# EFFICIENTLY SIMULATING HIGHER-ORDER ARITHMETIC BY A FIRST-ORDER THEORY MODULO

GUILLAUME BUREL

Max Planck Institute for Informatics, Saarland University, Saarbrücken, Germany
*e-mail address*: guillaume.burel@ens-lyon.org

ABSTRACT. In deduction modulo, a theory is not represented by a set of axioms but by a congruence on propositions modulo which the inference rules of standard deductive systems—such as for instance natural deduction—are applied. Therefore, the reasoning that is intrinsic of the theory does not appear in the length of proofs. In general, the congruence is defined through a rewrite system over terms and propositions. We define a rigorous framework to study proof lengths in deduction modulo, where the congruence must be computed in polynomial time. We show that even very simple rewrite systems lead to arbitrary proof-length speed-ups in deduction modulo compared to using axioms. As higher-order logic can be encoded as a first-order theory in deduction modulo, we also study how to reinterpret, thanks to deduction modulo, the speed-ups between higher-order and first-order arithmetics that were stated by Gödel. We define a first-order rewrite system with a congruence decidable in polynomial time such that proofs of higher-order arithmetic can be linearly translated into first-order arithmetic modulo that system. We also present the whole higher-order arithmetic as a first-order system without resorting to any axiom, where proofs have the same length as in the axiomatic presentation.

## 1. INTRODUCTION

The study of the length of the proofs produced by as logical system is of course interesting from a practical point of view. Indeed, shorter proofs seem to be easier to find out—either by hand or automatically—, to share and to maintain. Automated provers may be able to find proofs that are longer than proofs done by humans, they have nevertheless bounded capacities. Even if computing power is always increasing, so that one is no longer afraid to use SAT-solvers within verification tools (mainly because worst cases do not often occur in practice), it is not conceivable to build an automated theorem prover that produces only proofs of non-elementary length.

This study has also a theoretical interest. As remarked by Parikh in the introductory paragraph of Gödel (1986), "the celebrated P=NP? question can itself be thought of as a speed-up question." (See also Cook and Reckhow, 1979.) This explains the research

---

for speed-ups between proof systems—for instance, it is shown that Frege systems have an exponential speed-up over resolution for propositional logic (Buss, 1987)—and for new formalisms whose deductive systems provide smaller proofs, such as for instance the calculus of structures of Brünnler (2003) w.r.t. the sequent calculus of Gentzen (1934) (see Bruscoli and Guglielmi, 2009). The goal is to find a so-called super proof system (Cook and Reckhow, 1974), which can build polynomially sized proofs of each propositional tautology, or to refute the existence of such a system, in which case NP≠coNP, which would imply P≠NP. In this paper, the length of a proof corresponds to its number of steps (sometimes called lines), whatever the actual size of the propositions appearing in them is.

Proofs are rarely searched for without context: mathematical proofs rely on set theory, or Euclidean geometry, or arithmetic, etc.; proofs of program correctness are done using e.g. pointer arithmetic and/or theories defining data structures (chained lists, trees, . . . ); concerning security, theories are used for instance to model properties of encryption algorithms. In this paper, we are therefore interested in the length of proofs in a theory. This length may depend on several factors. First, the strength of the theory plays a key role, as shown by the following result: it has been proved by Parikh (1973) that second-order arithmetic provides shorter proofs than first-order arithmetic. (This result was stated earlier by Gödel (1936), unfortunately without proof.) This was generalized to all orders by Krajíček (1989), and was proved for the true language of arithmetic by Buss (1994). (The former results used an axiomatization of arithmetic using ternary predicates to represent addition and multiplication.) The theorem proved by Buss is stated as follow:

**Theorem 1.1** (Buss (1994, Theorem 3)). *Let $i \geq 0$. Then there is an infinite family $\mathcal{F}$ of $\prod_1^0$-propositions such that*

  (1) *for all $P \in \mathcal{F}$, $Z_i \vdash P$*
  (2) *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_{i+1} \vdash_{\overline{k \ steps}} P$*
  (3) *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i \vdash_{\overline{k \ steps}} P$.*

where $Z_i$ corresponds to the $i + 1^{\text{st}}$-order arithmetic (so $Z_0$ is in fact first-order arithmetic), and $Z_i \vdash_{\overline{k \ steps}} P$ means that $P$ can be proved in at most $k$ steps within a schematic system —i.e. a Hilbert-type (or Frege) system with a finite number of axiom schemata and inference rules— for $i + 1^{\text{st}}$-order arithmetic. (In fact, Buss proved this theorem also for weakly schematic systems, i.e. schematic systems in which every tautology can be used as an axiom, as well as generalizations of axioms, but we will not use this fact here.)

The length of the proofs depends also on the presentation of the theory. For instance, if we present the theory $\mathcal{T}$ by all the propositions that holds in that theory ($\{P : \mathcal{T} \models P\}$), then for all true propositions $P$ there is a one-step "proof", namely using the axiom $P$. Of course, we can argue whether those are really proofs. Indeed, in that case, proof checking consists in checking that $P$ holds in $\mathcal{T}$, and is therefore undecidable in general. On the other hand, using a finite first-order axiomatization of the theory does not seem optimal, in particular when computations are involved. For instance, a proof of $2 + 2 = 4$ should be straightforward and should not contain more than one step that consists in checking than computing $2 + 2$ makes 4. Then, it seems important to distinguish what part of a proof corresponds to computation and what part is real deduction, so as to better combine them. Such an idea is referred to as Poincaré's principle. Deduction modulo (Dowek, Hardin, and Kirchner, 2003) is a formalism deriving from this principle. The computational part of a proof is put in a congruence between propositions modulo which the application of the deduction rules takes place. This leads for instance to the sequent calculus modulo and to

the natural deduction modulo. The congruence is often defined as a set of rewrite rules that can rewrite terms but also *atomic propositions*: indeed, one wants for instance to consider the definition of the addition or multiplication using rewrite rules over terms as part of the computation, but also the following rewrite rule:

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

which rewrites an atomic proposition to a proposition, so that the following simple natural-deduction-modulo proof of $t \times t = 0$ can be deduced from a proof $\pi$ of $t = 0$:

$$\vee\text{-i} \ \frac{\dfrac{\pi}{t = 0}}{t \times t = 0} \ t \times t = 0 \longrightarrow t = 0 \vee t = 0 \quad .$$

Proposition rewriting is essential to be able to encode expressive theories in deduction modulo, as was done for first-order arithmetic (Dowek and Werner, 2005), Zermelo's set theory (Dowek and Miquel, 2006), simple type theory (a.k.a. higher-order logic) (Dowek, Hardin, and Kirchner, 2001) or pure type systems (Cousineau and Dowek, 2007; Burel, 2008).

As computations are not part of the deduction in the proof, they should not be counted in the length of the proof. Indeed, a proof in deduction modulo consists only of the deductive steps, and the computational steps are replayed during proof checking. However, this is too general if we are concerned with the notion of proof length. Because rewriting is Turing-complete, a whole proof system can be encoded in the computational part. This leads to the same problem as using all propositions of the theory as axioms: proof checking is no longer decidable. We therefore need a more rigorous framework to study proof length in deduction modulo. We argue that we should only call a proof an object that can be checked feasibly, that is, in polynomial time. This is of course an arbitrary criterion (we could for instance have chosen another complexity class), but it seems natural. Furthermore, this is requested if one wants to link proof theory with complexity theory. Indeed, Cook and Reckhow (1979) defined a framework in which a proof system for a theory $\mathcal{T}$ is an onto function computable in polynomial time from the words over some alphabet (representing the proofs) to the set of propositions that hold in $\mathcal{T}$. Starting from a more conventional proof system, the idea is to map a correct proof with its conclusion, and an incorrect proof to any proposition of $\mathcal{T}$. As the function must be computable in polynomial time, proof checking in the real system has to be feasible. In deduction modulo, this requirement implies that the congruence must be checkable in polynomial time. In this paper, we will consider rewrite systems that are confluent and that have a polynomial derivational complexity, i.e. the number of rewrite steps of a term of size $n$ must be bounded by a polynomial of $n$.

Deduction modulo is logically equivalent to the axiomatic theory corresponding to the congruence (Dowek et al., 2003, Proposition 1.8), but proofs are often considered as simpler, because the computation is hidden, letting the deduction clearly appear. Proofs are also claimed to be shorter for the same reason. Nevertheless, this fact was never quantified. Besides, it is possible, in deduction modulo, to build proofs of Higher-Order Logic using a first-order system (Dowek et al., 2001). Using this, a step of higher-order resolution is completely simulated by a step of ENAR, the resolution and narrowing method based on deduction modulo. It looks like this is also the case for the associated sequent calculi, although this was not clearly stated. Therefore, it seems reasonable to think that deduction modulo is able to give the same proof-length speed-ups as the ones occurring between $i+1^{\text{st}}$- and $i^{\text{th}}$-order arithmetic. This paper therefore investigates how to relate proof-length speed-ups in arithmetic with the computational content of the proofs.

Our first result is to show that even a very simple rewrite system can lead to arbitrary proof-length speedups (Theorem 2.3). Then, we show how to encode everything concerning higher orders up to $i > 0$ into a confluent rewrite system $\mathcal{HO}_i$ with polynomial derivational complexity. Modulo this rewrite system, we show that it is possible to stay in first-order arithmetic while preserving the proof lengths of higher-order arithmetic (Theorem 4.5). This shows that the origin of the speed-up theorem of Buss can be, at least to some extend, expressed as simple computations. Note that $\mathcal{HO}_i$ is not the restriction of the encoding of HOL by Dowek et al. (2001) up to the order $i$, because we were not able to prove that its derivational complexity is bounded.

In this paper, we are also concerned with extending the work of Dowek and Werner (2005), in which the whole first-order arithmetic is expressed as a rewrite system. In that case, we speak of a purely computational presentation of the theory. Thus, we show how to express *higher-order* arithmetic as a purely computational theory. This permits to recover desirable properties such as disjunction and witness properties for higher-order Heyting arithmetic (i.e. intuitionistic arithmetic). This is not just the combination of the encoding of higher orders and the formulation of first-order arithmetic by Dowek and Werner (2005), because the latter does not preserve the length of proofs. We define higher-order arithmetic as a purely computational theory $\mathcal{HHA}_i^{\mathrm{mod}}$ which has the same speed-up over first-order arithmetic as the axiomatic presentation. Unfortunately, the rewrite system of this presentation is not terminating. The rule that breaks the termination is the one encoding the induction principle, which is not surprising, because this is where the strength of arithmetic lies. We therefore advocate the use of a new inference rule corresponding to it.

In 2007, we also looked at the relations between computations and proof-length speedups. We work in a much more rigorous framework here. For instance, in 2007, we only stated that the rewrite systems we were using were "simple", whereas we request here that they are confluent and with a polynomially bounded derivational complexity. Also, in 2007, in the translation of $Z_i$ to $Z_{i-1}$ modulo, there remained axioms in which function symbols of order $i$ were involved, which is no longer the case here.

The next section will present the minimal knowledge needed on deduction modulo to make the paper self-contained, it defines the notion of polynomially bounded derivational complexity, and shows that arbitrary proof-length speed-ups naturally occur thanks to deduction modulo, even for very simple rewrite system with polynomially bounded derivational complexity. In Section 3 we present proof systems for higher-order arithmetic, and we prove that using schematic systems or natural deduction is not relevant w.r.t. arbitrary proof-length speed-ups. Then, Section 4 presents how to efficiently encode higher orders, and then higher-order arithmetic. Finally, in Section 5 we apply these results to investigate the origin of the speed-ups in arithmetic.

## 2. Proof Speed-ups in Deduction Modulo

2.1. **Rewriting propositions.** In this section, we recall the definition of deduction modulo, as introduced by Dowek et al. (2003) and Dowek and Werner (2003). In deduction modulo, propositions are considered modulo some congruence defined by some rules that rewrite not only terms but also propositions. We use standard definitions, as given by Baader and Nipkow (1998), and extend them to proposition rewriting (Dowek et al., 2003).

First, let us recall how to build many-sorted first-order propositions (see Gallier, 1986, Chapter 10), mainly to introduce the notations we will use. A (first-order) many-sorted signature consists of a set of function symbols and a set of predicates, all of them with their arity (and co-arity for function symbols). We denote by $\mathcal{T}(\Sigma, V)$ the set of *terms* built from a signature $\Sigma$ and a set of variables $V$. An *atomic proposition* is given by a predicate symbol $A$ of arity $[i_1, \ldots, i_n]$ and by $n$ terms $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, V)$ with matching sorts. It is denoted $A(t_1, \ldots, t_n)$. *propositions* can be built using the following grammar[1]:

$$\mathcal{P} \stackrel{!}{=} \bot \mid A \mid \mathcal{P} \wedge \mathcal{P} \mid \mathcal{P} \vee \mathcal{P} \mid \mathcal{P} \Rightarrow \mathcal{P} \mid \forall x.\ \mathcal{P} \mid \exists x.\ \mathcal{P}$$

where $A$ ranges over atomic propositions and $x$ over variables. $P \Leftrightarrow Q$ will be used as a syntactic sugar for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$, as well as $\neg P$ for $P \Rightarrow \bot$. Positions in a term or a proposition, free variables and substitutions are defined as usual (see Baader and Nipkow, 1998). The replacement of a variable $x$ by a term $t$ in a proposition $P$ is denoted by $\{t/x\}P$, the subterm or subproposition of $t$ at the position $\mathfrak{p}$ by $t_{|\mathfrak{p}}$, and its replacement in $t$ by a term or proposition $s$ by $t[s]_{\mathfrak{p}}$.

A *term rewrite rule* is the pair of terms $l, r$ such that all free variables of $r$ appear in $l$. It is denoted $l \rightarrow r$. A *term rewrite system* is a set of term rewrite rules. A term $s$ can be rewritten to a term $t$ by a term rewrite rule $l \rightarrow r$ if there exists some substitution $\sigma$ and some position $\mathfrak{p}$ in $s$ such that $\sigma l = s_{|\mathfrak{p}}$ and $t = s[\sigma r]_{\mathfrak{p}}$. An atomic proposition $A(s_1, \ldots, s_i, \ldots, s_n)$ can be rewritten to the atomic proposition $A(s_1, \ldots, t_i, \ldots, s_n)$ by a term rewrite rule $l \rightarrow r$ if $s_i$ can be rewritten to $t_i$ by $l \rightarrow r$. This relation is extended by congruence to all propositions.

