

# Algorithmic equality in Heyting Arithmetic Modulo

Lisa Allali

LogiCal - Ecole polytechnique - INRIA,  
www.lix.polytechnique.fr/Labo/Lisa.Allali/  
allali@lix.polytechnique.fr

## 1 Introduction

We present in this paper a version of Heyting arithmetic where all the axioms are dropped and replaced by rewrite rules. A previous work has been done by Gilles Dowek and Benjamin Werner presenting Heyting Arithmetic in such a way [3], but where equality was defined by a “Leibniz rule” : a proposition of the form  $x = y$  was rewritten in their system into  $\forall p (x \in p \Rightarrow y \in p)$ , that is provable if  $x$  and  $y$  are two equal closed terms, but not as simply as it could be expected. In this paper, in contrary, when  $x$  and  $y$  are closed terms, we considere checking equality between terms is just a computation :  $x = y$  rewrites directly to  $\top$  or  $\perp$ .

We followed a remark of Schwichtenberg, about how a set of rewrite rules could be (or not) enough to decide equality in Heyting Arithmetic. In the work we present here, we answer positively to this question and present a set of rewrite rules that define a new Heyting Arithmetic modulo  $\text{HA}_{\equiv}$ , that is

- an extension of axiomatic Heyting Arithmetic : all the theorems of arithmetic and, in particular, all instances of Leibniz’ scheme can be proved in  $\text{HA}_{\equiv}$
- this extension is conservative with respect to a very simple translation
- all proofs of  $\text{HA}_{\equiv}$ , strongly normalize.

This work suggests new ways to consider equality of inductive types in general, not anymore with Leibniz’s axiom as it is the case in Coq for instance, but building specific rewrite rules to define equality in an algorithmic way.

## 2 Definitions

### 2.1 Deduction Modulo

Modern type theories feature a rule called *conversion rule* which allows to identify proposition which are *equal modulo beta-equivalence*. It is often presented as follows :

$$\frac{\Gamma \vdash t : T \quad \Gamma \vdash T : \text{Type} \quad \Gamma \vdash T' : \text{Type}}{\Gamma \vdash t : T'} T \equiv_{\beta} T'$$

where  $T \equiv_{\beta} T'$  is read  $T$  is convertible to  $T'$ .

This convertibility is not checked by logical rules but by *computation* with the rule  $\beta$ . The idea of natural deduction modulo is to "import" this computation of convertibility inside the natural deduction but replacing  $\equiv_{\beta}$  by an arbitrary congruence  $\equiv$  defined by a confluent rewrite system. For instance, the axiom rule and the  $\Rightarrow$  elimination rules are the following :

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

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

The other rules of natural deduction modulo are built the same way upon natural deduction [2].

The convertibility  $\equiv$  is not fixed. It can be any congruence defined by the reflexive, symmetric and transitive closure of a rewrite system which is confluent, rewrites term to term and atomic proposition to proposition.

## 2.2 Theories in natural deduction modulo

### Definition 1 (Axiomatic theory)

An axiomatic theory is a set of axioms.

### Definition 2 (Modulo theory)

A modulo theory is a set of axioms and a congruence defined as the reflexive, transitive and symmetric closure of a set of rewrite rules.

### Definition 3 (Purely computational theory)

A purely computational theory is a modulo theory where the set of axioms is empty.

## 3 Heyting Arithmetic - from axioms to rewrite rules

### 3.1 The axiomatic presentation of Heyting Arithmetic

The language of arithmetic formed by the functional symbols 0 of arity 0,  $S$  of arity 1,  $+$  and  $\times$  of arity 2. The predicate symbol  $=$  of arity 2. The axioms are structured in four groups as follow :

The axioms of equality

*Reflexivity*  $\forall x x = x$       *Leibniz' axiom scheme*  
 $\forall x \forall y x = y \Rightarrow P(x) \Leftrightarrow P(y)$

The axioms 3 and 4 of Peano

$\forall x \forall y (S(x) = S(y) \Rightarrow x = y)$        $\forall x 0 = S(x) \Rightarrow \perp$

The induction scheme

$(P\{x := 0\} \wedge \forall y (P\{x := y\} \Rightarrow P\{x := S(y)\})) \Rightarrow \forall n P\{x := n\}$

The axioms of addition and multiplication.

