Implementation of Bourbaki's Elements of Mathematics in Coq: Part One, Theory of Sets

JOSÉ GRIMM
Inria Sophia-Antipolis, Apics Team
Email: Jose.Grimm@inria.fr

This paper presents a formalization of the first book of the series "Elements of Mathematics" by Nicolas Bourbaki, using the Coq proof assistant. It discusses formalization of mathematics, and explains in which sense a computer proof of a statement corresponds to a proof in the Bourbaki sense, given that the Coq quantifiers are not defined in terms of Hilbert's ϵ function. The list of axioms and axiom schemes of Bourbaki is compared to the more usual Zermelo-Fraenkel theory, and to those proposed by Carlos Simpson, which form the basis of the Gaia software. Some basic constructions (union, intersection, product, function, equivalence and order relation) are described, as well as some properties; this corresponds to Sections 1 to 6 of Chapter II, and the first two sections of Chapter III. A commented proof of Zermelo's theorem is also given. The code (including almost all exercises) is available on the Web, under http://www-sop.inria.fr/apics/gaia.

1. INTRODUCTION

1.1 Normal Form of One in Bourbaki

When I was young, I was intrigued by the following quote of Bourbaki [Bou68, p. 158]: The mathematical *term denoted* (Chapter 1, § 1, no. 1) by the symbol "1" is of course not to be confused with the *word* "one" in ordinary language. The term denoted by "1" is equal, by virtue of the definition above, to the term denoted by the symbol

$$\tau_Z((\exists u)(\exists U)(u = (U, \{\emptyset\}, Z) \text{ and } U \subset \{\emptyset\} \times Z \text{ and } \cdots)) \qquad (*)$$

As a rough estimate, the term so *denoted* is an assembly of several tens of thousands of signs (each of which is one of τ, $\square$, $\vee$, $\neg$, $=$, $\in$, $\supset$).

This assembly is too big to be written on a sheet of paper, but computing its size presents no difficulty; one can check that the size of $\emptyset$ is 12, the size of $\{\emptyset\}$ is 513, the size of $U \subset \{\emptyset\} \times Z$ is 3 000 000, etc. Thus, Bourbaki underestimated the size which can be shown to be of the order of 10^{15}. Some years later, I used a computer algebra system to get the exact number (see [Gri09a] for details); later on I discovered that the English edition of Bourbaki (1968) differs from the French one (1970, [Bou70]) by the removal of the sign $\supset$, that constructs the ordered pair (x, y), and the axiom A3 that governs the use of this symbol; a new definition has been added, namely $(x, y) = \{\{x\}, \{x, y\}\}$. This naturally increases somehow the

size of "1", which becomes

$$573306704401798033758237640367224116154353941968147665929668 9.$$

In [Mat02], the author says "$\{x\}$ is the term $\tau_y \forall z(z \in y \Leftrightarrow z = x)$ slightly simplified from the actual definition as $\{x, x\}$". The actual term is $\tau_y \forall z(z \in y \Leftrightarrow z = x \text{ or } z = x)$. With this definition $\{\emptyset\}$ has size 217, and "1" is of size 4.10^{12} (a thousand times smaller than our computations). He has also computed the size of "1" for the 1970 edition: it is smaller by a factor 10^9 to our estimation. This shows that a slight change in one definition may induce a huge change in subsequent constructions.

1.2 Outline of this Paper

In the first section we discuss formalization of mathematics in general, and that of Bourbaki in particular. In the second section we describe the list of axioms and axiom schemes of Bourbaki, compare them to the more usual Zermelo-Fraenkel theory, and introduce and discuss the axioms of Gaia, as designed by Carlos Simpson. The third section explains the basic constructions (union, intersection, product, functions); it corresponds to Sections 1 to 5 of Chapter II. It is followed by the description of equivalence and order relations (corresponding to Sections II.6, III.1 and III.2). Finally, we comment the proof of Zermelo's theorem.

1.3 Formalization of Mathematics

The discovery of paradoxes at the start of the twentieth century (Burali-Forti, Russell and others) obliged mathematicians and logicians to clarify some fundamental points in their theories; specifying the universe of discourse gave raise to Russell's theory of types, the theory of categories, as opposed to the simple theory of sets; on the other hand, intuitionists like Brouwer started to reject the principle of excluded middle, while other introduced the axioms of choice, of replacement, of foundation, or of anti-foundation. This led to different theories, in which one can formally decide that some proposition is true, with the disadvantage that there are formally undecidable propositions (Gödel).

People started to construct catalogs of formally proved statements. One example is the series "Elements of mathematics" of Bourbaki, that considers classical logic and the theory of sets. Some years later, people designed computer programs that check the formal proofs. One of these systems is Mizar[1], it is based on the Tarski-Grothendieck set theory. The catalog of all theorems (which has 50000 entries) is published in the (Journal of) Formalized Mathematics. As an example a proof of Zermelo's theorem (Theorem 1 of Bourbaki E.III.2.3) can be found in [NB89]; some properties of complete lattices (Exercise 17 of Bourbaki E.III.1) can be found in [Ban92]. However, given a theorem of Bourbaki, there is no easy way to find its equivalent in Mizar, or in any other proof assistant, that come with their own catalogs, and their own priorities (often, to proven hard theorems like the Kepler conjecture or the Feit-Thompson theorem).

[1] http://mizar.org/

Our word is based on Coq (see [BC04]), that uses intuitionist logic (but provides the excluded middle axiom in a module). It is based on the Calculus of Inductive Constructions. Each object has a type; in particular a type has a type, which is called a *sort*. There are two basic sorts Set, and Prop, that are sub-sorts of Type. If a is of type A and A of type Prop, then A is called a proposition and a is called a proof of A. The quantity $A \to B$ is the type of functions from A to B. If A and B are propositions and f is of type $A \to B$, then f maps a proof of A to a proof of B. Thus f is a proof of $A \Rightarrow B$. This is called the Curry-Howard isomorphism, and explains why implication is sometimes denoted by $\to$. The other main feature of Coq is that one can define inductive functions and objects. If one defines nat as the set of successors of zero, the associated induction principles (checking a property, defining a function) come for free. This feature of Coq is rarely used.

1.4 The GAIA Project

In Spring 2009, we started a collaboration with Alban Quadrat, called GAIA (Geometry, Algebra, Informatics and Applications)[2]. The long term objective was to formally prove the theorems presented in the HDR dissertation of A. Quadrat, implement part of the book of [Osb00] on Homological Algebra, and to show correctness of the implementation of these theorems as algorithms in the OreModules package[3]. We started to implement the "Elements of Mathematics" of Bourbaki into a computer, using Coq as proof assistant, and the work of Carlos Simpson as basis[4]. We started with Book I (theory of sets), a part of Book II (Algebra), and a part of Book III (Topology). We will describe in the paper our implementation of the theory of sets, functions, relations, and in a follow-up the theory of cardinals, from which one deduces the set $\mathbf{N}$ of natural integer (i.e., finite numbers). Implementing $\mathbf{Z}$ and $\mathbf{Q}$ is straightforward. Further details can be found in [Gri09b], [Gri09a] and the GAIA Web page.

1.5 Bourbaki's Elements of Mathematics

As mentioned above, the intent of the series "Elements of mathematics" was to give a rather exhaustive list of theorems, using coherent notations, and each book is followed by a summary of results. There are nevertheless some mistakes. For instance the formula (*) is wrong (injectivity of u is missing). The book contains some other small errors in the statements (for instance Bourbaki pretends that a finite ordered set has a maximal element, which is true only if the set is non-empty) or in the proofs (The English edition has $1 \leq i \leq j < n$ instead of $1 \leq i < j \leq n$ on page 182, it has an upper case X instead of a lower case x in the proof of the criteria C26 and C27 page 37). There are some differences between the English edition of 1968 and the French version of 1970, [Bou70]. For instance the definition of "partition" is not the same (in one case, all the sets are assumed non-empty), so that some theorems might be ambiguous.

[2] http://www-sop.inria.fr/apics/gaia
[3] http://wwwb.math.rwth-aachen.de/OreModules/
[4] http://math.unice.fr/~carlos/themes/verif.html

We are convinced that it is possible to prove all statements of this series, using a proof assistant such as Coq, using essentially the same arguments as those presented by Bourbaki, thanks to its rigour. The main drawback is that, before proving any theorem using real numbers, one has to implement 200 pages of the theory of sets (in order to get the set $\mathbf{N}$), 120 pages of algebra (in order to get the set $\mathbf{Q}$) and 300 pages of topology (in order to get the set $\mathbf{R}$); this paper presents half of the first 200 pages. The second drawback is the weird Bourbaki logic (the axiom of choice is implicit, the axiom of replacement is complicated). However, starting from Chapter III, there is hardly any reference to the symbol τ, and the replacement axiom is not used any more. In other terms, all these theorems can be proved in ZFC, using classical logic.

The main conclusion of our work is that it is *possible* to follow Bourbaki, but it is *more efficient* to define $\mathbf{N}$ and $\mathbf{R}$ via axioms, and in parallel, prove that these axioms are in fact theorems (in other terms: there is a model of these axioms). For instance, in the standard Coq library, an axiom asserts existence of a complete totally ordered archimedian field, where "complete" means that every non-empty bounded set has a least upper bound (rather than its topological equivalent: every Cauchy sequence converges). Such a definition is independent of algebra and topology, it requires only the theory of ordered sets.

1.6 The Formalization of Bourbaki

Chapter I of the theory of sets explains how, theoretically, one could be absolutely certain that all statements of the book are true. This is a two-step procedure. To every object x is associated a formal value $f(x)$ and a normal form $n(f(x))$. Consider for instance the statement: every cardinal x satisfies $x \geq 0$. The formal value of the statement would be $\forall x, \mathrm{isCard}(x) \Rightarrow x \geq_{\mathbf{N}} 0_{\mathbf{N}}$ (essentially, this disambiguates 0 and $\geq$). Consider the following proof of the statement "follows trivially from $\emptyset \subset X$". Its formal value is the list of formal values of the statements, after filling the holes (one has to say that $\emptyset \subset X$ implies $\mathrm{Card}(\emptyset) \leq \mathrm{Card}(X)$, that $\mathrm{Card}(\emptyset)$ is zero and that, since x is a cardinal, it has the form $\mathrm{Card}(X)$ for some X). Here is an example of formal value of the statement and its proof.

```
Lemma zero_smallest: forall x, is_cardinal x -> cardinal_le card_zero x.
Proof.
move=> x cx; rewrite - cardinal_emptyset - (cardinal_of_cardinal cx).
apply sub_smaller; apply emptyset_sub_any.
Qed.
```

The second step can be fully automatized, so that implementing Bourbaki in Coq (or any other book in any other proof assistant) consists in finding a formal value $f(x)$ for every theorem and proof. The non-trivial point here is "filling the holes". The proof assistant might be intelligent enough to convert the conclusion of the theorem into $\mathrm{Card}(\emptyset) \leq \mathrm{Card}(X)$, prove the hint $\emptyset \subset X$, and look at its data base, and notice that the theorem sub_smaller applied to the hint solves the goal.

The normal form of x is some quantity $n(x)$ such that, if unfolding some abbreviations in x yields x', then $n(x) = n(x')$. The same holds if x' is obtained

from x by removing superfluous parentheses (the normal form of Bourbaki uses no parentheses at all). Two otherwise distinct objects have distinct normal forms. In the Bourbaki framework $(\forall x)(\forall y)(x \in y)$ and $(\forall y)(\forall x)(y \in x)$ are "identical" so have the same normal form: it is obtained by replacing any bound variable (x or y) by the same symbol $\square$, and linking it to the $\forall$ (in fact, to the τ that results from unfolding the definition of $\forall$). Links are hard to formalize, but can be done in the following way. Let A be the alphabet (it is the finite set formed of τ, $\square$, $\vee$, $\neg$, $=$, $\in$, $\supset$, and the letters). An assembly without links is a word over A (an element of A^n for some n). The i-th character of a word w will be denoted by w_i. A link is a pair of integers (a, b) (it links w_a to w_b). One of the characters w_a, w_b is τ, the other one is $\square$. Since the square must be in the scope of τ, we get $a < b$. Since each square is linked to exactly one τ, an assembly of size n has at most n links. The links can be ordered by increasing value of the second component. This gives the normal form. (Since the set of finite lists of pairs of integers is a countable set, one could encode the lists of links by a single integer). It is algorithmically possible to decide whether or not a word w, together with a finite list of pairs on integers is an assembly-with-links associated to a valid formula, say $(\forall x)(\forall y)(x \in y)$. Note that Bourbaki cannot use this definition, since it presupposes existence of integers. Thus, he introduces a definition, that cannot be used in practice. There are some alternatives: for every link (a, b), the de Bruijn index of the bound variable w_b is the number of positions c such that $a < b < c$ and w_c is a τ. One can replace a by the de Bruijn index of b. One can avoid using integers by replacing each bound variable by a square followed by some prime signs (none if the index is zero, one if the index is 1, two if the index is zero, etc).

A theorem checker is an algorithm that takes as argument a sequence n_1, n_2, etc. It has a list of true statements a_1, a_2, etc, and adds each n_i to the list after checking. The easy case is when $n_i = a_j$ for some j. The next case is when some a_j is the normal form of $f(x)$, for some expression f that depends on one or more parameters x, and there is an instance X of the parameter such that the normal form of $f(X)$ is n_i. There might be conditions on X. A typical case is the syllogism: there are two parameters A and B, that have to be true, and moreover B must be of the form $A \Rightarrow C$, case where $f(X)$ is C. In this case n_i is accepted if there are a_n and a_m such that for some A, $a_n = n(A)$ for some B, $a_m = n(A \Rightarrow B)$ and $n_i = n(B)$. An *annotated proof* is a sequence n_i as above together with annotations that explain how each n_i has to be checked (for instance, that n_i can be proved by applying a_j with arguments a_n and a_m). In theory, annotations are not needed, in practice they are mandatory, as there are in general too many cases to check. A *proof tree* is just an annotated proof represented as a a tree rather than a flat list.

One example of normal form was used by Gödel [vH67, p592-617] in his famous paper on formally undecidable propositions. Instead of a sequence of signs, or a sequence of sequences, he uses natural numbers, and he shows that the checker is "recursive". One could prove the same for the Bourbaki encoding. Note that Bourbaki does not use the terms "parser", "normal form", "algorithm", and he does not mention the Gödel encoding.

The introduction of [Bou70] discusses formalization of mathematics and says "such a project is absolutely unrealizable", and deduces that it is "imperative to condense the formalized text by the introduction of a fairly large number of new words (called *abbreviating symbols*) and additional rules of syntax (called *deductive criteria*)." The "project" in the sentence above is the possibility for a mathematician to write down the normal form of a theorem and its proof, and to check it. That it is unrealizable is obvious if we consider the size of one. That abbreviations are needed is equally obvious (using similar symbols as $1_{\mathbf{N}}$ and $1_{\mathbf{Z}}$ makes some informal proofs easier to understand). That new syntax rules are needed is more dubious. In fact, all we need to do is change the definition of a checker: it is an algorithm that takes as input a sequence of formal statements x_1, x_2, etc, decides whether each $n(x_i)$ is true; this is possible without (in most cases) computing the normal form of each x_i, provided that a proof tree as defined above.

1.7 The Abuse of Language and Notations

Bourbaki advocates the need of *abuse of language* or *abuse of notations* to "allow us to write the rest of this series in the same way as all mathematical texts are written, that is to say partly in ordinary language and partly in formulae which constitute partial, particular and incomplete formalizations". For instance, in Section III.5.7, Bourbaki has: "writing a/b will imply that b divides a". If a theorem has the form $P \Rightarrow a/b = 2$, it is unclear whether b divides a is an assumption or a conclusion; completing the theorem is generally easy. An abuse of notations consists generally in using the same notation for two different things. For instance, the sum of a and b, denoted by $a + b$ may denote an operation between cardinals, elements of $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{C}$, order-types, ordinals, etc. Another example of abuse of language: Bourbaki defines the ordinal sum of the order-types λ_i as the order-type of the ordinal sum of the family λ_i (considered as ordered sets); he uses the notation $\sum_{i\in I} \lambda_i$ for both sums. These abuse of language or notations are harmless: one could formally write $a +_{\mathbf{N}} b$, $a +_{\mathbf{Z}} b$ (to indicate the sum on $\mathbf{N}$ or $\mathbf{Z}$) or $\sum_{i\in I;T} \lambda_i$, $\sum_{i\in I;O} \lambda_i$ to indicate the ordinal sum of type T or O, and omit the index when the context is clear. In a typed language like Coq, there is often a unique possibility for the operator, given the type of the arguments. In the Bourbaki system, where all objects have the same type, this is not possible. For instance, if a and b are natural integers, then $a +_{\mathbf{Z}} b$ is well-defined (but is not a natural number; in our attempt to formalize algebraic structures, we thought it interesting to define $a + b = \emptyset$ whenever one argument is not in the structure). Identifying distinct objects is a common abuse of language. For instance Bourbaki [Bou89, p. 21] writes $\mathbf{Z} = \mathbf{N} \cup (-\mathbf{N})$. This has to be interpreted as follows: if $i_{\mathbf{N}}$ is the natural injection $\mathbf{N} \to \mathbf{Z}$, $-_{\mathbf{Z}}$ the opposite function on $\mathbf{Z}$, and $f\langle X\rangle$ the set of all $f(x)$ for $x \in X$, then $\mathbf{Z} = i_{\mathbf{N}}\langle\mathbf{N}\rangle \cup -_{\mathbf{Z}}\langle i_{\mathbf{N}}\langle\mathbf{N}\rangle\rangle$.

