A Finite First-Order Theory of Classes

Florent Kirchner*

LIX, École Polytechnique, 91128 Palaiseau, France
florent.kirchner@lix.polytechnique.fr

Abstract. We expose a formalism that allows the expression of any theory with one or more axiom schemes using a finite number of axioms. This allows us to give finite first-order axiomatizations of arithmetic and real analysis, and a presentation of arithmetic in deduction modulo that has a finite number of rewrite rules. Overall, this formalization relies on a weak calculus of explicit substitutions to provide a simple and finite framework.

1 Introduction

In mathematics, some theories — such as arithmetic or set theory — are often expressed using an infinite number of axioms. This is achieved through the use of one or more *axiom schemes*, i.e. sets of axioms, often described within the meta-theory. Take for instance the induction scheme of arithmetic: for any proposition P,

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

This scheme, parametrized in P, generates an infinite number of axioms.

The use of these axiom schemes can be avoided though, by introducing a new sort of objects, *classes* (for which we will use uppercase letters), and a membership symbol $\in$. Using classes as representatives for propositions, the induction scheme is re-written as a single axiom:

$$\forall E\ (0 \in E \Rightarrow \forall y\ (y \in E \Rightarrow S(y) \in E) \Rightarrow \forall z\ z \in E)\ .$$

However, in order for classes to soundly emulate propositions, one needs to guarantee that belonging to a class is equivalent to satisfying a proposition. This is assured by the axiom scheme of comprehension, which states: for any proposition P that is well formed in the original language,

$$\exists E\ \forall x\ (x \in E \Leftrightarrow P)\ .$$

This extension of arithmetic is *conservative*, meaning that any formula that is provable in the theory of arithmetic plus classes, involving symbols of arithmetic only, was already provable in arithmetic. Also note that some axiom schemes

* Projet Logical, Pôle Commun de Recherche en Informatique du plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

might have propositions with *two* free variables or more, and that to deal with this in the most general way, one needs not only classes of objects but classes of n-tuples of objects. Therefore, one introduces a sort of lists, list constructors and adapts the symbol $\in$ and the axiom scheme of comprehension to this new structure. In the end, however, the theory still has an axiom scheme that generates an infinite number of axioms.

This is not a zero-sum game, though. Previous works [1, 2] have shown that, in the case of set theory, it is possible to reduce the comprehension axiom scheme to a finite number of axioms. However we believe that the notion of class is independent of set theory and can be extended to express any theory containing axiom schemes in a finite first-order axiomatization. Moreover we will see the axioms of the theory of classes can easily be oriented as rewrite rules.

This work is part of a long term project investigating the possibility to base proof checkers on weaker frameworks, such as first-order logic. The maturity of these frameworks make them very secure centerpieces of formal tools in general, and proof checkers in particular; some designs (e.g. [3, 4]) have already taken advantage of them. It is essential in this project to be able to implement strong theories, such as arithmetic, real analysis or set theory, with a finite number of axioms. The main contribution of this paper is a systematic way of conducting these implementations.

2 A Theory with the Comprehension Scheme

We consider a finite language $\mathscr{L}$ in first-order predicate logic, and $\mathcal{T}$ a theory of this language that has one or more axiom schemes, i.e. axioms of the form:

$$s(P(t_1^1/x_1, \ldots, t_n^1/x_n), \ldots, P(t_1^p/x_1, \ldots, t_n^p/x_n)) \ .$$

Definition 1 ($\mathscr{L}^{\mathsf{cs}}$). *Let $\mathscr{L}^{\mathsf{cs}}$ be a finite many-sorted language with three sort symbols: the sort of objects X, lists L^X, and classes CL^X. $\mathscr{L}^{\mathsf{cs}}$ extends $\mathscr{L}$ with a predicate symbol $\in$ of rank (L^X, CL^X) and two function symbols:*

$$\begin{aligned} nil &: L^X \\ :: &: (X, L^X)L^X \ . \end{aligned}$$

Remark 1. We use $\langle x_1, \ldots, x_n \rangle$ as syntactic sugar for the list construction $x_1 :: \ldots :: x_n :: nil$.

Definition 2 ($\mathcal{T}^{\mathsf{cs}}$). *Define $\mathcal{T}^{\mathsf{cs}}$ be the theory of the language $\mathscr{L}^{\mathsf{cs}}$ derived from $\mathcal{T}$ by adding the comprehension scheme:*

$$\exists E, \forall x_1, \ldots, \forall x_n, (\langle x_1, \ldots, x_n \rangle \in E \Leftrightarrow P) \qquad (\mathcal{T}_1^{\mathsf{cs}})$$

where P is built with the symbols of $\mathscr{L}$; and replacing the axiom schemes by axioms:

$$\forall E, s(\langle t_1^1, \ldots, t_n^1 \rangle \in E, \ldots, \langle t_1^p, \ldots, t_n^p \rangle \in E) \ . \qquad (\mathcal{T}_2^{\mathsf{cs}})$$

Example 1. Following the translation of Definition 2, the induction axiom scheme:

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

is replaced by the two axioms:

$$\exists E, \forall x_1, \ldots, \forall x_n, (\langle x_1, \ldots, x_n \rangle \in E \Leftrightarrow P)$$
$$\forall E, \langle 0 \rangle \in E \Rightarrow \forall y, (\langle y \rangle \in E \Rightarrow \langle S(y) \rangle \in E) \Rightarrow \forall z, \langle z \rangle \in E \ .$$

Proposition 1. *$\mathcal{T}^{\mathsf{cs}}$ is an extension of $\mathcal{T}$.*

Proof. We need to show that in $\mathcal{T}^{\mathsf{cs}}$, for any P,

$$s(P(t_1^1/x_1, \ldots, t_n^1/x_n), \ldots, P(t_1^p/x_1, \ldots, t_n^p/x_n)) \ .$$

is provable. This is immediate, by the comprehension scheme ($\mathcal{T}_1^{\mathsf{cs}}$) and the axioms ($\mathcal{T}_2^{\mathsf{cs}}$). □

