proofs modulo $\equiv$ strongly normalize.*

In this article we will be able to shorten some proofs using the following remark; it simply states that the previous definition can also be reformulated in a more conventional way.

Proposition 1. *A proof term σ belongs to $[\![A \Rightarrow B]\!]_\phi$ if and only if for any proof term $\pi \in [\![A]\!]_\phi$, $(\sigma\ \pi) \in [\![B]\!]_\phi$.*

A proof term σ belongs to $[\![\forall x_s A]\!]_\phi$ if and only if for any term t of the language and any element X of M_s, $(\sigma\ t) \in [\![A]\!]_{\phi, X/x}$.

2 An Alternative Presentation of Arithmetic

Heyting arithmetic is usually presented as a theory in predicate logic with the axioms of Definition 8 below. Before we give a presentation of arithmetic in deduction modulo, we shall give an alternative presentation HA$_{Class}$ of arithmetic in predicate logic in Definition 9 below and prove the equivalence with HA. This equivalence is proved in several steps using two intermediate theories. Let us first recall the usual presentation of arithmetic.

2.1 Heyting Arithmetic

Definition 8 (HA). *The language of the theory HA is formed with the symbols 0, S, $+$, $\times$ and $=$. The axioms are the axioms of equality corresponding to these symbols and the propositions*

$$\forall x\ \forall y\ (S(x) = S(y) \Rightarrow x = y)$$

$$\forall x\ \neg(0 = S(x))$$

$$((0/x)P \Rightarrow \forall y\ ((y/x)P \Rightarrow (S(y)/x)P) \Rightarrow \forall n\ (n/x)P)$$

$$\begin{array}{ll} \forall y\ (0 + y = y) & \forall x\ \forall y\ (S(x) + y = S(x + y)) \\ \forall y\ (0 \times y = 0) & \forall x\ \forall y\ (S(x) \times y = x \times y + y) \end{array}$$

2.2 A Symbol for Predecessor

The first step is to add a predecessor symbol to arithmetic and the axioms

$$\mathit{Pred}(0) = 0 \qquad \mathit{Pred}(S(x)) = x$$

$$\forall x \forall y\ (x = y \Rightarrow \mathit{Pred}(x) = \mathit{Pred}(y))$$

We prove that the theory obtained this way, called $\mathrm{HA}_{\mathit{Pred}}$ is a conservative extension of HA. This is a consequence of Skolem's theorem for constructive logic (see, for instance, [5]). But notice that in order to obtain the third axiom above, it is not sufficient to skolemize the theorem

$$\forall x \exists y\ ((x = 0 \wedge y = 0) \vee x = S(y))$$

but we need to skolemize the theorem

$$\forall x \exists y\ ((x = 0 \Rightarrow y = 0) \wedge \forall z\ (x = S(z) \Rightarrow y = z))$$

2.3 A Symbol for Natural Numbers

The second step is to introduce a theory HA_N where the universe of discourse is not restricted to the natural numbers and where we have a predicate symbol N to characterize the natural numbers. The language of this theory is formed with the symbols 0, S, $+$, $\times$, $=$, *Pred*, *Null* and N. The axioms are the axioms of equality (including those related to *Pred*, *Null* and N) and the propositions

$$(0/x)P \Rightarrow \forall y\ (N(y) \Rightarrow (y/x)P \Rightarrow (S(y)/x)P) \Rightarrow \forall n\ (N(n) \Rightarrow (n/x)P)$$

$$\begin{array}{cc} N(0) & \forall x\ (N(x) \Rightarrow N(S(x))) \\ \mathit{Pred}(0) = 0 & \forall x\ (\mathit{Pred}(S(x)) = x) \\ \mathit{Null}(0) & \forall x\ (\neg \mathit{Null}(S(x))) \\ \forall y\ (0 + y = y) & \forall x\ \forall y\ (S(x) + y = S(x + y)) \\ \forall y\ (0 \times y = 0) & \forall x\ \forall y\ (S(x) \times y = x \times y + y) \end{array}$$

In the induction scheme, all the symbols, including *Pred*, *Null* and N may occur in the proposition P

Because of the introduction of the predicate N, we must define a translation from the language of $\mathrm{HA}_{\mathit{Pred}}$ to the language of HA_N.

