# Efficiently Simulating Higher-Order Arithmetic by a First-Order Theory Modulo

Guillaume Burel

# Efficiently Simulating Higher-Order Arithmetic by a First-Order Theory Modulo

GUILLAUME BUREL
Nancy-Université & LORIA

Deduction modulo is a paradigm which consists in applying the inference rules of a deductive system—such as for instance natural deduction—modulo a rewrite system over terms and propositions. It has been shown that higher-order logic can be simulated into the first-order natural deduction modulo. However, a theorem stated by Gödel and proved by Parikh expresses that proofs in second-order arithmetic may be unboundedly shorter than proofs in first-order arithmetic, even when considering only formulæ provable in first-order arithmetic. We investigate how deduction modulo can be used to translate proofs of higher-order arithmetic into first-order proofs without inflating their length.

First we show how higher orders can be encoded through a quite simple (finite, terminating, confluent, left-linear) rewrite system. Then, a proof in higher-order arithmetic can be linearly translated into a proof in first-order arithmetic modulo this system. Second, in the continuation of a work of Dowek and Werner, we show how to express the whole higher-order arithmetic as a rewrite system. Then, proofs of higher-order arithmetic can be linearly translated into proofs in the empty theory modulo this rewrite system. These results show that the speed-up between first- and second-order arithmetic, and more generally between $i^{\text{th}}$- and $i+1^{\text{st}}$-order arithmetic, can in fact be expressed as computation, and does not lie in the really deductive part of the proofs.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Complexity of proof procedures*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Proof theory*; *Mechanical theorem proving*

General Terms: Theory

Additional Key Words and Phrases: Arithmetic, deduction modulo, higher-order logic, proof-length speed-ups, term rewriting

## 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 intro-

Author's address: INRIA Team Pareo – Bâtiment B, Campus Scientifique, BP 239, 54506 Vandœuvre-lès-Nancy cedex, France
Author's email: guillaume.burel@ens-lyon.org
Author's homepage: http://www.loria.fr/~burel/
LORIA is the UMR 7503 CNRS–INRIA–Nancy-Université.
This paper is an updated and extended version of Burel [2007].

ductory 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 [2008]). 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 formulæ appearing in them is.

An interesting open issue concerning proof lengths is whether adding Tseytin's extensions [Tseytin 1968] to a deductive systems leads to exponentially shorter proofs. Such extensions consist in having for each formula $P$ a fresh atomic proposition $A_P$ and new inference rules

$$\frac{P}{A_P} \qquad \frac{A_P}{P} \ .$$

These extensions can be related to the comprehension axiom schema of higher-order arithmetic:

$$\exists\alpha^{j+1}.\ \forall\beta^{j}.\ \beta^{j} \in^{j} \alpha^{j+1} \Leftrightarrow A(\beta^{j}) \qquad (\alpha^{j+1} \text{ is not free in } A(\beta^{j})) \tag{35}$$

Indeed, it can be used to get the following derivations in natural deduction:

$$\exists\text{-e}\ \frac{\exists\alpha^{j+1}.\ \forall\beta^{j}.\ \beta^{j} \in^{j} \alpha^{j+1} \Leftrightarrow A(\beta^{j})\ (35) \qquad \begin{array}{c}\forall\beta^{j}.\ \beta^{j} \in^{j} \alpha^{j+1} \Leftrightarrow A(\beta^{j})\ (\mathrm{i}) \\ \hline t \in^{j} \alpha^{j+1} \Leftrightarrow A(t) \\ \hline t \in^{j} \alpha^{j+1} \Rightarrow A(t) \end{array} \quad \begin{array}{c} t \in^{j} \alpha^{j+1} \\ \vdots \\ A(t) \end{array}}{A(t)}\ (\mathrm{i})$$

$$\exists\text{-e}\ \frac{\exists\alpha^{j+1}.\ \forall\beta^{j}.\ \beta^{j} \in^{j} \alpha^{j+1} \Leftrightarrow A(\beta^{j})\ (35) \qquad \begin{array}{c}\forall\beta^{j}.\ \beta^{j} \in^{j} \alpha^{j+1} \Leftrightarrow A(\beta^{j})\ (\mathrm{i}) \\ \hline t \in^{j} \alpha^{j+1} \Leftrightarrow A(t) \\ \hline A(t) \Rightarrow t \in^{j} \alpha^{j+1} \end{array} \quad \begin{array}{c} A(t) \\ \vdots \\ t \in^{j} \alpha^{j+1} \end{array}}{t \in^{j} \alpha^{j+1}}\ (\mathrm{i})$$

So, $\cdot \in^{j} \alpha^{j+1}$ plays the same role to $A(\cdot)$ as $A_P$ to $P$, except that it is a unary predicate instead of a proposition.

The comprehension axiom schema (35) is what separate first-order arithmetic from higher-order arithmetic, and it has been proved by Parikh [1973] that second-order arithmetic indeed 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 BUSS [1994, THEOREM 3]. *Let $i \geq 0$. Then there is an infinite family $\mathcal{F}$ of $\prod_1^0$-formulæ 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 $P \in \mathcal{F}$, $Z_{i+1} \vdash_{\overline{k \text{ steps}}} P$*
*(3) there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i \vdash_{\overline{k \text{ steps}}} P$.*

where $Z_i$ corresponds to the $i + 1^{\text{st}}$-order arithmetic (so $Z_0$ is in fact first-order arithmetic), and $Z_i \vdash_{\overline{k \text{ steps}}} P$ 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.)

Because this theorem is concerned in arithmetic, an intuitive notion of computation takes place in the proofs. Indeed, as remarked by Poincaré, establishing that $2 + 2 = 4$ using the definition of the addition is just a verification, and not a demonstration, so that in a proof occur in fact not only pure deduction but also computation. Therefore, the question arises whether this speed-up comes from the deductive or the computational part of the proofs, or both of them. Of course, the difference between computation and deduction cannot be clearly determined. Because of the Curry-Howard correspondence, the whole content of the proofs could be considered as computation. (Concerning proofs as programs and arithmetic, see Schwichtenberg [2007].) Here, this difference must be thought of as the distinction between what is straightforward (at least decidable), and what must be reasoned out.

Deduction modulo [Dowek et al. 2003] is a proposal to identify what corresponds to deduction and to computation in a proof. The computational part of a proof is put in a congruence between formulæ 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 better represented 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 formula, so that the following simple natural-deduction-modulo proof of $t \times t = 0$ can be deduced from a proof $\pi$ of $t = 0$:

$$\vee\text{-i}\,\frac{\begin{array}{c}\pi\\ t = 0\end{array}}{t \times t = 0}\; t \times t = 0 \;\longrightarrow\; t = 0 \vee t = 0 \quad .$$

In deduction modulo we can distinguish between two kind of theories (denominations are originated by Allali [2007]): axiomatic theories are usual sets of axioms used as assumptions in proofs; purely computational theories are term and proposition rewrite systems modulo which inference rules are applied. A modulo theory is the combination of an axiomatic theory and a purely computational theory.

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. This paper answers this issue. Of course, if there are no restriction on the rewrite rules that are used (for instance if it is allowed to use a rewrite system semi-deciding the validity of formulæ), it is not surprising that the length of the proofs can be unboundedly reduced. Notwithstanding, the first rewriting system that we will consider in this paper for encoding higher order is very simple: is finite, terminating, confluent (i.e. deterministic) and linear (variables in the left-hand side only appear once). Note that if infinite rewrite systems are allowed, it is easy to get Tseytin's extension by working modulo the rewrite rules $A_P \to P$ for all $P$.

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, however 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.

To prove that the origin of the speed-up theorem of Buss can be expressed as simple computation, we proceed in two steps: First, we show how to encode higher order using a rewrite system and a finite set of axioms, by generalizing the work of Kirchner [2006]. This rewrite system, despite its simplicity, permits to obtain the proof-length speed-up. Second, extending the work of Dowek and Werner [2005], we will express higher-order arithmetic as a purely computational theory, that is a rewrite system modulo which the inference rules of natural deduction will take place without the need of external assumptions. 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 order and the formulation of first-order arithmetic by Dowek and Werner [2005], because the latter do not preserve the length of proofs. We define a formulation of higher-order arithmetic which has the same speed-up over first-order arithmetic. As a technical detail, because Theorem 1 is proved only for schematic systems, we will begin by showing that proof lengths in schematic systems and natural deduction do not differ too much.

In [2007], we also looked at the relations between computations and proof-length speed-ups. There are three main improvements here. First, the encoding of higher order in [2007] was not completely done: there remained axioms in which higher-

order function symbols were involved. A proof of $Z_i$ was therefore not translated into a proof of $Z_{i-1}$ modulo, but in a proof of $Z_{i-1}+T_i$ modulo, where $T_i$ contained function symbols of order $i+1$. This is no longer the case when working modulo $\mathcal{HO}_i$ in this paper. Second, we had length-preserving translations from $Z_{i+1}$ to $Z_i$ modulo, and from $Z_i$ to $Z_{i-1}$ modulo, but it was not clear if they could be combined to get a length-preserving translation from $Z_{i+1}$ to $Z_{i-1}$ modulo. We show here that we can in fact have length-preserving translations from $Z_i$ to $Z_0$ modulo, for all $i \geq 0$. Third, we succeed in characterizing the whole higher-order arithmetic as a purely computational theory, while conserving the length of proofs.

In the next section, we will recall the definition of a schematic system, and we will present such a system for $i^{\text{th}}$-order arithmetic. The section 3 will define formally what deduction modulo, and in particular natural deduction modulo, consists in. In Section 4 we will give *bounded* translations between the schematic system for $i^{\text{th}}$-order arithmetic and natural deduction. Then, the main section 5 will present how to efficiently encode higher order, and then higher-order arithmetic. Finally, in Section 6 we will apply these results to determine the origin of the speed-ups in arithmetic. We will conclude about the interest of working within a first-order system modulo to simulate higher order.

## 2. A SCHEMATIC SYSTEM FOR $I^{\text{TH}}$-ORDER ARITHMETIC

### 2.1 Schematic systems

We recall here, using Buss' terminology [1994], what a schematic system consists in. It is essentially an Hilbert-type (or Frege) proof system, i.e. valid formulæ are derived from a finite number of axiom schemata using a finite number of inference rules. Theorem 1 is true on condition that proofs are performed using a schematic system.

First, we recall how to build many-sorted first-order formulæ (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 $A(t_1, \ldots, t_n)$. *Formulæ* can be built using the following grammar[1]:

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

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 formula, 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 formula $P$ is denoted by $\{t/x\}P$, the subterm or subformula of $t$ at the position $\mathfrak{p}$ by $t_{|\mathfrak{p}}$, and its replacement in $t$ by a term or formula $s$ by $t[s]_{\mathfrak{p}}$.