The next abuse of notation consists in writing $\mathbf{N} \subset \mathbf{Z}$ instead of $i_{\mathbf{N}}\langle\mathbf{N}\rangle \subset \mathbf{Z}$ (this statement appears in the proof of $\mathbf{Q}_+ \cap \mathbf{Z} = \mathbf{N}$, [Bou89, p. 117]), and identifying elements of $\mathbf{N}$ with elements of $\mathbf{Z}$. For instance $1 + 1 = 2$ may be interpreted as $1_{\mathbf{N}} +_{\mathbf{N}} 1_{\mathbf{N}} = 2_{\mathbf{N}}$ or $1_{\mathbf{Z}} +_{\mathbf{Z}} 1_{\mathbf{Z}} = 2_{\mathbf{Z}}$. These are two distinct true statements. In Coq, you can use the scope operator, and write this as $(1 + 1 = 2)_{\mathbf{N}}$ or $(1 + 1 = 2)_{\mathbf{Z}}$, which is unambiguous and readable. The statement "the interval $]1, 2[$ is empty" is

true in $\mathbf{N}$ and $\mathbf{Z}$, but false in $\mathbf{Q}$ or in $\mathbf{R}$, and has no clear meaning in $\mathbf{C}$ (there is no standard interpretation of $\leq_{\mathbf{C}}$). Identifying distinct objects may be problematic. By construction, there is an injection $i_{\mathbf{Z}} : \mathbf{Z} \to \mathbf{Q}$, that allows us to identify $\mathbf{Z}$ with a subset of $\mathbf{Q}$, thus write $\mathbf{Z} \subset \mathbf{Q}$. Now, $i_{\mathbf{Z}} \circ i_{\mathbf{N}} : \mathbf{N} \to \mathbf{Q}$ can be used to identify $\mathbf{N}$ with a subset of $\mathbf{Q}$. Thus $\mathbf{N} \subset \mathbf{Q}$. But $\mathbf{N} \subset \mathbf{Q}$ is a consequence of $\mathbf{Q}_+ \cap \mathbf{Z} = \mathbf{N}$. How can one be sure that this does not induce a contradiction? For instance some proofs rely on the fact that $1 \neq 0$. It might happen that $0_{\mathbf{Z}} = 1_{\mathbf{N}}$; identifying $0_{\mathbf{Z}}$ and $0_{\mathbf{N}}$ yields $0_{\mathbf{N}} = 1_{\mathbf{N}}$ which is absurd. We pretend that it is possible to rewrite the whole work of Bourbaki without false statements like $\mathbf{N} \subset \mathbf{Z}$, without abuse of language, without implicit assumptions, etc. (see the *ssreflect* library for examples of notations that are unambiguous but still readable by humans).

1.8 An Example of Definitions

We shall assume the reader familiar with Coq, see [BC04] for an introduction. Here is a code fragment:

```
choose
  (fun z : Set =>
   inc z BZp &
   Zo BZ (fun z0 : Set => exists u : Set, inc u BZ & z0 = BZmult z u) =
   Zo BZ
     (fun z0 : Set =>
      exists u : Set,
        exists v : Set,
          inc u BZ & inc v BZ & z0 = BZplus (BZmult x u) (BZmult y v)))
```

The quantity BZ denotes the set $\mathbf{Z}$ of rational numbers, BZp is the set of positive numbers, BZplus and BZmult the addition and multiplication. An expression of the form 'Zo A (fun z => p)' denotes the set of all elements z of A that satisfy p. The code fragment above chooses some positive integer z such that $\{t \in \mathbf{Z}, \exists u \in \mathbf{Z}, t = uz\}$ is equal to the set $\{t \in \mathbf{Z}, \exists u \in \mathbf{Z}, \exists v \in \mathbf{Z}, t = xu + yv\}$. We can restate this as: the $\mathbf{Z}$-ideal generated by z is the $\mathbf{Z}$-ideal generated by x and y. Thus, the code fragment defines the gcd of x and y.

There are other possible definitions of the gcd; for instance, the *ssreflect* library [GM08] defines the gcd on $\mathbf{N}$ as follows:

```
Fixpoint gcdn_rec (m n : nat) {struct m} :=
  let n' := n %% m in if n' is 0 then m else
  if m - n'.-1 is m'.+1 then gcdn_rec (m' %% n') n' else n'.
```

In our second implementation of the code, we have adopted the *ssreflect* way of organizing the proofs, but not the data structures; we use freely the axiom of choice, our equality is not decidable, we cannot use the if-then-else construction as in the example above. Instead, we define a function Yo, whose semantics is the same. The following fragment of code defines the quotient q of two rational integers a and b:

```
  let  q:= BZ_of_nat (card_quo (P a) (P b)) in
```

```
Yo (b = BZ_zero) BZ_zero
(Yo (Q a= TPb) (Yo (Q b = TPb) q (BZopp q))
  (Yo (Bnat_divides (P b) (P a)) (Yo (Q b = TPb) (BZopp q) q)
    (Yo (Q b = TPb) (BZopp (BZsucc q)) (BZsucc q))))
```

In this example, the quantity P a is the absolute value of a and Q a its sign, which is either TPa (negative) or TPb (positive). The quotient is zero if $b = 0$; otherwise, let q' be the rational integer $i_{\mathbf{N}}(|a|/|b|)$; if $a \geq 0$ and $b \geq 0$, then $q = q'$, if $a \geq 0$ and $b < 0$, $q = -_{\mathbf{Z}}(q')$. If $a < 0$, the definition is more involved, it depends on whether or not $|b|$ divides $|a|$. The division is characterized by the existence of a remainder r such that $0 \leq r < |b|$ and $a = bq + r$.

We noticed a strange phenomenon: Coq was much slower on **Q** than on **Z**. We think that this is because the normal forms of expressions of **Q** are much larger than those on **Z**. For instance, the sum of a/b and c/d is the reduced form of $(ad+bc, bd)$, where the reduced form of (x, y) is $(x/\gcd(x, y), y/\gcd(x, y))$. If we expand the let in the division, we see that b appears eleven times; thus in the reduced form of (x, y) the gcd appears 22 times. As a consequence, if we express addition on **Q** using only functions of **Z**, we get a large expression.

We solved our speed problem by removing from the data base used by the auto tactic all theorems of the form '**Q** is stable by addition'. We made some other changes, whose effects were to reduce the size of of the normal form of 1 from 16000 to 3000 tokens[5]. However, the normal form of the sum of two cardinals $a + b$ is too big for Coq to compute (it requires more 1G of memory). In particular, since **N** is the set of all n such that $n \neq n + 1$, its normal form cannot be computed.

2. AXIOMS

The Bourbaki theory deals with "assemblies" which are sequences of signs, classified as "terms", "relations" and "ill-typed". For instance '$x'' \in \neg$' is an ill-typed assembly. Any number of prime signs can follow the letter x, this will still be called a letter (thus, there is an unlimited number of distinct letters in the theory). Instead of '$\vee ab$', Bourbaki uses the infix notation '(a) or (b)', where parentheses are often omitted. There are only 7 non-letters in the alphabet, and hundreds of abbreviations, like $A \Rightarrow B$ or $\mathrm{Coll}_x P$ (which stands for $(\exists y)(\forall x)((x \in y) \Leftrightarrow P)$).

Some relations are "true" (either by deduction, or by convention, case where they are called "axioms"), and these are called "theorems". The Bourbaki theory is a first order logic theory, in that one cannot quantify over relations. Instead of $(\forall A)(A \text{ or } A \Rightarrow A)$, which is ill-typed, Bourbaki has: whenever A is a relation, then A or $A \Rightarrow A$ is true. This is an ill-typed assembly if A is considered as a letter, thus the bold face character $\boldsymbol{A}$ is used instead. Such statements are called "Axiom Schemes" (if admitted) or "Deductive Criteria" (if deduced).

[5]In the current version of Gaia, as described in this paper, this has increased, and Coq is again unable to print it.

2.1 The Bourbaki Axioms

These are the 8 Axiom Schemes and 5 Axioms of Bourbaki:

S1. If $\boldsymbol{A}$ is a relation, ($\boldsymbol{A}$ or $\boldsymbol{A}$) $\Rightarrow \boldsymbol{A}$ is an axiom.

S2. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are relations, the relation $\boldsymbol{A} \Rightarrow$ ($\boldsymbol{A}$ or $\boldsymbol{B}$) is an axiom.

S3. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are relations, the relation ($\boldsymbol{A}$ or $\boldsymbol{B}$) $\Rightarrow$ ($\boldsymbol{B}$ or $\boldsymbol{A}$) is an axiom.

S4. If $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are relations, the relation

$$(\boldsymbol{A} \Rightarrow \boldsymbol{B}) \Rightarrow ((\boldsymbol{C} \text{ or } \boldsymbol{A}) \Rightarrow (\boldsymbol{C} \text{ or } \boldsymbol{B}))$$

is an axiom.

S5. If $\boldsymbol{R}$ is a relation, $\boldsymbol{T}$ is a term, and $\boldsymbol{x}$ a letter, then the relation $(\boldsymbol{T}|\boldsymbol{x})\boldsymbol{R} \Rightarrow (\exists \boldsymbol{x})\boldsymbol{R}$ is an axiom.

S6. Let $\boldsymbol{x}$ be a letter, $\boldsymbol{T}$ and $\boldsymbol{U}$ terms, and $\boldsymbol{R}(\boldsymbol{x})$ a relation. Then the relation

$$(\boldsymbol{T} = \boldsymbol{U}) \Rightarrow (\boldsymbol{R}(\boldsymbol{T}) \Leftrightarrow \boldsymbol{R}(\boldsymbol{U}))$$

is an axiom.

S7. Let $\boldsymbol{R}$ and $\boldsymbol{S}$ be relations and $\boldsymbol{x}$ is a letter. Then the relation

$$((\forall \boldsymbol{x})(\boldsymbol{R} \Leftrightarrow \boldsymbol{S})) \Rightarrow (\tau_{\boldsymbol{x}}(\boldsymbol{R}) = \tau_{\boldsymbol{x}}(\boldsymbol{S}))$$

is an axiom.

S8. Let $\boldsymbol{R}$ be a relation, $\boldsymbol{x}$ and $\boldsymbol{y}$ distinct letters, $\boldsymbol{X}$ and $\boldsymbol{Y}$ letters distinct from $\boldsymbol{x}$ and $\boldsymbol{y}$ which do not appear in $\boldsymbol{R}$. Then the relation

$$(\forall \boldsymbol{y})(\exists \boldsymbol{X})(\forall \boldsymbol{x})(\boldsymbol{R} \Rightarrow (\boldsymbol{x} \in \boldsymbol{X})) \Rightarrow (\forall \boldsymbol{Y})\,\mathrm{Coll}_{\boldsymbol{x}}((\exists \boldsymbol{y})((\boldsymbol{y} \in \boldsymbol{Y}) \text{ and } \boldsymbol{R}))$$

is an axiom.

A1. $(\forall x)(\forall y)((x \subset y \text{ and } y \subset x) \Rightarrow (x = y))$.

A2. $(\forall x)(\forall y)\ \mathrm{Coll}_z(z = x \text{ or } z = y)$.

A3. $(\forall x)(\forall x')(\forall y)(\forall y')(((x, y) = (x', y')) \Rightarrow (x = x' \text{ and } y = y'))$.

A4. $(\forall \mathrm{X})\ \mathrm{Coll}_{\mathrm{Y}}(\mathrm{Y} \subset \mathrm{X})$.

A5. There exists an infinite set.

Schemes S1 to S4 say that we have a logical theory (with excluded middle). The notation $(\boldsymbol{T}|\boldsymbol{x})\boldsymbol{R}$ represents the assembly obtained by substituting all occurrences of $\boldsymbol{x}$ by $\boldsymbol{T}$ in $\boldsymbol{R}$. The notation $\tau_{\boldsymbol{x}}(\boldsymbol{R})$ denotes the assembly obtained by inserting τ in front of $\boldsymbol{R}$, replacing all occurrences of $\boldsymbol{x}$ by $\square$, and linking these $\square$ to the τ. If $\boldsymbol{y}$ does not appear in $\boldsymbol{R}$, then the two terms $\tau_{\boldsymbol{x}}(\boldsymbol{R})$ and $\tau_{\boldsymbol{y}}((\boldsymbol{y}|\boldsymbol{x})(\boldsymbol{R}))$ are identical.

Bourbaki defines $(\exists \boldsymbol{x})\boldsymbol{R}$ as '$(\tau_{\boldsymbol{x}}(\boldsymbol{R})|\boldsymbol{x})\boldsymbol{R}$' and $(\forall \boldsymbol{x})\boldsymbol{R}$ as 'not $((\exists \boldsymbol{x})$ not $\boldsymbol{R})$'. These assemblies are well-typed (and are then relations) only if $\boldsymbol{x}$ is replaced by a letter and $\boldsymbol{R}$ by a relation.

Bourbaki sometimes uses a special symbol that was replaced here by parentheses. In Scheme S6, the first occurrence of the symbol in $\boldsymbol{R}(\boldsymbol{x})$ says that further occurrences, like $\boldsymbol{R}(\boldsymbol{T})$, should be interpreted as $(\boldsymbol{T}|\boldsymbol{x})\boldsymbol{R}$. With these notations, S5 can be rewritten as $\boldsymbol{R}(\boldsymbol{T}) \Rightarrow \boldsymbol{R}(\tau_{\boldsymbol{x}}(\boldsymbol{R}))$. Intuitively, $\tau_{\boldsymbol{x}}(\boldsymbol{R})$ is a distinguished object that satisfies $\boldsymbol{R}$ (it corresponds to Hilbert's ϵ). The axiom says: if there is

any term satisfying the relation, then the distinguished term does. Axiom S7 says that this element is not randomly chosen: If $\boldsymbol{R}$ and $\boldsymbol{S}$ are equivalent relations, the distinguished terms satisfying $\boldsymbol{R}$ and $\boldsymbol{S}$ are equal.

There are three possible definitions of equality. The most restrictive one says that two objects are equal if their normal form is the same. In Coq, the natural numbers 1 and 2 have SO and SSO as normal form, thus are unequal. The numbers $1+1$ and 2 have SSO as normal form, thus are equal. The second notion is Leibniz equality: two objects are equal if replacing one by the other does not change anything. In Coq, one can construct a function that associates a to O, b to SO, and c to SSn, whatever n. This implies that 0, 1, and 2 are unequal in the Leibniz sense. There is a last possibility: define equality via axioms, for instance S6, S7, A1. One can prove $1 \neq 2$ in Bourbaki as follows: Let $p(x, y)$ be the property that the two sets x and y are equipotent. Then, as shown in the introduction, 1 is $\tau_Z(p(A, Z))$ where A is $\{\emptyset\}$ and 2 is $\tau_Z(p(B, Z)))$ for some set B with two elements. One can show that p is an equivalence, thus $p(A, A)$ is true, so that S5 implies $p(A, 1)$. The same argument gives $p(B, 2)$, hence by symmetry and transitivity of p, from $1 = 2$ and S6, one deduces $p(A, B)$, which is absurd (there is no bijection between a set with one element and a set with two elements). One can show $1 + 1 = 2$ in Bourbaki as follows: by definition, $1 + 1$ is $\tau_Z(p(C, Z)))$ where C is some set with two elements, thus equipotent to B (given two sets with two elements each, there is always a bijection), so that $p(B, C)$ holds. Since p is an equivalence $\forall Z, p(C, Z) \Leftrightarrow p(B, Z)$ holds, and the result is true by S7. The first theorem shown by Bourbaki asserts that $a = b$ is an equivalence (reflexive, symmetric and transitive).

The notation $x \in X$ can be read as: x is an element of the set X, x is a member of X, or x belongs to X. The notation $x \subset y$ is a short-hand for: $\forall z, z \in x \Rightarrow z \in y$, it is read as x is a subset of y, or x is contained in y, or y contains x. The Russell paradox says that there is no set whose elements are those x such that $x \notin x$. One can avoid this paradox in different ways, for instance by using complicated schemes as S8. One can use a hierarchy of objects (as is the case of Coq); each object has a level, and $x \in X$ is only possible if x is of level n and X of level $n + 1$. One might also distinguish between sets and classes: a class is a set, classes can be members of classes but only sets can be members of sets. Further subdivisions might be needed, for instance [Osb00] introduces "conglomerates". There is no such hierarchy in Bourbaki: every term (a well-typed assembly that is not a relation) is a set. Various denominations are then used for notions that do not enter in this framework like "class of objects equivalent to x" (for equivalence relations that have no graph), "functional symbols" (for functions without graph), a "set with an algebraic structure", etc.

In S8, A2 and A4, the notation $\mathrm{Coll}_x R(x)$ means that R is collectivizing in x, i.e., there is a set Z whose elements are those x such that $R(x)$; such a set is unique from Axiom A1. Axioms A2 says that, for all x and y, there is a set (unique by A1), denoted by $\{x, y\}$, whose elements are x and y. Taking $y = x$ shows that there exists a (unique) set, denoted by $\{x\}$, that has a single element x. Axiom A3 says that, given x and y, there is a set, denoted $J(x, y)$, or simply (x, y) such that $J(x, y) = J(x', y')$ implies $x = x'$ and $y = y'$. For instance $J(x, y) = \{\{x\}, \{x, y\}\}$

could be a candidate. Axiom A4 asserts existence of a set (the powerset of X), denoted $\mathfrak{P}(X)$, whose elements are the sets Y such that $Y \subset X$. Axiom A5 will be described later.

Axiom scheme S8 is a bit tricky. It deals with a relation $R(x, y, z_1, \ldots, z_n)$. Fix z_1, z_2, etc. If, for any y, there is a set X_y (that may depend on z_1, etc), such that $R(x, y, z_1, \ldots, z_n)$ implies $x \in X_y$, then for any Y there is a set whose elements are all x such that $R(x, y, z_1, \ldots, z_n)$ is true for at least one $y \in Y$. Bourbaki uses this scheme only in the following cases:

Example 1. Let X be a set, $R(x)$ a relation such that $R(x) \Rightarrow x \in X$. Then the assumption of S8 is clearly satisfied. Let z_0 be any set (for instance X). Then '$\exists y, y \in \{z_0\}$ and $R(x)$' is equivalent to $R(x)$, so that there exists a set, denoted by $\{x, R(x)\}$, whose elements are all x satisfying $R(x)$. In particular, if P is any property and R is '$x \in X$ and $P(x)$', there exists a set, denoted by $\{x \in X, P(x)\}$ whose elements are all x such that $x \in X$ and $P(x)$.

Example 2. Assume that $R(x, y)$ has the form $x = f(y)$. Choose $\{f(y)\}$ for X_y. Then, for any set Y there is a set formed of all elements of the form $f(y)$ for $y \in Y$. It is sometimes denoted by $f\langle Y\rangle$.