- $|P| = P$, if P is atomic, $|\top| = \top$, $|\bot| = \bot$, $|A \wedge B| = |A| \wedge |B|$, $|A \vee B| = |A| \vee |B|$, $|A \Rightarrow B| = |A| \Rightarrow |B|$,
- $|\forall x\ A| = \forall x\ (N(x) \Rightarrow |A|)$, $|\exists x\ A| = \exists x\ (N(x) \wedge |A|)$.

Then we prove that HA_N is a conservative extension of $\mathrm{HA}_{\mathit{Pred}}$ in the sense that if A is a closed proposition formed in the language of $\mathrm{HA}_{\mathit{Pred}}$ then A is provable in $\mathrm{HA}_{\mathit{Pred}}$ if and only if $|A|$ is provable in HA_N. Proving that HA_N is an extension of $\mathrm{HA}_{\mathit{Pred}}$ is relatively easy as it just requires to prove that if a proposition A is an axiom of $\mathrm{HA}_{\mathit{Pred}}$ then $|A|$ is provable in HA_N and an induction over proof structure. Proving that the extension is conservative is achieved using the completeness theorem by verifying that all constructive models of $\mathrm{HA}_{\mathit{Pred}}$ extend to models of HA_N.

2.4 A Sort for Classes of Numbers

Finally, we introduce a second sort for classes of natural numbers and use these number classes to express equality and the induction scheme.

Definition 9 ($\mathbf{HA}_{\mathit{Class}}$).
The theory HA_{Class} is a many sorted theory with two sorts ι and κ. The language contains a constant 0 of sort ι, function symbols S and $Pred$ of rank $\langle \iota, \iota \rangle$ and $+$ and $\times$ of rank $\langle \iota, \iota, \iota \rangle$, predicate symbols $=$ of rank $\langle \iota, \iota \rangle$, $Null$ and N of rank $\langle \iota \rangle$ and $\in$ of rank $\langle \iota, \kappa \rangle$ and for each proposition P in the language 0, S, $Pred$, $+$, $\times$, $=$, $Null$ and N and whose free variables are among $x, y_1, \ldots, y_n$ of sort ι, a function symbol $f_{x,y_1,\ldots,y_n,P}$ of rank $\langle \iota, \ldots, \iota, \kappa \rangle$. The symbol $f_{x,y_1,\ldots,y_n,P}$ is written f_P when the variables $x, y_1, ..., y_n$ are clear from the context. The axioms are

$$\forall y \forall z\ (y = z \Leftrightarrow \forall p\ (y \in p \Rightarrow z \in p))$$

$$\forall n\ (N(n) \Leftrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p))$$

$$\forall x \forall y_1 ... \forall y_n\ (x \in f_{x,y_1,\ldots,y_n} P(y_1, \ldots, y_n) \Leftrightarrow P)$$

$$\begin{array}{cc}
Pred(0) = 0 & \forall x\ (Pred(S(x)) = x) \\
Null(0) & \forall x\ (\neg Null(S(x))) \\
\forall y\ (0 + y = y) & \forall x \forall y\ (S(x) + y = S(x + y)) \\
\forall y\ (0 \times y = 0) & \forall y\ (S(x) \times y = x \times y + y)
\end{array}$$

The theory $\mathrm{HA}_{\mathit{Class}}$ is a conservative extension of HA_N. Again, proving that is is an extension is relatively simple, while proving that the extension is conservative requires to prove that prove that all constructive models of HA_N extend to models of $\mathrm{HA}_{\mathit{Class}}$.

The conclusion is the equivalence between HA and $\mathrm{HA}_{\mathit{Class}}$.

