ELSEVIER

Annals of Pure and Applied Logic 68 (1994) 53–78

ANNALS OF PURE AND APPLIED LOGIC

Classical logic, storage operators and second-order lambda-calculus

Jean-Louis Krivine*

Equipe de Logique, Université Paris VII, C.N.R.S., 2 Place Jussieu, 75251 Paris Cedex 05, France

Communicated by J.-Y. Girard; received 14 September 1992; revised 20 July 1993

Abstract

We describe here a simple method in order to obtain programs from proofs in second-order classical logic. Then we extend to classical logic the results about storage operators (typed λ-terms which simulate call-by-value in call-by-name) proved by Krivine (1990) for intuitionistic logic. This work generalizes previous results of Parigot (1992).

1. Introduction

In this paper, we consider the λ-calculus with head reduction (the call-by-name λ-calculus in [15]), equipped with a second-order type system: the system $\mathscr{F}$ of Girard [3], or a simple extension of it, which is nothing else than the full predicate calculus of second order [7, 9, 10].

We shall add a new constant, denoted by **c**, to the pure (i.e. untyped) λ-calculus, with two aims:

(1) To be able to modelize important features of imperative programming languages: the "exit" instruction, most often used in order to terminate a program from inside a procedure; and also "escape" instructions which are useful in order to handle errors and exceptions. Since we consider the call-by-name λ-calculus, each λ-term is considered as a program, whose execution is the head reduction of the term. It follows that the operational behaviour of the new operator **c** has to be defined only when it is in head position in the λ-term. This definition is given at the beginning of Section 2.3. It is a particular case of a rule of reduction for control operators given by Felleisen [2].

(2) In our type system, the constant **c** will be declared to have as its type the second-order formula $\forall X((\neg\neg X \to X)$, which axiomatizes classical logic over

* E-mail krivine@logique.jussieu.fr

SSDI 0168-0072(94)E0059-W

intuitionistic logic. In this way, our system extends to classical logic the usual "Curry–Howard correspondence" between intuitionistic proofs and programs. The type of the "exit" instruction itself will be the "absurdity rule" $\forall X(\bot \to X)$ which is an instance of this formula (because $\neg\neg X$ is $(X \to \bot) \to \bot$). The type of "escape" instructions is given by the formula $\forall X \forall Y[((X \to Y) \to X) \to Y]$ ("law of Pierce") which is valid in classical logic.

We talked previously about imperative programming languages, which may seem surprising, since λ-calculus is usually considered as a prototype for functional languages. But we believe that the results of this paper, together with those in [8] about storage operators, are good supports for the following claim: the second-order λ-calculus, together with the strategy of head reduction (call-by-name), is a model for *imperative* programming languages. The storage operators described in [8] can be used to modelize assignment instructions (like $y = \mathrm{fun}(x)$, where fun is a function, in the C programming language). In a forthcoming paper, we intend to treat the case of "input" instructions (like "scanf" in the C programming language), and of "while" loops.

The main result of the present paper is Theorem 4.4, which tells essentially that storage operators behave with "classical integers" in exactly the same way as they do with "intuitionistic integers". Let us explain this briefly.

For each data type, we can define a notion of storage operator. Let us take integers as an example of data type. Then, a closed λ-term T is called a *storage operator for integers* if, for each Church integer n, there is a λ-term $\alpha_n \simeq_\beta n$ such that, for every $\theta \simeq_\beta n$, the λ-term $Tf\theta$ (f is a fixed variable, or, in fact, any λ-term) reduces, *by head reduction*, to $f\alpha'_n$, where α'_n is obtained from α_n by means of some substitution.

For example, $T = \lambda f \lambda n((n)\lambda g.g \circ s)f\,0$, where s is any λ-term for the successor, is a storage operator for integers: indeed, it is easily checked that, if $\theta \simeq_\beta n$, then $Tf\theta$ reduces, by head reduction, to $(f)s^n 0$.