3 Finite Class Theory

3.1 Setting and Notations.

Definition 3 ($\mathscr{L}^{\mathsf{ws}}$). *We extend $\mathscr{L}^{\mathsf{cs}}$ with the function symbols:*

$$1 : X \qquad \varnothing : CL^X$$
$$S : (X)X \qquad \cap, \cup, \supset : (CL^X, CL^X)CL^X$$
$$\cdot[\cdot] : (X, L^X)X \qquad \mathscr{P}, \mathscr{C} : (CL^X)CL^X \ .$$

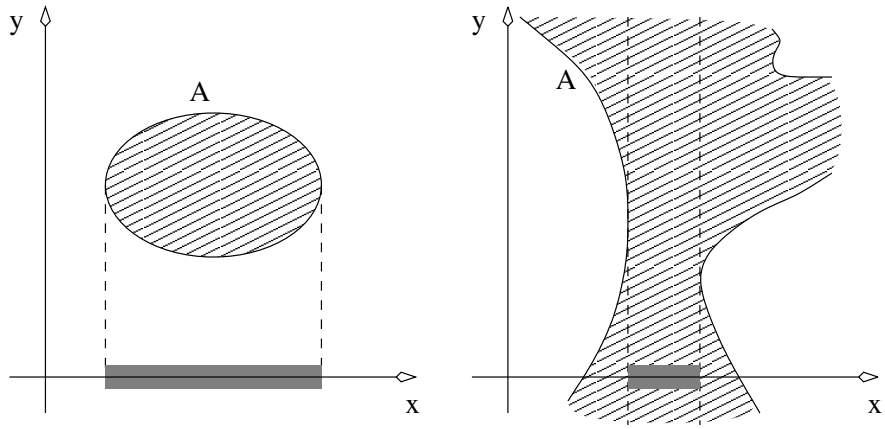
To each predicate symbol p is associated a function symbol $\dot{p}$ of similar arity, which constructs elements of sort CL^X. We call $\mathscr{L}^{\mathsf{ws}}$ this language.

Definition 4 ($\mathcal{T}^{\mathsf{ws}}$). *Let $\mathcal{T}^{\mathsf{ws}}$ be the theory of the language $\mathscr{L}^{\mathsf{ws}}$ formed with the following axioms for explicit substitutions:*

$$\forall x : X, \quad x[nil] = x \qquad (\mathcal{T}_1^{\mathsf{ws}})$$
$$\forall \ell : L^X, \forall x : X, \quad 1[x :: \ell] = x \qquad (\mathcal{T}_2^{\mathsf{ws}})$$
$$\forall \ell : L^X, \forall x, n : X, \quad S(n)[x :: \ell] = n[l] \qquad (\mathcal{T}_3^{\mathsf{ws}})$$
$$\forall \ell : L^X, \forall x_1, \ldots, x_n : X, \quad f(x_1, \ldots, x_n)[\ell] = f(x_1[\ell], \ldots, x_n[\ell]) \qquad (\mathcal{T}_4^{\mathsf{ws}})$$

and the axioms for proposition encoding:

$$\forall \ell : L^X, \forall x_1, \ldots, x_n : X, \quad \ell \in \dot{p}(x_1, \ldots, x_m) \Leftrightarrow p(x_1[\ell], \ldots, x_m[\ell]) \qquad (\mathcal{T}_5^{\mathsf{ws}})$$
$$\forall A, B : CL^X, \forall \ell : L^X, \quad \ell \in A \cap B \Leftrightarrow \ell \in A \wedge \ell \in B \qquad (\mathcal{T}_6^{\mathsf{ws}})$$
$$\forall A, B : CL^X, \forall \ell : L^X, \quad \ell \in A \cup B \Leftrightarrow \ell \in A \vee \ell \in B \qquad (\mathcal{T}_7^{\mathsf{ws}})$$
$$\forall A, B : CL^X, \forall \ell : L^X, \quad \ell \in A \supset B \Leftrightarrow \ell \in A \Rightarrow \ell \in B \qquad (\mathcal{T}_8^{\mathsf{ws}})$$
$$\forall \ell : L^X, \quad \ell \in \varnothing \Leftrightarrow \bot \qquad (\mathcal{T}_9^{\mathsf{ws}})$$
$$\forall A : CL^X, \forall \ell : L^X, \quad \ell \in \mathscr{P}(A) \Leftrightarrow \exists x, x :: \ell \in A \qquad (\mathcal{T}_{10}^{\mathsf{ws}})$$
$$\forall A : CL^X, \forall \ell : L^X, \quad \ell \in \mathscr{C}(A) \Leftrightarrow \forall x, x :: \ell \in A \ . \qquad (\mathcal{T}_{11}^{\mathsf{ws}})$$



(a) Projection (b) Cylindrification

Fig. 1: Two-dimensional operators

A couple of remarks on this formalization:

- The $\mathscr{P}$ and $\mathscr{C}$ operators are called *projection* and *cylindrification* respectively. Figure 1 illustrates the semantics of these combinators in a two-dimensional space.
- Axiom scheme ($\mathcal{T}_4^{\mathsf{ws}}$) (resp. ($\mathcal{T}_5^{\mathsf{ws}}$)) is *finite*, as there is one such axiom for each function (resp. predicate) symbol of arity n (resp. m) in the language $\mathscr{L}^{\mathsf{ws}}$, which is finite.
- This axiom system implements a weak calculus of explicit substitutions [5]: the substitutions are propagated over the elements of the language via the symbols $\in$ and $\cdot[\cdot]$, and no lift is introduced by the $\mathscr{P}$ or $\mathscr{C}$ binders (axioms ($\mathcal{T}_{10}^{\mathsf{ws}}$) and ($\mathcal{T}_{11}^{\mathsf{ws}}$)).