Proposition 2. *Let A be a closed proposition in the language of HA. Then A is provable in HA if and only if $|A|$ is provable in HA_{Class}.*

3 Arithmetic in Deduction Modulo

Definition 10 (The theory HA$_{\longrightarrow}$)).
The language of the theory HA$_{\longrightarrow}$ is the same as that of the theory HA$_{Class}$. This theory has no axioms and the rewrite rules

$$y = z \longrightarrow \forall p\ (y \in p \Rightarrow z \in p)$$

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

$$x \in f_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n) \longrightarrow P$$

$$\begin{array}{rlrl} \mathit{Pred}(0) \longrightarrow 0 & & \mathit{Pred}(S(x)) \longrightarrow x \\ \mathit{Null}(0) \longrightarrow \top & & \mathit{Null}(S(x)) \longrightarrow \bot \\ 0 + y \longrightarrow y & & S(x) + y \longrightarrow S(x+y) \\ 0 \times y \longrightarrow 0 & & S(x) \times y \longrightarrow x \times y + y \end{array}$$

Proposition 3. *The theory HA$_{\longrightarrow}$ is a conservative extension of HA.*

Proof. It is equivalent to HA$_{Class}$.

Remark 1. The variant of HA$_{\longrightarrow}$ where the rule

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

is replaced by

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

is also a conservative extension of HA.

We favor the first formulation that allows more natural induction proofs (see Section 6).

4 Cut Elimination

In this section, we build a pre-model to show that HA$_{\longrightarrow}$ has the cut elimination property.

Proposition 4. *The theory HA$_{\longrightarrow}$ has the cut elimination property.*

Proof. We build a pre-model as follows. We take $M_\iota = \mathbb{N}$, $M_\kappa = \mathcal{CR}^{\mathbb{N}}$. The denotations of 0, S, $+$, $\times$, *Pred* are obvious. We take $\hat{\mathit{Null}}(n) = \mathcal{SN}$. The denotation of $\in$ is the function mapping n and f to $f(n)$. Then we can define the denotation of

$$\forall p\ (y \in p \Rightarrow z \in p)$$

and the denotation of $=$ accordingly.

To define the denotation of N, for each function f of $\mathcal{CR}^{\mathbb{N}}$ we can define an interpretation $\mathcal{M}_f$ of the language of the proposition

$$\forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

where the symbol N is interpreted by the function f. We define the function Φ from $\mathcal{CR}^{\mathbb{N}}$ to $\mathcal{CR}^{\mathbb{N}}$ mapping f to the function mapping the natural number x to the candidate

$$[\![\forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)]\!]^{\mathcal{M}_f}_{x/n}$$

The order on $\mathcal{CR}^{\mathbb{N}}$ defined by $f \subseteq g$ if for all n, $f(n) \subseteq g(n)$ is a complete order and the function Φ is monotonous as the occurrence of N is positive in

$$\forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

Hence it has a fixpoint g. We interpret the symbol N by the function g.

Finally, the denotation of the symbols of the form f_P is defined in the obvious way.

This pre-model is a pre-model of each rule of $\mathrm{HA}_{\longrightarrow}$ by construction.

Remark 2. Building a premodel for the variant of $\mathrm{HA}_{\longrightarrow}$ with the rule

$$N(x) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

is even simpler, we do not need to use the fixpoint theorem and we just define the denotation of the proposition $N(n)$ as the denotation of

$$\forall p\ (0 \in p \Rightarrow \forall y\ (y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

5 A Predicative Cut Elimination Proof

The normalization proof of the previous section is essentially obtained by mapping arithmetic into second order arithmetic and then applying the usual normalization proof of second order arithmetic.

This proof is impredicative, indeed, to define the reducibility candidates interpreting the propositions $t = u$ and $N(t)$ we use a quantification over the set M_κ of functions mapping natural number to reducibility candidates.

We shall now see that it is possible to build also a predicative proof. In this proof, we restrict the set M_κ to contain only some functions from natural numbers to candidates, typically definable functions. Then these functions can be replaced by indices, for instance, a natural number and the quantification over functions from natural numbers to candidates can be replaced by a simple quantification over natural numbers. The difficulty here is that to define the denotation of $=$ and N we must use quantification on elements of the set M_κ. To define this set we need to define the notion of definable functions and as the symbols $=$ and N occur in the language, to define this notion we need to use the denotation of the symbols $=$ and N. To break this circularity, we give another definition of the interpretation of $t = u$ and $N(t)$ that does not use quantification over the elements of M_κ. Then the rewrite rules are not valid by construction anymore and we have to check their validity *a posteriori*.

Thus, we shall start by constructing the reducibility candidates E and E' used for interpreting equality and P_n used for interpreting the symbol N.