$\forall y (0 + y = y)$        $\forall x \forall y (S(x) + y = S(x + y))$   
 $\forall y (0 \times y = 0)$        $\forall x \forall y (S(x) \times y = x \times y + y)$

### 3.2 The steps to go from an axiomatic theory of Heyting Arithmetic (HA) to a purely computational one

We shall introduce four successive theories to reach the final purely computational theory we aim at, each of them being an equivalent or conservative extension of HA.

#### 3.2.1 $\mathbf{HA}_R$

Induction axiom scheme

$(P\{x := 0\} \wedge \forall y (P\{x := y\} \Rightarrow P\{x := S(y)\})) \Rightarrow \forall n P\{x := n\}$

Rewrite rules

$0 = 0 \longrightarrow \top$        $0 + y \longrightarrow y$   
 $0 = S(x) \longrightarrow \perp$        $S(x) + y \longrightarrow S(x + y)$   
 $S(x) = 0 \longrightarrow \perp$        $0 \times y \longrightarrow 0$   
 $S(x) = S(y) \longrightarrow x = y$        $S(x) \times y \longrightarrow x \times y + y$

The axioms of addition and multiplication are transformed in a very intuitive way into rewrite rules : for instance, the axiom  $\forall x 0 + x = x$  becomes the rewrite rule  $0 + x \longrightarrow x$ . We keep the induction axiom scheme. We drop the Leibniz axiom and add 4 rewrite rules to define equality. We prove this theory is equivalent to HA.

### 3.2.2 $\mathbf{HA}_N$

|                                                                                                                                               |                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Induction axiom scheme                                                                                                                        |                                                |
| $\forall n N(n) \Rightarrow (P\{x := 0\} \wedge \forall y (N(y) \Rightarrow P\{x := y\} \Rightarrow P\{x := S(y)\})) \Rightarrow P\{x := n\}$ |                                                |
| Axioms for $N$                                                                                                                                |                                                |
| $N(0)$                                                                                                                                        | $\forall x N(x) \Rightarrow N(S(x))$           |
| Rewrite rules                                                                                                                                 |                                                |
| $0 = 0 \longrightarrow \top$                                                                                                                  | $0 + y \longrightarrow y$                      |
| $0 = S(x) \longrightarrow \perp$                                                                                                              | $S(x) + y \longrightarrow S(x + y)$            |
| $S(x) = 0 \longrightarrow \perp$                                                                                                              | $0 \times y \longrightarrow 0$                 |
| $S(x) = S(y) \longrightarrow x = y$                                                                                                           | $S(x) \times y \longrightarrow x \times y + y$ |

We introduce a new predicate symbol  $N$  for the natural numbers and three axioms to define it. We prove this theory is a conservative extension of  $\mathbf{HA}_R$  modulo a certain translation .

### 3.2.3 $\mathbf{HA}_K$

|                                                                                                                                                    |                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Comprehension scheme                                                                                                                               |                                                |
| $\forall x \forall y_1 \dots \forall y_n (x \in f_{z, y_1, \dots, y_n, P}(y_1, \dots, y_n) \Leftrightarrow P\{z := x\})$                           |                                                |
| Induction axiom                                                                                                                                    |                                                |
| $\forall n (N(n) \Leftrightarrow \forall f (0 \in f \Rightarrow \forall y (N(y) \Rightarrow y \in f \Rightarrow S(y) \in f) \Rightarrow n \in f))$ |                                                |
| Rewriting rules                                                                                                                                    |                                                |
| $0 = 0 \longrightarrow \top$                                                                                                                       | $0 + y \longrightarrow y$                      |
| $0 = S(x) \longrightarrow \perp$                                                                                                                   | $S(x) + y \longrightarrow S(x + y)$            |
| $S(x) = 0 \longrightarrow \perp$                                                                                                                   | $0 \times y \longrightarrow 0$                 |
| $S(x) = S(y) \longrightarrow x = y$                                                                                                                | $S(x) \times y \longrightarrow x \times y + y$ |