Example 3. Assume that $R(x, y)$ has the form '$y \in I$ and $x \in f(y)$'. Applying the criterion with $X_y = f(y)$ gives a set, denoted by $\cup_{y\in I} f(y)$ formed of all x such that there is y such that $y \in I$ and $x \in f(y)$. If $f(y) = y$ the set is denote by $\bigcup I$.

Example 4. Let X be any set. The set $\{x \in X, x \notin X\}$ is denoted by $\emptyset$. It satisfies $\forall x, x \notin \emptyset$. By Axiom A1, it does not depend on X.

Example 5. Let X and Y be two sets. Let $f_x(y)$ be the pair (x, y) and A_x the set $f_x\langle Y\rangle$. Consider A as a function of x, and define $B = A\langle X\rangle$. Then $u \in \bigcup B$ if and only if there exists $x \in X$ and $y \in Y$ such that $u = (x, y)$. This is the cartesian product of X and Y. (This is essentially the Bourbaki argument, but he defines the product before the union).

2.2 The Zermelo-Fraenkel Axioms

An alternative to the Bourbaki theory is the Zermelo-Fraenkel theory. It has the usual interpretation of the quantifiers $\forall$ and $\exists$, but not the symbol τ, thus is missing a choice function. With the notations of [Kri72] the axioms are

B1. $\forall x \forall y[\forall z(z \in x \Leftrightarrow z \in y) \Rightarrow x = y]$ (Axiom of extent, A1).

B0. $\forall x \forall y \exists z \forall t[t \in z \Leftrightarrow (t = x \text{ or } t = y)]$ (Axiom of the pair, A2).

B2. $\forall x \exists y \forall z[z \in y \Leftrightarrow \exists t(t \in x \text{ and } z \in t)]$ (Axiom of the union).

B3. $\forall x \exists y \forall z[z \in y \Leftrightarrow z \subset x]$ (Axiom of the set of subsets, A4).

B4. $\exists x \exists y[\forall z(z \notin y)$ and $y \in x$ and $\forall u[u \in x \Rightarrow \exists v[v \in x$ and $\forall t(t \in v \Leftrightarrow t = u$ or $t \in u)]]]$ (Axiom of infinity).

AR. $\forall x_1 \ldots \forall x_k\{\forall x \forall y \forall y'[E(x, y, x_1, \ldots, x_k)$ and $E(x, y', x_1, \ldots, x_k) \Rightarrow y = y']$ $\Rightarrow \forall t \exists w \forall v[v \in w \Leftrightarrow \exists u[u \in t$ and $E(u, v, x_1, \ldots, x_k)]]\}$ (Replacement).

AS. $\forall x_1 \ldots \forall x_k \forall x \exists y \forall z[z \in y \Leftrightarrow (z \in x$ and $A(z, x_1, \ldots, x))]$ (Separation).

AC. $\forall a\{[\forall x(x \in a \Rightarrow x \neq \emptyset)$ and $\forall x \forall y(x \in a$ and $y \in a \Rightarrow x = y$ or $x \cap y = \emptyset)] \Rightarrow$
$\exists b \forall x \exists u(x \in a \Rightarrow b \cap x = \{u\})\}$ (Axiom of choice).

AF. $\forall x[x \neq \emptyset \Rightarrow \exists y(y \in x$ and $y \cap x = \emptyset)]$ (Axiom of foundation).

The Zermelo-Fraenkel theory consists in axioms B1, B2, B3, B4, and scheme AR. From AR, one can deduce AS and B0 (see the definition of a doubleton below). The Zermelo theory consists in B1, B0, B2, B3, B4 and AS. It is a weaker theory. Axiom AF is independent of all other axioms, it excludes some weird sets.

The set z defined by axiom B0 is denoted by $\{x, y\}$ or $\{x\}$ if $x = y$. If x is replaced by $\{x\}$ in B2, we get a set y such that $z \in y$ is equivalent to '$\exists t, t \in \{x\}$ and $z \in t$'. Since $t = x$, we get that $z \in y$ is equivalent to $z \in x$, hence $z = x$ by the axiom of extent. Thus we have shown $\bigcup\{x\} = x$.

Axiom scheme AR (axiom of replacement) depends on a relation E that takes at least two arguments. Fix all parameters but the first two ones. Assume that $E(x, y)$ is single-valued in y (i.e., $E(x, y) = E(x, y')$ implies $y = y'$). The scheme says that, for all t, there is a w whose elements are all v such that $E(u, v)$ holds for some $u \in t$. In particular, if E has the form $y = f(x)$, there is a set containing all $f(u)$ for $u \in t$. Note that axiom scheme S8 says: if for any x, there is a set Y_x such that $E(x, y)$ implies $y \in Y_x$, then the same conclusion holds. These two axiom schemes are clearly different; but every use by Bourbaki of S8 could be replaced by AR, with a single exception: the axiom of the union follows from S8.

Axiom Scheme AS (separation or specification) says that for every relation $A(z)$ (that may depend on other parameters) and for every set x, there is a set w whose elements are all $v \in x$ that satisfy A. This axiom is a consequence of AR. Let z be a set, x a function $z \to \{\alpha, \beta\}$ (the target of x is a fixed set with two distinct elements) and $E(x, y)$ the property that for all t, $t \in y$ is equivalent to '$t \in z$ and $p(t) = \alpha$'. The relation E is single-valued by B1, and AS implies that for any x, there exists y such that $E(x, y)$ holds. This set y is a subset of z, and all subsets of z are obtained in this way. Applying axiom AR yields the power set of z. Thus B3 is a consequence of AR, provided that quantification of over functions is allowed. The axiom of the union follows by a similar argument.

Consider the special case of AR where $E(x, y, x_1, \ldots, x_k)$ depends only on y, say $E(x, y, \ldots) = P(y)$, and choose $t = \{\emptyset\}$. Then '$\exists u, u \in t$ and $E(u, v, \ldots)$' is equivalent to $P(v)$. The replacement scheme says: if $P(y)$ and $P(y')$ imply $y = y'$, then there exists a set w such that $v \in w$ is equivalent to $P(v)$. If the set w is empty, then P is always false. Otherwise, if $y \in w$ and $y' \in w$ we have $y = y'$ so that w is a singleton, and $\bigcup w$ is some set y satisfying P. If we denote $\bigcup w$ by $\tau_y(P)$, we obtain the equivalent of Bourbaki's τ. If the predicate P is single-valued, and true for at least one x, we have a *definite description*; the object so defined will be called *the y that satisfies P*.

Consider now axiom B4. The condition $\forall z(z \notin y)$ is equivalent to $y = \emptyset$, and $\forall t(t \in v \Leftrightarrow t = u$ or $t \in u)$ is equivalent to $v = u \cup \{u\}$. Denote this set by $S(u)$. Now B4 says: there exists a set x, that has $\emptyset$ as element, and such that $u \in x \Rightarrow S(u) \in x$. This set is infinite, since S is injective and not surjective. The intersection of all subsets y of x such that $\emptyset \in y$ and y stable by S will be denoted by

ω. It satisfies this property. (In Isabelle/ZF, this is the set of natural integers, and S is the successor function, see [Pau10]). Conversely, assume that there is at least one infinite set; then one can construct the set of natural numbers $\mathbf{N}$, and $f\langle\mathbf{N}\rangle$ satisfies B4, where f is inductively defined by $f(0) = \emptyset$ and $f(n+1) = S(f(n))$. Thus B4 is equivalent to the axiom of infinity A5. Note that, if $\mathbf{N}$ exists, it cannot be finite.

Consider now axiom AC. It says that for every set a, if a is formed of non-empty, mutually disjoint sets, there exists a set b that meets each element of a exactly once. Denote by $f(x)$ the unique element of the intersection of x and b. Then (informally) f is a function such that $f(x) \in x$. More formally, the axiom is equivalent to: for every set A, there exists a function $f : \mathfrak{P}(A) - \emptyset \to A$ such that $f(x) \in x$. It is also equivalent to say that a product of non-empty sets is non-empty; it is also equivalent to Zermelo's Theorem (every set can be well-ordered). It implies that the collection of cardinals is well-ordered.

Consider the following sequence of sets: z_0 is empty, $z_{n+1} = \mathfrak{P}(z_n)$. Let V_ω be the union of these sets. The theory obtained by deciding that a set is an element of V_ω satisfies Zermelo-Fraenkel but not the axiom of infinity. One can define by transfinite induction z_i for every ordinal i, and the theory obtained by deciding that a set is an element of $V_{\omega+\omega}$ satisfies ZF+AF.

There is an implementation of the Zermelo-Fraenkel theorem with Axiom of Foundation in Isabelle, [Pau10]. The replacement axiom is implemented as follows: there is a function R, depending on two parameters A and P, such that, if $P(x, y)$ is single-valued in y (for any $x \in A$), then $R(A, P)$ is the set of all b such that $P(x, b)$ holds for at least one element of x of A. Thus there a function the that replaces Bourbaki's τ, and a function if(P,a,b) (that chooses a if P holds and b otherwise).

Consider a hierarchical theory of sets. Thus, to each set we associate a level, an integer i. In this theory, elements of z_{i+1} are of level i. Objects at level zero are special: For such a set X, we cannot write $x \in X$; thus $X \subset Y$ becomes meaningless and the axiom of extent does not apply. There are many useful sets in the Bourbaki theory that are, in practice, at level 0; they are objects for which $x \in X$ is never considered, and axiom A1 is never applied. In fact, if one tries to identify $\mathbf{N}$ as a subset of $\mathbf{Z}$ and $\mathbf{Z}$ as a subset of $\mathbf{Q}$, one is forced to considers elements of these sets as level-zero sets (and forget A1 for these; such elements are called "urelements" in [BM96]).

2.3 The Gaia Axioms

The theory presented here is based on a set of axioms proposed by Carlos Simpson [Sim04b, Sim04a], that extends the logic of Coq, and is sufficient to prove Bourbaki. In particular $\forall x$ and $\exists x$ are not defined through a symbol τ, this implies that x can be any type (that can be explicitly given, or inferred by Coq). The notation $A \to B$ will be used for implication; the statement $\forall A, A \to A$ is true, but is does not implies 'A or not A'. Axioms Schemes S1 to S6 are clearly true.

The principal idea is to translate the Coq statement $a : b$ (that says that a is of type b) into $a' \in b$ (that says that a' is a member of b) whenever b is a set and

where a' is a type-cast to set. In the work of Simpson, a set was any object of type Type, but in Gaia, it is an object of type Set. This idea is legitimated by the fact that all set constructors of Coq have an equivalent in Bourbaki. For instance, one can consider a type b without constructor. Then $a : b$ is impossible, and b is the empty set. If b is a type with two constant constructors a_1 and a_2, we get a set with two elements, which is disjoint from all sets obtained this way; this construction is possible in Bourbaki: it suffices to apply the axiom of the pair A2 to two fresh letters (each letter is a set, and there is an unlimited amount of letters).

The situation becomes more complicated when the constructors are allowed to be functions that take as arguments of the type to be constructed. Consider for example the case of nat. It has two constructors. The first one is O, it is constant, the second is S, it is a function with one argument of type nat. This type is isomorphic to the set of natural numbers $\mathbf{N}$, where O is identified to zero, and S to the successor function. As noticed above, the existence of $\mathbf{N}$ is a consequence of the axiom of infinity A5, and in Gaia, the axiom of infinity is consequence of our assumption that nat is a set.

The first Gaia axiom asserts existence of an injective function $\mathcal{R}$ (or Ro in the Coq code) such that, for any set x, for any object i of type x, $\mathcal{R}_x i$ is a set (according to Simpson, this axiom can be justified by the existence of an inaccessible cardinal, see [Wer97, Bar01] for some hints). Injectivity says that if i and j have the same type x, then $\mathcal{R}_x i = \mathcal{R}_x j$ implies $i = j$; it can however happen that i and j have different types (thus are unequal) and $\mathcal{R}_x i = \mathcal{R}_y j$ holds. For instance if $x \subset y$, we have such a relation for any $i \in x$ (and some unique j).

```
Parameter Ro : forall x : Set, x -> Set.
Axiom R_inj : forall (x : Set) (a b : x),
   Ro a = Ro b -> a = b.
```

Membership is defined by $\mathcal{R}_y i \in y$. In other words, $x \in y$ if and only if there exists i of type y such that $x = \mathcal{R}_y i$ (this element is unique). The definition of $x \subset y$ is the classical one.

```
Definition inc (x y : Set) :=
   exists a : y, Ro a = x.
Definition sub (a b : Set) :=
   forall x : Set, inc x a -> inc x b.
```

Now comes the axiom of extent that says $x = y$ if and only if the two sets have the same elements. Two functions f and g such that $f(x) = g(x)$ for every argument x are also called equal (in *ssreflect*, one would say f =1 g).

```
Axiom extensionality : forall a b : Set,
   sub a b -> sub b a -> a = b.
Axiom arrow_extensionality :
  forall (x y : Type) (u v : x -> y), (forall a : x, u a = v a) -> u = v.
```

A set x is said non-empty if there exists an element y such that $y \in x$. Any witness y (an element of x) proves that x is non-empty via the constructor nonempty_intro,

and if x is non-empty, the statement $\exists y, y \in x$ becomes true (since the witness satisfies the condition). A type T is said non-empty or inhabited if there is a witness $y : T$; the name y is omitted in the definition that follows since it is irrelevant.

```
Inductive nonempty (x : Set) : Prop :=
    nonempty_intro : forall y : Set, inc y x -> nonempty x.
Inductive nonemptyT (t : Type) : Prop :=
    nonemptyT_intro : t -> nonemptyT t.
```

For Bourbaki, the expression $\exists x, x \in y$ is equivalent to $r(y) \in y$ where $r(y)$ is $\tau_x(x \in y)$. Such a function r cannot be defined in Coq: from $\exists x, p(x)$ one cannot deduce a function $c(p)$ such that $p(c(p))$ (what is the meaning of $c(q)$ be defined for $q \neq p$?). To overcome this difficulty, we introduce an axiom, that will essentially be used in two cases: (a) it ensures existence a function r such that $r(y) \in y$ for every non-empty set y, and (b) in the case of a definite description, the axiom provides a name for the object definitely described.

We consider the general case where the argument of the property p is of type t. Then $c(p)$ is of type t, and this is problematic if the type t is empty. There are different ways to overcome this difficulty, which are all equivalent in practice (if $\exists x, p(x)$ holds, then the type t of the argument x is inhabited). One possibility consists in augmenting the type t (using the option mechanism of Coq); another possibility consists in introducing a second argument. For instance, one can consider $c(p, x)$, where x is any object of type t, case where $p(x) \Rightarrow p(c(p, x))$ would be a valid statement. The solution adopted here is to take as second argument a proof q that the type t is inhabited (and such a proof follows from a witness x). The axiom says: if $\exists x, p(x)$ then $\mathcal{C}_t(p, q)$ satisfies p. The choice function depends on q, but this is harmless.

```
Parameter chooseT :
  forall (t : Type) (p : t -> Prop) (q : nonemptyT t),  t.
Axiom chooseT_pr :
  forall (t : Type) (p : t -> Prop) (q : nonemptyT t),
    ex p -> p (chooseT p q).
```

Axiom Scheme S8 (replacement axiom AR[6]) is implemented as: for any set x, any function f that maps elements of x to sets, there is a set y, namely 'IM f', such that $z \in y$ if and only if z has the form $f(i)$ for some $i : x$.

```
Parameter IM : forall x : Set, (x -> Set) -> Set.

Axiom IM_exists : forall (x : Set) (f : x -> Set) (y : Set),
   inc y (IM f) -> exists a : x, f a = y.
Axiom IM_inc : forall (x : Set) (f : x -> Set) (y : Set),
   (exists a : x, f a = y) -> inc y (IM f).
```

Many proofs of Bourbaki are non-constructive because they use the excluded middle law. In order to prove that P is true, it suffices to prove that the negation

[6]This is a generalized replacement axiom as it implies the axiom of the union of of the powerset.

of P leads to a contradiction. There are different variants of the law; we chose one here. The proof irrelevance axiom in classical, but not needed here (it would imply that $\mathcal{C}_t(p,q)$ is independent of q). The last axiom allows us to use $=$ instead of $\Leftrightarrow$, thus use the autorewrite tactic in some cases.

```
Axiom excluded_middle : forall P : Prop, ~ ~ P -> P.
(* Axiom proof_irrelevance : forall (P : Prop) (q p : P), p = q. *)
Axiom iff_eq : forall P Q : Prop, (P -> Q) -> (Q -> P) -> P = Q.

Lemma p_or_not_p : forall P : Prop, P \/ ~ P.
Lemma equal_or_not: forall x y: Set, x= y \/ x<> y.
Lemma inc_or_not: forall x y: Set, inc x y \/ ~ (inc x y).
```

The following two axioms were introduced by Simpson. They say that $\mathcal{R}_{\mathsf{nat}}0 = \emptyset$ and $\mathcal{R}_{\mathsf{nat}}(Sn) = s(\mathcal{R}_{\mathsf{nat}}n)$ where $s(u) = u \cup \{u\}$ and S is the successor function on nat. If one defines $\mathbf{N}$ as the set of finite von Neumann ordinals, the axioms say that $\mathcal{R}_{\mathsf{nat}}$ induces a bijection $\mathsf{nat} \to \mathbf{N}$.

```
(*
Axiom nat_realization_0 : forall x : E, ~ inc x (Ro 0).
Axiom nat_realization_S :
          forall (n : nat) (x : E),
          inc x (Ro (S n)) = (inc x (Ro n) \/ x = Ro n).
*)
```

In the original work of Simpson, any Type was a set. In particular, False is the empty set (by extensionality, since there is no object of type False) and True is a singleton (there is at least one proof of it, and all proofs of True are equal by the proof irrelevance axiom). Simpson added an axiom that says that $\mathcal{R}t = \mathcal{R}0$, whenever t is a proof of True, so that True is $\{\emptyset\}$. In particular, $\{\mathsf{True}, \mathsf{False}\}$ is $\mathcal{R}2$.

