Cut elimination for Zermelo set theory

Gilles Dowek[1] and Alexandre Miquel[2]

[1] École polytechnique and INRIA
LIX, École polytechnique, 91128 Palaiseau Cedex, France
Gilles.Dowek@polytechnique.edu
[2] Université Paris 7,
PPS, 175 Rue du Chevaleret, 75013 Paris, France
Alexandre.Miquel@pps.jussieu.fr

Abstract. We show how to express intuitionistic Zermelo set theory in deduction modulo (*i.e.* by replacing its axioms by rewrite rules) in such a way that the corresponding notion of proof enjoys the normalization property. To do so, we first rephrase set theory as a theory of pointed graphs (following a paradigm due to P. Aczel) by interpreting set-theoretic equality as bisimilarity, and show that in this setting, Zermelo's axioms can be decomposed into graph-theoretic primitives that can be turned into rewrite rules. We then show that the theory we obtain in deduction modulo is a conservative extension of (a minor extension of) Zermelo set theory. Finally, we prove the normalization of the intuitionistic fragment of the theory.

The cut elimination theorem is a central result in proof theory that has many corollaries such as the disjunction property and the witness property for constructive proofs, the completeness of various proof search methods and the decidability of some fragments of predicate logic, as well as some independence results.

However, most of these corollaries hold for pure predicate logic and do not generally extend when we add axioms, because the property that cut-free proofs end with an introduction rule does not generalize in the presence of axioms. Thus, extensions of the normalization theorem have been proved for some axiomatic theories, for instance arithmetic, simple type theory [11, 12] or the so-called stratified foundations [4]. There are several ways to extend normalization to axiomatic theories: the first is to consider a special form of cut corresponding to a given axiom, typically the induction axiom. A second is to transform axioms into deduction rules, typically the β-equivalence axiom. A third way is to replace axioms by computation rules and consider deduction rules modulo the congruence generated by these computation rules [5, 7].

Unfortunately, extending the normalization theorem to set theory has always appeared to be difficult or even impossible: a counter example, due to M. Crabbé [3] shows that normalization does not hold when we replace the axioms of set theory by the obvious deduction rules, and in particular the Restricted Comprehension axiom by a deduction rule allowing to deduce the formula $a \in b \wedge P(x \leftarrow a)$ from $a \in \{x \in b \mid P\}$ and vice-versa. In the same way,

normalization fails if we replace the comprehension axiom by a computation rule rewriting $a \in \{x \in b \mid P\}$ to $a \in b \wedge P(x \leftarrow a)$. Calling C the set $\{x \in A \mid \neg x \in x\}$ the formula $C \in C$ rewrites to $C \in A \wedge \neg C \in C$ and it is not difficult to check that the formula $\neg C \in A$. This counterexample raises the following question: is the failure of normalization an artifact of this particular formulation of set theory, or do all formulations of this theory have a similar property?

More recently, interpretations of set theory in type theory have been proposed [18–20] that follow P. Aczel's "sets as pointed graphs" paradigm [1] by interpreting sets as pointed graphs and extensional equality as bisimilarity. One remarkable feature about these translations is that they express set theory in a framework that enjoys normalization. Another is that although the formulæ $a \in \{x \in b \mid P\}$ and $a \in b \wedge P(x \leftarrow a)$ are provably equivalent, their proofs are different. This suggests that the failure of normalization for set theory is not a property of the theory itself, but of some particular way to transform the axioms into deduction or computation rules.

In the type theoretic interpretation of set theory where sets are translated as pointed graphs, the membership relation $\in$ is no longer primitive, but defined in terms of other atomic relations such as the ternary relation $x \ \eta_a \ y$ expressing that two nodes x and y are connected by an edge in a pointed graph a.

In this paper, we aim at building a theory of pointed graphs in predicate logic—that we call $\mathsf{IZ}^{\mathrm{mod}}$—which is expressive enough to encode set theory in a conservative way. For that, we start from a simple extension of intuitionistic Zermelo set theory (without foundation) called $\mathsf{IZ}^{\mathrm{st}}$, namely, Zermelo set theory with the axioms of Strong Extensionality and Transitive Closure.

Instead of expressing the theory $\mathsf{IZ}^{\mathrm{mod}}$ with axioms, we shall directly express it with computation rules. It is well-known [5] that any theory expressed with computation rules can also be expressed with axioms, replacing every computation rule of the form $l \longrightarrow r$ by the axiom $l = r$ when l and r are terms, or by the axiom $l \Leftrightarrow r$ when l and r are formulæ. Expressing this theory with rewrite rules instead of axioms makes the normalization theorem harder to prove but is a key element for the cut-free proofs to end with an introduction rule. To prove our normalization theorem, we shall use two main ingredients: reducibility candidates as introduced by J.-Y. Girard [11] to prove normalization for higher-order logic, and the forcing/realizability method, following [4, 6].

1 Deduction modulo

In deduction modulo, the notions of language, term and formula are that of first-order predicate logic. But, a theory is formed with a set of axioms Γ *and a congruence* $\equiv$ defined on formulæ. Such a congruence may be defined by a rewrite systems on terms and on formulæ. Then, the deduction rules take this congruence into account. For instance, the *modus ponens* is not stated as usual

$$\frac{A \Rightarrow B \quad A}{B}$$

as the first premise need not be exactly $A \Rightarrow B$ but may be only congruent to this formula, hence it is stated

$$\frac{C \quad A}{B} \text{ if } C \equiv A \Rightarrow B$$

All the rules of natural deduction may be stated in a similar way. See, for instance, [7] for a complete presentation.

For example, arithmetic can be defined by a congruence defined by the following rewrite rules

$$\begin{array}{cc} 0 + y \longrightarrow y & 0 \times y \longrightarrow 0 \\ S(x) + y \longrightarrow S(x + y) & S(x) \times y \longrightarrow x \times y + y \end{array}$$

and some axioms, including the identity axiom $\forall x\ (x = x)$. In this theory, we can prove that the number 4 is even

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \forall x\ x = x}\text{ axiom}}{\Gamma \vdash 2 \times 2 = 4}\ (x, x = x, 4)\ \forall\text{-elim}}{\Gamma \vdash \exists x\ 2 \times x = 4}\ (x, 2 \times x = 4, 2)\ \exists\text{-intro}$$

Substituting the term 2 for the variable x in the formula $2 \times x = 4$ yields $2 \times 2 = 4$, that is congruent to $4 = 4$. The transformation of one formula into the other, that requires several proof steps in usual formulation of arithmetic, is dropped from the proof in deduction modulo.

Deduction modulo allows rules rewriting terms to terms, but also atomic formulæ to arbitrary ones. For instance

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

When we take the rewrite rules above, the axioms of addition and multiplication are not needed anymore as, for example, the formula $\forall y\ 0 + y = y$ is congruent to the axiom $\forall y\ y = y$. Thus, rewrite rules replace axioms.

This equivalence between rewrite rules and axioms is expressed by the *equivalence lemma*, which says that for every congruence $\equiv$ we can find a theory $\mathcal{T}$ such that $\Gamma \vdash A$ is provable in deduction modulo the congruence $\equiv$ if and only if $\mathcal{T}, \Gamma \vdash A$ is provable in ordinary first-order predicate logic [5]. Hence, deduction modulo is not a true extension of predicate logic, but rather an alternative formulation of predicate logic. Of course, the provable formulæ are the same in both cases, but the proofs are very different.

2 Variations on axiomatic Set Theory

In this section, we define the theory $\mathsf{IZ}^{\mathrm{st}}$. This theory is Zermelo set theory extended with two axioms: the Strong Extensionality axiom (which replaces the standard Extensionality axiom of set theory) and the Transitive Closure axiom. In Zermelo-Fraenkel set theory with the Foundation axiom, the Strong

Extensionality axiom can be derived from the Foundation axiom, but it is weaker. Similarly, the Transitive Closure axiom is a consequence of the Replacement scheme, but it is weaker.

Since the theory $\mathsf{IZ}^{\mathrm{st}}$ is expressed in the standard existential way, we also define two conservative extensions of $\mathsf{IZ}^{\mathrm{st}}$ plus a non-conservative extension. The first extension of $\mathsf{IZ}^{\mathrm{st}}$ is a theory called $\mathsf{IZ}^{\mathrm{class}}$ obtained by adding a conservative notion of class *à la* Von Neumann-Bernays-Gödel. The second extension, called $\mathsf{IZ}^{\mathrm{skol}}$, is built from the latter by adding Skolem symbols to denote sets and class, including notations to denote sets and class defined by comprehension. As we shall see in section 4, such a conservative extension of the language of set theory is convenient to define the translation which maps formulæ of the language $\mathsf{IZ}^{\mathrm{mod}}$ of pointed graphs back to the language of set theory.

The final extension, called $\mathsf{IZ}^{\mathrm{skol2}}$, is an extension of $\mathsf{IZ}^{\mathrm{skol}}$ with impredicative classes that will be used in section 5. This extension of $\mathsf{IZ}^{\mathrm{skol}}$ is nothing but a skolemized presentation of second order Zermelo set theory with strong extensionality and transitive closure.

2.1 The theory $\mathsf{IZ}^{\mathrm{st}}$

Definition 1 (The theory $\mathsf{IZ}^{\mathrm{st}}$). *The theory $\mathsf{IZ}^{\mathrm{st}}$ is expressed in predicate logic. Its language is the language of first-order predicate logic formed with two binary predicate symbols $=$ and $\in$, and its axioms are given in Table 1.*

We use the standard abbreviations:

$$\begin{aligned}
a \subseteq b &\equiv \forall x\ (x \in a \Rightarrow x \in b)\\
\mathit{Empty}(a) &\equiv \forall x\ \neg(x \in a)\\
\mathit{Succ}(a,b) &\equiv \forall x\ (x \in b \Leftrightarrow (x \in a \vee x = a))\\
\mathit{Ind}(c) &\equiv \forall a\ (\mathit{Empty}(a) \Rightarrow a \in c) \wedge \forall a\ (a \in c \Rightarrow \forall b\ (\mathit{Succ}(a,b) \Rightarrow b \in c))\\
\mathit{Nat}(a) &\equiv \forall b\ (\mathit{Ind}(b) \Rightarrow a \in b)
\end{aligned}$$

Notice that in $\mathsf{IZ}^{\mathrm{st}}$ the standard formulation of the Extensionality axiom is a consequence of the axiom of Strong Extensionality:

Proposition 1. — *In $\mathsf{IZ}^{\mathrm{st}}$, the following formula is provable:*

$$(\mathit{Extensionality}) \qquad \forall a\ \forall b\ (\forall x\ (x \in a \Leftrightarrow x \in b)\ \Rightarrow\ a = b)\,.$$

Proof. Using the instance of strong extensionality where the formula $R(x,y)$ is $(x = a \wedge y = b) \vee x = y$. □

2.2 A conservative extension with a sort for classes

Definition 2 (The theory $\mathsf{IZ}^{\mathrm{class}}$). *The theory $\mathsf{IZ}^{\mathrm{class}}$ is expressed in many-sorted predicate logic. It has two sorts* Set *and* Class*. Its language is formed with two binary predicate symbols $=$ and $\in$ of rank $\langle \mathrm{Set}, \mathrm{Set}\rangle$ and a binary predicate symbol* mem *of rank $\langle \mathrm{Set}, \mathrm{Class}\rangle$. The axioms of the theory $\mathsf{IZ}^{\mathrm{class}}$ are*

(Reflexivity)	$\forall x\ (x = x)$
(Equ. Compat.)	$\forall x \forall x' \forall y\ (x = x' \wedge x = y \Rightarrow x' = y)$
(Mem. Left Compat.)	$\forall x \forall x' \forall y\ (x = x' \wedge x \in y \Rightarrow x' \in y)$
(Mem. Right Compat.)	$\forall x \forall y \forall y'\ (y = y' \wedge x \in y \Rightarrow x \in y')$
(Strong Extensionality)	$\forall x_1 \cdots \forall x_n \forall a \forall b$ $(R(a,b)$ $\wedge\ \forall x \forall x' \forall y\ (x' \in x \wedge R(x,y) \Rightarrow \exists y'\ (y' \in y \wedge R(x',y')))$ $\wedge\ \forall y \forall y' \forall x\ (y' \in y \wedge R(x,y) \Rightarrow \exists x'\ (x' \in x \wedge R(x',y')))$ $\Rightarrow a = b)$
for each formula $R(x,y)$ whose free variables are among $x_1, \ldots, x_n$, x and y	
(Pairing)	$\forall a \forall b \exists e \forall x\ (x \in e \Leftrightarrow x = a \vee x = b)$
(Union)	$\forall a \exists e \forall x\ (x \in e\ \Leftrightarrow\ \exists y\ (x \in y \wedge y \in a))$
(Powerset)	$\forall a \exists e \forall x\ (x \in e \Leftrightarrow x \subseteq a)$
(Restr. Comprehension)	$\forall x_1 \cdots \forall x_n \forall a \exists e \forall x\ (x \in e \Leftrightarrow x \in a \wedge P(x))$
for each formula $P(x)$ whose free variables are among $x_1, \ldots, x_n$, a and x	
(Infinity)	$\exists e\ \ Ind(e)$
(Transitive closure)	$\forall a \exists e\ (a \subseteq e \wedge \forall x \forall y (x \in y \wedge y \in e \Rightarrow x \in e))$

Table 1. Axioms of the theory $\mathsf{IZ}^{\mathrm{st}}$

– *the axioms of equality of* $\mathsf{IZ}^{\mathrm{st}}$ *and the axiom*

$$\forall x\ \forall y\ \forall p\ (x = y \wedge \mathrm{mem}(x,p) \Rightarrow \mathrm{mem}(y,p))$$

– *the strong extensionality scheme, generalized to all formulæ possibly containing the symbol* mem *and free variables of sort* Class*, but no quantification on classes;*
– *the pairing axiom, the union axiom, the powerset axiom, the axiom of infinity, the axiom of transitive closure;*
– *the restricted comprehension scheme, generalized to all formulæ possibly containing the symbol* mem *and free variables of sort* Class*, but no quantification on classes;*
– *and finally, a* class comprehension scheme

$$\exists \alpha\ \forall x\ (\mathrm{mem}(x,\alpha) \Leftrightarrow P)$$

for each formula P *possibly containing the symbol* mem *and free variables of sort* Class*, but no quantification on classes.*

All the axioms of $\mathsf{IZ}^{\mathrm{st}}$ are axioms of $\mathsf{IZ}^{\mathrm{class}}$, thus $\mathsf{IZ}^{\mathrm{class}}$ is an extension of $\mathsf{IZ}^{\mathrm{st}}$. To prove that this is a conservative extension, we use a notion of intuitionistic model where formulæ are valuated in a Heyting algebra [22], and we prove that for every intuitionistic model of $\mathsf{IZ}^{\mathrm{st}}$ there is an intuitionistic model of $\mathsf{IZ}^{\mathrm{class}}$ validating the same formulæ of the language of $\mathsf{IZ}^{\mathrm{st}}$. Conservativity follows from the correctness and completeness of intuitionistic logic w.r.t. to its Heyting algebra valuated models.