We sort our theory with two sorts  $\iota$  and  $\kappa$ . We add an infinite number of function symbols of the form  $f_{z, y_1, \dots, y_n, P}$ , one for each proposition  $P$  that can be expressed in the language of  $\mathbf{HA}_N$ , where the free variables of  $P$  are  $y_1, \dots, y_n$ . Each of those symbols is of rank  $\langle \iota, \dots, \iota, \kappa \rangle$ . We add a symbol  $\in$  of rank  $\langle \iota, \kappa \rangle$ . Finally we add an axiom scheme expressing the equivalence of the proposition  $x \in f_{z, y_1, \dots, y_n, P}(y_1, \dots, y_n)$  with  $P$ . Notice that the free variables of these propositions are the same. The induction axiom scheme is replaced by a single axiom. We prove this theory is a conservative extension of  $\mathbf{HA}_N$ .

### 3.2.4 $\mathbf{HA}_\rightarrow$

We transform the remaining axioms of  $\mathbf{HA}_N$  into rewrite rules.  $\mathbf{HA}_\rightarrow$  is a purely computational presentation of Heyting Arithmetic.

### 3.3 $\text{HA}_{\rightarrow}$ , a purely computational presentation of Heyting Arithmetic

**Definition 4** ( $\text{HA}_{\rightarrow}$ )

|                                                                                                                                        |                                                |
|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| $x \in f_{z, y_1, \dots, y_n, P}(y_1, \dots, y_n) \longrightarrow P\{z := x\}$                                                         |                                                |
| $N(n) \longrightarrow \forall f (0 \in f \Rightarrow \forall y (N(y) \Rightarrow y \in f \Rightarrow S(y) \in f) \Rightarrow n \in f)$ |                                                |
| $0 = 0 \longrightarrow \top$                                                                                                           | $0 + y \longrightarrow y$                      |
| $0 = S(x) \longrightarrow \perp$                                                                                                       | $S(x) + y \longrightarrow S(x + y)$            |
| $S(x) = 0 \longrightarrow \perp$                                                                                                       | $0 \times y \longrightarrow 0$                 |
| $S(x) = S(y) \longrightarrow x = y$                                                                                                    | $S(x) \times y \longrightarrow x \times y + y$ |

We introduce the following **Translation**  $|\cdot|$  from  $\text{HA}$  to  $\text{HA}_{\rightarrow}$

$$\begin{aligned}
|P| &= P, \text{ if } P \text{ is atomic,} \\
|\top| &= \top, \\
|\perp| &= \perp, \\
|A \wedge B| &= |A| \wedge |B|, \\
|A \vee B| &= |A| \vee |B|, \\
|A \Rightarrow B| &= |A| \Rightarrow |B|, \\
|\forall x A| &= \forall x (N(x) \Rightarrow |A|), \\
|\exists x A| &= \exists x (N(x) \wedge |A|)
\end{aligned}$$

**Proposition 1**

$\text{HA}_{\rightarrow}$  is a conservative extension of  $\text{HA}$ , i.e. for all closed propositions  $A$ ,

$$\vdash_{\text{HA}} A \quad \text{if and only if} \quad \vdash_{\text{HA}_{\rightarrow}} |A|$$

We prove that each of the theories  $\text{HA}$ ,  $\text{HA}_R$ ,  $\text{HA}_N$ ,  $\text{HA}_K$ ,  $\text{HA}_{\rightarrow}$  is a conservative extension of the previous one.