5.1 The Construction of Some Candidates

Definition 11. *Let A be a set of strongly normalizing terms. The set $[A]$ is inductively defined by*

- *if $\pi \in A$ then $\pi \in [A]$,*
- *if $\pi \in [A]$ and $\pi \rhd^* \pi'$ then $\pi' \in [A]$,*
- *if π is an elimination and all its one step reducts are in $[A]$ then $\pi \in [A]$.*

It is routine to check that if A is a set of strongly normalizing proof-terms, then $[A]$ is the smallest reducibility candidate containing A.

The smallest reducibility candidate $\perp\!\!\!\perp$ can be defined by $\perp\!\!\!\perp = [\emptyset]$. For each strongly normalizing proof-term σ we define C_σ, the smallest reducibility candidate containing σ, by $C_\sigma = [\{\sigma\}]$.

Definition 12.

$$E = \{\pi \in \mathcal{SN} \mid \forall t\ \forall \sigma \in \mathcal{SN}\ (\pi\ t\ \sigma) \in C_\sigma\}$$
$$E' = \{\pi \in \mathcal{SN} \mid \forall t\ \forall \sigma \in \mathcal{SN}\ (\pi\ t\ \sigma) \in \perp\!\!\!\perp\}$$

Let $P = (P_i)_{i \in \mathbb{N}}$ and $Q = (Q_i)_{i \in \mathbb{N}}$ be two sequences of reducibility candidates, recall that the order defined by $P \subseteq Q$ if for all n, $P_n \subseteq Q_n$ is a complete order.

Definition 13. *Let σ_0 and σ_S be two proof terms and P be a sequence of reducibility candidates, we define the family of candidates $C_n^{\sigma_0,\sigma_S,P}$ by induction on n.*

$$C_0^{\sigma_0,\sigma_S,P} = [\{\pi \mid \pi = \sigma_0 \wedge \pi \in \mathcal{SN}\}]$$

$$C_{n+1}^{\sigma_0,\sigma_S,P} = [\{(\sigma_S\ t\ \rho\ \pi) \in \mathcal{SN} \mid \rho \in P_n \wedge \pi \in C_n^{\sigma_0,\sigma_S,P}\}]$$

It is easy to check that if $P \subseteq Q$ then $C_n^{\sigma_0,\sigma_S,P} \subseteq C_n^{\sigma_0,\sigma_S,Q}$.

Definition 14 (P-Peano pair). *A pair of proof-terms $\langle\sigma_0, \sigma_S\rangle$ is called a P-*Peano pair *if*

- *σ_0 is $\mathcal{SN}$,*
- *σ_S is $\mathcal{SN}$ and for every term t, for every natural number n, every proof-term $\rho \in P_n$, and for every proof-term π in $C_n^{\sigma_0,\sigma_S,P}$, the term $(\sigma_S\ t\ \rho\ \pi)$ is SN.*

It is easy to check that if $P \subseteq Q$ then ($\langle\sigma_0, \sigma_S\rangle$ is a P-Peano pair $\Leftarrow$ $\langle\sigma_0, \sigma_S\rangle$ is a Q-Peano pair).

Finally we define a family of candidates $\Phi(P)$.

Definition 15.

$$(\Phi(P))_n = \{\pi \in \mathcal{SN} \mid \forall t \forall \sigma_0 \forall \sigma_S\ \ \langle\sigma_0, \sigma_S\rangle \textit{ is a } P\textit{-Peano pair} \Rightarrow (\pi\ t\ \sigma_0\ \sigma_S) \in C_n^{\sigma_0,\sigma_S,P}\}.$$

It is easy to check that is $P \subseteq Q$ then $\Phi(P) \subseteq \Phi(Q)$, i.e. that the function Φ is monotonous.