Definition 3. — *Let* $\mathcal{M}$ *be an intuitionistic model of* $\mathsf{IZ}^{\mathrm{st}}$*, whose domain is still written* $\mathcal{M}$ *and whose underlying Heyting algebra is written* B*. A function* E *from* $\mathcal{M}$ *to* B *is said to be* definable *if there exists a formula* P *in the language of* $\mathsf{IZ}^{\mathrm{st}}$ *whose free variables are among* $x, y_1, \ldots, y_n$ *and elements* $b_1, \ldots, b_n$ *of* $\mathcal{M}$ *such that for all* a $[\![P]\!]_{a/x, b_1/y_1, \ldots, b_n/y_n} = E(a)$*.*

Definition 4. — *Let* $\mathcal{M}$ *be a model of* $\mathsf{IZ}^{\mathrm{st}}$*, and consider the structure* $\mathcal{M}'$ *(with the same underlying Heyting algebra* B*) defined as follows:* $[\![\mathrm{Set}]\!] = \mathcal{M}$ *and* $[\![\mathrm{Class}]\!]$ *is the set of definable functions from* $\mathcal{M}$ *to the underlying algebra* B*. The denotation of the symbols* $=$ *and* $\in$ *is the same as in* $\mathcal{M}$*, and the denotation of the symbol* mem *is function application.*

Proposition 2. — *The structure* $\mathcal{M}'$ *is a model of* $\mathsf{IZ}^{\mathrm{class}}$*.*

Proof. To prove that $\mathcal{M}'$ is a model of the class comprehension scheme, of the generalized extensionality scheme and of the the generalized restricted comprehension scheme, we prove that for any formula P containing no quantifiers on variable of the sort Class and assignment ϕ, there exists a formula Q in the language of $\mathsf{IZ}^{\mathrm{st}}$ and an assignment ϕ' such that for all a,

$$[\![P]\!]_{\phi + a/x} = [\![Q]\!]_{\phi' + a/x}$$

We proceed by induction over the structure of P. The only non trivial case is when $P = \mathrm{mem}(x, p)$ where p and x are variables. Then, the object $[\![p]\!]_\phi$ is a definable function from $\mathcal{M}$ to B. Let Q and ϕ' be the defining formula and assignment, for all a, we have

$$[\![P]\!]_{\phi+a/x} = [\![Q]\!]_{\phi'+a/x} \qquad \square$$

Obviously, a formula of the language of $\mathsf{IZ}^{\mathrm{st}}$ has the same denotation in $\mathcal{M}$ and in $\mathcal{M}'$, hence the conservativity of $\mathsf{IZ}^{\mathrm{class}}$ over $\mathsf{IZ}^{\mathrm{st}}$.

2.3 A conservative extension with Skolem symbols

The language of $\mathsf{IZ}^{\mathrm{skol}}$ is the following. Notice that the language of terms expressing sets and classes now contains binding symbols.

Terms	t, u	$::= \; x \mid \bigcup t \mid \{t_1, t_2\} \mid \mathfrak{P}(t) \mid \{x \in t \mid P'\} \mid \mathbb{N} \mid \mathrm{Cl}(t)$
Class terms	T, U	$::= \; X \mid \{\!\!\{x \mid P'\}\!\!\}$
Formulæ	P, Q	$::= \; t = u \mid t \in u \mid \mathrm{mem}(t, T) \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid \forall x\ P \mid \exists x\ P \mid \forall X\ P \mid \exists X\ P$
Restricted formulæ	P', Q'	$::= \; t = u \mid t \in u \mid \mathrm{mem}(t, T) \mid \top \mid \bot \mid P' \wedge Q' \mid P' \vee Q' \mid P' \Rightarrow Q' \mid \forall x\ P' \mid \exists x\ P'$

Definition 5. *We define three transformations:*

- *A transformation on* terms*, which maps each term t of $\mathsf{IZ}^{\mathrm{skol}}$ equipped with a variable z to a formula of $\mathsf{IZ}^{\mathrm{class}}$ written $z \in^\circ t$;*
- *A transformation on* class terms*, which maps each class term T of $\mathsf{IZ}^{\mathrm{skol}}$ equipped with a variable z to a formula of $\mathsf{IZ}^{\mathrm{class}}$ written $\mathrm{mem}^\circ(z, T)$;*
- *A transformation on* formulæ*, which maps each formula P of $\mathsf{IZ}^{\mathrm{skol}}$ to a formula of $\mathsf{IZ}^{\mathrm{class}}$ written P°.*

These transformations are defined by the following equations:

$$
\begin{array}{lcl}
z \in^\circ x & \equiv & z \in x \\
z \in^\circ \bigcup t & \equiv & \exists y\ (z \in y \wedge y \in^\circ t) \\
z \in^\circ \{t_1, t_2\} & \equiv & (z = t_1)^\circ \vee (z = t_2)^\circ \\
z \in^\circ \mathfrak{P}(t) & \equiv & \forall y\ (y \in z \Rightarrow y \in^\circ t) \\
z \in^\circ \{x \in t \mid P'\} & \equiv & z \in^\circ t \wedge P'^\circ(x \leftarrow z) \\
z \in^\circ \mathbb{N} & \equiv & Nat(z) \\
z \in^\circ \mathrm{Cl}(t) & \equiv & \forall x\ [\forall y_1 \forall y_2\ (y_1 \in y_2 \wedge y_2 \in x \Rightarrow y_1 \in x) \wedge \\
 & & \quad \forall y\ (y \in^\circ t \Rightarrow y \in x)\ \Rightarrow\ z \in x] \\
 & & \\
\mathrm{mem}^\circ(z, X) & \equiv & \mathrm{mem}(z, X) \\
\mathrm{mem}^\circ(z, \{\!|x \mid P'|\!\}) & \equiv & P'^\circ(x \leftarrow z) \\
 & & \\
(t = u)^\circ & \equiv & \forall z\ (z \in^\circ t \Leftrightarrow z \in^\circ u) \\
(t \in u)^\circ & \equiv & \exists x\ ((x = t)^\circ \wedge x \in^\circ u) \\
(\mathrm{mem}(t, U))^\circ & \equiv & \exists x\ ((x = t)^\circ \wedge \mathrm{mem}^\circ(x, U)) \\
(P \wedge Q)^\circ & \equiv & P^\circ \wedge Q^\circ \\
 & & \textit{etc.} \\
(\forall x\ P)^\circ & \equiv & \forall x\ P^\circ \\
(\exists x\ P)^\circ & \equiv & \exists x\ P^\circ
\end{array}
$$

Notice that if P is already in the language of $\mathsf{IZ}^{\mathrm{class}}$, then the equivalence $P \Leftrightarrow P^\circ$ is (intuitionistically) provable in $\mathsf{IZ}^{\mathrm{class}}$.

The notion of provability in $\mathsf{IZ}^{\mathrm{skol}}$ is defined by $\mathsf{IZ}^{\mathrm{skol}} \vdash P$ if $\mathsf{IZ}^{\mathrm{class}} \vdash P^\circ$. An equivalent solution would be to define provability in $\mathsf{IZ}^{\mathrm{skol}}$ directly from the expected deduction rules and from the skolemized versions of the axioms of $\mathsf{IZ}^{\mathrm{class}}$.

We shall use the following abbreviations:

$$
\begin{array}{rcl}
\varnothing & \equiv & \{x \in \mathbb{N} \mid \bot\} \\
X \cup Y & \equiv & \bigcup\{X, Y\} \\
\{a\} & \equiv & \{a, a\} \\
\langle a, b\rangle & \equiv & \{\{a\}, \{a, b\}\} \\
\pi_1(x) & \equiv & \bigcup\{x_1 \in \bigcup x \mid \exists x_2\ \ x \equiv \langle x_1, x_2\rangle\} \\
\pi_2(x) & \equiv & \bigcup\{x_2 \in \bigcup x \mid \exists x_1\ \ x \equiv \langle x_1, x_2\rangle\} \\
X \times Y & \equiv & \{z \in \mathfrak{P}(\mathfrak{P}(X \cup Y)) \mid \exists x\ \exists y\ (x \in X \wedge y \in Y \wedge z = \langle x, y\rangle\} \\
0 & \equiv & \varnothing \\
1 & \equiv & \{\varnothing\} \\
f(x) & \equiv & \bigcup\{y \in \bigcup\bigcup f \mid \langle x, y\rangle \in f\} \\
f_{|D} & \equiv & \{c \in f \mid \pi_1(c) \in D\}
\end{array}
$$

2.4 Second-order class quantification

The model construction of section 5 (which is devoted to the normalization proof of $\mathsf{IZ}^{\mathrm{mod}}$) is not done relatively to the theory $\mathsf{IZ}^{\mathrm{skol}}$, but relatively to the extension $\mathsf{IZ}^{\mathrm{skol2}}$ of $\mathsf{IZ}^{\mathrm{skol}}$ in which we drop the restriction on the formulæ that

may be used in set/class comprehension (thus allowing class quantification to appear everywhere in the language).

Of course, $\mathsf{IZ}^{\mathrm{skol2}}$ is definitely not a conservative extension of $\mathsf{IZ}^{\mathrm{skol}}$. Actually, it is a skolemized presentation of second-order Zermelo set theory (extended with Strong Extensionality and Transitive Closure), which is proof-theoretically stronger than $\mathsf{IZ}^{\mathrm{st}}$. However, $\mathsf{IZ}^{\mathrm{skol2}}$ has an obvious extensional model in ZF which is defined by setting

$$[\![\mathrm{Set}]\!] = V_{2\omega} \qquad \text{and} \qquad [\![\mathrm{Class}]\!] = V_{2\omega+1}\,,$$

where (V_α) denotes the cumulative hierarchy (indexed by ordinals).

2.5 Projective classes

Let A be class defined by a formula $A(x)$ with at most one free variable x, and $\phi(x,y)$ a formula with at most two free variables x and y. We say that ϕ is a projection onto A if the following formulæ are provable:

1. $\forall x\ \exists y\ \phi(x,y)$
2. $\forall x\ \forall y\ \forall y'\ (\phi(x,y) \wedge \phi(x,y') \Rightarrow y = y')$
3. $\forall x\ (A(x) \Rightarrow \phi(x,x))$
4. $\forall x\ \forall y\ (\phi(x,y) \Rightarrow A(y))$

A class A is *projective* if there is a projection ϕ onto A. Notice that in classical set theory every nonempty class A is projective, by taking

$$\phi(x,y) \ \equiv\ (A(x) \wedge y = x) \vee (\neg A(x) \wedge y = a)$$

where a is an arbitrary object such that $A(a)$. In intuitionistic set theory, it is not the case anymore. In some case, the formula $\phi(x,y)$ can be written $y = t(x)$ for some term t with at most one free variable x. In this case, conditions 1 and 2 vanish, and conditions 3 and 4 are rephrased as:

$3'$. $\forall x\ (A(x) \Rightarrow t(x) = x)$
$4'$. $\forall x\ A(t(x))$

In what follows, the term $t(x)$ will be written $\lfloor x \rfloor_A^t$, or simply $\lfloor x \rfloor_A$ when the term t is clear in the context.

3 A theory of pointed graphs

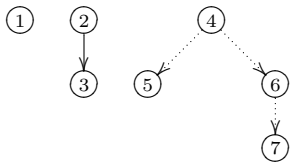
3.1 Informal presentation

The main definition in this paper is the theory $\mathsf{IZ}^{\mathrm{mod}}$ that is a presentation of set theory in deduction modulo with rewrite rules only, *i.e.* with no axioms. At a first glance, the theory $\mathsf{IZ}^{\mathrm{mod}}$ looks more like a theory of pointed graphs, where usual set theoretic notions such as membership and equality are derived notions.

Informally, a pointed graph is just a pair formed with a directed graph and a distinguished node, called the *root* of the pointed graph. In the theory $\mathsf{IZF}^{\mathrm{mod}}$, all pointed graphs share the same nodes, but may have different roots and edges. Thus we have a sort N for nodes and a sort G for pointed graphs. The main symbol of the theory is a ternary predicate symbol η, the formula $x\ \eta_a\ y$ expressing that there is a edge from y to x in the pointed graph a. We also have a function symbol root mapping each pointed graph to its root.

The easiest way to represent a set as a pointed graph is to represent it as a tree whose root is connected to the roots of the trees representing the elements of the set. For instance, the set $\varnothing$ is represented as a pointed graph with no edges. The set $\{\varnothing\}$ is represented as a tree whose root has one child that has no children, etc.

In the figure below, the pointed graph with no edges and root `1` is a representation of the set $\varnothing$, the pointed graph with root `2` and the plain edge is a representation of the set $\{\varnothing\}$ and the pointed graph with root `4` and the dotted edges is a representation of the set $\{\varnothing, \{\varnothing\}\}$.



We extend this idea by considering that any pointed graph represents a set, namely, the set of objects represented by all the pointed graphs obtained by shifting the root one level downwards.