Simpson added the axiom that says $\mathcal{R}x = x$ whenever x is a proposition. Since, by the excluded middle law, x is true or false (thus equivalent to True or False, thus equal to one of them because of the iff_eq axiom), $\mathcal{R}2$ is equal to Prop by extensionality! Here the two axioms:

```
(*
Axiom prop_realization : forall x : Prop, Ro x = x.
Axiom true_proof_realization_empty : forall t : True, Ro t = Ro 0.

Lemma nat_zero_emptyset : Ro 0 = emptyset.
Lemma R_one_singleton_emptyset : Ro 1 = singleton emptyset.
Lemma R_two_prop : Ro 2 = Prop.
*)
```

2.4 Proofs

For Bourbaki, a *proof* is a sequence expressions: A_1, A_2, A_3, etc, that can be terms or relations, such that each A_i is either (0) a term, (1a) an axiom, or (1b) an instance of an axiom scheme, or (2) is obtained by application of the modus ponens

to two relations preceding it in the list; a *theorem* (or *proposition*[7], *lemma*, etc) is any relation that appears in a proof. We shall call this a B-proof in order to avoid any confusion.

Bourbaki has over 60 criteria that help proving theorems. These are rules that can be deduced from the axioms. The first one C1 says: if $\boldsymbol{A}$ and $\boldsymbol{A} \to \boldsymbol{B}$ are theorems, then $\boldsymbol{B}$ is a theorem. This is also known as syllogism or Modus Ponens. The *ssreflect* library provides two tactics have and suff that implement this criterion, one starts with proving $\boldsymbol{A}$ and the other starts with $\boldsymbol{A} \to \boldsymbol{B}$. Criterion C14 is a converse to C1: if $\mathcal{T}$ is a theory, and $\mathcal{T}'$ the theory obtained by adjoining $\boldsymbol{A}$ to the list of axioms, and if $\boldsymbol{B}$ is a theorem in $\mathcal{T}'$, then $\boldsymbol{A} \to \boldsymbol{B}$ is a theorem in $\mathcal{T}$. This is called the *method of the auxiliary hypothesis*. This is implemented in *ssreflect* as follows: in order to prove that $A \to B$ is true, it suffices to say move=> A, then prove B in a context where A is true. Criterion C8 says that $\boldsymbol{A} \to \boldsymbol{A}$ is a theorem. This is by definition 'not $\boldsymbol{A}$ or $\boldsymbol{A}$', and is called the "Law of Excluded Middle".

Criterion C30 implies that, if $\boldsymbol{R}$ is a relation, $\boldsymbol{T}$ a term, $\boldsymbol{x}$ a letter, if $(\forall \boldsymbol{x})\boldsymbol{R}$ is a theorem, then $(\boldsymbol{T}|\boldsymbol{x})R$ is a theorem. Conversely, C36 says that, if $\boldsymbol{x}$ is not a constant of the theory $\mathcal{T}$, if $\boldsymbol{R}$ is a theorem of the theory $\mathcal{T}'$ obtained by adjoining $\boldsymbol{A}$ to the list of axioms, then $(\forall \boldsymbol{x})(\boldsymbol{A} \Rightarrow \boldsymbol{R})$ is a theorem of $\mathcal{T}$. In particular, if $\boldsymbol{R}$ is a theorem, so is $(\forall \boldsymbol{x})\boldsymbol{R}$. These criteria correspond to the *ssreflect* constructions move:x and move=>x.

Note that these criteria all say: some theorems can be deduced from other theorems. They never say: one gets a B-proof of A from a B-proof of B and other considerations. Thus a Bourbaki *proof* is just a list of criteria with their arguments. Let's consider two examples.

Proposition 1 [Bou68, Chapter II, section 1, p. 67] is "$x \subset x$", with "Obvious" as justification. Proposition 2 [Bou68, same page] is "$(x \subset y$ and $y \subset z) \Rightarrow (x \subset z)$", and the justification is: "Adjoin the hypotheses $x \subset y$, $y \subset z$, and $u \in x$. Then the relations

$$(u \in x) \Rightarrow (u \in y), \qquad (u \in y) \Rightarrow (u \in z),$$

are true, and therefore the relation $u \in z$ is true".

Notice that Bourbaki uses the expression "is true" instead of "is a theorem" and "is false" instead of "the negation is a theorem". These terms are somehow ambiguous, especially with expressions containing free variables. In fact, according to Criteria C27 and C30, the statement $P(x)$ is equivalent to $\forall x, P(x)$, whenever x is not a constant. The statement $x \subset x$ is logically equivalent to $\forall x, x \subset x$, hence to $\forall x, \forall y, (y \in x \Rightarrow y \in x)$, and true by C8. The statement $x \neq x$ is false, since $x = x$ true (Theorem 1, Chapter 1, section 5). Let A be the statement $x = 0$. It is not true, since it implies $1 = 0$ which is not true (see below). It is not false, since A false is the same as $\forall x, \neg A$, which is equivalent to $\neg(\exists x, A)$, which is the negation of a theorem. The excluded middle law (that says that A is either true or false),

[7]We shall use the term "Proposition" (with an upper case P) in the Bourbaki sense of "theorem" when discussing the Propositions 1 and 2 of [Bou68, p. 67]. In all other cases "proposition" is a synonym to "relation" in the Bourbaki sense.

has to be interpreted as: $\forall x, (A \vee \neg A)$, which not the same as $(\forall x, A) \vee (\forall x, \neg A)$. For this reason, all theorems proved in Gaia contain no free variables.

Proving that a property is false is not always easy as no axiom nor axiom scheme has $\neg A$ as consequence (in fact, the symbol $\neg$ does not appear in the list of axioms). In most cases, we use a proof by contradiction: if P and $\neg P$ are true, then any proposition is true. In particular, if $A \Rightarrow P$ and $A \Rightarrow \neg P$ then $A \Rightarrow \neg A$ is true; since $\neg A \Rightarrow \neg A$ is equally true, it follows (by the method of disjunction) that $\neg a$ is true and A is false. Since $A \Rightarrow B$ is equivalent to $\neg B \Rightarrow \neg A$, we deduce: if A implies a false proposition then A is false. Example. Let P the statement $\emptyset \in \emptyset$. It implies $\exists x, x \in \emptyset$, which is false by definition of the emptyset. Thus P is false. If $0 = \emptyset$ and $1 = \{\emptyset\}$, the proposition $0 = 1$ implies P, hence is false. In Coq $\neg P$ is $P \rightarrow$ False, where False is a statement without proof (i.e., a type without constructor).

The conclusion of Proposition 2 is $\forall u, u \in x \Rightarrow u \in y$. Criterion C36 then says: the conclusion holds if we can prove $u \in y$ in a context where $u \in x$ holds. The Coq proof that follows introduces first the three variables x, y and z, then the two assumptions $x \subset y$ and $y \subset x$, then the variable u and the assumption $u \in x$. The result is immediate by applying the two assumptions.

```
Lemma sub_refl : forall x, sub x x.
Proof. by rewrite /sub. Qed.
Lemma sub_trans : forall x y z, sub x y -> sub y z -> sub x z.
Proof. by move => x y z Sxy Syx u Iux; apply Syx; apply Sxy. Qed.
```

The text between Proof and Qed is formed of commands (and arguments) that construct the proof tree, which is the following for the two examples:

```
sub_refl = fun x x0 : Set => ssrfun.id
     : forall x : Set, sub x x
sub_trans =
fun (x y z : Set) (Sxy : sub x y) (Syx : sub y z) (u : Set) (Iux : inc u x)
 => Syx u (Sxy u Iux)
     : forall x y z : Set, sub x y -> sub y z -> sub x z
```

3. THE THEORY OF SETS

We explain here the first consequences of our axioms, showing how to implement Chapter II, Sections 1 to 5 of the Theory of Sets of Bourbaki, [Bou68]. The first section here is directly adapted from Carlos Simpson.

3.1 Basic Constructions

The emptyset is a set with no constructor. This means that x:emptyset holds for no term x, hence that $x \in$ emptyset is false and the set is empty.

```
Inductive emptyset : Set :=.
Definition empty (x : Set) := forall y : Set, ~ inc y x.
```

If $x \in y$ there is a unique i of type y such that $x = \mathcal{R}_y i$. The axiom of choice and the property $x \in y \Rightarrow y \neq \emptyset$ allows us to consider i as function $\mathcal{B}(H_{x;y})$ of x, y, and a proof $H_{x;y}$ of the statement $x \in y$, from which x and y can be deduced. The function is denoted by Bo in Coq; it satisfies:

$$\mathcal{R}(\mathcal{B}(H_{x;y})) = x; \qquad \mathcal{B}(H_{\mathcal{R}z;y}) = z.$$

```
Definition Bo (x y : Set) (hyp : inc x y) :=
  chooseT (fun a : y => Ro a = x) (inc_nonempty hyp).

Lemma B_eq : forall x y (hyp : inc x y), Ro (Bo hyp) = x.
Lemma B_back : forall (x:Set) (y:x) (hyp : inc (Ro y) x), Bo hyp = y.
```

The axiom of choice asserts existence of a function $\tau(p)$, such that $p(\tau(p))$ is true, provided that there exists a set x satisfying p. The function is extended to the case where no x satisfies p, by defining $\tau(p) = \emptyset$ in that case.

```
Definition choose (p:Set -> Prop) :=
  chooseT (fun x => (ex p -> p x) & ~(ex p) -> x = emptyset)
  (nonemptyT_intro emptyset).
Lemma choose_pr : forall p, ex p -> p (choose p).
```

The representative of a set X is $\tau(x \in X)$. Denoting it by $r(X)$, we have $r(X) \in X$, whenever X is nonempty.

```
Definition rep (x : Set) := choose (fun y : Set => inc y x).
Lemma nonempty_rep : forall x, nonempty x -> inc (rep x) x.
```

The next function mixes the axiom of choice and the law of excluded middle. Consider a property P, a function a that assigns a value $a(p)$ to each proof p of P and a function b that assigns a value $b(q)$ to each proof q of the negation of P. We define $\mathcal{C}_C(P, a, b)$ to be some value z such that, for all p we have $a(p) = z$ and for all q we have $b(q) = z$. In the case where $a(p)$ is the constant function x and b the constant function y, we denote it by $\mathcal{Y}(P, x, y)$, or Yo in the Coq code.

```
Definition by_cases (T : Type) (P : Prop) (a : P -> T)  (b : ~ P -> T) :=
  chooseT (fun x : T =>  (forall p : P, a p = x)
                       & (forall q : ~ P, b q = x))
    (by_cases_nonempty a b).
Definition Yo : Prop -> Set -> Set -> Set :=
  fun P x y => by_cases (fun _ : P => x) (fun _ : ~ P => y).
```

Let P, a and b be as above, and assume a and b constant (this is a consequence of the proof irrelevance axiom, which is generally admitted). Then, there is some z such that, for all p and q we have $a(p) = z$ and $b(q) = z$. In fact, if P is true, there is a proof p_0, and it suffices to take $z = a(p_0)$. That $b(q) = z$ follows from the fact that, if there is a proof of P and its negation, then every proposition is true. On the other hand, if P is not true, the excluded middle law asserts that its negation is true, and there is a proof q_0, it suffices to take $z = b(p_0)$. This argument shows that the axiom of choice can be applied.

In particular $\mathcal{Y}(P,x,y)$ is x if P is true, and is y otherwise.

```
Lemma Y_if_rw : forall (P:Prop) (hyp :P) x y, Yo P x y = x.
Lemma Y_if_not_rw : forall (P:Prop) (hyp : ~P) x y, Yo P x y = y.
```

The next definition is a bit complicated, for those not familiar with the Coq syntax. It implements the axiom of separation (the easy part of Scheme S8). Given a set x, a property p, Zo x (fun z => p) is the subset of all elements z of x that satisfy p. It is sometimes denoted by $\{z \in x, p\}$. We start with the original code of C. Simpson. Here Z and R correspond to Zo and Ro, while E is the type of a set, namely Type, and EP is the type E → Prop.

Let f be the function $a \mapsto p(\mathcal{R}_x a)$ of type $x \to$ Prop. For each a, we consider the pair $(a, f(a))$, considered as an instance of a record. The first element of the pair is known as head, the second as tail. The record is created via Rec (a function whose arguments are x and f), and an instance $(a, f(a))$ via the function join, that takes a as argument.

Let P be such a record, and consider the set of all $\mathcal{R}_x(a)$, where a is the head of an instance (a is head t for some t of type P). This set exists, thanks to the axiom of replacement (the function IM). If b is $\mathcal{R}_x a$, then $b \in x$ and the tail of t is $p(b)$, hence b satisfies p. Conversely, if b is an element of x that satisfies p, then b has the form $\mathcal{R}_x a$, $f(a)$ is true, $(a, f(a))$ is of type P and b is in the image.

```
(*
Record Rec (x : E) (f : x -> E) : E := join {head : x; tail : f head}.
Implicit Arguments join [x f].

Definition Z : E -> EP -> E.
intros x p.
pose (Rec (fun a : x => p (R a))).
exact (IM (fun t : P => R (head t))).
Defined.
*)
```

The Gaia definition is similar, except that the record has been replaced by a co-inductive type (this reduces the number of auxiliary functions). Since there is no function head anymore, one has to say something like let (a, _) := z in Ro a.

```
CoInductive Zorec (x : Set) (f : x -> Prop) : Set :=
   Zorec_c : forall a:x, f a -> Zorec f.
Definition Zo (x:Set) (p:Set -> Prop) :=
  IM (fun  (z : Zorec (fun (a : x) => p (Ro a))) => let (a, _) := z in Ro a).
```

The key relation is: $z \in \mathcal{Z}(x,p)$ if and only if $z \in x$ and $p(z)$. Simpson has defined a tactic Ztac, that implements this rule. The xd tactic splits all conjunctions (it replaces '$z \in x$ & $p(z)$' by two assumptions '$z \in x$' and '$p(z)$'). This is incompatible with the *ssreflect* way of organizing proofs, and the tactic has been removed in Gaia Version 2.

```
Lemma Z_rw: forall  x p y, inc y (Zo x p) = (inc y x & p y).
```

```
Lemma Z_inc : forall x (p: Set -> Prop) y, inc y x -> p y -> inc y (Zo x p).
(*
Ltac Ztac :=
  match goal with
  | id1:(inc ?X1 (Z _ _)) |- _ => pose (Z_all id1); xd
  |  |- (inc _ (Z _ _)) => ap Z_inc; au
  | _ => idtac
  end.
*)
```

We consider here a set D (called the *canonical doubleton*) with two constructors, say α and β. These objects are of type D, so that $\mathcal{R}_D\alpha$ and $\mathcal{R}_D\beta$ are elements of D; we shall denote them by TPa and TPb. The image of D by the function that associates x to α and y to β is denoted by $\{x, y\}$. Its elements are x and y (in the case $x = y$, it is denoted $\{x\}$). We mentioned above that the axiom of the pair is a consequence of the axiom of replacement, and the other axioms of ZF. In fact, the set $\mathfrak{P}(\mathfrak{P}(\emptyset))$ has exactly two elements, and can be used instead of D; this method is used in Isabelle/ZF [Pau10].

```
Inductive two_points : Set :=  | two_points_a | two_points_b.
Definition TPa := Ro two_points_a.
Definition TPb := Ro two_points_b.

Lemma two_points_pr: forall x, inc x two_points = (x = TPa \/ x = TPb).
Lemma two_points_distinct: TPa <> TPb.

Definition doubleton (x y : Set) :=
  IM (fun t => two_points_rect (fun _ : two_points => Set) x y t).
Definition singleton x := doubleton x x.

Lemma singleton_rw: forall x y, inc y (singleton x) = (y = x).
Lemma doubleton_rw : forall x y z : Set,
  inc z (doubleton x y) = (z = x \/ z = y).
```

Let X be any set, D the canonical doubleton, α of type D, and p of type $x \to D$. Then $X_p = \{x \in X,\ p(x) = \alpha\}$ is a subset of X, and all subsets of X have this form. The set of all X_p is called the powerset $\mathfrak{P}(X)$ of X. This shows that there is no need to introduce A4 as an axiom.

```
Definition powerset (x : Set) :=
  IM (fun p : x -> two_points =>
    Zo x (fun y : Set => forall hyp : inc y x, p (Bo hyp) = two_points_a)).
```

The definition that follows is similar to that of $\mathcal{Z}$ (it corresponds to example 3 of the discussion of Axiom scheme S8, see page 89). There is only one argument x, the predicate $p(y)$ being replaced by $y \in x$. The result is a set z such that $a \in z$ if and only if there exists y such that $y \in x$ and $a \in y$. It is the union of x.

```
Record Union_integral (x : Set) : Set :=
  {Union_param : x; Union_elt : Ro Union_param}.
```

```
Definition union (x : Set) :=
  IM (fun i : Union_integral x => Ro (Union_elt i)).
```

Let x be a set, and $p(a)$ the property $\forall y, y \in x \Rightarrow a \in y$. It is satisfied by all a if x is empty, and is equivalent to $a \in \{z \in E, p(z)\}$, whenever E is a superset of one element of x. This will be called the intersection of x. Originally, Simpson used rep x for E; in Gaia, we use the union of x instead (this avoids using the axiom of choice). The union and intersection of the doubleton $\{x, y\}$ is called the union or intersection of the two sets x and y, denoted $x \cup y$ and $x \cap y$.

```
Definition intersection (x : Set) :=
  Zo (union x) (fun y : Set => forall z : Set, inc z x -> inc y z).
Definition union2 (x y : Set) := union (doubleton x y).
Definition intersection2 (x y : Set) := intersection (doubleton x y).
```

In 1908, Zermelo (see [vH67, 199-215]) used only unordered pairs for the definition of a bijection; the source and target had to be disjoint; nevertheless, he was able to define cardinals and show the theorem of Cantor, i.e., $2^n > n$. In 1914, Wiener proposed $\{\{\{x\}, \emptyset\}, \{\{y\}\}\}$ for the ordered pair (x, y). It is well-typed (according to Russell's theory of types). In 1921, Kuratowski introduced $\{\{x\}, \{x, y\}\}$, which is now the classical definition. It was adopted by Bourbaki in 1970, but the English version [Bou68] of the book still ignores it, and uses an axiom.