3.2 Expressiveness

We want be able to associate to each proposition a characteristic class, constructed with the symbols exposed in Definition 3.

Example 2. Assume the predicate symbol $<$ and the associate characteristic set constructor $\dot{<}$. Using infix notations and 2 and 3 for respectively $S(1)$ and $S(S(1))$, the class of the x such that $\exists y, \exists z, (x < z \wedge z < y)$ is written $\mathscr{P}\mathscr{P}(3\dot{<}1 \cap 1\dot{<}2)$. Indeed,

$$\begin{aligned}
&\langle x \rangle \in \mathscr{P}\mathscr{P}(3\dot{<}1 \cap 1\dot{<}2) \\
&\quad \Leftrightarrow \exists y, \exists z, (\langle z, y, x \rangle \in 3\dot{<}1 \cap 1\dot{<}2) \\
&\quad \Leftrightarrow \exists y, \exists z, (\langle z, y, x \rangle \in 3\dot{<}1 \wedge \langle z, y, x \rangle \in 1\dot{<}2) \\
&\quad \Leftrightarrow \exists y, \exists z, (x < z \wedge z < y) \ .
\end{aligned}$$

We first need a little lemma to prove that the term substitution axioms are complete:

Lemma 1 (Completeness of substitution). *For all terms u and variables $x_1, \ldots, x_n$ of $\mathscr{L}$, there exists a term t of $\mathscr{L}^{\mathsf{ws}}$ in which none of the x_i appear, such that $u = t[x_1 :: \ldots :: x_n :: nil]$ is provable.*

Proof. We proceed inductively on the structure of u. If u is one of the x_i, then we take $S^{i-1}(1)$ for t and by using axioms $(\mathcal{T}_2^{\mathsf{ws}})$ and $(\mathcal{T}_3^{\mathsf{ws}})$, $x_i = S^{i-1}(1)[x_1 :: \ldots :: x_n :: nil]$ is provable. If u is a variable y different from the x_i, we take $t = S^n(y)$, and by axioms $(\mathcal{T}_1^{\mathsf{ws}})$, $(\mathcal{T}_2^{\mathsf{ws}})$ and $(\mathcal{T}_3^{\mathsf{ws}})$ we can prove $y = S^n(y)[x_1 :: \ldots :: x_n :: nil]$. Finally, if u is a function $f(u_1, \ldots, u_m)$, then by induction hypothesis there exist $t_1, \ldots, t_m$ such that for any $0 < i \leq m$, $u_i = t_i[x_1 :: \ldots :: x_n :: nil]$. Axiom $(\mathcal{T}_4^{\mathsf{ws}})$ allows to conclude $f(u_1, \ldots, u_m) = f(t_1, \ldots, t_m)[x_1 :: \ldots :: x_n :: nil]$. □

We now show that the axiom scheme of comprehension holds in our formalism. Translated in our framework, the scheme states:

Proposition 2. *For any formula P built with the symbols of $\mathscr{L}$, the formula:*

$$\exists E, \forall x_1, \ldots, \forall x_n, (\langle x_1, \ldots, x_n \rangle \in E \Leftrightarrow P)$$

is provable.

Proof. We show by induction on the structure of P that for any formula P there exists a term E of sort CL^X such that, for all $x_1, \ldots, x_n$ of sort X, $\langle x_1, \ldots, x_n \rangle \in E \Leftrightarrow P$.

If P consists in an atomic proposition $p(u_1, \ldots, u_m)$, Lemma 1 allows us to equate this proposition to $p(t_1[l], \ldots, t_m[l])$ where $l = \langle x_1, \ldots, x_n \rangle$. Then by Axiom $(\mathcal{T}_5^{\mathsf{ws}})$, E is $\dot{p}(t_1, \ldots, t_m)$.

For the propositional connectors, using Axioms $(\mathcal{T}_6^{\mathsf{ws}})$ to $(\mathcal{T}_9^{\mathsf{ws}})$, we prove that

$$\begin{aligned} \text{if } P = A \wedge B \text{ then } E &= E_A \cap E_B \\ \text{if } P = A \vee B \text{ then } E &= E_A \cup E_B \\ \text{if } P = \bot \text{ then } E &= \varnothing \\ \text{if } P = A \Rightarrow B \text{ then } E &= E_A \supset E_B \ . \end{aligned}$$

For the case $P = \exists y, A$, the bi-dimensional illustration of Fig. 1a provides the intuition: the class such that there exists a value y for which A holds — and E_A is characterized — is its projection $\mathscr{P}(E_A)$. Indeed, we can derive:

$$\begin{aligned} \langle x_1, \ldots, x_n \rangle \in E &\Leftrightarrow \langle x_1, \ldots, x_n \rangle \in \mathscr{P}(E_A) \\ &\Leftrightarrow \exists y, \langle y, x_1, \ldots, x_n \rangle \in E_A \qquad \text{by axiom } (\mathcal{T}_{10}^{\mathsf{ws}}) \\ &\Leftrightarrow \exists y, A \ . \end{aligned}$$

Similarly for the universal quantifier: if $P = \forall y, A$ then $E = \mathscr{C}(E_A)$. □

Remark that while our theory $\mathcal{T}^{\mathsf{ws}}$ is intuitionistic, this proof unfolds just as well in a classical setting. In this case, we can additionally discard the $\cup$, $\cap$ and $\mathscr{C}$ symbols and related axioms, and to use the De Morgan equivalences to carry out the proof.

Because this entails that $\mathcal{T}^{\mathsf{ws}}$ is an extension of $\mathcal{T}^{\mathsf{cs}}$, and that $\mathcal{T}^{\mathsf{cs}}$ is an extension of $\mathcal{T}$, we have:

Proposition 3 (Extensionality). *$\mathcal{T}^{\mathsf{ws}}$ is an extension of $\mathcal{T}$.*

This property assures that the formalism we proposed is expressive enough, however it should not be too strong: we should not be able to prove propositions that were not provable by using axiom schemes.