As this function is monotonous, it has a least fixpoint. Let $(P_i)_{i \in \mathbb{N}}$ be the least fixpoint of Φ. By definition

$$P_n = \{\pi \in \mathcal{SN} \mid \forall t \forall \sigma_0 \forall \sigma_S\ \langle\sigma_0, \sigma_S\rangle \text{ is a } P\text{-Peano pair} \Rightarrow (\pi\ t\ \sigma_0\ \sigma_S) \in C_n^{\sigma_0,\sigma_S,P}\}.$$

5.2 A Pre-model

As in the impredicative construction, we take $M_\iota = \mathbb{N}$, we interpret the symbols 0, S, $+$, $\times$, *Pred* in the obvious way and we take $\hat{Null}(n) = \mathcal{SN}$. Then, we define the interpretation of the symbols $=$ and N as follows.

$$\hat{=}(n,n) = E$$
$$\hat{=}(n,m) = E' \text{ if } n \neq m$$
$$\hat{N}(n) = P_n$$

Before we define the set M_κ and the interpretation of the symbol $\in$, we introduce a notion of definable function from the set of natural numbers to the set of candidates.

Definition 16 (Definable function). *A function f from $\mathbb{N}$ to $\mathcal{CR}$ is said to be definable if there exists a proposition P in the language of HA$_{\longrightarrow}$ without the symbol $\in$ and an assignment ϕ such that for all n $f(n) = [\![P]\!]_{\phi,n/x}$.*

We then define the set M_κ, as the (countable) set of functions from $\mathbb{N}$ to $\mathcal{CR}$ containing

- definable functions,
- constant functions taking the value C_σ for some proof-term σ,
- and functions mapping k to $C_k^{\sigma_0,\sigma_S,P}$ for some proof-terms σ_0 and σ_S.

Finally, we complete the construction of the pre-model by defining the denotation of $\in$ as the obvious application function and the denotation of the symbols of the form f_P accordingly. The validity of all the rewrite rules of HA$_{\longrightarrow}$ is routine, except that of the rules

$$y = z \longrightarrow \forall p \ (y \in p \Rightarrow z \in p)$$

$$N(x) \longrightarrow \forall p \ (0 \in p \Rightarrow (\forall y \ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p)) \Rightarrow x \in p)$$

each of them requiring a lemma.

Remark 3. **(Making the proof predicative).** The pre-model construction as presented above is not obviously predicative since to define the reducibility candidate associated to proposition $\forall p \ A$ we use quantification over M_κ that is a set of functions mapping natural numbers to reducibility candidates. However as the set M_κ is countable, it is not difficult to associate a natural number to each of its elements and to define a function U that maps each number to the associated function. Then we can replace M_κ by $\mathbb{N}$ and define the interpretation of $\in$ as the function mapping n and m to $U(m)(n)$. The construction obtained this way is predicative. For instance, it could be formalized in Martin-Löf's Type Theory with one universe.

Remark 4. For the variant of $\text{HA}_{\longrightarrow}$ with the rule

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

the proof is simpler as we do not need to apply the fixpoint theorem. If σ_0 and σ_S be two proof terms, we define the family of candidates $C_n^{\sigma_0,\sigma_S,P}$ by induction on n without the parameter P. Peano pairs and the family P_n can be defined directly and the rest of the proof is similar.

6 The System T

More traditional cut elimination proofs for arithmetic use the normalization of Gödel system T. We show here that the normalization of system T also can be obtained as a corollary of the normalization theorem of [3] although the system T contains a specific rewrite rule on proofs and [3] allows only specific rewrite rules on terms and propositions but uses fixed rewrite rules on proofs.

Consider the symbol $nat = f_{N(x)}$ and $\rightarrow = f_{x \in y \Rightarrow x \in z}$. In $\text{HA}_{\longrightarrow}$, we have

$$x \in nat \longrightarrow N(x)$$

$$x \in (y \rightarrow z) \longrightarrow x \in y \Rightarrow x \in z$$

and of course

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

We can drop the first rule, replacing all propositions of the form $N(x)$ the proposition $x \in nat$ and we get this way the rewrite system with two rules

$$n \in nat \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (y \in nat \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

$$x \in (y \rightarrow z) \longrightarrow x \in y \Rightarrow x \in z$$

In this system, we get rid of all terms of type ι. We get the following theory

Definition 17 (The theory $\mathcal{T}$).

$$\varepsilon(nat) \longrightarrow \forall p\ (\varepsilon(p) \Rightarrow (\varepsilon(nat) \Rightarrow \varepsilon(p) \Rightarrow \varepsilon(p)) \Rightarrow \varepsilon(p))$$

$$\varepsilon(y \rightarrow z) \longrightarrow \varepsilon(y) \Rightarrow \varepsilon(z)$$

Proposition 5. *The theory $\mathcal{T}$ has the cut elimination property.*

The proof is structurally similar to the one of Section 5.

