Unbounded Proof-Length Speed-up in Deduction Modulo

Guillaume Burel[1]

[1] Université Henri Poincaré & LORIA[2]
Campus scientifique BP 239 — 54506 Vandœuvre-lès-Nancy Cedex France
guillaume.burel@ens-lyon.org
[2] UMR 7503 CNRS-INPL-INRIA-Nancy2-UHP

Abstract. In 1973, Parikh proved a speed-up theorem conjectured by Gödel 37 years before: there exist arithmetical formulæ that are provable in first order arithmetic, but whose shorter proof in second order arithmetic is arbitrarily smaller than any proof in first order. On the other hand, resolution for higher order logic can be simulated step by step in a first order narrowing and resolution method based on deduction modulo, whose paradigm is to separate deduction and computation to make proofs clearer and shorter.
We prove that $i+1$-th order arithmetic can be linearly simulated into i-th order arithmetic modulo some confluent and terminating rewrite system. We also show that there exists a speed-up between i-th order arithmetic modulo this system and i-th order arithmetic without modulo. All this allows us to prove that the speed-up conjectured by Gödel does not come from the deductive part of the proofs, but can be expressed as simple computation, therefore justifying the use of deduction modulo as an efficient first order setting simulating higher order.

Key words: proof theory, rewriting, higher order logic, arithmetic

1 Introduction

Even if two logical systems are shown to be expressively equivalent, i.e. they can prove exactly the same formulæ, they can lead to very different proofs, in particular in terms of length. For instance, it is shown that Frege systems have an exponential speed-up over resolution for propositional logic [5]. However in mechanized theorem proving, the length of proofs has an importance: First, computers have limited capacities, and this can lead to a difference between the practical expressiveness of theoretically equivalent systems. 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 proofs of non-elementary length. Second, the length of a proof is one (among others) criterion for defining the quality of a proof. Indeed, a smaller proof is often more readable and, in the case for instance of software certification and proof engineering, more communicable and often also more maintainable. In [10, 2], this notion of "good proofs" is translated through a proof ordering, which of course may correspond to the comparison of proof lengths.

Obtaining a speed-up can also have a theoretical interest, because, as remarked by Parikh in the introductory paragraph of [16], "the celebrated P=NP? question can itself be thought of as a speed-up question." (See [7].) All this explains the research for new formalisms whose deductive systems provide smaller proofs, such as for instance the calculus of structures w.r.t. the sequent calculus [17].

In this paper, the length of a proof will correspond to its number of steps (sometimes called lines), whatever the actual size of the formulæ appearing in them is. Considering only the minimal length of proofs, the definition of a speed-up is the following: given some function h over natural numbers, a system has a speed-up for h over another one, if there exists an infinite set of formulæ provable in both of them, such that, if the length of the proofs in the first system is l and the length in the second system is k, then $k > h(l)$.

In 1936, Gödel [16] conjectured that there exists such a speed-up for all recursive functions between i-th order and $i+1$-th order arithmetic, no matter the formal system actually used. In other words, he stated that for all recursive functions h, it is possible to find an infinite set of formulæ such that, for each of them, denoted by P, if k is the minimal number of steps in the proofs of P in the i-th order arithmetic (k is assumed to exist, so that P is provable in it), and l is the minimal number of steps in the proofs of P in the $i+1$-th order arithmetic, then $k > h(l)$.

This result was proved for first-order arithmetic by Parikh [21], who actually proved a stronger theorem: this proof-length speed-up exists in fact also for non-recursive functions. This was generalized to all orders by Krajíček , and was proved for the true language of arithmetic by Buss [6] (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 ([6, 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 $P \in \mathcal{F}$, $Z_{i+1} \vdash_{\overline{k\ steps}} P$*
3. *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i \vdash_{\overline{k\ steps}} P$.*

Z_i corresponds to the $i+1$-th 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 $i+1$-th 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 take 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. 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 [12] is a presentation of a given logic —and the formalisms associated with it— identifying what corresponds to computation. The computational part of a proof is put in a congruence between formulæ modulo whom the application of the deduction rules takes place. This leads to the sequent calculus modulo and 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 proof of $t \times t = 0$ can be deduced from a proof π of $t = 0$:

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

Deduction modulo is logically equivalent to the considered logic [12, 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 smaller 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, we will consider in this paper only very simple rewrite systems: they will be finite, terminating, confluent (i.e. deterministic) and linear (variables in the left-hand side only appear once).