The interest of such operators lies in two facts:

(1) In λ-calculus with head reduction (call-by-name λ-calculus), they simulate call-by-value for integers. This follows from their very definition: if f is any λ-term, and if we have to compute $f\theta$, let us compute $Tf\theta$ (by head reduction) instead. This will give first $f\alpha'_n$, and then we will reduce $f\alpha'_n$, which is clearly the same computation as to reduce $f\alpha_n$. This means that we computed θ first (in the form α_n), and then that we applied the "function" f to the result of this computation. Thus, the integer θ has been called by value.

(2) We can find simple second-order types for some of these operators. In fact, let $\mathrm{Int}[x]$ be the type of integers ($\mathrm{Int}[x]$ is the formula $\forall X\{\forall y(Xy \to Xsy), X0 \to Xx\}$, which says that x belongs to any set containing 0 and closed by successor). Then, it is proved in [8] that any λ-term T, which is of type $\forall x\{\neg \mathrm{Int}[x] \to \neg \mathrm{Int}^*[x]\}$, is a storage operator for integers. The operation * is the Gödel translation, which associates, to every formula F, the formula F^* obtained by negating each atomic formula of F. It has the well-known property of turning every classical proof into an intuitionistic one.

Now, let us say that a λ-term θ is an *intuitionistic integer*, if $\vdash \theta$: Int$[n]$ for some n (i.e. if θ is obtained by an intuitionistic proof of Int$[n]$), and that θ is a *classical integer*, if $\mathbf{c}$: $\forall X(\neg\neg X \to X) \vdash \theta$: Int$[n]$ (i.e. if θ is obtained by means of a classical proof of Int$[n]$, which we shall denote by $\vdash^{c} \theta$: Int$[n]$). Then, Theorem 4.4 asserts the following: let T be of type $\forall x\{\neg \text{Int}[x] \to \neg \text{Int}^*[x]\}$. Then, even if θ is a classical integer, $Tf\theta$ will still reduce, by head reduction, to $f\alpha'_n$, i.e. to the same result as for an intuitionistic integer.

The intuitive meaning of such a result can be explained as follows: let us consider that the atomic formula $\perp$ is the type of executable programs. Suppose you write a program θ in order to compute some integer, say the 100 000th prime number, for example. This program is of type Int, not of type $\perp$, and, as such, is not executable alone. Indeed, some operating system must take care of it, in order, first, to launch it, then to supervise its execution (hardware or software errors may occur), and finally to display its result in some form, or to pass it to another program, and so on. Let us represent this operating system by E. Then, the executable program is $E\theta$, which is of type $\perp$, so that E has, naturally, the type $\neg$ Int. A program like E is usually called a "continuation".

Now, an essential feature of E is the fact that it must call the program θ *by value*: in fact, it is clear that during the execution of $E\theta$, we want that θ be computed first, i.e. that the operating system begins by dealing with θ, not by carrying out its own internal procedures, which may be very long and numerous. But, by Theorem 4.4, there is a way to ensure this, and it is to use storage operators: if we know that E is of the form Tf, where T is of type $\forall x\{\neg \text{Int}[x] \to \neg \text{Int}^*[x]\}$, then the continuation E will behave as we want. This explains the interest of considering the head reduction of terms like $Tf\theta$.

The extraction of programs from classical proofs, which has long been considered as impossible, has raised a lot of interest since two or three years. The fact that control operators can be given types like $\neg\neg F \to F$ was discovered by Griffin [6], and exploited by Murthy [11] via translations of classical into intuitionistic logic. Girard [4, 5] has recently obtained a new universal system which embodies, as fragments, linear logic, intuitionistic logic and classical logic.

The deduction system devised by Parigot for classical logic in [12, 13] allows extraction of programs in an extension of λ-calculus with two abstractors, called $\lambda\mu$-calculus. This is close to the method described in the present paper because head $\mathbf{c}$-reduction is equivalent to head reduction of $\lambda\mu$-calculus. The idea of using storage operators in order to "decode" classical integers is due to him [13]. In fact, he has proved Theorem 4.3 [14], in the frame of $\lambda\mu$-calculus, and the particular case of Theorem 4.4 when $T = \lambda f \lambda n((n)\lambda g.g \circ s)f0$. It should be noted that it is this result which ensures the correctness of programs obtained from classical proofs.