A *proposition rewrite rule* is the pair of an atomic proposition $A$ and a proposition $P$, such that all free variables of $P$ appear in $A$. It is denoted $A \rightarrow P$. A *proposition rewrite system* is a set of proposition rewrite rules. A proposition $Q$ can be rewritten to a proposition $R$ by a proposition rewrite rule $A \rightarrow P$ if there exists some substitution $\sigma$ and some position $\mathfrak{p}$ in $Q$ such that $\sigma A = Q_{|\mathfrak{p}}$ and $R = Q[\sigma P]_{\mathfrak{p}}$. Semantically, this proposition rewrite relation must be seen as a logical equivalence between propositions.

A *rewrite system* is the union of a term rewrite system and a proposition rewrite system. The fact that $P$ can be rewritten to $Q$ either by a term or by a proposition rewrite rule of a rewrite system $\mathcal{R}$ will be denoted by $A \xrightarrow{\mathcal{R}} P$. The transitive (resp. reflexive transitive) closure of this relation will be denoted by $\xrightarrow[\mathcal{R}]{*}$ (resp. $\xleftrightarrow[\mathcal{R}]{*}$).

The *derivational length* of a term or proposition $t$ w.r.t. a rewrite system $\mathcal{R}$ is the maximal length of a derivation starting from $t$ using $\mathcal{R}$. The *derivational complexity* of a rewrite system $\mathcal{R}$ is the function that maps a natural number $n$ to the maximal derivational length w.r.t. $\mathcal{R}$ of the terms and propositions of size at most $n$. In this paper, we are interested in rewrite systems that are confluent and whose derivational complexity can be bounded by a polynomial. This implies that $\xleftrightarrow[\mathcal{R}]{*}$ is decidable in polynomial time.

2.2. **Natural deduction modulo.** Using some equivalence $\xleftrightarrow[\mathcal{R}]{*}$ defined by a term and proposition rewrite system $\mathcal{R}$, we can define natural deduction modulo as do Dowek and Werner (2003). Its inference rules are represented in Figure 1. They are the same as the one introduced by Gentzen (1934), except that we work modulo the rewrite relation. Leaves

---

[1] $\stackrel{!}{=}$ is used for definitions.

$$\Rightarrow\text{-i}\ \dfrac{\overset{[A]}{\underset{}{B}}}{C}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \Rightarrow B \qquad\qquad \Rightarrow\text{-e}\ \dfrac{A \qquad C}{B}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \Rightarrow B$$

$$\wedge\text{-i}\ \dfrac{A \qquad B}{C}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \wedge B \qquad\qquad \wedge\text{-e}\ \dfrac{C}{A}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \wedge B \ \text{or}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} B \wedge A$$

$$\vee\text{-i}\ \dfrac{A}{C}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \vee B \ \text{or}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} B \vee A \qquad\qquad \vee\text{-e}\ \dfrac{C \qquad \overset{[A]}{D} \qquad \overset{[B]}{D}}{D}\ \text{if}\ C \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} A \vee B$$

$$\forall\text{-i}\ \dfrac{\{y/x\}A}{B}\ \begin{array}{l}\text{if}\ B \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \forall x.\ A \ \text{and}\ y \ \text{is not free in}\ A \\ \text{nor in the assumptions of the proof} \\ \text{above}\end{array} \qquad\qquad \forall\text{-e}\ \dfrac{A}{B}\ \text{if}\ A \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \forall x.\ C \ \text{and}\ B \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \{t/x\}C$$

$$\exists\text{-i}\ \dfrac{B}{A}\ \text{if}\ A \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \exists x.\ C \ \text{and}\ B \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \{t/x\}C \qquad\qquad \exists\text{-e}\ \dfrac{B \qquad \overset{[\{y/x\}A]}{C}}{C}\ \begin{array}{l}\text{if}\ B \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \exists x.\ A \ \text{and}\ y \ \text{is not free in}\ C \\ \text{nor in the assumption of the proof} \\ \text{above except}\ \{y/x\}A\end{array}$$

$$\top\text{-i}\ \dfrac{}{A}\ \text{if}\ A \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \top \qquad\qquad \bot\text{-e}\ \dfrac{A}{B}\ \text{if}\ A \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} \bot \qquad\qquad classical\ \dfrac{}{A}\ \text{if}\ A \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} B \vee (B \Rightarrow \bot)$$

Figure 1: Inference Rules of Natural Deduction Modulo $\mathcal{R}$.

of a proof that are not discarded by some inference rules (on the contrary to $A$ in $\Rightarrow$-i for instance) are the assumptions of the proof.

The length of a proof is the number of inferences used in it. We will denote by $\mathcal{T} \vdash^{\mathsf{N}}_{k\ \mathcal{R}} P$ the fact that there exists a proof of $P$ of length at most $k$ using a finite subset of $\mathcal{T}$ ($\mathcal{T}$ can be infinite) as assumptions. In the case where $\mathcal{R} = \emptyset$, we are back to pure natural deduction, and we will use $\mathcal{T} \vdash^{\mathsf{N}}_{k} P$.

**Definition 2.1** (Compatible presentation (Dowek et al., 2003, Definition 1.4))**.** An axiomatic presentation $\Gamma$ of a theory is said *compatible* with a rewrite system $\mathcal{R}$ if:

- $P \underset{\mathcal{R}}{\overset{*}{\leftrightarrow}} Q$ implies $\Gamma \vdash^{\mathsf{N}} P \Leftrightarrow Q$;
- for every proposition $P \in \Gamma$, we have $\vdash^{\mathsf{N}}_{\mathcal{R}} P$.

For instance, $B \Rightarrow A$ is compatible with $A \to A \vee B$: it possible to prove $A \Leftrightarrow A \vee B$ assuming $B \Rightarrow A$ with the proof:

$$\wedge\text{-i}\ \dfrac{\Rightarrow\text{-i}\ \dfrac{\vee\text{-i}\ \dfrac{A \ (\text{i})}{A \vee B}}{A \Rightarrow A \vee B}\ (\text{i}) \qquad \vee\text{-e}\ \dfrac{A \vee B \ (\text{ii}) \qquad \Rightarrow\text{-i}\ \dfrac{\Rightarrow\text{-e}\ \dfrac{B \ (\text{iii}) \qquad B \Rightarrow A}{A}}{A \vee B \Rightarrow A}\ (\text{ii})}{A \Leftrightarrow A \vee B}}{}$$

(other cases of equivalent propositions can be derived from it), and reciprocally, $B \Rightarrow A$ has the following proof modulo $A \to A \vee B$:

$$\Rightarrow\text{-i}\ \dfrac{\vee\text{-i}\ \dfrac{B \ (\text{i})}{A}\ A \longrightarrow A \vee B}{B \Rightarrow A}\ (\text{i})$$

Given a rewrite system, a compatible presentation always exists, and one can show that proving modulo a rewrite system is the same as proving without modulo but using a compatible presentation as assumptions (Dowek et al., 2003, Proposition 1.8).

Proof lengths in finite compatible presentations are essentially the same:

**Proposition 2.2.** *Let $\Gamma_1$ and $\Gamma_2$ be two finite presentations compatible with the same rewrite system $\mathcal{R}$. It is possible to translate a proof of length $n$ in $\Gamma_1$ into a proof of length $O(n)$ in $\Gamma_2$.*

$$\Gamma_1 \vdash^{\mathsf{N}}_{k} P \;\rightsquigarrow\; \Gamma_2 \vdash^{\mathsf{N}}_{O(k)} P$$

*Proof.* We show that every axiom of $\Gamma_1$ can be translated by proof of bounded depth in $\Gamma_2$. By definition of compatibility, for all $P \in \Gamma_1$, we have $\vdash^{\mathsf{N}}_{\mathcal{R}} P$. Then, whenever the congruence is used in that proof, we replace it by a cut with the corresponding proof in $\Gamma_2$ thanks to compatibility. For instance, if we have

$$\forall\text{-e}\ \frac{\overset{\varpi}{A}}{B}\ \text{with}\ A \xleftrightarrow{*}_{\mathcal{R}} \forall x.\ C\ \text{and}\ B \xleftrightarrow{*}_{\mathcal{R}} \{t/x\}C$$

we know by compatibility that there exists proofs $\pi_1$ of $\Gamma_2 \vdash^{\mathsf{N}} A \Leftrightarrow \forall x.\ C$ and $\pi_2$ of $\Gamma_2 \vdash^{\mathsf{N}} B \Leftrightarrow \{t/x\}C$, so that we have

$$\Rightarrow\text{-e}\ \frac{\wedge\text{-e}\ \dfrac{\overset{\pi_2}{B \Leftrightarrow \{t/x\}C}}{\{t/x\}C \Rightarrow B} \qquad \forall\text{-e}\ \dfrac{\Rightarrow\text{-e}\ \dfrac{\wedge\text{-e}\ \dfrac{\overset{\pi_1}{A \Leftrightarrow \forall x.\ C}}{A \Rightarrow \forall x.\ C} \quad \overset{\varpi}{A}}{\forall x.\ C}}{\{t/x\}C}}{B}$$

Transforming all applications of the congruence in that way, we obtain a proof of $\Gamma_2 \vdash^{\mathsf{N}} P$. As $\Gamma_1$ is finite, there is a maximum $K$ on the length of such proofs, and a proof of length $n$ in $\Gamma_1$ can be transformed into a proof of length at most $K \times n$ in $\Gamma_2$ by replacing an axiom $P$ by its corresponding proof in $\Gamma_2$.  $\square$

2.3. **A Simple Proof-Length Speed-up.** Because part of the proofs are put into the congruence, it is quite easy to get arbitrary proof-length speed-ups in deduction modulo, even for very simple rewrite systems.

Consider the proposition rewrite system

$$\mathcal{A}dd \overset{!}{=} \left\{ \begin{array}{l} Add(O, y, y) \to \top \\ Add(s(x), y, s(z)) \to Add(x, y, z) \end{array} \right. .$$

It is easy to prove that the derivational complexity of $\mathcal{A}dd$ is polynomially bounded. Furthermore, it is confluent, and $\xleftrightarrow{*}_{\mathcal{A}dd}$ is clearly decidable in polynomial time. However, proving modulo $\mathcal{A}dd$ leads to an arbitrary proof-length speed-up compared to proving using a finite compatible presentation.

**Theorem 2.3.** *There is an infinite family $\mathcal{F}$ of propositions such that for all finite presentation $\Gamma$ compatible with $\mathcal{A}dd$,*

(1) *for all $P \in \mathcal{F}$, $\Gamma \vdash^{\mathsf{N}} P$*

(2) *for all $P \in \mathcal{F}$, $\vdash^{\mathsf{N}}_{1\ step} \mathcal{A}dd\ P$*

(3) *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $\Gamma \vdash^{\mathsf{N}}_{k\ steps} P$.*

*Proof.* Let $\underline{n}$ denote $\underbrace{s(\cdots s(O)\cdots)}_{n\ \text{times}}$ for $n \in \mathbb{N}$. Consider the family of propositions $(Add(\underline{i},\underline{i},\underline{2i}))_{i \in \mathbb{N}}$.

Clearly, (1) holds. Since $Add((\underline{i},\underline{i},\underline{2i}) \xrightarrow[\mathcal{A}dd]{*} Add(O,\underline{i},\underline{i}) \xrightarrow[\mathcal{A}dd]{} \top$, we have the following proof modulo $\mathcal{A}dd$:

$$\top\text{-i} \ \frac{}{Add(\underline{i},\underline{i},\underline{2i})} \ Add((\underline{i},\underline{i},\underline{2i}) \xleftrightarrow[\mathcal{A}dd]{*} \top$$

Hence, (2) holds. Consider the presentation containing the two axioms $\forall x.\ Add(x,O,x)$ and $\forall x\ y\ z.\ Add(s(x),y,s(z)) \Leftrightarrow Add(x,y,z)$. It is easy to prove that this finite presentation is compatible with $\mathcal{A}dd$. To prove $Add(\underline{i},\underline{i},\underline{2i})$ in this presentation, we need to use the second axiom at least $i$ times, so that the length of the proofs cannot be bounded by a constant. Now consider another finite presentation compatible with $\mathcal{A}dd$, Proposition 2.2 tells us that the length of the proofs cannot be bounded by a constant in that presentation neither. $\square$

**Note 1.** This is not true for infinite compatible presentations, since such presentations can contain $\mathcal{F}$.

## 3. Proof systems for $i^{\text{TH}}$-order arithmetic

### 3.1. Schematic systems.
We recall here, using Buss' terminology 1994, what a schematic system consists of. It is essentially an Hilbert-type (or Frege) proof system, i.e. valid propositions are derived from a finite number of axiom schemata using a finite number of inference rules. Theorem 1.1 is true on condition that proofs are performed using a schematic system.

Given a many-sorted signature of first-order logic, we can consider infinite sets of *metavariables* $\alpha^i$ for each sort $i$ (which will be substituted by variables), of *term variables* $\tau^i$ for each sort $i$ (which will be substituted by terms) and *proposition variables* $A(x_1, \ldots, x_n)$ for each arity $[i_1, \ldots, i_n]$ (which will be substituted by propositions).

Metaterms are built like terms, except that they can contain metavariables and term variables. Metapropositions are built like propositions, except that they can contain proposition variables (which play the same role as predicates) and metaterms.

A *schematic system* is a finite set of inference rules, where an inference rule is a triple of a finite set of metapropositions (the *premises*), a metapropositions (the *conclusion*), and a set of side conditions of the forms $\alpha^j$ *is not free in* $\Phi$ or $s$ *is freely substitutable for* $\alpha^j$ *in* $\Phi$ where $\Phi$ is a metaproposition and $s$ a metaterm of sort $j$. It is denoted by

$$\frac{\Phi_1 \quad \cdots \quad \Phi_n}{\Psi} \ (R)$$

An inference with an empty set of premises will be called an *axiom schema*. An axiom schema without metaproposition is an *axiom*.

