HAL open science

Realizability in classical logic

Jean-Louis Krivine

▶ To cite this version:

Jean-Louis Krivine. Realizability in classical logic. Panoramas et synthèses, 2009, 27, pp.197-229. hal-00154500

HAL Id: hal-00154500
https://hal.science/hal-00154500

Submitted on 13 Jun 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Realizability in classical logic

Jean-Louis Krivine

University Paris VII, C.N.R.S.

P.P.S. Team

krivine@pps.jussieu.fr

Lessons in Marseille-Luminy, may 2004

(last revision : june 27, 2005)

Introduction

The essential aim is to explore the *Curry-Howard correspondence* : we want to associate a program with each mathematical proof and to consider each theorem as a *specification*, i.e. to find the common behavior of the programs associated with every proof of this theorem. It is a very difficult and very interesting problem, which I call the ‘‘ specification problem ’’ in what follows.

In the first place, we must give the programming language in which we write these programs, and also explain in which way they are executed. This is done in the first section, it is the ‘‘ program side ’’ of the correspondence. As we shall see, this programming frame is very similar to usual *imperative* programming. As the theory develops, many usual and important notions of imperative and system programming will naturally appear, such as : storage and call-by-value for datas, pointers, signature of files, system clock, boot, ...

Then we must give a computational content to each logical rule and each axiom. We do this by means of *elementary instructions*. The instructions for the rules of intuitionistic propositional logic have been found long time ago, at the very discovery of the Curry-Howard correspondence ; the programming language was Church's lambda-calculus. Then the instructions for intuitionistic *second order logic* were obtained and the programming language was still the same.
But mathematical proofs are not done in intuitionistic logic, not even in pure classical logic. We need *axioms* and the usual axiom systems are :

1. Second order classical Peano arithmetic with the axiom of dependent choice ; this system is also called ‘‘ Analysis ’’.
2. Zermelo-Frænkel set theory with the axiom of choice.

In this paper, we consider the first case. We shall give new elementary instructions for the lacking axioms, which are : the excluded middle, the axiom of recurrence and the axiom of dependent choice. It is necessary, for that, to define an extension of lambda-calculus. We notice that some of these instructions are *incompatible with β-reduction*. Therefore the execution strategy is deterministic and is given in the form of a weak head reduction machine.
The same machine is used for Zermelo-Frænkel set theory. The instructions associated with

the axioms of ZF are given in [4]. The full axiom of choice remained an open problem until recently (may 2005). The new instructions necessary for this axiom and also for the continuum hypothesis will be given in a forthcoming paper.

Terms, stacks, processes, execution

We denote by PL the set of *closed* λ-terms built with some set of constants which are called *instructions* ; one of these instructions is denoted by cc. We shall denote the application of t to u by $(t)u$ or tu ; the application of t to n arguments $u_1,\ldots,u_n$ is denoted by $(t)u_1\ldots u_n$ or $tu_1\ldots u_n$. Therefore, we have $tuv = (t)uv = (tu)v$ with this notation.
Elements of PL are called *proof-like terms*.
Let $\mathsf{L} \supset \mathsf{PL}$ be the set of *closed* λ-terms built with a new constant k and a set $\Pi_0 \neq \emptyset$ of new constants called *stack constants*.
A closed λ-term of L of the form $\mathsf{k}t_1\ldots t_n\pi_0$ with $n \in \mathbb{N}$, $t_1,\ldots,t_n \in \mathsf{L}$ and $\pi_0 \in \Pi_0$ is called a *continuation*.
A λ_c*-term* is, by definition, a closed λ-term $\tau \in \mathsf{L}$ with the following properties :
- each occurrence of k in τ appears at the beginning of a subterm of τ which is a continuation ;
- each occurrence of a stack constant in τ appears at the end of a subterm of τ which is a continuation.

Remark. Proof-like terms are therefore λ_c-terms which do not contain the symbol k or, which amounts to the same thing, which do not contain any stack constant.

The set of λ_c-terms is denoted by Λ_c. In what follows, we almost always consider only λ_c-terms ; so, they will be called simply " terms " or " closed terms ".

Lemma 1. *i) $\mathsf{PL} \subset \Lambda_c$;*
ii) If $tu \in \Lambda_c$ and $u \notin \Pi_0$, then $t \in \Lambda_c$ and $u \in \Lambda_c$;
iii) If $tu \in \Lambda_c$ and $u \in \Pi_0$, then tu is a continuation ;
iv) If $\lambda x\, t, u \in \Lambda_c$, then $t[u/x] \in \Lambda_c$;
v) If $t_1,\ldots,t_n \in \Lambda_c$ ($n \in \mathbb{N}$) and $\pi_0 \in \Pi_0$, then $(\mathsf{k}\, t_1\ldots t_n)\pi_0 \in \Lambda_c$.

i) Trivial.
ii) Consider an occurrence of k (resp. of a stack constant π_0) in t or u ; it is therefore in tu. Since $tu \in \Lambda_c$, this occurrence is at the beginning (resp. at the end) of a continuation $\mathsf{k}t_1\ldots t_n\pi_0 = (\mathsf{k}t_1\ldots t_n)\pi_0$ which is a subterm of tu. Now $u \neq \pi_0$, so that this subterm is not tu itself ; thus, $\mathsf{k}t_1\ldots t_n\pi_0$ is a subterm of t or u.
iii) We have $u = \pi_0 \in \Pi_0$ and this occurrence of π_0 is at the end of a subterm of $t\pi_0$ which is a continuation $(\mathsf{k}t_1\ldots t_n)\pi_0$. Therefore, this subterm is $t\pi_0$ itself.
iv) Consider an occurrence of k (resp. π_0) in $t[u/x]$; thus, it is in $\lambda x\, t$ or u. But $\lambda x\, t, u \in \Lambda_c$, so that this occurrence is at the beginning (resp. at the end) of a continuation $\mathsf{k}t_1\ldots t_n\pi_0$ which is a subterm of $\lambda x\, t$ or u. If this continuation is a subterm of u, it is also a subterm of $t[u/x]$. If it is a subterm of $\lambda x\, t$, then this occurrence of k (resp. π_0) in $t[u/x]$ is at the beginning (resp. at the end) of the subterm $\mathsf{k}t_1[u/x]\ldots t_n[u/x]\pi_0$, which is a continuation.
v) Trivial.
Q.E.D.

If $t_1, \ldots, t_n \in \Lambda_c$ ($n \in \mathbb{N}$) and $\pi_0 \in \Pi_0$, the sequence $\pi = (t_1, \ldots, t_n, \pi_0)$ is called a *stack* and is denoted by $t_1.t_2 \ldots t_n.\pi_0$; the set of stacks is denoted by Π. The continuation $\mathsf{k}t_1 \ldots t_n\pi_0$ is denoted by k_π. If $t \in \Lambda_c$ and $\pi = t_1.t_2 \ldots t_n.\pi_0 \in \Pi$, then the stack $t.t_1.t_2 \ldots t_n.\pi_0$ is denoted by $t.\pi$. Thus, the dot is an application of $\Lambda_c \times \Pi$ in Π.

Every term $\tau \in \Lambda_c$ is either an application, or an abstraction, or a constant *which is an instruction* (indeed, this term can be neither k, nor a stack constant, by definition of λ_c-terms). From lemma 1, it follows that, for every $\tau \in \Lambda_c$, we are in one and only one of the following cases :
i) τ is an application tu with $u \notin \Pi_0$ (and therefore $t, u \in \Lambda_c$) ;
ii) τ is an abstraction $\lambda x\, t$;
iii) τ is an instruction (particular case : $\tau = \mathsf{cc}$) ;
iv) τ is an application tu with $u \in \Pi_0$, i.e. $\tau = \mathsf{k}_\pi$ for some stack π.

A *process* is an ordered pair (τ, π) with $\tau \in \Lambda_c$, $\pi \in \Pi$. It is denoted by $\tau \star \pi$; τ is called the *head* or the *active part* of the process. The set of all processes will be denoted by $\Lambda_c \star \Pi$.
We describe now the execution of processes, which is denoted by $\succ$. We give the rule to perform *one execution step* of the process $\tau \star \pi$. It depends on the form (i), ..., (iv) of τ given above. In the four rules that follow, $t, u, \lambda x\, v$ denote elements of Λ_c ; π, ρ denote stacks.

i) $tu \star \pi \succ t \star u.\pi$ (push) ;
ii) $\lambda x\, v \star u.\pi \succ v[u/x] \star \pi$ (pop) ;
iii) $\mathsf{cc} \star t.\pi \succ t \star \mathsf{k}_\pi.\pi$ (store the stack) ;
the rule for other instructions will be given in due time ;
iv) $\mathsf{k}_\pi \star t.\rho \succ t \star \pi$ (restore the stack).

We say that the process $t \star \pi$ *reduce to* $t' \star \pi'$ (notation $t \star \pi \succ t' \star \pi'$) if we get $t' \star \pi'$ from $t \star \pi$ by means of a finite (possibly null) number of execution steps.

Remark. By lemma 1, we can check that the four execution rules give processes when applied to processes.

Truth values and types

Consider an arbitrary set of processes, which is denoted by $\perp\!\!\!\perp$ and which we suppose cc-*saturated* ; it means that :

$$t \star \pi \in \perp\!\!\!\perp, t' \star \pi' \succ t \star \pi \Rightarrow t' \star \pi' \in \perp\!\!\!\perp.$$

$\mathcal{P}(\Pi)$ is called the *set of truth values.* If $U \subset \Pi$ is a truth value and $t \in \Lambda_c$, we say that t *realizes* U (notation $t \Vdash U$) if $(\forall \pi \in U)\, t \star \pi \in \perp\!\!\!\perp$. The set $\{t \in \Lambda_c;\ t \Vdash U\}$ will be denoted by $|U|$. Thus, we have $|\bigcup_{i \in I} U_i| = \bigcap_{i \in I} |U_i|$.
The truth value $\emptyset$ (resp. Π) is called *true* (resp. *false*) and denoted by $\top$ (resp. $\bot$). Thus, we have $|\top| = \Lambda_c$; $t \in |\bot| \Leftrightarrow t \star \pi \in \perp\!\!\!\perp$ for every stack $\pi \in \Pi$.
Whenever U, V are truth values, we define the truth value :
$(U \to V) = \{t.\pi;\ t \Vdash U,\ \pi \in V\}$; we put $\neg U = (U \to \bot)$.
We shall sometimes use the following notation, when V is a truth value and $A \subset \Lambda_c$:
$(A \to V) = \{t.\pi;\ t \in A,\ \pi \in V\}$.
For example, $\{t\} \to V$ is the truth value $\{t.\pi;\ \pi \in V\}$ if $t \in \Lambda_c$ and $V \subset \Pi$.
With this notation, $U \to V$ is the same as $|U| \to V$, for $U, V \subset \Pi$.
Remarks. If $\perp\!\!\!\perp = \emptyset$, then either $|U| = |\top| = \Lambda_c$ or $|U| = |\bot| = \emptyset$ for every truth value U.
$|\bot| = \emptyset$ is equivalent to $\perp\!\!\!\perp = \emptyset$. Indeed, the implication $\Leftarrow$ is obvious. Conversely, if $\perp\!\!\!\perp \neq \emptyset$, let $t \star \pi \in \perp\!\!\!\perp$; then $\mathsf{k}_\pi t \star \rho \succ t \star \pi \in \perp\!\!\!\perp$ for every stack ρ, and therefore $\mathsf{k}_\pi t \in |\bot|$.

Types are usual formulas of second order logic, written in the following language : the only logical symbols are $\to$ and $\forall$; we have function symbols of arity k which are functions from $\mathbb{N}^k$ into $\mathbb{N}$ and predicate symbols of arity k which are applications from $\mathbb{N}^k$ into $\mathcal{P}(\Pi)$ (k is any integer ≥ 0). First order variables (also called *individual variables*) are denoted by $x, y, \ldots$; second order variables (also called *predicate* variables) are denoted by $X, Y, \ldots$ Each second order variable has an arity which is an integer. Predicate variables of arity 0 are also called *propositional variables*.

Notations.

The formula $F_0 \to (F_1 \to \ldots \to (F_n \to G) \ldots)$ is denoted by $F_0, F_1, \ldots, F_n \to G$.
$\perp$ is defined by $\forall X\, X$; $\neg F$ by $F \to \perp$;
$F \vee G$ by $\forall X[(F \to X), (G \to X) \to X]$;
$F \wedge G$ by $\forall X[(F, G \to X) \to X]$;
$\exists Y\, F[Y]$ by $\forall X\{\forall Y(F[Y] \to X) \to X\}$; $\exists y\, F[y]$ by $\forall X\{\forall y(F[y] \to X) \to X\}$;
We use the notation $\exists Y\{F_1[Y], \ldots, F_k[Y]\}$ for the formula :
$\forall X\{\forall Y(F_1[Y], \ldots, F_k[Y] \to X) \to X\}$;
we have the same notation for the first order existential quantifier.
(In all these formulas, X is a propositional variable and Y has an arbitrary arity).
$x = y$ is defined by $\forall X(Xx \to Xy)$ (where X has arity 1).

Let $x_1, \ldots, x_k$ be individual variables, X a predicate variable of arity k, A and F arbitrary formulas. Then, we define $A[F/Xx_1 \ldots x_k]$ by induction on (the length of) A :

If X is not free in A, then $A[F/Xx_1 \ldots x_k]$ is A.
If A is $X(t_1, \ldots, t_k)$, then $A[F/Xx_1 \ldots x_k]$ is $F[t_1/x_1, \ldots, t_k/x_k]$.
$(A \to B)[F/Xx_1 \ldots x_k]$ is $A[F/Xx_1 \ldots x_k] \to B[F/Xx_1 \ldots x_k]$.
$(\forall y\, A)[F/Xx_1 \ldots x_k]$ is $\forall y\, A[F/Xx_1 \ldots x_k]$.
$(\forall Y\, A)[F/Xx_1 \ldots x_k]$ is $\forall Y\, A[F/Xx_1 \ldots x_k]$ if Y is a predicate variable which is $\neq X$
(as usual, we assume that y et Y are not free in F).

When F is an atomic formula of the form $R(x_1, \ldots, x_k)$, where R is either a second order variable of arity k, or a parameter ($R \in \mathcal{P}(\Pi)^{\mathbb{N}^k}$), we write also $A[R/X]$ instead of $A[F/Xx_1 \ldots x_k]$.

We now give the deduction rules for classical second order logic and, at the same time, the typing rules for λcc-terms, which are the usual λ-terms (possibly not closed) written with the constant cc ; in such an expresssion as " $t : A$ ", t is a λcc-term and A is a type, i.e. a second order formula.

Let Γ be a *context*, i.e. an expression of the form $x_1 : A_1, \ldots, x_n : A_n$.

1. $\Gamma \vdash x_i : A_i$ ($1 \leq i \leq n$)
2. $\Gamma \vdash t : A \to B,\ \Gamma \vdash u : A \ \Rightarrow\ \Gamma \vdash tu : B$.
3. $\Gamma, x : A \vdash t : B \ \Rightarrow\ \Gamma \vdash \lambda x\, t : A \to B$.
4. $\Gamma \vdash t : (A \to B) \to A \ \Rightarrow\ \Gamma \vdash \mathsf{cc}\, t : A$.
5. $\Gamma \vdash t : A \ \Rightarrow\ \Gamma \vdash t : \forall x\, A$ (resp. $\forall X\, A$) if x (resp. X) is not free in Γ.
6. $\Gamma \vdash t : \forall x\, A \Rightarrow \Gamma \vdash t : A[\tau/x]$ for every term τ of $\mathcal{L}$.
7. $\Gamma \vdash t : \forall X\, A \Rightarrow \Gamma \vdash t : A[F/Xx_1 \ldots x_k]$ for every formula F.

Rule 4 uses the interpretation of Griffin [2] for the law of Peirce.
Rule 7 uses Takeuti's interpretation for the comprehension scheme.