Of course, a set may have several and non-isomorphic representations. For instance, the graph with root `2` and plain edges and the graph with root `6` and dotted edges both represent the set $\{\varnothing\}$. To recover the property of extensionality, we have to define equality in such a way to identify these two pointed graphs. Thus set equality is defined as bisimilarity. Introducing a third sort for binary relations on nodes and a predicate symbol rel (such that $\mathrm{rel}(x, y, r)$ means that x and y are related by the relation r), we then define $a \approx b$ as

$$\begin{array}{l}\exists r\ (\mathrm{rel}(\mathrm{root}(a), \mathrm{root}(b), r)\\ \quad \wedge \quad \forall x \forall x' \forall y\ (x'\ \eta_a\ x \wedge \mathrm{rel}(x, y, r) \Rightarrow \exists y'\ (y'\ \eta_b\ y \wedge \mathrm{rel}(x', y', r)))\\ \quad \wedge \quad \forall y \forall y' \forall x\ (y'\ \eta_b\ y \wedge \mathrm{rel}(x, y, r) \Rightarrow \exists x'\ (x'\ \eta_a\ x \wedge \mathrm{rel}(x', y', r))))\end{array}$$

In deduction modulo, this definition can be handled by introducing a predicate symbol $\approx$ and a rule rewriting the atomic formula $a \approx b$ to this one.

Next, we want to define the membership relation. We first introduce in the language a binary function symbol / associating a pointed graph to each pair formed with a pointed graph and a node. The pointed graph a/x has the same graph as a but its root is x. This is expressed in deduction modulo by the rules

$$\mathrm{root}(a/x) \longrightarrow x \qquad (a/x)/y \longrightarrow a/y$$
$$x\ \eta_{a/z}\ y \longrightarrow x\ \eta_a\ y$$

Now, an object a is a member of a set b if the root of b has a child x in b, such that a is bisimilar to b/x. In deduction modulo, this definition can be handled by introducing a predicate symbol $\in$ and a rule

$$a \in b \longrightarrow \exists x\ (x\ \eta_b\ \mathrm{root}(b) \wedge a \approx (b/x))$$

As equality on pointed graphs is not defined as the smallest substitutive relation, but as bisimilarity, substitutivity has to be proved. In fact, equality is only substitutive with respect to the predicates $\in$ and $\approx$, but not with respect to the symbol "/", for instance. Fortunately, substitutivity with respect to $\in$ and $\approx$ is all we need to prove that $\mathsf{IZ}^{\mathrm{mod}}$ extends $\mathsf{IZ}^{\mathrm{st}}$.

Equality on nodes is defined in a more usual way, introducing a fourth sort for classes of nodes. The comprehension schemes expressing the existence of classes and relations are handled by introducing a function symbol for each formula and the rewrite rules

$$\begin{array}{rcl} \mathrm{mem}(x, g_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n)) & \longrightarrow & P \\ \mathrm{rel}(x, x,', g'_{x,x',y_1,\ldots,y_n,P}(y_1,\ldots,y_n)) & \longrightarrow & P \end{array}$$

Now, we want to build graphs for the usual set theoretic constructions: pairing, union, powerset, restricted comprehension, infinity and transitive closure. Let us take the example of the union. If a is a pointed graph, we want $\bigcup(a)$ to be a pointed graph with a fresh root o related to all the grand children of the root of a. That the root of $\bigcup(a)$ is the node o can be expressed in deduction modulo with the rule

$$\mathrm{root}(\textstyle\bigcup(a)) \longrightarrow o$$

Then, we want the formula $x\ \eta_{\bigcup(a)}\ x'$ to hold if either x and x' are related in the graph a or x' is o and x is a grand child of the root of a. This could be expressed by the naive rule

$$\begin{array}{l} x\ \eta_{\bigcup(a)}\ x' \longrightarrow \\ \quad x\ \eta_a\ x' \vee \exists z\ (x' = o \wedge x\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a)) \end{array}$$

However, with such as rule, we fail to express that the root o must be fresh. If it were already a node of a, for instance, the properties of the set $\bigcup(a)$ would not be as expected. To build the pointed graph $\bigcup(a)$, we must first *relocate* the graph a in a space where there is no o. This is achieved by introducing in the language a relocation function i, that is injective but not surjective and a node o that is not in the image of i. Then the set $\bigcup(a)$ can be defined by the rule

$$\begin{array}{l} x\ \eta_{\bigcup(a)}\ x' \longrightarrow \\ \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ \vee\ \ (\exists y \exists z\ (x = i(y) \wedge x' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a))) \,. \end{array}$$

The fact that i is injective is expressed in deduction modulo, following [8] by introducing a left inverse i' and the rule

$$i'(i(x)) \rightarrow x$$

To express that o is not in the image of i, we introduce a predicate I that contains the image of i but not o. This is expressed by the rules

$$I(i(x)) \to \top \qquad I(o) \to \bot$$

Some other constructions, such as pairing or powerset, need two relocation functions i and j such that the images of i, j and o are disjoint. To express the axiom of infinity, we also need a copy of arithmetic at the level of nodes, thus we introduce also symbols Nat, 0, S, *Pred*, *Null*, and $<$ and related rules. For the powerset axiom, we need also an injection ρ embedding pointed graphs into nodes.

3.2 The theory $\mathsf{IZ}^{\mathrm{mod}}$

Let us now turn to the formal definition of $\mathsf{IZ}^{\mathrm{mod}}$. The main symbol of this theory is a ternary predicate symbol η, the formula $x\ \eta_a\ y$ meaning that there is a edge from x and y in the pointed graph a.

The sorts of the theory $\mathsf{IZ}^{\mathrm{mod}}$ are the following:

Sort	**Usage**
G	pointed graphs
N	Nodes
C	Classes of nodes
R	Binary relations on nodes

The predicate (`pre`), function (`fun`) and constant (`cst`) symbols with their arities are given in Table 2.

The function symbols $f_{x,y_1,\ldots,y_n,P}$ are defined for each formula P with free variables $x, y_1, \ldots, y_n$ of sort G formed with the predicate symbols $\in$ and $\approx$, and quantifiers on G only. The function symbols $g_{x,y_1,\ldots,y_n,P}$ (resp. $g'_{x,x',y_1,\ldots,y_n,P}$) are defined for each formula P whose free variables are among $x, y_1, \ldots, y_n$ (resp. $x, x', y_1, \ldots, y_n$), with x (resp. x, x') of sort N, formed in the restriction of the language containing all the symbols, except $g_{\ldots}$ and $g'_{\ldots}$. The theory $\mathsf{IZ}^{\mathrm{mod}}$ contains no axioms but the rewrite rules that are given in Table 3.

Example 1. Let $\varnothing = f_{x,y,\neg(x\in y)}(y, y)$.

3.3 Translating $\mathsf{IZ}^{\mathrm{st}}$ into $\mathsf{IZ}^{\mathrm{mod}}$

We prove that $\mathsf{IZ}^{\mathrm{mod}}$ is an extension of set theory. To do so, we define a translation $P \mapsto P^\dagger$ from $\mathsf{IZ}^{\mathrm{st}}$ to $\mathsf{IZ}^{\mathrm{mod}}$ which simply maps $\in$ (of $\mathsf{IZ}^{\mathrm{st}}$) to $\in$ (of $\mathsf{IZ}^{\mathrm{mod}}$) and $=$ (of $\mathsf{IZ}^{\mathrm{st}}$) to $\approx$ (of $\mathsf{IZ}^{\mathrm{mod}}$), the rest of the structure of the formula being preserved. We then prove that $\mathsf{IZ}^{\mathrm{mod}}$ is an extension of $\mathsf{IZ}^{\mathrm{st}}$, in the sense that for any formula ϕ of $\mathsf{IZ}^{\mathrm{st}}$, if $\mathsf{IZ}^{\mathrm{st}} \vdash \phi$, then $\mathsf{IZ}^{\mathrm{mod}} \vdash \phi^\dagger$.

To prove this formula, we first prove that all axioms of $\mathsf{IZ}^{\mathrm{st}}$ are theorems of $\mathsf{IZ}^{\mathrm{mod}}$. We begin with fifty-three elementary lemmas.

General		
η	$\mathtt{pre}(G, N, N)$	Local membership
root	$\mathtt{fun}(G)N$	Root of a pointed graph
$/$	$\mathtt{fun}(G, N)G$	Change the root of a pointed graph
$=$	$\mathtt{pre}(N, N)$	Node equality

Sets and relations on nodes		
mem	$\mathtt{pre}(N, C)$	Node membership
rel	$\mathtt{pre}(N, N, R)$	Node relation
$g_{x,y_1,\ldots,y_n,P}$	$\mathtt{fun}(N^n)C$	Construction of sets of nodes
$g'_{x,x',y_1,\ldots,y_n,P}$	$\mathtt{fun}(N^n)R$	Construction of relations on nodes

Relocations		
o	$\mathtt{cst}\ N$	Distinguished node
i	$\mathtt{fun}(N)N$	First injection
i'	$\mathtt{fun}(N)N$	Left-inverse of i
I	$\mathtt{pre}(N)$	Image of i
j	$\mathtt{fun}(N)N$	Second injection
j'	$\mathtt{fun}(N)N$	Left-inverse of j
J	$\mathtt{pre}(N)$	Image of j
0	$\mathtt{cst}\ N$	zero
S	$\mathtt{fun}(N)N$	successor
Pred	$\mathtt{fun}(N)N$	Left-inverse of S
Null	$\mathtt{pre}(N)$	Singleton 0
Nat	$\mathtt{pre}(N)$	Natural number nodes
$<$	$\mathtt{pre}(N, N)$	Strict ordering over nodes
ρ	$\mathtt{fun}(G)N$	Injection from pointed graphs to nodes
ρ'	$\mathtt{fun}(N)G$	Left-inverse of ρ

Equality and membership		
$\approx$	$\mathtt{pre}(G, G)$	Equality as bisimilarity
$\in$	$\mathtt{pre}(G, G)$	Membership as shifted bisimilarity

Constructions		
$\bigcup$	$\mathtt{fun}(G)G$	Construction of the union
$\{\_, \_\}$	$\mathtt{fun}(G, G)G$	Construction of the pair
$\mathfrak{P}$	$\mathtt{fun}(G)G$	Construction of the powerset
$f_{x,y_1,\ldots,y_n,P}$	$\mathtt{fun}(G^n, G)G$	Construction of sets by comprehension
Ω	$\mathtt{cst}\ G$	Pointed graph of Von Neumann numerals
Cl	$\mathtt{fun}(G)G$	Construction of the transitive closure

Table 2. The signature of $\mathsf{IZ}^{\mathrm{mod}}$

General

$$x\ \eta_{a/z}\ y \longrightarrow x\ \eta_a\ y \qquad \mathrm{root}(a/x) \longrightarrow x$$
$$y = z \longrightarrow \forall p\ (\mathrm{mem}(y,p) \Rightarrow \mathrm{mem}(z,p)) \qquad (a/x)/y \longrightarrow a/y$$

Sets and relations on nodes

$$\mathrm{mem}(x, g_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n)) \longrightarrow P$$
$$\mathrm{rel}(x,x',g'_{x,x',y_1,\ldots,y_n,P}(y_1,\ldots,y_n)) \longrightarrow P$$

Relocations

$$i'(i(x)) \to x \quad I(i(x)) \to \top \quad I(j(x)) \to \bot \quad I(o) \to \bot$$
$$j'(j(x)) \to x \quad J(j(x)) \to \top \quad J(i(x)) \to \bot \quad J(o) \to \bot$$
$$Pred(S(x)) \to x \quad Null(0) \to \top \quad Null(S(x)) \to \bot \quad \rho'(\rho(x)) \to x$$
$$Nat(0) \to \top \quad Nat(S(x)) \to Nat(x) \quad x < 0 \to \bot \quad x < S(y) \to x < y \vee x = y$$

Equality and membership

$$\begin{aligned} a \approx b \longrightarrow\ & \exists r\ (\mathrm{rel}(\mathrm{root}(a),\mathrm{root}(b),r) \\ & \wedge\ \forall x \forall x' \forall y\ (x'\ \eta_a\ x \wedge \mathrm{rel}(x,y,r) \Rightarrow \exists y'\ (y'\ \eta_b\ y \wedge \mathrm{rel}(x',y',r))) \\ & \wedge\ \forall y \forall y' \forall x\ (y'\ \eta_b\ y \wedge \mathrm{rel}(x,y,r) \Rightarrow \exists x'\ (x'\ \eta_a\ x \wedge \mathrm{rel}(x',y',r)))) \\ a \in b \longrightarrow\ & \exists x\ (x\ \eta_b\ \mathrm{root}(b) \wedge a \approx (b/x)) \end{aligned}$$

Constructions

$$\begin{aligned} & x\ \eta_{\bigcup(a)}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ (\exists y \exists z\ (x = i(y) \wedge x' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a))) \end{aligned}$$

$$\begin{aligned} & x\ \eta_{\{a,b\}}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ (\exists y \exists y'\ (x = j(y) \wedge x' = j(y') \wedge y\ \eta_b y')) \\ & \vee\ (x = i(\mathrm{root}(a)) \wedge x' = o) \\ & \vee\ (x = j(\mathrm{root}(b)) \wedge x' = o) \end{aligned}$$

$$\begin{aligned} & x\ \eta_{\mathfrak{P}(a)}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ (\exists y \exists c\ (x = i(y) \wedge x' = j(\rho(c)) \wedge y\ \eta_a\ \mathrm{root}(a) \wedge (a/y) \in c)) \\ & \vee\ (\exists c\ (x = j(\rho(c)) \wedge x' = o)) \end{aligned}$$

$$\begin{aligned} & x\ \eta_{f_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n,a)}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ (\exists y\ (x = i(y) \wedge x' = o \wedge y\ \eta_a\ \mathrm{root}(a) \wedge P(x \leftarrow (a/y)))) \end{aligned}$$

$$\begin{aligned} & x\ \eta_{\Omega}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y < y')) \\ & \vee\ (\exists y\ (x = i(y) \wedge x' = o \wedge Nat(y))) \end{aligned}$$

$$\begin{aligned} & x\ \eta_{\mathrm{Cl}(a)}\ x' \longrightarrow \\ & \quad (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ (\exists y\ (x = i(y) \wedge x' = o \wedge \\ & \qquad \forall c\ [\forall z\ (z\ \eta_a\ \mathrm{root}(a) \Rightarrow \mathrm{mem}(z,c)) \wedge \\ & \qquad\quad \forall z\ \forall z'\ ((z\ \eta_a\ z' \wedge \mathrm{mem}(z',c)) \Rightarrow \mathrm{mem}(z,c)) \ \Rightarrow\ \mathrm{mem}(y,c)])) \end{aligned}$$

$$\mathrm{root}(\textstyle\bigcup(a)) \longrightarrow o \qquad \mathrm{root}(\{a,b\}) \longrightarrow o$$
$$\mathrm{root}(\mathfrak{P}(a)) \longrightarrow o \qquad \mathrm{root}(f_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n,a)) \longrightarrow o$$
$$\mathrm{root}(\Omega) \longrightarrow o \qquad \mathrm{root}(\mathrm{Cl}(a)) \longrightarrow o$$