Besides, it is possible, in deduction modulo, to build proofs of Higher Order Logic using a first order system [11]. 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. Therefore, it seems reasonable to think that deduction modulo is able to give the same proof-length speed-ups as the ones occurring between $i+1$-th and i-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 speed-up theorem of Buss comes from the computational part of the proofs, we first define a linear translation between proofs in $i+1$-th order arithmetic and i-th order arithmetic modulo some rewrite system $\mathcal{R}_i$. Second, using this translation and Buss' theorem, we prove that there is no proof-length speed-up between $i+1$-th order arithmetic and i-th order arithmetic modulo, whereas there exists such a speed-up between i-th order arithmetic modulo and i-th order arithmetic without modulo. Therefore, we conclude that the speed-up between $i+1$-th order arithmetic and i-th order arithmetic lies in the modulo, i.e. the computational part of the proofs.

In the next section, we will recall the definition of a schematic system, and we will present such a system for i-th order arithmetic. The section 3 will define formally what deduction modulo, and in particular natural deduction modulo consists of. In Section 4 we will give the exact translations between a proof in the schematic system for i-th order arithmetic and a proof in natural deduction, modulo or not, as well as the simulation in natural deduction of $i+1$-th order arithmetic in i-th order arithmetic modulo. An upper bound of the increase in the length of the proofs due to these translations will be given. Finally, in Section 5 we will use these translations to determine the origin of the speed-up in arithmetic, and we will conclude about the interest of working within a first-order system modulo to simulate higher order. All the details can be found in the full version of this paper [4].

2 A Schematic System for *i*-th Order Arithmetic

2.1 Schematic Systems

We recall here, using Buss' terminology [6], 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, mainly to introduce the notations we will use. A (first order) many-sorted signature consists in 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 Σ 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[3]:

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

where A ranges over atomic propositions and x over variables. $P \Leftrightarrow Q$ will be used as a syntactic sugar for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$. Positions in a term or a formula, free variables and substitutions are defined as usual (see [1]). The

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

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

Then, given a many-sorted signature of first order logic, we can consider infinite sets of *metavariables* α^i for each sort i (which will be substituted by variables), of *term variables* τ^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 α^j *is not free in* Φ or s *is freely substitutable for* α^j *in* Φ where Φ is a metaformula and s a metaterm of sort j. It is denoted by

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

An inference with an empty set of premises will be called an *axiom schema*.

2.2 *i*-th Order Arithmetic

i-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 can be found in its totality in the full version [4]. The most representative inference rules are given here as examples:

$14 + 2 \times \mathbf{i}$ axiom schemata of classical logic. We take the one used by Gentzen [15, Chapter 5] to prove the equivalence of his formalisms with an Hilbert-type proof system:

$$(A \Rightarrow A \Rightarrow B) \Rightarrow A \Rightarrow B \quad (1)$$

$$(A \Rightarrow B) \Rightarrow (B \Rightarrow C) \Rightarrow A \Rightarrow C \quad (2)$$

$$(A \wedge B) \Rightarrow A \quad (3)$$

$$A(\tau^j) \Rightarrow \exists \alpha^j.\ A(\alpha^j) \quad \left(\tau^j \text{ is freely substitutable for } \alpha^j \text{ in } A(\alpha^j)\right) \quad (4)$$

$$A \vee (A \Rightarrow \bot) \quad (5)$$

$1 + 2 \times \mathbf{i}$ inference rules of classical logic. Again, we consider the one of [15]:

$$\frac{A \quad A \Rightarrow B}{B} \quad (6)$$

$$\frac{B(\beta^j) \Rightarrow A}{(\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A}\ (\beta^j \text{ is not free in } (\exists \alpha^j.\ B(\alpha^j)) \Rightarrow A) \quad (7)$$

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

$$\forall\alpha^0.\ \alpha^0 = \alpha^0 \tag{8}$$
$$\forall\alpha^0\beta^0\gamma^0.\ \alpha^0 = \beta^0 \Rightarrow \alpha^0 + \gamma^0 = \beta^0 + \gamma^0 \tag{9}$$
$$\forall\alpha^0\beta^0.\ \alpha^0 = \beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0) \tag{10}$$

7 Robinson's axioms. They are the axioms defining the function symbols of arithmetic [19]:

$$\forall\alpha^0\beta^0.\ s(\alpha^0) = s(\beta^0) \Rightarrow \alpha^0 = \beta^0 \tag{11}$$
$$\forall\alpha^0.\ \alpha^0 \times 0 = 0 \tag{12}$$
$$\forall\alpha^0\beta^0.\ \alpha^0 \times s(\beta^0) = \alpha^0 \times \beta^0 + \alpha^0 \tag{13}$$

$i+1$ induction and comprehension axiom schemata.

$$A(0) \Rightarrow \left(\forall\beta^0.\ A(\beta^0) \Rightarrow A(s(\beta^0))\right) \Rightarrow \forall\alpha^0.\ A(\alpha^0) \tag{14}$$

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

$$\exists\alpha^{j+1}.\ \forall\beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j) \quad (\alpha^{j+1} \text{ is not free in } A) \tag{15}$$

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 can be found in [12, 13]. 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 can be found in [1], and extend them to proposition rewriting [12].