Proposition 4 (Conservativity). *$\mathcal{T}^{\mathsf{ws}}$ is a conservative extension of $\mathcal{T}$.*

Proof. We prove this proposition by showing that for each model $\mathcal{M}$ of $\mathcal{T}$ there is a model of $\mathcal{T}^{\mathsf{ws}}$ validating the same $\mathcal{T}$-built formulas.

Let $\mathcal{M}$ be a model of $\mathcal{T}$, and consider the structure $\mathcal{M}'$ defined as follows:

- $[\![X]\!]^{\mathcal{M}'}$ is the set of functions mapping $\mathcal{M}^\omega$ to $\mathcal{M}$, where $\mathcal{M}^\omega$ is the set of infinite lists of elements of $\mathcal{M}$.
- For any function symbol f of $\mathscr{L}$, $[\![f]\!]^{\mathcal{M}'}$, is the function mapping elements $a_1, \ldots, a_n$ of $[\![X]\!]^{\mathcal{M}'}$ to an element b of $[\![X]\!]^{\mathcal{M}'}$ if and only if for all infinite list u, $[\![f]\!]^{\mathcal{M}}(a_1(u), \ldots, a_n(u)) = b(u)$.
- The denotation of a predicate symbol p of $\mathscr{L}$ in $\mathcal{M}'$ is the function mapping elements $a_1, \ldots, a_n$ of $[\![X]\!]^{\mathcal{M}'}$ to 1 if and only if for all infinite list u, $[\![p]\!]^{\mathcal{M}}(a_1(u), \ldots, a_n(u)) = 1$.
- We say that a set e of sequences of elements of $[\![X]\!]^{\mathcal{M}'}$ is definable if and only if there exists a formula P in $\mathscr{L}$ such that the sequence $a_1, \ldots, a_n$ belongs to e if and only if $[\![P]\!]^{\mathcal{M}'}_{a_1/x_1,\ldots,a_n/x_n} = 1$.
- $[\![L^X]\!]^{\mathcal{M}'}$ is the set of finite sequences of elements of $[\![X]\!]^{\mathcal{M}'}$, and $[\![CL^X]\!]^{\mathcal{M}'}$ is the set of definable subsets of $[\![L^X]\!]^{\mathcal{M}'}$.

Note that all we need to define the denotation of a formula P of $\mathscr{L}$ in $\mathcal{M}'$ are $[\![X]\!]^{\mathcal{M}'}$ and the denotation of the symbols to $\mathscr{L}$.

Upon such a basis, we can define denotations for the symbols of $\mathscr{L}^{\mathsf{ws}}$:

- $[\![1]\!]^{\mathcal{M}'}$ is the function that to each infinite list associates its first element, and $[\![S]\!]^{\mathcal{M}'}$ maps a function f to a function g such that $g(u_1, u_2, \ldots) = f(u_2, u_3, \ldots)$.
- The denotation of $t[\ell]$ is the element of $[\![X]\!]^{\mathcal{M}'}$ that associates to any list $u = u_1, u_2, \ldots$ the term $[\![t]\!]^{\mathcal{M}'}(b_1(u), \ldots, b_n(u), u_1, u_2, \ldots)$ where the finite sequence $b_1, \ldots, b_n$ is the denotation of ℓ.
- The denotation of the rest of the symbols of $\mathscr{L}^{\mathsf{ws}}$ is their self-evident counterpart: $\cap$ is set intersection, $\in$ is set membership, etc.

Remark that this definition of definability and the denotation of the symbols of $\mathscr{L}^{\mathsf{ws}}$ make the following sentence a tautology: if e is the set of sequences defined by a proposition P then for all $a_1, \ldots, a_n$,

$$[\![\langle x_1, \ldots, x_n\rangle \in E]\!]^{\mathcal{M}'}_{a_1/x_1,\ldots,a_n/x_n,e/E} = [\![P]\!]^{\mathcal{M}'}_{a_1/x_1,\ldots,a_n/x_n} \quad . \qquad (\star)$$

Indeed, both are interpreted as 1 when $a_1, \ldots, a_n$ belongs to e.

To prove that $\mathcal{M}'$ is a model of $\mathcal{T}^{\mathsf{ws}}$, we check that it validates axioms $(\mathcal{T}_1^{\mathsf{ws}})$ to $(\mathcal{T}_{11}^{\mathsf{ws}})$. But that is not enough, as we need to ensure that the translation we did of the original axiom schemes (e.g. the induction scheme of Example 1) is also valid in $\mathcal{M}'$. Let $s(P(t_1^1, \ldots, t_{n_1}^1), \ldots, P(t_1^p, \ldots, t_{n_p}^p))$ be an axiom scheme of $\mathcal{T}$, and $\forall E, s(\langle t_1^1, \ldots, t_{n_1}^1 \rangle \in E, \ldots, \langle t_1^p, \ldots, t_{n_p}^p \rangle \in E)$ its translation in $\mathcal{T}^{\mathsf{ws}}$. We prove that for any definable set e and sequence $a_1, \ldots, a_n$,

$$[\![s(\langle t_1^1, \ldots, t_{n_1}^1 \rangle \in E, \ldots, \langle t_1^p, \ldots, t_{n_p}^p \rangle \in E)]\!]^{\mathcal{M}'}_{a_1/x_1, \ldots, a_n/x_n, e/E} = 1 \ .$$