2. Preliminaries on λ-calculus and second-order type systems

2.1. Pure λ-calculus

We shall denote by Λ the set of terms of pure (i.e. untyped) λ-calculus, also called *λ-terms*. A variable of the λ-calculus will be called a *λ-variable*; by convention, a λ-variable which is never used under a λ will be called a *λ-constant*. Given $t, u \in \Lambda$, the result of the application of t to u will be denoted by $(t)u$ or simply tu. In the same way, we shall write $(t)uv$ or tuv instead of $((t)u)v$, etc.

The β (resp. $\beta\eta$)-equivalence between two λ-terms $t, u \in \Lambda$ will be denoted by $t \simeq_\beta u$ (resp. $t \simeq_{\beta\eta} u$).

If $n \in \mathbb{N}$ the notation $(t)^n u$ will mean $(t) \ldots (t)u$ (t being repeated n times). We shall denote the booleans $\lambda x\lambda y.y$ and $\lambda x\lambda y.x$ by $\mathbf{0}$ and $\mathbf{1}$.

The notation $t[u_1/x_1, \ldots, u_n/x_n]$ will represent the result of the simultaneous substitution of u_i to the free occurrences of x_i in t (after a suitable renaming of the bounded variables of t). The notation for the successive substitution of u_1 to $x_1, \ldots, u_n$ to x_n will be $t[u_1/x_1] \ldots [u_n/x_n]$. Let us recall that a λ-term t either has a *head redex* (i.e. $t = \lambda x_1 \ldots \lambda x_k(\lambda x.v)uu_1 \ldots u_n$, the head redex being $(\lambda x.v)u$), or is in *head normal form* (*h.n.f.* in short), i.e. $t = \lambda x_1 \ldots \lambda x_k(x)u_1 \ldots u_n$. A *solvable* term is a λ-term which is β-equivalent to a h.n.f. A *head reduction* is a β-reduction in which we only contract head redexes. It is well known that, if t is a solvable term, the head reduction of t results in a h.n.f. called the principal head normal form of t. We shall mainly use a restricted form of head reduction, which stops as soon as one obtains a term beginning by a λ. Let us say that t' is obtained from t by one step of *restricted head reduction* if $t = (\lambda x.u)vt_1 \ldots t_n$ and $t' = u[v/x]t_1 \ldots t_n$. The notation $t \succ t'$ will mean that t' is obtained from t by a finite number of steps of restricted head reduction.

The following lemma is immediate.

Lemma 2.1. (i) *If $t \succ t'$, then $tu_1 \ldots u_k \succ t'u_1 \ldots u_k$.*

(ii) *If $t \succ t'$, then $t[u_1/x_1 \ldots, u_k/x_k] \succ t'[u_1/x_1, \ldots, u_k/x_k]$.*

A *substitution* S is a map from the set of λ-variables into Λ. It has a unique natural extension into a map (which we shall also denote by S) from Λ into itself, such that $S(tu) = S(t)S(u)$ and $S(\lambda x.t) = \lambda y.S_y(t[y/x])$ for any variable y except a finite number (where S_y is the substitution defined by $S_y(y) = y$; $S_y(z) = S(z)$ for any variable $z \neq y$). The second part of the preceding lemma can be rewritten as follows.

Lemma 2.2. *If $t \succ t'$ then $S(t) \succ S(t')$ for any substitution S.*

2.2. Second-order type systems

The types will be formulas of the (usual) second-order predicate calculus over a given language $\mathscr{L}$. The logical connectives used are only $\bot$, $\to$, $\forall$. We shall use the notation $A_1, A_2, \ldots, A_k \to B$ for $A_1 \to (A_2 \to (\cdots \to (A_k \to B) \ldots))$.