Table 3. Rewrite rules of $\mathsf{IZ}^{\mathrm{mod}}$

Node identity

1. $x = x$
2. $y = z \Rightarrow (P(x \leftarrow y) \Rightarrow P(x \leftarrow z))$ $(*)$

Bisimilarity

3. $a \approx a$
4. $a \approx b \Rightarrow b \approx a$
5. $(a \approx b \wedge b \approx c) \Rightarrow a \approx c$
6. $a \approx (a/\operatorname{root}(a))$

Injectivity and non confusion

7. $S(x) = S(y) \Rightarrow x = y$
8. $\neg 0 = S(x)$
9. $i(x) = i(y) \Rightarrow x = y$
10. $j(x) = j(y) \Rightarrow x = y$
11. $\neg i(x) = o$
12. $\neg j(x) = o$
13. $\neg i(x) = j(y)$

Eta simplification

14. $x\ \eta_{\bigcup(a)}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ y')$
15. $x\ \eta_{\bigcup(a)}\ o \Leftrightarrow \exists y\ \exists z\ (x = i(y) \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \operatorname{root}(a))$
16. $x\ \eta_{\{a,b\}}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ y')$
17. $x\ \eta_{\{a,b\}}\ j(y') \Leftrightarrow \exists y\ (x = j(y) \wedge y\ \eta_b\ y')$
18. $x\ \eta_{\{a,b\}}\ o \Leftrightarrow (x = i(\operatorname{root}(a)) \vee x = j(\operatorname{root}(b)))$
19. $x\ \eta_{\wp(a)}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ y')$
20. $x\ \eta_{\wp(a)}\ j(\rho(c)) \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ \operatorname{root}(a) \wedge (a/y) \in c)$
21. $x\ \eta_{\wp(a)}\ o \Leftrightarrow \exists c\ (x = j(\rho(c)))$
22. $x\ \eta_{f_{x,y_1,\ldots,y_n,P(y_1,\ldots,y_n,a)}}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ y')$
23. $x\ \eta_{f_{x,y_1,\ldots,y_n,P(y_1,\ldots,y_n,a)}}\ o \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ \operatorname{root}(a) \wedge P(x \leftarrow (a/y)))$
24. $x\ \eta_{\Omega}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y < y')$
25. $x\ \eta_{\Omega}\ o \Leftrightarrow \exists y\ (x = i(y) \wedge Nat(y))$
26. $x\ \eta_{\mathrm{Cl}(a)}\ i(y') \Leftrightarrow \exists y\ (x = i(y) \wedge y\ \eta_a\ y')$
27. $x\ \eta_{\mathrm{Cl}(a)}\ o \Leftrightarrow$
    $\exists y\ (x = i(y) \wedge$
    $\forall c\ [\forall z\ (z\ \eta_a\ \operatorname{root}(a) \Rightarrow \operatorname{mem}(z, c)) \wedge$
    $\forall z\ \forall z'\ ((z\ \eta_a\ z' \wedge \operatorname{mem}(z', c)) \Rightarrow \operatorname{mem}(z, c))\ \Rightarrow\ \operatorname{mem}(y, c)])$

$(*)$ Where P is any formula of the language of $\mathsf{IZ}^{\mathrm{mod}}$ that contains no function symbol of the form $g_{\ldots}$ or $g'_{\ldots}$.

Table 4.

Membership

28. $x\ \eta_a\ \mathrm{root}(a) \Rightarrow (a/x) \in a$
29. $a \approx b \Rightarrow \forall x\ (x\ \eta_a\ \mathrm{root}(a) \Rightarrow \exists y\ (y\ \eta_b\ \mathrm{root}(b) \wedge (a/x) \approx (b/y)))$
30. $(a \in b \wedge a \approx c) \Rightarrow c \in b$
31. $(a \in b \wedge b \approx c) \Rightarrow a \in c$

Substitutivity

32. $(P(x \leftarrow a) \wedge a \approx b) \Rightarrow P(x \leftarrow b) \quad (*)$

Bisimilarity by relocation

33. $(\mathrm{root}(b) = i(\mathrm{root}(a)) \wedge \forall x \forall y'\ (y'\ \eta_b\ i(x) \Leftrightarrow \exists x'\ (y' = i(x') \wedge x'\ \eta_a\ x))) \Rightarrow a \approx b$
34. $(\mathrm{root}(b) = j(\mathrm{root}(a)) \wedge \forall x \forall y'\ (y'\ \eta_b\ j(x) \Leftrightarrow \exists x'\ (y' = j(x') \wedge x'\ \eta_a\ x))) \Rightarrow a \approx b$

Embedding

35. $\bigcup(a)/i(y) \approx (a/y)$
36. $(\{a, b\}/i(\mathrm{root}(a))) \approx a$
37. $(\{a, b\}/j(\mathrm{root}(b))) \approx b$
38. $\mathfrak{P}(a)/i(y) \approx (a/y)$
39. $f_{x,y_1,...,y_p,P}(a_1, ..., a_p, b)/i(y) \approx (b/y)$
40. $\mathrm{Cl}(a)/i(y) \approx (a/y)$

Extensionality

41. $P(c, d)$
    $\wedge\ \ (\forall a \forall a' \forall b\ ((a' \in a \wedge P(a, b)) \Rightarrow \exists b'(b' \in b \wedge P(a', b'))))$
    $\wedge\ \ (\forall a \forall b \forall b'\ ((b' \in b \wedge P(a, b)) \Rightarrow \exists a'(a' \in a \wedge P(a', b'))))$
    $\Rightarrow\ \ (c \approx d) \quad (*)$

Finitary existence axioms

42. $c \in \bigcup(a) \Leftrightarrow \exists b\ (c \in b \wedge b \in a)$
43. $c \in \{a, b\} \Leftrightarrow (c \approx a \vee c \approx b)$
44. $a \in \mathfrak{P}(b) \Leftrightarrow \forall c\ (c \in a \Rightarrow c \in b)$
45. $a \in f_{x,y_1,...,y_p,P}(y_1, ..., y_p, b) \Leftrightarrow a \in b \wedge P(x \leftarrow a) \quad (*)$

Infinity

46. $\neg a \in \varnothing$
47. $\varnothing \approx (\Omega/i(0))$
48. $(a \approx (\Omega/i(y))) \Rightarrow \bigcup(\{a, \{a\}\}) \approx (\Omega/i(S(y)))$
49. $\varnothing \in \Omega$
50. $a \in \Omega \Rightarrow \bigcup(\{a, \{a\}\}) \in \Omega$
51. $Ind(\Omega)$

Transitive closure

52. $a \in c \Rightarrow a \in \mathrm{Cl}(c)$
53. $a \in b \Rightarrow b \in \mathrm{Cl}(c) \Rightarrow a \in \mathrm{Cl}(c)$

$(*)$ Where P is any formula expressed in the language $\approx$, $\in$ and where all the quantifiers are of sort G.

Table 5.

Theorem 1. *If* $\mathsf{IZ}^{\mathrm{st}} \vdash P$ *then* $\mathsf{IZ}^{\mathrm{mod}} \vdash P^{\dagger}$.

Proof. We first prove the fifty three easy lemmas of tables 4 and 5, from which we deduce that the axioms of $\mathsf{IZ}^{\mathrm{st}}$ are provable in $\mathsf{IZ}^{\mathrm{mod}}$. We conclude with a simple induction on proof structure. □

3.4 An example

To understand the benefit of using pointed graphs and not directly sets, take an arbitrary set A and consider the set C (built using the restricted comprehension scheme) formed by the elements of A that are not members of themselves. The naive computation rule

$$a \in C \quad \longrightarrow \quad a \in A \wedge \neg a \in a$$

makes the formula $C \in C$ reduce to $C \in A \wedge \neg C \in C$. Consequently, the rewrite system is non terminating, and the underlying proof system is non normalizing too: a simple adaptation of the proof of Russell's paradox yields a non normalizable (but non paradoxical) proof of $\neg C \in A$.

A simple attempt to solve the problem would be to replace the former rule by

$$a \in C \quad \longrightarrow \quad \exists b\ (b = a \wedge (b \in A \wedge \neg b \in b))\,.$$

This way, the atomic formula $C \in C$ would reduce to the formula $\exists b\ (b = C \wedge (b \in A \wedge \neg b \in b))$, and instead of the formula $\neg C \in C$ we would get the formula $\neg b \in b$ (where b is a variable). Using this trick, the rewrite system would be terminating, but we could still build a non normalizable proof using the method of [7]. The reason is that although we know that b is an element of A and that A is structurally smaller than C (since C is built from A), nothing prevents us from substituting an arbitrary term to the variable b in the sub-formula $\neg b \in b$ during some deduction step.

In $\mathsf{IZ}^{\mathrm{mod}}$, in contrast, the formula $C \in C$ reduces to

$$\begin{array}{l}\exists x\ ((\exists y \exists y'\ (x = i(y) \wedge o = i(y') \wedge y\ \eta_A\ y')\ \ \vee \\ \qquad \exists y\ (x = i(y) \wedge o = o \wedge y\ \eta_A\ \mathrm{root}(A) \wedge \neg(A/y) \in (A/y))) \\ \qquad \wedge\ \ C \approx (C/x))\end{array}$$

During this reduction step, we have evolved from $C \in C$ to $\neg(A/y) \in (A/y)$, where A is structurally smaller than C. This breaks the circularity and, in the normalization proof, we shall be able to interpret A first and then C according to this rule.

4 Translating back $\mathsf{IZ}^{\mathrm{mod}}$ into $\mathsf{IZ}^{\mathrm{skol}}$

To complete the proof that $\mathsf{IZ}^{\mathrm{mod}}$ is actually a formulation of set theory, we prove that it is a conservative extension of $\mathsf{IZ}^{\mathrm{st}}$. Since $\mathsf{IZ}^{\mathrm{skol}}$ is itself a conservative extension of $\mathsf{IZ}^{\mathrm{st}}$, all we need to prove is that $\mathsf{IZ}^{\mathrm{mod}}$ is a conservative extension of $\mathsf{IZ}^{\mathrm{skol}}$. This proof is organized in two steps. First, we define a translation $P \mapsto P^*$ from $\mathsf{IZ}^{\mathrm{mod}}$ to $\mathsf{IZ}^{\mathrm{skol}}$ and we prove that if $\mathsf{IZ}^{\mathrm{mod}} \vdash P$ then $\mathsf{IZ}^{\mathrm{skol}} \vdash P^*$. Then, we prove that the formula $P \Leftrightarrow P^{\dagger *}$ is provable in $\mathsf{IZ}^{\mathrm{skol}}$.

4.1 Pointed graphs and reifications

The translation $P \mapsto P^*$ is based on the fact that the notions of pointed graph and bisimilarity can be defined in set theory.

Definition 6 (Pointed graph). *A (directed)* graph *is a set of pairs. A* pointed graph *is a pair $\langle A, a\rangle$ where A is a graph.*

Notice that we do not include a carrier set in our graphs, since the carrier $\overline{A}$ of a graph A can always be reconstructed as

$$\overline{A} = \left\{x \in \bigcup\bigcup A \mid \exists y \;\; \langle x, y\rangle \in A \vee \langle y, x\rangle \in A\right\},$$

whereas the carrier of a pointed graph can be reconstructed as $\overline{\langle A, a\rangle} = \overline{A} \cup \{a\}$. The formula '$A$ is a graph' is

$$\mathrm{Graph}(A) \;\equiv\; \forall c{\in}A \;\exists x \;\exists y \; c = \langle x, y\rangle$$

and the formula 'g is a pointed graph' is

$$\mathrm{Pgraph}(g) \equiv \exists A \;\exists a \;(g = \langle A, a\rangle \wedge \mathrm{Graph}(A)).$$

Definition 7 (Bisimilarity). — *Let $\langle A, a\rangle$ and $\langle B, b\rangle$ be two pointed graphs. A set r is called a* bisimulation *from $\langle A, a\rangle$ to $\langle B, b\rangle$ if*

1. *$\langle a, b\rangle \in r$;*
2. *for all x, x' and y such that $\langle x', x\rangle \in A$ and $\langle x, y\rangle \in r$, there exists y' such that $\langle x', y'\rangle \in r$ and $\langle y', y\rangle \in B$;*
3. *for all y, y' and x such that $\langle y', y\rangle \in B$ and $\langle x, y\rangle \in r$, there exists x' such that $\langle x', y'\rangle \in r$ and $\langle x', x\rangle \in A$.*

Two pointed graphs $\langle A, a\rangle$ and $\langle B, b\rangle$ are said to be bisimilar *if there exists a bisimulation from $\langle A, a\rangle$ to $\langle B, b\rangle$.*