Assume e is defined by the proposition Q of $\mathscr{L}$. Since:

$$\begin{aligned}
&[\![s(Q(t_1^1, \ldots, t_{n_1}^1), \ldots, Q(t_1^p, \ldots, t_{n_p}^p))]\!]^{\mathcal{M}'}_{a_1/x_1, \ldots, a_n/x_n} \\
&= [\![s(Q(t_1^1, \ldots, t_{n_1}^1), \ldots, Q(t_1^p, \ldots, t_{n_p}^p))]\!]^{\mathcal{M}}_{a_1/x_1, \ldots, a_n/x_n} \\
&= 1 \ ,
\end{aligned}$$

then all we need to do is to prove:

$$\begin{aligned}
&[\![s(\langle t_1^1, \ldots, t_{n_1}^1 \rangle \in E, \ldots, \langle t_1^p, \ldots, t_{n_p}^p \rangle \in E)]\!]^{\mathcal{M}'}_{a_1/x_1, \ldots, a_n/x_n, e/E} \\
&= [\![s(Q(t_1^1, \ldots, t_{n_1}^1), \ldots, Q(t_1^p, \ldots, t_{n_p}^p))]\!]^{\mathcal{M}'}_{a_1/x_1, \ldots, a_n/x_n} \ .
\end{aligned}$$

But this is simply a consequence of $(\star)$. Hence $\mathcal{M}'$ is a model of $\mathcal{T}^{\mathsf{ws}}$.

Finally, a formula of the language of $\mathscr{L}$ has, obviously, the same denotation in $\mathcal{M}$ and in $\mathcal{M}'$. Thus we can conclude to the conservativity of $\mathcal{T}^{\mathsf{ws}}$ over $\mathcal{T}$. □

This last proof is classical in essence. However it can be extended to an intuitionistic proof by using Heyting algebras instead of classical models.

4 Applications

The result of Propositions 3 and 4 can be applied to theories that use all kinds of axiom schemes. For instance, Zermelo's set theory accepts a conservative extension, built by applying these propositions to the traditional formulation of the theory. The same holds for the binary replacement axiom scheme of Zermelo-Fraenkel's theory, or the three schemes that result from Dowek and Miquel's encoding of set theory in a theory of pointed graphs [6]. We detail two examples: arithmetic and real analysis.

4.1 A Finite Theory of Arithmetic

In the following, we will explore Heyting's arithmetic. While our formalism applies to the original formulation of the theory, HA, we consider here a slightly more elaborate presentation of the theory where the universe of discourse is not restricted to natural numbers. This theory, called HA_N, was presented in [7] by Dowek and Werner.

Definition 5 (HA_N). *The theory HA_N of arithmetic is defined in first-order logic using the symbols 0, Succ, +, ×, Pred, =, Null and N. It counts the axioms:*

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

the axioms for equality:

$$\begin{array}{c} \forall x, (x = x) \\ \forall x, \forall y, (x = y \Rightarrow P(x) \Rightarrow P(y)) \end{array}$$

and the induction scheme:

$$P(0/x) \Rightarrow \forall y, (P(y/x) \Rightarrow P(Succ(y)/x)) \Rightarrow \forall n, (N(n) \Rightarrow P(n/x)) \ .$$

[7] defines a translation $|\cdot|$ between the languages of HA and HA_N, and proves:

Proposition 5. *HA_N is a conservative extension of HA in the sense that if A is a closed proposition formed in the language of HA then A is provable in HA if and only if $|A|$ is provable in HA_N.*

Finitizing the presentation of these axiom schemes is achieved by introducing lists and classes and a set of axioms that allows to express comprehension, as per Sect. 3.

Definition 6 ($\mathsf{HA}_N^{\mathsf{ws}}$). *Define $\mathsf{HA}_N^{\mathsf{ws}}$ as an extension of HA_N, composed of the ranked signature:*

$$\begin{array}{rr} 0, 1 : X & S, Succ, Pred : (X)X \\ +, \times : (X, X)X & = : (X, X) \\ N, Null : (X) & \cdot[\cdot] : (X, L^X)X \\ \varnothing : CL^X & \mathscr{P}, \mathscr{C} : (CL^X)\, CL^X \\ \cap, \cup, \supset : (CL^X, CL^X)\, CL^X & \in : (L^X, CL^X) \ . \end{array}$$

axioms ($\mathcal{T}_1^{\mathsf{ws}}$) to ($\mathcal{T}_{11}^{\mathsf{ws}}$), the axioms of arithmetic:

$$\begin{array}{rr} N(0) & \forall x,(N(x) \Rightarrow N(Succ(x))) \\ Pred(0) = 0 & \forall x,(Pred(Succ(x)) = x) \\ Null(0) & \forall x,(\neg Null(Succ(x))) \\ \forall y,(0 + y = y) & \forall x, \forall y,(Succ(x) + y = Succ(x + y)) \\ \forall y,(0 \times y = 0) & \forall y,(Succ(x) \times y = x \times y + y) \end{array}$$

the equality axioms:

$$\forall x, (x = x)$$
$$\forall x, \forall y, (x = y \Rightarrow \forall A, (\langle x \rangle \in A \Rightarrow \langle y \rangle \in A))$$

and the induction axiom:

$$\forall n, \forall A, (\langle 0 \rangle \in A \Rightarrow \forall y, (\langle y \rangle \in A \Rightarrow \langle Succ(y) \rangle \in A) \Rightarrow \forall n, (N(n) \Rightarrow \langle n \rangle \in A)) \ .$$

In particular, axiom ($\mathcal{T}_4^{\mathsf{ws}}$) has four instances (*Succ*, *Pred*, + and ×) and axiom ($\mathcal{T}_5^{\mathsf{ws}}$) three (=, N and *Null*), for a total of 29 axioms. Propositions 3 and 4 applied to HA_N, composed with Proposition 5, allow us to state:

Proposition 6. $\mathsf{HA}_N^{\mathsf{ws}}$ *is as a conservative extension of* HA.

We can define a slight variant of $\mathsf{HA}_N^{\mathsf{ws}}$ by replacing the class induction axiom and Leibniz's equality axiom by the equivalences:

$$\forall x, \forall y, (x = y \Leftrightarrow \forall A, (\langle x \rangle \in A \Rightarrow \langle y \rangle \in A))$$
$$\forall n, (N(n) \Leftrightarrow \forall A, (\langle 0 \rangle \in A \Rightarrow \forall y, (\langle y \rangle \in A \Rightarrow \langle Succ(y) \rangle \in A) \Rightarrow \forall n, \langle n \rangle \in A)) \ .$$

And we can drop the three axioms $\forall x, (x = x)$, $N(0)$ and $\forall x, (N(x) \Rightarrow N(Succ(x)))$ that have become superfluous.