### 3.2. $i^{\text{th}}$-order arithmetic.

$i^{\text{th}}$-order arithmetic $(Z_{i-1})$ is a many-sorted theory with sorts $0, \ldots, i-1$ and the signature

$$
\begin{array}{lll}
0 : 0 & + : [0;0] \to 0 & = \; : [0;0] \\
s : [0] \to 0 & \times : [0;0] \to 0 & \in^j \; : [j;j+1]
\end{array}
\quad .
$$

The schematic system we use here consists of the following inference rules:

**$15 + 2 \times \mathbf{i}$ axiom schemata of classical logic.** We take those used by Gentzen (1934, Chapter 5) to prove the equivalence of his formalisms with an Hilbert-type proof system:

$$A \Rightarrow A \tag{I}$$

$$A \Rightarrow B \Rightarrow A \tag{K}$$

$$(A \Rightarrow A \Rightarrow B) \Rightarrow A \Rightarrow B \tag{W}$$

$$(A \Rightarrow B \Rightarrow C) \Rightarrow B \Rightarrow A \Rightarrow C \tag{C}$$

$$(A \Rightarrow B) \Rightarrow (B \Rightarrow C) \Rightarrow A \Rightarrow C \tag{B}$$

$$(A \wedge B) \Rightarrow A \tag{$\text{Proj}_l$}$$

$$(A \wedge B) \Rightarrow B \tag{$\text{Proj}_r$}$$

$$(A \Rightarrow B) \Rightarrow (A \Rightarrow C) \Rightarrow A \Rightarrow (B \wedge C) \tag{Pair}$$

$$A \Rightarrow (A \vee B) \tag{$\text{Inj}_l$}$$

$$B \Rightarrow (A \vee B) \tag{$\text{Inj}_r$}$$

$$(A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow (A \vee B) \Rightarrow C \tag{Case}$$

$$(A \Rightarrow B) \Rightarrow (A \Rightarrow B \Rightarrow \bot) \Rightarrow A \Rightarrow \bot \tag{Contradiction}$$

$$(A \Rightarrow \bot) \Rightarrow A \Rightarrow B \tag{EFSQ}$$

$$\top \tag{T}$$

$$(\forall \alpha^j.\ A(\alpha^j)) \Rightarrow A(\tau^j) \tag{UI}$$

$$\left( \tau^j \text{ is freely substitutable for } \alpha^j \text{ in } A(\alpha^j) \right)$$

$$A(\tau^j) \Rightarrow \exists \alpha^j.\ A(\alpha^j) \tag{EI}$$

$$\left( \tau^j \text{ is freely substitutable for } \alpha^j \text{ in } A(\alpha^j) \right)$$

$$A \vee (A \Rightarrow \bot) \tag{TND}$$

**$1 + 2 \times \mathbf{i}$ inference rules of classical logic.** Again, we take thouse used by Gentzen (1934):

$$\frac{A \qquad A \Rightarrow B}{B} \tag{MP}$$

$$\frac{A \Rightarrow B(\beta^j)}{A \Rightarrow \forall \alpha^j.\ B(\alpha^j)} \; (\beta^j \text{ is not free in } A \Rightarrow \forall \alpha^j.\ B(\alpha^j)) \tag{Gen}$$

$$\frac{B(\beta^j) \Rightarrow A}{(\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A} \; (\beta^j \text{ is not free in } (\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A) \tag{Part}$$

**2 identity axiom schemata.** They define the particular relation $=$:

$$\forall \alpha^0.\ \alpha^0 = \alpha^0 \tag{Refl}$$

$$\forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0) \tag{Leibniz}$$

**7 Robinson's axioms.** They are the axioms defining the function symbols of arithmetic (Mostowski, Robinson, and Tarski, 1953):

$$\forall \alpha^0. \; \neg \; 0 = s(\alpha^0) \tag{$0 \neq s$}$$

$$\forall \alpha^0 \beta^0. \; s(\alpha^0) = s(\beta^0) \Rightarrow \alpha^0 = \beta^0 \tag{$\mathrm{Inj}_s$}$$

$$\forall \alpha^0. \; (\neg \; \alpha^0 = 0) \Rightarrow \exists \beta^0. \; \alpha^0 = s(\beta^0) \tag{$\mathrm{Onto}_s$}$$

$$\forall \alpha^0. \; \alpha^0 + 0 = \alpha^0 \tag{$+0$}$$

$$\forall \alpha^0 \beta^0. \; \alpha^0 + s(\beta^0) = s(\alpha^0 + \beta^0) \tag{$+s$}$$

$$\forall \alpha^0. \; \alpha^0 \times 0 = 0 \tag{$\times 0$}$$

$$\forall \alpha^0 \beta^0. \; \alpha^0 \times s(\beta^0) = \alpha^0 \times \beta^0 + \alpha^0 \tag{$\times s$}$$

**$i+1$ induction and comprehension axiom schemata.**

$$A(0) \Rightarrow \big(\forall \beta^0. \; A(\beta^0) \Rightarrow A(s(\beta^0))\big) \Rightarrow \forall \alpha^0. \; A(\alpha^0) \tag{Ind}$$

For all $0 \leq j < i - 1$,

$$\exists \alpha^{j+1}. \; \forall \beta^j. \; \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j) \qquad (\alpha^{j+1} \text{ is not free in } A(\beta^j)) \tag{$\mathrm{Comp}^j$}$$

From this point on, we will denote by $Z_{i-1} \models^{\mathsf{S}}_k P$ the fact that there exists a proof of $P$ of length at most $k$ in this schematic system, i.e. $P$ can be derived using at most $k$ instances of these inference rules. Abusing notations, we will write $Z_{i-1} \models^{\mathsf{N}}_{k \; \mathcal{R}} P$ to say that there is a proof of $P$ in natural deduction of length at most $k$ using as assumptions a finite subset of instances of the axiom schemata (Refl) to $(\mathrm{Comp}^j)$.

3.3. **Translations between schematic systems and natural deduction.** Buss' theorem is true in schematic systems, but deduction modulo is defined for natural deduction. It is important to get bounded translations between these formalisms to show that the speed-ups we will be considering are not artifacts of the deductive system.

3.3.1. *From $Z_i \models^{\mathsf{S}}$ to $Z_i \models^{\mathsf{N}}$.* We want to translate a proof in the schematic system of $Z_i$ into a proof in pure natural deduction using as assumptions instances of the axiom schemata (Refl) to $(\mathrm{Comp}^j)$.

For the axiom schemata and inference rules of classical logic, we use the same translation as Gentzen, for instance the axiom schema (C) is translated into the natural deduction proof

$$
\Rightarrow\text{-i} \cfrac{
\Rightarrow\text{-i} \cfrac{
\Rightarrow\text{-e} \cfrac{B \;\text{(ii)} \qquad \Rightarrow\text{-i} \cfrac{C}{A \Rightarrow C}\text{(iii)}}{
\Rightarrow\text{-i} \cfrac{\Rightarrow\text{-e} \cfrac{A \;\text{(iii)} \qquad A \Rightarrow B \Rightarrow C \;\text{(i)}}{B \Rightarrow C}}{}}
}{B \Rightarrow A \Rightarrow C}\text{(ii)}
}{(A \Rightarrow B \Rightarrow C) \Rightarrow B \Rightarrow A \Rightarrow C}\text{(i)}
$$

and the inference rule (Part) into

$$
\Rightarrow\text{-i} \cfrac{
\exists\text{-e} \cfrac{\exists \alpha^j. \; B(\alpha^j) \;\text{(i)} \qquad \Rightarrow\text{-e} \cfrac{B(\beta^j) \;\text{(ii)} \qquad B(\beta^j) \Rightarrow A}{A}\text{(ii)}}{A}
}{\exists \alpha^j. \; B(\alpha^j) \Rightarrow A}\text{(i)}
$$

(note that the side condition ensure that it is possible to consider that what will be substituted for $\beta$ is free in $A$ and the assumptions of the proof above $B(\beta^j) \Rightarrow A$). All these inference rules have a translation whose length does not depend on the propositions finally substituted in the proof.

In a schematic system proof, there is also a finite number of instances of the axiom schemata for identity, Robinson's axioms and induction and comprehension schemata. We keep these instances as assumptions in natural deduction, so that we obtain a proof in natural deduction using as assumptions a finite subset of instances of the axiom schemata (Refl) to (Comp$^j$), and whose length is linear compared to the schematic system proof:

**Proposition 3.1.** *It is possible to translate a proof of length $n$ in the schematic system for $Z_i$ into a proof of length $O(n)$ in (pure) natural deduction using assumptions in $Z_i$.*

$$Z_i \vdash^{\mathsf{S}}_{k} P \ \rightsquigarrow \ Z_i \vdash^{\mathsf{N}}_{O(k)} P$$

3.3.2. *From $Z_i \vdash^{\mathsf{N}}$ to $Z_i \vdash^{\mathsf{S}}$.* In this section, we consider a proof of $P$ in natural deduction, using as assumption finite instances of (Refl) to (Comp$^j$) in the language of $Z_i$. We translate it into a proof in the schematic system for $Z_i$.

This is essentially a generalization of the translation from the $\lambda$-calculus to combinatory logic (see Curry, Feys, and Craig, 1958). We define mutually recursively two functions by induction on the inference rules: $\mathrm{T}$ transforms a proof of $P$ in natural deduction using assumptions $\Gamma$ into a proof of $P$ in the schematic system (I) to (Part) plus $\Gamma$. $\mathrm{T}_A$ transform a proof of $P$ in natural deduction using assumptions $\Gamma, A$ into a proof of $A \Rightarrow P$ in the schematic system (I) to (Part) plus $\Gamma$. The translation can be found in the appendix.

It can be verified that this transformation is at most exponential in the length of proofs. Due to Cook and Reckhow (1979, Corollary 3.4), we could have found, at least for the propositional part, a polynomial translation. Nevertheless all we need in this paper is the fact that the increase of the proof length in the translation is bounded.

**Proposition 3.2.** *There exists some constant $K$ such that it is possible to translate a proof of length $n$ in the (pure) natural deduction using assumptions in $Z_i$ into a proof of length $O(K^n)$ in the schematic system for $Z_i$.*

$$Z_i \vdash^{\mathsf{N}}_{k} P \ \rightsquigarrow \ Z_i \vdash^{\mathsf{S}}_{O(K^k)} P$$

*Proof.* Let $K$ be the maximum number of steps that appear in addition of the recursive calls in the definition of $\mathrm{T}_A$ (note that it does not depend on $A$). First, if a proof $\varpi$ does not contain $\Rightarrow$-i, $\vee$-e or $\exists$-e, then $|T_A(\varpi)| \leq K|\varpi|$. We prove that by induction on $\varpi$. Let us detail $\Rightarrow$-e only, using notations of the appendix, the other cases being similar:

$$|\mathrm{T}_A(\varpi)| = |\mathrm{T}_A(\pi_1)| + |\mathrm{T}_A(\pi_2)| + 7$$
$$\leq K|\pi_1| + K|\pi_2| + K \qquad \text{by induction hypothesis, and by definition of } K$$
$$\leq K(|\pi_1| + |\pi_2| + 1)$$
$$\leq K|\varpi|$$

Now let us show that in all cases $|T_A(\varpi)| \leq K^{|\varpi|}$. This is also proved by induction on $\varpi$. We only detail the case of $\Rightarrow$-i. $|\mathrm{T}_A(\varpi)| = |\mathrm{T}_A(\mathrm{T}_B(\pi))|$. By induction hypothesis, $|\mathrm{T}_B(\pi)| \leq K^{|\pi|}$. Furthermore, $\mathrm{T}_B(\pi)$ does contain neither $\Rightarrow$-i, $\vee$-e nor $\exists$-e, so that

$|\mathrm{T}_A\left(\mathrm{T}_B\left(\pi\right)\right)| \le K|\mathrm{T}_B\left(\pi\right)| \le K \times K^{|\pi|} = K^{|\pi|+1} = K^{|\varpi|}$. From this result, we can deduce the bound for T. $\qquad\square$

## 4. Higher-order arithmetic as a first-order theory modulo

4.1. **Encoding higher orders using classes.** First, we translate a proof in the schematic system for $Z_i$ into a proof in natural deduction modulo with as assumptions a finite presentation using only first-order function symbols, resorting on the congruence to get the higher orders again.

To do so, we first consider the theory consisting in the *axioms* in (Refl) to ($\times s$), so without the axiom schemata (Leibniz), (Ind) and (Comp$^j$). Those are replaced by three new axioms (Leibniz$_{ax}$), (Ind$_{ax}$) and (Comp$^j_{ax}$). To do so, we use the work of Kirchner (2007) which permits to express first-order theories using a finite number of axioms. The idea is to transform some metaproposition $A(t_1, \ldots, t_n)$ used in an axiom schema into a proposition of the form $\langle t_1, \ldots, t_n \rangle \in \gamma$ where $\gamma$ is some term representing what proposition is actually substituted for $A$. Note that is long known that using classes permits to have finite first-order axiomatization, but Kirchner's work shows how to handle the classes with a simple rewrite system of weak explicit substitutions.