Formally, the formula 'g and g' are bisimilar' is

$$\begin{aligned} g \approx g' \equiv\; & \exists A \exists a \exists B \exists b \exists r \;(\\ & \mathrm{Graph}(A) \wedge \mathrm{Graph}(B) \wedge g = \langle A, a\rangle \wedge g' = \langle B, b\rangle \\ & \wedge \langle a, b\rangle \in r \\ & \wedge \forall x \forall x' \forall y \;((\langle x', x\rangle \in A \wedge \langle x, y\rangle \in r) \Rightarrow \exists y' \;(\langle y', y\rangle \in B \wedge \langle x', y'\rangle \in r)) \\ & \wedge\; \forall y \forall y' \forall x \;((\langle y', y\rangle \in B \wedge \langle x, y\rangle \in r) \Rightarrow (\exists x' \;\langle x', x\rangle \in A \wedge \langle x', y'\rangle \in r))) \end{aligned}$$

In the following definition, we will need a shorthand for 'ϕ is a function'

$$\begin{aligned} \mathrm{Function}(\phi) \quad \equiv \quad & \forall z \;(z \in \phi \;\Rightarrow \exists x \;\exists y \;\; z = \langle x, y\rangle) \;\; \wedge \\ & \forall x \;\forall y \;\forall y' \;(\langle x, y\rangle \in \phi \wedge \langle x, y'\rangle \in \phi \Rightarrow y = y') \end{aligned}$$

as well as terms $\mathrm{Dom}(\phi)$ and $\mathrm{Cod}(\phi)$ defined as

$$\begin{aligned} \mathrm{Dom}(\phi) \quad &\equiv \quad \{x \in \bigcup\bigcup \phi \mid \exists y \;\langle x, y\rangle \in \phi\} \\ \mathrm{Cod}(\phi) \quad &\equiv \quad \{y \in \bigcup\bigcup \phi \mid \exists x \;\langle x, y\rangle \in \phi\} \end{aligned}$$

Definition 8 (Collapse). *A* Mostovski collapse *of a graph A is a function ϕ of domain $\mathrm{Dom}(\phi) = \overline{A}$ such that for any vertex $i \in \mathrm{Dom}(\phi)$ and for any x, we have $x \in \phi(i)$ if and only if there exists $j \in \mathrm{Dom}(\phi)$ such that $\langle j, i\rangle \in A$ and $x = \phi(j)$.*

Formally, the formula 'ϕ is a collapse of A' is

$$\begin{array}{l}\mathrm{Collapse}(A,\phi) \quad \equiv \\ \quad \mathrm{Graph}(A) \wedge \mathrm{Function}(\phi) \wedge \mathrm{Dom}(\phi) = \overline{A} \wedge \\ \quad \forall i\ \forall y'\ \forall y\ [y' \in y \wedge \langle i, y\rangle \in \phi\ \Leftrightarrow \exists i'\ (\langle i', i\rangle \in A\ \wedge \langle i', y'\rangle \in \phi)]\end{array}$$

The collapse of a graph, when it exists, is unique. In ZF, this property is a consequence of the Foundation axiom. However, the weaker Strong Extensionality axiom is sufficient.

Proposition 3. — *The formula*

$$\forall A\ \forall \phi\ \forall \psi\ (\mathrm{Collapse}(A,\phi) \wedge \mathrm{Collapse}(A,\psi) \Rightarrow \phi = \psi)$$

is derivable in $\mathsf{IZ}^{\mathrm{skol}}$.

Proof. Let A be a graph with two collapse functions ϕ and ψ. As a consequence of the instance of Strong Extensionality corresponding to the relation r defined by

$$r(u,v) \quad \equiv \quad \exists i\ (\langle i, u\rangle \in \phi \wedge \langle i, v\rangle \in \psi)$$

we get $x = x'$ for all x, x' and i such that $\langle i, x\rangle \in \phi$ and $\langle i, x'\rangle \in \psi$. □

The domain of the collapse ϕ of a graph A is the carrier $\overline{A}$ of A. We extend it on the whole universe by introducing the notation

$$\hat{\phi}_A(i)\ \equiv\ \{y \in \mathrm{Cod}(\phi) \mid \exists j\ \langle j, i\rangle \in A \wedge \langle j, y\rangle \in \phi\}$$

Proposition 4. — *The following formulæ are provable in* $\mathsf{IZ}^{\mathrm{skol}}$*:*

1. $\forall A\ \forall \phi\ \forall i\ (\mathrm{Collapse}(A,\phi) \wedge i \in \overline{A}\ \Rightarrow\ \phi(i) = \hat{\phi}_A(i))$
2. $\forall A\ \forall \phi\ (\mathrm{Collapse}(A,\phi)\ \Rightarrow\ \forall i\ \forall y\ (y \in \hat{\phi}_A(i)\ \Leftrightarrow\ \exists j\ (\langle j, i\rangle \in A \wedge y = \hat{\phi}_A(j))))$

Proof. 1. Assume that ϕ is a collapse of A and $i \in \overline{A}$. Then, by definition of $\hat{\phi}_A$, we have $\phi(i) = \hat{\phi}_A(i)$.

2. If $\langle j, i\rangle \in A$ then $j \in \overline{A}$, hence by the first part of the proposition, the formula $\langle j, i\rangle \in A \wedge y = \hat{\phi}_A(j)$ is equivalent to $\langle j, i\rangle \in A \wedge y = \phi(j)$. □

Definition 9 (Reification). — *Let $\langle A, a\rangle$ be a pointed graph whose underlying graph has a collapse ϕ. We say that an object x is a reification of $\langle A, a\rangle$ if $x = \hat{\phi}_A(a)$.*

Formally, the formula 'x is a reification of g' is

$$\mathrm{Reif}(g,x) \quad \equiv \quad \exists A\ \exists a\ \exists \phi\ (g = \langle A, a\rangle \wedge \mathrm{Collapse}(A,\phi) \wedge x = \hat{\phi}_A(a))$$

The formula 'g is a reifiable pointed graph is'

$$\mathrm{Rgraph}(g) \equiv \exists x\ \ \mathrm{Reif}(g,x)$$

As an immediate corollary of Prop. 3 we get the following proposition.

Proposition 5. *The formula*

$$\forall g\ \forall x\ \forall y\ ((\mathrm{Reif}(g,x) \wedge \mathrm{Reif}(g,y)) \Rightarrow x = y)$$

is derivable in $\mathsf{IZ}^{\mathrm{st}}$.

Proposition 6. — *The formula*

$$\forall x\ \forall g\ \forall h\ ((\mathrm{Reif}(g,x) \wedge \mathrm{Reif}(h,x)) \Rightarrow\ g \approx h)$$

is derivable in $\mathsf{IZ}^{\mathrm{st}}$.

Proof. Let x be a set, and $g = \langle A, a\rangle$ and $h = \langle B, b\rangle$ be two pointed graphs such that $\mathrm{Reif}(g,x)$ and $\mathrm{Reif}(h,x)$. Assume that ϕ is a collapse of $\langle A, a\rangle$ such that $\phi(a) = x$ and ψ is a collapse of $\langle B, b\rangle$ such that $\psi(b) = x$. We then define the relation r by

$$r = \{\langle y, z\rangle \in \mathrm{Dom}(\phi) \times \mathrm{Dom}(\psi) \mid \hat{\phi}_A(y) = \hat{\psi}_B(z)\}$$

and check that this is a bisimulation of $g = \langle A, a\rangle$ with $h = \langle B, b\rangle$. □

By definition, a reifiable pointed graph has a reification. We prove that, conversely, every set is the reification of some pointed graph. This existence property can be proved with the Replacement Scheme of ZF. However, the weaker Transitive Closure axiom is sufficient.

Proposition 7. *The formula*

$$\forall x\ \exists g\ \ \mathrm{Reif}(g,x)$$

is derivable in $\mathsf{IZ}^{\mathrm{st}}$.

Proof. Let A be the set $\mathrm{Cl}(x) \cup \{x\}$ and r the relation on A defined by $r(u,v)$ if and only if $u \in v$, the set x is the reification of the pointed graph $\langle r, x\rangle$. □

We now want to show that the class Rgraph is projective. To do so, we first have to project any set A to a collapsible graph $G(A)$. Intuitively, the graph $G(A)$ is defined as the largest subgraph of A that has a collapse. This relies on the following definition:

Definition 10 (Initial subgraph).

$$\mathrm{ISeg}(G,A) \quad \equiv \quad \mathrm{Graph}(G) \wedge \forall x\ \forall\ y\ ((\langle x, y\rangle \in A \wedge y \in \overline{G}) \Rightarrow \langle x, y\rangle \in G)$$

Proposition 8. *If* $\mathrm{Collapse}(A,\phi)$ *and* $\mathrm{ISeg}(G,A)$ *then* $\mathrm{Collapse}(G, \phi_{|\overline{G}})$

Proof. Let $\psi = \phi_{|\overline{G}}$. It is routine to check that if $i \in \overline{G}$, then the formulæ

$$\exists j\ (\langle j, i\rangle \in A \wedge \langle j, y\rangle \in \phi) \qquad \text{and} \qquad \exists j\ (\langle j, i\rangle \in G \wedge \langle j, y\rangle \in \psi)$$

are equivalent. Thus, if $i \in \overline{G}$, then we have $y \in \psi(i)$ if and only if $y \in \phi(i)$ if and only if $\exists j\ (\langle j, i\rangle \in A \wedge \langle j, y\rangle \in \phi)$ if and only if $\exists j\ (\langle j, i\rangle \in G \wedge \langle j, y\rangle \in \psi)$. Thus ψ is a collapse of G. □

Proposition 9. *Let A be a graph, and G_1 and G_2 two initial subgraphs of A with collapses ϕ_1 and ϕ_2. Then ϕ_1 and ϕ_2 coincide on $D = \mathrm{Dom}(\phi_1) \cap \mathrm{Dom}(\phi_2)$.*

Proof. Let $G = A \cap (D \times D)$. Notice that $D = \overline{G_1} \cap \overline{G_2}$. It is routine to check that G is an initial subgraph of G_1 (resp. G_2). Let $D' = \overline{G} \subseteq D$. By Prop. 8, $\phi_{1|D'}$ and $\phi_{2|D'}$ are collapses of G hence they are equal by Prop. 3. We now want to prove that ϕ_1 and ϕ_2 coincide on the full set D. Consider an element $i \in D$. We have $y \in \phi_1(i)$ if and only if $\exists j\ \langle j, i\rangle \in G_1 \wedge y = \phi_1(j)$ if and only if $\exists j\ \langle j, i\rangle \in G_2 \wedge y = \phi_2(j)$ if and only if $y \in \phi_2(i)$. The equivalence $\exists j\ \langle j, i\rangle \in G_1 \wedge y = \phi_1(j)$ if and only if $\exists j\ \langle j, i\rangle \in G_2 \wedge y = \phi_2(j)$ is justified by noticing that the proposition $\langle j, i\rangle \in G_1$ and $\langle j, i\rangle \in G_2$ are equivalent when $i \in D$ (since both G_1 and G_2 are initial subgraphs of A), and that in this case, we have $j \in D'$, hence $\phi_1(j) = \phi_2(j)$. □

As an immediate corollary, we get:

Proposition 10. *The union of all the initial subgraphs of a set A that have a collapse has a collapse.*

Definition 11 (Largest collapsible subgraph). — *The largest collapsible subgraph of a set A is given by*

$$G(A) = \bigcup \{G \in \mathfrak{P}(A) \mid \mathrm{ISeg}(G, A) \wedge \exists \psi\ \ \mathrm{Collapse}(G, \psi)\}$$

The projection of any set x onto the class Rgraph of reifiable pointed graphs is then defined as

$$\lfloor x \rfloor_{\mathrm{Rgraph}} \ = \ \langle G(\pi_1(x), \pi_2(x)\rangle$$

4.2 Translation

We are now ready to define a translation from $\mathsf{IZ}^{\mathrm{mod}}$ to $\mathsf{IZ}^{\mathrm{skol}}$. Each sort s of $\mathsf{IZ}^{\mathrm{mod}}$ is interpreted as a sort of $\mathsf{IZ}^{\mathrm{skol}}$ written s_* accompanied with a relativization predicate written $s^*(x)$ (where x is of sort s_*). We take

- $G_* = \mathrm{Set}$, with $G^*(x) \equiv \mathrm{Rgraph}(x)$
- $N_* = \mathrm{Set}$, with $N^*(x) \equiv \top$
- $C_* = \mathrm{Class}$, with $C^*(x) \equiv \top$
- $R_* = \mathrm{Class}$, with $R^*(c) \equiv \forall x\ (\mathrm{mem}(x, c) \Rightarrow \exists y\ \exists z\ (x = \langle y, z\rangle))$

Each term t (resp. formula P) of $\mathsf{IZ}^{\mathrm{mod}}$ is translated as a term t^* (resp. formula P^*) of $\mathsf{IZ}^{\mathrm{skol}}$. These translations are defined by mutual induction in Tables 6 and 7.