Let A be a closed second order formula with first order parameters in $\mathbb{N}$ and second order parameters of arity k in $\mathcal{P}(\Pi)^{\mathbb{N}^k}$ for each integer k. We define below its truth value, which is

denoted by $\|A\|$. We denote by $|A|$ the set of terms in Λ_c which realize A, i.e. $|A| = \{t \in \Lambda_c;$ $(\forall \pi \in \|A\|)\, t \star \pi \in \perp\!\!\!\perp\}$. We write $t \Vdash A$ (read " t realizes A ") if $t \in |A|$.
The definition of $\|A\|$ is given by induction on A :

If A is atomic, i.e. $A \equiv R(t_1, \ldots, t_k)$, then $t_1, \ldots, t_k$ are closed terms and $R \in \mathcal{P}(\Pi)^{\mathbb{N}^k}$ is a second order parameter. Let $a_i \in \mathbb{N}$ be the value of t_i ; then, we put :

$\|R(t_1, \ldots, t_k)\| = R(a_1, \ldots, a_k) \subset \Pi$.

The other steps of induction are as follows :

$\|A \to B\| = \{t.\pi \ ; \ t \in |A|, \ \pi \in \|B\|\}$;

$\|\forall x\, A\| = \bigcup_{a \in \mathbb{N}} \|A[a/x]\|$; (and therefore $|\forall x\, A| = \bigcap_{a \in \mathbb{N}} |A[a/x]|$) ;

$\|\forall X\, A\| = \bigcup \{\|A[R/X]\|; \ R \in \mathcal{P}(\Pi)^{\mathbb{N}^k}\}$ if X is a predicate variable of arity k

(and therefore $|\forall X\, A| = \bigcap \{|A[R/X]|; \ R \in \mathcal{P}(\Pi)^{\mathbb{N}^k}\}$).

We get $\|\perp\!\!\!\perp\| = \Pi$ and $|\perp\!\!\!\perp|$ is the least truth value. There is a greatest truth value, denoted by $\top$, which is $\emptyset$; thus, $|\top| = \Lambda_c$.

The following theorem shows that realizability is compatible with deduction in classical second order logic. It is an essential tool in this theory.

Theorem 2 (Adequation lemma). *Let $A_1, \ldots, A_k, A$ be closed formulas such that $x_1 : A_1, \ldots, x_k : A_k \vdash t : A$ is provable with the rules above.*
If $t_i \Vdash A_i$ for $1 \leq i \leq k$, then $t[t_1/x_1, \ldots, t_k/x_k] \Vdash A$.

In particular, if A is a closed formula and $\vdash t : A$ is provable, then $t \Vdash A$.

The following lemma is a stronger, but more complicated, form of this theorem ; it is suitable for a proof by induction.

Lemma 3.
Let $A_1, \ldots, A_k, A$ be formulas, free variables of which are amongst $y_1, \ldots, y_m, Y_1, \ldots, Y_n$. Let $b_i \in \mathbb{N}$ $(1 \leq i \leq m)$ and $P_j \in \mathcal{P}(\Pi)^{\mathbb{N}^{k_j}}$ $(1 \leq j \leq n)$, where k_j is the arity of Y_j. Suppose that $x_1 : A_1, \ldots, x_k : A_k \vdash t : A$ is provable with the above rules.
If $t_i \Vdash A_i[b_1/y_1, \ldots, b_m/y_m, P_1/Y_1, \ldots, P_n/Y_n]$ for $1 \leq i \leq k$, then $t[t_1/x_1, \ldots, t_k/x_k] \Vdash A[b_1/y_1, \ldots, b_m/y_m, P_1/Y_1, \ldots, P_n/Y_n]$.

We make an induction on the length of the proof of $\Gamma \vdash t : A$
where Γ is the context $x_1 : A_1, \ldots, x_k : A_k$. We shall use the notation t' for $t[t_1/x_1, \ldots, t_k/x_k]$ and A' for $A[b_1/y_1, \ldots, b_m/y_m, P_1/Y_1, \ldots, P_n/Y_n]$. Consider the rule used in the last step of the proof :

If it is rule 1 the result is trivial ;

If it is rule 2, we have $t = uv$ and $\Gamma \vdash u : A \to B$, $v : A$. We must show that $t' \in |B'|$, that is $u'v' \star \pi \in \perp\!\!\!\perp$ for every stack $\pi \in \|B'\|$. But $u'v' \star \pi \succ u' \star v'.\pi$, hence the result since, by induction hypothesis, we have $u' \in |A' \to B'|$, $v' \in |A'|$ and thus $v'.\pi \in \|A' \to B'\|$.

If it is rule 3, we have $t = \lambda x\, u$, $A = (B \to C)$ and $\Gamma, x : B \vdash u : C$. We must show that $\lambda x\, u' \in |B' \to C'|$, that is $\lambda x\, u' \star \pi \in \perp\!\!\!\perp$ for every $\pi \in \|B' \to C'\|$. Now, we have $\pi = v.\varpi$ with $v \in |B'|$ and $\varpi \in \|C'\|$. By induction hypothesis, we have $u'[v/x] \Vdash C'$ and therefore $u'[v/x] \star \varpi \in \perp\!\!\!\perp$. Thus $\lambda x\, u' \star v.\varpi \in \perp\!\!\!\perp$, because $\perp\!\!\!\perp$ is cc-saturated.

If it is rule 4, we have $t = \mathsf{cc}\, u$ and $\Gamma \vdash u : (A \to B) \to A$. We must show that $t' \in |A'|$, that is $\mathsf{cc}\, u' \star \pi \in \perp\!\!\!\perp$ for every $\pi \in \|A'\|$. Since $\perp\!\!\!\perp$ is cc-saturated, it is sufficient to show that $u' \star \mathsf{k}_\pi.\pi \in \perp\!\!\!\perp$. We first show that $\mathsf{k}_\pi \in |A' \to B'|$: indeed, if $v \in |A'|$ et $\rho \in \|B'\|$, then $\mathsf{k}_\pi \star v.\rho \succ v \star \pi \in \perp\!\!\!\perp$.
Now, by induction hypothesis, we know that $u' \in |(A' \to B') \to A'|$, hence the result.

If it is rule 5 for a first order x, we have $A = \forall x\, B$ and $\Gamma \vdash t : B$. By induction hypothesis, we get $t' \in |B'[a/x]|$ for each $a \in \mathbb{N}$. Thus $t' \in \bigcap_{a\in\mathbb{N}} |B'[a/x]| = |\forall x\, B'| = |A'|$.

If it is rule 5 for a second order variable X of arity k, we have $A = \forall X\, B$ and $\Gamma \vdash t : B$. By induction hypothesis, we have $t' \in |B'[R/X]|$ for every $R \in \mathcal{P}(\Pi)^{\mathbb{N}^k}$.
Thus $t' \in |\forall X\, B'| = |A'|$.

If it is rule 6, we have $A = B[\tau/x]$ and $\Gamma \vdash t : \forall x\, B$. By induction hypothesis, we get $t' \in |\forall x\, B'|$. But, if $a \in \mathbb{N}$ is the value of τ, we have $|B'[\tau/x]| = |B'[a/x]| \supset |\forall x\, B'|$ and therefore $t \in |B'[\tau/x]| = |A'|$.

If it is rule 7, we have $A = B[\Phi(z_1, \dots, z_p)/Xz_1 \dots z_p]$ and therefore :
$A' = B'[\Phi'(z_1, \dots, z_p)/Xz_1 \dots z_p]$. We have $\Gamma \vdash t : \forall X\, B$ and we must show that $t' \in |A'|$. By induction hypothesis, we know that $t' \in |\forall X\, B'|$ and the result follows from lemma 4.

Q.E.D.

Lemma 4. *Let Φ (resp. A) be a formula with parameters, with the only free variables $z_1, \dots, z_p$ (resp. X of arity p). Define $R \in \mathcal{P}(\Pi)^{\mathbb{N}^p}$ by :*
$R(a_1, \dots, a_p) = \|\Phi[a_1/z_1, \dots, a_p/z_p]\|$ *for* $a_1, \dots, a_p \in \mathbb{N}$.
Then $\|A[\Phi/Xz_1 \dots z_p]\| = \|A[R/X]\|$ *and therefore :*
$\|A[\Phi/Xz_1 \dots z_p]\| \subset \|\forall X\, A\|$ *and* $|\forall X\, A| \subset |A[\Phi/Xz_1 \dots z_p]|$.

We show $\|A[\Phi/Xz_1 \dots z_p]\| = \|A[R/X]\|$ by induction on the length of A.
The result is trivial if X is not free in A, if A is atomic, or if $A = (B \to C)$.

If $A = \forall x\, B$, then $\|A[\Phi/Xz_1 \dots z_p]\| = \bigcup_{a\in\mathbb{N}} \|B[\Phi/Xz_1 \dots z_p][a/x]\|$
$= \bigcup_{a\in\mathbb{N}} \|B[a/x][\Phi/Xz_1 \dots z_p]\| = \bigcup_{a\in\mathbb{N}} \|B[a/x][R/X]\|$ by induction hypothesis
$= \bigcup_{a\in\mathbb{N}} \|B[R/X][a/x]\| = \|\forall x\, B[R/X]\| = \|A[R/X]\|$.

If $A = \forall Y\, B$, with Y of arity q and $Y \neq X$, then :
$\|A[\Phi/Xz_1 \dots z_p]\| = \bigcup\{\|B[\Phi/Xz_1 \dots z_p][S/Y]\|;\, S \in \mathcal{P}(\Pi)^{\mathbb{N}^q}\}$
$= \bigcup\{\|B[S/Y][\Phi/Xz_1 \dots z_p]\|;\, S \in \mathcal{P}(\Pi)^{\mathbb{N}^q}\}$
$= \bigcup\{\|B[S/Y][R/X]\|;\, S \in \mathcal{P}(\Pi)^{\mathbb{N}^q}\}$ by induction hypothesis
$= \bigcup\{\|B[R/X][S/Y]\|;\, S \in \mathcal{P}(\Pi)^{\mathbb{N}^q}\} = \|\forall Y\, B[R/X]\| = \|A[R/X]\|$.

Q.E.D.

Remark. The terms we obtain by means of proofs in classical second order logic are $\lambda\mathsf{cc}$-terms, that is λ-terms with the constant cc, not necessarily closed. On the other hand, the terms which appear in processes are in Λ_c (i.e. closed terms with instructions and continuations). The common elements are proof-like terms with the only instruction cc (hence the name " proof-like ").

Theorem 5. *For $U, V \in \mathcal{P}(\Pi)$, we put $U \leq V \Leftrightarrow (\exists \theta \in \mathsf{PL})\, \theta \Vdash U \to V$.*
Suppose that $\perp\!\!\!\perp$ is coherent, which means that $(\forall \theta \in \mathsf{PL})\, \theta \nVdash \bot$. Then $\leq$ is a preorder (i.e. reflexive and transitive) and $(\mathcal{P}(\Pi), \leq)$ is a Boolean algebra.

Proof that $\leq$ is a preorder : if $U \leq V$ and $V \leq W$, then there are $\xi, \eta \in \mathsf{PL}$ such that $\xi \Vdash U \to V$, $\eta \Vdash V \to W$. Therefore $\eta \circ \xi \Vdash U \to W$, hence $U \leq W$.

Proof that $(\mathcal{P}(\Pi), \leq)$ is a Boolean algebra : we have $U \wedge V \leq U$ and $U \wedge V \leq V$ (the proofs of $U \wedge V \to U$ and $U \wedge V \to V$ give the requested terms). Moreover, if $Z \leq U, V$, then $\xi \Vdash Z \to U$, $\eta \Vdash Z \to V$ thus $\lambda z \lambda f((f)(\xi)z)(\eta)z \Vdash Z \to U \wedge V$. Therefore we have $Z \leq U \wedge V$ and $U \wedge V = \inf(U, V)$. In the same way, we get $U \vee V = \sup(U, V)$.
We have $I \Vdash \bot \to U$ and $I \Vdash U \to \top$. Therefore $\bot$ is the least element and $\top$ is the greatest.
We have $\neg U \wedge U \leq \bot$ and $\top \leq \neg U \vee U$ since $\neg U \wedge U \to \bot$ and $\top \to \neg U \vee U$ are provable. Thus $\neg U$ is the complement of U.
It remains to show that $\top \not\leq \bot$. This clearly follows from the hypothesis that $\perp\!\!\!\perp$ is coherent.

Q.E.D.

As soon as we choose a coherent set $\perp\!\!\!\perp$, each closed formula with parameters is given a value in this Boolean algebra. The set $\mathcal{T}_{\perp\!\!\!\perp}$ of formulas with value 1, i.e. formulas which are realized by a proof-like term, is a consistent theory which contains second order logic. We are interested in the models of this theory ; we call them *generic models*.
The case when $\perp\!\!\!\perp = \emptyset$ is trivial : the Boolean algebra is $\{0, 1\}$ and we get back the standard model we started with.
We say that a closed formula F, with parameters, is *realized* if there exists a proof-like term θ such that $\theta \Vdash F$ *for any choice of the* CC*-saturated set* $\perp\!\!\!\perp$.
All axioms of second order logic are realized. Thus, generic models satisfy second order logic.

The tools we have built up to now are sufficient in order to solve the specification problem for some very simple valid formulas.

Theorem 6. *If* $\vdash \theta : \forall X(X \to X)$*, then* $\theta \star t.\pi \succ t \star \pi$ *for every* $t \in \Lambda_c$ *and* $\pi \in \Pi$.

Given t and π, we put $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi;\ p \succ t \star \pi\}$ and $\|X\| = \{\pi\}$. Thus, we have $t \Vdash X$. But, by theorem 2, we have $\theta \Vdash X \to X$, hence $\theta \star t.\pi \in \perp\!\!\!\perp$. It is exactly what we wanted to show.

Q.E.D.

We denote by *Bool*(x) the formula $\forall X(X1, X0 \to Xx)$, which is equivalent to $x = 0 \vee x = 1$ in classical secod order logic.

Theorem 7. *If* $\vdash \theta$: *Bool*(1) *(resp.* *Bool*(0)*), then* $\theta \star t.u.\pi \succ t \star \pi$ *(resp.* $u \star \pi$*) for every* $t, u \in \Lambda_c$ *and* $\pi \in \Pi$.

Given t and π, we put $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi;\ p \succ t \star \pi\}$ and we define a unary parameter X by putting $\|X1\| = \{\pi\}$, $\|Xn\| = \emptyset$ for $n \neq 1$. Thus, we have $t \Vdash X1$ and $u \Vdash X0$.
If $\vdash \theta$: *Bool*(1), by theorem 2, we have $\theta \Vdash X1, X0 \to X1$, hence $\theta \star t.u.\pi \in \perp\!\!\!\perp$, which is the result.

Q.E.D.

Remark. The meaning of these theorems is that the terms of type $\forall X(X \to X)$ (resp. *Bool*(1), *Bool*(0)) behave essentially like $I = \lambda x\, x$ (resp. $\lambda x \lambda y\, x$, $\lambda x \lambda y\, y$).

Axiom of recurrence and call by value

We should now realize the axiom of recurrence, which is : $\forall x\, Int(x)$ where $Int(x)$ is the formula $\forall X[\forall y(Xy \to Xsy), X0 \to Xx]$ that says that the individual x is an integer. Unfortunately, this axiom is not realized. It means that, in a generic model, there may be individuals which

are not integers. Moreover, we shall see later that, in general, there may be also non standard integers in generic models.
In order to solve this problem, we introduce now some simple tools, which will appear essential in what follows.

We define a binary predicate constant, denoted by $\neq$, by putting $\|m \neq n\| = \emptyset$ if $m \neq n$ and $\|m \neq n\| = \Pi$ if $m = n$. The following theorem shows that we can profitably use the formula $x \neq y$ instead of $x = y \to \bot$.

Theorem 8. *The formula* $\forall x \forall y[x \neq y \leftrightarrow (x = y \to \bot)]$ *is realized. Indeed :*
$\lambda x\, xI \Vdash \forall x \forall y[(x = y \to \bot) \to x \neq y]$ *and* $\lambda x \lambda y\, yx \Vdash \forall x \forall y[x \neq y \to (x = y \to \bot)]$.