Definition 7 (HA^+)**.** *Let* HA^+ *be this slimmed-down theory.*

HA^+ is equivalent to $\mathsf{HA}_N^{\mathsf{ws}}$, and counts 26 axioms instead of 29.

4.2 Arithmetic as a Theory Modulo

A theory modulo is a theory in which formulas are identified modulo a congruence [8]. In particular, the thoery arithmetic has been expressed in such a framework [7], but this formalization had an infinite number of rewrite rules. The goal of this section is to show how our system allows a finite formulation of arithmetic modulo.

Definition 8 ($\mathsf{HA}^{\mathsf{mod}}$)**.** *The language of the theory* $\mathsf{HA}^{\mathsf{mod}}$ *is the same as the theory* HA^+*. The congruence* $\equiv_{\mathcal{R}}$ *associated with this theory is given by the rewrite system* $\mathcal{R}$ *of Table 1.*

The system is parted between rules dealing with de Bruijn indices, rules for arithmetic operations and rules defining relations (equality, etc.). This formalism counts a total of 26 rules, which is reduced to 22 or 23 in classical logic using the fact that all the connectors and quantifiers can be defined from 2 or 3 primitive ones.

Table 1: Rewrite system $\mathcal{R}$ for arithmetic

Substitutions rules

$$t[nil] \to t \qquad (\mathsf{HA}_1^{\mathsf{mod}})$$
$$1[t :: \ell] \to t \qquad (\mathsf{HA}_2^{\mathsf{mod}})$$
$$S(n)[t :: \ell] \to n[\ell] \qquad (\mathsf{HA}_3^{\mathsf{mod}})$$
$$Succ(t)[\ell] \to Succ(t[\ell]) \qquad (\mathsf{HA}_4^{\mathsf{mod}})$$
$$Pred(t)[\ell] \to Pred(t[\ell]) \qquad (\mathsf{HA}_5^{\mathsf{mod}})$$
$$(t_1 + t_2)[\ell] \to t_1[\ell] + t_2[\ell] \qquad (\mathsf{HA}_6^{\mathsf{mod}})$$
$$(t_1 \times t_2)[\ell] \to t_1[\ell] \times t_2[\ell] \qquad (\mathsf{HA}_7^{\mathsf{mod}})$$

Arithmetic rules

$$Pred(0) \to 0 \qquad (\mathsf{HA}_8^{\mathsf{mod}})$$
$$Pred(Succ(x)) \to x \qquad (\mathsf{HA}_9^{\mathsf{mod}})$$
$$0 + y \to y \qquad (\mathsf{HA}_{10}^{\mathsf{mod}})$$
$$0 \times y \to 0 \qquad (\mathsf{HA}_{11}^{\mathsf{mod}})$$
$$Succ(x) + y \to Succ(x + y) \qquad (\mathsf{HA}_{12}^{\mathsf{mod}})$$
$$Succ(x) \times y \to x \times y + y \qquad (\mathsf{HA}_{13}^{\mathsf{mod}})$$

Relation rules

$$\ell \in \dot{Null}(t) \to Null(t[\ell]) \qquad (\mathsf{HA}_{14}^{\mathsf{mod}})$$
$$\ell \in \doteq(t_1, t_2) \to t_1[\ell] = t_2[\ell] \qquad (\mathsf{HA}_{15}^{\mathsf{mod}})$$
$$\ell \in \dot{N}(t) \to N(t[\ell]) \qquad (\mathsf{HA}_{16}^{\mathsf{mod}})$$
$$x = y \to \forall A, (\langle x \rangle \in A \Rightarrow \langle y \rangle \in A)) \qquad (\mathsf{HA}_{17}^{\mathsf{mod}})$$
$$N(n) \to \forall A, (\langle 0 \rangle \in A \Rightarrow \forall y, (\langle y \rangle \in A \Rightarrow \langle Succ(y) \rangle \in A) \Rightarrow \langle n \rangle \in A) \qquad (\mathsf{HA}_{18}^{\mathsf{mod}})$$
$$Null(0) \to \top \qquad (\mathsf{HA}_{19}^{\mathsf{mod}})$$
$$Null(Succ(x)) \to \bot \qquad (\mathsf{HA}_{20}^{\mathsf{mod}})$$
$$\ell \in A \cap B \to \ell \in A \wedge \ell \in B \qquad (\mathsf{HA}_{21}^{\mathsf{mod}})$$
$$\ell \in A \cup B \to \ell \in A \vee \ell \in B \qquad (\mathsf{HA}_{22}^{\mathsf{mod}})$$
$$\ell \in A \supset B \to \ell \in A \Rightarrow \ell \in B \qquad (\mathsf{HA}_{23}^{\mathsf{mod}})$$
$$\ell \in \varnothing \to \bot \qquad (\mathsf{HA}_{24}^{\mathsf{mod}})$$
$$\ell \in \mathscr{P}(A) \to \exists n, n :: \ell \in A \qquad (\mathsf{HA}_{25}^{\mathsf{mod}})$$
$$\ell \in \mathscr{C}(A) \to \forall n, n :: \ell \in A \ . \qquad (\mathsf{HA}_{26}^{\mathsf{mod}})$$

4.3 Application to Real Analysis

The applications of real analysis in computer science are too numerous to cite here. Real numbers are used, e.g., in exact arithmetic, programming languages, formal calculus and formal systems.The following formal development is quite common [9], and is used e.g. in the proof assistant Coq to implement the theory of real numbers.