Proposition 11. — *If a is a well-formed term of sort s in $\mathsf{IZ}^{\mathrm{mod}}$ with free variables $x_1, \ldots, x_n$ of sorts $s_1, \ldots, s_n$ respectively, then*

$$\mathsf{IZ}^{\mathrm{skol}} \vdash \forall x_1\ \cdots\ \forall x_n\ (s_1^*(x_1) \wedge \cdots \wedge s_n^*(x_n) \Rightarrow s^*(a^*))$$

$$
\begin{array}{lll}
x^* \equiv x & & \\
(\mathrm{root}(a))^* \equiv \pi_2(a^*) & (i(a))^* \equiv \langle 0, a^* \rangle & (j(a))^* \equiv \langle 1, a^* \rangle \\
(a/b)^* \equiv \langle \pi_1(a^*), b^* \rangle & (i'(a))^* \equiv \pi_2(a^*) & (j'(a))^* \equiv \pi_2(a^*) \\
o^* \equiv 0 & S(x)^* \equiv x^* \cup \{x^*\} & (\rho(a))^* \equiv a^* \\
0^* \equiv 0 & \mathit{Pred}(x)^* \equiv \bigcup x^* & (\rho'(a))^* \equiv \lfloor a^* \rfloor_{\mathrm{Rgraph}}
\end{array}
$$

$$(g_{x,y_1,\ldots,y_n,P}(b_1,\ldots,b_n))^* \equiv \{\!|x \mid P^*(y_1 \leftarrow b_1^*, \ldots, y_n \leftarrow b_n^*)|\!\}$$
$$(g'_{x,x',y_1,\ldots,y_n,P}(b_1,\ldots,b_n))^* \equiv \{\!|z \mid \exists x \exists x' \ (z = \langle x, x' \rangle \wedge P^*(y_1 \leftarrow b_1^*, \ldots, y_n \leftarrow b_n^*))|\!\}$$

$(\bigcup(a))^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \overline{a^*}) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(a^*)) \\ \vee \exists y' \exists y \ (c = \langle \langle 0, y' \rangle, 0 \rangle \wedge \langle y', y \rangle \in \pi_1(a^*) \ \wedge \langle y, \pi_2(a^*) \rangle \in \pi_1(a^*))\}$$

$(\{a, b\})^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \overline{a^*}) \cup (\{1\} \times \overline{b^*}) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(a^*)) \\ \vee \exists y \exists y' \ (c = \langle \langle 1, y' \rangle, \langle 1, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(b^*)) \\ \vee c = \langle \langle 0, \pi_2(a^*) \rangle, 0 \rangle \vee c = \langle \langle 1, \pi_2(b^*) \rangle, 0 \rangle\}$$

$(\mathfrak{P}(a))^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \overline{a^*}) \cup (\{1\} \times \mathfrak{P}(\overline{a^*})) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(a^*)) \\ \vee \exists y \exists p \ (c = \langle \langle 0, y \rangle, \langle 1, p \rangle \rangle \wedge \langle y, \pi_2(a^*) \rangle \in \pi_1(a^*) \wedge y \in p) \\ \vee \exists p \ (c = \langle \langle 1, p \rangle, 0 \rangle)\}$$

$(f_{x,y_1,\ldots,y_n,P}(a_1,\ldots,a_n,a))^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \overline{a^*}) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(a^*)) \\ \vee \ \exists y \ (c = \langle \langle 0, y \rangle, 0 \rangle \wedge \langle y, \pi_2(a^*) \rangle \in \pi_1(a^*) \\ \wedge \ P^*(x \leftarrow \langle \pi_1(a^*), y \rangle, y_{1..n} \leftarrow a^*_{1..n}))\}$$

$\Omega^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \mathbb{N}) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge y' \in y) \\ \vee \exists y \ (c = \langle \langle 0, y \rangle, 0 \rangle)\}$$

$(\mathrm{Cl}(a))^* \equiv \langle R, 0 \rangle$ where $X \equiv (\{0\} \times \overline{a^*}) \cup \{0\}$ and
$$R \equiv \{c \in X \times X \mid \exists y \exists y' \ (c = \langle \langle 0, y' \rangle, \langle 0, y \rangle \rangle \wedge \langle y', y \rangle \in \pi_1(a^*)) \\ \vee \exists y \ (c = \langle \langle 0, y \rangle, 0 \rangle \wedge \langle y, \pi_2(a^*) \rangle \in \mathrm{Clos}(\pi_1(a^*)))$$
where $\mathrm{Clos}(r)$ is the term
$$\{c \in \overline{r} \times \overline{r} \mid \forall r' \ (r \subseteq r' \ \wedge \ \forall x \forall y \forall z \ (\langle x, y \rangle \in r' \wedge \langle y, z \rangle \in r' \Rightarrow \langle x, z \rangle \in r') \ \Rightarrow \ c \in r')\}$$

Table 6. Translation of terms

$(t\ \eta_a\ u)^*$	$\equiv$	$\langle t^*, u^*\rangle \in \pi_1(a^*)$
$(t = u)^*$	$\equiv$	$t^* = u^*$
$(\mathrm{mem}(t,p))^*$	$\equiv$	$\mathrm{mem}(t^*, p^*)$
$(\mathrm{rel}(t,u,r))^*$	$\equiv$	$\mathrm{mem}(\langle t^*, u^*\rangle, r^*)$
$(I(t))^*$	$\equiv$	$\exists y\ t^* = \langle 0, y\rangle$
$(J(t))^*$	$\equiv$	$\exists y\ t^* = \langle 1, y\rangle$
$(Null(t))^*$	$\equiv$	$t^* = 0$
$(t < u)^*$	$\equiv$	$t^* \in u^* \wedge u^* \in \mathbb{N}$
$(Nat(t))^*$	$\equiv$	$t^* \in \mathbb{N}$
$(t \approx u)^*$	$\equiv$	$t^* \approx u^*$
$(t \in u)^*$	$\equiv$	$\exists z\ (\langle z, \pi_2(u^*)\rangle \in \pi_1(u^*)\ \wedge t^* \approx \langle \pi_1(u^*), z\rangle)$
$\top^*$	$\equiv$	$\top$
$\bot^*$	$\equiv$	$\bot$
$(A \Rightarrow B)^*$	$\equiv$	$A^* \Rightarrow B^*$
$(A \wedge B)^*$	$\equiv$	$A^* \wedge B^*$
$(A \vee B)^*$	$\equiv$	$A^* \vee B^*$
$(\forall x\ A)^*$	$\equiv$	$\forall x\ (s_*(x) \Rightarrow A^*)$
$(\exists x\ A)^*$	$\equiv$	$\exists x\ (s_*(x) \wedge A^*)$

Table 7. Translation of formulæ

Proof. By induction on the structure of the term a. The only non trivial case is when t is of sort G, in which case we have to check that t^* is a term of sort Set and that the formula $\mathrm{Reif}(t)$ is provable in $\mathsf{IZ}^{\mathrm{skol}}$. If a has the form $\bigcup(b)$, $\{b,c\}$, $\mathfrak{P}(b)$, $f_{x,y_1,\ldots,y_n,P}(b_1,\ldots,b_n,b)$, Ω or $\mathrm{Cl}(a)$, then we just apply the induction hypothesis and prove that the pointed graph built in the translation is reifiable (which needs to use the corresponding axioms of $\mathsf{IZ}^{\mathrm{skol}}$). If a has the form b/x then we have to prove that the pointed graph built in the translation is reifiable which is obvious because reifiability does not depend on the position of the root in the graph. If the term has the form $\rho'(a)$. We have to check that the pointed graph built in the translation is reifiable, and this holds because $G(a)$ is built in order to have a collapse. □

Proposition 12 (Correction of rules). — *If $P \longrightarrow Q$, where the free variables of P are among $x_1,\ldots,x_n$ of sorts $s_1,\ldots,s_n$ respectively, then the formula*

$$\mathsf{IZ}^{\mathrm{skol}} \vdash s_1^*(x_1) \wedge \cdots \wedge s_n^*(x_n)\ \Rightarrow (P^* \Leftrightarrow Q^*)$$

Proof. We check this rule by rule. Let us give a few examples.

– The rule

$$\begin{array}{rl} x\ \eta_{\bigcup(a)}\ x' \longrightarrow & (\exists y\ \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\ & \vee\ \ (\exists y\ \exists z\ (x = i(y) \wedge x' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a))) \end{array}$$

Consider an atomic formula of the form $t\ \eta_{\bigcup(a)}\ t'$ that reduces to

$$\begin{array}{l} (\exists y\ \exists y'\ (t = i(y) \wedge t' = i(y') \wedge y\ \eta_a\ y')) \\ \vee\ \ (\exists y\ \exists z\ (t = i(y) \wedge t' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a))) \end{array}$$

The translation of the formula $t\ \eta_{\bigcup(a)}\ t'$ is $\langle t^*, t'^* \rangle \in \pi_1(\langle R, 0 \rangle)$ where

$$\begin{array}{l} R = \{c \in X \times X \mid \\ \qquad \exists y\ \exists y'\ (c = \langle \langle 0, y \rangle, \langle 0, y' \rangle \rangle \wedge \langle y, y' \rangle \in \pi_1(a^*)) \\ \quad \vee\ \exists y\ \exists z\ (c = \langle \langle 0, y \rangle, 0 \rangle \wedge \langle y, z \rangle \in \pi_1(a^*)\ \wedge \langle z, \pi_2(a^*) \rangle \in \pi_1(a^*))\} \end{array}$$

where $X = (\{0\} \times \overline{a^*}) \cup \{0\}$. This formula is provably equivalent in $\mathsf{IZ}^{\mathrm{skol}}$ to

$$\begin{array}{l} t^* \in X \wedge t'^* \in X \wedge \\ (\exists y\ \exists y'\ (t^* = \langle 0, y \rangle \wedge t'^* = \langle 0, y' \rangle \wedge \langle y, y' \rangle \in \pi_1(a^*)) \\ \vee\ \exists y\ \exists z\ (t^* = \langle 0, y \rangle \wedge t'^* = 0 \wedge \langle y, z \rangle \in \pi_1(a^*) \wedge \langle z, \pi_2(a^*) \rangle \in \pi_1(a^*))) \end{array}$$

that is provably equivalent in $\mathsf{IZ}^{\mathrm{skol}}$ to

$$\begin{array}{l} \exists y\ \exists y'\ (t^* = \langle 0, y \rangle \wedge t'^* = \langle 0, y' \rangle \wedge \langle y, y' \rangle \in \pi_1(a^*)) \\ \vee\ \exists y\ \exists z\ (t^* = \langle 0, y \rangle \wedge t'^* = 0 \wedge \langle y, z \rangle \in \pi_1(a^*) \wedge \langle z, \pi_2(a^*) \rangle \in \pi_1(a^*)) \end{array}$$

that is the translation of

$$\begin{array}{l} (\exists y\ \exists y'\ (t = i(y) \wedge t' = i(y') \wedge y\ \eta_a\ y')) \\ \vee\ (\exists y\ \exists z\ (t = i(y) \wedge t' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a)))\,. \end{array}$$

– The rule

$$y = z \longrightarrow \forall p\ (\mathrm{mem}(y, p) \Rightarrow \mathrm{mem}(z, p))$$

Consider an atomic formula $t = u$ that reduces to

$$\forall p\ (\mathrm{mem}(t, p) \Rightarrow \mathrm{mem}(t, p))\,.$$

The l.h.s. translates to the formula $t^* = u^*$ whereas the r.h.s. translates to

$$\forall p\ (\top \Rightarrow \mathrm{mem}(t^*, p) \Rightarrow \mathrm{mem}(u^*, p))\,.$$

Both formulæ are equivalent in $\mathsf{IZ}^{\mathrm{class}}$.

Proposition 13 (Correction of the translation). — *Let P be a formula of $\mathsf{IZ}^{\mathrm{mod}}$ with free variables $x_1, \ldots, x_n$ of sorts $s_1, \ldots, s_n$ respectively. If $\mathsf{IZ}^{\mathrm{mod}} \vdash P$, then*

$$\mathsf{IZ}^{\mathrm{skol}} \vdash s_1^*(x_1) \wedge \cdots \wedge s_n^*(x_n) \Rightarrow P^*$$

Proof. By induction over proof structure, using Prop. 11 to justify the rules of quantifiers and Prop. 12 to justify conversion steps. □

4.3 Conservative extension

In Section 3.3, we have proved that $\mathsf{IZ}^{\mathrm{mod}}$ was an extension of $\mathsf{IZ}^{\mathrm{st}}$. We are now ready to prove that this extension is conservative.

Proposition 14. — *For any formula $P(x_1, \ldots, x_n)$ of $\mathsf{IZ}^{\mathrm{st}}$, the universal closure of the formula (with free variables $x_1, \ldots, x_n, g_1, \ldots, g_n$)*

$$\bigwedge_{i=1}^{n} \mathrm{Reif}(x_i, g_i) \;\Rightarrow\; \big(P(x_1, \ldots, x_n) \Leftrightarrow P^{\dagger *}(g_1, \ldots, g_n)\big)$$

is a theorem of $\mathsf{IZ}^{\mathrm{skol}}$.

Proof. By structural induction on P.

- If $P(x, y)$ has the form $x = y$, let us assume $\mathrm{Reif}(x, g)$ and $\mathrm{Reif}(y, h)$. We have to prove
$$x = y \Leftrightarrow g \approx h$$
and this is a consequence of Prop. 6 and 5.
- If $P(x, y)$ has the form $x \in y$, let us assume $\mathrm{Reif}(x, g)$ and $\mathrm{Reif}(y, h)$. The formula $P^{\dagger *}(g, h)$ is
$$\exists z\ (\langle z, \pi_2(h)\rangle \in \pi_1(h) \wedge g \approx \langle \pi_1(h), z\rangle)$$
and we have to prove the formula
$$x \in y \quad \Leftrightarrow \quad \exists z\ (\langle z, \pi_2(h)\rangle \in \pi_1(h) \wedge g \approx \langle \pi_1(h), z\rangle)$$
Let $B = \pi_1(h)$, $b = \pi_2(h)$, and $a = \pi_2(g)$. We have to prove
$$x \in y \quad \Leftrightarrow \quad \exists z\ (\langle z, b\rangle \in B \wedge g \approx \langle B, z\rangle)$$
Let ϕ be a collapse of g such that $x = \phi(a)$ and ψ a collapse of h such that $y = \psi(b)$.
  - Assume $x \in y$. Then there exists $z \in \mathrm{Dom}(\psi)$ such that $\psi(z) = x$ and $\langle z, b\rangle \in B$. But x is obviously a reification of the pointed graph $\langle B, z\rangle$. Since the pointed graphs g and $\langle B, z\rangle$ have the same reification x, they are bisimilar (Proposition 6).
  - Conversely, assume z such that $\langle z, b\rangle \in B$ and $\langle B, z\rangle \approx g$. From $\langle z, b\rangle \in B$, we get $\psi(z) \in \psi(b) = y$. Since the pointed graphs $\langle B, z\rangle$ and g are bisimilar, their reifications $\psi(z)$ and x are equal from Proposition 5.
- If $P(x_1, \ldots, x_n)$ has the form $Q(x_1, \ldots, x_n) \wedge R(x_1, \ldots, x_n)$, then, by induction hypothesis, under the hypotheses $\mathrm{Reif}(x_i, g_i)$, we have $Q(x_1, \ldots, x_n) \Leftrightarrow Q^{\dagger *}(g_1, \ldots, g_n)$ and $R(x_1, \ldots, x_n) \Leftrightarrow R^{\dagger *}(g_1, \ldots, g_n)$. We deduce $(Q(x_1, \ldots, x_n) \wedge R(x_1, \ldots, x_n)) \Leftrightarrow (Q^{\dagger *}(g_1, \ldots, g_n) \wedge R^{\dagger *}(g_1, \ldots, x_n))$, *i.e.* $(Q(x_1, \ldots, x_n) \wedge R(x_1, \ldots, x_n)) \Leftrightarrow (Q(g_1, \ldots, g_n) \wedge R(g_1, \ldots, g_n))^{\dagger *}$.
- If $P(x_1, \ldots, x_n)$ has the form $Q(x_1, \ldots, x_n) \vee R(x_1, \ldots, x_n)$ or $Q(x_1, \ldots, x_n) \Rightarrow R(x_1, \ldots, x_n)$, the proof is similar.
- If $P(x_1, \ldots, x_n)$ has the form $\forall x\ Q(x, x_1, \ldots, x_n)$, then $P^{\dagger *}(g_1, \ldots, g_n)$ is
$$\forall g\ [\mathrm{Rgraph}(g) \Rightarrow Q^{\dagger *}(g, g_1, \ldots, g_n)]\,.$$