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 σ and some position $\mathfrak{p}$ in s such that $\sigma l = s_{|\mathfrak{p}}$ and $t = s[\sigma r]_{\mathfrak{p}}$. An atomic proposition $A(s_1, \dots, s_i, \dots, s_n)$ can be rewritten to the atomic proposition $A(s_1, \dots, t_i, \dots, s_n)$ 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

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

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

$$\vee\text{-i}\ \frac{A}{C}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \vee B \text{ or } C \xleftrightarrow[\mathcal{R}]{*} B \vee A \qquad \vee\text{-e}\ \frac{C \quad \begin{matrix}[A]\\ D\end{matrix} \quad \begin{matrix}[B]\\ D\end{matrix}}{D}\ \text{if } C \xleftrightarrow[\mathcal{R}]{*} A \vee B$$

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

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

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

Fig. 1. Inference Rules of Natural Deduction Modulo.

exists some substitution σ 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 $A \xrightarrow[\mathcal{R}]{} P$. The transitive (resp. reflexive transitive) closure of this relation will be denoted by $\xrightarrow[\mathcal{R}]{*}$ (resp. $\xleftrightarrow[\mathcal{R}]{*}$).

3.2 Natural Deduction Modulo

Using some equivalence $\xleftrightarrow[\mathcal{R}]{*}$ defined by a rewrite system $\mathcal{R}$, we can define natural deduction modulo as in [13]. Its inference rules are represented in Fig. 1. They are the same as the one introduced by Gentzen [15], 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 (6).

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 (8) to (15).

4 Translations

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 (8) to (15).

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

$$\Rightarrow\text{-i}\ \dfrac{\exists\text{-i}\ \dfrac{A(\tau^j)\ \text{(i)}}{\exists \alpha^j.\ A(\alpha^j)}}{A(\tau^j) \Rightarrow \exists \alpha^j.\ A(\alpha^j)}\ \text{(i)}$$

and the inference rule (7) into

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

(note that the side condition ensures that it is possible to consider that what will be substituted for β^j 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 (8) to (15), and whose length is linear compared to the schematic system proof:

Proposition 1. *It is possible to translate a proof of length n in the schematic system for Z_i into a proof of length $O(n)$ in (pure) natural deduction using assumptions in Z_i.*

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

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 (8) to (15) 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 λ-calculus to combinatory logic (see [9]). We define mutually recursively two functions by induction on the inference rules: T transforms a proof of P in natural deduction using assumptions Γ into a proof of P in the schematic system (1) to (7) plus Γ. T_A transform a proof of P in natural deduction using assumptions Γ, A into a proof of $A \Rightarrow P$ in the schematic system (1) to (7) plus Γ.

Due to lack of space, the definition of T and T_A is given here only for the existential quantifier, but can be entirely found in the full version of this paper [4].

$$T\left(\begin{array}{c}\pi\\ \exists\text{-i}\ \dfrac{\{t/x\}A}{\exists x.\ A}\end{array}\right) \overset{!}{=} (6)\ \dfrac{\begin{array}{c}T(\pi)\\ \{t/x\}A\end{array} \qquad \{t/x\}A \Rightarrow \exists x.\ A\ (4)}{\exists x.\ A}$$

$$T\left(\exists\text{-e}\ \dfrac{\begin{array}{c}\pi_1\\ \exists x.\ A\end{array} \qquad \pi_2\left\{\begin{array}{c}[\{y/x\}A]\\ B\end{array}\right.}{B}\right) \overset{!}{=} (6)\ \dfrac{\begin{array}{c}T(\pi_1)\\ \exists x.\ A\end{array} \qquad (7)\ \dfrac{\begin{array}{c}T_A(\pi_2)\\ \{y/x\}A \Rightarrow B\end{array}}{(\exists x.\ A) \Rightarrow B}}{B}$$

$$T_A\left(\begin{array}{c}\pi\left\{\begin{array}{c}[A]\\ \{t/x\}B\end{array}\right.\\ \exists\text{-i}\ \dfrac{}{\exists x.\ B}\end{array}\right) \overset{!}{=}$$

$$(6)\ \dfrac{\{t/x\}B \Rightarrow \exists x.\ B\ (4) \qquad (6)\ \dfrac{\begin{array}{c}T_A(\pi)\\ A \Rightarrow \{t/x\}B \qquad \cdots\ (2)\end{array}}{(\{t/x\}B \Rightarrow \exists x.\ B) \Rightarrow A \Rightarrow \exists x.\ B}}{A \Rightarrow \exists x.\ B}$$