We first check that $|m = n| = |\forall X(X \to X)|$ if $m = n$ and $|m = n| = |\top \to \bot|$ if $m \neq n$. It is immediate, since $m = n$ is the formula $\forall X(Xm \to Xn)$.
We have to show that $\lambda x\, xI \Vdash (m = n \to \bot) \to m \neq n$ and $\lambda x \lambda y\, yx \Vdash m \neq n, m = n \to \bot$ for every $m, n \in \mathbb{N}$. In the first case, we only need to check that :
$\lambda x\, xI \Vdash (m = m \to \bot) \to \bot$, which is obvious since $I \Vdash m = m$.
In the second case, we must check that :
$\lambda x \lambda y\, yx \Vdash \top, m = n \to \bot$ if $m \neq n$ and $\lambda x \lambda y\, yx \Vdash \bot, m = n \to \bot$ if $m = n$
which is immediate, using the above computation of $|m = n|$.

Q.E.D.

Theorem 9. *Suppose that* $\theta \star \mathsf{k}_\pi.\rho \in \perp\!\!\!\perp$ *for any stack* $\pi \in \|X\|$ *and* $\rho \in \|Y\|$. *Then :*
$V\theta \Vdash \neg X \to Y$ *with* $V = \lambda x \lambda y(\mathsf{cc})\lambda h(y)(\mathsf{cc})\lambda k(h)(x)k$.

Let $t \Vdash \neg X$ et $\rho \in \|Y\|$. We must show that $V\theta \star t.\rho \in \perp\!\!\!\perp$; but $V\theta \star t.\rho \succ V \star \theta.t.\rho \succ (\mathsf{cc})\lambda h(t)(\mathsf{cc})\lambda k(h)(\theta)k \star \rho \succ t \star (\mathsf{cc})\lambda k(\mathsf{k}_\rho)(\theta)k.\rho$. Since $t \Vdash X \to \bot$, it is therefore sufficient to prove that $(\mathsf{cc})\lambda k(\mathsf{k}_\rho)(\theta)k \Vdash X$, in other words $(\mathsf{cc})\lambda k(\mathsf{k}_\rho)(\theta)k \star \pi \in \perp\!\!\!\perp$ for every $\pi \in \|X\|$. This is true because this process reduces into $\theta \star \mathsf{k}_\pi.\rho$.

Q.E.D.

Remark. With the notation introduced page 3, the hypothesis of theorem 9 is : $\theta \Vdash \{\mathsf{k}_\pi; \pi \in \|X\|\} \to Y$. Theorem 9 says that the set $|\neg X|$ behaves, in some sense, like its subset $\{\mathsf{k}_\pi; \pi \in \|X\|\}$; and it gives a simple way to check that $X \vee Y$ is realized.

Theorem 10 (Storage of integers). *We put* $T = \lambda f \lambda n(n \lambda g\, g \circ s)f0$, *where* s *is a* λ*-term for the successor (for instance* $s = \lambda n \lambda f \lambda x(f)(n)fx$*).*
If $\phi \in \Lambda_c$ *is such that* $\phi \star s^n 0.\pi \in \perp\!\!\!\perp$ *for every* $\pi \in \|X\|$, *then* $T\phi \Vdash Int[s^n 0] \to X$.

Remarks.
1. With the notation introduced page 3, the hypothesis of theorem 10 is : $\phi \Vdash \{s^n 0\} \to X$.
Theorem 10 says that the set $|Int[s^n 0]|$ behaves, in some sense, like its subset $\{s^n 0\}$.
2. T is called a *storage operator* [3]. We understand intuitively its behavior by comparing the weak head reductions of $\phi\nu$ and $T\phi\nu$, where ϕ and ν are λ-terms, $\nu \simeq_\beta \lambda f \lambda x(f)^n x$ (Church integer). We get $T\phi\nu \succ (\phi)(s)^n 0$, which means that the integer argument ν of ϕ is computed first. In other words, T simulate call by value in a weak head reduction, that is a call by name execution.

Note that we use the same symbol s for a λ-term and a function symbol of $\mathcal{L}$.

Proof. Let $\nu \Vdash Int[s^n 0]$ and $\pi \in \|X\|$; we must show that $T\phi \star \nu.\pi \in \perp\!\!\!\perp$.
We define a unary predicate P by putting :

$\|P(j)\| = \{s^{n-j}0.\pi\}$ for $0 \leq j \leq n$ and $\|P(j)\| = \emptyset$ for $j > n$.
We have $\phi \Vdash P0$ by hypothesis on ϕ.
Let us show that $\lambda g\, g\circ s \Vdash \forall x(Px \to Psx)$, which means that $\lambda g\, g\circ s \in |P(j) \to P(j+1)|$ for every $j \in \mathbb{N}$. It is trivial for $j \geq n$.
If $j < n$, let $\xi \in |P(j)|$; we must show that $\lambda g\, g\circ s \star \xi.s^{n-j-1}0.\pi \in \perp\!\!\!\perp$. But we have :
$\lambda g\, g\circ s \star \xi.s^{n-j-1}0.\pi \succ \xi\circ s \star s^{n-j-1}0.\pi \succ \xi \star s^{n-j}0.\pi \in \perp\!\!\!\perp$ by hypothesis on ξ.

Now, we have $T\phi \star \nu.\pi \succ \nu \star \lambda g\, g\circ s.\phi.0.\pi$ which is in $\perp\!\!\!\perp$ because :
$\nu \Vdash \forall x(Px \to Psx), P0 \to Ps^n0$ by hypothesis, $\phi \Vdash P0$ and $0.\pi \in \|Ps^n0\|$.

Q.E.D.

We are interested in the theory called *Analysis* or *Second order Peano arithmetic*, which is classical second order logic with the following supplementary axioms :

1. A set of *equational* formulas, i.e. formulas of the form :
$\forall x_1 \ldots \forall x_k[t(x_1, \ldots, x_k) = u(x_1, \ldots, x_k)]$, where t, u are terms which represent functions from $\mathbb{N}^k$ into $\mathbb{N}$. Of course, all these formulas are supposed to be true in $\mathbb{N}$. For instance, we can take the following equational formulas, where $0, s, p, +, .$ are function symbols that represent respectively the integer 0, the successor function, the predecessor, the addition and multiplication on integers :
$p0 = 0$; $\forall x(psx = x)$; $\forall x(x + 0 = x)$; $\forall x[x + sy = s(x + y)]$;
$\forall x(x.0 = 0)$; $\forall x\forall y[x.sy = x.y + x]$.
2. $\forall x(0 \neq sx)$.
3. The recurrence axiom $\forall x\, Int(x)$.
4. The axiom of dependent choice.

All equational axioms 1 are realized by $I = \lambda x\, x$. Axiom 2 is realized by any proof-like term.
But there is a problem with axiom 3, as is shown by the following theorem :

Theorem 11. *There is no proof-like term and even no instruction which realizes* $\forall x\, Int(x)$.

Let ξ be such an instruction ; thus, we have $\xi \Vdash Int(0)$ and $\xi \Vdash Int(1)$ for every choice of the set $\perp\!\!\!\perp$. Let $\delta = \lambda x\, xx$, $\alpha_0 = \delta\delta 0$, $\alpha_1 = \delta\delta 1$; let π be some stack.
We first take $\perp\!\!\!\perp = \{\mathsf{p} \in \Lambda_c \star \Pi;\ \mathsf{p} \succ \alpha_0 \star \pi\}$ (recall that $\Lambda_c \star \Pi$ is the set of processes). We define a unary predicate X by putting $\|X0\| = \{\pi\}$, $\|X(i+1)\| = \emptyset$ for every $i \in \mathbb{N}$. Thus, we have $\alpha_0 \Vdash X0$ and $t \Vdash \forall y(Xy \to Xsy)$ for every $t \in \Lambda_c$. But $\xi \Vdash Int(0)$, thus $\xi \Vdash \forall y(Xy \to Xsy), X0 \to X0$. It follows that $\xi \star \lambda x\, \alpha_1.\alpha_0.\pi \in \perp\!\!\!\perp$. In other words, $\xi \star \lambda x\, \alpha_1.\alpha_0.\pi \succ \alpha_0 \star \pi$.
We now take $\perp\!\!\!\perp = \{\mathsf{p} \in \Lambda_c \star \Pi;\ \mathsf{p} \succ \alpha_1 \star \pi\}$ and we define the unary predicate X by putting $\|X0\| = \emptyset$, $\|X(i+1)\| = \{\pi\}$ for every $i \in \mathbb{N}$. Therefore, we have $t \Vdash X0$ for every $t \in \Lambda_c$ and $\lambda x\, \alpha_1 \Vdash \forall y(Xy \to Xsy)$. But $\xi \Vdash Int(1)$, thus $\xi \Vdash \forall y(Xy \to Xsy), X0 \to X1$. It follows that $\xi \star \lambda x\, \alpha_1.\alpha_0.\pi \in \perp\!\!\!\perp$. In other words, $\xi \star \lambda x\, \alpha_1.\alpha_0.\pi \succ \alpha_1 \star \pi$. This is obviously a contradiction with the preceding result, because no process can reduce both to $\alpha_0 \star \pi$ and $\alpha_1 \star \pi$.

Q.E.D.

A solution for axiom 3 (we shall give another method later) is simply to remove it by means of the following well known property :

Every proof of a formula Θ in second order classical logic, using axioms 1, 2 and 3, can be converted into a proof of Θ^{Int} using axioms 1, 2 and the following axioms :

3'. $\forall x_1 \ldots \forall x_k \{Int[x_1], \ldots, Int[x_k] \to Int[f(x_1, \ldots, x_k)]\}$ *for each function symbol* f *of* $\mathcal{L}$.
Θ^{Int} is the formula that we obtain by restricting to Int all first order quantifiers in Θ. It is inductively defined as follows :
If A is atomic, then $A^{Int} \equiv A$; $(A \to B)^{Int} \equiv A^{Int} \to B^{Int}$;
$(\forall X\, A)^{Int} \equiv \forall X\, A^{Int}$; $(\forall x\, A)^{Int} \equiv \forall x(Int[x] \to A^{Int})$.
It remains to find for which functions from $\mathbb{N}^k$ into $\mathbb{N}$ the formulas 3' are realised. The solution is given by the following theorem :

Theorem 12. *Let* f *be a function symbol of arity* k*, which represents a (total) recursive function. Then the formula* $\forall x_1 \ldots \forall x_k \{Int[x_1], \ldots, Int[x_k] \to Int[f(x_1, \ldots, x_k)]\}$ *is realised.*

We first need a result about usual λ-calculus.

Notations.
Λ is the set of usual λ-terms.
If $t, u \in \Lambda$, then $t \simeq_\beta t'$ means that t is β-equivalent to t' ;
$t \succ t'$ means that t reduce to t' by *weak head reduction* : a step of weak head reduction is $(\lambda x\, u)vv_1 \ldots v_n \succ (u[v/x])v_1 \ldots v_n$.
The following lemma explains why we use the same symbol $\succ$ for weak head reduction of usual λ-terms and for the execution des processes.

Lemma 13.
Let $\xi, \eta, t_1, \ldots, t_k \in \Lambda$ *some usual closed* λ*-terms and let* $\pi \in \Pi$*. If* η *is not an application (in other words,* η *is a* λ*-constant or* $\eta = \lambda x\, \eta'$*) and if* $\xi \succ (\eta)t_1 \ldots t_k$*, then*
$\xi \star \pi \succ \eta \star t_1. \ldots .t_k.\pi$.

The proof is by induction on the length of the weak head reduction $\xi \succ (\eta)t_1 \ldots t_k$. The first step is $\xi = (\lambda y\, u)vv_1 \ldots v_n \succ (u[v/y])v_1 \ldots v_n$ and, by induction hypothesis :
$(u[v/y])v_1 \ldots v_n \star \pi \succ \eta \star t_1. \ldots .t_k.\pi$. Now, during the first $n-1$ reduction steps of the left hand side, the head of the process is an application ; since η is not, we did not reach yet the right hand process after these steps. Thus, we have :
$(u[v/y]) \star v_1 \ldots v_n.\pi \succ \eta \star t_1. \ldots .t_k.\pi$. It follows that :
$\xi \star \pi = (\lambda y\, u)vv_1 \ldots v_n \star \pi \succ (u[v/y]) \star v_1 \ldots v_n.\pi \succ \eta \star t_1. \ldots .t_k.\pi$.
Q.E.D.

Notations.
For $t, u \in \Lambda_c$, we define :
$(t)^n u$ pour $n \in \mathbb{N}$ by : $(t)^0 u = u$, $(t)^{n+1} u = (t)(t)^n u$.
$t \circ u$ by $\lambda x(t)(u)x$, where x does appear in t, u.

Lemma 14. *Let* f, a *be* λ*-constants and* $\xi \in \Lambda$ *such that* $\xi \simeq_\beta (f)^n a$.
Suppose that $\phi \Vdash \forall y(Xy \to Xsy)$ *and* $\alpha \Vdash X0$*. Then* $\xi[\phi/f, \alpha/a] \Vdash Xs^n 0$.

Proof by induction on n. If $n = 0$, then $\xi \simeq_\beta a$ and therefore $\xi \succ a$. If $\pi \in \|X0\|$, then $\xi \star \pi \succ a \star \pi$, by lemma 13. Thus $\xi[\phi/f, \alpha/a] \star \pi \succ \alpha \star \pi \in \perp\!\!\!\perp$.
If $n > 0$, then $\xi \succ (f)\eta$ with $\eta \simeq_\beta (f)^{n-1} a$. Let $\pi \in \|Xs^n 0\|$; then $\xi \star \pi \succ f \star \eta.\pi$, by lemma 13 and therefore $\xi[\phi/f, \alpha/a] \star \pi \succ \phi \star \eta[\phi/f, \alpha/a].\pi$. Now, $\phi \in |Xs^{n-1}0 \to Xs^n 0|$ and, by induction hypothesis, $\eta[\phi/f, \alpha/a] \in |Xs^{n-1}0|$. Thus $\phi \star \eta[\phi/f, \alpha/a].\pi \in \perp\!\!\!\perp$.
Q.E.D.

Theorem 15. *Let $n \in \mathbb{N}$ and $\nu \in \Lambda$ such that $\nu \simeq_\beta \lambda f \lambda x (f)^n x$. Then $\nu \Vdash Int[s^n 0]$.*

Let $\phi \Vdash \forall y(Xy \to Xsy)$, $\alpha \Vdash X0$ and $\pi \in \|Xs^n 0\|$; we must show that $\nu \star \phi.\alpha.\pi \in \perp\!\!\!\perp$. Since $\nu \simeq_\beta \lambda f \lambda x (f)^n x$, we have $\nu \succ \lambda f\, \eta$, $\eta \succ \lambda a\, \xi$ and $\xi \simeq_\beta (f)^n a$. By lemma 13, we have $\nu \star \phi.\alpha.\pi \succ \lambda f\, \eta \star \phi.\alpha.\pi \succ \eta[\phi/f] \star \alpha.\pi$. Again by lemma 13, we have :
$\eta \star \alpha.\pi \succ \lambda a\, \xi \star \alpha.\pi \succ \xi[\alpha/a] \star \pi$ and thus $\eta[\phi/f] \star \alpha.\pi \succ \xi[\phi/f, \alpha/a] \star \pi$. Finally, we have $\nu \star \phi.\alpha.\pi \succ \xi[\phi/f, \alpha/a] \star \pi$. But, by lemma 14, we have $\xi[\phi/f, \alpha/a] \Vdash Xs^n 0$ and therefore $\nu \star \phi.\alpha.\pi \succ \xi[\phi/f, \alpha/a] \star \pi \in \perp\!\!\!\perp$.

Q.E.D.