Definition 18 (The system T). *The system T is the extension of simply typed lambda-calculus with a constant* 0*, a unary function symbol S and a ternary function symbol Rec^A for each type A and the rules*

$$Rec(a, f, 0) \longrightarrow a$$

$$Rec(a, f, S(b)) \longrightarrow (f\ b\ Rec(a, f, b))$$

Proof normalization for the theory $\mathcal{T}$ implies normalization for the system T. Indeed, types of the system T are terms of the theory $\mathcal{T}$ and terms of type A in the system T can be translated into proofs of $\varepsilon(A)$ in the theory $\mathcal{T}$ (Parigot's numbers [11]):

- $|x| = x$, $|u\ v| = |u|\ |v|$, $|\lambda x : A\ u| = \lambda x : \varepsilon(A)\ |u|$,
- $|0| = \lambda p\ \lambda x : \varepsilon(p)\ \lambda f : \varepsilon(nat) \Rightarrow \varepsilon(p) \Rightarrow \varepsilon(p)\ x$,
- $|S(n)| = \lambda p\ \lambda x : \varepsilon(p)\ \lambda f : \varepsilon(nat) \Rightarrow \varepsilon(p) \Rightarrow \varepsilon(p)\ (f\ |n|\ (|n|\ p\ x\ f))$,
- $|Rec^A(x, f, n)| = (|n|\ A\ x\ f)$.

It is routine to check that if $t \longrightarrow^1 u$ in the system T then $|t| \longrightarrow^+ |u|$ in the theory $\mathcal{T}$. For instance:

$$|Rec^A(x, f, 0)| = (|0|\ A\ x\ f) = (\lambda p\ \lambda x : \varepsilon(p)\ \lambda f : \varepsilon(nat) \Rightarrow \varepsilon(p) \Rightarrow \varepsilon(p)\ x)\ A\ x\ f \longrightarrow^+ x.$$

Here we reap the benefit of having chosen the rule

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (N(y) \Rightarrow y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

and not

$$N(n) \longrightarrow \forall p\ (0 \in p \Rightarrow \forall y\ (y \in p \Rightarrow S(y) \in p) \Rightarrow n \in p)$$

that would have given us only the termination of the variant of system T where the recursor is replaced by an iterator.

References

1. M. Crabbé. Non-normalisation de la théorie de Zermelo. Manuscript (1974).
2. G. Dowek, Th. Hardin, and C. Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31 (2003) pp. 33-72.
3. G. Dowek and B. Werner. Proof normalization modulo, *The Journal of Symbolic Logic*, 68, 4 (2003) pp. 1289-1316.
4. G. Dowek and B. Werner. Arithmetic as a theory modulo. Manuscript (2004).
5. G. Dowek and B. Werner. A constructive proof of Skolem theorem for constructive logic, Manuscript (2004).
6. G. Dowek. *La part du Calcul.* Habilitation thesis, Université de Paris 7 (1999).
7. J.Y. Girard. Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur, *Thèse d'État*, Université de Paris 7 (1972).
8. J.Y. Girard, Y. Lafont and P. Taylor. *Proofs and Types*, Cambridge University Press (1989).
9. K. Gödel. Über eine bisher noch nicht benüzte Erweiterung des finiten Standpunktes, *Dialectica*, 12 (1958) pp. 280-287. Reproduced in S. Feferman *et al.* (eds.), *Collected Works*, vol. II, Oxford University Press (1990) pp. 241-251.
10. J.-W. Klop, V. van Oostrom and F. van Raamsdonk. Combinatory reduction systems: introduction and survey. *Theoretical Computer Science*, 121, (1993) pp. 279-308.

11. M. Parigot. Programming with proofs: A second order type theory. *European Symposium on Programming*, H. Ganzinger (ed.), Lecture Notes in Computer Science, 300, (1988) pp. 145-159.
12. D. Prawitz, Natural deduction. A proof-theoretical study. Almqvist & Wiksell (1965).
13. H. Rasiowa and R. Sikorski, *The mathematics of metamathematics*, Polish Scientific Publishers (1963).