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

$$(6)\ \dfrac{(6)\ \dfrac{(7)\ \dfrac{\begin{array}{c}T_{\{y/x\}B}\begin{pmatrix}T_A(\pi_2)\\ A \Rightarrow C\end{pmatrix}\\ \{y/x\}B \Rightarrow A \Rightarrow C\end{array}}{\exists x.\ B \Rightarrow A \Rightarrow C} \qquad (6)\ \dfrac{\begin{array}{c}T_A(\pi_1)\\ A \Rightarrow \exists x.\ B \qquad \cdots\ (2)\end{array}}{(\exists x.\ B \Rightarrow A \Rightarrow C) \Rightarrow A \Rightarrow A \Rightarrow C}}{A \Rightarrow A \Rightarrow C} \qquad \cdots\ (1)}{A \Rightarrow C}$$

It can be verified that this definition transforms a proof of size n into a proof of size $O(3^n)$. Due to [7, 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 2. *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.*

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

4.3 From $Z_{i+1} \vdash^{\mathsf{S}}$ and $Z_{i+1} \vdash^{\mathsf{N}}$ to $Z_i \vdash^{\mathsf{N}}_{\mathcal{R}_i}$

This time, we translate a proof in the schematic system for Z_{i+1} into a proof in natural deduction modulo using as assumption instances of the axiom schemata (8) to (15), but in the language of Z_i. The point is that, using modulo, it is possible to downshift an order.

We follow the translation of Section 4.1, except for the axiom schemata (10), (14) and (15) that are instantiated by formulæ that are in the language of Z_{i+1} but not in the language of Z_i. To translate these schemata, we will use the work of F. Kirchner [18] which permits to express first-order theories using a finite number of axioms. The idea is to transform some metaformula $A(t_1, \dots, t_n)$ used in an axiom schema into a formula of the form $\langle t_1, \dots, t_n \rangle \mathrel{\epsilon} \gamma$ where γ will be some term representing what formula will be actually substituted for A.

Following F. Kirchner's method, we add new sorts ℓ for lists and c for classes, as well as new function symbols and predicate

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

$\langle \alpha_1, \dots, \alpha_n \rangle$ 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 (10), (14) and (15) into the following *axioms*:

$$\forall \gamma^c.\ \forall \alpha^0 \beta^0.\ \alpha^0 = \beta^0 \Rightarrow \langle \alpha^0 \rangle \mathrel{\epsilon} \gamma^c \Rightarrow \langle \beta^0 \rangle \mathrel{\epsilon} \gamma^c \tag{16}$$

$$\forall \gamma^c. \langle 0 \rangle \mathrel{\epsilon} \gamma^c \Rightarrow \left(\forall \beta^0.\ \langle \beta^0 \rangle \mathrel{\epsilon} \gamma^c \Rightarrow \langle s(\beta^0) \rangle \mathrel{\epsilon} \gamma^c \right) \Rightarrow \forall \alpha^0.\ \langle \alpha^0 \rangle \mathrel{\epsilon} \gamma^c \tag{17}$$

For all $0 \leq j < i$,

$$\forall \gamma^c.\ \exists \alpha^{j+1}.\ \forall \beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow \langle \beta^j \rangle \mathrel{\epsilon} \gamma^c \tag{18}$$

The rewrite system $\mathcal{R}_i$ is then the following:

$$\begin{array}{rcl} t[nil]^j & \to & t \\ 1^j[t ::^j l]^j & \to & t \\ S^j(n)[t ::^j l]^j & \to & n[l]^j \\ s(n)[l]^0 & \to & s(n[l]^0) \\ (t_1 + t_2)[l]^0 & \to & t_1[l]^0 + t_2[l]^0 \\ (t_1 \times t_2)[l]^0 & \to & t_1[l]^0 \times t_2[l]^0 \\ l \mathrel{\epsilon} \doteq (t_1, t_2) & \to & t_1[l]^0 = t_2[l]^0 \end{array} \qquad \begin{array}{rcl} l \mathrel{\epsilon} \dot{\in}^j(t_1, t_2) & \to & t_1[l]^j \in^j t_2[l]^{j+1} \\ l \mathrel{\epsilon} A \cup B & \to & l \mathrel{\epsilon} A \vee l \mathrel{\epsilon} B \\ l \mathrel{\epsilon} A \cap B & \to & l \mathrel{\epsilon} A \wedge l \mathrel{\epsilon} B \\ l \mathrel{\epsilon} A \supset B & \to & l \mathrel{\epsilon} A \Rightarrow l \mathrel{\epsilon} B \\ l \mathrel{\epsilon} \emptyset & \to & \bot \\ l \mathrel{\epsilon} \mathcal{P}^j(A) & \to & \exists x.\ x ::^j l \mathrel{\epsilon} A \\ l \mathrel{\epsilon} \mathcal{C}^j(A) & \to & \forall x.\ x ::^j l \mathrel{\epsilon} A \end{array}$$

Note that this system is finite, terminating (either the size of a list decreases, or else a $\cdot[\cdot]$ or an ϵ goes more inside or disappears), confluent (the only critical pairs, of the form: $f(t_1, \dots, t_n) \underset{\mathcal{R}_i}{\longleftarrow} f(t_1, \dots, t_n)[nil] \underset{\mathcal{R}_i}{\longrightarrow} f(t_1[nil], \dots, t_n[nil])$, are easily joinable), and linear (variables appears only once in the left hand side of the rewrite rules).