The main difficulty lies in the first step: proving that  $\text{HA}_R$  is an extension of  $\text{HA}$ , and more specifically that the Leibniz's axiom scheme of  $\text{HA}$  is derivable in  $\text{HA}_R$ . This requires to prove successively the following properties of  $\text{HA}_R$  equality:

$$\begin{aligned}
&\forall x (x = x) \\
&\forall x \forall y (x = y \Rightarrow y = x) \\
&\forall x \forall y \forall z (x = y \Rightarrow y = z \Rightarrow x = z) \\
&\forall x \forall y \forall z x = y \Rightarrow x + z = y + z \\
&\forall x \forall y \forall z x = y \Rightarrow z + x = z + y \\
&\forall x \forall y \forall z x = y \Rightarrow x \times z = y \times z \\
&\forall x \forall y \forall z x = y \Rightarrow z \times x = z \times y \\
&\forall x x \times 0 = 0 \\
&\forall y \forall x (y \times S(x) = y \times x + y) \\
&\forall x \forall y (x \times y = y \times x)
\end{aligned}$$

Thanks to those properties of equality in  $\text{HA}_R$ , we can prove by induction on  $t$  that for each term  $t$ , the proposition  $\forall a \forall b (a = b \Rightarrow t\{y := a\} = t\{y := b\})$  is provable in  $\text{HA}_R$ .

This proposition is the basic case of an induction on  $P$  we made to prove that each instance of Leibniz' scheme

$$\forall x \forall y x = y \Rightarrow P(x) \Leftrightarrow P(y)$$

is provable in  $\text{HA}_R$ .

Finally we also prove two other results :

**Proposition 2**

*The congruence defined by the rewrite rules of  $\text{HA}_{\rightarrow}$  is decidable.*

**Proposition 3**

*$\text{HA}_{\rightarrow}$  has cut elimination property.*

## 4 Discussion

One can ask if this system is really efficient in practice: in one hand, the proof of  $x = y$  are shorter, in the other hand the proof of  $x = y \Rightarrow P(x) \Rightarrow P(y)$  is longer. There is no theoretical answer to that question, it's only by making tests that we would see how the size of proof terms would change. An good indication is that the way we manage to "simulate" an application of Leibniz principle with our rewrite rules (the way it is shown in [1]) is linear in the size of the proposition.

## 5 Conclusion

We have reached a presentation of Heyting Arithmetic without any axiom, simply defined by a rewrite rule system. A cornerstone of this presentation is that it makes use of the decidability of the equality in Heyting Arithmetic, indeed the equality is *defined* as a decision procedure, rather than as Leibniz's proposition which becomes a consequence of the congruence of the system.

## References

- [1] Lisa Allali, Memoire de DEA, [http://www.lix.polytechnique.fr/Labo/Lisa.Allali/rapport\\_MPRI.pdf](http://www.lix.polytechnique.fr/Labo/Lisa.Allali/rapport_MPRI.pdf).
- [2] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31:32–72, 2003.
- [3] Gilles Dowek and Benjamin Werner. Arithmetic as a theory modulo. J. Giesel (Ed.), *Term rewriting and applications (RTA)*, Lecture Notes in Computer Science 3467, Springer-Verlag, 2005, pp. 423-437.
- [4] Gilles Dowek. Truth values algebras and normalization, to appear in TYPES 2006, 2007
- [5] Gilles Dowek. La part du calcul. *Mémoire d'Habilitation à Diriger des Recherches*, Université Paris 7, 1999.
- [6] Vincent van Oostrom, Femke van Raamsdonk. Weak Orthogonality Implies Confluence : The High-Order Case. *Technical Report: ISRL-94-5*, December, 1994.
- [7] The Coq Development Team . Manuel de Référence de Coq V8.0. <http://coq.inria.fr/doc/main.html>, LogiCal Project, 2004-2006