Following Kirchner's method, we add new sorts $\ell$ for lists and $c$ for classes, as well as new function symbols and predicate

$$
\begin{array}{llll}
1^j : j & nil : \ell & \cup : [c;c] \to c & \emptyset : c \\
S^j : [j] \to j & ::^j : [j;\ell] \to \ell & \cap : [c;c] \to c & \mathcal{P}^j : [c] \to c \\
\cdot[\cdot]^j : [j;\ell] \to j & \doteq : [0;0] \to c & \supset : [c;c] \to c & \mathcal{C}^j : [c] \to c \\
& \dot{\in}^j : [j;j+1] \to c & & \epsilon : [\ell;c]
\end{array}
$$

$\langle \alpha_1, \ldots, \alpha_n \rangle$ will be syntactic sugar for $\alpha_1 ::^{j_1} \cdots :: \alpha_n ::^{j_n} nil$ for the appropriate $j_m$. Note that we only need one sort of class, and not one for each orders, as we could have done. That way, all substitutions are done in the same setting. We change the axiom schemata (Leibniz), (Ind) and (Comp$^j$) into the following *axioms*:

$$\forall \gamma^c.\ \forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow \langle \alpha^0 \rangle \in \gamma^c \Rightarrow \langle \beta^0 \rangle \in \gamma^c \qquad (\text{Leibniz}_{ax})$$

$$\forall \gamma^c. \langle 0 \rangle \in \gamma^c \Rightarrow \left( \forall \beta^0.\ \langle \beta^0 \rangle \in \gamma^c \Rightarrow \langle s(\beta^0) \rangle \in \gamma^c \right) \Rightarrow \forall \alpha^0.\ \langle \alpha^0 \rangle \in \gamma^c \qquad (\text{Ind}_{ax})$$

For all $0 \le j < i$,

$$\forall \gamma^c.\ \exists \alpha^{j+1}.\ \forall \beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow \langle \beta^j \rangle \in \gamma^c \qquad (\text{Comp}^j_{ax})$$

We also need weak-substitution axioms which permit to decode the classes (see Kirchner, 2007, Definition 4).

$$\forall \alpha^j.\ \alpha^j[nil]^j = \alpha^j \qquad (\text{WS}_{nil})$$

$$\forall \alpha^j.\ \forall l^\ell.\ 1^j[\alpha^j ::^j l^\ell]^j = \alpha^j \qquad (\text{WS}_{1^j})$$

$$\forall \alpha^j.\ \forall \beta^k.\ \forall l^\ell.\ S^j(\alpha^j)[\beta^k ::^k l^\ell]^j = \alpha^j[l^\ell]^j \qquad (\text{WS}_{S^j})$$

$$\forall \alpha^0.\ \forall l^\ell.\ s(\alpha^0)[l^\ell]^0 = s(\alpha^0[l^\ell]^0) \qquad (\text{WS}_s)$$

$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ (\alpha^0 + \beta^0)[l^\ell]^0 = \alpha^0[l^\ell]^0 + \beta^0[l^\ell]^0 \qquad (\text{WS}_+)$$

$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ (\alpha^0 \times \beta^0)[l^\ell]^0 = \alpha^0[l^\ell]^0 \times \beta^0[l^\ell]^0 \qquad (\text{WS}_\times)$$

$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ l^\ell \in \doteq (\alpha^0, \beta^0) \Leftrightarrow \alpha^0[l^\ell]^0 = \beta^0[l^\ell]^0 \qquad (\text{WS}_=)$$

$$t[nil]^j \to t \qquad\qquad l \ \epsilon \ \dot{\in}^{j'}(t_1, t_2) \to t_1[l]^{j'} \in^{j'} t_2[l]^{j'+1}$$
$$1^j[t ::^j l]^j \to t \qquad\qquad l \ \epsilon \ A \cup B \to l \ \epsilon \ A \vee l \ \epsilon \ B$$
$$S^j(n)[t ::^k l]^j \to n[l]^j \qquad\qquad l \ \epsilon \ A \cap B \to l \ \epsilon \ A \wedge l \ \epsilon \ B$$
$$s(n)[l]^0 \to s(n[l]^0) \qquad\qquad l \ \epsilon \ A \supset B \to l \ \epsilon \ A \Rightarrow l \ \epsilon \ B$$
$$(t_1 + t_2)[l]^0 \to t_1[l]^0 + t_2[l]^0 \qquad\qquad l \ \epsilon \ \emptyset \to \bot$$
$$(t_1 \times t_2)[l]^0 \to t_1[l]^0 \times t_2[l]^0 \qquad\qquad l \ \epsilon \ \mathcal{P}^j(A) \to \exists x.\ x ::^j l \ \epsilon \ A$$
$$l \ \epsilon \ \dot{=} (t_1, t_2) \to t_1[l]^0 = t_2[l]^0 \qquad\qquad l \ \epsilon \ \mathcal{C}^j(A) \to \forall x.\ x ::^j l \ \epsilon \ A$$
$$x \in^{j'} comp^{j'+1}(A) \to x ::^{j'} nil \ \epsilon \ A$$

for all $0 \leq j \leq i$, $0 \leq k \leq i$ and $0 \leq j' < i$.

Figure 2: Rewrite rules of $\mathcal{HO}_i$

$$\forall \alpha^j.\ \forall \beta^{j+1}.\ \forall l^\ell.\ l^\ell \ \epsilon \ \dot{\in}^j(\alpha^j, \beta^{j+1}) \Leftrightarrow \alpha^j[l^\ell]^j \in^j \beta^{j+1}[l^\ell]^{j+1} \qquad\qquad (\text{WS}_{\in^j})$$

$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ l^\ell \ \epsilon \ \alpha^c \cup \beta^c \Leftrightarrow l^\ell \ \epsilon \ \alpha^c \vee l^\ell \ \epsilon \ \beta^c \qquad\qquad (\text{WS}_\vee)$$

$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ l^\ell \ \epsilon \ \alpha^c \cap \beta^c \Leftrightarrow l^\ell \ \epsilon \ \alpha^c \wedge l^\ell \ \epsilon \ \beta^c \qquad\qquad (\text{WS}_\wedge)$$

$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ l^\ell \ \epsilon \ \alpha^c \supset \beta^c \Leftrightarrow l^\ell \ \epsilon \ \alpha^c \Rightarrow l^\ell \ \epsilon \ \beta^c \qquad\qquad (\text{WS}_\Rightarrow)$$

$$\forall l^\ell.\ l^\ell \ \epsilon \ \emptyset \Leftrightarrow \bot \qquad\qquad (\text{WS}_\bot)$$

$$\forall \alpha^c.\ \forall l^\ell.\ (l^\ell \ \epsilon \ \mathcal{P}^j(\alpha^c) \Leftrightarrow \exists \beta^j.\ \beta^j ::^j l^\ell \ \epsilon \ \alpha^c) \qquad\qquad (\text{WS}_{\exists^j})$$

$$\forall \alpha^c.\ \forall l^\ell.\ (l^\ell \ \epsilon \ \mathcal{C}^j(\alpha^c) \Leftrightarrow \forall \beta^j.\ \beta^j ::^j l^\ell \ \epsilon \ \alpha^c) \qquad\qquad (\text{WS}_{\forall^j})$$

We call $Z_i^{\text{ws}}$ the theory consisting of all the axioms from (Refl) to $(\text{WS}_{\forall^j})$, but not the axioms schemata (Leibniz), (Ind) and $(\text{Comp}^j)$.

**Proposition 4.1.** *The theory $Z_i^{\text{ws}}$ is a conservative extension of $Z_i$.*

*Proof.* This is the Proposition 4 of Kirchner (2007). $\qquad\square$

Now, we use skolemization to transform $(\text{Comp}_{ax}^j)$ (see van Dalen, 1989, Section 3.4). We add new function symbols $comp^j : [c] \to j$ for all $0 < j \leq i$. We then consider the skolemized version of $(\text{Comp}_{ax}^j)$:

$$\forall \gamma^c.\ \forall \beta^j.\ \beta^j \in^j comp^{j+1}(\gamma^c) \Leftrightarrow \langle \beta^j \rangle \ \epsilon \ \gamma^c \qquad\qquad (\text{Comp}_{sk}^j)$$

We denote by $Z_i^{\text{sk}}$ the theory $Z_i^{\text{ws}}$ where $(\text{Comp}_{ax}^j)$ is replaced by $(\text{Comp}_{sk}^j)$.

**Proposition 4.2.** *The theory $Z_i^{\text{sk}}$ is a conservative extension of $Z_i^{\text{ws}}$.*

*Proof.* According to van Dalen (1989, Corollary 3.4.5), $Z_i^{\text{sk}} \cup \{(\text{Comp}_{ax}^j)\}$ is a conservative extension of $Z_i^{\text{ws}}$. But $(\text{Comp}_{ax}^j)$ can be proved in $Z_i^{\text{sk}}$ so that we can drop it. $\qquad\square$

We can then transform each axiom where a higher-order function symbol or predicate appears, as well as axioms decoding classes, into rewrite rules, and work modulo the resulting rewrite system. We denote by $\mathcal{HO}_i$ the rewrite system defined in Figure 2. This rewrite system has the following properties:

- It is finite (for a given $i$).
- It is terminating in a polynomial number of steps (Proposition 4.3).

- It is confluent: it terminates and it is locally confluent, since the only critical pairs, of the form $f(t_1, \ldots, t_n) \overset{}{\underset{\mathcal{HO}_i}{\longleftarrow}} f(t_1, \ldots, t_n)[nil] \overset{}{\underset{\mathcal{HO}_i}{\longrightarrow}} f(t_1[nil], \ldots, t_n[nil])$ for $f \in \{+; \times; s\}$, are easily joinable.
- It is left-linear, i.e. variables appears only once on the left-hand side of each rule.

Before showing that $\mathcal{HO}_i$ has a polynomially bounded derivational complexity, let us first see how $\mathcal{HO}_i$ works and, in particular, how it can be used to encode propositions as terms. Proposition 2 of Kirchner (2007) says that, for all propositions $P$ of the language of $i^{\text{th}}$-order arithmetic, and for all finite lists of variables $\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}$, it is possible to prove

$$\exists E^c. \; \forall \alpha_1^{j_1} \cdots \alpha_n^{j_n}. \; \langle \alpha_1^{j_1}, \ldots, \alpha_n^{j_n} \rangle \in E^c \Leftrightarrow P \; .$$

Moreover, the proof of this proposition shows us how to construct the witness for $E^c$. We will denote it by $E_P^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$, and it is therefore defined as:

$$||\alpha^j||^{\emptyset} \overset{!}{=} \alpha^j$$

$$||\alpha_1^{j_i}||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} 1^{j_1}$$

$$||\alpha^j||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} S^j(||\alpha^j||^{\alpha_2^{j_2}, \ldots, \alpha_n^{j_n}}) \qquad \text{if } \alpha^j \neq \alpha_1^{j_1}$$

$$||0||^{\emptyset} \overset{!}{=} 0$$

$$||0||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} S^0(||0||^{\alpha_2^{j_2}, \ldots, \alpha_n^{j_n}})$$

$$||s(t)||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} s(||t||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}})$$

$$||t_1 + t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} ||t_1||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} + ||t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$$

$$||t_1 \times t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} ||t_1||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \times ||t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$$

$$E_{t_1 = t_2}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} \; \dot{=} \, (||t_1||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}, ||t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}})$$

$$E_{t_1 \in^j t_2}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} \; \dot{\in}^j(||t_1||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}, ||t_2||^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}})$$

$$E_{\perp}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} \emptyset$$

$$E_{P \Rightarrow Q}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} E_P^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \supset E_q^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$$

$$E_{P \wedge Q}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} E_P^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \cap E_q^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$$

$$E_{P \vee Q}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} E_P^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \cup E_q^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}$$

$$E_{\forall \alpha^j. \, P}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} \mathcal{C}^j(E_P^{\alpha^j, \alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}) \qquad \text{if } \alpha^j \notin \{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}\}$$

$$E_{\exists \alpha^j. \, P}^{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}} \overset{!}{=} \mathcal{P}^j(E_P^{\alpha^j, \alpha_1^{j_1}, \ldots, \alpha_n^{j_n}}) \qquad \text{if } \alpha^j \notin \{\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}\}$$

Then, one can prove that $\langle t_1, \ldots, t_n \rangle \in E_P^{\alpha_1, \ldots, \alpha_n} \overset{*}{\longrightarrow} \{t_1/\alpha_1, \ldots, t_n/\alpha_n\}P$. For instance, consider the proposition $P \overset{!}{=} x = 0 \vee \exists y. \, x \in^0 y$. Then $E_P^x$ equals $\dot{=} (1^0, S^0(0)) \cup \mathcal{P}^1 \left( \dot{\in}^0(S^0(1^0), 1^1) \right)$

and $\langle t \rangle \in E_P^x$ can be rewritten to $t = 0 \vee \exists x^1.\ t \in^0 x^1$:

$$\langle t \rangle \in E_P^x \longrightarrow \langle t \rangle \in\ \dot{=}\ (1^0, S^0(0)) \vee \langle t \rangle \in \mathcal{P}^1 \left( \dot{\in}^0(S^0(1^0), 1^1) \right)$$

$$\xrightarrow{2} \left( 1^0[t ::^0 nil] = S^0(0)[t ::^0 nil] \right) \vee \exists x^1.\ \langle x^1, t \rangle \in\ \dot{\in}^0(S^0(1^0), 1^1)$$

$$\xrightarrow{3} (t = 0[nil]) \vee \exists x^1.\ S^0(1^0)[x^1 ::^1 t ::^0:: nil] \in^0 1^1[x^1 ::^1 t ::^0:: nil]$$

$$\xrightarrow{3} t = 0 \vee \exists x^1.\ 1^0[t ::^0:: nil] \in^0 x^1$$

$$\longrightarrow t = 0 \vee \exists x^1.\ t \in^0 x^1$$

**Proposition 4.3.** *The derivational complexity of $\mathcal{HO}_i$ is polynomially bounded.*

*Proof.* Let us note $\mathcal{WS}_i$ the system $\mathcal{HO}_i$ without the last rule. $\mathcal{WS}_i$ is computing the application of a substitution to (the encoding of) a proposition. It cannot be applied more than the size of the right-hand side of $\in$ and the left-hand side of $\cdot[\cdot]$ (by simple induction on the derivation). Therefore, the derivational complexity of $\mathcal{WS}_i$ is linear. Now, note that a substitution is blocked by all $comp^j$, i.e. $comp^j(t)[l]$ cannot be reduced if neither $t$ nor $l$ can. Therefore, the last rule of $\mathcal{HO}_i$ can only be applied to the outermost $comp^j$s: due to the sort constraints, $\in^j$ cannot appear inside a $comp^{j'}$, and if a $\dot{\in}^j$ is transformed into a $\in^j$ by the rule $l \in \dot{\in}^j(t_1, t_2) \to t_1[l]^j \in^j t_2[l]^{j+1}$, the substitution applied to $t_2$ blocks $comp^{j+1}$ if it is its function symbol. Applying $\mathcal{WS}_i$ can duplicate the initially outermost $comp^j$s, but not more than the total number of $1^{j'}$ in the initial term. Once the last rule of $\mathcal{HO}_i$ is applied to all these copies of the outermost $comp^j$s, only $\mathcal{WS}_i$ can be applied. Therefore, the derivational complexity of $\mathcal{HO}_i$ is polynomially bounded. $\square$

The axiom schemata (Leibniz), (Ind) and (Comp$^j$) can be replaced by the proofs in Figure 3. Note that the replacement for (Comp$^j$) does not need extra axioms, because all is done in the congruence.