Proposition 2 of [18] says that it is possible, for any formula P of the language of i-th order arithmetic, to prove $\exists E.\ \forall x_1 \cdots x_n.\ \langle x_1, \dots, x_n \rangle \mathrel{\epsilon} E \Leftrightarrow P$. Moreover, the proof of this proposition shows us how to construct the witness E. We will denote it by $E_P^{x_1,\dots,x_n}$. Then, one can prove that $\langle t_1, \dots, t_n \rangle \mathrel{\epsilon} E_P^{x_1,\dots,x_n} \xrightarrow[\mathcal{R}_i]{*} \{t_1/x_1, \dots, t_n/x_n\} P$. For instance, consider the formula $x = 0 \vee \exists y.\ x \in^0 y$, which will be denoted by P. Then E_P^x equals $\doteq (1, S(0)) \ \cup \ \mathcal{P}^1 \left(\dot{\in}^0(S(1), 1) \right)$ and $\langle t \rangle \mathrel{\epsilon} E_P^x$ can be rewritten to $t = 0 \vee \exists x.\ t \in^0 x$.

$$\forall\text{-e}\ \frac{\forall\gamma^c.\ \forall\alpha^0\beta^0.\ \alpha^0=\beta^0 \Rightarrow \langle\alpha^0\rangle \mathrel{\epsilon} \gamma^c \Rightarrow \langle\beta^0\rangle \mathrel{\epsilon} \gamma^c\ (16)}{\forall\alpha^0\beta^0.\ \alpha^0=\beta^0 \Rightarrow A(\alpha^0) \Rightarrow A(\beta^0)}\ \ \langle\alpha^0\rangle \mathrel{\epsilon} E_A^x \Rightarrow \langle\beta^0\rangle \mathrel{\epsilon} E_A^x \xrightarrow[\mathcal{R}_i]{*} A(\alpha^0) \Rightarrow A(\beta^0)$$

$$\forall\text{-e}\ \frac{\forall\gamma^c.\langle 0\rangle \mathrel{\epsilon} \gamma^c \Rightarrow \big(\forall\beta^0.\ \langle\beta^0\rangle \mathrel{\epsilon} \gamma^c \Rightarrow \langle s(\beta^0)\rangle \mathrel{\epsilon} \gamma^c\big) \Rightarrow \forall\alpha^0.\ \langle\alpha^0\rangle \mathrel{\epsilon} \gamma^c\ (17)}{A(0) \Rightarrow \big(\forall\beta^0.\ A(\beta^0) \Rightarrow A(s(\beta^0))\big) \Rightarrow \forall\alpha^0.\ A(\alpha^0)}\ \ \text{for all } t,\ \langle t\rangle \mathrel{\epsilon} E_A^x \xrightarrow[\mathcal{R}_i]{*} A(t)$$

$$\forall\text{-e}\ \frac{\forall\gamma^c.\ \exists\alpha^{j+1}.\ \forall\beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow \langle\beta^j\rangle \mathrel{\epsilon} \gamma^c\ (18)}{\exists\alpha^{j+1}.\ \forall\beta^j.\ \beta^j \in^j \alpha^{j+1} \Leftrightarrow A(\beta^j)}\ \ \langle\beta^j\rangle \mathrel{\epsilon} E_A^x \xrightarrow[\mathcal{R}_i]{*} A(\beta^j)$$

Fig. 2. Translations of the axiom schemata (10), (14) and (15).

Consequently, the axiom schemata (10), (14) and (15) for formulæ of the language of Z_{i+1} but not in the language of Z_i are replaced by the proofs in Fig. 2. In these translations, we need to instantiate γ^c with some E_A^x. It is well-known that the instantiations are the most problematic rules in deductive 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.