We can now prove theorem 12. For simplicity, we suppose $k = 1$; thus, we have a recursive function $f : \mathbb{N} \to \mathbb{N}$. Let $\phi \in \Lambda$ be a λ-term which represents f : for each $n \in \mathbb{N}$, we have $\phi s^n 0 \simeq_\beta \lambda f \lambda x (f)^p x$ with $p = f(n)$. Thus $\phi s^n 0 \succ \lambda f\, \xi$ and, by theorem 15, we have $\lambda f\, \xi \Vdash Int[s^p 0]$. Let $\pi \in \|Int[s^p 0]\|$; we have $\lambda f\, \xi \star \pi \in \perp\!\!\!\perp$. Now, by lemma 13, we have $\phi s^n 0 \star \pi \succ \lambda f\, \xi \star \pi$; but this reduction has necessarily at least one step, because $\phi s^n 0 \neq \lambda f\, \xi$ ($\phi s^n 0$ does not begin by λ). Therefore $\phi \star s^n 0.\pi \succ \lambda f\, \xi \star \pi \in \perp\!\!\!\perp$. Since this holds for each $\pi \in \|Int[s^p 0]\|$, theorem 10 shows that $T\phi \Vdash Int[s^n 0] \to Int[s^p 0]$, that is $T\phi \Vdash Int[n] \to Int[f(n)]$. Since this is true for every $n \in \mathbb{N}$, it follows that
$T\phi \Vdash \forall x(Int[x] \to Int[f(x)])$.

Q.E.D.

Arithmetical formulas

We can now consider the specification problem for arithmetical theorems. We begin by two simple forms : $\forall x \exists y[f(x,y) = 0]$ and $\exists x \forall y[f(x,y) = 0]$ where f is a recursive function. We suppose that these formulas are proved in classical second order arithmetic, using equational axioms written with total recursive function symbols (so that theorem 12 applies). In order to remove the recurrence axiom, we write these theorems in this form :
$\forall x[Int(x) \to \exists y\{Int(y), f(x,y) = 0\}]$ and $\exists x\{Int(x), \forall y[Int(y) \to f(x,y) = 0]\}$.

Theorem 16. *If $\vdash \theta : \forall x[Int(x), \forall y\{Int(y) \to f(x,y) \neq 0\} \to \bot]$, then for every $n \in \mathbb{N}$, $\kappa \in \Lambda_c$ and $\pi \in \Pi$, we have $\theta \star \underline{n}.T\kappa.\pi \succ \kappa \star s^p 0.\pi'$ with $f(n,p) = 0$ (T is defined in theorem 10).*

Remark. If we take for κ a " stop " instruction, we see that the program θ can compute, for each integer n, an integer p such that $f(n,p) = 0$. A " stop " instruction has no reduction rule when it comes in head position ; therefore, execution stops at this moment.
Of course, the theorem remains true if we use other axioms in the proof of the formula $\forall x \exists y[f(x,y) = 0]$, provided that these axioms are realized by suitable instructions. For instance, as we shall see later, we can use the axiom of dependent choice for this proof ; then, the program θ will contain a *clock instruction* $\hbar$ the reduction rule of which is :
$\hbar \star t.\pi \succ t \star \underline{n}.\pi$, where n is an integer which is the current time.

Proof. We choose $n \in \mathbb{N}$ and we put $\perp\!\!\!\perp = \{\mathsf{p} \in \Lambda_c \star \Pi;\ \mathsf{p} \succ \kappa \star s^p 0.\rho$ with $\rho \in \Pi$ and $f(n,p) = 0\}$. By theorem 2 (adequation lemma), we have :
$(*) \quad \theta \Vdash Int(n), \forall y\{Int(y) \to f(n,y) \neq 0\} \to \bot$.
We have $\underline{n} \Vdash Int(n)$ by theorem 15. We show that $T\kappa \Vdash \forall y\{Int(y) \to f(n,y) \neq 0\}$, in other words that $T\kappa \Vdash Int(p) \to f(n,p) \neq 0$ for every integer p. By theorem 10, it is sufficient to

prove that $\kappa \star s^p 0.\rho \in \perp\!\!\!\perp$ for every $\rho \in \|f(n,p) \neq 0\|$. It is trivial if $f(n,p) \neq 0$ since $\|f(n,p) \neq 0\| = \emptyset$. If $f(n,p) = 0$, it is again true by definition of $\perp\!\!\!\perp$.
It follows from $(*)$ that $\theta \star T\kappa.\pi \in \perp\!\!\!\perp$ for every $\pi \in \Pi$, which is the desired result.

Q.E.D.

We consider now an arithmetical theorem Φ of the form :
$\exists x\{Int(x), \forall y[Int(y) \to f(x,y) = 0]\}$.
We define a game between two players called $\exists$ and $\forall$: $\exists$ plays an integer m, $\forall$ answers by an integer n ; the play stops as soon as $f(m,n) = 0$ and then $\exists$ won ; therefore $\forall$ wins if and only if the play does not stop.
Intuitively, $\exists$ is the " defender " of the theorem and $\forall$ " attacks " this theorem, searching to exhibit a counter-example. It is clear that $\exists$ has a winning strategy if and only if $\mathbb{N} \models \Phi$; moreover, in this case, there is an obvious strategy for $\exists$: simply play successively $0, 1, 2, \ldots$

We shall show (theorem 17) that the program associated with a proof of Φ behaves exactly as a winning strategy for $\exists$ in this game. For this, we need to add an instruction κ to our programming language, in order to allow an interactive execution. The execution rule of κ is the following :

$$\kappa \star s^n 0.\xi.\pi \succ \xi \star s^p 0.\kappa_{np}.\pi'$$

for $n, p \in \mathbb{N}$, $\xi \in \Lambda_c$, $\pi, \pi' \in \Pi$; s is a fixed λ-term for the successor in Church integers ; κ_{np} is a double sequence of " stop " instructions.
Observe that *this execution rule is non deterministic*, since the integer p and the stack π' are arbitrary. The intuitive meaning of this rule is as follows : in the left hand side, the program, which is also the player $\exists$, plays the integer n ; in the right hand side, the attacker $\forall$ answers by playing p and the execution goes on ; κ_{np} keeps the trace of the ordered pair (n,p) of integers.

Theorem 17. *If $\vdash \theta : [\exists x \forall y(f(x,y) = 0)]^{Int}$, then every reduction of $\theta \star T\kappa.\pi$ ends up into $\kappa_{np} \star \pi'$ with $f(n,p) = 0$. T is the storage operator defined in theorem 10.*

We take for $\perp\!\!\!\perp$ the set of processes *every* reduction of which ends up into $\kappa_{np} \star \pi'$ *with* $f(n,p) = 0$ and $\pi' \in \Pi$. We must show that $\theta \star T\kappa.\pi \in \perp\!\!\!\perp$ for every $\pi \in \Pi$.
By theorem 2, we have $\theta \Vdash \forall x[Int(x), \forall y(Int(y) \to f(x,y) = 0) \to \bot] \to \bot$.
Therefore, by definition of $\Vdash$, it is sufficient to show that :
$T\kappa \Vdash \forall x[Int(x), \forall y(Int(y) \to f(x,y) = 0) \to \bot]$.
Let $n \in \mathbb{N}$; we must show that $T\kappa \Vdash Int[s^n 0] \to [\forall y(Int(y) \to f(n,y) = 0) \to \bot]$.
By theorem 10, we only need to show that, if $\pi \in \Pi$ and $\xi \Vdash \forall y(Int(y) \to f(n,y) = 0)$ then $\kappa \star s^n 0.\xi.\pi \in \perp\!\!\!\perp$. By definition of $\perp\!\!\!\perp$, it is sufficient to show that $\xi \star s^p 0.\kappa_{np}.\pi \in \perp\!\!\!\perp$ for every $p \in \mathbb{N}$ and $\pi \in \Pi$. But, by hypothesis on ξ, for every $p \in \mathbb{N}$ and $\varpi \in \|f(n,p) = 0\|$, we have $\xi \star s^p 0.\varpi \in \perp\!\!\!\perp$. Therefore, it suffices to show that $\kappa_{np}.\pi \in \|f(n,p) = 0\|$ for every $p \in \mathbb{N}$ and $\pi \in \Pi$.
If $f(n,p) = 0$, then $\|f(n,p) = 0\|$ is $\|\forall X(X \to X)\|$ that is to say :
$\{t.\rho;\, t \in \Lambda_c, \rho \in \Pi, t \star \rho \in \perp\!\!\!\perp\}$. But, by definition of $\perp\!\!\!\perp$, we have also $\kappa_{np} \star \pi \in \perp\!\!\!\perp$ and therefore $\kappa_{np}.\pi \in \|f(n,p) = 0\|$.
If $f(n,p) \neq 0$, then $\|f(n,p) = 0\| = \|\top \to \bot\|$ that is to say $\{t.\rho;\, t \in \Lambda_c, \rho \in \Pi\}$; thus, we have again $\kappa_{np}.\pi \in \|f(n,p) = 0\|$.

Q.E.D.

It follows that every proof of Φ in classical analysis gives rise to an interactive program which wins against any attacker.

Indeed, at each answer of the attacker $\forall$, the program, which is nobody else than the player $\exists$, provides an *object* $(s^n 0, \xi)$ constituted with an integer n (the " provisional solution ") and an *exception handler* ξ, which will be used in case of a pertinent answer of the opponent.
They are the two arguments of κ which can therefore be considered as a *pointer* towards this object.

We can, without any supplementary effort, add a universal quantifier, which gives the

Theorem 18. *If* $\vdash \theta : [\forall x \exists y \forall z (f(x,y,z) = 0)]^{Int}$*, then every reduction of* $\theta \star \underline{m}.T\kappa.\pi$ *ends up into* $\kappa_{np} \star \pi'$ *with* $f(m,n,p) = 0$.

The generalization to the case of an arithmetical formula with an arbitrary number of alternating quantifiers is given in [5].

We shall now prove that *every arithmetical formula which is true in* $\mathbb{N}$ *is realized.* We use for this the trivial winning strategy of the player $\exists$, which we express by a program.

Theorem 19.
Let θ *be a closed* λ*-term such that* $\theta ti \succ (ti)(\theta t)(s)i$*. If* $\mathbb{N} \models \forall x \exists y \forall z [f(x,y,z) \neq g(x,y,z)]$*, then* $\lambda x\, \theta x 0 \Vdash \forall x \exists y \{Int(y), \forall z (f(x,y,z) \neq g(x,y,z))\}$.

Remark. This formula is obviously stronger than $[\forall x \exists y \forall z (f(x,y,z) \neq g(x,yz))]^{Int}$ which is therefore also realized.

Suppose that there exists an integer m such that :
$\lambda x\, \theta x 0 \nVdash \forall y [Int(y), \forall z (f(m,y,z) \neq g(m,y,z)) \to \bot] \to \bot$. Thus, there exists $t \in \Lambda_c$ and $\pi_0 \in \Pi$ such that $t \Vdash \forall y [Int(y), \forall z (f(m,y,z) \neq g(m,y,z)) \to \bot]$ and $\theta t 0 \star \pi_0 \notin \perp\!\!\!\perp$.
Thus, we have $t \star 0.\theta t 1.\pi_0 \notin \perp\!\!\!\perp$ and therefore $\theta t 1 \nVdash \forall z (f(m,0,z) \neq g(m,0,z))$. This means that there exists $p_0 \in \mathbb{N}$ and $\pi_1 \in \Pi$ such that $f(m,0,p_0) = g(m,0,p_0)$ and $\theta t 1 \star \pi_1 \notin \perp\!\!\!\perp$.
Therefore, we have $t \star 1.\theta t 2.\pi_1 \notin \perp\!\!\!\perp$ and so on. Therefore we build, in this way, a sequence of integers $p_i (i \in \mathbb{N})$ such that $f(m,i,p_i) = g(m,i,p_i)$. This is a contradiction with the hypothesis that $\mathbb{N} \models \forall x \exists y \forall z [f(x,y,z) \neq g(x,y,z)]$.

Q.E.D.

This theorem generalizes to an arbitrary number of alternating quantifiers (see below the subsection " Arbitrary arithmetical formulas ").

Axiom of foundation in generic models

We obviously have $\mathbb{N} \models \forall x \exists y \forall z (x \neq y + z)$ (take $y = sx$). Thus, by theorem 19, we have $\lambda x\, \theta x 0 \Vdash \forall x \exists y \{Int(y), \forall z (x \neq y + z)\}$. We shall deduce that generic models are " well founded " :

Theorem 20. *The formula* $\forall x \exists! n \exists! y \{Int(n), y \simeq 0, x = n + y\}$ *is realized, where* $y \simeq 0$ *is the formula* $\forall z (y \neq sz)$.

It is sufficient to show that this formula is a logical consequence of the above formula and of identities $\forall x \forall y \forall z (f(x,y,z) = g(x,y,z))$ which are true in $\mathbb{N}$.
Existence : let p be the least integer such that $\forall z (x \neq z + p)$. Then $p \neq 0$ because $\forall z (x \neq z + 0)$ is false, by $z + 0 = z$. Threfore $p = sn$ and, by definition of p, there exists y such that $x = y + n$. We cannot have $y = sz$, otherwise $x = sz + n = z + sn = z + p$.

Unicity : suppose that $x = y + n = y' + n'$, with $Int(n)$, $Int(n')$ $y, y' \simeq 0$. Suppose $n \leq n'$, thus $n' = n + k$ with $Int(k)$. Therefore, we have $y + n = y' + n + k$, thus $y = y' + k$; since y is not a successor, we have $k = 0$, thus $n = n'$ and $y = y'$.

Q.E.D.

With each individual of the model, we can therefore associate an integer (of the model) and only one. In this way, we can consider integers as equivalence classes. It is another interpretation of the recurrence scheme : instead of restricting individuals to integers, we keep individuals and we restrict to predicates which are saturated for this equivalence relation. The two models of second order arithmetic obtained in this way are obviously isomorphic.

The following theorem is also an expression of well foundedness. It will be used later.

Theorem 21. *Let* Y *be the Turing combinator* : $\mathsf{Y} = AA$ *with* $A = \lambda a \lambda f(f)(a)af$. *Then* $\mathsf{Y} \Vdash \forall X\{\forall n(\bigcap_{m<n} |Xm| \to Xn) \to \forall n\, Xn\}$.

Let $X : \mathbb{N} \to \mathcal{P}(\Pi)$, $k \in \mathbb{N}$ and $t \Vdash \forall n(\bigcap_{m<n} |Xm| \to Xn)$. We show, by induction on k, that $\mathsf{Y} \star t.\pi \in \perp\!\!\!\perp$ for every $\pi \in Xk$. By induction hypothesis, we have $\mathsf{Y} \star t.\rho \in \perp\!\!\!\perp$ for every $\rho \in \bigcup_{m<k} Xm$ and therefore $\mathsf{Y}t \Vdash \bigcap_{m<n} |Xm|$. Thus, by hypothesis on t, we have $t \star \mathsf{Y}t.\pi \in \perp\!\!\!\perp$ for every $\pi \in Xk$. But we have $\mathsf{Y} \star t.\pi \succ t \star \mathsf{Y}t.\pi$ hence the result.

Q.E.D.

We define the predicate $x \neq_X y$ (predicate of order 3, where X is a propositional variable) by putting, for $m, n \in \mathbb{N}$: $\|m \neq_X n\| = \emptyset$ if $m \neq n$ and $\|m \neq_X n\| = \|X\|$ if $m = n$.
Lemma 22 tells us that this predicate is equivalent to $x = y \to X$.

Lemma 22. *The formula* $\forall X \forall x \forall y\{x \neq_X y \leftrightarrow (x = y \to X)\}$ *is realized. Indeed, we have :*
$\lambda x\, xI \Vdash \forall X \forall x \forall y[(x = y \to X) \to x \neq_X y]$ *and*
$\lambda x \lambda y\, yx \Vdash \forall X \forall x \forall y[x \neq_X y \to (x = y \to X)]$.

Same proof as for theorem 8, which is the particular case $X = \perp$.

Q.E.D.

The following theorem gives the specification associated with a formula of the form :
$\forall x \exists y\{Int(y), \forall z[f(x, y, z) \neq g(x, y, z)]\}$ when $\mathbb{N} \models \forall x \exists y \forall z[f(x, y, z) \neq g(x, y, z)]$.