In Gaia, we tried different variants. The first definition was introduced by Simpson, its is a variant of the Wiener definition with less braces; with this definition a pair is a set with exactly two elements. The last definition is that of Kuratowski, and the other one uses an axiom.

```
(*
Definition spair x y :=
  doubleton (singleton x) (doubleton emptyset (singleton y)).
Parameter bpair : Set -> Set -> Set.
Axiom axiom_of_pair : forall x y x' y' : Set,
  (bpair x y = bpair x' y') -> (x = x' & y = y').
*)
Definition kpair x y := doubleton (singleton x) (doubleton x y).
```

Let J be any of the three functions defined above. A set z is called a *pair* if it is of the form $J(u, v)$ for some u and v. The function J satisfies the property that if $z = J(u', v')$, then $u = u'$ and $v = v'$. Consider the x such that there exists v such that $z = (x, v)$, and the y such that there is u such that $z = (u, y)$. Then $z = J(x, y)$. Moreover, if we write $x = P(z)$ and $y = P(z)$ we get $P(J(u, v)) = u$ and $Q(J(u, v)) = v$. The two projectors are written $\mathrm{pr}_1 z$ and $\mathrm{pr}_2 z$ in Bourbaki.

```
Notation J := kpair.
Definition is_pair (u : Set) := exists x, exists y, u = J x y.
Lemma pr1_pair: forall x y, P (J x y) = x.
Lemma pr2_pair: forall x y, Q (J x y) = y.
```

In Gaia, *the x such that R* is implemented $\tau_x R(x)$. Isabelle/ZF (see [Pau10]) has a *the* function, and considers, for function f, any pair z, the y such that there exists u and v such that $z = J(u, v)$ and $y = f(u, v)$. It is denoted by $\mathsf{split}(f, z)$. If $f_1(u, v) = u$ and $f_2(u, v) = v$, then the two projectors are just $\mathsf{split}(f_1)$ and $\mathsf{split}(f_2)$. Note that $x \neq (x, y)$ and $y \neq (x, y)$ are theorems of Isabelle/ZF (as trivial consequence of the axiom of foundations); these statements are unprovable in Gaia.

```
(* Original definition of Simpson
Definition pr1 (u : Set) :=
  choose (fun x : Set => exists y : Set, u = J x y).
Definition pr2 (u : Set) :=
  choose (fun y : Set => exists x : Set, u = J x y).
*)
```

Assume that is pair is defined by the Kuratowski formula $z = \{\{x\}, \{x, y\}\}$. Let A be union of z, so that $A = \{x, y\}$. Let D and E be the sets of elements $t \in A$ such that either $\{t\} \in z$ or $\{x, t\} = A$. We have $D = \{x\}$ and $E = \{y\}$. It follows that x is the union of D and y is the union of E. If one replaces x in the definition of E by the union of D, one obtains a definition of the projectors that does not use the axiom of choice. Here is the code of [Bar01]:

```
(*
Definition fst p := union (subset (union p) (fun x => singl x \in p)).
Definition snd p :=
  union (subset (union p) (fun z => pair (fst p) z == union p)).
*)
```

Let B the intersection of z, and C the complement of the intersection in the union. We have $B = \{x\}$, so that $C = \{y\}$ (case $x \neq y$) or $C = \emptyset$ (otherwise). Thus, x is the union of B, and y is either x (in the case where C is empty) or the union of C (otherwise). Note that the axiom of choice in hidden in Yo. We can get rid of it, by using the idea of [Bar01]; this gives kpr3.

```
Notation P := kpr1.
Notation Q := kpr2.
Definition kpr1 x := union (intersection x).
Definition kpr2 x := let a := complement (union x) (intersection x) in
  Yo (a = emptyset) (kpr1 x) (union a).
(*
  Definition kpr3 x :=
    union (Zo (union x) (fun z => (doubleton (kpr1 x) z) = (union x))).
*)
```

Given two sets A and B, the set of all (a, b) for $a \in A$ and $b \in B$ is called the product and denoted by $A \times B$. This is a great simplification over Simpson's definition. The quantity fun_image X f is the set of all $f(x)$ for $x \in X$ and denoted by $f\langle X \rangle$.

```
Definition fun_image (x : Set) (f : Set -> Set) :=
  IM (fun a : x => f (Ro a)).
```

```
Definition product (A B : Set) :=
  union (fun_image A (fun x =>  (fun_image B (fun y => J x y)))).
```

3.2 Correspondences

A set G whose elements are pairs is called a graph. If $(x, y) \in G$ one says that x and y are related by G. The domain of the graph is $\mathrm{pr}_1\langle G\rangle$, it is the set of all x related to some y; the range $\mathrm{pr}_2\langle G\rangle$ satisfies a similar property. The substrate of the graph is the union of the domain and range.

```
Definition is_graph r :=
  forall y, inc y r -> is_pair y.
Definition domain f  := fun_image f P.
Definition range f := fun_image f Q.
Definition substrate r :=  union2 (domain r) (range r).
```

For Bourbaki [Bou68, p. 76], a correspondence is a triple $\Gamma = ((G, S), T)$, satisfying

$$G \text{ is a graph and } \mathrm{pr}_1 G \subset S \text{ and } \mathrm{pr}_2 G \subset T. \tag{C}$$

This is is equivalent to

$$G \subset S \times T. \tag{C'}$$

If $\Gamma = (G, S, T)$ is a triple, the three quantities G, S and T are called the graph, source and target of the triple. Such triples are used to define functions, equivalences and orders. In the case of an equivalence or order, we have $S = T$ and this is the substrate E of G. Bourbaki says [Bou68, p. 132] "By abuse of language we shall sometimes refer to the graph G of Γ as an ordering on E" (he does this almost everywhere). Condition (C') becomes $G \subset E \times E$, which is true whenever E is a superset of the substrate of G. In these two cases, there is no need to distinguish between Γ and G, and we shall not introduce Γ. A function is a triple satisfying

$$G \text{ is a functional graph and } \mathrm{pr}_1 G = S \text{ and } \mathrm{pr}_2 G \subset T. \tag{F}$$

A graph is functional if each element is related to at most one other element. The axiom of choice asserts existence of a mapping $\mathcal{V}_G$ (the *evaluation function* of the graph) such that each x of the domain is associated to $\mathcal{V}_G(x)$. This element is in the range and is uniquely defined by this property if the graph is functional. Here are some properties.

```
Definition V (x f : Set) := choose (fun y : Set => inc (J x y) f).
Lemma in_graph_V : forall f x,
  fgraph f -> inc x f -> x = J (P x) (V (P x) f).
Lemma fdomain_pr1 : forall f x,
  fgraph f -> inc x (domain f) -> inc (J x (V x f)) f.
Lemma inc_V_range:  forall f x,
  fgraph f -> inc x (domain f) -> inc (V x f) (range f)
```

If f is a function that maps sets to sets, and X is a set, one can consider the set of all pairs $(x, f(x))$ for $x \in X$. This is a functional graph, denoted by $\mathcal{L}_X f$. Its domain is X, and its evaluation function is f.

```
Definition graph_constructor (x : Set) (p : Set -> Set) :=
  fun_image x (fun y => J y (p y)).
Notation L := graph_constructor.

Lemma L_fgraph : forall p x, fgraph (L x p).
Lemma L_domain : forall x p, domain (L x p) = x.
Lemma L_V_rewrite :  forall x p y, inc y x -> V y (L x p) = p y.
```

We tried different implementations for a correspondence. In the first implementation, a correspondence was a functional graph, whose domain was a subset of the type string[8]. We then thought of using an inductive structure containing the three fields together with the assumption (C). A function would be a correspondence with additional assumptions (F); as a consequence functions and correspondences were of different type. This mechanism was too complicated to use, and we decided that a correspondence and a function should have the same type. With this definition, a function is not a set, and one cannot consider the set of all functions satisfying some properties.

```
(*
  Record correspondenceC: Type :=
    corresp { source:Set; target:Set; graph :Set }.
Definition is_correspondence f: correspondenceC := ...
Definition is_function f: correspondenceC := ...
*)
```

In the current definition a correspondence is a triple satisfying (C'). This matches the definition of 1 given at the start of this paper (see [Bou68, p. 158]). The set of correspondences between A and B is just $\mathfrak{P}(A \times B) \times \{A\} \times \{B\}$.

```
Definition is_correspondence f  :=
  is_triple f &  sub (graph f) (product (source f) (target f)).

Definition set_of_correspondences (x y:Set) :=
  product(powerset (product x y))
  (product (singleton x) (singleton y)).
```

To each function f, for instance the injection $\mathbb{N} \to \mathbb{Z}$, one can associate a correspondence, denoted $\mathcal{L}f$. Its source is $\mathbb{N}$, its target is $\mathbb{Z}$.

```
Definition gacreate (a b:Set) (f:a->b) := IM (fun y:a => J (Ro y) (Ro (f y))).
Definition acreate (a b:Set) (f:a->b)  := corresp a b (gacreate f).
Lemma source_acreate : forall (a b :Set)(f:a->b),
   source (acreate f) = a.
Lemma target_acreate : forall (a b:Set) (f:a->b),
   target (acreate f) = b.
```

The inverse graph of G, denoted by G^{-1} is the set of all pairs (x, y) such that $(y, x) \in G$. The composition $G \circ H$ of two graphs G and H is the set of all (x, y)

[8]This is an unordered triple, where each item is an ordered pair, formed of a string and a general set. This can be generalized to any algebraic structure, as proposed by C. Simpson

such that there exists z such that $(x, z) \in H$ and $(z, y) \in G$. It is associative and the inverse of composition is the composition of inverses (in reverse order). If the graphs are composable, composition is a functional graph whose evaluation function is $\mathcal{V}_{G\circ H}(x) = \mathcal{V}_G(\mathcal{V}_H(x))$.

```
Definition inverse_graph r :=
  Zo(product(range r)(domain r))
  (fun y=> inc (J (Q y)(P y)) r).
Definition compose_graph r' r :=
 Zo(product(domain r)(range r'))(fun w => exists y,
   (inc (J (P w) y) r & inc (J y (Q w)) r')).
Theorem composition_associative:forall r r' r'',
  is_graph r -> is_graph r' -> is_graph r'' ->
  compose_graph r'' (compose_graph r' r) =
  compose_graph (compose_graph r'' r') r.
Theorem inverse_compose:forall r r',
  inverse_graph (compose_graph r' r) =
  compose_graph (inverse_graph r)(inverse_graph r').
```

The inverse correspondence of $\Gamma = (G, S, T)$ is $\Gamma^{-1} = (G^{-1}, T, S)$, the composition of $f = (G, A, B)$ and $f' = (G', B, C)$ is $f' \circ f = (G' \circ G, A, C)$. The same definitions apply to functions (defined below). Composition is associative, and the identity is a unit for composition. The inverse of a function is in general not a function.

```
Definition composableC r' r :=
  is_correspondence r & is_correspondence r' & source r' = target r.
Definition inverse_fun m :=
  corresp(target m) (source m)(inverse_graph (graph m)).

Definition compose r' r :=
  corresp (source r)(target r') (compose_graph (graph r')(graph r)).
Definition identity x := corresp x x (identity_g x).
```

3.3 Functions

The definition of a function in the "Summary of results" [Bou68, p. 351] is a bit fuzzy: "We give the name *function* to the operation which in this way associates with every element $x \in E$ the element $y \in F$ which is in the given relation with x". This definition applies to S (the successor function of Coq, it defines $\mathbb{N}$), to $+_{\mathbb{N}}$ (addition on $\mathbb{N}$, a recursively defined function by Coq), to $+_{\mathrm{Card}}$ (addition on cardinals, an operation that has no graph), to $x \mapsto 1/x$ (inverse function on $\mathbf{R}$, partially defined), to the evaluation function $x \mapsto \mathcal{V}_G x$, etc.

The formal definition is a triple that satisfies (F), a correspondence whose graph is functional, and whose source is equal to the domain of the graph. Thus, for any x in the source, there is a unique y, denoted $\mathcal{W}_f x$, or simply $f(x)$, such that (x, y) is in the graph.

```
Definition is_function f :=
```

```
    is_correspondence f & fgraph (graph f) & source f = domain (graph f).
Definition W x f := V x (graph f).
Lemma inc_W_target: forall f x, is_function f -> inc x (source f)
  -> inc (W x f) (target f).
Lemma W_pr: forall f x y, is_function f ->
  inc (J x y) (graph f) -> W x f = y.
Lemma inc_pr1graph_source: forall f x y, is_function f ->
  inc (J x y) (graph f) ->  inc x (source f).
Lemma inc_pr2graph_target: forall f x y, is_function f ->
  inc (J x y) (graph f) ->  inc y (target f).
```

For any two sets A and B, any f of type $A \to B$, there is a function g (in the Bourbaki sense) associated to f; its source A and its target B; it was introduced above as acreate f or $\mathcal{L}f$. Assume $a \in A$. There is i of type A such that $\mathcal{R}_A i = a$, and $f(i)$ is of type B. Thus there is a unique b such that $\mathcal{R}_B f(i) = b$. This is is by definition $g(a)$. Conversely, to any function g, one can associate a function $\mathcal{M}g$, such that $\mathcal{L}\mathcal{M}g = g$. This function is not uniquely defined because if $A' = A$ but is distinct from A, then the two types $A \to B$ and $A' \to B$ are unequal. On the other have, if A is equal to the source of g and B to the target, one can define $\mathcal{M}_{A;B}g$, and this is an inverse of $\mathcal{L}$, in the sense that $\mathcal{L}\mathcal{M}_{A;B}g = g$ and $\mathcal{M}_{A;B}\mathcal{L}f = f$. Using $\mathcal{L}$ and $\mathcal{M}$ sometimes reduces the size of a proof.

```
Lemma acreate_function : forall (A B:Set) (f:A->B),
  is_function (acreate f).
Lemma acreate_W : forall (A B:Set) (f:A->B) (x:A),
  W (Ro x) (acreate f) = Ro (f x).
Definition bcreate1 f (H:is_function f) :=
  fun x:source f => Bo (inc_W_target H (R_inc x)).
Definition bcreate f A B
  (H:is_function f)(Ha:source f =A)(Hb:target f =B) :=
  fun x:A => Bo (W_mapping Ha Hb H (R_inc x)).

Lemma bcreate_inv2: forall f A B
  (H:is_function f) (Ha:source f=A)(Hb:target f=B),
  acreate (bcreate H Ha Hb) = f.
Lemma bcreate_inv3: forall (A B:Set) (f:A->B),
  bcreate  (acreate_function f) (acreate_source f)(acreate_target f) = f.
```

A function is injective if $f(x) = f(y)$ implies $x = y$ (this means that the inverse graph is function), it is surjective is each element of the target is associated to an element of the source (this means that the target is the range of the graph), it is bijective if injective and surjective (so that the inverse correspondence is a function). If $f : A \to B$ is bijective, there is a function g such $f \circ g = I_B$ and $g \circ f = I_A$. The converse holds and g is unique. If f is only injective or surjective, there exists g such that one of the two equalities holds (sometimes $A \neq \emptyset$ is required). If $h = g \circ f$, if f and g are injective (surjective, bijective), so is h; the converse is partially true. Assume that f and g are two functions with the same source; does there exists h such that $f = h \circ g$? If g is bijective, the answer is yes and h is unique. A necessary condition is that if $g(x) = g(y)$ then $f(x) = f(y)$. If g is surjective, the condition

is sufficient and h is unique. Otherwise it suffices to restrict the function to its target. The dual problem is: assume that f and g have the same target; what about $f = g \circ h$. Obviously, the range of f must be a subset of the range of g. If g is injective, then there is always a unique solution. These theorems, and many others are implemented in Gaia.

Two sets are said equipotent when there is a bijection between them. This is an equivalence relation.

```
Definition equipotent x y :=
  exists z, bijective z & source z = x & target z = y.
Lemma equipotent_reflexive: forall x, equipotent x x.
Lemma equipotent_symmetric:  forall a b,
  equipotent a b -> equipotent b a.
Lemma equipotent_transitive: forall a b c,
  equipotent a b -> equipotent b c -> equipotent a c.
```

Consider a mapping f like: $x \mapsto (x, x)$ or $x \mapsto \{x\}$ defined for $x \in E$. It can be converted into a functional graph (denoted by $\mathcal{L}_E f$). If F is any superset of the range of this graph ($E \times E$ or $\mathfrak{P}(E)$ in the example), then there is a unique function with source E, target F and graph $\mathcal{L}_E f$. It it denoted by $\mathcal{L}_{E;F} f$, or 'BL f E F' in Coq.

```
Definition BL f a b := corresp a b (L a f).
Definition transf_axioms f a b :=
  forall c, inc c a -> inc (f c) b.

Lemma bl_function: forall f a b,
  transf_axioms f a b -> is_function (BL f a b).
Lemma bl_W: forall f a b c,
  transf_axioms f a b -> inc c a -> W c (BL f a b) = f c.
Lemma bl_bijective: forall f a b, transf_axioms f a b ->
  (forall u v, inc u a-> inc v a -> f u = f v -> u = v) ->
  (forall y, inc y b -> exists x, inc x a & f x = y) ->
  bijective (BL f a b).
```

3.4 Union and Intersection of a Family of Sets

Consider a function f, with source I, target T and graph X. In some cases, the target of f is irrelevant; in this case the word family is used indifferently for f and X (note that I is both the source of f and the domain of X). To be precise, a family is just a functional graph. The value of f at i (obtained by the Coq function W) or the value of X at ι (obtained by the Coq function V), generally denoted by $f(i)$ and $X(\iota)$, are denoted by f_i and X_ι in the context of families. Bourbaki defines the union, intersection, product (and later on, disjoint union, cardinal product, etc.) of a family, and instead of denoting it by $P(X)$, the strange notation $\mathop{\mathrm{P}}_{\iota \in I} X_\iota$ will be used. In order to save space, this generally rendered as $\mathrm{P}_{\iota \in I}\, X_\iota$.

Note the use of the Greek letter iota. This is an attempt to assign a type to some sets. There is no rule, but one can notice the following conventions. A

correspondence is denoted by a Greek letter Γ, but by a Latin lower case letter as f when it is a function. A cardinal is denoted by a Greek letter α, but by a Latin letter n when it is finite. A set is denoted by a German letter like $\mathfrak{G}$ when its elements are considered as sets (i.e., when one writes $x \in Y$ and $Y \in \mathfrak{G}$); an example being the powerset $\mathfrak{P}(X)$. In the case of $\iota \in I$ or $x \in X_\iota$, a lower case letter is used on the left, an upper case letter on the right of the operator. Greek letters are used for indices (remember that the two elements of the canonical doubleton are called α and β); this explains why the index set is never called J. We try, in our documentation, to use notations similar to those of Bourbaki, but attach absolutely no importance to the name of dummy variables. For us a letter is a letter, be it Latin, Greek or German, uppercase or lowercase, bold or italic.