Using this, a proof in the schematic system for Z_{i+1} can be translated into a proof of P in natural deduction modulo $\mathcal{R}_i$ using as assumptions the axioms (10), (14) and (15) as well as a finite subset of instances the axiom schemata (8) to (15) for i-th order arithmetic, and whose length is linear compared to the schematic system proof:

Proposition 3. *It is possible to translate a proof of length n in the schematic system for Z_{i+1} into a proof of length $O(n)$ in the natural deduction modulo $\mathcal{R}_i$ using assumptions in Z_i, (16), (17) and (18).*

$$Z_{i+1} \mathrel{\substack{\mathsf{S}\\ \vdash \\ k}} P \quad\rightsquigarrow\quad Z_i, (16), (17), (18) \mathrel{\substack{\mathsf{N}\\ \vdash \\ O(k)}}_{\mathcal{R}_i} P$$

This result can also be stated entirely in natural deduction

Theorem 2. *For all $i \geq 0$, there exists a (finite) rewrite system $\mathcal{R}_i$ and a finite set of axioms Γ such that for all formulæ P, if $Z_{i+1} \mathrel{\substack{\mathsf{N}\\ \vdash \\ k}} P$ then $Z_i, \Gamma \mathrel{\substack{\mathsf{N}\\ \vdash \\ O(k)}}_{\mathcal{R}_i} P$.*

Proof. Let Γ be $\{(16), (17), (18)\}$. We replace the instance of the axiom schemata (10), (14) and (15) by the axioms (16), (17) and(18) as indicated in Fig. 2. □

Note that, contrarily to HOL-$\lambda\sigma$ which permits to simulate Higher Order Logic, the rewrite system purposed here is finite and terminating.

The fact to add the finite set of axioms Γ could be seen as some deceit, because we do not work really in Z_i, but in a theory strictly stronger. By the way, due to Theorem 2, it is possible to prove the consistency of Z_i in Z_i, Γ modulo

$\mathcal{R}_i$. Nevertheless, the point here is that it is possible, by working modulo $\mathcal{R}_i$, to simulate Z_{i+1} using a finite set of axioms, and not axiom schemata, without exploding the length of the proofs. If we were not working modulo this rewrite system, but using a finite theory compatible with it (i.e. proving exactly the same formulæ), then it would not be possible to give a bound to the translation:

Proposition 4. *For all $i \geq 0$, for all finite theories T_i compatible with $\mathcal{R}_i$, there is an infinite family $\mathcal{F}$ such that*

1. *for all $P \in \mathcal{F}$, $Z_i, \Gamma, T_i \vdash^{\mathsf{S}} P$*
2. *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_{i+1} \vdash^{\mathsf{S}}_{k\ steps} P$*
3. *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i, \Gamma, T_i \vdash^{\mathsf{S}}_{k\ steps} P$.*

It could also have been possible to translate the formulæ that one wants to prove, as is done in [14], where a formula of first order arithmetic is transformed by adding the information that some variable n is an integer using some predicate $N(n)$ which can be rewritten into an axiom corresponding to the induction schema for first order arithmetic. Here, P could be translated into (16) $\Rightarrow$ (17) $\Rightarrow$ (18) $\Rightarrow P$.

5 Application to Speed-ups in Arithmetic

5.1 Bypassing Buss' Speed-up using Modulo

The goal of this section is to prove that one can work in Z_i modulo some rewrite system $\mathcal{R}_i$ to be able to build proof as small as the one of Z_{i+1}. Indeed, Theorem 2 permits to show that Gödel's theorem does not extend if one works modulo $\mathcal{R}_i$ (what is formulated here in a positive way):

Corollary 1 (of Theorem 2). *For all $i \geq 0$, there exists a (finite) rewrite system $\mathcal{R}_i$ and a finite set of axioms Γ such that for all infinite family $\mathcal{F}$ of $\prod_1^0$-formulæ, if*

- *for all $P \in \mathcal{F}$, $Z_i \vdash^{\mathsf{N}} P$*
- *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_{i+1} \vdash^{\mathsf{N}}_{k\ steps} P$*

then there is a fixed $k' \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i, \Gamma \vdash^{\mathsf{N}}_{k'\ steps}{}_{\mathcal{R}_i} P$.

5.2 Speed-up Due to Computation

On the contrary, we want to show that it is possible to achieve the same speed-up as the one between i-th order and $i+1$-th order arithmetic just by working modulo some rewrite system in i-th order arithmetic:

Theorem 3. *For all $i \geq 0$, there is a rewrite system $\mathcal{R}_i$ such that there is an infinite family $\mathcal{F}$ such that*

1. *for all $P \in \mathcal{F}$, $Z_i \vdash^{\mathsf{N}} P$*

2. *there is a fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i \vdash^{\mathsf{N}}_{k\ steps\ \mathcal{R}_i} P$*
3. *there is no fixed $k \in \mathbb{N}$ such that for all $P \in \mathcal{F}$, $Z_i \vdash^{\mathsf{N}}_{k\ steps} P$.*