Definition 9 ($\mathbb{R}^{\mathsf{cs}}$). *The language of the theory of real numbers $\mathbb{R}^{\mathsf{cs}}$ is formed by the symbols 0, 1, $+$, $\times$, the opposite $-$, inverse $1/\cdot$, the symbol $\lceil\cdot\rceil$ that maps real numbers to natural numbers, and the predicates $<$ and $=$. We note $\leq$ the disjunction of the two aforementioned predicates. The axioms follow:*

$$(1 = 0) \Rightarrow \bot \tag{$\mathbb{R}_1$}$$
$$\forall x, \forall y, \quad x + y = y + x \tag{$\mathbb{R}_2$}$$
$$\forall x, \forall y, \forall z, \quad (x + y) + z = x + (y + z) \tag{$\mathbb{R}_3$}$$
$$\forall x, \quad x + (-x) = 0 \tag{$\mathbb{R}_4$}$$
$$\forall x, \quad x + 0 = x \tag{$\mathbb{R}_5$}$$
$$\forall x, \forall y, \quad x \times y = y \times x \tag{$\mathbb{R}_6$}$$
$$\forall x, \forall y, \forall z, \quad (x \times y) \times z = x \times (y \times z) \tag{$\mathbb{R}_7$}$$
$$\forall x, \quad ((x = 0) \Rightarrow \bot) \Rightarrow (1/r) \times r = 1 \tag{$\mathbb{R}_8$}$$
$$\forall x, \quad 1 \times x = x \tag{$\mathbb{R}_9$}$$
$$\forall x, \forall y, \forall z, \quad x \times (y + z) = x \times y + x \times z \tag{$\mathbb{R}_{10}$}$$
$$\forall x, \forall y, \quad (x < y) \vee (x = y) \vee (y < x) \tag{$\mathbb{R}_{11}$}$$
$$\forall x, \forall y, \quad x < y \Rightarrow y < x \Rightarrow \bot \tag{$\mathbb{R}_{12}$}$$
$$\forall x, \forall y, \forall z, \quad x < y \Rightarrow y < z \Rightarrow x < z \tag{$\mathbb{R}_{13}$}$$
$$\forall x, \forall y, \forall z, \quad y < z \Rightarrow x + y < x + z \tag{$\mathbb{R}_{14}$}$$
$$\forall x, \forall y, \forall z, \quad 0 < x \Rightarrow y < z \Rightarrow x \times y < x \times z \tag{$\mathbb{R}_{15}$}$$
$$\forall x, \quad x < \lceil x \rceil \wedge (\lceil x \rceil + (-x) \leq 1) \ . \tag{$\mathbb{R}_{16}$}$$

One way of formulating the completeness theorem of real analysis is to use classes and bounds. Thus we consider classes of reals, manipulated using the nil, $::$ and $\in$ symbols, and the axiom scheme of comprehension:

$$\forall E, \forall x_1, \ldots, x_n, (\langle x_1, \ldots, x_n \rangle \in E \Leftrightarrow P) \ . \tag{$\mathbb{R}_{17}$}$$

The last four axioms of this theory follow: the first three define the semantics of the predicate symbols $isUB(\cdot,\cdot)$, $bounded(\cdot)$ and $isLUB(\cdot,\cdot)$; the fourth is the axiom of completeness.

$$\forall E, \forall m, (\exists x, \langle x \rangle \in E \Rightarrow x \leq m) \Leftrightarrow isUB(E, m) \tag{$\mathbb{R}_{18}$}$$
$$\forall E, (\exists m, isUB(E, m)) \Leftrightarrow bounded(E)) \tag{$\mathbb{R}_{19}$}$$
$$\forall E, \forall m, (\exists x, isUB(E, m) \wedge (\forall b, isUB(E, b) \Rightarrow m \leq b)) \Leftrightarrow isLUB(E, m) \tag{$\mathbb{R}_{20}$}$$
$$\forall E, (bounded(E) \Rightarrow (\exists x, \langle x \rangle \in E) \Leftrightarrow (\exists m, isLUB(E, m))) \ . \tag{$\mathbb{R}_{21}$}$$

Following the protocol defined in Sect. 3, we give a finite first-order presentation of this theory.

Definition 10 ($\mathbb{R}^{\mathsf{ws}}$). *Define* $\mathbb{R}^{\mathsf{ws}}$ *as an extension of* $\mathbb{R}^{\mathsf{cs}}$*, formed with the ranked signature:*

$$\begin{aligned} 0, 1 &: X & S, -, 1/\cdot &: (X)X \\ +, \times &: (X, X)X & <, = &: (X, X) \\ \lceil\cdot\rceil &: (X)X & \cdot[\cdot] &: (X, L^X)X \\ \varnothing &: CL^X & \mathscr{P}, \mathscr{C} &: (CL^X)\,CL^X \\ \cap, \cup, \supset &: (CL^X, CL^X)\,CL^X & \in &: (L^X, CL^X) \ . \end{aligned}$$

axioms $(\mathcal{T}_1^{\mathsf{ws}})$ *to* $(\mathcal{T}_{11}^{\mathsf{ws}})$*, the axioms* $(\mathbb{R}_1)$ *to* $(\mathbb{R}_{16})$ *and* $(\mathbb{R}_{18})$ *to* $(\mathbb{R}_{21})$*.*

In particular, axiom $(\mathcal{T}_4^{\mathsf{ws}})$ has six instances (S, $-$, $1/\cdot$, $+$, $\times$ and $\lceil\cdot\rceil$) and axiom $(\mathcal{T}_5^{\mathsf{ws}})$ two ($=$ and $<$), for a total of 37 axioms. Propositions 3 and 4 applied to $\mathbb{R}$ allow us to state:

Proposition 7. $\mathbb{R}^{\mathsf{ws}}$ *is as a conservative extension of* $\mathbb{R}^{\mathsf{cs}}$*.*

Remark that in the ranked signature of $\mathbb{R}^{\mathsf{ws}}$, $\lceil\cdot\rceil$ has the (potentially faulty) rank $(X)X$. This is because there is no notion of natural numbers in the formalism of Definition 10. This can be rectified by introducing the appropriate sort X', and the language and theory of natural arithmetic as in the previous section; then writing $\lceil\cdot\rceil : (X)X'$. However a more lightweight way of solving this issue is to emulate natural numbers within the sort of real numbers X. Indeed, we can define N as the smallest class of real numbers that satisfy the conjunction of the formulas:

$$0 \in N \qquad\qquad \forall x : \mathbb{R}, x \in N \Rightarrow x + 1 \in N \ .$$

Now the signature of $\lceil\cdot\rceil$ would still read $(X)X$, however the semantics of the operator would restrict its values to elements of the class N.