Theorem 23. *Suppose that* $\mathbb{N} \models \forall x \exists y \forall z[f(x, y, z) \neq g(x, y, z)]$ *; let* θ *be a proof-like term such that* $\theta \Vdash \forall x \exists y\{Int(y), \forall z[f(x, y, z) \neq g(x, y, z)]\}$ *for every choice of* $\perp\!\!\!\perp$ *(such a term exists, by theorem 19). Then we have, for every stack* π *:*
$\theta \star T\kappa.\pi \succ \kappa \star \underline{n}_0.t_0.\pi$, $t_0 \star \pi \succ \kappa \star \underline{n}_1.t_1.\pi$, ..., $t_i \star \pi \succ \kappa \star \underline{n}_{i+1}.t_{i+1}.\pi$, ... *and for every* $m \in \mathbb{N}$ *there exists* i *such that* $\mathbb{N} \models \forall z[f(m, n_i, z) \neq g(m, n_i, z)]$.

By lemma 22, the formula $\forall X \forall x\{\forall y[Int(y), \forall z(f(x, y, z) \neq_X g(x, y, z)) \to X] \to X\}$ is equivalent to $\forall X \forall x\{\forall y[Int(y) \to \exists z(f(x, y, z) = g(x, y, z)) \vee X] \to X\}$, and therefore to :
$\forall x \exists y\{Int(y), \forall z[f(x, y, z) \neq g(x, y, z)]\}$. We can thus suppose :
$\theta \Vdash \forall X \forall x\{\forall y[Int(y), \forall z(f(x, y, z) \neq_X g(x, y, z)) \to X] \to X\}$.
Consider an integer m and a stack π ; we put $\|X\| = \{\pi\}$ and
$\perp\!\!\!\perp = \{\mathsf{p} \in \Lambda_c \star \Pi;\ \mathsf{p} \succ \kappa \star \underline{n}_0.t_0.\pi$, $t_0 \star \pi \succ \kappa \star \underline{n}_1.t_1.\pi$, ..., $t_i \star \pi \succ \kappa \star \underline{n}_{i+1}.t_{i+1}.\pi$, ... and

$(\exists i \in \mathbb{N})(\forall z \in \mathbb{N})\, f(m, n_i, z) \neq g(m, n_i, z)\}$.
It is therefore sufficient to prove that $T\kappa \Vdash \forall y[Int(y), \forall z(f(m, y, z) \neq_X g(m, y, z)) \to X$ for every $m \in \mathbb{N}$. By theorem 10, we have to show that $\kappa \star \underline{n}.t.\pi \in \perp\!\!\!\perp$ for every $n \in \mathbb{N}$ and every t such that $t \Vdash \forall z(f(m, n, z) \neq_X g(m, n, z))$. In other words, we must check :
i) $(\forall z \in \mathbb{N})(f(m, n, z) \neq_X g(m, n, z)) \Rightarrow \kappa \star \underline{n}.t.\pi \in \perp\!\!\!\perp$; it is clear, by definition of $\perp\!\!\!\perp$.
ii) $t \Vdash X \Rightarrow \kappa \star \underline{n}.t.\pi \in \perp\!\!\!\perp$, or else $t \star \pi \in \perp\!\!\!\perp \Rightarrow \kappa \star \underline{n}.t.\pi \in \perp\!\!\!\perp$; it is clear again by definition of $\perp\!\!\!\perp$.

Q.E.D.

For instance, the formula $\forall x \exists y\{Int(y), \forall z(x \neq y + z)\}$ is a specification for the " streams of integers " which take arbitrarily high values. If we define $e : \mathbb{N}^2 \to \{0, 1\}$ by $e(m, n) = 1 \Leftrightarrow m = n$, the formula $\forall x \exists y\{Int(y), e(x, y) \neq 0\}$ is a specification for the " streams of integers " which take every possible value. It is the same for the formula (which is equivalent to the former one) : $\forall x \exists y\{Int(y), \forall z(x^2 + y^2 \neq 2xy + sz)\}$.

Arbitrary arithmetical formulas

Remark. This subsection may be skipped at first reading.

Let us now consider a formula Φ of the form :
$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k(f(x_1, y_1, \ldots, x_k, y_k) \neq 0)$ where $f : \mathbb{N}^{2k} \to \mathbb{N}$ is an *arbitrary* function.
We define a game associated with Φ (players $\exists$ and $\forall$ are respectively the " defender " and the " attacker " of the formula Φ) ; we also define the notion of *reached position* at a given moment :
A *position* of the game is a finite sequence of integers $n_1 p_1 \ldots n_i p_i$ $(0 \leq i \leq k)$. The player $\exists$ chooses first an *already reached* position $n_1 p_1 \ldots n_i p_i$ with $0 \leq i < k$ and an integer n_{i+1} ; then, the player $\forall$ chooses an integer p_{i+1}. The position $n_1 p_1 \ldots n_{i+1} p_{i+1}$ is then reached.
If $i + 1 = k$ and $f(n_1, p_1, \ldots, n_k, p_k) \neq 0$, then the play stops and $\exists$ *won*. In every other case, the play goes on. Therefore $\forall$ wins if and only if the play does not stop.

It is easy to see that $\mathbb{N} \models \Phi$ if and only if the player $\exists$ has a winning strategy for this game. Moreover, we can effectively (and very simply) describe such a strategy. It does not even depend on the function f, but only on the number k of quantifiers :
The player $\exists$ uses an effective enumeration of $\mathbb{N}^k$. When he reaches the k-uple $n_1 \ldots n_k$, he chooses the longest already reached position of the form $n_1 p_1 \ldots n_i p_i$. We have $i < k$ because this position was reached for a k-uple of integers which is different from $n_1 \ldots n_k$. Then, he plays successively $n_{i+1}, \ldots, n_k$ without taking any account of the choices of player $\forall$. Then, he continues with the next k-uple of integers.
If the play is infinite, we obtain k functions $\phi_i(x_1, \ldots, x_i)$ such that :

$$\mathbb{N} \models \forall x_1 \ldots \forall x_k \{f[x_1, \phi_1(x_1), x_2, \phi_2(x_1, x_2), \ldots, x_k, \phi_k(x_1, \ldots, x_k)] = 0\}$$

which is the Skolem form of $\neg\Phi$; thus $\mathbb{N} \models \neg\Phi$.

Conversely, if $\mathbb{N} \models \neg\Phi$, there exist k functions $\phi_i(x_1, \ldots, x_i)$ such that the Skolem form of $\neg\Phi$ is satisfied. They obviously provide a winning strategy for the opponent $\forall$.

Theorem 24. *Suppose que* $\mathbb{N} \models \exists x_1 \forall y_1 \ldots \exists x_k \forall y_k(f(x_1, y_1, \ldots, x_k, y_k) \neq 0)$
where $f : \mathbb{N}^{2k} \to \mathbb{N}$ *is an arbitrary function. Then the formula :*
$\exists x_1\{Int(x_1), \forall y_1 \exists x_2\{Int(x_2), \ldots \exists x_k\{Int(x_k), \forall y_k(f(x_1, y_1, \ldots, x_k, y_k) \neq 0)\} \ldots\}$
is realized by a proof-like term which is independent of f *(it is, in fact, a usual closed* λ*-term).*

Remarks.
This formula is obviously stronger than Φ^{Int}, which is therefore also realized. Theorem 24 is a generalization of theorem 19.
The integers of a generic model give therefore a first order model which is elementarily equivalent to $\mathbb{N}$.
The term which realizes Φ is obtain by means of a program for the winning strategy given above.

We shall only consider the case where $\mathbb{N} \models \Phi \equiv \exists m \forall n \exists p (f(m,n,p) \neq 0)$.
We then write, in the following way, a proof-like term *independent of* f which realizes the formula $\exists m\{Int(m), \forall n \exists p\{Int(p), f(m,n,p) \neq 0)\}\}$:

m, p are λ-variables, which represent Church integers ; s_0 and s_1 are closed usual λ-terms such that the ordered pair (s_0mp, s_1mp) is the successor of the pair (m,p) in a fixed recursive enumeration of $\mathbb{N}^2$ which begins with $(0,0)$.
σ is a variable which represents a finite sequence of ordered pairs $(m_0, \eta_0), \ldots, (m_k, \eta_k)$ where m_i is a Church integer and η_i a λ_c-term. In other words, σ represents a λ_c-term of the form $\lambda h \lambda x (hm_0\eta_0) \cdots (hm_k\eta_k)x$.
H and Σ are closed usual λ-terms, defined as follows :
$H\sigma\eta m \succ \eta_i$ for the first i such that $m = m_i$, if m is equal to one of the integers m_i of the sequence σ ; otherwise $H\sigma\eta m \succ \eta$.
$\Sigma\sigma\eta m \succ \sigma$ if m is equal to one of the integers m_i of the sequence σ ; otherwise $\Sigma\sigma\eta m \succ \tau$, where τ is the sequence obtained by adding the pair (m, η) at the end of the sequence σ.

Theorem 25. *Let $f : \mathbb{N}^3 \to \mathbb{N}$ be an arbitrary function such that $\mathbb{N} \models \exists m \forall n \exists p (f(m,n,p) \neq 0)$.*
Define a closed usual λ-term θ by $(\theta\xi)\sigma mp \succ (\xi m)\lambda\eta(\eta' p)(\theta\xi)\sigma' m' p'$
with $\eta' = H\sigma\eta m$, $\sigma' = \Sigma\sigma\eta m$, $m' = s_0mp$, $p' = s_1mp$.
Then $\lambda x\, \theta x 000 \Vdash \forall m\{Int(m), \forall n[\forall p(Int(p) \to f(m,n,p) = 0) \to \bot] \to \bot\} \to \bot$.

Let $\xi \Vdash \forall m\{Int(m), \forall n[\forall p(Int(p) \to f(m,n,p) = 0) \to \bot] \to \bot\}$;
we have to show that $\theta\xi 000 \Vdash \bot$.

Lemma 26. *Let $\eta_0, \ldots, \eta_k \in \Lambda_c$ and $m, p, m_0, n_0, \ldots, m_k, n_k \in \mathbb{N}$, $m_0, \ldots, m_k$ being distinct integers ; suppose that $\eta_i \Vdash \forall p\{Int(p) \to f(m_i, n_i, p) = 0\}$ for $0 \leq i \leq k$. We put :*
$\sigma = \lambda h \lambda x (hm_0\eta_0) \cdots (hm_k\eta_k)x$. If $\theta\xi\sigma mp \nVdash \bot$, then there exist $n \in \mathbb{N}$ and $\eta \in \Lambda_c$ such that $\theta\xi\sigma' m' p' \nVdash \bot$, $\eta \Vdash \forall p\{Int(p) \to f(m,n,p) = 0\}$ and $f(m,n',p) = 0$ with $n' = n$ if $m \neq m_i$ for $0 \leq i \leq k$; otherwise, $n' = n_i$ for the (unique) integer i such that $m = m_i$.

By definition of θ, we have $\xi m t_{\sigma mp} \nVdash \bot$ with $t_{\sigma mp} = \lambda\eta(\eta' p)(\theta)\xi\sigma' m' p'$.
By hypothesis on ξ, it follows that $t_{\sigma mp} \nVdash \forall n[\forall p(Int(p) \to f(m,n,p) = 0) \to \bot]$.
Thus, there exist $n \in \mathbb{N}$ and $\eta \in \Lambda_c$, $\eta \Vdash \forall p\{Int(p) \to f(m,n,p) = 0\}$, such that $t_{mp\sigma}\eta \nVdash \bot$.
We have $\eta' = H\sigma\eta m$ and therefore $\eta' \Vdash \forall p\{Int(p) \to f(m,n',p) = 0\}$, by definition of H.
Now, we have : $t_{\sigma mp}\eta \succ \eta' p.\theta\xi\sigma' m' p'$.
If $f(m,n',p) \neq 0$, then $\eta' p \Vdash \top \to \bot$; therefore $t_{\sigma mp}\eta \Vdash \bot$ which is a contradiction.
Therefore, we have $f(m,n',p) = 0$ and it follows that $\eta' p \Vdash 0 = 0$, that is $\eta' p \Vdash \bot \to \bot$;
if $\theta\xi\sigma' m' p' \Vdash \bot$, we get once more the contradiction $t_{\sigma mp}\eta \Vdash \bot$.
Q.E.D.

Now suppose that $\theta\xi 000 \nVdash \bot$; applying iteratively lemma 26, we obtain a sequence $(m_i, n_i, p_i, \eta_i)_{i \in \mathbb{N}}$ such that :
• $(m_i, p_i)_{i \in \mathbb{N}}$ is the fixed enumeration of $\mathbb{N}^2$ which begins with $(0,0)$;

• if we put $\phi(m) = n_j$ for the first j such that $m = m_j$, then $f(m_i, \phi(m_i), p_i) = 0$ for every $i \in \mathbb{N}$.
It follows that $\mathbb{N} \models \forall m \forall p[f(m, \phi(m), p) = 0]$ and therefore $\mathbb{N} \not\models \exists m \forall n \exists p[f(m, n, p) = 1]$.
Q.E.D.

Without any supplementary effort, we can add two universal quantifiers : in head position, a quantifier $\forall l$ is justified because the λ-term which we got does not depend on f. The last quantifier $\forall q$ is justified, because the equation $f(m, n, p) = 0$ can be written $f(m, n, p) \neq 0 \to \bot$, which only takes the truth values $\top \to \bot$ and $\bot \to \bot$. We replace it with :
$\forall q(f(l, m, n, p, q) \neq 0) \to \bot$ which has the same property. Therefore, we get :

Theorem 27. *Let $f : \mathbb{N}^5 \to \mathbb{N}$ be an arbitrary function such that*
$\mathbb{N} \models \forall l \exists m \forall n \exists p \forall q(f(l, m, n, p, q) \neq 0)$ *and let θ be the closed λ-term defined in theorem 25. Then we have :* $\lambda x\, \theta x 000 \Vdash \forall l \exists m\{Int(m), \forall n \exists p\{Int(p), \forall q[f(l, m, n, p, q) \neq 0]\}\}$
which is the formula :
$\forall l[\forall m\{Int(m), \forall n[\forall p(Int(p), \forall q(f(l, m, n, p, q) \neq 0) \to \bot) \to \bot] \to \bot\} \to \bot]$.

The axiom of choice in analysis

In order to get programs from proofs in classical analysis, it remains to realize *the axiom of dependent choice* or a slightly weaker form : *the axiom of countable choice.* This axiom is, indeed, used very often in analysis.
We consider first the axiom of countable choice which is technically simpler to realize. It consists of the following axiom scheme :

$$\forall \vec{Y} \exists Z \forall x(\exists X\, F[x, X, \vec{Y}] \to F[x, Z(x, y)/Xy, \vec{Y}]).$$

F is a second order formula in which the binary predicate variable Z does not appear ; X is of arity 1 ; $\vec{Y} = (Y_1, \ldots, Y_n)$ is a finite sequence of second order variables (which are, in fact, parameters).
We shall rather write this axiom in contrapositive form :

(CAC) $$\forall \vec{Y} \exists Z \forall x(F[x, Z(x, y)/Xy, \vec{Y}] \to \forall X\, F[x, X, \vec{Y}]).$$

This means that $Z(x, y)$ is a counter-example to $\forall X\, F[x, X]$, if there is one.
In order to realize this axiom scheme, we add a new instruction, which is denoted by χ ; its reduction rule is given as follows : consider a fixed bijection $n \mapsto \pi_n$ from $\mathbb{N}$ onto Π (which we do not even need to suppose recursive) ; let $\pi \mapsto n_\pi$ be the inverse function. For each $n \in \mathbb{N}$, we denote by $\underline{n}$ a fixed λ-term which is $\simeq_\beta \lambda f \lambda x(f)^n x$, for instance $\lambda f \lambda x(f)^n x$ itself, or $s^n 0$, where s is a λ-term for the successor. The reduction rule for χ is then the following :

$$\chi \star t.\pi \succ t \star \underline{n_\pi}.\pi \quad \text{for every } t \in \Lambda_c \text{ and } \pi \in \Pi.$$

Remark. We shall examine later the intuitive meaning of this reduction rule.
Of course, from now on, the set $\perp\!\!\!\perp$ of processes is supposed to be cc-saturated for this new notion of reduction.