Let $FZ$ be the theory consisting of (Refl), Robinson's axioms $(0 \neq s)$ to $(\times s)$, (Leibniz$_{ax}$) and (Ind$_{ax}$), consisting only of a finite number of axioms, all of them in the language of $Z_0$ plus the language of Kirchner's classes. A proof $\pi$ of $P$ in the schematic system for $Z_i$ can be translated into a proof of $P$ in natural deduction modulo $\mathcal{HO}_i$ using assumptions in $FZ$ whose length is linear compared to the length of $\pi$.

**Proposition 4.4.** *It is possible to translate a proof of length $n$ in the schematic system for $Z_i$ into a proof of length $O(n)$ in the natural deduction modulo $\mathcal{HO}_i$ using assumptions in $FZ$.*

$$Z_i \vdash^{\mathsf{S}}_k P \ \rightsquigarrow \ FZ \vdash^{\mathsf{N}}_{O(k)\ \mathcal{HO}_i} P$$

*Proof.* Instances of axiom schemata in the proof in $Z_i$ are replaced by the proofs in Figure 3, whose length is fixed. $\square$

This result can also be stated entirely in natural deduction

**Theorem 4.5.** *For all $i \geq 0$, there exists a finite confluent rewrite system with polynomially bounded derivational complexity $\mathcal{HO}_i$ such that for all propositions $P$, if $Z_i \vdash^{\mathsf{N}}_k P$ then $FZ \vdash^{\mathsf{N}}_{O(k)\ \mathcal{HO}_i} P$.*

*Proof.* We replace the instance of the axiom schemata (Leibniz), (Ind) and (Comp$^j$) by proofs using the axioms (Leibniz$_{ax}$) and (Ind$_{ax}$) as indicated in Figure 3. $\square$

$$\forall\text{-e} \ \frac{\forall\gamma^c. \ \forall\alpha^0\beta^0. \ \alpha^0 = \beta^0 \Rightarrow \langle\alpha^0\rangle \ \epsilon \ \gamma^c \Rightarrow \langle\beta^0\rangle \ \epsilon \ \gamma^c \ \text{(Leibniz}_{ax})}{\forall\alpha^0\beta^0. \ \alpha^0 = \beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0)}$$

$$(\text{because } \langle\alpha^0\rangle \ \epsilon \ E^x_{A(x)} \Rightarrow \langle\beta^0\rangle \ \epsilon \ E^x_{A(x)} \stackrel{*}{\longrightarrow} A(\alpha^0) \Rightarrow A(\beta^0))$$

$$\forall\text{-e} \ \frac{\forall\gamma^c.\langle 0\rangle \ \epsilon \ \gamma^c \Rightarrow \left(\forall\beta^0. \ \langle\beta^0\rangle \ \epsilon \ \gamma^c \Rightarrow \langle s(\beta^0)\rangle \ \epsilon \ \gamma^c\right) \Rightarrow \forall\alpha^0. \ \langle\alpha^0\rangle \ \epsilon \ \gamma^c \ \text{(Ind}_{ax})}{A(0) \Rightarrow \left(\forall\beta^0. \ A(\beta^0) \Rightarrow A(s(\beta^0))\right) \Rightarrow \forall\alpha^0. \ A(\alpha^0)}$$

$$(\text{because for all } t, \ \langle t\rangle \ \epsilon \ E^x_{A(x)} \stackrel{*}{\longrightarrow} A(t))$$

$$\Rightarrow\text{-i} \ \frac{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \ \text{(ii)}}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Rightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})} \ \text{(ii)}$$

$$\wedge\text{-i} \ \frac{\Rightarrow\text{-i} \ \dfrac{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \ \text{(i)}}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Rightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})} \ \text{(i)} \qquad \vdots}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Leftrightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}$$

$$\exists\text{-i} \ \frac{\forall\text{-i} \ \dfrac{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Leftrightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}{\forall\beta^j. \ \beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Leftrightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})} \quad \begin{array}{c}\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \\ \longrightarrow \langle\beta^j\rangle \ \epsilon \ E^x_{A(x)} \stackrel{*}{\longrightarrow} A(\beta^j)\end{array}}{\exists\alpha^{j+1}. \ \forall\beta^j. \ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j)}$$

Figure 3: Translations of the axiom schemata (Leibniz), (Ind) and (Comp$^j$).

4.2. **Higher-order arithmetic as purely computational theory.** We define here higher-order arithmetic entirely as a rewrite system modulo which inference will be applied. This is in line with the work of Dowek and Werner (2005) who express first-order arithmetic as a theory modulo. The idea is to combine this with the rewrite system of the previous section, to get a characterization of higher-order arithmetic. Notwithstanding, we will look carefully at the length of proofs in the translations.

Dowek and Werner (2005) use the following method to introduce the induction schema for first-order arithmetic: they add a new predicate $N$ of arity $[0]$ which essentially says that an element is a natural number, and thus can be used in the induction schema. $N(n)$ can therefore be rewritten to $\forall p. \ 0 \in p \Rightarrow (\forall y. \ N(y) \Rightarrow y \in p \Rightarrow s(y) \in p) \Rightarrow n \in p$. Then, function symbols $f_P^{x,y_1,\ldots,y_n}$ for each proposition $P$ of first-order arithmetic with free variables $x, y_1, \ldots, y_n$ are added, as well as rewrite rules $x \in f_P^{x,y_1,\ldots,y_n}(y_1, \ldots, y_n) \to P$. To prove a proposition using induction, we need to know that the variables used in the proof are natural numbers, hence quantifiers are relativized with the predicate $N$ (i.e. $\forall x. \ P$ becomes $\forall x. \ N(x) \Rightarrow P$, and $\exists x. \ P$ becomes $\exists x. \ N(x) \wedge P$). Using this, it is proved (Dowek and Werner, 2005, Proposition 13) that we obtain a conservative extension of first-order arithmetic. Nevertheless, the length of the proofs is not preserved by the relativization. Indeed, to translate a proof whose last step is

$$\forall\text{-e} \ \frac{\begin{array}{c}\pi \\ \forall x. \ P\end{array}}{\{t/x\}P}$$

we have to transform it into a proof

Arithmetic rules:

$$pred(0) \rightarrow 0 \qquad\qquad 0 \times y \rightarrow y$$
$$pred(s(x)) \rightarrow x \qquad\qquad s(x) \times y \rightarrow x \times y + y$$
$$0 + y \rightarrow y \qquad\qquad Null(0) \rightarrow \top$$
$$s(x) + y \rightarrow s(x+y) \qquad\qquad Null(s(x)) \rightarrow \bot$$

Axiom schemata:

$$x = y \rightarrow \forall z^c.\ \langle x \rangle \ \epsilon \ z \Rightarrow \langle y \rangle \ \epsilon \ z \qquad x \in^j comp^{j+1}(y) \rightarrow x ::^j nil \ \epsilon \ y$$

$$x ::^0 nil \ \epsilon \ p \rightarrow \langle x \rangle \ \epsilon \ p \vee (\langle 0 \rangle \ \epsilon \ p \wedge \forall y.\ \langle y \rangle \ \epsilon \ p \Rightarrow \langle s(y) \rangle \ \epsilon \ p)$$

Substitutions and classes: $\mathcal{WS}_i +$

$$pred(n)[l]^0 \rightarrow pred(n[l]^0) \qquad\qquad \ell \ \epsilon \ \dot{Null}(t) \rightarrow Null(t[\ell]^0)$$

Figure 4: Rewrite rules of $\mathcal{HHA}_i^{\mathrm{mod}}$

$$\Rightarrow\text{-e} \ \dfrac{\varpi \quad N(t) \qquad \forall\text{-e} \ \dfrac{\begin{matrix}|\pi|\\ \forall x.\ N(x) \Rightarrow P\end{matrix}}{N(t) \Rightarrow \{t/x\}P}}{\{t/x\}P}$$

The problem is that the length of the proof $\varpi$ depends on the size of $t$.

Hence, we use a different approach. Starting from $FZ$ modulo $\mathcal{HO}_i$, it remains to orient the axioms of $FZ$ into rewrite rules. Axioms $(+0)$ to $(\times s)$ can be easily oriented. To orient (Refl) and (Leibniz$_{ax}$), we use the axiom

$$\forall \alpha^0\ \beta^0.\ \alpha^0 = \beta^0 \Leftrightarrow (\forall \gamma^c.\ \langle \alpha^0 \rangle \ \epsilon \ \gamma^c \Rightarrow \langle \beta^0 \rangle \ \epsilon \ \gamma^c) \qquad\qquad (=_{def})$$

which is equivalent to their conjunction. (Onto$_s$) is redundant if the induction principle is present, so it can be dropped. To encode $(0 \neq s)$ and (Inj$_s$), we use the same technique as Dowek and Werner (2005): we introduce a new function symbol $pred : [0] \rightarrow 0$ and a new predicate $Null : [0]$, as well as new axioms defining them:

$$pred(0) = 0 \qquad\qquad\qquad\qquad (pred_0)$$

$$\forall \alpha^0.\ pred(s(\alpha^0)) = \alpha^0 \qquad\qquad\qquad (pred_s)$$

$$Null(0) \qquad\qquad\qquad\qquad (Null_0)$$

$$\forall \alpha^0.\ \neg Null(s(\alpha^0)) \qquad\qquad\qquad (Null_s)$$

which can be easily oriented. It remains to orient the induction principle (Ind$_{ax}$). The most problematic part is that this axiom is the universal closure of an implication, whereas proposition rewrite rules are compatible with universal closures of logical equivalences. We use the fact that $B \Rightarrow A$ is intuitionistically equivalent to $A \Leftrightarrow A \vee B$, so that (Ind$_{ax}$) is equivalent to

$$\forall \alpha^0\ \gamma^c.\ \langle \alpha^0 \rangle \ \epsilon \ \gamma^c \Leftrightarrow \left(\langle \alpha^0 \rangle \ \epsilon \ \gamma^c \vee \left(\langle 0 \rangle \ \epsilon \ \gamma^c \wedge (\forall \beta^0.\ \langle \beta^0 \rangle \ \epsilon \ \gamma^c \Rightarrow \langle s(\beta^0) \rangle \ \epsilon \ \gamma^c)\right)\right) \quad (\text{Ind}_{mod})$$

If we do not use (TND) as axiom (i.e. if we work in intuitionistic logic), we therefore obtain a formulation of higher-order Heyting arithmetic through the rewrite system $\mathcal{HHA}_i^{\mathrm{mod}}$ defined in Figure 4. With this rewrite system, we can linearly simulate higher-order arithmetic in deduction modulo:

**Theorem 4.6.** *For all $i$ there exists a finite rewrite system $\mathcal{HHA}_i^{mod}$ such that for all propositions $P$ in the language of $Z_i$, if $Z_i \vdash^{\mathsf{N}}_{k \ steps} P$ then $\vdash^{\mathsf{N}}_{O(k) \ steps} \mathcal{HHA}_i^{mod} P$.*

*Proof.* It is sufficient to prove that all instances of the axiom schemata of $Z_i$ can be proved in a bounded number of steps.

(Refl) can be proved by

$$\forall\text{-i} \ \cfrac{\forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\langle\alpha^0\rangle \in p^c \ \text{(i)}}{\langle\alpha^0\rangle \in p^c \Rightarrow \langle\alpha^0\rangle \in p^c} \ \text{(i)}}{\forall p^c.\ \langle\alpha^0\rangle \in p^c \Rightarrow \langle\alpha^0\rangle \in p^c}}{\forall\alpha^0.\ \alpha^0 = \alpha^0} \qquad \alpha^0 = \alpha^0 \longrightarrow \forall p^c.\ \langle\alpha^0\rangle \in p^c \Rightarrow \langle\alpha^0\rangle \in p^c$$

$(0 \neq s)$ is proved using $x = y \to \forall z^c.\ \langle x\rangle \in z \Rightarrow \langle y\rangle \in z$. $(0 \neq s)$

$$\forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\Rightarrow\text{-e} \ \cfrac{\forall\text{-e} \ \cfrac{0 = s(\alpha^0) \ \text{(i)}}{\langle 0\rangle \in N\dot{u}ll(1^0) \Rightarrow \langle s(\alpha^0)\rangle \in N\dot{u}ll(1^0)} \qquad \top\text{-i} \ \cfrac{}{N\dot{u}ll(0)}}{\bot}}{0 = s(\alpha^0) \Rightarrow \bot} \ \text{(i)}}{\forall\alpha^0.\ \neg\, 0 = s(\alpha^0)}$$

$$\langle 0\rangle \in N\dot{u}ll(1^0) \xrightarrow{*} N\dot{u}ll(0) \longrightarrow \top$$
$$\langle s(\alpha^0)\rangle \in N\dot{u}ll(1^0) \xrightarrow{*} N\dot{u}ll(s(\alpha^0)) \longrightarrow \bot$$