- Let us assume $P(x_1, \ldots, x_n)$, *i.e.* $\forall x\ Q(x, x_1, \ldots, x_n)$, and prove $(P^\dagger)^*(g_1, \ldots, g_n)$, *i.e.* $\forall g\ [\mathrm{Rgraph}(g) \Rightarrow Q^{\dagger *}(g, g_1, \ldots, g_n)]$.
  Let g be a reifiable pointed graph, and a a reification of g. From our assumption, one has $Q(a, x_1, \ldots, x_n)$ By induction hypothesis, we have $(Q^\dagger)^*(g, g_1, \ldots, g_n)$.
- Conversely, assume $(P^\dagger)^*(g_1, \ldots, g_n)$, *i.e.* $\forall g\ [\mathrm{Rgraph}(g) \Rightarrow Q^{\dagger *}(g, g_1, \ldots, g_n)]$, and prove $P(x_1, \ldots, x_n)$, *i.e.* $\forall x\ Q(x, x_1, \ldots, x_n)$. Let x be a set. From Prop. 7, there exists a reifiable pointed graph h such that $\mathrm{Reif}(x, h)$. By induction hypothesis we have $Q(x, x_1, \ldots, x_n)$.

– If P has the form $\exists x\ Q$, the proof is similar. □

Theorem 2 (Conservativity). — *Let P be a closed formula in the language of $\mathsf{IZ}^{\mathrm{st}}$. If $\mathsf{IZ}^{\mathrm{mod}} \vdash P^\dagger$, then $\mathsf{IZ}^{\mathrm{st}} \vdash P$.*

Proof. Assume $\mathsf{IZ}^{\mathrm{mod}} \vdash P^\dagger$. By Proposition 13, we have $\mathsf{IZ}^{\mathrm{skol}} \vdash (P^\dagger)^*$, by Proposition 14 we get $\mathsf{IZ}^{\mathrm{skol}} \vdash P$ and we conclude using the fact that $\mathsf{IZ}^{\mathrm{skol}}$ is a conservative extension of $\mathsf{IZ}^{\mathrm{st}}$. □

5 Normalization

In this section, we prove that all proofs in the theory $\mathsf{IZ}^{\mathrm{mod}}$ are strongly normalizable. As this theorem implies the consistency of $\mathsf{IZ}^{\mathrm{mod}}$, it cannot be proved in set theory itself. In [6] we have generalized the usual notion of relative consistency proof to a notion of relative normalization proof. Technically, our normalization theorem is proved under the assumption that $\mathsf{IZ}^{\mathrm{skol2}}$ is 1-consistent.

5.1 Reducibility candidates

To prove normalization, we shall use the result proved in [6]. For that, we need to define a translation from $\mathsf{IZ}^{\mathrm{mod}}$ to $\mathsf{IZ}^{\mathrm{skol2}}$ associating to each term t of $\mathsf{IZ}^{\mathrm{mod}}$ a term t^* of $\mathsf{IZ}^{\mathrm{skol2}}$ and to each atomic formula P of $\mathsf{IZ}^{\mathrm{mod}}$ a formula $\pi \Vdash P$ of $\mathsf{IZ}^{\mathrm{skol2}}$. This translation is then extended to all formulæ as shown in [6]. To define the formula $\pi \Vdash P$ we shall first define a term P^* expressing a reducibility candidate and then we shall define $\pi \Vdash P$ as $\pi \in P^*$.

We refer to [6] for the definition of all notations related to reducibility candidates. In particular, we shall denote Proof the set of all proof-terms, $\mathcal{CR}$ the set of all reducibility candidates, SN the set of all strongly normalizable proofs (which is the largest reducibility candidate), and $\tilde{\Rightarrow}$, $\tilde{\wedge}$, $\tilde{\vee}$, etc. the binary operations on $\mathcal{CR}$ that interpret the corresponding intuitionistic connectives.

An important property of the class of reducibility candidates is that it is projective. Indeed, if we define $\lfloor X \rfloor_{\mathcal{CR}}$ as the intersection of all reducibility candidates containing $X \cap \mathrm{SN}$

$$\lfloor X \rfloor_{\mathcal{CR}} \ = \ \{\pi \in \mathrm{SN} \mid \forall r \in \mathcal{CR}\ (X \cap \mathrm{SN} \subseteq r \Rightarrow \pi \in r)\}$$

we easily check that $\mathsf{IZ}^{\mathrm{skol2}}$ proves

1. For all X, $\lfloor X \rfloor_{\mathcal{CR}} \in \mathcal{CR}$
2. If $X \in \mathcal{CR}$, then $X = \lfloor X \rfloor_{\mathcal{CR}}$.

Moreover, if X is a set of strongly normalizable proofs, then $\lfloor X \rfloor_{\mathcal{CR}}$ is the smallest reducibility candidate containing X.

5.2 Saturated pointed graphs

Definition 12 (Saturated pointed graph). *A* saturated graph *is a function R whose domain is a set of pairs and whose codomain is $\mathcal{CR}$. A* saturated pointed graph *is a pair $\langle R, r\rangle$ formed by a saturated graph R and an arbitrary object r.*

The formulæ 'x is a saturated graph' and 'x is a saturated pointed graph' are written $\mathrm{Sgraph}(x)$ and $\mathrm{Spgraph}(x)$, respectively. Again, it is easy to check that the class of saturated graphs and the class of saturated pointed graphs are projective, using the projections:

$$\begin{aligned} \lfloor X \rfloor_{\mathrm{Sgraph}} &\equiv \{c \in \mathrm{Dom}(X) \times \mathcal{CR} \mid \pi_2(c) = \lfloor X(\pi_1(c)) \rfloor_{\mathcal{CR}}\} \\ \lfloor X \rfloor_{\mathrm{Spgraph}} &\equiv \langle \lfloor \pi_1(X) \rfloor_{\mathrm{Sgraph}},\ \pi_2(X)\rangle \end{aligned}$$

We check that $\mathsf{IZ}^{\mathrm{skol2}}$ proves

1. For all X, $\mathrm{Spgraph}(\lfloor X \rfloor_{\mathrm{Spgraph}})$
2. If $\mathrm{Spgraph}(X)$, then $X = \lfloor X \rfloor_{\mathrm{Spgraph}}$.

(and similarly for Sgraph).

The carrier $\overline{a}$ of a saturated pointed graph a is defined as

$$\overline{a} \quad \equiv \quad \{x \in \bigcup\bigcup \pi_1(x) \mid \exists y\ \exists r\ \langle\langle x, y\rangle, r\rangle \in \pi_1(a) \vee \langle\langle y, x\rangle, r\rangle \in \pi_1(a)\}\,.$$

5.3 Translation of sorts

We now define the translation of $\mathsf{IZ}^{\mathrm{mod}}$ into $\mathsf{IZ}^{\mathrm{skol2}}$.

Each sort s of $\mathsf{IZ}^{\mathrm{mod}}$ is translated as a sort of $\mathsf{IZ}^{\mathrm{skol2}}$ written s_* accompanied with a relativization predicate written $s^*(x)$ (where x is of sort s_*). We then set:

- $G_* = \mathrm{Set}$, with $G^*(x) \equiv \mathrm{Spgraph}(x)$
- $N_* = \mathrm{Set}$, with $N^*(x) \equiv \top$
- $C_* = \mathrm{Class}$, with

$$C^*(c) \quad \equiv \quad \forall z\ (\mathrm{mem}(z, c) \Rightarrow \exists x\ \exists r\ (z = \langle x, r\rangle \wedge r \in \mathcal{CR}))$$

- $R_* = \mathrm{Class}$, with

$$R^*(c) \quad \equiv \quad \forall z\ (\mathrm{mem}(z, c) \Rightarrow \exists x\ \exists y\ \exists r\ (z = \langle\langle x, y\rangle, r\rangle \wedge r \in \mathcal{CR}))$$

If c is an element of C^* and if x is any object, we write

$$c[x] \quad = \quad \lfloor \bigcup \{r \in \mathcal{CR} \mid \mathrm{mem}(\langle x, r\rangle, c)\} \rfloor_{\mathcal{CR}}$$

the candidate associated to x in c (or the smallest candidate if there is no candidate associated to x in c). Similarly, if c is an element of R^* and if x, y are arbitrary objects, we write

$$c[x, y] \quad = \quad \lfloor \bigcup \{r \in \mathcal{CR} \mid \mathrm{mem}(\langle\langle x, y\rangle, r\rangle, c)\} \rfloor_{\mathcal{CR}}$$

the candidate associated to $\langle x, y\rangle$ in c (or the smallest candidate otherwise).

5.4 Translation of function and predicate symbols

To each function symbol f of arity n of $\mathsf{IZ}^{\mathrm{mod}}$, we associate a term $\tilde{f}(x_1, ..., x_n)$ possibly containing the free variables $x_1, ..., x_n$. These "macros" will be used later to translate full terms, setting $(f(t_1, ..., t_n))^* \equiv \tilde{f}(t_1^*, ..., t_n^*)$.

We start by some easy function symbols:

$$\begin{array}{rclrclrcl}
\tilde{\mathrm{root}}(x) & \equiv & \pi_2(x) & \tilde{x/y} & \equiv & \langle \pi_1(x), y\rangle & \tilde{o} & \equiv & 0 \\
\tilde{i}(x) & \equiv & \langle 0, x\rangle & \tilde{j}(x) & \equiv & \langle 1, x\rangle & \tilde{\rho}(x) & \equiv & x \\
\tilde{i'}(x) & \equiv & \pi_2(x) & \tilde{j'}(x) & \equiv & \pi_2(x) & \tilde{\rho'}(x) & \equiv & \lfloor x \rfloor_{\mathrm{Spgraph}} \\
\tilde{0} & \equiv & 0 & \tilde{S}(x) & \equiv & x \cup \{x\} & \tilde{Pred}(x) & \equiv & \bigcup x
\end{array}$$

In the same way, to each predicate symbol P of arity n of $\mathsf{IZ}^{\mathrm{mod}}$, we associate a term $\tilde{P}(x_1, ..., x_n)$ possibly containing the free variables $x_1, ..., x_n$. From these macros we will translate atomic formulæ by setting $(P(t_1, \ldots, t_n))^* \equiv \tilde{P}(t_1^*, \ldots, t_n^*)$. We set

$$\begin{array}{rcl}
\tilde{\mathrm{mem}}(x, p) & \equiv & p(x) \\
\tilde{\mathrm{rel}}(x, y, p) & \equiv & p(x, y) \\
\tilde{I}(x) \equiv \tilde{J}(x) \equiv \tilde{Null}(x) \equiv \tilde{Nat}(x) & \equiv & \mathrm{SN} \\
(x\ \tilde{\eta}_a\ y) & \equiv & \lfloor \pi_1(a)(x, y) \rfloor_{\mathcal{CR}}
\end{array}$$

Using both definitions $(f(t_1, ..., t_n))^* \equiv \tilde{f}(t_1^*, ..., t_n^*)$ and $(P(t_1, ..., t_n))^* \equiv \tilde{P}(t_1^*, ..., t_n^*)$, we can now translate all the terms and formulæ containing the symbols for which we have already introduced a translation. In particular, we can translate the formula $\forall p\ (\mathrm{mem}(x, p) \Rightarrow \mathrm{mem}(y, p))$. We thus translate the predicate symbol $=$ as

$$x \tilde{=} y \equiv (\forall p\ (\mathrm{mem}(x, p) \Rightarrow \mathrm{mem}(y, p)))^*$$

In a similar way we take

$$\begin{array}{rcl}
a \tilde{\approx} b & \equiv & [\exists r\ (\mathrm{rel}(\mathrm{root}(a), \mathrm{root}(b), r) \\
& & \quad \wedge\ \forall x \forall x' \forall y\, (x'\ \eta_a\ x \wedge \mathrm{rel}(x, y, r) \Rightarrow \\
& & \qquad \exists y'\, (y'\ \eta_b\ y \wedge \mathrm{rel}(x', y', r))) \\
& & \quad \wedge\ \forall y \forall y' \forall x\, (y'\ \eta_b\ y \wedge \mathrm{rel}(x, y, r) \Rightarrow \\
& & \qquad \exists x'\, (x'\ \eta_a\ x \wedge \mathrm{rel}(x', y', r))))]^* \\
a \tilde{\in} b & \equiv & [\exists x\ (x\ \eta_b\ \mathrm{root}(b) \wedge a \approx (b/x))]^*
\end{array}$$

To define $x \tilde{<} y$ and $(t < u)^*$, we proceed as follows. Fix x_0 and y_0, and consider the sequence of functions $(f_n)_{n\in\mathbb{N}} : \mathrm{Cl}(\{y_0\}) \to \mathcal{CR}$ defined by induction on n as follows:

- f_0 is defined as the constant function that maps all the elements of $\mathrm{Cl}(\{y_0\})$ to the smallest candidate.
- f_{n+1} is defined from f_n in two steps as follows:

- First we consider the functional graph f'_n defined as

$$f'_n = \{(0, \bot)\} \cup \{\langle s(z),\ f_n(z) \mathbin{\tilde{\vee}} x_0 = z\rangle \mid y \in \mathrm{Cl}(\{y\})\}$$

- Then we set

$$f_{n+1}(z) = \lfloor \{\pi \in \mathrm{Proof} \mid \exists c\ (\langle z, c\rangle \in f'_n \wedge \pi \in c)\}\rfloor_{\mathcal{CR}}$$

  for all $z \in \mathrm{Cl}(\{y\})$.