Then, given a many-sorted signature of first-order logic, we can consider infinite

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

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 formulæ).

Metaterms are built like terms, except that they can contain metavariables and term variables. Metaformulæ are built like formulæ, 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 metaformulæ (the *premises*), a metaformulæ (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 metaformula and $s$ a metaterm of sort $j$. It is denoted by

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

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

### 2.2 $i^{\text{th}}$-order arithmetic

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

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

The schematic system we use here consists of the following inference rules:
**$14 + 2 \times$ i axiom schemata of classical logic.** We take the one used by Gentzen [1934, Chapter 5] to prove the equivalence of his formalisms with an Hilbert-type proof system:

$$A \Rightarrow A \tag{1}$$
$$A \Rightarrow B \Rightarrow A \tag{2}$$
$$(A \Rightarrow A \Rightarrow B) \Rightarrow A \Rightarrow B \tag{3}$$
$$(A \Rightarrow B \Rightarrow C) \Rightarrow B \Rightarrow A \Rightarrow C \tag{4}$$
$$(A \Rightarrow B) \Rightarrow (B \Rightarrow C) \Rightarrow A \Rightarrow C \tag{5}$$
$$(A \wedge B) \Rightarrow A \tag{6}$$
$$(A \wedge B) \Rightarrow B \tag{7}$$
$$(A \Rightarrow B) \Rightarrow (A \Rightarrow C) \Rightarrow A \Rightarrow (B \wedge C) \tag{8}$$
$$A \Rightarrow (A \vee B) \tag{9}$$
$$B \Rightarrow (A \vee B) \tag{10}$$
$$(A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow (A \vee B) \Rightarrow C \tag{11}$$
$$(A \Rightarrow B) \Rightarrow (A \Rightarrow B \Rightarrow \bot) \Rightarrow A \Rightarrow \bot \tag{12}$$
$$(A \Rightarrow \bot) \Rightarrow A \Rightarrow B \tag{13}$$

$$(\forall \alpha^j.\ A(\alpha^j)) \Rightarrow A(\tau^j) \tag{14}$$
$$\big(\tau^j \text{ is freely substitutable for } \alpha^j \text{ in } A(\alpha^j)\big)$$
$$A(\tau^j) \Rightarrow \exists \alpha^j.\ A(\alpha^j) \tag{15}$$
$$\big(\tau^j \text{ is freely substitutable for } \alpha^j \text{ in } A(\alpha^j)\big)$$
$$A \vee (A \Rightarrow \bot) \tag{16}$$

**$1 + 2 \times$ i inference rules of classical logic.** Again, we take the one used by Gentzen [1934]:

$$\frac{A \qquad A \Rightarrow B}{B} \tag{17}$$

$$\frac{A \Rightarrow B(\beta^j)}{A \Rightarrow \forall \alpha^j.\ B(\alpha^j)}\ (\beta^j \text{ is not free in } A \Rightarrow \forall \alpha^j.\ B(\alpha^j)) \tag{18}$$

$$\frac{B(\beta^j) \Rightarrow A}{(\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A}\ (\beta^j \text{ is not free in } (\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A) \tag{19}$$

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

$$\forall \alpha^0.\ \alpha^0 = \alpha^0 \tag{20}$$
$$\forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow s(\alpha^0) = s(\beta^0) \tag{21}$$
$$\forall \alpha^0 \beta^0 \gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \alpha^0 + \gamma^0 = \beta^0 + \gamma^0 \tag{22}$$
$$\forall \alpha^0 \beta^0 \gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \gamma^0 + \alpha^0 = \gamma^0 + \beta^0 \tag{23}$$
$$\forall \alpha^0 \beta^0 \gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \alpha^0 \times \gamma^0 = \beta^0 \times \gamma^0 \tag{24}$$
$$\forall \alpha^0 \beta^0 \gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \gamma^0 \times \alpha^0 = \gamma^0 \times \beta^0 \tag{25}$$
$$\forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0) \tag{26}$$

**7 Robinson's axioms.** They are the axioms defining the function symbols of arithmetic [Mostowski et al. 1953]:

$$\forall \alpha^0.\ \neg\ 0 = s(\alpha^0) \tag{27}$$
$$\forall \alpha^0 \beta^0.\ s(\alpha^0) = s(\beta^0) \Rightarrow \alpha^0 = \beta^0 \tag{28}$$
$$\forall \alpha^0.\ (\neg\ \alpha^0 = 0) \Rightarrow \exists \beta^0.\ \alpha^0 = s(\beta^0) \tag{29}$$
$$\forall \alpha^0.\ \alpha^0 + 0 = \alpha^0 \tag{30}$$
$$\forall \alpha^0 \beta^0.\ \alpha^0 + s(\beta^0) = s(\alpha^0 + \beta^0) \tag{31}$$
$$\forall \alpha^0.\ \alpha^0 \times 0 = 0 \tag{32}$$
$$\forall \alpha^0 \beta^0.\ \alpha^0 \times s(\beta^0) = \alpha^0 \times \beta^0 + \alpha^0 \tag{33}$$

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

$$A(0) \Rightarrow \big(\forall \beta^0.\ A(\beta^0) \Rightarrow A(s(\beta^0))\big) \Rightarrow \forall \alpha^0.\ A(\alpha^0) \tag{34}$$

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

$$\exists \alpha^{j+1}.\ \forall \beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j) \qquad (\alpha^{j+1} \text{ is not free in } A(\beta^j)) \tag{35}$$

From this point on, we will denote by $Z_{i-1} \vdash^{\mathsf{S}}_{k} P$ 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.

## 3. DEDUCTION MODULO

### 3.1 Rewriting formulæ

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, formulæ are considered modulo some congruence defined by some rules that rewrite not only terms but also formulæ. We use standard definitions, as given by Baader and Nipkow [1998], and extend them to proposition rewriting [Dowek et al. 2003].

A *term rewrite rule* is the pair of terms $l, r$ such that all free variables of $r$ appear in $l$. It is denoted $l \to r$. 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 $l \to r$ 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 $l \to r$ if $s_i$ can be rewritten to $t_i$ by $l \to r$. This relation is extended by congruence to all formulæ.

A *proposition rewrite rule* is the pair of an atomic proposition $A$ and a formula $P$, such that all free variables of $P$ appear in $A$. It is denoted $A \to P$. A *proposition rewrite system* is a set of proposition rewrite rules. A formula $Q$ can be rewritten to a formula $R$ by a proposition rewrite rule $A \to P$ 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 formulæ.

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 $A \underset{\mathcal{R}}{\longrightarrow} P$. The transitive (resp. reflexive transitive) closure of this relation will be denoted by $\underset{\mathcal{R}}{\overset{*}{\longrightarrow}}$ (resp. $\underset{\mathcal{R}}{\overset{*}{\longleftrightarrow}}$).

### 3.2 Natural deduction modulo

Using some equivalence $\underset{\mathcal{R}}{\overset{*}{\longleftrightarrow}}$ 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 of a proof that are not introduced by some inference rules (contrary to $A$ in $\Rightarrow$-i for instance) are the assumptions of the proof. Note that if we do not work modulo, $\Rightarrow$-e is exactly the same as (17).

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 $\mathcal{T} \vdash^{\mathsf{N}}_{k} P$. Abusing notations, we will write $Z_i \vdash^{\mathsf{N}}_{k\,\mathcal{R}} P$ to say that there is a proof of $P$ of length at most $k$ using as assumptions a finite subset of instances of the axiom schemata (20) to (35).

*Definition* 1 *Compatible theory [Dowek et al. 2003, Definition 1.4].* A theory $\mathcal{T}$ is said *compatible* with a rewrite system $\mathcal{R}$ if:

$$\Rightarrow\text{-i}\ \frac{\begin{array}{c}[A]\\ B\end{array}}{C}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \Rightarrow B$$

$$\Rightarrow\text{-e}\ \frac{A \qquad C}{B}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \Rightarrow B$$

$$\wedge\text{-i}\ \frac{A \qquad B}{C}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \wedge B$$

$$\wedge\text{-e}\ \frac{C}{A}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \wedge B \text{ or } C \xleftrightarrow[\mathcal{R}]{*} B \wedge A$$

$$\vee\text{-i}\ \frac{A}{C}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \vee B \text{ or } C \xleftrightarrow[\mathcal{R}]{*} B \vee A$$

$$\vee\text{-e}\ \frac{C \qquad \begin{array}{c}[A]\\ D\end{array} \qquad \begin{array}{c}[B]\\ D\end{array}}{D}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \vee B$$

$$\forall\text{-i}\ \frac{\{y/x\}A}{B}\ \begin{array}{l}\text{if } B \xleftrightarrow[\mathcal{R}]{*} \forall x.\ A \text{ and } y \text{ is not free in}\\ A \text{ nor in the assumptions of the proof}\\ \text{above}\end{array}$$

$$\forall\text{-e}\ \frac{A}{B}\ \text{if } A \xleftrightarrow[\mathcal{R}]{*} \forall x.\ C \text{ and } B \xleftrightarrow[\mathcal{R}]{*} \{t/x\}C$$

$$\exists\text{-i}\ \frac{B}{A}\ \text{if } A \xleftrightarrow[\mathcal{R}]{*} \exists x.\ C \text{ and } B \xleftrightarrow[\mathcal{R}]{*} \{t/x\}C$$

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

$$\textit{classical}\ \frac{}{B}\ \text{if } A \xleftrightarrow[\mathcal{R}]{*} B \vee (B \Rightarrow \bot)$$

$$\bot\text{-e}\ \frac{A}{B}\ \text{if } A \xleftrightarrow[\mathcal{R}]{*} \bot$$

Fig. 1. Inference Rules of Natural Deduction Modulo.

—$P \xleftrightarrow[\mathcal{R}]{*} Q$ implies $\mathcal{T} \vdash^{\mathsf{N}} P \Leftrightarrow Q$;

—for every formula $P \in \mathcal{T}$, we have $\vdash^{\mathsf{N}}_{\mathcal{R}} P$.

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

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

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

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

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

## 4. 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.

### 4.1 From $Z_i \vdash^{\mathsf{S}}$ to $Z_i \vdash^{\mathsf{N}}$

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 (20) to (35).

For the axiom schemata and inference rules of classical logic, we use the same translation as Gentzen, for instance the axiom schema (4) is translated into the natural deduction proof

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

and the inference rule (19) into

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

(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 formulæ finally substituted in the proof.

In a schematic system proof, there is also a finite number of instances of the axioms 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 (20) to (35), and whose length is linear compared to the schematic system proof:

PROPOSITION 2. *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 $Z_i$.*

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

### 4.2 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 (20) to (35) 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 et al. [1958]). We define mutually recursively two functions by induction on the inference rules: T transforms a proof of $P$ in natural deduction using assumptions $\Gamma$ into a proof of $P$ in the schematic system (1) to (19) 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 (1) to (19) plus $\Gamma$. The translation can be found in the appendix.