There are individual (or first-order) variables, denoted by $x, y, \ldots$, and predicate (or second-order) variables, denoted by $X, Y, \ldots$ There may be symbols of function, but no variables of function. As usual, each predicate variable or constant and each symbol of function is associated with an integer called its arity. Predicate variables (resp. constants) of arity 0 are also called propositional variables (resp. constants), and function symbols of arity 0 are called individual constants. The terms of the language $\mathscr{L}$ (also called $\mathscr{L}$-terms, to avoid confusion with λ-terms) are built in the usual way, using individual variables (also called $\mathscr{L}$-variables for the same reason) and symbols of function.

For convenience, we shall suppose that there are infinitely many individual constants, and, for each $n \geqslant 0$, infinitely many predicate constants of arity n. This will allow us to consider only closed formulas in the deduction rules and the definition of realizability.

The logical connectives $\neg$, $\wedge$, $\vee$, $\exists$ are defined as follows: $\neg F$ is $F \to \bot$; $F \wedge G$ is $\forall X[(F, G \to X) \to X]$; $F \vee G$ is $\forall X[(F \to X), (G \to X) \to X]$; $\exists \xi F$ is $\forall X[\forall \xi(F \to X) \to X]$ (ξ is any variable of first or second order, X is a propositional variable).

We do not suppose that the language $\mathscr{L}$ has a special predicate constant for equality. Instead, we define the formula $t = u$ (where t, u are $\mathscr{L}$-terms) to be $\forall Y(Yt \to Yu)$, where Y is a 1-ary predicate variable. Such a formula will be called an *equation of* $\mathscr{L}$.

Let $\mathscr{E}$ be a (finite) set of equations of $\mathscr{L}$. An equation $t_0 = u_0$ is said to be an *equational consequence* of $\mathscr{E}$ if it can be obtained by the following rules, called rules of equational deduction from $\mathscr{E}$:

(i) If $t = u$ is an equation of $\mathscr{E}$, then $t[v_1/x_1, \ldots, v_k/x_k] = u[v_1/x_1, \ldots, v_k/x_k]$ is deduced ($v_1, \ldots, v_k$ are arbitrary $\mathscr{L}$-terms and $x_1, \ldots, x_k$ are $\mathscr{L}$-variables).

(ii) For every $\mathscr{L}$-terms t, u, v, $t = t$ is deduced; if $t = u$ is deduced, so is $u = t$; if $t = u$, $u = v$ are deduced, so is $t = v$.

(iii) If $u_1 = v_1, \ldots, u_k = v_k$ are deduced, so is $fu_1 \ldots u_k = fv_1 \ldots v_k$, for every k-ary function symbol f.

The following rules are the rules of "intuitionistic natural deduction" from the equations of $\mathscr{E}$. They operate on expressions of the form $\mathscr{A} \vdash_{\mathscr{E}} A$, where A is a closed formula, and $\mathscr{A}$ a finite set of closed formulas.

(D1) $\mathscr{A}, A \vdash_{\mathscr{E}} A$ for every closed formulas A.

(D2) If $\mathscr{A}, A \vdash_{\mathscr{E}} B$, then $\mathscr{A} \vdash_{\mathscr{E}} A \to B$ for all closed formulas A, B.

(D3) If $\mathscr{A} \vdash_{\mathscr{E}} A$ and $\mathscr{A} \vdash_{\mathscr{E}} A \to B$, then $\mathscr{A} \vdash_{\mathscr{E}} B$ for all closed formulas A, B.

(D4) If $\mathscr{A} \vdash_{\mathscr{E}} \forall x\, A$, then $\mathscr{A} \vdash_{\mathscr{E}} A[t/x]$ for every closed $\mathscr{L}$-term t; x is an individual variable, which is the only free variable of A.

(D5) If $\mathscr{A} \vdash_{\mathscr{E}} A[a/x]$, a being an individual constant which does not appear in $\mathscr{A}$, then $\mathscr{A} \vdash_{\mathscr{E}} \forall x\, A$; x is an individual variable, which is the only free variable of A.