Theorem 28.
Let $F[x, X]$ be a formula with parameters, X being a unary predicate variable. There exists a function $U : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ such that :
$\chi \Vdash \forall x\{\forall n(Int[n] \to F[x, U(x, n, y)/Xy]) \to \forall X\, F[x, X]\}$.

For each individual x and each stack π, we have :

$$\pi \in \|\forall X\, F[x,X]\| \Leftrightarrow (\exists R \in \mathcal{P}(\Pi)^{\mathbb{N}})\, \pi \in \|F[x,R/X]\|.$$

By the axiom of countable choice, there exists a function $U : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ such that : $\pi \in \|\forall X\, F[x,X]\| \Leftrightarrow \pi \in \|F[x,U(x,n_\pi,y)/Xy]\|$. Let us show that U has the desired property : we consider an individual $x \in \mathbb{N}$, $t \in |\forall n(Int[n] \to F[x,U(x,n,y)/Xy])|$ and $\pi \in \|\forall X\, F[x,X]\|$. We have to show that $\chi \star t.\pi \in \perp\!\!\!\perp$. By the reduction rule of χ, it is sufficient to prove that $t \star \underline{n}_\pi.\pi \in \perp\!\!\!\perp$. But this follows immediately from :

- $t \Vdash Int(s^{n_\pi}0) \to F[x,U(x,n_\pi,y)/Xy]$ which is true by hypothesis on t ;
- $\underline{n}_\pi \Vdash Int(s^{n_\pi}0)$ by theorem 15 ;
- $\pi \in \|F[x,U(x,n_\pi,y)/Xy]\|$ by hypothesis on π and by definition of U.

Q.E.D.

The formula $\exists U \forall x\{\forall n(Int[n] \to F[x,U(x,n,y)/Xy]) \to \forall X\, F[x,X]\}$ is therefore realized by $\lambda x\, x\chi$. But this formula implies, in classical second order logic, the countable axiom of choice (CAC), that is $\exists Z \forall x\{F[x,Z(x,y)/Xy]) \to \forall X\, F[x,X]\}$.

The proof is easy : it is sufficient, by means of the comprehension scheme, to define $Z(x,y)$ by the following formula : " $U(x,n,y)$ for the first integer n such that $\neg F[x,U(x,n,y)/Xy]$ if there is one ". Notice that this proof uses the excluded middle ; therefore, the λ_c-term we get to realize CAC contains the instructions χ and cc.

However, a careless formalization of this proof will give a complicated and not very comprehensible term. It is therefore interesting to give an explicit and more refined method, in order to obtain a simple term. This is done below in the section " A program for the countable choice axiom ".

The axiom of dependent choice

It is the following axiom scheme :

$$\forall X \exists Y\, H[X,Y] \to \exists Z \forall k(Int[k] \to H[Z(k,y)/Xy, Z(k+1,y)/Yy])$$

for every formula H which does not contain the variable Z.

We write this scheme in the following form :

(DC) $\quad \exists Z \forall k(Int(k), F[Z(k,y)/Xy, Z(k+1,y)/Yy] \to \forall Y\, F[Z(k,y)/Xy, Y])$

for every formula F which does not contain the variable Z.

The first form follows immediately from this one, by setting $F \equiv \neg H$. Conversely, we get the second form if we set $H(X,Y) \equiv [F(X,Y) \to \forall Y\, F(X,Y)]$ in the first one : indeed, $\forall X \exists Y\, H[X,Y]$ is then provable.

Let $<x,y>$ be a symbol of binary function, which represents a bijection of $\mathbb{N}^2$ onto $\mathbb{N}$. We suppose it to be recursive so that theorem 12 applies.

Lemma 29. *For every $U : \mathbb{N}^2 \to \mathcal{P}(\Pi)$, there exists $V : \mathbb{N}^2 \to \mathcal{P}(\Pi)$ such that :*
$\chi \Vdash \forall n\{Int[n] \to F[U(x,y)/Xy, V(<n,x>,y)/Yy]\} \to \forall Y\, F[U(x,y)/Xy, Y]$.

Same proof as for theorem 28. For each individual x and each stack π, we have :

$$\pi \in \|\forall Y\, F[U(x,y)/Xy, Y]\| \Leftrightarrow (\exists R \in \mathcal{P}(\Pi)^{\mathbb{N}})\, \pi \in \|F[U(x,y)/Xy, R/Y]\|.$$

It follows, by the countable choice axiom, that there exists a function $V : \mathbb{N}^2 \to \mathcal{P}(\Pi)$ such that $\pi \in \|\forall Y\, F[U(x,y)/Xy, Y]\| \Leftrightarrow \pi \in \|F[U(x,y)/Xy, V(<n_\pi,x>,y)/Yy]\|$.

We now show that V has the desired property : consider $\pi \in \|\forall Y\, F[U(x,y)/Xy, Y]\|$ and

$t \in |\forall n\{Int[n] \to F[U(x,y)/Xy, V(<n,x>,y)/Yy]\}|$. We have to show that $\chi \star t.\pi \in \perp\!\!\!\perp$, that is $t \star \underline{n}_\pi.\pi \in \perp\!\!\!\perp$. But this follows immediately from :

- $t \Vdash Int(s^{n_\pi}0) \to F[U(x,y)/Xy, V(<n_\pi,x>,y)/Yy]$ by hypothesis on t ;
- $\underline{n}_\pi \Vdash Int(s^{n_\pi}0)$ by theorem 15 ;
- $\pi \in \|F[U(x,y)/Xy, V(<n_\pi,x>,y)/Yy]\|$ by hypothesis on π and by definition of V.

Q.E.D.

Theorem 30. *Let $F[X,Y]$ be a formula with parameters, X, Y being unary predicate variables. Then, there exists $A : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ such that :*

$$\chi \Vdash \forall x \forall k(\forall n\{Int[n] \to F[A(k,x,y)/Xy, A(k+1,<n,x>,y)/Yy]\} \to \forall Y\, F[A(k,x,y)/Xy, Y]).$$

By lemma 29, we have $(\forall U \in \mathcal{P}(\Pi)^{\mathbb{N}^2})(\exists V \in \mathcal{P}(\Pi)^{\mathbb{N}^2})\Phi(U,V)$ where $\Phi(U,V)$ is the formula $\chi \Vdash \forall n\{Int[n] \to F[U(x,y)/Xy, V(<n,x>,y)/Yy]\} \to \forall Y\, F[U(x,y)/Xy, Y]$.
Therefore, we get the result by application of the axiom of dependent choice (in the usual form, i.e. the first one) to the formula $\Phi(U,V)$.

Q.E.D.

By theorem 30, in order to realize the axiom scheme (DC), it is sufficient to derive it, in classical second order logic, from the formula :

$$\widehat{F}[A] \equiv \forall x \forall k(\forall n\{Int[n] \to F[A(k,x,y)/Xy, A(k+1,<n,x>,y)/Yy]\} \to \forall Y F[A(k,x,y)/Xy, Y]).$$

But the formula $\widehat{F}[A]$ is trivially equivalent to $\forall x \forall k \exists n\{Int[n], G[x,k,n,A]\}$ with :
$G[x,k,n,A] \equiv F[A(k,x,y)/Xy, A(k+1,<n,x>,y)/Yy] \to \forall Y F[A(k,x,y)/Xy, Y]$.
Thus, we can define inductively a sequence of integers n_k, by the conditions $n_0 = 0$ and $n_{k+1} = <n, n_k>$ for the first integer n such that $G[n_k, k, n, A]$. Now, if we define $Z(k,y)$ by $A(k, n_k, y)$, we get $F[Z(k,y)/Xy, Z(k+1,y)/Yy] \to \forall Y\, F[Z(k,y)/Xy, Y]$ for each integer k, that is (DC).

Variants and interpretations

We get the same results using, instead of χ, a dual instruction χ^*, which operates on terms instead of stacks. Consider a bijection $n \mapsto \tau_n$ from $\mathbb{N}$ onto Λ_c and let $t \mapsto n_t$ be the inverse function. Then the reduction rule for χ^* is the following :

$$\chi^* \star t.\pi \succ t \star \underline{n}_t.\pi \quad \text{for every } t \in \Lambda_c \text{ and } \pi \in \Pi.$$

We have the analogue of theorem 28, the proof being a little more complicated.

Theorem 31.
Let $F[x,X]$ be a formula with parameters, X being a unary predicate variable. There exists $U : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ such that : $\chi^* \Vdash \forall x\{\forall n(Int[n] \to F[x, U(x,n,y)/Xy]) \to \forall X\, F[x,X]\}$.

For each $n \in \mathbb{N}$ we put $P_n(\perp\!\!\!\perp) = \{\pi \in \Pi;\ \tau_n \star \underline{n}.\pi \notin \perp\!\!\!\perp\}$. For each individual x, we have $\|\forall X\, F[x,X]\| = \bigcup\{\|F[x,R/X]\|;\ R \in \mathcal{P}(\Pi)^{\mathbb{N}}\}$. It follows that, given $x, n \in \mathbb{N}$ such that $P_n(\perp\!\!\!\perp) \cap \|\forall X\, F[x,X]\| \neq \emptyset$, there exists a function $R : \mathbb{N} \to \mathcal{P}(\Pi)$ such that :
$P_n(\perp\!\!\!\perp) \cap \|F[x, Ry/Xy]\| \neq \emptyset$. By the axiom of countable choice, there exists a function
$U : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ which has the following property :
if $P_n(\perp\!\!\!\perp) \cap \|\forall X\, F[x,X]\| \neq \emptyset$ then $P_n(\perp\!\!\!\perp) \cap \|F[x, U(x,n,y)/Xy]\| \neq \emptyset$.

Now, let $x \in \mathbb{N}$, $\pi \in \|\forall X\, F[x,X]\|$ and $t \in |\forall n(Int[n] \to F[x, U(x,n,y)/Xy])|$. We have to show that $\chi^* \star t.\pi \in \perp\!\!\!\perp$ and, by the reduction rule of χ^*, it is sufficient to show that : $t \star \underline{n}_t.\pi \in \perp\!\!\!\perp$. If it is not true, we put $n = n_t$, and therefore $\tau_n = t$; we have then : $\pi \in P_n(\perp\!\!\!\perp) \cap \|\forall X\, F[x,X]\|$.
By definition of U, there exists $\pi' \in P_n(\perp\!\!\!\perp) \cap \|F[x, U(x,n,y)/Xy]\|$. From $\pi' \in P_n(\perp\!\!\!\perp)$, we deduce $\tau_n \star \underline{n}.\pi' \notin \perp\!\!\!\perp$. But, since $\pi' \in \|F[x, U(x,n,y)/Xy]\|$, it follows, by hypothesis on t, that $t \star \underline{n}.\pi' \in \perp\!\!\!\perp$, because $\underline{n} \Vdash Int[s^n 0]$ (theorem 15). It is a contradiction, because $t = \tau_n$.

Q.E.D.

We show, in the same way, the analogue of lemma 29. We can then realize the axioms of denumerable and dependent choice, by a pure repetition of the above arguments, using χ^* instead of χ.

We now remark that instructions analogous to χ and χ^* are very common in programmation :

The 'quote' instruction

The reduction rule of instruction χ^*, which is :

$$(*) \qquad \chi^* \star t.\pi \succ t \star \underline{n}_t.\pi$$

shows that χ is very similar to the instruction `quote` of LISP. Indeed, $\underline{n}_t$ can be used in the same way that `(quote t)`, if we assume that $t \mapsto n_t$ is a recursive bijection from Λ_c onto $\mathbb{N}$. For instance, since n_u is, in this case, a recursive function of n_{tu}, we can define, by means of χ^*, an instruction χ' such that $\chi' \star \phi.\psi.\pi \succ \phi \star n_\psi.\pi$: let a be a closed λ-term such that $(a)\underline{n}_{tu} \simeq_\beta \underline{n}_u$ and let $\chi' = \lambda x \lambda y (\chi^*)(\lambda d\, x \circ a) y$. As this example shows, the instruction χ^* is not compatible with β-reduction : in fact, we cannot replace $(\lambda d\, x \circ a)y$ with $x \circ a$.

The signature

An objection to the interpretation above is that `quote` is a reversible instruction, which is almost always used with its reverse `unquote` or `eval`. On the contrary, even if we suppose that χ (resp. χ^*) is one-to-one from Π (resp. Λ_c) onto $\mathbb{N}$, *we do not make use of the inverse function*. It is therefore more relevant (and much easier to implement) to interpret $\underline{n}_t$, in the reduction rule $(*)$ above, as the *signature* of t : it is an integer which is computed from the text of t and which is supposed to characterize it. Examples of well known usual programs for signature are MD-5 (for " Message Digest ") or SHA-1 (for " Secure Hash Algorithm ").
Since these programs give integers of fixed length (128 bits for MD-5 and 160 bits for SHA-1), they clearly do not provide an injective function from Λ_c into $\mathbb{N}$. But, during the execution of a given process, it is extremely improbable that two different terms with the same signature occur ; therefore, we consider such an event as impossible. And it is all we need in order to realize the axiom of dependent choice.

The clock

Here is another possible interpretation for the instructions χ and χ^*.
We observe that the application $n \mapsto \tau_n$ may be any *surjective* function from $\mathbb{N}$ onto Λ_c. Then the reduction rule of χ^* is :

$$\chi^* \star t.\pi \succ t \star \underline{n}.\pi.$$

where n is any integer such that $\tau_n = t$. This suggests the following interpretation :
χ^* is an input instruction ; when it comes in head position, the process $\chi^* \star t.\pi$ waits for an integer n which is provided by a human operator or some external process. Then the reduction $\chi^* \star t.\pi \succ t \star \underline{n}.\pi$ takes place and the execution goes on. The only condition is that we must be able to *retrieve* t *from* n ; in other words, the integers which are provided to the processes $\chi^* \star t.\pi$ and $\chi^* \star t'.\pi'$ with $t \neq t'$ must be different.
A very simple and natural way to obtain this behavior is by providing the integer n by means of a *clock*, since two different λ_c-terms cannot appear at the same time. In other words, let us be given a second process, which is executed in parallel with the principal process, and which only increments an integer at each reduction step. It is this process which provides the integer n when necessary, that is when χ^* arrives in head position in the principal process.
For an example of a model with a clock, look at the section below : " The standard generic model ".

A program for the countable choice axiom

We give here an explicit λ_c-term, containing the instruction χ, which realizes the axiom of countable choice and we explain its behaviour.

In the proof of theorem 28, we defined a function $U : \mathbb{N}^3 \to \mathcal{P}(\Pi)$ (ternary predicate).
We define now the binary predicate $V(x, y)$ by putting :
$V(x, y) \equiv \forall n\{\bigcap_{m<n} |\{\underline{m}\} \to F_U(x, m)|, \{\underline{n}\}, \neg F_U(x, n) \to U(x, n, y)\}$.
We use the notations $F_U(x, n)$ for $F[x, U(x, n, y)/Xy]$ and $F_V(x)$ for $F[x, V(x, y)/Xy]$; the notation $\{\underline{m}\} \to F_V(x, m)$ is defined page 3.
We propose to realize the formula $F_V(x) \to \forall X\, F[x, X]$, which is the axiom of denumerable choice. Now, we have :

Theorem 32 (Extensionality scheme).
i) For every formula $G[X]$, the formula : $Ext[G] \equiv \forall X\{G[X] \to G^*[X]\}$ is provable in intuitionistic second order logic, with :
$G^*[X] \equiv \forall U\{\forall y(Xy \to Uy), \forall y(Uy \to Xy) \to G[Uy/Xy]\}$.
ii) $\vdash \lambda g \lambda u \lambda u' \lambda v \lambda v'((g)v \circ u)u' \circ v' : Ext[G^*]$.

i) This is immediate, by recurrence on G.
ii) We have $g : G^*[X]$, $u : \forall y(Xy \to Uy)$, $u' : \forall y(Uy \to Xy)$,
$v : \forall y(Uy \to Vy)$, $v' : \forall y(Vy \to Uy) \vdash v \circ u : \forall y(Xy \to Vy)$, $u' \circ v' : \forall y(Vy \to Xy)$,
and therefore $((g)v \circ u)u' \circ v' : G[Vy/Xy]$.
Q.E.D.