It can be verified that this definition transforms a proof of size $n$ into a proof of size $O(3^n)$. 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. *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 $O(3^n)$ in the schematic system for $Z_i$.*

$$Z_i \vdash^{\mathsf{N}}_{k} P \quad \rightsquigarrow \quad Z_i \vdash^{\mathsf{S}}_{O(3^k)} P$$

## 5. HIGHER-ORDER ARITHMETIC AS A FIRST-ORDER THEORY MODULO

### 5.1 Encoding higher order using classes

First, we translate a proof in the schematic system for $Z_i$ into a proof in natural deduction modulo with as assumption a finite theory using only first-order function symbols, using the modulo to get the higher order again.

To do so, we first consider the theory consisting in the *axioms* in (20) to (33), so without the axiom schemata (26), (34) and (35). They are replaced by three new axioms (36), (37) and (38). To do so, we use the work of Kirchner [2006] which permits to express first-order theories using a finite number of axioms. The idea is to transform some metaformula $A(t_1, \ldots, t_n)$ used in an axiom schema into a formula of the form $\langle t_1, \ldots, t_n \rangle \in \gamma$ where $\gamma$ is some term representing what formula is actually substituted for $A$.

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

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

$\langle \alpha_1, \ldots, \alpha_n \rangle$ will be syntactic sugar for $\alpha_1 ::^{j_1} \cdots :: \alpha_n ::^{j_n} nil$ for the appropriate $j_m$. We change the axiom schemata (26), (34) and (35) into the following *axioms*:

$$\forall \gamma^c.\ \forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow \langle \alpha^0 \rangle \in \gamma^c \Rightarrow \langle \beta^0 \rangle \in \gamma^c \tag{36}$$

$$\forall \gamma^c. \langle 0 \rangle \in \gamma^c \Rightarrow \left(\forall \beta^0.\ \langle \beta^0 \rangle \in \gamma^c \Rightarrow \langle s(\beta^0) \rangle \in \gamma^c\right) \Rightarrow \forall \alpha^0.\ \langle \alpha^0 \rangle \in \gamma^c \tag{37}$$

For all $0 \leq j < i$,

$$\forall \gamma^c.\ \exists \alpha^{j+1}.\ \forall \beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow \langle \beta^j \rangle \in \gamma^c \tag{38}$$

We also need axioms which permits to decode the classes (see Kirchner [2006, Definition 4]).

$$\forall \alpha^j.\ \ \alpha^j[nil]^j = \alpha^j \tag{39}$$
$$\forall \alpha^j.\ \forall l^\ell.\ \ 1^j[\alpha^j ::^j l^\ell]^j = \alpha^j \tag{40}$$
$$\forall \alpha^j.\ \forall \beta^k.\ \forall l^\ell.\ \ S^j(\alpha^j)[\beta^k ::^k l^\ell]^j = \alpha^j[l^\ell]^j \tag{41}$$
$$\forall \alpha^0.\ \forall l^\ell.\ \ s(\alpha^0)[l^\ell]^0 = s(\alpha^0[l^\ell]^0) \tag{42}$$
$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ \ (\alpha^0 + \beta^0)[l^\ell]^0 = \alpha^0[l^\ell]^0 + \beta^0[l^\ell]^0 \tag{43}$$
$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ \ (\alpha^0 \times \beta^0)[l^\ell]^0 = \alpha^0[l^\ell]^0 \times \beta^0[l^\ell]^0 \tag{44}$$
$$\forall \alpha^0.\ \forall \beta^0.\ \forall l^\ell.\ \ l^\ell \in \doteq(\alpha^0, \beta^0) \Leftrightarrow \alpha^0[l^\ell]^0 = \beta^0[l^\ell]^0 \tag{45}$$
$$\forall \alpha^j.\ \forall \beta^{j+1}.\ \forall l^\ell.\ \ l^\ell \in \dot{\in}^j(\alpha^j, \beta^{j+1}) \Leftrightarrow \alpha^j[l^\ell]^j \in^j \beta^{j+1}[l^\ell]^{j+1} \tag{46}$$
$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ \ l^\ell \in \alpha^c \cup \beta^c \Leftrightarrow l^\ell \in \alpha^c \vee l^\ell \in \beta^c \tag{47}$$
$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ \ l^\ell \in \alpha^c \cap \beta^c \Leftrightarrow l^\ell \in \alpha^c \wedge l^\ell \in \beta^c \tag{48}$$
$$\forall \alpha^c.\ \forall \beta^c.\ \forall l^\ell.\ \ l^\ell \in \alpha^c \supset \beta^c \Leftrightarrow l^\ell \in \alpha^c \Rightarrow l^\ell \in \beta^c \tag{49}$$
$$\forall l^\ell.\ \ l^\ell \in \emptyset \Leftrightarrow \bot \tag{50}$$
$$\forall \alpha^c.\ \forall l^\ell.\ \ (l^\ell \in \mathcal{P}^j(\alpha^c) \Leftrightarrow \exists \beta^j.\ \beta^j ::^j l^\ell \in \alpha^c) \tag{51}$$
$$\forall \alpha^c.\ \forall l^\ell.\ \ (l^\ell \in \mathcal{C}^j(\alpha^c) \Leftrightarrow \forall \beta^j.\ \beta^j ::^j l^\ell \in \alpha^c) \tag{52}$$

We call $Z_i^{\mathrm{ws}}$ the theory consisting of the axioms (20) to (52) but the axioms schemata (26), (34) and (35).

PROPOSITION 4. *The theory $Z_i^{\mathrm{ws}}$ is a conservative extension of $Z_i$.*

PROOF. This is the Proposition 4 of Kirchner [2006]. □

Now, we use skolemization to transform (38) (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 (38):

$$\forall \gamma^c.\ \forall \beta^j.\ \beta^j \in^j comp^{j+1}(\gamma^c) \Leftrightarrow \langle \beta^j \rangle \in \gamma^c \tag{53}$$

We denote by $Z_i^{\mathrm{sk}}$ the theory $Z_i^{\mathrm{ws}}$ where (38) is replaced by (53).

PROPOSITION 5. *The theory $Z_i^{\mathrm{sk}}$ is a conservative extension of $Z_i^{\mathrm{ws}}$.*

PROOF. According to van Dalen [1989, Corollary 3.4.5], $Z_i^{\mathrm{sk}} \cup \{(38)\}$ is a conservative extension of $Z_i^{\mathrm{ws}}$. But (38) can be proved in $Z_i^{\mathrm{sk}}$ so that we can drop it. □

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 following rewrite system:

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

$$x \in^{j'} comp^{j'+1}(A) \to x ::^{j'} nil \in A$$

for all $0 \leq j \leq i$, $0 \leq k \leq i$ and $0 \leq j' < i$.
This rewrite system has the following properties:

—It is finite (for a given $i$).
—It is locally confluent: the only critical pairs, of the form: $f(t_1,\ldots,t_n) \underset{\mathcal{HO}_i}{\longleftarrow} f(t_1,\ldots,t_n)[nil] \underset{\mathcal{HO}_i}{\longrightarrow} f(t_1[nil],\ldots,t_n[nil])$ 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.
—It is terminating for $i = 1$, that is if we are interested in the gap between first- and second order arithmetic. More generally, it is terminating if we consider the last rule only for $j' = i - 1$. In that case, we do not encode all higher orders into first order, but only $i^{\text{th}}$ order into $i - 1^{\text{st}}$. As there exists a speed-up between $i - 1^{\text{st}}$- and $i^{\text{th}}$-order arithmetic, this is sufficient for our claim that deduction modulo permits to bypass such speed-ups. We nonetheless conjecture that it is terminating also without this restriction.

PROPOSITION 6. *The term rewrite system $\mathcal{HO}_1$ is terminating*

PROOF. See the output of AProVE in appendix B. □

CONJECTURE 1. *For all $i > 1$, the term rewrite system $\mathcal{HO}_i$ is terminating.*

Proposition 2 of Kirchner [2006] says that it is possible, for any formula $P$ of the language of $i^{\text{th}}$-order arithmetic, to prove

$$\exists E.\ \forall x_1 \cdots x_n.\ \langle x_1,\ldots,x_n\rangle \in E \Leftrightarrow P\ .$$

Moreover, the proof of this proposition shows us how to construct the witness $E$. We will denote it by $E_P^{x_1,\ldots,x_n}$. Remark that no $\cdot[\cdot]$ appears in it. Then, one can prove that $\langle t_1,\ldots,t_n\rangle \in E_P^{x_1,\ldots,x_n} \xrightarrow{*} \{t_1/x_1,\ldots,t_n/x_n\}P$. For instance, consider the formula $P \stackrel{!}{=} x = 0 \vee \exists y.\ x \in^0 y$. Then $E_P^x$ equals $\doteq(1,S(0)) \cup \mathcal{P}^1\left(\dot{\in}^0(S(1),1)\right)$ and $\langle t\rangle \in E_P^x$ can be rewritten to $t = 0 \vee \exists x.\ t \in^0 x$.

Consequently, the axiom schemata (26), (34) and (35) are replaced by the proofs in Figure 2. In these translations, we need to instantiate $\gamma$ with some $E_{A(x)}^x$. It is well-known that the instantiations are the most problematic rules in deductive

$$\forall\text{-e}\,\frac{\forall\gamma^c.\ \forall\alpha^0\beta^0.\ \alpha^0 = \beta^0 \Rightarrow \langle\alpha^0\rangle\ \epsilon\ \gamma^c \Rightarrow \langle\beta^0\rangle\ \epsilon\ \gamma^c \ (36)}{\forall\alpha^0\beta^0.\ \alpha^0 = \beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0)}$$

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

$$\forall\text{-e}\,\frac{\forall\gamma^c.\langle 0\rangle\ \epsilon\ \gamma^c \Rightarrow \big(\forall\beta^0.\ \langle\beta^0\rangle\ \epsilon\ \gamma^c \Rightarrow \langle s(\beta^0)\rangle\ \epsilon\ \gamma^c\big) \Rightarrow \forall\alpha^0.\ \langle\alpha^0\rangle\ \epsilon\ \gamma^c \ (37)}{A(0) \Rightarrow \big(\forall\beta^0.\ A(\beta^0) \Rightarrow A(s(\beta^0))\big) \Rightarrow \forall\alpha^0.\ A(\alpha^0)}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\beta^j \in^j comp^{j+1}(E^x_{A(x)})\ \text{(i)}}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Rightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}\ \text{(i)} \quad \dfrac{\dfrac{\beta^j \in^j comp^{j+1}(E^x_{A(x)})\ \text{(ii)}}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Rightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}\ \text{(ii)} \quad \vdots}{\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Leftrightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}\ \wedge\text{-i} \quad \vdots}{\forall\beta^j.\ \beta^j \in^j comp^{j+1}(E^x_{A(x)}) \Leftrightarrow \beta^j \in^j comp^{j+1}(E^x_{A(x)})}\ \forall\text{-i}}{\exists\alpha^{j+1}.\ \forall\beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j)}\ \exists\text{-i} \quad \begin{array}{l}\beta^j \in^j comp^{j+1}(E^x_{A(x)}) \\ \longrightarrow \langle\beta^j\rangle\ \epsilon\ E^x_A \xrightarrow{*} A(\beta^j)\end{array}$$

Fig. 2. Translations of the axiom schemata (26), (34) and (35).

systems, at least for automated provers (e.g. they are what leads to nondeterminism and/or nontermination of tableaux methods for first-order logic), because the instantiated term must be somehow guessed. Nevertheless, the instantiation here is entirely and automatically determined by the formula used in the schema, so that no harm is done. Note that the replacement for (35) do not need extra axioms, because all is done in the modulo.