Finally we set $x_0 \mathrel{\tilde{<}} y_0 \equiv \lfloor \bigcup_{n\in\mathbb{N}} f_n(y_0)\rfloor_{\mathcal{CR}}$.

We then translate the symbols $\bigcup$, $\{,\}$, $\mathfrak{P}$, $f_{x,y_1,\ldots,y_n}$, Ω and Cl.

The formula $x\ \eta_{\bigcup(a)}\ x'$ reduces to the formula P which is:

$$\begin{array}{l}(\exists y\ \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y'))\\ \vee\ \ (\exists y\ \exists z\ (x = i(y) \wedge x' = o \wedge y\ \eta_a\ z \wedge z\ \eta_a\ \mathrm{root}(a)))\end{array}$$

Consider the translation P^* of this formula. We let

$$\tilde{\bigcup}(a) \equiv \langle R, 0\rangle$$

where $R = \{c \in (X \times X) \times \mathcal{CR} \mid \exists x\ \exists x'\ c = \langle\langle x, x'\rangle, P^*\rangle\}$ and $X = (1 \times \overline{a}) \cup \{0\}$. We do the same thing for the other constructions.

Finally, remains to define the translation of the symbols $g_{x,y_1,\ldots,y_n,P}$ and $g'_{x,x',y_1,\ldots,y_n,P}$. We set

$$\tilde{g}_{x,y_1,\ldots,y_n,P}(y_1,\ldots,y_n) \equiv \{\!|z \mid \exists x\ z = \langle x, P^*\rangle|\!\}$$

$$(\tilde{g}'_{x,x',y_1,\ldots,y_n,P}(y_1,\ldots,y_n))^* \equiv \{\!|z \mid \exists x\ \exists x'\ z = \langle\langle x, x'\rangle, P^*\rangle|\!\}$$

From [6], to get normalization, we need to prove the following two lemmas:

Proposition 15. — *For any atomic formula A of $\mathsf{IZ}^{\mathrm{mod}}$, the formula $A^* \in \mathcal{CR}$ is provable in $\mathsf{IZ}^{\mathrm{skol2}}$.*

Proof. By induction on the structure of A. □

Proposition 16. — *If $A \longrightarrow B$, then $A^* = B^*$ is provable in $\mathsf{IZ}^{\mathrm{skol2}}$ under the assumptions $s_i^*(x_i)$ for each variable x_i of sort s_i that appears in one of the formulæ A and B.*

Proof. It suffices to prove the formula for each rewrite rule $A \longrightarrow B$ (for which A is always an atomic formula). In most cases, this is obvious, since the denotation of the left-hand side has been precisely defined as the denotation of the right-hand side. □

Thus we get our final theorem.

Theorem 3. — *If $\mathsf{IZ}^{\mathrm{skol2}}$ is 1-consistent, then the theory $\mathsf{IZ}^{\mathrm{mod}}$ has the normalization property.*

6 Witness properties

Corollary 1 (Witness property in $\mathsf{IZ}^{\mathrm{mod}}$). *If a closed formula $\exists x\ P(x)$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$, then there exists a term t in $\mathsf{IZ}^{\mathrm{mod}}$ (of the same sort as the variable x) such that the formula $P(t)$ is provable.*

Proof. A cut-free proof ends with an introduction rule.

Corollary 2 (Non-numerical witness in $\mathsf{IZ}^{\mathrm{st}}$). *If a closed formula $\exists x\ P(x)$ is provable in $\mathsf{IZ}^{\mathrm{st}}$, then there exists a formula $D(y)$ with one free variable y such that*

1. *The formula $\exists!x\ D(x)$ is provable in $\mathsf{IZ}^{\mathrm{st}}$.*
2. *The formula $\forall x\ (D(x) \Rightarrow P(x))$ is provable in $\mathsf{IZ}^{\mathrm{st}}$.*

Proof. The formula $\exists x\ P^\dagger(x)$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$, hence by corollary 1 there exists a term t such that $P^\dagger(t)$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$. Hence the formula $P^{\dagger *}(t^*)$ is provable in $\mathsf{IZ}^{\mathrm{st}}$. Consider the formula $D(y) \equiv \mathrm{Reif}(y, t^*)$. From Prop. 11, we have $\mathrm{Rgraph}(t^*)$, that is: $\exists x\ D(x)$. Uniqueness follows from Prop. 5. From Prop. 14, we have

$$\forall x\ \forall g\ (\mathrm{Reif}(x, g) \Rightarrow (P(x) \Leftrightarrow P^{\dagger *}(g)))$$

hence

$$\forall x\ (\mathrm{Reif}(x, t^*) \Rightarrow (P^{\dagger *}(t^*) \Rightarrow P(x)))$$

As we have $P^{\dagger *}(t^*)$, we get $\forall x\ (D(x) \Rightarrow P(x))$. □

Corollary 3 (Numerical witness in $\mathsf{IZ}^{\mathrm{st}}$). *If a closed formula of the form $\exists\ x(Nat(x) \wedge P(x))$ is provable in $\mathsf{IZ}^{\mathrm{st}}$, then there exists a natural number n such that the formula*

$$\exists x\ (\mathrm{Is}_n(x) \wedge P(x))$$

is provable in $\mathsf{IZ}^{\mathrm{st}}$, where the formula $\mathrm{Is}_n(x)$ is inductively defined by

$$\mathrm{Is}_0(x) \equiv \mathit{Empty}(x) \qquad \textit{and} \qquad \mathrm{Is}_{n+1}(x) \equiv \exists y\ (\mathrm{Is}_n(y) \wedge \mathit{Succ}(y, x))$$

Proof. The formula $\exists\ x(Nat^\dagger(x) \wedge P^\dagger(x))$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$, hence there exists a term t in $\mathsf{IZ}^{\mathrm{mod}}$ such that $Nat^\dagger(t)$ and $P^\dagger(t)$ are provable. We check that the formula $\forall x\ (Nat^\dagger(x) \Rightarrow x \in \Omega)$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$. Hence the formula $t \in \Omega$, i.e. $\exists x\ (x\ \eta_\Omega\ o) \wedge t \approx (\Omega/x)$ is provable. Again there exists a term u such that the formulae $u\ \eta_\Omega\ o$ and $t \approx (\Omega/u)$ are provable. The formula $u\ \eta_\Omega\ o$ is equivalent by elementary means to $\exists y\ (u = i(y) \wedge Nat(y))$.

Thus there exists a term v such that $u = i(v)$ and $Nat(v)$ are provable.

A cut free proof of the formula $Nat(v)$ ends with an introduction rule. Hence $Nat(v)$ reduce to a non atomic formula and v has the form $S^n(0)$ for some n.

Thus the formula $t \approx (\Omega/i(S^n(0)))$ is provable.

We check, by induction on n that the formula $\mathrm{Is}_n^\dagger(\Omega/i(S^n(0)))$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$. Hence the formula $\exists x\ (\mathrm{Is}_n^\dagger(x) \wedge P^\dagger(x))$ is provable in $\mathsf{IZ}^{\mathrm{mod}}$. Hence $\exists x\ (\mathrm{Rgraph}(x) \wedge \mathrm{Is}_n^{\dagger *}(x) \wedge P^{\dagger *}(x))$ is provable in $\mathsf{IZ}^{\mathrm{st}}$ and by Prop. 14, the formula $\exists x\ (\mathrm{Is}_n(x) \wedge P(x))$ is provable in $\mathsf{IZ}^{\mathrm{st}}$.

7 Conclusion

In this paper we have given a normalization proof for Zermelo set theory extended with Strong Extensionality and Transitive Closure.

This theorem can also be attained as a corollary of the existence of a translation of $\mathsf{IZ}^{\mathrm{st}}$ into type theory [20] (using stronger assumptions than the 1-consistency of $\mathsf{IZ}^{\mathrm{skol2}}$). However, instead of expressing set theory on top of a theory of graphs defined in type theory, we expressed it on top of a theory of graphs simply expressed in predicate logic. The fact that this theory can be expressed with computation rules only and no axioms is a key element for the cut-free proof to end with an introduction rule. This shows that the key feature of type theory used in these translations is the feature captured by Deduction modulo: the possibility to mix computation and deduction.

Along the way we have proposed a typed lambda-calculus where all terms normalize and where all provably total functions of set theory can be expressed. It should be noticed that the syntax of lambda-calculus is exactly that of the proofs of predicate logic (*i.e.* variables, abstractions and applications, pairs and projections, disjoints union and definition by cases, ...). No new construction is needed, only the type system is new.

One striking feature of this expression of set theory in Deduction modulo is the presence of the extensionality axiom. Extensionality axioms are usually difficult to transform into computation rules. For instance, for extensional simple type theory there is, as far as we know, no expression in Deduction modulo and no normalization proof. The idea is to define equality in such a way that it is extensional and then prove that it is substitutive on the considered part of the language. Whether this method can be generalized to extensional simple type theory still remains to be investigated.

Our investigation on normalization has lead us to consider an extension of Zermelo set theory with Strong Extensionality and Transitive Closure. This raises the question of the interest *per se* of this theory. In particular, the fact that transitive closure cannot always be constructed in Zermelo set theory [9] can be seen as a weakness of this theory, which is repaired by the transitive closure axiom. However, we leave open the question of the various axioms of set theory that can be added or removed from our choice of axioms: both for weaker theories, for instance without the Transitive Closure axiom and for stronger theories, for instance with the collection scheme, the axiom of choice or the continuum hypothesis.

Finally, the fact that set operations need to be decomposed into more atomic operations raises the question of the relevance of the choice of the notion of set for the foundation of mathematics. It might be the case that founding mathematics directly on the notion of graph would be more convenient.

References

1. P. Aczel. Non well-founded sets. *Center for the Study of Language and Information, Stanford*, 1988.

2. P. Aczel. On relating type theories and set theories. In T. Altenkirch, W. Naraschewski, and B. Reus, editors, *Types for proofs and programs*, volume 1657 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 1999.
3. M. Crabbé. Non-normalisation de ZF. Manuscript, 1974.
4. M. Crabbé. Stratification and cut-elimination. *The Journal of Symbolic Logic*, 56(1):213–226, 1991.
5. G. Dowek, Th. Hardin, and C. Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31:33–72, 2003.
6. G. Dowek and A. Miquel. Relative normalization. Manuscript, available from the web pages of the authors, 2006.
7. G. Dowek and B. Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.
8. G. Dowek and B. Werner. Arithmetic as a theory modulo. In J. Giesel, editor, *Term Rewriting and Applications*, volume 3467 of *Lecture Notes in Computer Science*, pages 423–437. Springer, 2005.
9. O. Esser and R. Hinnion. Antifoundation and transitive closure in the system of Zermelo. *Notre Dame Journal of Formal Logic*, 40(2):197–205, 1999.
10. H. Friedman. Some applications of Kleene's methods for intuitionistic systems. In *Cambridge Summer School in Mathematical Logic*, volume 337 of *Springer Lecture Notes in Mathematics*, pages 113–170. Springer-Verlag, 1973.
11. J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types. In *J.E. Fenstad (Ed.), Second Scandinavian Logic Symposium*. North-Holland, 1970.
12. J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur*. PhD thesis, Université de Paris 7, 1972.
13. J.-L. Krivine. *Théorie des ensembles*. Cassini, 1998.
14. J.-L. Krivine. Typed lambda-calculus in classical Zermelo-Fraenkel set theory. *Archive for Mathematical Logic*, 40(3):189–205, 2001.
15. J.-L. Krivine. Dependent choice, 'quote' and the clock. *Theoretical Computer Science*, 308:259–276, 2003.
16. D. McCarty. *Realizability and Recursive Mathematics*. PhD thesis, Ohio State University, 1984.
17. P.-A. Melliès and B. Werner. A generic normalization proof for pure type systems. In E. Gimenez and Ch. Paulin-Mohring, editors, *Types for Proofs and Programs*, Lecture Notes in Computer Science, pages 254–276, 1998.
18. A. Miquel. *Le calcul des constructions implicite: syntaxe et sémantique*. PhD thesis, Université de Paris 7, 2001.
19. A. Miquel. A strongly normalising Curry-Howard correspondence for IZF set theory. In *Proceedings of CSL'03*, volume 2803 of *Lecture Notes in Computer Science*, pages 441–454, 2003.
20. A. Miquel. Lamda-Z: Zermelo's Set Theory as a PTS with 4 sorts. In Jean-Christophe Filliâtre, Christine Paulin-Mohring, and Benjamin Werner, editors, *TYPES*, volume 3839 of *Lecture Notes in Computer Science*, pages 232–251. Springer, 2004.
21. J. Myhill. Some properties of intuitionistic Zermelo-Fraenkel set theory. In *Cambridge Summer School in Mathematical Logic*, volume 337 of *Springer Lecture Notes in Mathematics*, pages 206–231. Springer-Verlag, 1973.
22. H. Rasiowa and R. Sikorski. *The mathematics of metamathematics*. Polish Scientific Publishers, 1963.
23. B. Werner. Sets in types, types in sets. In *Theoretical Aspects of Computer Software*, volume 1281 of *Lecture Notes in Computer Science*, pages 530–546, 1997.