Remark. $Ext[G]$ tells us that the formulas G and G^* are equivalent. The λ-term which is associated to the proof of this equivalence depends, in general, on G. It does not depend on it for formulas of the form G^*.

Therefore we shall, in fact, realize the denumerable axiom of choice in the form :

$$F_V^*(x) \to \forall X\, F[x, X]$$

where $F_V^*(x)$ is the formula $F^*[x, V(x, y)/Xy]$ that is to say :
$\forall X\{\forall y(V(x, y) \to Xy), \forall y(Xy \to V(x, y)) \to F[x, X]\}$.

A trivial variant of theorem 28 gives :

$$\chi \Vdash \forall x\{\forall n|\{\underline{n}\} \to F_U(x,n)| \to \forall X\, F[x,X]\}$$

(we replaced $Int[n]$ with $\{\underline{n}\}$, which has the advantage to avoid the introduction of a storage operator).

Therefore, we have now to realize : $F_V^*(x) \to \forall n|\{\underline{n}\} \to F_U(x,n)|$.

We have seen (theorem 21) that, if Y is the fixed point combinator of Turing, then :

Y $\Vdash \forall X\{\forall n(\bigcap_{m<n} |Xm| \to Xn) \to \forall n\, Xn\}$.

It follows that, by replacing Xn with $|\{\underline{n}\} \to F_U(x,n)|$, we have :

Y $\Vdash \forall n(\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\} \to F_U(x,n)) \to \forall n|\{\underline{n}\} \to F_U(x,n)|$.

So, we must now realize : $F_V^*(x), \bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\} \to F_U(x,n)$.

By the law of Peirce, it is sufficient to realize :

$F_V^*(x), \bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to F_U(x,n)$ which is :

$$\forall X\{\forall y(V(x,y) \to Xy), \forall y(Xy \to V(x,y)) \to F[x,X]\},$$
$$\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to F_U(x,n).$$

But n is now a fixed integer, so that we can put $Xy \equiv U(x,n,y)$; our goal is now :

$$\{\forall y(V(x,y) \to U(x,n,y)), \forall y(U(x,n,y) \to V(x,y)) \to F_U(x,n)\},$$
$$\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to F_U(x,n).$$

It is clearly a consequence of the two following formulas, that we have therefore to realize :

$$\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to (V(x,y) \to U(x,n,y))$$
$$\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to (U(x,n,y) \to V(x,y)).$$

By definition of V, it is immediate that the first formula is realized by $\lambda x \lambda n \lambda k \lambda v\, vxnk$.

Lemma 33. *There exists a closed usual λ-term, denoted by Comp, such that, for every $n, n' \in \mathbb{N}$: Comp $\underline{n}\,\underline{n'}xyz \succ x$ if $n < n'$; Comp $\underline{n}\,\underline{n'}xyz \succ y$ if $n' < n$; Comp $\underline{n}\,\underline{n'}xyz \succ z$ if $n = n'$.*

Trivial.

Q.E.D.

It follows that, if we put $\alpha = (k)(x')n$ and $\alpha' = (k')(x)n'$, we have :

$$\lambda x \lambda n \lambda k \lambda x' \lambda n' \lambda k'\, \mathrm{Comp}\, nn'\alpha\alpha'$$
$$\Vdash \bigcap_{m<n} |\{\underline{m}\} \to Xm|, \{\underline{n}\}, \neg Xn, \bigcap_{m'<n'} |\{\underline{m'}\} \to Xm'|, \{\underline{n'}\}, \neg Xn' \to (Yn \to Yn').$$

By replacing Xn with $F_U(x,n)$ and Yn with $U(x,n,y)$, we see that we have realized :

$$\bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n), \bigcap_{m'<n'} |\{\underline{m'}\} \to F_U(x,m')|, \{\underline{n'}\}, \neg F_U(x,n')$$
$$\to (U(x,n,y) \to U(x,n',y))$$

which, by the definition of V, can also be written as :

$$\lambda x \lambda n \lambda k \lambda u \lambda x' \lambda n' \lambda k'\, \mathrm{Comp}\, nn'\alpha\alpha' u$$
$$\Vdash \bigcap_{m<n} |\{\underline{m}\} \to F_U(x,m)|, \{\underline{n}\}, \neg F_U(x,n) \to (U(x,n,y) \to V(x,y)).$$

This is exactly the desired result and we obtain finally the following term which realizes the axiom of denumerable choice :

Theorem 34. *There exists a binary predicate* $V : \mathbb{N}^2 \to \mathcal{P}(\Pi)$ *such that :*
$\lambda f(\chi)(\mathsf{Y})\lambda x\lambda n(\mathsf{cc})\lambda k\, f\tau_0\tau_1 \Vdash \forall X\{\forall y(V(x,y) \leftrightarrow Xy) \to F[x,X]\} \to \forall X\, F[x,X]$
with $\tau_0 = \lambda v\, vxnk,\ \ \tau_1 = \lambda u\lambda x'\lambda n'\lambda k'\, \mathit{Comp}\, nn'\alpha\alpha' u,\ \ \alpha = (k)(x')n$ *and* $\alpha' = (k')(x)n'$.

Let us look at the behaviour of the term $\gamma = \lambda f(\chi)(\mathsf{Y})\lambda x\lambda n(\mathsf{cc})\lambda k\, f\tau_0\tau_1$. Consider a process $\gamma \star f.\pi$ in which γ is in head position. We have :
$\gamma \star f.\pi \succ \chi \star \mathsf{Y}\xi.\pi$ (where $\xi = \lambda x\lambda n(\mathsf{cc})\lambda k\, f\tau_0\tau_1$ depends only on f)
$\succ \mathsf{Y}\xi \star n_\pi.\pi \succ \xi \star \eta.n_\pi.\pi$ (with $\eta = \mathsf{Y}\xi$). Therefore

$$\gamma \star f.\pi \succ f \star \tau_0^{f\pi}.\tau_1^{f\pi}.\pi$$

with $\tau_i^{f\pi} = \tau_i[\eta/x, n_\pi/n, k_\pi/k]$.
Now $\tau_0^{f\pi}$ is simply the triple $<\eta, n_\pi, k_\pi>$. In other words $\tau_0^{f\pi}$ stores the current state $f.\pi$ at the present execution of γ.
$\tau_1^{f\pi}$ performs the real job : it looks at two such states $f.\pi$ and $f'.\pi'$ and compare their indexes n and n'. If $n = n'$ it does nothing.
If $n < n'$ (resp. $n' < n$) it restarts with $\gamma \star f'.\pi$ (resp. $\gamma \star f.\pi'$) :
the second file with the first stack.
Thus, the main function of this program is to update files (if χ is a clock)
or to choose a good version of a file (if χ is a signature).

Non standard integers

We give here sufficient conditions in order that every generic model contains non standard individuals. Let us define a unary predicate G by putting $\|Gn\| = \{\pi_n\}$ ($n \mapsto \pi_n$ is here a one-to-one recursive function from $\mathbb{N}$ onto Π).

Theorem 35.
i) We suppose that :
$(H_\perp)$ *For each stack constant* ϖ*, there exists a proof-like term* ξ *such that* $\xi \star \varpi \in \perp\!\!\!\perp$.
Then $I \Vdash \forall x\, Gx \to \perp$ *and* $(\forall n \in \mathbb{N})(\exists \xi \in \mathsf{PL})\, \xi \Vdash Gn$.
ii) $\chi \Vdash \forall x[Int(x) \to Gx] \to \perp$.

i) We have immediately $\|\forall x\, Gx\| = \{\pi_n;\, n \in \mathbb{N}\} = \Pi$ and therefore $I \Vdash \forall x\, Gx \to \perp$.
We are looking at some $\xi \in \mathsf{PL}$ such that $\xi \star \pi_n \in \perp\!\!\!\perp$. Now, we have $\pi_n = t_0 \ldots t_k.\varpi$, ϖ being a stack constant. By $(H_\perp)$, we have some $\eta \in \mathsf{PL}$ such that $\eta \star \varpi \in \perp\!\!\!\perp$. Il is sufficient to set $\xi = \lambda x_0 \ldots \lambda x_k\, \eta$.
ii) Let $t \Vdash \forall x[Int(x) \to Gx]$ and $\pi \in \Pi$, $\pi = \pi_n$. We must show that $\chi \star t.\pi \in \perp\!\!\!\perp$ that is $t \star \underline{n}.\pi \in \perp\!\!\!\perp$, by the reduction rule of χ. It is immediate, by hypothesis on t, since $\underline{n} \Vdash Int(n)$ and $\pi \in \|Gn\|$.
Q.E.D.

Thus, with hypothesis $(H_\perp)$, a generic model satisfies Gn for each standard integer n and also $\exists x\, \neg Gx$. It contains therefore *necessarily* a non standard individual. If, moreover, the instruction χ is present, it satisfies $\exists x\{Int(x), \neg Gx\}$. Then, it contains *necessarily* non standard integers.

The standard generic model

Consider a fixed one-to-one recursive function $\xi \mapsto \pi_\xi$ from PL (the set of proof-like terms) onto the set of stack constants. Every set $\perp\!\!\!\perp$ such that $(\forall \xi \in \mathsf{PL})\, \xi \star \pi_\xi \notin \perp\!\!\!\perp$ is clearly *coherent*, which means that $(\forall \xi \in \mathsf{PL})\, \xi \Vdash \perp$.
In this section, we take for $\perp\!\!\!\perp$ the greatest saturated set of processes which has this property. In fact, it is simpler to consider *the complement $\perp\!\!\!\perp^c$ of* $\perp\!\!\!\perp$ in the set $\Lambda_c \star \Pi$ of processes. Therefore, we set :
$\perp\!\!\!\perp^c =$ the least subset U of $\Lambda_c \star \Pi$ such that $(\forall \xi \in \mathsf{PL})\, \xi \star \pi_\xi \in U$ and
$(\forall \mathsf{p} \in U)(\forall \mathsf{p}' \in \Lambda_c \star \Pi)(\mathsf{p} \succ \mathsf{p}' \Rightarrow \mathsf{p}' \in U)$;
For each $\xi \in \mathsf{PL}$, the set of processes which are obtained by reduction of $\xi \star \pi_\xi$ will be called the *thread generated by* ξ ; $\perp\!\!\!\perp^c$ is thus the union of all threads. We have :

Lemma 36. *A process is in $\perp\!\!\!\perp$ if and only if it does not appear in any thread.*
A term $t \in \Lambda_c$ realizes $\perp$ if and only if it does not come in head position in any thread.

Trivial.
Q.E.D.

Remember that $\mathcal{T}_{\perp\!\!\!\perp}$ is the set of closed formulas with parameters which are realized by some proof-like term. The models of this theory (which is coherent if and only if $\perp\!\!\!\perp$ is) are called *generic models* . With this particular choice of $\perp\!\!\!\perp$, we shall call them *standard generic models.*
We now show some properties of these models, with the following assumptions :
1. There is no instruction which changes the current stack constant. In other words, in the thread generated by $\xi \in \mathsf{PL}$, the only stack constant which appears is π_ξ.
2. Every instruction is deterministic, that is to say that each process has at most one successor.
We then say that execution is *sequential.*

Theorem 37. *Let $a_0 = \delta\delta 0$ and $a_1 = \delta\delta 1$, with $\delta = \lambda x\, xx$. Then :*
$\lambda x(\mathsf{cc})\lambda k((x)(k)a_0)(k)a_1 \Vdash \forall x(x \neq 1, x \neq 0 \to x^2 \neq x) \to \perp$.

There are three possible values for $\|n \neq 1, n \neq 0 \to n^2 \neq n\|$:
if $n = 0$, it is $\|\top, \perp \to \perp\|$; if $n = 1$, it is $\|\perp, \top \to \perp\|$; if $n \neq 0, 1$, it is $\|\top\| = \emptyset$.
It follows that $|\forall x(x \neq 1, x \neq 0 \to x^2 \neq x)| = |\top, \perp \to \perp| \cap |\perp, \top \to \perp|$.
Let $t \in |\top, \perp \to \perp| \cap |\perp, \top \to \perp|$ and $\pi \in \Pi$. We must show that :
$\lambda x(\mathsf{cc})\lambda k((x)(k)a_0)(k)a_1 \star t.\pi \in \perp\!\!\!\perp$ that is $t \star \mathsf{k}_\pi a_0.\mathsf{k}_\pi a_1.\pi \in \perp\!\!\!\perp$. If it is not true, by hypothesis on t, we have $\mathsf{k}_\pi a_0, \mathsf{k}_\pi a_1 \not\Vdash \perp$. Therefore, both terms appear in head position in some thread ; since they both contain the stack constant at the bottom of π, these thread are the same one (by hypothesis 1 above). But then a_0 and a_1 appear in head position in the same thread. Now, by hypothesis 2, one of them appears in head position before the other, for instance a_0. But it is clear that a_1 can never come in head position after that.
Q.E.D.

The formula $\exists x \exists y\{x^2 = x, x \neq 0, x \neq 1\}$ is therefore satisfied in the generic model. Thus, there are individuals which are not integers and the axiom of recurrence is not satisfied. We shall see later that there are also non standard integers.
Let $\mathcal{B}$ be the set of individuals x such that $x^2 = x$. Since all equational formulas which are true in $\mathbb{N}$ are also true in generic models, it is clear that $\mathcal{B}$ is a Boolean algebra, which is therefore $\neq \{0, 1\}$. We now show that $\mathcal{B}$ is, indeed, infinite and even *atomless.*

Theorem 38. *Let* $\Theta = \lambda x \lambda y \mathsf{cc} \lambda k((x)(k)y0)((x)(k)y1)(k)y2$. *Then we have :*
$\Theta \Vdash \forall x[\forall y(xy \neq 0, xy \neq x \to y^2 \neq y), x \neq 0 \to x^2 \neq x]$
(This formula says that the Boolean algebra $\mathcal{B}$ *is atomless).*

By a simple computation, we see that we must show the following two properties :
i) $\Theta \Vdash (\bot, \bot \to \bot), \bot \to \bot$.
ii) $\Theta \Vdash |\top, \bot \to \bot| \cap |\bot, \top \to \bot|, \top \to \bot$.
Proof of (i) : let $t \in |\bot, \bot \to \bot|$ and $u \in |\bot|$. We have to show that $\Theta \star t.u.\pi \in \perp\!\!\!\perp$ that is, by reduction, $t \star \mathsf{k}_\pi u0.((t)(\mathsf{k}_\pi)u1)(\mathsf{k}_\pi)u2.\pi \in \perp\!\!\!\perp$. But, from $u \Vdash \bot$, we easily deduce that $\mathsf{k}_\pi u\xi \Vdash \bot$ for every $\xi \in \Lambda_c$. Since $t \Vdash \bot, \bot \to \bot$, it follows that $((t)(\mathsf{k}_\pi)u1)(\mathsf{k}_\pi)u2 \Vdash \bot$ and therefore $t \star \mathsf{k}_\pi u0.((t)(\mathsf{k}_\pi)u1)(\mathsf{k}_\pi)u2.\pi \in \perp\!\!\!\perp$.
Proof of (ii) : let $t \in |\top, \bot \to \bot| \cap |\bot, \top \to \bot|$ and $u \in \Lambda_c$. Again, we have to show that $t \star \mathsf{k}_\pi u0.((t)(\mathsf{k}_\pi)u1)(\mathsf{k}_\pi)u2.\pi \in \perp\!\!\!\perp$. If it is not true, we have :
$\mathsf{k}_\pi u0 \not\Vdash \bot$ (because $t \Vdash \bot, \top \to \bot$) and $((t)(\mathsf{k}_\pi)u1)(\mathsf{k}_\pi)u2 \not\Vdash \bot$ (because $t \Vdash \top, \bot \to \bot$).
But, since $t \Vdash \bot, \top \to \bot$ (resp. $\top, \bot \to \bot$), we deduce $\mathsf{k}_\pi u1 \not\Vdash \bot$ (resp. $\mathsf{k}_\pi u2 \not\Vdash \bot$).
It follows that $\mathsf{k}_\pi u0, \mathsf{k}_\pi u1, \mathsf{k}_\pi u2$ appear all in head position in some thread. Since they contain k_π, these threads are the same (their stack constant is the one of π). Suppose, for example, that $\mathsf{k}_\pi u0$ appears first in head position, then $\mathsf{k}_\pi u1$, and then $\mathsf{k}_\pi u2$. We have thus :
$\mathsf{k}_\pi u0 \star \pi_0 \succ u \star \pi \succ \cdots \succ \mathsf{k}_\pi u1 \star \pi_1 \succ u \star \pi \succ \cdots \succ \mathsf{k}_\pi u2 \star \pi_2 \succ u \star \pi \succ \cdots$
But such an execution is clearly impossible because, at the second appearance of the process $u \star \pi$, we enter in a loop and can never arrive at $\mathsf{k}_\pi u2 \star \pi_2$.
Q.E.D.