Let $FZ$ be the theory (20)–(25), (27)–(33), (36)–(37), consisting only of a finite number of axioms, all of them in the language of $Z_0$ plus the language of Kirchner's classes.

Using this, 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 7. *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 $FZ$.*

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

Proof. Instances of axiom schemata in the proof in $Z_i$ are replaced by the proofs in Figure 2, whose length is fixed. □

This result can also be stated entirely in natural deduction

Theorem 8. *For all $i \geq 0$, there exists a (finite) rewrite system $\mathcal{HO}_i$ such that for all formulæ $P$, 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 (26), (34) and (35) by proofs using the axioms (36) and (37) as indicated in Figure 2. □

*Note* 1. As remarked in the introduction, deduction modulo can simulate Tseytin's extensions in propositional logic through an infinite number of rules of the form $A_P \longrightarrow P$. Using Kirchner [2006], this can be done using the following finite (if the signature is), terminating and confluent rewrite system:

$$\begin{gathered}\epsilon(\dot{A}) \to A \quad \text{for } A \text{ atomic}\\ \epsilon(A \cup B) \to \epsilon(A) \vee \epsilon(B)\\ \epsilon(A \cap B) \to \epsilon(A) \wedge \epsilon(B)\\ \epsilon(A \supset B) \to \epsilon(A) \Rightarrow \epsilon(B)\\ \epsilon(\emptyset) \to \bot\end{gathered}$$

with new function symbols $\dot{A}$ for each atomic proposition $A$ of the signature.

### 5.2 Higher-order arithmetic as purely computational theory

We define here higher-order arithmetic entirely as a rewrite system modulo whom 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 formula $P$ 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 formula using induction, we need to know that the variables used in the proof are natural numbers, hence the need for a translation $|\cdot|$ of the formulæ: $\forall x.\ P$ is translated into $\forall x.\ N(x) \Rightarrow |P|$, $\exists x.\ P$ into $\exists x.\ N(x) \wedge |P|$, the translation of the other logical connectors being just the connection of the translations of the subformulæ. 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 in the given translation is not conserved. 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

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

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

Here we use a different approach: the induction axiom will be directly translated into the following rewrite rule

$$x \in p \to x \in p \vee (0 \in p \wedge \forall y.\ y \in p \Rightarrow s(y) \in p) \quad .$$

This rule permits to express the induction schema as the intuitionistically equivalent schema $\forall x.\ P(x) \Leftrightarrow P(x) \vee (P(0) \wedge \forall y.\ P(y) \Rightarrow P(s(y)))$, combined with the rule for $f_P$. Note that doing so, we lose the confluence of our system.

Kirchner [2006] already applied his method to first-order arithmetic, to get a finite rewrite system, contrarily to [Dowek and Werner 2005]. The preceding rule that we used for the induction can be applied only on first-order terms, and therefore becomes

$$x ::^0 nil \in p \to x ::^0 nil \in p \vee (0 ::^0 nil \in p \wedge \forall y.\ y ::^0 nil \in p \Rightarrow s(y) ::^0 nil \in p) \quad .$$

Here we extend this method to all orders, as done in the previous section. If we do not use (16) as axiom (i.e. if we work in intuitionistic logic), we therefore obtain a formulation of higher-order Heyting arithmetic through the following rewrite system $\mathcal{HHA}_i^{\mathrm{mod}}$:

Arithmetic rules:

$$\begin{array}{rcl@{\qquad}rcl}
pred(0) & \to & 0 & 0 \times y & \to & y \\
pred(s(x)) & \to & x & s(x) \times y & \to & x \times y + y \\
0 + y & \to & y & Null(0) & \to & \top \\
s(x) + y & \to & s(x + y) & Null(s(x)) & \to & \bot
\end{array}$$

Axiom schemata:

$$\begin{array}{rcl@{\qquad}rcl}
x = y & \to & \forall z^c.\ \langle x \rangle \in z \Rightarrow \langle y \rangle \in z & x \in^j comp^{j+1}(y) & \to & x ::^j nil \in y \\
x ::^0 nil \in p & \to & \langle x \rangle \in p \vee (\langle 0 \rangle \in p \wedge \forall y.\ \langle y \rangle \in p \Rightarrow \langle s(y) \rangle \in p) & & &
\end{array}$$

Substitutions and classes: we use $\mathcal{HO}_i$, but as the signature is bigger, we also need the following rules:

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

With this rewrite system, we can linearly simulate higher-order arithmetic in deduction modulo:

THEOREM 9. *For all $i$ there exists a finite rewrite system $\mathcal{HHA}_i^{mod}$ such that for all formula $P$ 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.

(20) can be proved by

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

(21) to (28) are proved using $x = y \to \forall z^c.\ \langle x \rangle \in z \Rightarrow \langle y \rangle \in z$ in at most 8 steps. We give here the proof for (24), with $E_\times \quad \stackrel{!}{=} \quad \doteq (S(\alpha^0) \times S(\gamma^0), 1 \times S(\gamma^0))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle \alpha^0 \times \gamma^0\rangle \in p^c \text{ (i)}}{\langle \alpha^0 \times \gamma^0\rangle \in p^c \Rightarrow \langle \alpha^0 \times \gamma^0\rangle \in p^c}\Rightarrow\text{-i}^{\text{(i)}}}{\alpha^0 \times \gamma^0 = \alpha^0 \times \gamma^0}\forall\text{-i} \qquad \dfrac{\alpha^0 = \beta^0 \text{ (i)}}{\langle \alpha^0\rangle \in E_\times \Rightarrow \langle \beta^0\rangle \in E_\times}\forall\text{-e}}{\alpha^0 \times \gamma^0 = \beta^0 \times \gamma^0}\Rightarrow\text{-e}}{\alpha^0 = \beta^0 \Rightarrow \alpha^0 \times \gamma^0 = \beta^0 \times \gamma^0}\Rightarrow\text{-i}^{\text{(i)}}}{\forall \alpha^0\ \beta^0\ \gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \alpha^0 \times \gamma^0 = \beta^0 \times \gamma^0}\forall\text{-i}^{\times 3}$$

Let $E \stackrel{!}{=} (\doteq (1, S(0)) \supset \emptyset) \supset \mathcal{P}(\doteq (S(1), s(1)))$. (29) is proved by

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

(30) to (33) are easy to prove using the arithmetical rules and the rule for $=$.
(34) has the following proof:

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

(35) has the following proof:

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

□

What we obtain is a conservative extension:

THEOREM 10. *For all formula $P$ in the language of $Z_i$, if $\vdash^{\mathsf{N}}_{\mathcal{HHA}_i^{mod}} P$ then $Z_i \vdash^{\mathsf{N}} P$.*

PROOF. For first-order arithmetic, the only difference between $HA_N$ of Dowek and Werner [2005] and our system is the induction schema. If we translate $N(n)$ into $\forall p.\ 0 \in p \Rightarrow (\forall y.\ y \in p \Rightarrow s(y) \in p) \Rightarrow n \in p$, because we use an intuitionistically equivalent formulation of the induction schema, we can proof that we have a conservative extension of $HA_N$ (of its variant in fact, see Dowek and Werner [2005, Remark 2]).

Then we apply the method of [Kirchner 2006], 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]). □

*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 2006]. This remains currently an open question whether our system normalizes or not.

## 6. APPLICATIONS TO PROOF-LENGTH SPEED-UPS

Because of Theorem 8 and Theorem 9, there is obviously no proof-length speed-up between $Z_i$, $FZ$ modulo $\mathcal{HO}_i$ and $\emptyset$ modulo $\mathcal{HHA}_i^{\mathrm{mod}}$. Furthermore, there exists a speed-up between all these and $Z_{i-1}$, which can be decomposed as follow.

### 6.1 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 $\mathcal{HO}_i$. But first, we prove that there is no need for a complicated rewrite system to get such a speed-up.

PROPOSITION 11. *Consider the rewrite system $\mathcal{R}$ consisting only of the rule $s(x) + y \to x + s(y)$, there is an infinite family $\mathcal{F}$ such that such that for all finite theories $\mathcal{T}$ compatible with $\mathcal{R}$,*

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

*(2) there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $\vdash^{\mathsf{N}}_{k\ steps\ \mathcal{R}} P$*

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

PROOF. Consider the family of formulæ $P(\underline{n} + \underline{n}) \Rightarrow P(\underline{n+n})$ for some unary predicate $P$, where $\underline{n}$ denotes the usual representation of the natural number $n$ using 0 and $s$. Then it is quite clear that $\vdash^{\mathsf{N}}_{1\ \mathcal{R}} P(\underline{n} + \underline{n}) \Rightarrow P(\underline{n+n})$. Let $\mathcal{T}$ be a finite theory compatible with $\mathcal{R}$. By definition $\mathcal{T} \vdash^{\mathsf{N}} P(\underline{n} + \underline{n}) \Rightarrow P(\underline{n+n})$, but it is impossible to find a proof that takes less than $O(n)$ steps. (In the theory we may have some formulæ such as $s^m(x) + y = x + s^m(y)$ but they will only divides the minimal number of steps by $m$, and we can only have a finite number of such formulæ. The theorem is of course wrong if infinite theories are allowed, because one could add $\mathcal{F}$ to some theory compatible with $\mathcal{R}$ to get proofs with a bounded number of steps.) □

This is therefore no surprise 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. (The same kind of proof permits also to show the existence of a speed-up between $\mathcal{HHA}_i^{\mathrm{mod}}$ and $FZ$ plus a finite theory compatible with $\mathcal{HO}_i$.)