5 Related Work

The work we present in the previous sections is related to von Neumann, Bernays and Gödel's formalism for set theory (NBG) [1] that rehabilitated the notion of class used by 19th century mathematicians [10]. However it improves on a couple of points:

- Classes and the NBG approach have always been associated with set theory [10]. We generalize it to any theory that has axiom schemes.
- By clarifying the classes/set distinction, not only is the system simplified, but we also allow a more structured hierarchy of objects. In $\mathcal{T}^{\mathsf{ws}}$ the sorts X, L^X and CL^X are clearly separate entities, while in NBG the sorts of objects and classes are indistinctly embedded into one another.

– Using lists and explicit substitutions to instantiate predicate free variables also greatly clarifies the argument-passing process, and allows us to bypass a couple of permutation axioms. Also, because we use native lists we are spared the tedious process of re-encoding them using sets, as is done in NBG.
– There is no easy way to orient NBG's permutation axioms to generate a well-behaved rewrite system. On the contrary, and as demonstrated in Section 4.2, the rules $(\mathcal{T}_1^{\mathsf{ws}})$ to $(\mathcal{T}_{11}^{\mathsf{ws}})$ are easily orientable.

What is more, our formalism applied to Zermelo's set theory would use 15 axioms (in the classical case) vs. 14 for NBG, thus we feel it allows for a more understandable presentation of set theory without being overly bloated.

Vaillant [2] gives a presentation of set theory using explicit substitutions (in the form of the $\lambda\sigma$-calculus) to manipulate classes. The axiomatization we propose differs from it in the following ways:

– While Vaillant's paper, following NBG, was focused on set theory, our method applies to any theory with one or more axiom schemes.
– We have shown that a weak substitution calculus is strong enough to allow the complete manipulation of substitutions in this type of framework. This allows us to greatly reduce both the language's signature (neither lift, shift nor compose operators) and the number of axioms in our presentation.

Finally, if a comparison of the size of the two formalizations could reinforce the reader's opinion that our system is lighter, consider that Vaillant's theory uses a total of 42 axioms, while ours would only require 18 and still express full-blown Zermelo set theory.

6 Conclusion

We have exposed a generic formalization of theories with axiom schemes, which has the property of being finite. This was achieved through the use of classes and the recourse to weak explicit substitutions to cope straightforwardly with variable instantiation. This operating protocol was applied to give a finite axiomatization of the theories of arithmetic and of real analysis, and a finite formalization of the former in deduction modulo.

Comparing to other methods such as von Neumann, Bernays and Gödel's or Vaillant's, it appears this way of formalizing theories using axiom schemes has links with both worlds. The use of a weak calculus allows us to keep a reduced number of axioms, and provides an intuitive, direct mechanism for substitutions — all of which are highly desirable properties in a proof checking environment.

Such a result easily fits into the trend set by the previous works done to formalize arithmetic and set theory into computer proof assistants [11–14, 6]. Moreover, the fact that our axioms are easily orientable is a major asset when dealing with theories in deduction modulo, which is a rapidly growing research topic showing strong potential. An implementation of the theory of real numbers into the first-order proof manager Fellowship [3] using this technique is currently underway. We will then be able to tell if this attempt at a better understanding of class theory leads to simpler real-world implementations.

References

1. Mendelson, E.: Introduction to mathematical logic (4th ed.). Chapman & Hall (1997)
2. Vaillant, S.: A finite first-order presentation of set theory. In: TYPES. (2002) 316–330
3. Kirchner, F.: Fellowship: who needs a manual anyway? (2005)
4. Ridge, T., Margetson, J.: A mechanically verified, sound and complete theorem prover for first order logic. In Hurd, J., Melham, T., eds.: TPHOLs. Volume 3603 of Lecture Notes in Computer Science., Springer (2005) 294–309
5. Hardin, T., Maranget, L., Pagano, B.: Functional back-ends within the lambda-sigma calculus. In: Proc. of the 1996 International Conference on Functional Programming, ACM Press (1996)
6. Dowek, G., Miquel, A.: Cut elimination for Zermelo's set theory. Submitted to RTA 2006 (2006)
7. Dowek, G., Werner, B.: Arithmetic as a theory modulo. In: RTA. (2005) 423–437
8. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo, revised version. Rapport de Recherche 4861, Institut National de Recherche en Informatique et en Automatique (2003)
9. Lelong-Ferrand, J., Arnaudies, J.M.: Cours de Mathématiques. Tome 2 : Analyse. Dunod (1972)
10. Bourbaki, N.: Éléments de mathématique – Théorie des ensembles. Volume 1 à 4. Masson, Paris (1968)
11. Belinfante, J.: Computer proofs in Gödel's class theory with equational definitions for composite and cross. Journal of Automated Reasoning (1999)
12. Boyer, R., Lusk, E., McCune, W., Overbeek, R., Stickel, M., Wos, L.: Set theory in first-order logic: Clauses for Gödel's axioms. Journal of Automated Reasoning (1986)
13. Paulson, L.: Set theory for verification: I. From foundations to functions. Journal of Automated Reasoning **11**(3) (1993) 353–389
14. Quaife, A.: Automated deduction in von Neumann-Bernays-Gödel set theory. Journal of Automated Reasoning (1992)