The generic thread

Consider a fixed one-to-one recursive function $n \mapsto \xi_n$ from $\mathbb{N}$ onto PL. We define a unary predicate P by taking for $\|Pn\|$ the set of stacks the bottom constant of which is π_{ξ_n}.

Theorem 39. *In the generic model,* $\neg P$ *contains exactly one individual and it is not a standard integer. In other words, the following formulas are realized :*
i) Pn *for each* $n \in \mathbb{N}$ *;*
ii) $\forall x\, Px \to \bot$ *;*
iii) $\forall x \forall y[\neg Px, x \neq y \to Py]$.

i) Let $n \in \mathbb{N}$ be such that $\delta\delta 0 \not\Vdash Pn$ and $\delta\delta 1 \not\Vdash Pn$. Thus, there exist $\rho_0, \rho_1 \in \Pi$ which have the same stack constant π_{ξ_n}, such that $\delta\delta 0 \star \rho_0 \notin \perp\!\!\!\perp$ and $\delta\delta 1 \star \rho_1 \notin \perp\!\!\!\perp$. We have therefore $\xi_n \star \pi_{\xi_n} \succ \delta\delta 0 \star \rho_0$ and $\xi_n \star \pi_{\xi_n} \succ \delta\delta 1 \star \rho_1$, which is clearly impossible.
ii) We have $\|\forall x Px\| = \bigcup_{n \in \mathbb{N}} \|Pn\| = \Pi$. Therefore $I \Vdash \forall x\, Px \to \bot$.
iii) We show, by means of theorem 9, that $V\lambda x \lambda y\, y \Vdash \forall x \forall y[\neg Px, x \neq y \to Py]$ that is to say $V\lambda x \lambda y\, y \Vdash \neg Pm, m \neq n \to Pn$ for $m, n \in \mathbb{N}$. So, let $\pi \in \|Pm\|$, $t \Vdash m \neq n$ and $\pi' \in \|Pn\|$; we must show that $\lambda x \lambda y\, y \star \mathsf{k}_\pi.t.\pi' \in \perp\!\!\!\perp$. It is obvious if $m \neq n$, since then the stacks π and π', which have different stack constants, cannot appear both in the same thread. If $m = n$, we have $t \Vdash \bot$, which give the desired result, because $\lambda x \lambda y\, y \star \mathsf{k}_\pi.t.\pi' \succ t \star \pi'$.
Q.E.D.

We now add to λ_c-calculus an instruction σ with the following reduction rule :

$$\sigma \star t.\pi \succ t \star \underline{n}.\pi$$

where n is the number of the stack constant of π (this constant is therefore π_{ξ_n}).

Theorem 40. $\sigma \Vdash \forall x[Int(x) \to Px] \to \bot$.

In the generic model, the unique individual which satisfies $\neg Px$ is therefore an integer. By theorem 39, it is a *non standard integer*.

Proof. Let $t \Vdash \forall x[Int(x) \to Px]$ and $\pi \in \Pi$. We must show that $\sigma \star t.\pi \in \perp\!\!\!\perp$; it is sufficient to prove that $t \star \underline{n}.\pi \in \perp\!\!\!\perp$, n being the number of the stack constant of π. Therefore, we have $\pi \in \|Pn\|$ and $t \Vdash Int(n) \to Pn$. Hence the result, since $\underline{n} \Vdash Int(n)$.

Q.E.D.

We can therefore add to our second order language a symbol of constant g (for *generic*), with the following axioms :

$Int(\mathsf{g})$, $\forall x[x \neq \mathsf{g} \leftrightarrow Px]$, $\mathsf{g} \neq n$ for each integer n.

The predicate Px will be denoted by $x \neq \mathsf{g}$; thus $\|n \neq \mathsf{g}\|$ is, for each integer n, the set of stacks which have π_{ξ_n} as their stack constant.

It is interesting to study the properties of the (non standard) proof-like λ_c-term ξ_{g} and of the thread $\xi_{\mathsf{g}} \star \pi_{\xi_{\mathsf{g}}}$, which will be, rather naturally, called the *generic thread*.

Remark. With the help of the instruction σ, we can therefore prove the existence of a non standard integer. By means of the second fixed point theorem of λ-calculus, we can do without this instruction, but we have a somewhat weaker result :

Theorem 41. $(\forall \theta \in \mathsf{PL})\, \theta \not\Vdash \forall x[Int(x) \to Px]$.

Let $\Delta : \mathbb{N} \to \mathbb{N}$ be the recursive function defined by $\Delta m = n \Leftrightarrow \xi_n = \xi_m \underline{m}$ and let δ be a closed λ-term which represents this function. We define the integers p, q by setting $\xi_p = \lambda x(\theta)(\delta)x$ and $q = \Delta p$. We have therefore $\xi_q \star \pi_{\xi_q} = \xi_p \underline{p} \star \pi_{\xi_q} \succ \theta \star \delta \underline{p}.\pi_{\xi_q}$, which shows that $\theta \star \delta \underline{p}.\pi_{\xi_q} \notin \perp\!\!\!\perp$.

Now, we have $\pi_{\xi_q} \in \|Pq\|$ and, by theorem 15, $\delta \underline{p} \Vdash Int(q)$. It follows that $\theta \not\Vdash Int(q) \to Pq$.

Q.E.D.

The clock

With the present definition of $\perp\!\!\!\perp$, in order to add a new instruction ι, it is sufficient to give its reduction rule only when it comes in head position *in a thread*. Indeed, $\perp\!\!\!\perp$ is the complement of the union of all threads and the only condition it has to satisfy is to be cc-*saturated* (see the section " Truth values and types "). It is therefore sufficient to define each thread, i.e. the sequence of processes obtained by the execution of $\xi \star \pi_\xi$, for each proof-like term ξ (possibly containing the instruction ι).

It is in this way that we define *the clock instruction* $\hbar$. Its reduction rule is

$\hbar \star t.\pi \succ t \star \underline{n}.\pi$ where $n \in \mathbb{N}$ is obtained in the following way : let $\xi \star \pi_\xi$ be the (unique) thread in which the process $\hbar \star t.\pi$ appears. Then n is the number of reduction steps from $\xi \star \pi_\xi$ (" the boot ") to $\hbar \star t.\pi$ (if $\hbar \star t.\pi$ appears more than once in the thread $\xi \star \pi_\xi$, we consider *its first appearance* ; notice that, in this case, the execution ends in a loop).

Definition. A proof-like term θ will be called *strongly solvable* iff $\theta \Vdash \bot \to \bot$. In other words : for every $t \in \Lambda_c \setminus \mathsf{PL}$, if $\theta \star t$ comes in head position in a thread, then t also comes in head position in this thread.

When θ is a closed usual λ-term, this means that it has a head normal form which is $\lambda x\, x t_1 \ldots t_n$ (hence the terminology " strongly solvable ").

The following theorem shows that *every standard strongly solvable proof-like term comes in head position in the generic thread.*

Theorem 42. *Let θ be a strongly solvable proof-like term and $\phi_\theta : \mathbb{N}^2 \to \{0,1\}$ the recursive function such that : $\phi_\theta(n,p) = 1 \Leftrightarrow \theta$ comes in head position in the thread $\xi_p \star \pi_{\xi_p}$ at the $(n+4)$-th step. Then $\hbar\lambda x\lambda y(\theta)(y)x \Vdash \forall p\{\forall n[Int(n) \to \phi_\theta(n,p) \neq 1] \to p \neq \mathsf{g}\}$*
(in other words, $\exists n\{Int(n), \phi_\theta(n,\mathsf{g}) = 1\}$).

Let $p \in \mathbb{N}$, $\pi \in \|p \neq \mathsf{g}\|$ and $t \Vdash \forall n[Int(n) \to \phi_\theta(n,p) \neq 1]$. We prove the result by contradiction : suppose that $\hbar\lambda x\lambda y(\theta)(y)x \star t.\pi \notin \perp\!\!\!\perp$. Therefore, this process appears in a thread, which is certainly $\xi_p \star \pi_{\xi_p}$ since $\pi \in \|p \neq \mathsf{g}\|$. We have thus :
$\xi_p \star \pi_{\xi_p} \succ \hbar\lambda x\lambda y(\theta)(y)x \star t.\pi \succ \theta \star t\underline{n}.\pi$, where the integer n, which is provided by $\hbar$, is the number of steps in the reduction of $\xi_p \star \pi_{\xi_p}$ until $\hbar\lambda x\lambda y(\theta)(y)x \star t.\pi$. Thus, we obtain $\theta \star t\underline{n}.\pi$ at the $(n+4)$-th step of reduction. Since θ is in head position at this moment, we have $\phi_\theta(n,p) = 1$. By hypothesis on t, it follows that $t\underline{n} \Vdash \perp$. Now, by hypothesis, $\theta \Vdash \perp \to \perp$ and therefore $\theta \star t\underline{n}.\pi \in \perp\!\!\!\perp$. It is a contradiction, because $\theta \star t\underline{n}.\pi$ appears in the thread $\xi_p \star \pi_{\xi_p}$.
Q.E.D.

Corollary 43. *Let θ be a proof-like term such that $\theta\hat{m}$ is strongly solvable for each integer m (with $\hat{m} = s^m 0$) and let $\psi_\theta : \mathbb{N}^3 \to \{0,1\}$ be the recursive function defined by $\psi_\theta(m,n,p) = 1$ iff $\theta\hat{m}$ comes in head position at the $(n+4)$-th step in the thread $\xi_p \star \pi_{\xi_p}$. Then we have :*
$T\lambda m\, \hbar\lambda x\lambda y(\theta m)(y)x \Vdash \forall p\forall m\{Int(m), \forall n[Int(n) \to \psi_\theta(m,n,p) \neq 1] \to p \neq \mathsf{g}\}$
(in other words $\forall m(Int(m) \to \exists n\{Int(n), \psi_\theta(m,n,\mathsf{g}) = 1\})$).

It means that the following formula is realized : " for every integer m, $\theta\hat{m}$ comes in head position in the generic thread $\xi_{\mathsf{g}} \star \pi_{\xi_{\mathsf{g}}}$ ". Therefore also the formula : " the generic thread neither stops nor loops " (take, for instance, $\theta = \lambda x\lambda y\, y$).

Proof. By theorem 10, it is sufficient to prove that, for every integers m, p and every stack $\rho \in \|\forall n[Int(n) \to \psi_\theta(m,n,p) \neq 1] \to p \neq \mathsf{g}\|$, we have :
$\lambda m\, \hbar\lambda x\lambda y(\theta m)(y)x \star \hat{m}.\rho \in \perp\!\!\!\perp$, that is $\hbar\lambda x\lambda y(\theta\hat{m})(y)x \star \rho \in \perp\!\!\!\perp$. But this results from theorem 42 and from the fact that $\psi_\theta(m,n,p) = \phi_{\theta\hat{m}}(n,p)$.
Q.E.D.

The denumerable axiom of choice

Here we check that, by means of the clock instruction $\hbar$, we can realize the axiom of denumerable choice.

Theorem 44. *Let $F[x,X]$ be a formula with parameters, X being a unary predicate variable. There exists $\Phi : \mathbb{N}^4 \to \mathcal{P}(\Pi)$ such that :*
$\hbar \Vdash \forall x\forall p\forall X\{\forall n(Int[n], F[x,\Phi(n,p,x,y)/Xy] \to \perp), F[x,X] \to p \neq \mathsf{g}\}$.

We define $v : \mathbb{N}^2 \to \Lambda_c$ by putting : $v(n,p) =$ the λ_c-term u which is in second position in the stack, at the n-th execution step in the thread $\xi_p \star \pi_{\xi_p}$. At the n-th step of this execution, we have therefore a process of the form $\tau \star t.u.\pi$.
We define now $\Phi(n,p,x,y)$, by means of the denumerable axiom of choice, in such a way that :
If there exists $X : \mathbb{N} \to \mathcal{P}(\Pi)$ such that $v(n,p) \Vdash F[x,X]$
then $v(n,p) \Vdash F[x,\Phi(n,p,x,y)/Xy]$.
We show that Φ has the desired property : consider $x, p \in \mathbb{N}$, an application $X : \mathbb{N} \to \mathcal{P}(\Pi)$, λ_c-terms t, u such that $t \Vdash \forall n(Int[n], F[x,\Phi(n,p,x,y)/Xy] \to \perp)$, $u \Vdash F[x,X]$ and a stack

$\pi \in \|p \neq \mathsf{g}\|$. Then, we have to show that $\hbar \star t.u.\pi \in \perp\!\!\!\perp$; suppose, on the contrary, that $\hbar \star t.u.\pi$ appears in a thread, at the n-th step. By hypothesis on π, this thread is $\xi_p \star \pi_{\xi_p}$; we have therefore $u = v(n,p)$, by definition of v, hence $v(n,p) \Vdash F[x,X]$. By definition of Φ, we have then $u = v(n,p) \Vdash F[x, \Phi(n,p,x,y)/Xy]$. But, since $\underline{n} \Vdash Int[n]$, it follows that $t \star \underline{n}.u.\pi \in \perp\!\!\!\perp$, by hypothesis on t. It is a contradiction, because this process appears at the $(n+1)$-th step in the thread $\xi_p \star \pi_{\xi_p}$.

Q.E.D.

It follows that the generic model satisfies the formula :
$\forall x \forall X (F[x,X] \to \exists n \{Int[n], F[x, \Phi(n, \mathsf{g}, x, y)/Xy]\})$. We can then define the binary predicate $\Psi(x,y)$ by the formula :
" $\Phi(n, \mathsf{g}, x, y)$ for the first integer n such that $F[x, \Phi(n, \mathsf{g}, x, y)/Xy]$, or $\perp$ if there is no such integer ". Then, we have, in the generic model :
$\forall x \forall X (F[x,X] \to F[x, \Psi(x,y)/Xy])$, which gives the denumerable axiom of choice.
By the same methods as above, we can also easily get the axiom of dependent choice.

References

[1] T. Coquand. *A semantics of evidence for classical arithmetic.*
J. Symb. Log. 60, p. 325-337, 1995.
[2] T. Griffin. *A formulæ-as-type notion of control.*
Conf. Record of the 17th A.C.M. Symp. on Principles of Programming Languages, 1990.
[3] J.-L. Krivine. *A general storage theorem for integers in call-by-name λ-calculus.*
Th. Comp. Sc. 129, p. 79-94, 1994.
[4] J.-L. Krivine. *Typed lambda-calculus in classical Zermelo-Fraenkel set theory.*
Arch. Math. Log. 40, 3, p. 189-205, 2001.
[5] J.-L. Krivine. *Dependent choice, 'quote' and the clock.*
Th. Comp. Sc. 308, p. 259-276, 2003.

My papers are at `http://www.pps.jussieu.fr/~krivine/`