PROPOSITION 12. *For all $i$, there is an infinite family $\mathcal{F}$ such that such that for all finite theories $\mathcal{T}_i$ compatible with $\mathcal{HO}_i$,*

*(1) for all $P \in \mathcal{F}$, we have $FZ, \mathcal{T}_i \vdash^{\mathsf{N}} P$*

*(2) there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $FZ \vdash^{\mathsf{N}}_{k \text{ steps}}{}_{\mathcal{HO}_i} P$*

*(3) there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, we have $FZ, \mathcal{T}_i \vdash^{\mathsf{N}}_{k \text{ steps}} P$*

PROOF. Consider the set of formulæ corresponding to all instantiations of the comprehension schema for $j = i - 1$. Obviously, $Z_{i-1}$ is not enough to prove all of them, so that (38) has to be used in the proofs in $FZ, \mathcal{T}_i$. Nevertheless, the term of sort $c$ instantiated in it cannot have a bounded size. Then, the decomposition of this term using the finite theory $\mathcal{T}_i$ compatible with $\mathcal{HO}_i$ cannot be done in a bounded number of steps. In $FZ$ modulo $\mathcal{HO}_i$ this formulæ can be proved in one step as done in Fig. 2. □

### 6.2 Speed-up in arithmetic modulo

It is also possible to get a speed-up between $FZ$ plus any theory compatible with $\mathcal{HO}_i$ and $Z_{i-1}$.

PROPOSITION 13. *For all $i > 0$, there is an infinite family $\mathcal{F}$ such that for all theory $\mathcal{T}_i$ 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 $FZ, \mathcal{T}_i \vdash^{\mathsf{N}}_{k \text{ steps}} P$*

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

PROOF. If we look at Buss' proof of Theorem 1, the infinite family of formulæ 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 $\mathcal{T}_i$, which is the case because of Theorem 8 and [Dowek et al. 2003, Proposition 1.8]. We also need Proposition 3 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. □

## 7. CONCLUSION AND PERSPECTIVES

The contribution of this paper are summarized in Figure 3. We have shown how to encode higher-order arithmetic through the rewrite system $\mathcal{HHA}_i^{\text{mod}}$ without inflating the length of proofs. In particular, using a generalization of the work of Kirchner [2006], we have translated higher-order axiom schemata into many-sorted first-order axioms modulo $\mathcal{HO}_i$. Using this, we have proved that the speed-up in arithmetic already occurs using a first-order finite axiomatization of $Z_{i+1}$ (Proposition 13), but that it is better to express this theory as a rewrite system modulo which rules will be applied (Proposition 12). This shows the power of separating computation and deduction. This kind of speed-ups must 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

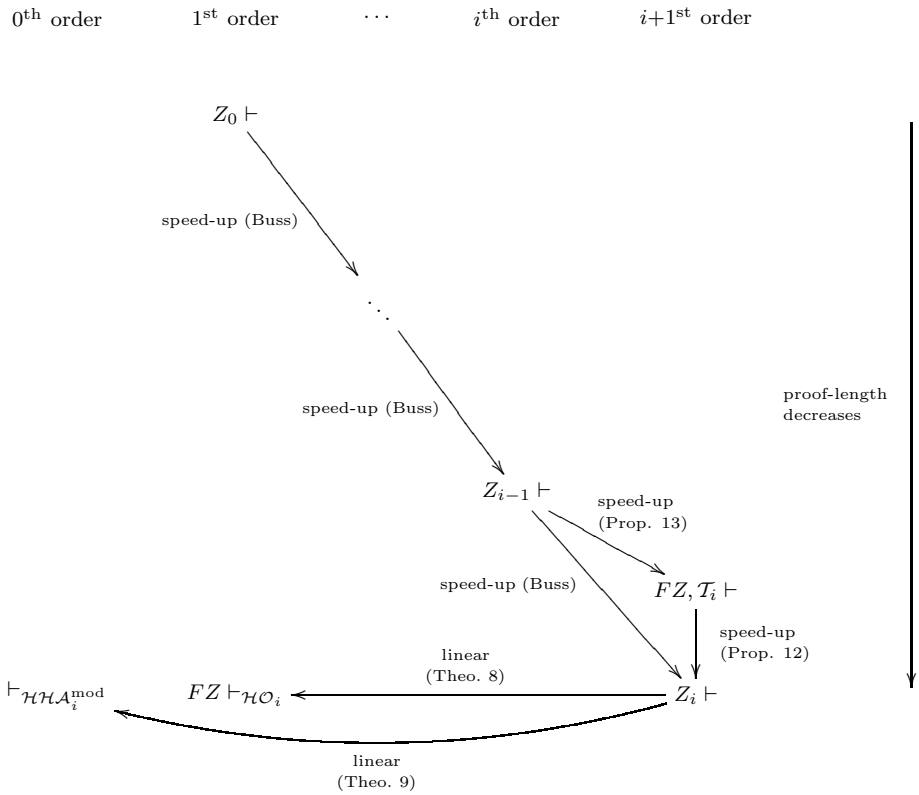



Fig. 3. Speed-ups in higher-order arithmetic and deduction modulo

(hence the need to have a decidable congruence, even better if it is determined by simple deterministic algorithm). This can be applied to automated and interactive theorem proving, to proof-carrying code [Necula 1997], as well as in representation of proofs in natural languages (where all computational details are often implicitly left the reader).

The fact that the difference between first-order and higher-order arithmetic can be expressed as computation is not surprising, because, if one looks carefully, the proof of Theorem 1 given by Buss [1994] deeply relies on the fact that it is possible to define some truth predicate for the formulæ of the preceding order. Therefore, in a sense, it is possible, in $i + 1^{\mathrm{st}}$-order arithmetic, to compute the validity of a formula in $i^{\mathrm{th}}$-order arithmetic. It should be looked at if this truth predicate could be expressed as a rewrite system, therefore relating more strongly Buss' proof and the speed-up obtained using deduction modulo.

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 [2006] or Isabelle/HOL [Nipkow et al. 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 et al. [2007]). It must

also be proved that such results extends to pure higher-order logic. Such a result was achieved by proving that function 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.

We also have to look whether it could be possible to find a rewrite system with the same proof length as $Z_i$, at least for formulæ of the language of $Z_{i-1}$, but such that compatible theories do not have a speed-up over $Z_{i-1}$. Were it possible, it would prove that we can find a speed-up between $Z_i$ and $Z_{i-1}$ that cannot be decomposed as we shown, but lies entirely in the computational part of the proofs.

## APPENDIX

## A. TRANSLATION FROM $Z_I \vdash^{\mathsf{N}}$ TO $Z_I \vdash^{\mathsf{S}}$

$$\mathrm{T}\left(\Rightarrow\text{-i}\,\dfrac{\pi\{\begin{smallmatrix}[A]\\B\end{smallmatrix}}{A\Rightarrow B}\right) \stackrel{!}{=} \mathrm{T}_A\left(\pi\{\begin{smallmatrix}[A]\\B\end{smallmatrix}\right)$$

$$\mathrm{T}\left(\Rightarrow\text{-e}\,\dfrac{\pi_1\quad\pi_2}{B}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi_1)\quad\mathrm{T}(\pi_2)}{B}$$

$$\mathrm{T}\left(\wedge\text{-i}\,\dfrac{\pi_1\quad\pi_2}{A\wedge B}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi_1)\quad (17)\,\dfrac{\mathrm{T}(\pi_2)\quad (17)\,\dfrac{B\quad B\Rightarrow A\Rightarrow B\ (2)}{A\Rightarrow B}}{A\Rightarrow(A\wedge B)}}{A\wedge B}\quad (17)\,\dfrac{A\Rightarrow A\ (1)\quad\cdots\ (8)}{(A\Rightarrow B)\Rightarrow A\Rightarrow(A\wedge B)}$$

$$\mathrm{T}\left(\wedge\text{-e}\,\dfrac{\pi}{A}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi)\quad A\wedge B\quad A\wedge B\Rightarrow A\ (6)}{A}$$

and similarly with (7) for the other side.

$$\mathrm{T}\left(\vee\text{-i}\,\dfrac{\pi}{A\vee B}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi)\quad A\quad A\Rightarrow(A\vee B)\ (9)}{A\vee B}$$

and similarly with (10) for the other side.

$$\mathrm{T}\left(\vee\text{-e}\,\dfrac{\pi_1\quad A\vee B\quad \pi_2\{\begin{smallmatrix}[A]\\C\end{smallmatrix}\quad \pi_3\{\begin{smallmatrix}[B]\\C\end{smallmatrix}}{C}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi_1)\quad A\vee B\quad (17)\,\dfrac{\mathrm{T}_B(\pi_3)\quad B\Rightarrow C\quad (17)\,\dfrac{\mathrm{T}_A(\pi_2)\quad A\Rightarrow C\quad\cdots\ (11)}{(B\Rightarrow C)\Rightarrow(A\vee B)\Rightarrow C}}{(A\vee B)\Rightarrow C}}{C}$$

$$\mathrm{T}\left(\forall\text{-i}\,\dfrac{\pi}{\forall x.\ A}\right) \stackrel{!}{=} (17)\,\dfrac{\mathrm{T}(\pi)\quad \{y/x\}A\quad (18)\,\dfrac{\{y/x\}A\Rightarrow\{y/x\}A\ (1)}{\{y/x\}A\Rightarrow\forall x.\ A}}{\forall x.\ A}$$

Note that the side conditions are satisfied.

$$\mathrm{T}\left(\forall\text{-e}\,\dfrac{\begin{array}{c}\pi\\ \forall x.\ A\end{array}}{\{t/x\}A}\right) \quad \stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}\,(\pi) & \\ \forall x.\ A & \forall x.\ A \Rightarrow \{t/x\}A\ (14)\end{array}}{\{t/x\}A}$$

$$\mathrm{T}\left(\exists\text{-i}\,\dfrac{\begin{array}{c}\pi\\ \{t/x\}A\end{array}}{\exists x.\ A}\right) \quad \stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}\,(\pi) & \\ \{t/x\}A & \{t/x\}A \Rightarrow \exists x.\ A\ (15)\end{array}}{\exists x.\ A}$$

$$\mathrm{T}\left(\exists\text{-e}\,\dfrac{\begin{array}{ccc}\pi_1 & & [\{y/x\}A]\\ \exists x.\ A & \pi_2\{ & B\end{array}}{B}\right) \quad \stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}\,(\pi_1) & \mathrm{T}_A\,(\pi_2)\\ \exists x.\ A & (19)\,\dfrac{\{y/x\}A \Rightarrow B}{(\exists x.\ A) \Rightarrow B}\end{array}}{B}$$

Note that the side conditions are satisfied.