Proof. The rewrite system $\mathcal{R}_i$ is the one defined in Section 4.3. Let $\mathcal{F}$ be the family of formulæ obtained by Theorem 1. Let $\mathcal{F}' \stackrel{!}{=} \{(16) \Rightarrow (17) \Rightarrow (18) \Rightarrow P : P \in \mathcal{F}\}$. Then:

1. For all $P' \in \mathcal{F}'$, $Z_i \vdash^{\mathsf{N}} P'$: we know that $Z_i \vdash^{\mathsf{S}} P$, therefore using Proposition 1, $Z_i \vdash^{\mathsf{N}} P$ and, adding to this proof $2+i$ times $\Rightarrow$-i, $Z_i \vdash^{\mathsf{N}} P'$.
2. There is a k such that for all $P' \in \mathcal{F}'$, $Z_i \vdash^{\mathsf{N}}_{k\ \mathcal{R}_i} P'$: there exists some k such that for all $P \in \mathcal{F}$, $Z_{i+1} \vdash^{\mathsf{S}}_{k} P$. Using Proposition 3, there exists some K such that for all $P \in \mathcal{F}$, we have $Z_i, (16), (17), (18) \vdash^{\mathsf{S}}_{K\ \mathcal{R}_i} P$ and one can add $2+i$ times $\Rightarrow$-i to obtain a proof of P'.
3. There is no k such that for all $P' \in \mathcal{F}'$, $Z_i \vdash^{\mathsf{N}}_{k} P'$: Suppose by contradiction that there is a k such that for all $P' \in \mathcal{F}'$, $Z_i \vdash^{\mathsf{N}}_{k} P'$, then using $2+i$ times $\Rightarrow$-e, we have $Z_i, (16), (17), (18) \vdash^{\mathsf{N}}_{k+2+i} P$. But (16), (17) and (18) use function symbols not appearing in P nor Z_i (for instance ϵ). Therefore they cannot be used in a proof of P in Z_i, so that in fact $Z_i \vdash^{\mathsf{N}}_{k+2+i} P$. Then, using Proposition 2, $Z_i \vdash^{\mathsf{S}}_{O(3^k)} P$, and that will be in contradiction with the fact that there is no K such that for all P, $Z_i \vdash^{\mathsf{S}}_{K} P$.

Schematically,

$$\begin{array}{ccccc} Z_{i+1} \vdash^{\mathsf{S}}_{k} P & \stackrel{\text{Prop. 3}}{\rightsquigarrow} & Z_i, (16), (17), (18) \vdash^{\mathsf{N}}_{K\ \mathcal{R}_i} P & \rightsquigarrow & Z_i \vdash^{\mathsf{N}}_{K+2+i\ \mathcal{R}_i} P' \\ \text{Theo. 1} \updownarrow & & & & \\ Z_i \vdash^{\mathsf{S}}_{\not{3}^{\not{k}}} P & \underset{\text{Prop. 2}}{\overset{\text{Prop. 1}}{\rightsquigarrow \atop \leftsquigarrow}} & Z_i, (16), (17), (18) \vdash^{\mathsf{N}}_{\not{k}} P & \rightsquigarrow & Z_i \vdash^{\mathsf{N}}_{\not{k}} P' \end{array}$$

□

Note that it is possible to get speed-ups in deduction modulo w.r.t pure natural deduction with systems much more simpler than for higher order arithmetic. (Take for instance the rule $s(x) + y \rightarrow x + s(y)$ and consider the formulæ $\underline{n} + \underline{n} = \underline{n+n}$ where $\underline{n}$ denotes the usual representation of the natural number n using 0 and s, for all natural numbers n.) Our last result however, combined with Corollary 1, permits to conclude that proof-length speed-ups *in arithmetic* result from the computational part of the proofs, which is expressed by the rewrite systems $\mathcal{R}_i$.

6 Conclusion and Perspectives

We have first proved that it is possible to use some rewrite system to simulate the difference between i-th and $i+1$-th order arithmetic at the condition to add three extra axioms which replace the missing axiom schemata. This simulation is linear in terms of proof length, which permits to prove that there is no proof-length speed-up between $i+1$-th order arithmetic and its simulation, on the contrary to without modulo as it is expressed in Buss' theorem. Furthermore, this simulation allows to get the same proof speed-up for deduction modulo over non modulo systems than the one of Buss' theorem. Together with the first result, this proves that the gap between i-th and $i+1$-th order arithmetic is in fact due to the computational part of the proofs. In this particular case, we also clearly identify the computation occurring in the proofs with a finite, terminating and confluent (so, in a sense, deterministic) rewrite system. This is not surprising, because, if one looks carefully, the proof of Theorem 1 given by Buss in [6] 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 $i+1$-th order arithmetic, to compute the validity of a formula in i-th order arithmetic.