(D6) If $\mathscr{A} \vdash_{\mathscr{E}} \forall X\, A$, X being an n-ary predicate variable, which is the only free variable of A, then $\mathscr{A} \vdash_{\mathscr{E}} A[F/Xx_1 \ldots x_n]$; F is any formula whose only free variables

are $x_1, \ldots, x_n$ and $A[F/Xx_1 \ldots x_n]$ is the formula obtained by replacing, in A, each atomic formula of the form $Xt_1 \ldots t_n$, by $F[t_1/x_1, \ldots, t_n/x_n]$.
This rule is the "comprehension scheme".

(D7) If $\mathscr{A} \vdash_{\mathscr{E}} A[P/X]$, P being an n-ary predicate constant which does not appear in $\mathscr{A}$, then $\mathscr{A} \vdash_{\mathscr{E}} \forall X\, A$; X is an n-ary predicate variable which is the only free variable of A.

(D8) If $t = u$ is an equational consequence of $\mathscr{E}$ and if $\mathscr{A} \vdash_{\mathscr{E}} A[t/x]$, then $\mathscr{A} \vdash_{\mathscr{E}} A[u/x]$; t, u are closed $\mathscr{L}$-terms; x is an individual variable, which is the only free variable of A.

The propositional constant $\perp$ plays no particular role with respect to these rules (they constitute the so-called "minimal logic").

We obtain the "classical natural deduction" by adding the axiom $\forall X(\neg\neg X \to X)$ (X is a propositional variable). So, the closed formula A is a consequence of $\mathscr{A}$ (a finite set of closed formulas) and $\mathscr{E}$ (a set of equations) in classical logic if and only if $\mathscr{A}, \forall X(\neg\neg X \to X) \vdash_{\mathscr{E}} A$ is obtained by means of the preceding deduction rules.

By the well known Curry–Howard isomorphism, the rules for intuitionistic logic are easily translated into rules of construction of typed terms. More precisely, let us define a *context* Γ to be an expression of the form $x_1: A_1, \ldots, x_k: A_k$ where $x_1, \ldots, x_k$ are distinct λ-variables, and $A_1, \ldots, A_k$ are closed formulas; a *typed term* is an expression of the form $\Gamma \vdash_{\mathscr{E}} \tau: A$, where Γ is a context, τ a λ-term, and A a closed formula (it is read as "τ is of type A in the context Γ, with respect to the equations of $\mathscr{E}$").

The following are the rules of construction of typed terms:

(T1) $\Gamma, x: A \vdash_{\mathscr{E}} x: A$ for every closed formula A.

(T2) If $\Gamma, x: A \vdash_{\mathscr{E}} \tau: B$, then $\Gamma \vdash_{\mathscr{E}} \lambda x.\tau: A \to B$ for all closed formulas A, B.

(T3) If $\Gamma \vdash_{\mathscr{E}} \tau: A$ and $\Gamma \vdash_{\mathscr{E}} \tau': A \to B$, then $\Gamma \vdash_{\mathscr{E}} \tau'\tau: B$ for all closed formulas A, B.

(T4) If $\Gamma \vdash_{\mathscr{E}} \tau: \forall x\, A$, then $\Gamma \vdash_{\mathscr{E}} \tau: A[t/x]$ for every closed $\mathscr{L}$-term t; x is an individual variable, which is the only free variable of A.

(T5) If $\Gamma \vdash_{\mathscr{E}} \tau: A[a/x]$, a being an individual constant which does not appear in Γ, then $\Gamma \vdash_{\mathscr{E}} \tau: \forall x A$; x is an individual variable, which is the only free variable of A.

(T6) If $\Gamma \vdash_{\mathscr{E}} \tau: \forall X\, A$, X being an n-ary predicate variable, which is the only free variable of A, and F a formula whose only free variables are $x_1, \ldots, x_n$, then $\Gamma \vdash_{\mathscr{E}} \tau: A[F/Xx_1 \ldots x_n]$.

(T7) If $\Gamma \vdash_{\mathscr{E}} \tau: A[P/X]$, P being an n-ary predicate constant which does not appear in Γ, then $\Gamma \vdash_{\mathscr{E}} \tau: \forall X\, A$; X is an n-ary predicate variable which is the only free variable of A.