$$\mathrm{T}\left(\begin{array}{c}\mathit{classical}\ \dfrac{}{A \vee (A \Rightarrow \bot)}\end{array}\right) \quad \stackrel{!}{=} \quad A \vee (A \Rightarrow \bot)\ (16)$$

$$\mathrm{T}\left(\perp\text{-e}\,\dfrac{\begin{array}{c}\pi\\ \bot\end{array}}{A}\right) \quad \stackrel{!}{=}$$

$$(17)\,\dfrac{\begin{array}{cc}\mathrm{T}\,(\pi) & \\ \bot & \bot \Rightarrow (A \Rightarrow A) \Rightarrow \bot\ (2)\end{array}}{\begin{array}{cc}(A \Rightarrow A) \Rightarrow \bot & \cdots\ (13)\\ (17)\,\dfrac{}{(A \Rightarrow A) \Rightarrow A} & \end{array}}$$
$$(17)\,\dfrac{A \Rightarrow A\ (1)}{A}$$

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

$$\mathrm{T}_A\left(\Rightarrow\text{-i}\,\dfrac{\begin{array}{c}[B]\\ \pi\{\\ C\end{array}}{B \Rightarrow C}\right) \quad \stackrel{!}{=} \quad \mathrm{T}_A\left(\begin{array}{c}\mathrm{T}_B\,(\pi)\\ B \Rightarrow C\end{array}\right)$$

$$\mathrm{T}_A\left(\Rightarrow\text{-e}\,\dfrac{\begin{array}{ccc}\pi_1\{ & [A] & \pi_2\{ \quad [A]\\ & B & \quad B \Rightarrow C\end{array}}{C}\right)$$

$$\stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}_A\,(\pi_2) & \\ A \Rightarrow B \Rightarrow C & \cdots\ (4)\end{array}}{\begin{array}{cc}B \Rightarrow A \Rightarrow C & \\ (17)\,\dfrac{}{(A \Rightarrow A \Rightarrow C) \Rightarrow A \Rightarrow A \Rightarrow C} & \end{array}} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}_A\,(\pi_1) & \\ A \Rightarrow B & \cdots\ (5)\end{array}}{\begin{array}{cc}(A \Rightarrow A \Rightarrow C) \Rightarrow A \Rightarrow A \Rightarrow C & \\ (17)\,\dfrac{}{A \Rightarrow A \Rightarrow C} & \cdots\ (3)\end{array}}$$
$$A \Rightarrow C$$

$$\mathrm{T}_A\left(\wedge\text{-i}\,\dfrac{\begin{array}{ccc}\pi_1\{ & [A] & \pi_2\{ \quad [A]\\ & B & \quad C\end{array}}{B \wedge C}\right)$$

$$\stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}_A\,(\pi_2) & \\ A \Rightarrow C & \end{array}}{\begin{array}{cc}(A \Rightarrow C) \Rightarrow A \Rightarrow (B \wedge C) & \\ (17)\,\dfrac{}{A \Rightarrow (B \wedge C)} & \end{array}} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}_A\,(\pi_1) & \\ A \Rightarrow B & (A \Rightarrow B) \Rightarrow (A \Rightarrow C) \Rightarrow A \Rightarrow (B \wedge C)\ (8)\end{array}}{\begin{array}{cc}(A \Rightarrow C) \Rightarrow A \Rightarrow (B \wedge C) & \\ (17)\,\dfrac{}{A \Rightarrow (B \wedge C)} & \end{array}}$$

$$\mathrm{T}_A\left(\wedge\text{-e}\,\dfrac{\begin{array}{c}[A]\\ \pi\{\\ B \wedge C\end{array}}{B}\right) \quad \stackrel{!}{=} \quad (17)\,\dfrac{\begin{array}{cc}\mathrm{T}_A\,(\pi) & \\ A \Rightarrow (B \wedge C) & \cdots\ (5)\end{array}}{\begin{array}{cc}((B \wedge C) \Rightarrow B) \Rightarrow A \Rightarrow B & \\ (17)\,\dfrac{}{A \Rightarrow B} & \end{array}}\ (17)\,\dfrac{(B \wedge C) \Rightarrow B\ (6)}{}$$

and similarly with (7) for the other side.

$$\mathrm{T}_A\left(\begin{array}{c}\pi\{\begin{array}{c}[A]\\B\end{array}\\ \vee\text{-i}\ \dfrac{}{B\vee C}\end{array}\right) \stackrel{!}{=} (17)\ \dfrac{B\Rightarrow(B\vee C)\ (9) \qquad (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi)\\A\Rightarrow B\end{array}\qquad\cdots\ (5)}{(B\Rightarrow(B\vee C))\Rightarrow A\Rightarrow(B\vee C)}}{A\Rightarrow(B\vee C)}$$

and similarly with (10) for the other side.

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

$$(17)\ \dfrac{\mathrm{T}_C\left(\begin{array}{c}\mathrm{T}_A(\pi_3)\\A\Rightarrow D\end{array}\right) \quad (17)\ \dfrac{\mathrm{T}_B\left(\begin{array}{c}\mathrm{T}_A(\pi_2)\\A\Rightarrow D\end{array}\right) \quad (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi_1)\\A\Rightarrow(B\vee C)\end{array}\qquad\cdots\ (5)}{((B\vee C)\Rightarrow A\Rightarrow D)\Rightarrow A\Rightarrow A\Rightarrow D}}{(17)\ \dfrac{(B\vee C)\Rightarrow A\Rightarrow D}{A\Rightarrow A\Rightarrow D}}}{(17)\ \dfrac{\vdots}{A\Rightarrow D}\qquad\cdots\ (3)}$$

$$\mathrm{T}_A\left(\begin{array}{c}\pi\{\begin{array}{c}[A]\\\{y/x\}B\end{array}\\ \forall\text{-i}\ \dfrac{}{\forall x.\ B}\end{array}\right) \stackrel{!}{=} (18)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi)\\A\Rightarrow\{y/x\}B\end{array}}{A\Rightarrow\forall x.\ B}$$

Note that the side conditions are satisfied.

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

$$\stackrel{!}{=} (17)\ \dfrac{(\forall x.\ B)\Rightarrow\{t/x\}B\ (14) \qquad (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi)\\A\Rightarrow\forall x.\ B\end{array}\qquad\cdots\ (5)}{((\forall x.\ B)\Rightarrow\{t/x\}B)\Rightarrow A\Rightarrow\{t/x\}B}}{A\Rightarrow\{t/x\}B}$$

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

$$\stackrel{!}{=} (17)\ \dfrac{\{t/x\}B\Rightarrow\exists x.\ B\ (15) \qquad (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi)\\A\Rightarrow\{t/x\}B\end{array}\qquad\cdots\ (5)}{(\{t/x\}B\Rightarrow\exists x.\ B)\Rightarrow A\Rightarrow\exists x.\ B}}{A\Rightarrow\exists x.\ B}$$

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

$$\stackrel{!}{=} (19)\ \dfrac{\{y/x\}B\Rightarrow A\Rightarrow C}{\exists x.\ B\Rightarrow A\Rightarrow C} \quad \mathrm{T}_B\left(\begin{array}{c}\mathrm{T}_A(\pi_2)\\A\Rightarrow C\end{array}\right) \quad (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi_1)\\A\Rightarrow\exists x.\ B\end{array}\qquad\ldots\ (5)}{(\exists x.\ B\Rightarrow A\Rightarrow C)\Rightarrow A\Rightarrow A\Rightarrow C}$$

$$(17)\ \dfrac{A\Rightarrow A\Rightarrow C}{A\Rightarrow C}\qquad\cdots\ (3)$$

Note that the side conditions are satisfied.

$$\mathrm{T}_A\left(\begin{array}{c}\pi\{\begin{array}{c}[A]\\\bot\end{array}\\ \bot\text{-e}\ \dfrac{}{B}\end{array}\right) \stackrel{!}{=} (17)\ \dfrac{\begin{array}{c}\mathrm{T}_A(\pi)\\A\Rightarrow\bot\end{array}\qquad(A\Rightarrow\bot)\Rightarrow A\Rightarrow B\ (13)}{A\Rightarrow B}$$

$$\mathrm{T}_A\,(A) \quad \stackrel{!}{=} \quad A \Rightarrow A\ (1)$$

$$\mathrm{T}_A\begin{pmatrix} \pi \\ B \end{pmatrix} \quad \stackrel{!}{=} \quad \dfrac{\begin{matrix}\mathrm{T}\,(\pi) \\ B\end{matrix} \qquad B \Rightarrow A \Rightarrow B\ (2)}{A \Rightarrow B}\,(17)$$

if the assumption $A$ is not actually used in $\pi$.

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 (18) and (19). (The translation of (17) is already defined because (17) is equal to $\Rightarrow$-e.)

$$\mathrm{T}_A\left(\begin{array}{c}\pi\{\ \begin{matrix}[A] \\ B \Rightarrow C(\tau)\end{matrix} \\ \hline B \Rightarrow \forall\alpha.\ C(\alpha)\end{array}\,(18)\right)$$

$$\stackrel{!}{=} \quad \dfrac{\dfrac{\begin{matrix}\mathrm{T}_A\,(\pi) \\ A \Rightarrow B \Rightarrow C(\tau)\end{matrix} \qquad \begin{matrix}\varpi_1 \\ (A \Rightarrow B \Rightarrow C(\tau)) \Rightarrow (A \wedge B) \Rightarrow C(\tau)\end{matrix}}{(A \wedge B) \Rightarrow C(\tau)}\,(17) \qquad \begin{matrix}\varpi_2 \\ \cdots\end{matrix}}{(A \wedge B) \Rightarrow \forall\alpha.\ C(\alpha)}\,(18)}{A \Rightarrow B \Rightarrow \forall\alpha.\ C(\alpha)}\,(17)$$

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 (1) to (8) and the inference rule (17). (Indeed, they are valid formulæ of the intuitionistic propositional logic.)

$$\mathrm{T}_A\left(\begin{array}{c}\pi\{\ \begin{matrix}[A] \\ B(\tau) \Rightarrow C\end{matrix} \\ \hline (\exists\alpha.\ B(\alpha)) \Rightarrow C\end{array}\,(19)\right)$$

$$\stackrel{!}{=} \quad \dfrac{\dfrac{\begin{matrix}\mathrm{T}_A\,(\pi) \\ A \Rightarrow B(\tau) \Rightarrow C\end{matrix} \qquad \begin{matrix}(A \Rightarrow B(\tau) \Rightarrow C) \Rightarrow B(\tau) \Rightarrow A \Rightarrow C\ (4)\end{matrix}}{B(\tau) \Rightarrow A \Rightarrow C}\,(19) \qquad \cdots\ (4)}{\exists\alpha.\ B(\alpha) \Rightarrow A \Rightarrow C}\,(17)}{A \Rightarrow \exists\alpha.\ B(\alpha) \Rightarrow C}$$