Let $E_{\text{Inj}_s}$ be $\dot{=}(S^0(\alpha^0), pred(1^0))$, $(\text{Inj}_s)$ is proved by

$$\forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\Rightarrow\text{-e} \ \cfrac{\forall\text{-e} \ \cfrac{s(\alpha^0) = s(\beta^0) \ \text{(i)}}{\langle s(\alpha^0)\rangle \in E_{\text{Inj}_s} \Rightarrow \langle s(\beta^0)\rangle \in E_{\text{Inj}_s}} \qquad \forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\langle\alpha^0\rangle \in p^c \ \text{(i)}}{\langle\alpha^0\rangle \in p^c \Rightarrow \langle\alpha^0\rangle \in p^c} \ \text{(i)}}{\alpha^0 = \alpha^0}}{\alpha^0 = \beta^0}}{s(\alpha^0) = s(\beta^0) \Rightarrow \alpha^0 = \beta^0} \ \text{(i)}}{\forall\alpha^0\beta^0.\ s(\alpha^0) = s(\beta^0) \Rightarrow \alpha^0 = \beta^0} \ 2\times$$

Let $E_{\text{Onto}_s} \overset{!}{=} (\dot{=}(1, S(0)) \supset \emptyset) \supset \mathcal{P}(\dot{=}(S(1), s(1)))$. $(\text{Onto}_s)$ is proved by

$$\forall\text{-i} \ \cfrac{\vee\text{-i} \ \cfrac{\wedge\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\bot\text{-e} \ \cfrac{\Rightarrow\text{-e} \ \cfrac{\forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\langle 0\rangle \in p^c \ \text{(ii)}}{\langle 0\rangle \in p^c \Rightarrow \langle 0\rangle \in p^c} \ \text{(ii)}}{0 = 0} \qquad 0 = 0 \Rightarrow \bot \ \text{(i)}}{\bot}}{\exists\beta^0.\ \alpha^0 = s(\beta^0)}}{\neg 0 = 0 \Rightarrow \exists\beta^0.\ \alpha^0 = s(\beta^0)} \ \text{(i)} \qquad \forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\exists\text{-i} \ \cfrac{\forall\text{-i} \ \cfrac{\Rightarrow\text{-i} \ \cfrac{\langle s(y)\rangle \in p^c \ \text{(iii)}}{\langle s(y)\rangle \in p^c \Rightarrow \langle s(y)\rangle \in p^c} \ \text{(iii)}}{s(y) = s(y)}}{\exists\beta^0.\ s(y) = s(\beta^0)}}{s(y) \in E_{\text{Onto}_s}}}{y \in E_{\text{Onto}_s} \Rightarrow s(y) \in E_{\text{Onto}_s}}}{\forall y.\ y \in E_{\text{Onto}_s} \Rightarrow s(y) \in E_{\text{Onto}_s}}}{\langle 0\rangle \in E_{\text{Onto}_s} \wedge \forall y.\ \langle y\rangle \in E_{\text{Onto}_s} \Rightarrow \langle s(y)\rangle \in E_{\text{Onto}_s}}}{\alpha^0 \in E_{\text{Onto}_s}}}{\forall\alpha^0.\ (\neg\, \alpha^0 = 0) \Rightarrow \exists\beta^0.\ \alpha^0 = s(\beta^0)}$$

$(+0)$ to $(\times s)$ are easy to prove using the arithmetical rules and the rule for $=$.

(Ind) has the following proof:

$$\wedge\text{-i} \cfrac{\cfrac{P(0)\ \text{(i)} \qquad \forall\beta^0.\ P(\beta^0) \Rightarrow P(s(\beta^0))\ \text{(ii)}}{\langle 0\rangle \in E_P^x \wedge \forall\beta^0.\ \langle\beta^0\rangle \in E_P^x \Rightarrow \langle s(\beta^0)\rangle \in E_P^x}}{}$$

$$\vee\text{-i} \cfrac{}{\cfrac{\langle\alpha^0\rangle \in E_P^x}{\forall\text{-i}\ \cfrac{}{\forall\alpha^0.\ P(\alpha^0)}}}$$

$$\Rightarrow\text{-i} \cfrac{}{P(0) \Rightarrow (\forall\beta^0.\ P(\beta^0) \Rightarrow P(s(\beta^0))) \Rightarrow \forall\alpha^0.\ P(\alpha^0)}\ \text{(i),(ii)}$$

(Comp$^j$) has the following proof:

$$\Rightarrow\text{-i} \cfrac{\beta^j \in^j comp^{j+1}(E_A^x)\ \text{(i)}}{\beta^j \in^j comp^{j+1}(E_A^x) \Rightarrow \langle\beta^j\rangle \in E_A^x}\ \text{(i)} \quad \Rightarrow\text{-i} \cfrac{\langle\beta^j\rangle \in E_A^x\ \text{(ii)}}{\beta^j \in E_A^x \Rightarrow \langle\beta^j\rangle \in^j comp^{j+1}(E_A^x)}\ \text{(ii)}$$

$$\wedge\text{-i} \cfrac{}{\cfrac{\beta^j \in^j comp^{j+1}(E_A^x) \Leftrightarrow \langle\beta^j\rangle \in E_A^x}{\forall\text{-i}\ \cfrac{\forall\beta^j.\ \beta^j \in^j comp^{j+1}(E_A^x) \Leftrightarrow A(\beta^j)}{\exists\text{-i}\ \cfrac{}{\exists\alpha^{j+1}.\ \forall\beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j)}}}}$$

$\square$

What we obtain is a conservative extension:

**Theorem 4.7.** *For all proposition $P$ in the language of $Z_i$, if $\vdash^{\mathsf{N}}_{\mathcal{HHA}_i^{mod}} P$ then $Z_i \vdash^{\mathsf{N}} P$.*

*Proof.* First, we can show, as do Dowek and Werner (2005), that adding $pred$, $Null$ and the axioms $(pred_0)$ to $(Null_s)$ gives a conservative extension. This can be done by Skolemizing the proposition $\forall\alpha^0.\ \exists\beta^0.\ (\alpha^0 = 0 \Rightarrow \beta^0 = 0) \wedge (\forall\gamma^0.\ \alpha^0 = s(\gamma^0) \Rightarrow \beta^0 = \gamma^0)$, which holds in first-order arithmetic, and by interpreting $Null(x)$ as $x = 0$.

Then, we apply the method of Kirchner (2007), which gives a conservative extension. Finally we skolemize the axioms corresponding to the comprehension schemata, and thus we obtain a conservative extension (see van Dalen, 1989). Finally, we just have to prove the equivalence of (Refl) and (Leibniz$_{ax}$) with ($=_{def}$), which is easy.

It can be remarked that the presentation obtained is compatible with $\mathcal{HHA}_i^{\text{mod}}$, hence the conclusion of the theorem. $\square$

Compared to $\mathcal{HO}_i$, the main issue is that the derivational complexity of $\mathcal{HHA}_i^{\text{mod}}$ is not polynomially bounded—actually, it does not even terminate. The non-termination is due to the rule encoding the induction principle, since it can be proved that the complexity of $\mathcal{HHA}_i^{\text{mod}}$ without this rule is polynomially bounded. It is not too surprising, since the real power of arithmetic lies in this principle. Note that Dowek and Werner (2005, Remark 1) propose a terminating rule to encode the induction principle, but, as said before, proof length is not kept. In fact, it can be proved that there is an arbitrary proof-length speed-up between the axiomatic presentation of first-order arithmetic and the presentation of Dowek and Werner: $\exists\alpha^0.\ \alpha^0 = \underline{n}$ can be proved in at most 7 steps in first-order arithmetic, whereas it need a proof-length linear in $n$ in the system of Dowek and Werner.

Poincaré (1902) advocates that everything in first-order arithmetic but the induction principle should be presented as computation, because the induction principle represents the only real deductive axiom of the theory. Following this idea, we want to keep all rewrite rules of $\mathcal{HHA}_i^{\text{mod}}$ but the rule for the induction principle, and present this latter rule in another way. Instead of using it as an axiom, we can apply the ideas within supernatural deduction (Wack, 2005) on it. Supernatural deduction consists in transforming proposition rewrite rules into new inference rules. It cannot be applied in our case, since $\vee$ cannot be handled by supernatural deduction. However, it instigates the new inference rule

$$[\langle \beta^0 \rangle \ \epsilon \ \sigma^c]$$

$$\text{Ind-i} \ \frac{\langle 0 \rangle \ \epsilon \ \sigma^c \qquad \langle s(\beta^0) \rangle \ \epsilon \ \sigma^c}{\langle \tau^0 \rangle \ \epsilon \ \sigma^c} \ \begin{array}{l} \beta^0 \text{ not free in } \langle \tau^0 \rangle \ \epsilon \ \sigma^c \\ \text{nor in the assumptions above} \end{array}$$

Proving with this new inference rules is equivalent to proving using the axiom $(\text{Ind}_{mod})$. We obtain a first-order proof system for higher-order arithmetic which is axiom-free, whose proofs can be checked in polynomial time, and whose proof lengths are the same as in the axiomatic presentations of higher-order arithmetic.

**Note 2.** With the rule that we use for arithmetic, we cannot extend the proof of the normalization through reducibility candidates as done by Dowek and Werner (2005), or through super consistency by Dowek (2007). This remains currently an open question whether proofs of the natural deduction modulo $\mathcal{HHA}_i^{\text{mod}}$ normalize or not.

## 5. Applications to proof-length speed-ups

Because of Theorem 4.5 and Theorem 4.6, there is obviously no proof-length speed-up between $Z_i$, $FZ$ modulo $\mathcal{HO}_i$ and $\emptyset$ modulo $\mathcal{HHA}_i^{\text{mod}}$. Furthermore, there exists a speed-up between all these and $Z_{i-1}$, which can be decomposed as follow.

5.1. **Speed-up over compatible theories.** In this section, we prove that there exists a speed-up between $FZ$ modulo $\mathcal{HO}_i$ and $FZ$ plus any finite theory compatible with $\mathcal{HO}_i$. Theorem 2.3 makes it not surprising that, if we consider $FZ$ plus a finite theory compatible with $\mathcal{HO}_i$, we get a speed-up with $Z_i$ (or with $FZ$ modulo $\mathcal{HO}_i$). That shows the interest of using deduction modulo.

**Proposition 5.1.** *For all $i$, there is an infinite family $\mathcal{F}$ such that such that for all finite presentations $\Gamma_i$ compatible with $\mathcal{HO}_i$,*

   (1) *for all $P \in \mathcal{F}$, we have $FZ, \Gamma_i \overset{\mathsf{N}}{\vdash} P$*
   (2) *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $FZ \overset{\mathsf{N}}{\underset{k \ steps}{\vdash}}_{\mathcal{HO}_i} P$*
   (3) *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $FZ, \Gamma_i \overset{\mathsf{N}}{\underset{k \ steps}{\vdash}} P$*

*Proof.* As in the proof of Theorem 2.3, we first consider the standard finite presentation $HO_i$ compatible with $\mathcal{HO}_i$, that is, axioms from $(\text{WS}_{nil})$ to $(\text{WS}_{\forall j})$ and axioms $(\text{Comp}_{sk}^j)$. Consider the set of propositions corresponding to all instantiations of the comprehension schema $(\text{Comp}^{i-1})$. In $FZ$ modulo $\mathcal{HO}_i$, these propositions can be proved in five steps as done in Fig. 3. Obviously, $Z_{i-1}$ is not enough to prove all of them, so that $(\text{Comp}_{sk}^{i-1})$ has to be used in the proofs in $FZ, HO_i$. Nevertheless, the term of sort $c$ instantiated in it cannot have a bounded size. Then, the decomposition of this term using $HO_i$ cannot be done in a bounded number of steps. We then use Proposition 2.2 to extend this to any finite presentation compatible with $\mathcal{HO}_i$. $\qquad \square$
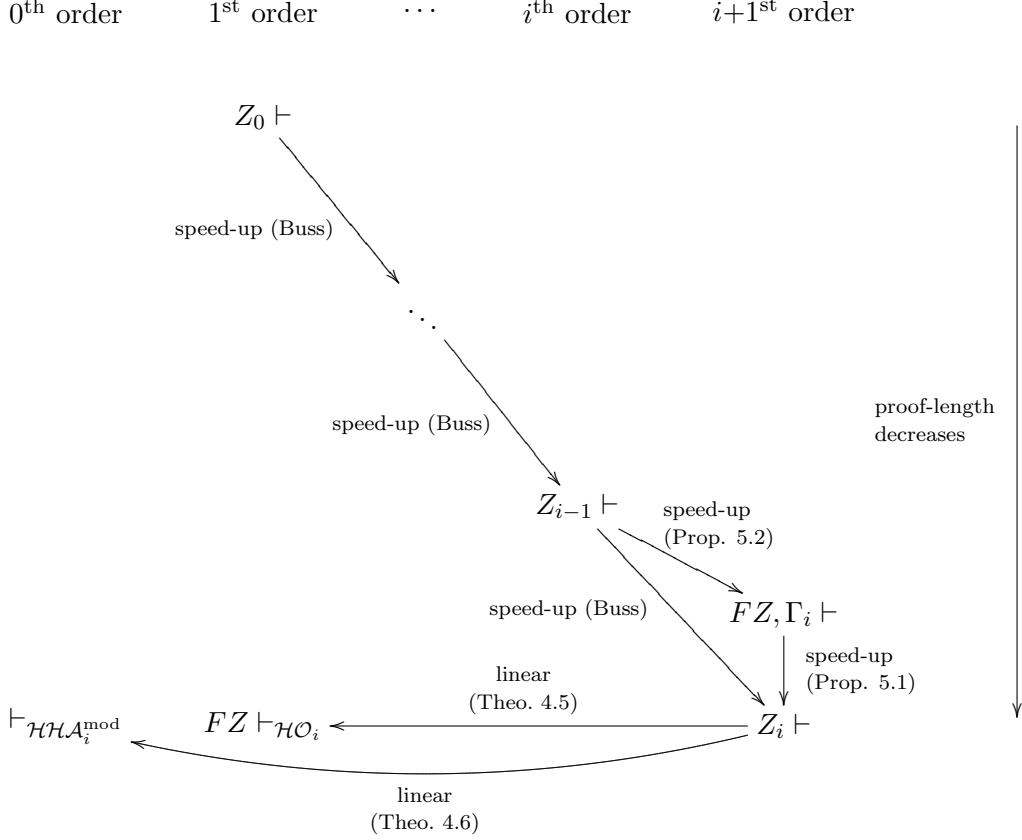
Figure 5: Speed-ups in higher-order arithmetic and deduction modulo

5.2. **Speed-up due to higher orders.** It is also possible to get a speed-up between $FZ$ plus any presentation compatible with $\mathcal{HO}_i$ and $Z_{i-1}$.