If X is a set, the union $\bigcup X$ is formed of all x such that $x \in y$ for some $y \in X$. In the case of a family, the union, written as $\bigcup_{i\in I} X_i$, is the set of all x such that $x \in X_i$ for some $i \in I$. In some cases, it is interesting to have typed objects, so that we define the union for a map $I \to \mathsf{Set}$. In some cases, we use a map $\mathsf{Set} \to \mathsf{Set}$. This makes four definitions of the union. They are all equivalent; for instance $\bigcup X$ is the union of the identity family on X. Most theorems will use unionb, since it has only one argument. The disjoint union of the family is $\bigcup_{i\in I}(X_i \times \{i\})$; it is also called the sum, and is used for defining the sum of cardinals.

The case of intersection is interesting. Bourbaki [Bou68, Def. 3, page 91] defines the intersection of a family $(X_i)_{i\in I}$ of subsets of E as the set of all elements of E that belong to each X_i. If I is empty, the intersection is E, otherwise it is independent of E, and the definition agrees with Definition 2 (same page) that uses the axiom of choice. In the case of intersection of sets of sets, C. Simpson used rep X instead of E (this is not a superset of all X_i, but a superset of the intersection). We cannot use $X_{\mathsf{rep(I)}}$ in the definition that follows, since the type I might be empty. We successively used: a proof that I is non-empty, a test (is I empty?), a witness i of type I. In the current version, the intersection is simply the subset of the union that contains all elements that belong to each element of the family:

$$\bigcap_{i\in I} X_i = \Big\{ y \in \bigcup_{i\in I} X_i, \quad \forall i, i \in I \Rightarrow y \in X_i \Big\}.$$

```
Definition intersectiont_V1 (In:E)(f : In->E)(H:nonemptyT In):=
  Zo (f (chooseT_any H)) (fun y : E => forall z : In, inc y (f z)).
Definition intersectiont_V2 (I:Set)(f : I-> Set):=
  by_cases(fun H:nonemptyT I =>
      Zo (f (chooseT_any H)) (fun y => forall z : In, inc y (f z)))
  (fun _:~ nonemptyT I => emptyset).
Definition intersectiont_V3 (I:Set)(i:I) (f : I->Set):=
  Zo (f i) (fun y => forall z : I, inc y (f z)).

Record Uintegral (I :Set)(f :I ->Set) : Set := {UI_z : In; UI_elt : f UI_z}.
Definition uniont (I:Set)(f : I ->Set) :=
   IM (fun i : Uintegral f => Ro (UI_elt i)).
Definition intersectiont  (I:Set) (f : I->Set):=
  Zo (uniont f) (fun y => forall z : I, inc y (f z)).
```

```
Definition unionf (x:Set)(f: Set->Set) := uniont (fun a:x => f (Ro a)).
Definition unionb (g:Set) := unionf (domain g)(fun a=> V a g).
Definition intersectionf (x:Set)(f: Set->Set):=
  intersectiont(fun a:x => f (Ro a)).
Definition intersectionb (g:Set) :=
  intersectionf (domain g) (fun a=> V a g).
```

We list here some properties. The important difference between union and intersection is that intersection requires sometimes $I \neq \emptyset$.

```
Lemma uniont_rw: forall (I:Set) (f:I->Set) x,
  inc x (uniont f) = exists z,  inc x (f z).
Lemma intersectionb_rw : forall g x,
  nonempty g ->
  inc x (intersectionb g) = (forall i inc i (domain g) -> inc x (V i g)).
Lemma intersectionb_forall : forall g x i,
  inc x (intersectionb g) -> inc i (domain g) -> inc x (V i g).
Lemma unionb_identity: forall x, unionb (identity_g x) = union x.
```

We have $\bigcup_{i\in I} X_i = \bigcup_{j\in J} X_{f(j)}$ whenever f is a function from J onto I (note that injectivity is not required). The union is monotone with respect to I and X_i; there are similar properties for intersection.

```
Lemma unionb_rewrite1: forall f g,
  is_function f -> fgraph g -> range (graph f) = domain g ->
  unionb g = unionb (fcompose g (graph f)).
Lemma union_monotone: forall (I:Set) (f g:I->Set),
  (forall i, sub (f i) (g i)) -> sub (uniont f) (uniont g).
Lemma union_monotone2: forall a b f,
  sub a b -> sub (unionf a f) (unionf b f).
Lemma intersection_monotone2: forall a b f,
  sub a b -> nonempty a -> sub (intersectionf b f) (intersectionf a f).
```

There are many properties, let's just show associativity.

$$\bigcup_{i\in I} X_i = \bigcup_{l\in L}\left(\bigcup_{i\in J_l} X_i\right), \qquad \bigcap_{i\in I} X_i = \bigcap_{l\in L}\left(\bigcap_{i\in J_l} X_i\right) \qquad (\text{if } I = \bigcup_{l\in L} J_l).$$

```
Theorem union_assoc: forall sf sg f g,
  sf = unionf sg g ->
  unionf sf f = unionf sg (fun l => unionf (g l) f).
Theorem intersection_assoc: forall sf sg f g,
  (forall i, inc i sg -> nonempty (g i)) ->
  sf = unionf sg g ->
  intersectionf sf f = intersectionf sg (fun l => intersectionf (g l) f).
```

A covering of X is a family X_i whose union contains X. A partition is a family where the union is X and the sets are mutually disjoint (there is a variant, where the sets are assumed to be non-empty). Assume that f_i is a function defined on each X_i (with target T). There is then a unique function f (with target T) defined

on X such that $f(x) = f_i(x)$ if $x \in X_i$, provided that some compatibility conditions hold. In the case of a partition, no compatibility conditions are required.

```
Definition covering f x :=  fgraph f & sub x (unionb f).
Definition partition_fam f x:=
  fgraph f & mutually_disjoint f & unionb f = x.

Lemma extension_covering1: forall sf f t h,
  (forall i, inc i sf -> function_prop (h i)(f i) t ) ->
  (forall i j, inc i sf -> inc j sf ->
    agrees_on (intersection2 (f i) (f j)) (h i) (h j)) ->
  exists g, function_prop g (unionf sf f) t &
    graph g = (unionf sf (fun i => (graph (h i)))) &
    range(graph g) = (unionf sf (fun i => (range (graph (h i))))) &
    (forall i, inc i sf -> agrees_on (f i) g (h i)).
Theorem extension_partition: forall f x t h,
  partition_fam f x ->
  (forall i, inc i (domain f) -> function_prop (h i) (V i f) t) ->
  exists_unique (fun g => function_prop g x t &
    (forall i, inc i (domain f) -> agrees_on (V i f) g (h i))).
```

This will be needed later: for any x and y, there is a family defined on the canonical doubleton that maps the first element to x and the second to y.

```
Definition variant a x y := (fun z:Set => Yo (z = a) x y).
Definition variantL a b x y := L (doubleton a b) (variant a x y).
Definition variantLc f g:=  variantLc TPa TPb f g.
```

3.5 Product of a Family of Sets

Given a family $(X_i)_{i\in I}$, we shall define (three variants of) the product of the family, denoted by $\prod_{i\in I} X_i$. This is the subset of $\mathfrak{P}(I \times \bigcup X_i)$ formed of all functional graphs x, with domain I, such that the value of x at i belongs to the value of X at i, in short $x_i \in X_i$. If f is a bijection, the product $\prod X_i$ is equipotent to the product $\prod X_{f(i)}$ (recall that for the union, we have equality if f is merely surjective).

```
Definition productb f:=
  Zo (powerset (product (domain f) (unionb f)))
  (fun z => fgraph z & domain z = domain f
    & forall i, inc i (domain f) -> inc (V i z) (V i f)).
Definition producttt (I:Set) (f:I->Set):= productb (gbcreate f).
Definition productf sf f:=  productb (L sf f).
```

The mapping $\prod_{i\in I} X_i \to X_j$ that associates to each $(x_i)_i$ the quantity x_j is called the j-th projection, and denoted by pr_j. If J is a subset of I, the mapping $\prod_{i\in I} X_i \to \prod_{j\in J} X_j$ that associates to each x its restriction to J is denoted by pr_J. We show here that this function is surjective when the sets X_i are non-empty. This can also be restated as: the product of non-empty sets is non-empty and is equivalent

to the axiom of choice. This result could be interpreted as: given two families of non-zero integers indexed by I and J where $J \subset I$; the product over the larger set is greater than (or equal to) the product over the smaller one.

```
Definition pr_i f i:=  BL(V i) (productb f) (V i f).
Lemma pri_W: forall f i x,
  fgraph f -> inc i (domain f) -> inc x (productb f) ->
  W x (pr_i f i) = V i x.
Definition restriction_product f j := productb (restr f j).
Definition pr_j f j :=
  BL (restriction_graph j) (productb f)(restriction_product f j).
Theorem prj_surjective: forall f j,
  fgraph f -> (forall i, inc i (domain f)  -> nonempty (V i f)) ->
  sub j (domain f) -> surjective (pr_j f j).
```

Proving associativity of the product is a bit more difficult than for the union; in fact $\prod X$ is not equal to $\prod(\prod X)$, it is merely equipotent, via the following canonical isomorphism (we assume that the family J_l is a partition of I):

$$f \mapsto (\mathrm{pr}_{J_l} f)_{l\in L}, \qquad \prod_{i\in I} X_i \to \prod_{l\in L} (\prod_{i\in J_l} X_i).$$

Let $((X_{l,i})_{i\in J_l})_{l\in L}$ be a family of families of sets. Let $I = \prod J_l$. Then

$$\bigcup_{l\in L} (\bigcap_{i\in J_l} X_{l,i}) = \bigcap_{f\in I} (\bigcup_{l\in L} X_{l,f(l)}).$$

$$\prod_{l\in L} (\bigcup_{i\in J_l} X_{l,i}) = \bigcup_{f\in I} (\prod_{l\in L} X_{l,f(l)}).$$

We say that one operation is *distributive* over the other. There are similar formulas obtained by exchanging union and intersection. However, the next formula is invalid for union.

$$(\prod_{i\in I} X_i) \cap (\prod_{i\in I} Y_i) = \prod_{i\in I} (X_i \cap Y_i), \qquad (\bigcap_{i\in I} X_i) \times (\bigcap_{i\in I} Y_i) = \bigcap_{i\in I} (X_i \times Y_i).$$

```
Theorem distrib_prod_intersection: forall f,
  fgraph f -> nonempty (domain f) ->
  (forall l, inc l (domain f) -> fgraph (V l f)) ->
  (forall l, inc l (domain f) -> nonempty (domain (V l f))) ->
  productf (domain f) (fun l => intersectionb (V l f)) =
  intersectionf (productf (domain f) (fun l => (domain (V l f))))
  (fun g => (productf (domain f) (fun l => V (V l g) (V l f)))).
```

Consider a family $(X_i)_{i\in I}$. If I is empty, then the union is empty, the intersection is undefined, and the product has a single element (the empty graph). If I has a single element a, then the union and intersection is I_a, and the product is equipotent to X_a. If I has two elements, a and b, then the union and intersections are $X_a \cup X_b$,

$X_a \cap X_b$, and the product is equipotent to $X_a \times X_b$. Given two sets X and Y, the set $X \times Y$ is equipotent in many ways to a product of the form $\prod(X_i)_{i\in I}$. When I is the canonical doubleton, we call this the canonical product of two sets. If we identify equipotent sets, we get the notion of *cardinal*. We can restate our result as: the cardinal product of two sets is the cardinal product of a family of two sets. The distributivity formulas shown above, applied to the case of two elements, gives $a \times (b + c) = (a \times b) + (a \times c)$.

```
Definition product2 x y :=
  productf two_points (variant TPa x y).
Definition product2_canon x y :=
  BL (fun z => (Lvariantc (P z) (Q z)))  (product x y) (product2 x y).
Lemma product2_canon_bijective: forall x y,
  bijective (product2_canon x y).
```

The set of all graphs of functions from E to F is denoted by F^E: this is a subset of the powerset of $E \times F$. It is the product of the constant family F over the index set E, thus the notation. The set of all functions, namely the set of triples (G, E, F) where $G \in F^E$, is denoted by $\mathcal{F}(E; F)$. These sets are canonically equipotent. We have canonical bijections from $\mathcal{F}(B \times C; A)$ into $\mathcal{F}(B; \mathcal{F}(C; A)$ or $\mathcal{F}(C; \mathcal{F}(B; A))$. Given $f \in \mathcal{F}(E; F)$, we construct $f' \in \mathcal{F}(E'; F')$ via $f' = v \circ f \circ u$, provided that u is a function from E' to E and v is a function from F into F'. If u and v are bijections, the two sets are isomorphic (see diagram below). Assume that X_i, Y_i and f_i are families with the same index I, f_i maps X_i into Y_i. If $x \in \prod X_i$ then mapping $y : i \mapsto f_i(x_i)$ is in $\prod Y_i$. This induces a function $\prod X_i \to \prod Y_i$ called the *extension* of the functions f_i. The two sets $(\prod X_i)^E$ and $\prod X_i^E$ are equivalent.

$$\begin{array}{ccc} E & \xrightarrow{f} & F \\ {\scriptstyle u}\uparrow & & \downarrow{\scriptstyle v} \\ E' & \xrightarrow[f']{} & F' \end{array} \qquad \begin{array}{ccc} X_i & \xrightarrow{f_i} & Y_i \\ {\scriptstyle \mathrm{pr}_i}\uparrow & & \uparrow{\scriptstyle \mathrm{pr}_i} \\ \prod X_i & \xrightarrow[(f_i)_{i\in I}]{} & \prod Y_i \end{array}$$

4. RELATIONS

A binary predicate r (of type Set → Set → Prop, denoted EEP in the definitions that follow) is called a relation. If $r(x, y)$ holds, then x and y are said *related* by r. If G is a graph, then x and y are said related by G if $(x, y) \in G$. This is sometimes denoted by $x \prec_G y$ or $x \sim_r y$, or without subscripts as $x \prec y$ or $x \sim y$.

Relations may be reflexive, symmetric, transitive, etc. Note that reflexive means that if x and y are related by r, then they are related to themselves; a symmetric and transitive relation is reflexive. A relation is said reflexive in a set E if every element of E is related to itself.

```
Definition reflexive_r (r:EEP) x :=
  forall y, inc y x = r y y.
Definition reflexive_rr r:EEP :=
  forall x y, r x y -> (r x x &  r y y).
```

```
Definition symmetric_r (r: EEP) :=
  forall x y, r x y -> r y x.
Definition antisymmetric_r (r: EEP) :=
  forall x y, r x y -> r y x -> x = y.
Definition transitive_r (r: EEP) :=
  forall x y z, r x y -> r y z -> r x z .
```

A relation is said to be an *equivalence* if it is symmetric and transitive, it is called an *order* if it is reflexive, antisymmetric and transitive. An equivalence is generally denoted by $\sim$ and an order by $\prec$. We say that r is an equivalence of E, or an order on E, if it is moreover reflexive on E.

```
Definition equivalence_r (r:EEP) :=
  symmetric_r r & transitive_r r.
Definition order_r(r:EEP) :=
  transitive_r r &  antisymmetric_r r & reflexive_rr r.
Definition order_re  (r:EEP) x :=
  order_r r & reflexive_r r x.
Definition equivalence_re (r:EEP) x :=
  equivalence_r r & reflexive_r r x.
```

Given a relation r and a set E we define r_E as the set of pairs of E related by r. This is a graph. Recall that the substrate of the graph is the set of all x and y that are related by the graph; obviously the substrate of r_E is a subset of E. Equality holds if r is reflexive on E. We say that G is the graph of r if two elements are related by r if and only if they are related by G. If such a set exists, we say that r has a graph. In this case, if E is the substrate of G, then r_E is G. For any set E, the relation "$x \in E$ and $y \in E$ and $r(x, y)$" has a graph.

```
Definition graph_on (r:EEP) x :=
   Zo(product x x)(fun w => r (P w)(Q w)).
Lemma is_graph_graph_on: forall r x,
  is_graph(graph_on r x).
Lemma substrate_graph_on: forall r x,
  sub (substrate (graph_on r x)) x.
```

If G is a graph, we say that it is an order or an equivalence (it satisfies order or is_equivalence) if the relation $(x, y) \in G$ is an order relation (satisfies order_r or is_equivalence_r). The relation is moreover reflexive on the substrate of G. Given an order of equivalence relation r and a set E, the graph of "$x \in E$ and $y \in E$ and $r(x, y)$" is an order or equivalence.

```
Definition is_equivalence r: Set :=
  is_reflexive r & is_transitive r &  is_symmetric r.
Definition order (r: Set) :=
  is_reflexive r & is_transitive r & is_antisymmetric r.

Lemma order_from_rel: forall r x,
  order_r r -> order (graph_on r x).
```

An equivalence or an order can always be considered as an equivalence relation or an order relation (via the relation (x, y) is in the graph), the converse is false; for instance $x \subset y$ is an order relation and "x equipotent to y" is an equivalence relation, they have no graph, since any set would be in the substrate. The relation "x and y are ordinals and x is isomorphic to a subset of y" has no graph (its substrate would be well-ordered, and this is not possible). In the same fashion, the relation "x and y are cardinals and x is equipotent to a subset of y" has no graph. These two relations are linear orders (whenever x and y are related to themselves, then one of them is related to the other).

Assume that r is an equivalence relation with a graph G, and thus a substrate E. If y is related to x, then $y \in E$. Thus one can consider the set of all y related to x, and call it the *class* of x. If r has no graph, this might be false. For instance, if x is a set, there is no set containing all y equipotent to x. For this reason the cardinal of x is defined to be $\tau_y(\mathrm{Eq}(x, y))$, where Eq stands for "is equipotent"; it has the form: there exists a bijection between x and y (see the introduction, where we show a part of the expansion of Eq). In the same vein, given an order x, there is no set containing all y such that y is order isomorphic to x. For this reason the ordinal of x is defined by $\tau_y(\mathrm{Ord}(x, y))$, for some Ord. Note the the axiom of choice is not needed for the theory of Ordinals, neither for finite Cardinals; it is required for infinite cardinals (consider for instance the comments about the axiom of choice in the paper of Zermelo in [vH67]).