Speed-ups in deduction modulo 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 (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, as well as in representation of proofs in natural language (where all computational details are often implicitly left the reader).

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 [22] or Isabelle/HOL [20], 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, see [3]). This paper gives clues to answer positively this question, although we were interested in the step between i-th order and $i+1$-th order arithmetic, and not between first order and higher order logic. The fact that higher order resolution can be simulated step by step by ENAR [11] is not a solution, because there may exist some other higher order proof system that produce proofs that cannot be conveniently translated in a first order system modulo. So, our next challenge will be, starting from the current results, to investigate how exactly higher order logic prevails or not over first order logic, by studying more closely the simulation of higher order logic.

A first direction to do so will be to prove that it is possible to apply transitivity between the simulation of Z_{i+1} in Z_i and the one of Z_{i+2} in Z_{i+1}, in order to get a simulation of Z_{i+2} in Z_i, for instance by combining $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$. In addition to the expression of first order arithmetic as a theory modulo [14], this would lead to the linear simulation of higher order arithmetic entirely as a theory modulo. It should however be noted that one of the main advantage of our rewrite systems w.r.t. HOL-$\lambda\sigma$, i.e. its finiteness, will be lost because of the need for a rule to decompose $\dot{\in}^i$ for all orders i.

Another direction would be to look directly at the difference of the lengths of proofs in the expression of HOL in the sequent calculus modulo [11], or of every PTS in $\lambda\Pi$ modulo [8].

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

References

1. Baader, F., Nipkow, T.: Term *Re*writing and *all That*. CUP (1998)
2. Bonacina, M.P., Dershowitz, N.: Abstract canonical inference. ACM Trans. Comput. Logic **8** (2007)
3. Brauner, P., Houtmann, C., Kirchner, C.: Principles of superdeduction. In: LICS. IEEE Computer Society (2007) To appear.
4. Burel, G.: Unbounded proof-length speed-up in deduction modulo. Research report (2007) Available at `http://hal.inria.fr/inria-00138195`.
5. Buss, S.R.: Polynomial size proofs of the propositional pigeonhole principle. The Journal of Symbolic Logic **52** (1987) 916–927
6. Buss, S.R.: On Gödel's theorems on lengths of proofs I: Number of lines and speedup for arithmetics. The Journal of Symbolic Logic **59** (1994) 737–756
7. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. The Journal of Symbolic Logic **44** (1979) 36–50
8. Cousineau, D., Dowek, G.: Embedding pure type systems in the lambda-pi-calculus modulo. In: TLCA. (2007) To appear.
9. Curry, H.B., Feys, R., Craig, W.: Combinatory Logic. Vol. 1. Elsevier Science Publishers B. V. (North-Holland), Amsterdam (1958)
10. Dershowitz, N., Kirchner, C.: Abstract Canonical Presentations. Theoretical Computer Science **357** (2006) 53–69
11. Dowek, G., Hardin, T., Kirchner, C.: HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. Math. Structures Comput. Sci. **11** (2001) 1–25
12. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo. Journal of Automated Reasoning **31** (2003) 33–72
13. Dowek, G., Werner, B.: Proof normalization modulo. The Journal of Symbolic Logic **68** (2003) 1289–1316
14. Dowek, G., Werner, B.: Arithmetic as a theory modulo. In: Giesl, J. (ed.): RTA. Lecture Notes in Computer Science, Vol. 3467. Springer-Verlag (2005) 423–437
15. Gentzen, G.: Untersuchungen über das logische Schliessen. Mathematische Zeitschrift **39** (1934) 176–210, 405–431 Translated in: Szabo, M.E. (ed.): *The Collected Papers of Gerhard Gentzen* as "Investigations into Logical Deduction".
16. Gödel, K.: On the length of proofs. In: Feferman, S., et al. (eds.): Kurt Gödel: Collected Works. Vol. 1. Oxford University Press, Oxford (1986) 396–399
17. Guglielmi, A.: Polynomial size deep-inference proofs instead of exponential size shallow-inference proofs. Av. at `http://cs.bath.ac.uk/ag/p/AG12.pdf` (2004)
18. Kirchner, F.: A finite first-order theory of classes. Available at `http://www.lix.polytechnique.fr/Labo/Florent.Kirchner/doc/fotc2006.pdf` (2006)
19. Mostowski, A., Robinson, R.M., Tarski, A.: Undecidable Theories. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam (1953)
20. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL — A Proof Assistant for Higher-Order Logic. LNCS, Vol. 2283. Springer (2002)
21. Parikh, R.J.: Some results on the length of proofs. Transactions of the ACM **177** (1973) 29–36
22. The Coq Development Team: The Coq Proof Assistant Reference Manual. INRIA. (2006) Version 8.0, available at `http://coq.inria.fr/doc/main.html`.