**Proposition 5.2.** *For all $i > 0$, there is an infinite family $\mathcal{F}$ such that for all presentations $\Gamma_i$ compatible with $\mathcal{HO}_i$,*

    (1) *for all $P \in \mathcal{F}$, we have $Z_{i-1} \vdash^{\mathsf{N}} P$*
    (2) *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $FZ, \Gamma_i \vdash^{\mathsf{N}}_{k\ steps} P$*
    (3) *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $Z_{i-1} \vdash^{\mathsf{N}}_{k\ steps} P$*

*Proof.* If we look at Buss' proof of Theorem 1.1, the infinite family of propositions he use are of the form $P(n)$ where $\forall n.\ P(n)$ can be proved in $Z_i$ whereas in $Z_{i-1}$, $P(n)$ can be proved, but not with less than $n$ steps. So to get a speed-up it is sufficient to prove that $\forall n.\ P(n)$ can be proved in $FZ$ plus $\Gamma_i$, which is the case because of Theorem 4.5 and (Dowek et al., 2003, Proposition 1.8). We also need Proposition 3.2 to show that if the length of the proofs in $Z_{i-1} \vdash^{\mathsf{N}}$ was bounded, it would be the same in $Z_{i-1} \vdash^{\mathsf{S}}$, hence a contradiction with Theorem 1.1. $\qquad\square$

    The links between the different systems for higher-order arithmetic presented in this paper are summarized in Figure 5.

## 6. Conclusion and discussion

In this paper, we have proposed a rigorous framework to study proof lengths in deduction modulo, by imposing that proofs must be checkable in polynomial time. We have shown that even with this strict condition, proofs in deduction modulo can be arbitrarily shorter than proofs using axiomatizations. We have applied these ideas to study the length of proofs in higher-order arithmetic. We have encoded higher orders as a first-order rewrite system, and proved that proofs have the same length in higher-order arithmetic and in first-order arithmetic modulo this system. We have also defined a system for higher-order arithmetic without resorting to any axiom, where proofs can be checked in polynomial time and have the same length as in the higher-order axiomatization. All these results open interesting issues that we are now discussing.

The first question that arises from this work is the definition of what should be considered as a proof. Until recently, automated theorem provers only answered yes or no (or maybe), and if the prover was correct, this could be considered as a proof. Of course, the correction of such provers, often implemented using low-level tricks to increase the efficiency, is hard to prove. Therefore, many provers now generate certificates that can be checked in more trustworthy provers (such as proof assistants as Coq or Isabelle). These certificates can therefore be considered as proofs, although they may not contain every steps that would be included in an usual formal proof, but only the hints that make it possible to build the formal proof. This idea is also important for proof-carrying codes (Necula, 1997): in this setting, the code of an application is distributed with a certificate proving its correctness. The user of the code can therefore check the correctness using the code and its certificates. It is crucial to have certificates that are small enough, because they are distributed with the code, but that can be checked efficiently, because such codes are often distributed to low-resource systems such as mobile phones. Here again, a tradeoff has to be found between the details present in the certificates and the complexity of their checking. In this paper, we have advocated that the natural criterion to define what a proof is, is that it can be feasibly checked. Of course, depending on the context, this criterion could be relaxed or strengthened.

Another question concerns the role of computation in the speed-ups in higher-order arithmetic. We have proved, at least to some extend, that part of these speed-ups originates from the computation (Proposition 5.1). However, it seems that what really makes proofs shorter is the fact to be able to reason about higher-order objects, even if they are encoded by first-order ones (Proposition 5.2). The real point of our results is that it is possible to use such finite first-order encodings while preserving the length of proofs, at the condition to work modulo some computation.

Another point is that rewrite steps are not counted into the length of the proofs. However, these steps have to be performed when searching for the proofs. We think that the speed-ups we obtained should not be considered as cheating, by hiding part of the proofs in the congruence. This must be thought of as a way to separate what is deduced and what is computed. To find a proof, both parts need to be built. To check the proof however, only the deductive part is necessary, because the rest can be effectively computed during the verification (hence the need to have a decidable congruence, even better if it can be decided in polynomial time). Note that it is also possible to obtain proof-length speed-ups even when counting the rewrite steps in the length of the proofs, as can be shown by transposing

a result of Bruscoli and Guglielmi (2009) where an exponential proof-length speed-up is achieved by applying deduction steps deeply inside propositions.

These results are encouraging indicators that it is as good to work directly in higher-order logics, as is done in the current interactive theorem provers, such as Coq (`http://coq.inria.fr/`) and Isabelle/HOL (Nipkow, Paulson, and Wenzel, 2002), or using a first-order implementation of these logics, as could be done in a proof assistant based on deduction modulo (or on its sequel named superdeduction developed by Brauner, Houtmann, and Kirchner, 2007). It must also be proved that our results extend to the higher-order systems basing the interactive provers. This was partly achieved by proving that functional pure type systems can be encoded in superdeduction in a manner such that typing inferences in the pure type system are translated into proofs in superdeduction of the same length (Burel, 2008). It should also be noticed that in the expression of HOL in the sequent calculus modulo (Dowek et al., 2001), the length of proofs are preserved too, although it was not highlighted by the authors.

## APPENDIX

**Translation from $Z_i \vdash^{\mathbf{N}}$ to $Z_i \vdash^{\mathbf{S}}$.**

$$\mathrm{T}\left( \underset{\Rightarrow\text{-i}}{\dfrac{\pi\{\dfrac{[A]}{B}}{A \Rightarrow B}} \right) \overset{!}{=} \mathrm{T}_A\left( \pi\{ \dfrac{[A]}{B} \right)$$

$$\mathrm{T}\left( \underset{\Rightarrow\text{-e}}{\dfrac{\overset{\pi_1}{A} \qquad \overset{\pi_2}{A \Rightarrow B}}{B}} \right) \overset{!}{=} \underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}(\pi_1)}{A} \qquad \overset{\mathrm{T}(\pi_2)}{A \Rightarrow B}}{B}}$$

$$\mathrm{T}\left( \underset{\wedge\text{-i}}{\dfrac{\overset{\pi_1}{A} \qquad \overset{\pi_2}{B}}{A \wedge B}} \right)$$

$$\overset{!}{=} \underset{\text{(MP)}}{\dfrac{\mathrm{T}(\pi_1) \qquad \underset{\text{(MP)}}{\dfrac{\underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}(\pi_2)}{B} \qquad B \Rightarrow A \Rightarrow B \text{ (K)}}{A \Rightarrow B}} \qquad \underset{\text{(MP)}}{\dfrac{A \Rightarrow A \text{ (I)} \qquad \cdots \text{ (Pair)}}{(A \Rightarrow B) \Rightarrow A \Rightarrow (A \wedge B)}}}{A \Rightarrow (A \wedge B)}}}{A \wedge B}}$$

$$\mathrm{T}\left( \underset{\wedge\text{-e}}{\dfrac{\overset{\pi}{A \wedge B}}{A}} \right) \overset{!}{=} \underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}(\pi)}{A \wedge B} \qquad A \wedge B \Rightarrow A \text{ (Proj}_l)}{A}}$$

and similarly with (Proj$_r$) for the other side.

$$\mathrm{T}\left( \underset{\vee\text{-i}}{\dfrac{\overset{\pi}{A}}{A \vee B}} \right) \overset{!}{=} \underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}(\pi)}{A} \qquad A \Rightarrow (A \vee B) \text{ (Inj}_l)}{A \vee B}}$$

and similarly with (Inj$_r$) for the other side.

$$\mathrm{T}\left( \underset{\vee\text{-e}}{\dfrac{\overset{\pi_1}{A \vee B} \qquad \pi_2\{\overset{[A]}{C} \qquad \pi_3\{\overset{[B]}{C}}{C}} \right)$$

$$\overset{!}{=} \underset{\text{(MP)}}{\dfrac{\mathrm{T}(\pi_1) \qquad \underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}_B(\pi_3)}{B \Rightarrow C} \qquad \underset{\text{(MP)}}{\dfrac{\overset{\mathrm{T}_A(\pi_2)}{A \Rightarrow C} \qquad \cdots \text{ (Case)}}{(B \Rightarrow C) \Rightarrow (A \vee B) \Rightarrow C}}}{(A \vee B) \Rightarrow C}}}{C}}$$

$$\mathrm{T}\left(\forall\text{-i}\ \dfrac{\overset{\pi}{\{y/x\}A}}{\forall x.\ A}\right)\ \overset{!}{=}\ (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}\,(\pi)}{\{y/x\}A}\quad (\mathrm{Gen})\ \dfrac{\{y/x\}A\Rightarrow\{y/x\}A\ (\mathrm{I})}{\{y/x\}A\Rightarrow\forall x.\ A}}{\forall x.\ A}$$

Note that the side conditions are satisfied.

$$\mathrm{T}\left(\forall\text{-e}\ \dfrac{\overset{\pi}{\forall x.\ A}}{\{t/x\}A}\right)\ \overset{!}{=}\ (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}\,(\pi)}{\forall x.\ A}\quad \forall x.\ A\Rightarrow\{t/x\}A\ (\mathrm{UI})}{\{t/x\}A}$$

$$\mathrm{T}\left(\exists\text{-i}\ \dfrac{\overset{\pi}{\{t/x\}A}}{\exists x.\ A}\right)\ \overset{!}{=}\ (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}\,(\pi)}{\{t/x\}A}\quad \{t/x\}A\Rightarrow\exists x.\ A\ (\mathrm{EI})}{\exists x.\ A}$$

$$\mathrm{T}\left(\exists\text{-e}\ \dfrac{\overset{\pi_1}{\exists x.\ A}\quad \pi_2\{\overset{[\{y/x\}A]}{B}}{B}\right)\ \overset{!}{=}\ (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}\,(\pi_1)}{\exists x.\ A}\quad (\mathrm{Part})\ \dfrac{\overset{\mathrm{T}_A\,(\pi_2)}{\{y/x\}A\Rightarrow B}}{(\exists x.\ A)\Rightarrow B}}{B}$$

Note that the side conditions are satisfied.

$$\mathrm{T}\left(classical\ \dfrac{}{A\vee(A\Rightarrow\bot)}\right)\ \overset{!}{=}\ A\vee(A\Rightarrow\bot)\ (\mathrm{TND})$$

$$\mathrm{T}\left(\bot\text{-e}\ \dfrac{\overset{\pi}{\bot}}{A}\right)\ \overset{!}{=}$$

$$(\mathrm{MP})\ \dfrac{A\Rightarrow A\ (\mathrm{I})\quad (\mathrm{MP})\ \dfrac{(\mathrm{MP})\ \dfrac{\overset{\mathrm{T}\,(\pi)}{\bot}\quad \bot\Rightarrow(A\Rightarrow A)\Rightarrow\bot\ (\mathrm{K})}{(A\Rightarrow A)\Rightarrow\bot}\quad \cdots\ (\mathrm{EFSQ})}{(A\Rightarrow A)\Rightarrow A}}{A}$$

$$\mathrm{T}\,(\mathrm{A})\ \overset{!}{=}\ \mathrm{A}$$

$$\mathrm{T}_A\left(\Rightarrow\text{-i}\ \dfrac{\pi\{\overset{[B]}{C}}{B\Rightarrow C}\right)\ \overset{!}{=}\ \mathrm{T}_A\left(\dfrac{\mathrm{T}_B\,(\pi)}{B\Rightarrow C}\right)$$

$$\mathrm{T}_A\left(\Rightarrow\text{-e}\ \dfrac{\pi_1\{\overset{[A]}{B}\quad \pi_2\{\overset{[A]}{B\Rightarrow C}}{C}\right)$$

$$\overset{!}{=}\ (\mathrm{MP})\ \dfrac{(\mathrm{MP})\ \dfrac{\overset{\mathrm{T}_A\,(\pi_2)}{A\Rightarrow B\Rightarrow C}\quad \cdots\ (\mathrm{C})}{B\Rightarrow A\Rightarrow C}\quad (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}_A\,(\pi_1)}{A\Rightarrow B}\quad \cdots\ (\mathrm{B})}{(B\Rightarrow A\Rightarrow C)\Rightarrow A\Rightarrow A\Rightarrow C}}{\dfrac{A\Rightarrow A\Rightarrow C\quad \cdots\ (\mathrm{W})}{A\Rightarrow C}}$$

$$\mathrm{T}_A\left(\wedge\text{-i}\ \dfrac{\pi_1\{\overset{[A]}{B}\quad \pi_2\{\overset{[A]}{C}}{B\wedge C}\right)$$

$$\overset{!}{=}\ (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}_A\,(\pi_2)}{A\Rightarrow C}\quad (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}_A\,(\pi_1)}{A\Rightarrow B}\quad (A\Rightarrow B)\Rightarrow(A\Rightarrow C)\Rightarrow A\Rightarrow(B\wedge C)\ (\mathrm{Pair})}{(A\Rightarrow C)\Rightarrow A\Rightarrow(B\wedge C)}}{A\Rightarrow(B\wedge C)}$$

$$\mathrm{T}_A\left(\wedge\text{-e}\ \dfrac{\pi\{\overset{[A]}{B\wedge C}}{B}\right)\ \overset{!}{=}\ (\mathrm{MP})\ \dfrac{(B\wedge C)\Rightarrow B\ (\mathrm{Proj}_l)\quad (\mathrm{MP})\ \dfrac{\overset{\mathrm{T}_A\,(\pi)}{A\Rightarrow(B\wedge C)}\quad \cdots\ (\mathrm{B})}{((B\wedge C)\Rightarrow B)\Rightarrow A\Rightarrow B}}{A\Rightarrow B}$$

and similarly with $(\mathrm{Proj}_r)$ for the other side.