4.1 Equivalence Relations

We give here an important example of equivalence. Consider a function $f : A \to B$. Then $f(x) = f(y)$ is an equivalence on A, it is called the equivalence associated to f.

```
Definition eq_rel_associated f :=
  fun x => fun y => (inc x (source f) & (inc y (source f)) & (W x f = W y f)).
Definition equivalence_associated f :=
  compose_graph (inverse_graph (graph f)) (graph f).
Lemma related_ea: forall f x y,
  is_function f ->
  related (equivalence_associated f) x y =
  (inc x (source f) & inc y (source f) &  W x f = W y f).
```

Conversely, each equivalence is associated to a function f. It suffices to take for $f(x)$ the class of x, and for the target of f the quotient set. These two quantities are defined here. Given x, we can consider the set A of all $z \in r$ such that $\mathrm{pr}_1 z = a$, and the set B formed of all b of the form $\mathrm{pr}_2 z$ for some $a \in A$. This is the class of x. The quotient set E/r is the set of all classes $C(x)$ where x is in the substrate.

```
Definition class (r x:Set) := fun_image (Zo r (fun z => P z = x)) Q.
Definition quotient r := fun_image (substrate r) (class r).
```

We shall not list all theorems about equivalence relations, just select randomly some properties.

```
Lemma inc_class : forall r x y,
  is_equivalence r -> inc y (class r x) = related r x y.
Lemma related_class_eq1: forall r u v, is_equivalence r ->
  related r u v -> class r u = class r v.
Lemma inc_quotient : forall r x,  is_equivalence r ->
  inc x (quotient r) = is_class r x.
Lemma class_rep : forall r x, is_equivalence r ->
  inc x (quotient r) -> class r (rep x) = x.
```

The quotient set is a partition of the substrate. Conversely, given a partition $(X_i)_i$ of E, the relation: "there is i such that $x \in E_i$ and $y \in E_i$" is an equivalence.

```
Lemma partition_from_equivalence: forall r,
  is_equivalence r ->
  partition(quotient r)(substrate r).
Definition in_same_coset f x y:=
   exists i, inc i (source f) & inc x (W i f) & inc y (W i f).

Definition partition_relation f x :=
  graph_on (in_same_coset f) x.
Lemma partition_is_equivalence: forall f x,
  is_function f -> partition_fam (graph f) x ->
  is_equivalence (partition_relation f x).
```

Consider the following example: E is the set $\mathbf{Z} \times \mathbf{Z}_+$ formed of pairs (x, y). Consider the relation $(x, y) \sim (x', y')$ if $xy' = x'y$. One may define $\mathbf{Q}$ as the quotient of this relation. Let S be the set of elements of E where x and y are coprime. Bourbaki calls this a "system of representatives" for the partition associated to the equivalence (the intersection of S and any class is a singleton). In some sense S and E/r are isomorphic. In the standard library Coq, $\mathbf{Q}$ is defined as E, and we define $\mathbf{Q}$ as S.

```
Definition representative_system s f x :=
   is_function f & partition_fam (graph f) x & sub s x
  & (forall i, inc i (source f) -> is_singleton (intersection2 (W i f) s)).
```

We say that a relation P is compatible with an equivalence if $x \sim y$ implies that $P(x)$ becomes equivalent to $P(y)$. This allows us to define P on the quotient. A trivial example would be $\mathrm{pr}_1 x > 0$ on E (for the example above). One could then define $x > 0$ on $\mathbf{Q}$ (then define $x > y$ as $x - y > 0$). One could also define "compatible" for a relation with two arguments, and define directly $x > y$ on $\mathbf{Q}$. One says that a function f is compatible with the equivalence if $f(x) = f(y)$ whenever x and y are related. For instance, one can consider the sign of $\mathrm{pr}_1 x$, thus construct a sign function on $\mathbf{Q}$. Given two equivalence relations, one says that f is compatible with both relations if $x \sim y$ implies $f(x) \sim f(y)$. For instance, if f denotes the pair $(-\mathrm{pr}_1 x, \mathrm{pr}_2 x)$, this allows us to define the opposite function on $\mathbf{Q}$. More generally, one can construct $\mathbf{Q}$ from $\mathbf{Z}$ because operations are compatible with the equivalence.

Let's finish this section by showing how one can construct some equivalence relations. If $f : A \to B$ is a function, and there is an equivalence on B then $f(x) \sim f(y)$ is an equivalence on A called the inverse image of the equivalence. A trivial example is when $C \subset B$ and f is the canonical injection. This is called the equivalence induced on C. Assume that a set E has two equivalences S and R where one is finer than the other (this means S implies R). It implies that S may be considered as an equivalence on the quotient set E/R. Finally, given two equivalences on E and E', one can define an equivalence on the product.

4.2 Orders

Remember that an order is a graph G with some properties, and an ordered set is a pair (E, G) where G is an order with substrate E. By abuse of language, one term is often omitted (generally G). The order induced by G on F (for some subset F of E) is the intersection of G with $F \times F$. This notion is more important for orders than for equivalences.

```
Definition induced_order (r a:Set):=
  intersection2 r (coarse a).
Lemma related_induced_order: forall r a x y,
  inc x a -> inc y a ->
  gle (induced_order r a) x y = gle r x y.
Lemma order_induced_order: forall r a,   sub a (substrate r) ->
  order r -> order (induced_order r a).
Lemma induced_order_substrate: forall r,  order r ->
  induced_order r (substrate r) = r.
```

Very often, we consider relations of the form "$x \in A$ and $y \in A$ and $x \subset y$". This is an order on A. For instance, Bourbaki shows that "a function extents another one" or "a partition is finer than another one" are orders by using this property. If f is an injective function, we get an order by replacing $x \subset y$ with $f(x) \subset f(y)$. If the set A is stable by union or intersection, we get a lattice.

```
Definition inclusion_order (a:Set) :=
  graph_on (fun u v => sub u a & sub v a & sub u v) (powerset a).
Definition inclusion_suborder (b:Set) :=
  graph_on (fun u v => inc u b & inc v b & sub u v) b.
Lemma inclusion_is_order: forall a,
  order (inclusion_order a).
Lemma subinclusion_is_order: forall a,
  order (inclusion_suborder a).
```

Given a family of ordered sets, there is a natural order on the product. We give here the definition. Since a function can be identified with its graph, which is an element of a product, we can order functions. We have $f \leq g$ if $\forall x, f(x) \leq g(x)$. This requires an ordering on the target. If we have an ordering on both the source and the target we may consider $\forall x \forall y, x \leq y \Rightarrow f(x) \leq f(y)$. In such a case f is called increasing. If we replace $\Rightarrow$ by $\Leftrightarrow$ and assume f bijective then f will be called an order isomorphism. One can show interesting theorems: for instance

[Bou70, exercise 1.6] defines $\mathcal{A}(E,F)$ as the sets of increasing mappings of E into F (ordered as above) and asks to show that $\mathcal{A}(E,F\times G)$ is isomorphic to the set $\mathcal{A}(E,F)\times\mathcal{A}(E,G)$ and $\mathcal{A}(E\times F,G)$ is isomorphic to set $\mathcal{A}(E,\mathcal{A}(F,G))$. The proof of the whole exercise required nearly 1000 lines of code.

```
Definition product_order_r (g:Set): Set -> Set -> Prop :=
  fun x x' => forall i, inc i (domain g) -> gle (V i g) (V i x) (V i x').
Definition product_order f g:=
  graph_on (product_order_r f g)(productb f).
Definition product_order_axioms g:=
  fgraph g & (forall i, inc i (domain g) -> order (V i g)).

Definition order_isomorphism f r r':=
  (order r) & (order r') &
  (bijective f) & (substrate r = source f) &  (substrate r' = target f) &
  (forall x y, inc x (source f) -> inc y (source f) ->
    gle r x y = gle r' (W x f) (W y f)).
```

Let's introduce here a few basic facts abouts orders. We consider an ordered set E, where the order is denoted $x\leq y$. If $x\leq y$ implies $x=y$ then x is called maximal. If $\forall y$ we have $y\leq x$ then x is called the greatest element; there are similar definitions for minimal, and least element. If $\forall x\in F$ we have $x\leq y$, then y is called the upper bound of F. If the set of upper bounds has a least element it is called the least upper bound. If each doubleton has an upper bound we have a directed set; if each doubleton has a least upper bound and greatest lower bound we have a lattice. For instance, in a set where $x\leq y$ means $x\subset y$, the quantities $x\cup y$ and $x\cap y$ might be the least upper bound and a greatest lower bound. Interesting distributivity property may hold (see the Exercises of [Bou68]). A linear set (or totally ordered set) is such that two elements can always be compared; it is thus a lattice. The set of all a such that $x\leq a$ and $a\leq y$ is called the (closed) interval $[x,y]$; there are other intervals. For instance, the set of all a such that $a<x$ is an interval, called the segment with end-point x, denoted S_x. In a lattice, the intersection of two intervals is an interval. The original proof of this statement was 841 lines long, we reduced it to 150 by noticing that every interval is the intersection of two unbounded intervals.

4.3 Well-ordered Sets

A well-ordering is an ordering $\leq$ on a set E, such that each non-empty subset F of E has a least element for the induced order; it is linearly ordered. The set of rational numbers satisfies: for all a and b, if $a<b$ there exists c such that $a<c<b$, thus is not well-ordered.

```
Definition worder r :=
  order r & forall x, sub x (substrate x) -> nonempty x ->
    exists y, least_element (induced_order r x) y.
```

A segment S is a set such that if $x\in S$ and $y<x$, then $y\in S$. In a well-ordered set, a segment is either the whole set or of the form S_x (the segment with end-point

x). Thus $x \mapsto S_x$ is an order isomorphism between E and the set of all segments of E distinct from E. Thus, the set of segments of a well-ordered set is well-ordered.

```
Definition is_segment r s :=
  sub s (substrate r) & forall x y, inc x s -> gle r y x -> inc y s.
Definition segment r x := interval_uo r x.
Theorem well_ordered_segment: forall r s, worder r -> is_segment r s ->
  s = substrate r \/ (exists x,  inc x (substrate r) &  s = segment r x).
Theorem set_of_segments_worder: forall r, worder r ->
  worder (inclusion_suborder (set_of_segments r)).
```

Consider a family of sets A_i, that form a covering of A. We know how to extent functions defined on each A_i to a function defined on A. We can do the same for relations. If the covering is not a partition we need some compatibility conditions. Assume that for each i and j there is k such that $A_i \subset A_k$ and $A_j \subset A_k$. The compatibility conditions then say that if $A_i \subset A_j$, the restriction to the smaller set of the ordering of the larger set agrees with the ordering of the smaller set. Under these conditions, there is unique extension. More precisely, if orderings are considered as graphs, the unique extension is the union of these graphs.

```
Definition common_extension_order g h:=
  order h & substrate h = unionf (domain g) (fun a =>  (substrate (V a g))) &
  (forall a, inc a (domain g) -> V a g = induced_order h (substrate (V a g))).
Lemma order_merge1: forall g,
  common_extension_order_axiom g -> common_extension_order g (unionb g).
```

With some additional conditions we get ([Bou68, p 149] *Let $(X_\iota)_{\iota\in I}$ be a family of well-ordered sets such that for each pair of indices (ι, κ) one of the sets X_ι, X_κ is a segment of the other. Then there exists a unique ordering on the set $E = \bigcup_{\iota\in I} X_\iota$ which induces the given ordering on each of the X_ι. Endowed with this ordering, E is a well-ordered set. Every segment of X_ι is a segment of E; for each $x \in X_\iota$, the segment with endpoint x in X is equal to the segment with endpoint x in E; and each segment of E is either E itself or a segment of one of the X_ι.* This is how we state this theorem (without uniqueness).

```
Definition common_worder_axiom g:=
  fgraph g &
  (forall x, inc x (domain g) -> worder (V x g)) &
  (forall a b, inc a (domain g) -> inc b (domain g) ->
    is_segment (V a g) (substrate (V b g))
    \/ is_segment (V b g) (substrate (V a g))) &
  (forall a b, inc a (domain g) -> inc b (domain g) ->
    sub (substrate (V a g)) (substrate (V b g)) ->
    V a g = induced_order (V b g) (substrate (V a g))).
Theorem worder_merge: forall g, common_worder_axiom g ->
 ( (common_extension_order g (unionb g)) &
   worder (unionb g) &
   (forall a x, inc a (domain g) -> is_segment (V a g) x
     -> is_segment (unionb g) x ) &
```

```
  (forall a x, inc a (domain g) ->  inc x (substrate (V a g)) ->
    segment (V a g) x = segment (unionb g) x) &
  (forall x, is_segment (unionb g) x ->
    x = substrate (unionb g) \/
      exists a,  inc a (domain g) & is_segment (V a g) x)).
```

Let E be a well-ordered set, T be a subset of $\mathfrak{P}(E)$. Assume T contains only segments, and is stable by union; assume that if $S_x \in T$ then $S_x \cup \{x\} \in T$ (recall that S_x is the set of elements $< x$ while $S_x \cup \{x\}$ is the set of elements $\leq x$). Then all segments are in T (including E). This is called the *transfinite principle*. A weaker form is: let p be a property on E. Consider the set of all x for which p is true on the whole of S_x. The first condition is obviously true. The second reads: if, from $\forall y < x,\ p(y)$, one can deduce $p(x)$, then p holds everywhere. A weaker form is the following: if $p(x)$ implies $p(s(x))$, where $s(x)$ is the successor of x, and if p is true for all elements that are not successors, then p is true everywhere. In particular, on $\mathbf{N}$, we have $s(x) = x + 1$ and zero is the only element that is not a successor.

```
Theorem transfinite_principle: forall r (p:Set-> Prop),
  worder r ->
  (forall x, inc x (substrate r) ->
       (forall y, inc y (substrate r) -> glt r y x -> p y)
     -> p x)
   ->  forall x, inc x (substrate r) -> p x.
```

Given a function g, we denote by g_x the restriction of g to S_x. We consider a functional term p (the definition is a bit fuzzy, all that we need is that if $x = y$, then $p(x) = p(y)$; in the Coq code, p is any function of type Set→Set). Consider the equation $p(g_x) = g(x)$. The previous theorem says that it has at most one solution. The next theorem says that it has one solution. The idea of the proof is the following. Denote by s_x the union $S_x \cup \{x\}$. If g is defined on S_x we can extend it to s_x. We say that x is the successor of y if $S_x = s_y$. If x is not a successor, we first have to give a meaning to g_x; this will be the unique extension of all the restrictions of g to the sets s_y for $y < x$. There is a technical difficulty here. We have shown that such a function exists if all these functions have the same target T. In some cases, we might add the restriction $p(g) \in T$ for all g (this is how we define, by induction, a function $\mathbf{N} \to \mathbf{N}$). Since p might be anything, we drop the condition on the target; we require the function to be surjective, and in our process, all partial functions will be surjective. This makes the proof a bit complicated.

```
Definition transfinite_def r p f:=
  surjective f & source f = substrate r &
  forall x, inc x (substrate r) -> W x f = p (restriction_to_segment r x f).
Theorem transfinite_definition: forall r p,
  worder r -> exists_unique (fun f => transfinite_def r p f).
```

We consider now a similar problem: it consists in finding a well-ordering on a subset F of E, that satisfies $p(S_x) = x$; each segment S_x must belong to a given set $\mathfrak{S}$ that satisfies $p(S) \notin S$ for $S \in \mathfrak{S}$. We want F as large as possible, and express

this as $F \notin \mathfrak{S}$. If $\emptyset \in \mathfrak{S}$, then F is non-empty and has a least element $p(\emptyset)$. For any $x \in F$ if $s_x \in \mathfrak{S}$ then the successor of x is $p(s_x)$, otherwise x is the greatest element. That such an ordering exists will be explained in section 5.2

```
Lemma Zermelo_aux:  forall E s p, sub s (powerset E) ->
  (forall x, inc x s -> inc (p x) E) & ~ (inc (p x) x))->
  exists r, Zermelo_axioms E p s r & (~ (inc (substrate r) s)).
```

As a consequence, every set E is the support of a well-ordering: the axiom of choice asserts that if S is a strict subset of E there is a $p(S)$ in $E - S$. Take for $\mathfrak{S}$ the set of strict subsets; the support of the well-ordering defined by the previous lemma cannot be but E.

```
Theorem Zermelo: forall E, exists r, worder r & substrate r = E.
```

For completeness, we cite Zorn's lemma (every inductive set has a maximal element, which is equivalent to Zermelo's theorem and the axiom of choice). We also list two easy theorems (but whose Coq proofs are rather longish). Write $p(A, B)$ for: there exists a function $f : A \to B$, whose range C is a segment of B (either B or a proper segment S_x), which is an order isomorphism between A and C. The first theorem says that, given two well-ordered sets E and F, either $p(E, F)$ or $p(F, E)$. The second says that if A is a subset of E, we have $p(A, E)$. In all these cases, the order isomorphism (as well as its range) is unique.

```
Definition inductive_set r :=
  forall X, sub X (substrate r) -> total_order (induced_order r X) ->
    exists x, upper_bound r X x.
Theorem Zorn_lemma: forall r, order  r -> inductive_set  r ->
  exists a, maximal_element r a.
Theorem isomorphism_worder: forall r r',
 worder r -> worder r' ->
  let iso:= (fun u v f =>
    is_segment v (range (graph f)) & order_morphism f u v) in
  exists_unique (fun f => iso r r' f) \/ exists_unique (fun f => iso r' r f).
Lemma isomorphic_subset_segment: forall r a,
  worder r -> sub a (substrate r) ->
  exists w, exists f, is_segment r w &
    order_isomorphism f (induced_order r a) (induced_order r w).
```