## B. TERMINATION PROOF WITH APROVE

Termination of $\mathcal{HO}_1$ could successfully be *proven* by AProVE (`http://www-i2.informatik.rwth-aachen.de/AProVE`):

**Term Rewriting System $\mathcal{HO}_1$ :**

$$\begin{aligned}
\mathsf{subst0}(t, \mathsf{nil}) &\to t \\
\mathsf{subst0}(\mathsf{un0}, \mathsf{cons0}(t, l)) &\to t \\
\mathsf{subst0}(\mathsf{S0}(n), \mathsf{cons0}(t, l)) &\to \mathsf{subst0}(n, l) \\
\mathsf{subst0}(\mathsf{S0}(n), \mathsf{cons1}(t, l)) &\to \mathsf{subst0}(n, l) \\
\mathsf{subst0}(\mathsf{s}(n), l) &\to \mathsf{s}(\mathsf{subst0}(n, l)) \\
\mathsf{subst0}(\mathsf{plus}(t1, t2), l) &\to \mathsf{plus}(\mathsf{subst0}(t1, l), \mathsf{subst0}(t2, l)) \\
\mathsf{subst0}(\mathsf{mult}(t1, t2), l) &\to \mathsf{mult}(\mathsf{subst0}(t1, l), \mathsf{subst0}(t2, l)) \\
\mathsf{subst1}(t, \mathsf{nil}) &\to t \\
\mathsf{subst1}(\mathsf{un1}, \mathsf{cons1}(t, l)) &\to t \\
\mathsf{subst1}(\mathsf{S1}(n), \mathsf{cons0}(t, l)) &\to \mathsf{subst1}(n, l) \\
\mathsf{subst1}(\mathsf{S1}(n), \mathsf{cons1}(t, l)) &\to \mathsf{subst1}(n, l) \\
\mathsf{inc}(l, \mathsf{doteq}(t1, t2)) &\to \mathsf{eq}(\mathsf{subst0}(t1, l), \mathsf{subst0}(t2, l)) \\
\mathsf{inc}(l, \mathsf{dotin0}(t1, t2)) &\to \mathsf{in0}(\mathsf{subst0}(t1, l), \mathsf{subst1}(t2, l))
\end{aligned}$$

$$\begin{array}{rcl}
\mathsf{inc}(l, \mathsf{supset}(A, B)) & \to & \mathsf{imp}(\mathsf{inc}(l, A), \mathsf{inc}(l, B)) \\
\mathsf{inc}(l, \mathsf{cup}(A, B)) & \to & \mathsf{or}(\mathsf{inc}(l, A), \mathsf{inc}(l, B)) \\
\mathsf{inc}(l, \mathsf{cap}(A, B)) & \to & \mathsf{and}(\mathsf{inc}(l, A), \mathsf{inc}(l, B)) \\
\mathsf{inc}(l, \mathsf{empty}) & \to & \mathsf{bot} \\
\mathsf{inc}(l, \mathsf{P0}(A)) & \to & \mathsf{ex}(\mathsf{inc}(\mathsf{cons0}(\mathsf{fresh}, l), A)) \\
\mathsf{inc}(l, \mathsf{P1}(A)) & \to & \mathsf{ex}(\mathsf{inc}(\mathsf{cons1}(\mathsf{fresh}, l), A)) \\
\mathsf{inc}(l, \mathsf{C0}(A)) & \to & \mathsf{all}(\mathsf{inc}(\mathsf{cons0}(\mathsf{fresh}, l), A)) \\
\mathsf{inc}(l, \mathsf{C1}(A)) & \to & \mathsf{all}(\mathsf{inc}(\mathsf{cons1}(\mathsf{fresh}, l), A)) \\
\mathsf{in0}(x, \mathsf{comp1}(A)) & \to & \mathsf{inc}(\mathsf{cons0}(x, \mathsf{nil}), A)
\end{array}$$

**Termination** of $\mathcal{HO}_1$ to be shown.

R
↪**Dependency Pair Analysis**

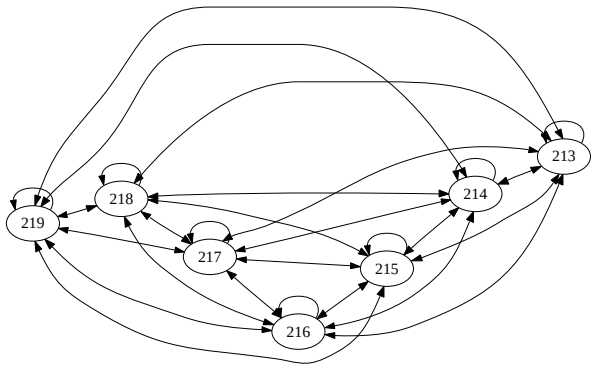
$\mathcal{HO}_1$ contains the following *Dependency Pairs:*

$$\begin{array}{rcl}
\mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons0}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons1}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{s}(n), l) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l) \\
\mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l) \\
\mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l) \\
\mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l) \\
\mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons0}(t, l)) & \to & \mathsf{SUBST1}(n, l) \\
\mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons1}(t, l)) & \to & \mathsf{SUBST1}(n, l) \\
\mathsf{INC}(l, \mathsf{doteq}(t1, t2)) & \to & \mathsf{SUBST0}(t1, l) \\
\mathsf{INC}(l, \mathsf{doteq}(t1, t2)) & \to & \mathsf{SUBST0}(t2, l) \\
\mathsf{INC}(l, \mathsf{dotin0}(t1, t2)) & \to & \mathsf{IN0}(\mathsf{subst0}(t1, l), \mathsf{subst1}(t2, l)) \\
\mathsf{INC}(l, \mathsf{dotin0}(t1, t2)) & \to & \mathsf{SUBST0}(t1, l) \\
\mathsf{INC}(l, \mathsf{dotin0}(t1, t2)) & \to & \mathsf{SUBST1}(t2, l) \\
\mathsf{INC}(l, \mathsf{supset}(A, B)) & \to & \mathsf{INC}(l, A) \\
\mathsf{INC}(l, \mathsf{supset}(A, B)) & \to & \mathsf{INC}(l, B) \\
\mathsf{INC}(l, \mathsf{cup}(A, B)) & \to & \mathsf{INC}(l, A) \\
\mathsf{INC}(l, \mathsf{cup}(A, B)) & \to & \mathsf{INC}(l, B) \\
\mathsf{INC}(l, \mathsf{cap}(A, B)) & \to & \mathsf{INC}(l, A) \\
\mathsf{INC}(l, \mathsf{cap}(A, B)) & \to & \mathsf{INC}(l, B) \\
\mathsf{INC}(l, \mathsf{P0}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A) \\
\mathsf{INC}(l, \mathsf{P1}(A)) & \to & \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A) \\
\mathsf{INC}(l, \mathsf{C0}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A) \\
\mathsf{INC}(l, \mathsf{C1}(A)) & \to & \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A) \\
\mathsf{IN0}(x, \mathsf{comp1}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(x, \mathsf{nil}), A)
\end{array}$$

Furthermore, $\mathcal{HO}_1$ contains three SCCs.

R

↪DPs
→DP Problem 1
↪**Polynomial Ordering**
→DP Problem 2
↪Polo
→DP Problem 3
↪Polo




The node numbers are specified as follows:

$$
\begin{array}{lrcl}
213: & \mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons0}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
214: & \mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons1}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
215: & \mathsf{SUBST0}(\mathsf{s}(n), l) & \to & \mathsf{SUBST0}(n, l) \\
216: & \mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l) \\
217: & \mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l) \\
218: & \mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l) \\
219: & \mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l)
\end{array}
$$

The following dependency pairs can be strictly oriented:

$$
\begin{array}{rcl}
\mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons1}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{S0}(n), \mathsf{cons0}(t, l)) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l) \\
\mathsf{SUBST0}(\mathsf{s}(n), l) & \to & \mathsf{SUBST0}(n, l) \\
\mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l) \\
\mathsf{SUBST0}(\mathsf{mult}(t1, t2), l) & \to & \mathsf{SUBST0}(t2, l) \\
\mathsf{SUBST0}(\mathsf{plus}(t1, t2), l) & \to & \mathsf{SUBST0}(t1, l)
\end{array}
$$

There are no usable rules w.r.t. to the implicit AFS that need to be oriented.

Used ordering: Polynomial ordering with Polynomial interpretation:

$$\begin{aligned} POL(\mathsf{mult}(x_1, x_2)) &= 1 + x_1 + x_2 \\ POL(\mathsf{plus}(x_1, x_2)) &= 1 + x_1 + x_2 \\ POL(\mathsf{cons1}(x_1, x_2)) &= 0 \\ POL(\mathsf{SUBST0}(x_1, x_2)) &= 1 + x_1 \\ POL(\mathsf{s}(x_1)) &= 1 + x_1 \\ POL(\mathsf{cons0}(x_1, x_2)) &= 0 \\ POL(\mathsf{S0}(x_1)) &= 1 + x_1 \end{aligned}$$

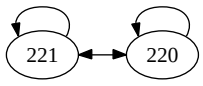
resulting in one new DP problem.

R
↪DPs
→DP Problem 1
↪Polo
→DP Problem 4
↪**Dependency Graph**
→DP Problem 2
↪Polo
→DP Problem 3
↪Polo

The node numbers are specified as follows:

Using the Dependency Graph resulted in no new DP problems.