$$\mathrm{T}_A \left( \begin{array}{c} \pi\{ \begin{array}{c} [A] \\ B \end{array} \\ \vee\text{-i} \,\dfrac{}{B \vee C} \end{array} \right) \;\overset{!}{=}\; (\mathrm{MP}) \dfrac{B \Rightarrow (B \vee C) \;(\mathrm{Inj}_l) \qquad (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi) \\ A \Rightarrow B \qquad \cdots\;(\mathrm{B})}{(B \Rightarrow (B \vee C)) \Rightarrow A \Rightarrow (B \vee C)}}{A \Rightarrow (B \vee C)}$$

and similarly with $(\mathrm{Inj}_r)$ for the other side.

$$\mathrm{T}_A \left( \begin{array}{c} \pi_1\{ \begin{array}{cc} [A] \\ B \vee C \end{array} \quad \pi_2\{ \begin{array}{c} [A,B] \\ D \end{array} \quad \pi_3\{ \begin{array}{c} [A,C] \\ D \end{array} \\ \vee\text{-e} \,\dfrac{}{D} \end{array} \right) \;\overset{!}{=}\;$$

$$(\mathrm{MP}) \dfrac{\mathrm{T}_C \left( \begin{array}{c} \mathrm{T}_A\,(\pi_3) \\ A \Rightarrow D \end{array} \right) \\ C \Rightarrow A \Rightarrow D \qquad (\mathrm{MP}) \dfrac{\mathrm{T}_B \left( \begin{array}{c} \mathrm{T}_A\,(\pi_2) \\ A \Rightarrow D \end{array} \right) \\ B \Rightarrow A \Rightarrow D \qquad \cdots\;(\mathrm{Case})}{(C \Rightarrow A \Rightarrow D) \Rightarrow (B \vee C) \Rightarrow A \Rightarrow D}}{(\mathrm{MP}) \dfrac{(B \vee C) \Rightarrow A \Rightarrow D \qquad (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi_1) \\ A \Rightarrow (B \vee C) \qquad \cdots\;(\mathrm{B})}{((B \vee C) \Rightarrow A \Rightarrow D) \Rightarrow A \Rightarrow A \Rightarrow D}}{A \Rightarrow A \Rightarrow D}}$$

$$(\mathrm{MP}) \dfrac{\vdots \qquad\qquad \cdots\;(\mathrm{W})}{A \Rightarrow D}$$

$$\mathrm{T}_A \left( \begin{array}{c} \pi\{ \begin{array}{c} [A] \\ \{y/x\}B \end{array} \\ \forall\text{-i} \,\dfrac{}{\forall x.\ B} \end{array} \right) \;\overset{!}{=}\; (\mathrm{Gen}) \dfrac{\mathrm{T}_A\,(\pi) \\ A \Rightarrow \{y/x\}B}{A \Rightarrow \forall x.\ B}$$

Note that the side conditions are satisfied.

$$\mathrm{T}_A \left( \begin{array}{c} \pi\{ \begin{array}{c} [A] \\ \forall x.\ B \end{array} \\ \forall\text{-e} \,\dfrac{}{\{t/x\}B} \end{array} \right)$$

$$\overset{!}{=}\; (\mathrm{MP}) \dfrac{(\forall x.\ B) \Rightarrow \{t/x\}B \;(\mathrm{UI}) \qquad (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi) \\ A \Rightarrow \forall x.\ B \qquad \cdots\;(\mathrm{B})}{((\forall x.\ B) \Rightarrow \{t/x\}B) \Rightarrow A \Rightarrow \{t/x\}B}}{A \Rightarrow \{t/x\}B}$$

$$\mathrm{T}_A \left( \begin{array}{c} \pi\{ \begin{array}{c} [A] \\ \{t/x\}B \end{array} \\ \exists\text{-i} \,\dfrac{}{\exists x.\ B} \end{array} \right)$$

$$\overset{!}{=}\; (\mathrm{MP}) \dfrac{\{t/x\}B \Rightarrow \exists x.\ B \;(\mathrm{EI}) \qquad (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi) \\ A \Rightarrow \{t/x\}B \qquad \cdots\;(\mathrm{B})}{(\{t/x\}B \Rightarrow \exists x.\ B) \Rightarrow A \Rightarrow \exists x.\ B}}{A \Rightarrow \exists x.\ B}$$

$$\mathrm{T}_A \left( \begin{array}{c} \pi_1\{ \begin{array}{cc} [A] \\ \exists x.\ B \end{array} \quad \pi_2\{ \begin{array}{c} [A,\{y/x\}B] \\ C \end{array} \\ \exists\text{-e} \,\dfrac{}{C} \end{array} \right)$$

$$\overset{!}{=}\; (\mathrm{MP}) \dfrac{(\mathrm{Part}) \dfrac{\mathrm{T}_B \left( \begin{array}{c} \mathrm{T}_A\,(\pi_2) \\ A \Rightarrow C \end{array} \right) \\ \{y/x\}B \Rightarrow A \Rightarrow C}{\exists x.\ B \Rightarrow A \Rightarrow C} \quad (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi_1) \\ A \Rightarrow \exists x.\ B \qquad \dots\;(\mathrm{B})}{(\exists x.\ B \Rightarrow A \Rightarrow C) \Rightarrow A \Rightarrow A \Rightarrow C}}{(\mathrm{MP}) \dfrac{A \Rightarrow A \Rightarrow C \qquad \cdots\;(\mathrm{W})}{A \Rightarrow C}}$$

Note that the side conditions are satisfied.

$$\mathrm{T}_A \left( \begin{array}{c} \pi\{ \begin{array}{c} [A] \\ \bot \end{array} \\ \bot\text{-e} \,\dfrac{}{B} \end{array} \right) \;\overset{!}{=}\; (\mathrm{MP}) \dfrac{\mathrm{T}_A\,(\pi) \\ A \Rightarrow \bot \qquad (A \Rightarrow \bot) \Rightarrow A \Rightarrow B \;(\mathrm{EFSQ})}{A \Rightarrow B}$$

$$\mathrm{T}_A\,(A) \;\overset{!}{=}\; A \Rightarrow A \;(\mathrm{I})$$

$$\mathrm{T}_A \left( \begin{array}{c} \pi \\ B \end{array} \right) \overset{!}{=} \;\; \text{(MP)} \dfrac{\overset{\displaystyle \mathrm{T}(\pi)}{B} \qquad B \Rightarrow A \Rightarrow B \;\text{(K)}}{A \Rightarrow B} \quad \begin{array}{l} \text{if the assumption } A \text{ is not actually} \\ \text{used in } \pi. \end{array}$$

The definition of $\mathrm{T}_A$ for $\Rightarrow$-i is not looping, because they are no longer $\Rightarrow$-i in $\mathrm{T}_B(\pi)$. Nevertheless, this case impose use to define what $\mathrm{T}_A$ means for a proof using the inference rules (Gen) and (Part). (The translation of (MP) is already defined because (MP) is equal to $\Rightarrow$-e.)

$$\mathrm{T}_A \left( \;\; \text{(Gen)} \dfrac{\overset{\displaystyle \pi\{ \overset{\displaystyle [A]}{B \Rightarrow C(\tau)}}{}}{B \Rightarrow \forall \alpha.\ C(\alpha)} \;\; \right)$$

$$\overset{!}{=} \;\; \text{(MP)} \dfrac{\dfrac{\overset{\displaystyle \mathrm{T}_A(\pi)}{A \Rightarrow B \Rightarrow C(\tau)} \qquad (A \Rightarrow B \Rightarrow C(\tau)) \Rightarrow (A \wedge B) \Rightarrow C(\tau)}{\text{(Gen)} \dfrac{(A \wedge B) \Rightarrow C(\tau)}{(A \wedge B) \Rightarrow \forall \alpha.\ C(\alpha)}} \qquad \overset{\displaystyle \varpi_1}{} \qquad \overset{\displaystyle \varpi_2}{\cdots}}{A \Rightarrow B \Rightarrow \forall \alpha.\ C(\alpha)} \;\text{(MP)}$$

where $\varpi_1$ is any proof of $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \wedge B) \Rightarrow C$, and $\varpi_2$ of $((A \wedge B) \Rightarrow C) \Rightarrow A \Rightarrow B \Rightarrow C$, using the axiom schemata (I) to (Pair) and the inference rule (MP). (Indeed, they are valid propositions of the intuitionistic propositional logic.)

$$\mathrm{T}_A \left( \;\; \text{(Part)} \dfrac{\overset{\displaystyle \pi\{ \overset{\displaystyle [A]}{B(\tau) \Rightarrow C}}{}}{(\exists \alpha.\ B(\alpha)) \Rightarrow C} \;\; \right)$$

$$\overset{!}{=} \;\; \text{(MP)} \dfrac{\dfrac{\overset{\displaystyle \mathrm{T}_A(\pi)}{A \Rightarrow B(\tau) \Rightarrow C} \qquad (A \Rightarrow B(\tau) \Rightarrow C) \Rightarrow B(\tau) \Rightarrow A \Rightarrow C \;\text{(C)}}{\text{(Part)} \dfrac{B(\tau) \Rightarrow A \Rightarrow C}{\exists \alpha.\ B(\alpha) \Rightarrow A \Rightarrow C}} \qquad \cdots \;\text{(C)}}{A \Rightarrow \exists \alpha.\ B(\alpha) \Rightarrow C} \;\text{(MP)}$$

## References

BAADER, F. AND NIPKOW, T. 1998. *Term Rewriting and* all That. Cambridge University Press.

BRAUNER, P., HOUTMANN, C., AND KIRCHNER, C. 2007. Principle of superdeduction. In *Proceedings of LICS*, L. Ong, Ed. IEEE Computer Society, 41–50.

BRÜNNLER, K. 2003. Deep inference and symmetry in classical proofs. Ph.D. thesis, Technische Universität Dresden.

BRUSCOLI, P. AND GUGLIELMI, A. 2009. On the proof complexity of deep inference. *ACM Transactions on Computational Logic 10*, 2, 1–34.

BUREL, G. 2007. Unbounded proof-length speed-up in deduction modulo. In *CSL*, J. Duparc and T. Henziger, Eds. Lecture Notes in Computer Science, vol. 4646. Springer-Verlag, 496–511.

BUREL, G. 2008. A first-order representation of pure type systems using superdeduction. In *LICS*, F. Pfenning, Ed. IEEE Computer Society, 253–263.

BUSS, S. R. 1987. Polynomial size proofs of the propositional pigeonhole principle. *The Journal of Symbolic Logic 52*, 4, 916–927.

BUSS, S. R. 1994. On Gödel's theorems on lengths of proofs I: Number of lines and speedup for arithmetics. *The Journal of Symbolic Logic 59*, 3, 737–756.

COOK, S. A. AND RECKHOW, R. A. 1974. On the lengths of proofs in the propositional calculus (preliminary version). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*. ACM, 135–148.

COOK, S. A. AND RECKHOW, R. A. 1979. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic 44,* 1, 36–50.

COUSINEAU, D. AND DOWEK, G. 2007. Embedding pure type systems in the lambda-pi-calculus modulo. In *TLCA*, S. Ronchi Della Rocca, Ed. Lecture Notes in Computer Science, vol. 4583. Springer-Verlag, 102–117.

CURRY, H. B., FEYS, R., AND CRAIG, W. 1958. *Combinatory Logic.* Vol. 1. Elsevier Science Publishers B. V. (North-Holland), Amsterdam.

DOWEK, G. 2007. Truth values algebras and proof normalization. In *TYPES*, T. Altenkirch and C. McBride, Eds. Lecture Notes in Computer Science, vol. 4502. Springer-Verlag, 110–124.

DOWEK, G., HARDIN, T., AND KIRCHNER, C. 2001. HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science 11,* 1, 1–25.

DOWEK, G., HARDIN, T., AND KIRCHNER, C. 2003. Theorem proving modulo. *Journal of Automated Reasoning 31,* 1, 33–72.

DOWEK, G. AND MIQUEL, A. 2006. Cut elimination for Zermelo's set theory. Available on authors' web page.

DOWEK, G. AND WERNER, B. 2003. Proof normalization modulo. *The Journal of Symbolic Logic 68,* 4, 1289–1316.

DOWEK, G. AND WERNER, B. 2005. Arithmetic as a theory modulo. In *RTA*, J. Giesl, Ed. LNCS, vol. 3467. Springer, 423–437.

GALLIER, J. H. 1986. *Logic for Computer Science: Foundations of Automatic Theorem Proving.* Computer Science and Technology Series, vol. 5. Harper & Row, New York. Revised On-Line Version (2003), http://www.cis.upenn.edu/~jean/gbooks/logic.html.

GENTZEN, G. 1934. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift 39*, 176–210, 405–431.

GÖDEL, K. 1936. Über die Länge von Beweisen. *Ergebnisse eines Mathematischen Kolloquiums 7*, 23–24. English translation in Gödel (1986).

GÖDEL, K. 1986. On the length of proofs. In *Kurt Gödel: Collected Works*, S. Feferman et al., Eds. Vol. 1. Oxford University Press, Oxford, 396–399.

KIRCHNER, F. 2007. A finite first-order theory of classes. In *TYPES*, T. Altenkirch and C. McBride, Eds. Lecture Notes in Computer Science, vol. 4502. Springer-Verlag, 188–202.

KRAJÍČEK, J. 1989. On the number of steps in proofs. *Annals of Pure and Applied Logic 41,* 2, 153–178.

MOSTOWSKI, A., ROBINSON, R. M., AND TARSKI, A. 1953. *Undecidable Theories.* Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam.

NECULA, G. C. 1997. Proof-carrying code. In *Proceedings of the 24th ACM Symposium on Principles of Programming Languages*. ACM.

NIPKOW, T., PAULSON, L. C., AND WENZEL, M. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic.* Lecture Notes in Computer Science, vol. 2283. Springer-Verlag.

PARIKH, R. J. 1973. Some results on the length of proofs. *Transactions of the ACM 177*, 29–36.

POINCARÉ, H. 1902. *La Science et l'Hypothèse.* Flammarion.

van Dalen, D. 1989. *Logic and Structure*, Second ed. Universitext. Springer-Verlag.

Wack, B. 2005. Typage et déduction dans le calcul de réécriture. Ph.D. thesis, Université H. Poincaré (Nancy 1).