5. PROOFS

5.1 Proving Theorems in Coq

We have shown in section 2.4 a theorem (transitivity of inclusion) with its proof. Here is a non-trivial proof, showing that a well-ordering is a total ordering.

```
worder_total =
fun (r : Set) (H : worder r) =>
and_ind
  (fun (H0 : order r)
```

```
     (H1 : forall x, sub x C ->
           nonempty x -> exists y , least_element (induced_order r x) y) =>
   conj H0
     (fun (x y) (H2 : inc x C) (H3 : inc y C) =>
      let u := doubleton x y in
      let H4 := sub_doubleton H2 H3 in
      let H5 := nonempty_doubleton x y in
      let e := H1 u H4 H5 in
      ex_ind
        (fun (x0) (H6 : least_element A x0) =>
         and_ind
           (fun (H7 : inc x0 B) (H8 : forall x1, inc x1 B -> gle A x0 x1) =>
            let H9 := eq_ind B (fun P  => inc x0 P) H7 u D in
            let H10 :=
              eq_ind B
                (fun P  => forall x1 , inc x1 P -> gle A x0 x1) H8 u D in
            let H11 := doubleton_second x y in
            let H12 := doubleton_first x y in
            let o := doubleton_or H9 in
            or_ind
              (fun H13 : x0 = x =>
               eq_ind x0 (fun x1  => gle r x1 y \/ gge r x1 y)
                 (or_introl (gge r x0 y)
                    (let g := H10 y H11 in related_induced_order1 g)) x H13)
              (fun H13 : x0 = y =>
               eq_ind x0 (fun y0  => gle r x y0 \/ gge r x y0)
                 (or_intror (gle r x x0)
                    (let g := H10 x H12 in related_induced_order1 g)) y H13)
              o) H6) e)) H
     : forall r , worder r -> total_order r
A== induced_order r u
B== substrate A
C== substrate r
D== substrate_induced_order H0 H4
```

This is how we actually prove the worder_total theorem.

```
Lemma worder_total: forall r, worder r -> total_order r.
Proof. ir. red. ir. red in H. ee. am. ir. set (u:=doubleton x y).
  assert (sub u (substrate r)). uf u. app sub_doubleton. assert (nonempty u).
  uf u. app nonempty_doubleton. nin (H0 _ H3 H4). red in H5.
  ee. awii H5.  awii H6. ufi u H5.
  assert (inc y u). uf u. fprops.  assert (inc x u). uf u. fprops.
  nin (doubleton_or H5); wr H9 ;
  [left; set (H6 _ H7) | right; set (H6 _ H8) ] ;
  app (related_induced_order1 g).
Qed.
```

One may compare this with the Bourbaki proof: "A well-ordered set E is totally ordered because every subset $\{x, y\}$ of E has a least element." The proof makes reference to assumptions denoted H0, H1, H2, etc, that were automatically generated

by Coq. In Gaia version 2, we use the *ssreflect* style, and assumptions names xu, yu are explicit. This proof makes use of three tactics: fprops that tells Coq to try to apply theorems of the fprops data base, aw that tells Coq to try to use rewriting rules of the aw data base, and ex_tac that proves $u \neq \emptyset$ by using yu (which is $y \in u$). The new proof is a bit longer than the previous one. For instance, instead of 'uf u' we use 'rewrite /u' (this unfolds u). Instead of ir (a short-hand for intros) we use move=>x y xs ys. Moreover, the proof uses more lines, as they are shorter (each line contains exactly one period). The proof is as follows: for any x and y, if $u = \{x, y\}$, this is a non-empty subset of the substrate of r. Thus, there is z such that $z \in u$ and z is the least element of u. In particular $z \leq x$ and $z \leq y$. Since $z \in u$ we have $z = x$ or $z = y$, and rewriting this equality in the two previous inequalities proves the result.

```
Lemma worder_total: forall r, worder r -> total_order r.
Proof.
move=> r [or wo]; split=>//.
move=> x y xs ys.
set (u:=doubleton x y).
have xu: (inc x u) by rewrite /u; fprops.
have yu: (inc y u) by rewrite /u; fprops.
have ur: sub u (substrate r) by rewrite /u; apply sub_doubleton.
have si: substrate (induced_order r u) = u by aw.
have neu: nonempty u by ex_tac.
move: (wo _ ur neu)=> [z [zu zle]].
rewrite si in zu zle.
move: (zle _ xu)(zle _ yu); aw=> p1 p2.
case (doubleton_or zu) =><-; auto.
Qed.
```

5.2 A Commented Proof

In 1904, Zermelo presented a paper titled "Proof that every set can be well-ordered" (reprinted in [vH67], pages 139-141). The main assumption is that any non-empty set A has a "distinguished" element, say $\gamma(A) \in A$. Zermelo calls γ-set any well-ordered subset M of E such that, for all $a \in M$, we have $\gamma(E - S_a) = a$. He first notices that, given two γ-sets M and M', one is a segment of the other. In particular $a \leq_M b$ is equivalent to $a \leq_{M'} b$ if a and b are in $M \cap M'$.

Thus, for any family of γ-sets, there is an extension of the ordering of the elements of the family to their union U, which is a well-ordering on U (theorem worder_merge), and makes U a γ-set. Assume $U \neq E$; we can extend U by adjoining a greatest element, namely $\gamma(E - U)$. This remains a γ-set, absurd. Thus E is well-ordered.

The Coq implementation requires 400 lines of proof (including sublemmas). In 1908, Zermelo published "a new proof of the possibility of a well-ordering" (reprinted in [vH67], 183-198), with a discussion of the axiom of choice. The interesting point is that "it presupposes no specific theorems of set theory"; his two proofs have nearly the same same size (60 lines), and the implementation in Coq of the new proof is only 150 lines long.

The theorem states (using modern notations, Ω instead of M and E instead of M): "for any set E, and choice function ϕ, there exists a unique set $\Omega \subset \mathfrak{P}(E)$ such that, for all $P \subset E$, there is unique $P_0 \in \Omega$ such that $P \subset P_0$ and $\phi(P_0) \in P$. The set E is well-ordered by Ω."

Consider a well-ordering $\leq$ on E, and let $\phi(P)$ be the least element of P. Denote by $\mathfrak{R}(a)$ the remainder of a, i.e., the set of all x such that $a \leq x$, and by Ω the set of all remainders. The least remainder containing P is $d(P) = \bigcap\{R \in \Omega, P \subset R\}$. It has the same least element as P, thus $\phi(d(P)) = \phi(P)$, and $d(P) = \mathfrak{R}(\phi(P))$. The set Ω satisfies the assumptions of the theorem, with $P_0 = d(P)$.

Conversely, given Ω, we can define $\mathfrak{R}(a)$ as the P_0 associated to $P = \{a\}$, and the relation $b \in \mathfrak{R}(a)$, denoted $a \leq_\Omega b$. The last sentence of Zermelo's theorem has to be understood as "the graph of $\leq_\Omega$ is a well-ordering on E" (Using the set of remainders is a clever way to represent an ordering as a set, in the absence of the notion of graph; ordered pairs have been introduced by Wiener in 1914 as a "simplification of the logic of relations" in the Principia Mathematica, [vH67, p. 224-227]).

We start with some definitions. Recall that rep is the representative of a set, it plays the role of ϕ. The quantity $X - \{\phi(X)\}$ is denoted by $p(X)$. A subset Θ of $\mathfrak{P}(M)$ that (1) contains M, (2) contains $p(A)$ if it contains A, (3) contains $\bigcap A$ if it contains each element of A is called a Θ-chain. The property that A is a subset or a superset of all elements of Ω is denoted by $m(A)$. The quantity d is defined as above, and $\mathfrak{R}(x) = d(\{x\})$.

```
Lemma Zermelo_bis: forall E, exists r, worder r & substrate r = E.
Proof.
move=> E.
set (p := fun a => complement a (singleton (rep a))).
set (chain:= fun F => sub F (powerset E) & inc E F &
    (forall A, inc A F -> inc (p A) F)
    & (forall A, sub A F -> nonempty A -> inc (intersection A) F)).
set (om:= intersection (Zo (powerset (powerset E)) chain)).
set (m:= fun a => forall x, inc x om -> sub x a \/ sub a x).
set (d:= fun p => intersection (Zo om (fun x => sub p x))).
set (R:= fun x => d (singleton x)).
```

We show that $p(\emptyset) = \emptyset$ and $p(X) \subset X$ for any X. The second statement holds since $p(X)$ is the complement of some set in X. The first statement is equivalent to: $\forall x, x \in p(\emptyset) \rightarrow \mathsf{False}$. Unfolding p and using the property of the complement gives ($x \in \emptyset$ and $x \in A$) $\rightarrow \mathsf{False}$ (for some A). The case tactic transforms this into $x \in \emptyset \rightarrow x \in A \rightarrow \mathsf{False}$. A second call to case replaces $x \in \emptyset$ by $x = \mathcal{R}_\emptyset a$ for some a of type $\emptyset$. A third call to case considers all constructors of $\emptyset$. Since there is none, the result is trivially true.

```
have pe: p emptyset = emptyset.
  by empty_tac x xe; move: xe; rewrite /p; srw; case; case; case.
have sp: forall a, sub (p a) a by move=> t; rewrite /p; apply sub_complement.
```

We show here that the powerset of E is a Θ-chain. We must show that it is a subset of the powerset (trivial), that it contains E (easy), that it is stable by p (this uses transitivity of $\subset$ and property sp) and this it is stable by intersection (this is a bit more complicated, it requires two lines of proof.)

```
have cp:chain (powerset E).
  split; fprops; split; first  by apply powerset_inc; fprops.
  split; first by  move=> A; aw; move=>  AE; apply sub_trans with A.
  move=> A AP [x xA]; move: (AP _ xA); aw.
  move=> xE t tA; exact (xE _ (intersection_forall tA xA)).
```

We now show that Ω is a Θ-chain. This presents no difficulty; the non-trivial point is that one has to show that there is at least one Θ-chain (this follows trivially from cp). By construction, Ω is a subset of every Θ-chain.

```
have co :chain om.
  have aux: (nonempty (Zo (powerset (powerset E)) chain)).
    by exists (powerset E); apply Z_inc; aw; fprops.
  ...
have cio: forall x, chain x -> sub om x.
```

Consider a set A satisfying m. Let F be the set of elements B of Ω satisfying $B \subset p(A)$ or $A \subset B$. Assume $B \in F$; we pretend that $p(B) \in F$. Since $p(B) \subset B$, the result is obvious if $B \subset A$; so assume $A \subset B$. If $\phi(B) \notin A$ we deduce $A \subset p(B)$. Since A satisfies m, it suffices to consider the case $p(B) \subset A$. Now, $\phi(B) \in A$ implies $B \subset A$, thus $A = B$, and this completes the proof. A simpler argument shows that F is stable by intersection, so that F is a Θ-chain thus equal to Ω. One deduces that $p(A)$ satisfies m. Since the set of all $A \in \Omega$ that satisfy m is stable by intersection, it is a Θ-chain, thus Ω. We omit the 50 lines of proof. We shall also omit the proofs of all the statements that follow.

```
have am: om = Zo om m.
```

A consequence of the previous property is that $\subset$ is a total ordering on Ω. The function d satisfies the obvious properties: if $X \subset E$, then $d(X) \in \Omega$, $X \subset d(X)$, $d(X) \subset Y$ whenever $Y \in \Omega$ and $X \subset Y$.

Assume $\phi(d(X)) \notin X$. Then $X \subset p(d(X))$ thus $d(X) \subset p(d(X))$ and $\phi(d(X)) \notin d(X)$. We use here the assumption that ϕ *is a choice function*, i.e., that either $\phi(Z) \in Z$ or Z is empty, with $Z = d(X)$. Since $X \subset d(X)$ we get: whenever X is non-empty, then $\phi(d(X)) \in X$.

```
have st: forall a b, inc a om -> inc b om -> sub a b \/ sub b a.
have dpo: forall q, sub q E -> inc (d q) om.
have pdp: forall q, sub q E -> sub q (d q).
have dpq: forall q r, inc r om -> sub q r -> sub q E-> sub (d q) r.
have rdq: forall q, sub q E -> nonempty q -> inc (rep (d q)) q.
```

Note that $d(X)$ is the least Y such that $\phi(Y) \in X$. The function $\mathfrak{R}$ satisfies: $\mathfrak{R}(x) \in \Omega$, $x \in \mathfrak{R}(x)$ and $\phi(\mathfrak{R}(x)) = x$. In particular, $\mathfrak{R}$ is injective. If $Y \in \Omega$, then $\mathfrak{R}(\phi(Y)) = Y$.

```
have qdp: forall q r, inc r om -> sub q r -> inc (rep r) q -> r = d q.
have Rp: forall x, inc x E ->
    (inc (R x) om & inc x (R x) &  rep (R x) = x).
have Ri:forall x y, inc x E -> inc y E -> R x = R y -> x = y.
have Rrq: forall q, inc q om -> nonempty q -> R (rep q) = q.
```

Write $x \leq y$ instead of $\mathfrak{R}(y) \subset \mathfrak{R}(x)$. Since $\mathfrak{R}$ is injective and Ω is totally ordered by inclusion, this is a total ordering on E. If $x \in \mathfrak{R}(y)$ then $y \leq x$ (hint: assume $x \leq y$; take $q = \{x, y\}$, apply `qdp` with $r = \mathfrak{R}(x)$ and $r = \mathfrak{R}(y)$). Let X be a non-empty subset of E, and $y = \phi(d(X))$. This is some element of X. We have $\mathfrak{R}(y) = \mathfrak{R}(\phi(d(X))) = d(X)$, so that $x \in X$ implies $x \in \mathfrak{R}(y)$, thus $y \leq x$, so that y is the least element of X, and we have a well-ordering.

```
have sRR: forall x y, inc x E -> inc y E -> inc x (R y) -> (sub (R x) (R y)).
set (r := graph_on (fun x y => (sub (R y) (R x))) E).
exists r.
...
Qed.
```

6. CONCLUSION

The work presented here corresponds to pages 65–157 of the theory of sets of Bourbaki. It is split into 6 files, 400 definitions, 1600 theorems, 53 tactics and 18000 lines of code. The code was rewritten in Summer 2010 using the *ssreflect* syntax, and requires 22000 lines of code (the number of characters increased from 715k to 770k, less than 8%). The associated exercises use 10000 lines of code (most exercises of the second chapter are easy, Exercise 5.3 as well a Exercise 5.11.(c) are out of scope (they require properties of the integers, defined only in the third chapter). There are more exercises to the third chapter and they are often harder. Some of them are formally incorrect: for instance $\sum(-1)^i\binom{n}{i} = 0$ is valid over $\mathbf{Z}$ but makes no sense on $\mathbf{N}$, since this is $1 - n + n(n-1)/2 - \ldots$, and partial sums are not naturals. The correct version would be $\sum\binom{n}{2i} = \sum\binom{n}{2i+1}$. We think that is possible to implement almost all exercises, but this will take more time than to implement the main text.

Our work shows that the Bourbaki notion of correspondence is an unnecessary complication: in theory, an order (or an equivalence) is a correspondence, in practice it is just a graph. The other use of correspondences is to define functions; but it suffices simply to say that a function is a triple, formed of a source, a target and a graph.

We are also convinced that functions should not have the same type as a set (Coq has no trouble with objects of type $A \to B$), but the theory of Bourbaki works only if one can consider the set $\mathcal{F}(A; B)$ of functions $A \to B$. This set would have unusual properties: the elements of this set are not sets (in particular they do not satisfy the axiom of extent), and $A \in \mathcal{F}(A; B)$ becomes illegal (ill-typed). Note that the statement is false if one admits the axiom of foundation, so that making it illegal is harmless. It might be interesting to check whether this idea can be turned

into a coherent set of axioms, and in that case, if this offers any advantage over the current implementation of sets and other algebraic structures of Coq.

References

[Ban92] Grzegorz Bancerek. Complete lattices. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/lattice3.html.

[Bar01] Bruno Barras. Sets in Coq, Coq in sets. *Journal of Formalized Reasoning*, 3(1):29–48, 201.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development*. Springer, 2004.

[BM96] Jon Barwise and Lawrence Moss. *Vicious Circles*. Number 60. CLSI Publications, 1996.

[Bou68] N. Bourbaki. *Elements of Mathematics, Theory of Sets*. Springer, 1968.

[Bou70] N. Bourbaki. *Éléments de mathématiques, Théorie des ensembles*. Diffusion CCLS, 1970.

[Bou89] N. Bourbaki. *Elements of Mathematics, Algebra I*. Springer, 1989.

[GM08] Georges Gonthier and Assia Mahboubi. A Small Scale Reflection Extension for the Coq system. Research Report RR-6455, INRIA, 2008. http://hal.inria.fr/inria-00258384/en/.

[Gri09a] José Grimm. Implementation of Bourbaki's Elements of Mathematics i n Coq: Part Two; Ordered Sets, Cardinals, Integers. Research Report RR-7150, INRIA, 2009. http://hal.inria.fr/inria-00440786/en/.

[Gri09b] José Grimm. Implementation of Bourbaki's Elements of Mathematics in Coq: Part One, Theory of Sets. Research Report RR-6999, INRIA, 2009. http://hal.inria.fr/inria-00408143/en/.

[Kri72] Jean-Louis Krivine. *Théorie axiomatique des ensembles*. Presses Universitaires de France, 1972.

[Mat02] Alan Mathias. A term of length 4 523 659 424 929. *Synthese*, 133:75–86, 2002. DOI: 10.1023/A:1020827725055.

[NB89] Bogdan Nowak and Sławomir Białecki. Zermelo's theorem. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/wellset1.html.

[Osb00] M. Scott Osborne. *Basic Homological Algebra*. Springer Verlag, 2000.

[Pau10] Lawrence Paulson. *Isabelle's logics: FOL and ZF*, 2010. http://isabelle.in.tum.de/doc/logics-ZF.pdf.

[Sim04a] Carlos Simpson. Computer theorem proving in math. Technical report, CNRS, Laboratoire J.A. Dieudonne, 2004. arXiv:math/0311260v2.

[Sim04b] Carlos Simpson. Set-theorical mathematics in Coq. Technical report, CNRS, Laboratoire J.A. Dieudonne, 2004. arXiv:math/0402336v1.

[vH67] Jean von Heijenoort, editor. *From Frege to Gödel: a source book in mathematical logic, 1879-1931*. Harvard University Press, 1967.

[Wer97] Benjamin Werner. Sets in types, types in sets. In *Proceedings of TACS'97*, pages 530–546. Springer-Verlag, 1997.