R
↪DPs
→DP Problem 1
↪Polo
→DP Problem 2
↪**Polynomial Ordering**
→DP Problem 3
↪Polo



The node numbers are specified as follows:

$$\begin{array}{lrcl} 220: & \mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons0}(t,l)) & \to & \mathsf{SUBST1}(n,l) \\ 221: & \mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons1}(t,l)) & \to & \mathsf{SUBST1}(n,l) \end{array}$$

The following dependency pairs can be strictly oriented:

$$\begin{array}{lcl} \mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons1}(t,l)) & \to & \mathsf{SUBST1}(n,l) \\ \mathsf{SUBST1}(\mathsf{S1}(n), \mathsf{cons0}(t,l)) & \to & \mathsf{SUBST1}(n,l) \end{array}$$

There are no usable rules w.r.t. to the implicit AFS that need to be oriented. Used ordering: Polynomial ordering with Polynomial interpretation:

$$\begin{aligned} POL(\mathsf{cons1}(x_1,x_2)) &= 0 \\ POL(\mathsf{S1}(x_1)) &= 1 + x_1 \\ POL(\mathsf{cons0}(x_1,x_2)) &= 0 \\ POL(\mathsf{SUBST1}(x_1,x_2)) &= 1 + x_1 \end{aligned}$$

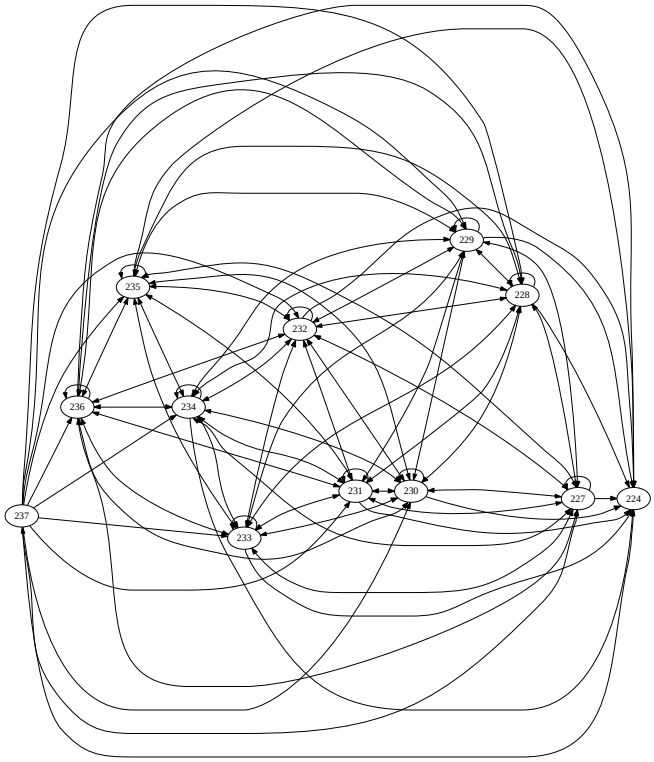
resulting in one new DP problem.

R
↪DPs
→DP Problem 1
↪Polo
→DP Problem 2
↪Polo
→DP Problem 5
↪**Dependency Graph**
→DP Problem 3
↪Polo

The node numbers are specified as follows:

Using the Dependency Graph resulted in no new DP problems.

R
↪DPs
→DP Problem 1
↪Polo
→DP Problem 2
↪Polo
→DP Problem 3
↪**Polynomial Ordering**




The node numbers are specified as follows:

$$
\begin{array}{rrcl}
224: & \mathsf{INC}(l, \mathsf{dotin0}(t1, t2)) & \to & \mathsf{IN0}(\mathsf{subst0}(t1, l), \mathsf{subst1}(t2, l)) \\
227: & \mathsf{INC}(l, \mathsf{supset}(A, B)) & \to & \mathsf{INC}(l, A) \\
228: & \mathsf{INC}(l, \mathsf{supset}(A, B)) & \to & \mathsf{INC}(l, B) \\
229: & \mathsf{INC}(l, \mathsf{cup}(A, B)) & \to & \mathsf{INC}(l, A) \\
230: & \mathsf{INC}(l, \mathsf{cup}(A, B)) & \to & \mathsf{INC}(l, B) \\
231: & \mathsf{INC}(l, \mathsf{cap}(A, B)) & \to & \mathsf{INC}(l, A) \\
232: & \mathsf{INC}(l, \mathsf{cap}(A, B)) & \to & \mathsf{INC}(l, B) \\
233: & \mathsf{INC}(l, \mathsf{P0}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A) \\
234: & \mathsf{INC}(l, \mathsf{P1}(A)) & \to & \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A) \\
235: & \mathsf{INC}(l, \mathsf{C0}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A) \\
236: & \mathsf{INC}(l, \mathsf{C1}(A)) & \to & \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A) \\
237: & \mathsf{IN0}(x, \mathsf{comp1}(A)) & \to & \mathsf{INC}(\mathsf{cons0}(x, \mathsf{nil}), A)
\end{array}
$$

The following dependency pairs can be strictly oriented:

$$
\begin{array}{rcl}
\mathsf{INC}(l, \mathsf{cup}(A, B)) & \to & \mathsf{INC}(l, A) \\
\mathsf{INC}(l, \mathsf{P1}(A)) & \to & \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A)
\end{array}
$$

$$\begin{aligned}
\mathsf{INC}(l, \mathsf{cap}(A, B)) &\to \mathsf{INC}(l, A)\\
\mathsf{INC}(l, \mathsf{C0}(A)) &\to \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A)\\
\mathsf{INC}(l, \mathsf{cup}(A, B)) &\to \mathsf{INC}(l, B)\\
\mathsf{INC}(l, \mathsf{supset}(A, B)) &\to \mathsf{INC}(l, B)\\
\mathsf{INC}(l, \mathsf{C1}(A)) &\to \mathsf{INC}(\mathsf{cons1}(\mathsf{fresh}, l), A)\\
\mathsf{INC}(l, \mathsf{dotin0}(t1, t2)) &\to \mathsf{IN0}(\mathsf{subst0}(t1, l), \mathsf{subst1}(t2, l))\\
\mathsf{IN0}(x, \mathsf{comp1}(A)) &\to \mathsf{INC}(\mathsf{cons0}(x, \mathsf{nil}), A)\\
\mathsf{INC}(l, \mathsf{cap}(A, B)) &\to \mathsf{INC}(l, B)\\
\mathsf{INC}(l, \mathsf{P0}(A)) &\to \mathsf{INC}(\mathsf{cons0}(\mathsf{fresh}, l), A)\\
\mathsf{INC}(l, \mathsf{supset}(A, B)) &\to \mathsf{INC}(l, A)
\end{aligned}$$

Additionally, the following usable rules w.r.t. to the implicit AFS can be oriented:

$$\begin{aligned}
\mathsf{subst1}(t, \mathsf{nil}) &\to t\\
\mathsf{subst1}(\mathsf{un1}, \mathsf{cons1}(t, l)) &\to t\\
\mathsf{subst1}(\mathsf{S1}(n), \mathsf{cons0}(t, l)) &\to \mathsf{subst1}(n, l)\\
\mathsf{subst1}(\mathsf{S1}(n), \mathsf{cons1}(t, l)) &\to \mathsf{subst1}(n, l)
\end{aligned}$$

Used ordering: Polynomial ordering with Polynomial interpretation:

$$\begin{aligned}
POL(\mathsf{mult}(x_1, x_2)) &= 0\\
POL(\mathsf{supset}(x_1, x_2)) &= 1 + x_1 + x_2\\
POL(\mathsf{plus}(x_1, x_2)) &= 0\\
POL(\mathsf{subst0}(x_1, x_2)) &= 0\\
POL(\mathsf{IN0}(x_1, x_2)) &= x_2\\
POL(\mathsf{cup}(x_1, x_2)) &= 1 + x_1 + x_2\\
POL(\mathsf{un1}) &= 0\\
POL(\mathsf{C1}(x_1)) &= 1 + x_1\\
POL(\mathsf{subst1}(x_1, x_2)) &= x_1 + x_2\\
POL(\mathsf{cons0}(x_1, x_2)) &= x_2\\
POL(\mathsf{P0}(x_1)) &= 1 + x_1\\
POL(\mathsf{S0}(x_1)) &= 0\\
POL(\mathsf{INC}(x_1, x_2)) &= x_1 + x_2\\
POL(\mathsf{comp1}(x_1)) &= 1 + x_1\\
POL(\mathsf{C0}(x_1)) &= 1 + x_1\\
POL(\mathsf{cons1}(x_1, x_2)) &= x_1 + x_2\\
POL(\mathsf{P1}(x_1)) &= 1 + x_1\\
POL(\mathsf{un0}) &= 0\\
POL(\mathsf{cap}(x_1, x_2)) &= 1 + x_1 + x_2\\
POL(\mathsf{S1}(x_1)) &= 1 + x_1\\
POL(\mathsf{nil}) &= 0\\
POL(\mathsf{s}(x_1)) &= 0
\end{aligned}$$

$$POL(\mathsf{dotin0}(x_1, x_2)) = 1 + x_2$$
$$POL(\mathsf{fresh}) = 0$$

resulting in one new DP problem.

R
↪DPs
→DP Problem 1
↪Polo
→DP Problem 2
↪Polo
→DP Problem 3
↪Polo
→DP Problem 6
↪**Dependency Graph**

The node numbers are specified as follows:

Using the Dependency Graph resulted in no new DP problems.
**Termination** of $\mathcal{HO}_1$ successfully shown.
Duration:
0:02 minutes

ACKNOWLEDGMENTS

The author wishes to thank G. Dowek, T. Hardin and C. Kirchner for many discussions and comments about this paper.

REFERENCES

Allali, L. 2007. Algorithmic equality in heyting arithmetic modulo. In *TYPES*, M. Miculan, I. Scagnetto, and F. Honsell, Eds. LNCS. Springer. To appear.

Altenkirch, T. and McBride, C., Eds. 2007. *Types for Proofs and Programs, International Workshop, TYPES 2006, Nottingham, UK, April 18-21, 2006, Revised Selected Papers*. LNCS, vol. 4502. Springer.

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. 41–50.

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

Bruscoli, P. and Guglielmi, A. 2008. On the proof complexity of deep inference. *ACM Trans. Comput. Logic*. To appear.

Burel, G. 2007. Unbounded proof-length speed-up in deduction modulo. In *CSL 2007*, J. Duparc and T. A. Henziger, Eds. LNCS, vol. 4646. Springer, 496–511.

Burel, G. 2008. A first-order representation of pure type systems using superdeduction. In *LICS*, F. Pfenning, Ed. IEEE Computer Society. To appear, available at `http://www.loria.fr/~burel/download/NJ+asLF.pdf`.

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. and Reckhow, R. 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, New York, NY, USA, 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.

Curry, H. B., Feys, R., and Craig, W. 1958. *Combinatory Logic*. Vol. 1. Elsevier Science Publishers B. V. (North-Holland), Amsterdam.

Dowek, G. 2006. Truth values algebras and proof normalization. See Altenkirch and McBride [2007], 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 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. Translated in Szabo, editor., *The Collected Papers of Gerhard Gentzen* as "Investigations into Logical Deduction".

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. 2006. A finite first-order theory of classes. See Altenkirch and McBride [2007], 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*. Paris, France.

Nipkow, T., Paulson, L. C., and Wenzel, M. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. LNCS, vol. 2283. Springer.

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

Schwichtenberg, H. 2007. *New Computational Paradigms*. Springer, Chapter New Develoments in Proofs and Computations. To appear, available at `http://www.mathematik.uni-muenchen.de/~schwicht/papers/cie05/cie05.pdf`.

The Coq Development Team. 2006. *The Coq Proof Assistant Reference Manual*. INRIA. Version 8.0, available at `http://coq.inria.fr/doc/main.html`.

Tseytin, G. S. 1968. On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic II*. Zapiski Nauchnykh Seminarov LOMI, vol. 8. Nauka, Leningrad, 235–259. In Russian.

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