(T8) If $t = u$ is an equational consequence of $\mathscr{E}$, and if $\Gamma \vdash_{\mathscr{E}} \tau: A[t/x]$, then $\Gamma \vdash_{\mathscr{E}} \tau: A[u/x]$; t, u are closed $\mathscr{L}$-terms; x is an individual variable, which is the only free variable of A.

The system $\mathscr{F}$ of Girard is the subsystem where we only have propositional variables and constants (predicate variables or constants of arity 0). So, first-order

variables, function symbols and finite sets of equations are useless. The rules for typed terms are T1, T2, T3, and T6, T7 restricted to propositional variables.

It is clear that every construction of a typed term with the full system can be brought back to system $\mathscr{F}$ by simply erasing first-order symbols. So the normalization theorem for the system $\mathscr{F}$ [3] gives the same theorem for the full system.

Theorem 2.3. *If* $\Gamma \vdash_{\mathscr{E}} \tau : A$ *can be obtained by the rules* $T1, \ldots, T8$, *then* τ *is a strongly normalizable* λ*-term.*

Some methods to use such a system in order to write programs are explained in [7, 9].

2.3. *The* λ*-calculus with the new operator* **c**

We add a λ-constant **c** to the pure λ-calculus. Let us consider the following two rules of reduction, called rules of *head* **c***-reduction*:

(1) $(\lambda x.u)tt_1 \ldots t_k \to (u[t/x])t_1 \ldots t_k$ for every $u, t, t_1, \ldots, t_k \in \Lambda$,

(2) $\mathbf{c}tt_1 \ldots t_k \to (t)\lambda x(x)t_1 \ldots t_k$, for every $t, t_1, \ldots, t_k \in \Lambda$, x being a λ-variable not appearing in $t_1, \ldots, t_k$.

For any λ-terms t, t', we shall write $t \succ_{\mathbf{c}} t'$ if t' is obtained from t by applying these rules finitely many times. We say that t' is obtained from t by head **c**-reduction.

Remark. We shall consider λ-terms written with **c** as programs whose execution is head **c**-reduction. Then, we can see that a λ-term like $(\mathbf{c})\lambda x.\tau$, where τ is a λ-term which does not contain the variable x, modelizes an instruction like exit(τ) ("exit" is to be understood as in the C programming language), which executes the program τ at the top level. In fact, consider a λ-term t, in which there is a subterm $(\mathbf{c})\lambda x.\tau$. If during the execution (i.e. the head **c**-reduction) of t, this subterm arrives in the active position, which is the head position, the term t has taken the form $((\mathbf{c})\lambda x.\tau)\tau_1 \ldots \tau_n$, which gives τ in two steps of head **c**-reduction. So, the instruction exit$(\cdot)$ can be represented by the λ-term $\lambda y(\mathbf{c})\lambda x.y$.

The following lemma is clear.

Lemma 2.4. *If* $t \succ_{\mathbf{c}} t'$, *then* $t[t_1/x_1, \ldots, t_n/x_n] \succ_{\mathbf{c}} t'[t_1/x_1, \ldots, t_n/x_n]$ (*in other words,* $S(t) \succ_{\mathbf{c}} S(t')$ *for any substitution* S).

Let us consider now typed λ-calculus of second order, as defined before, in which the λ-constant **c** is always declared of type $\forall X(\neg\neg X \to X)$. By the Curry–Howard isomorphism, a construction of a typed term of the form $x_1 : A_1, \ldots, x_k : A_k$, $\mathbf{c} : \forall X(\neg\neg X \to X) \vdash_{\mathscr{E}} \tau : A$, with the rules T1 to T8, correspond precisely to an intuitionistic proof of $A_1, \ldots, A_k, \forall X(\neg\neg X \to X) \vdash_{\mathscr{E}} A$, i.e. to a classical proof of $A_1, \ldots, A_k \vdash_{\mathscr{E}} A$.

Notation. We shall often write $x_1 : A_1, \ldots, x_k : A_k \vdash^{\mathbf{c}}_{\mathscr{E}} \tau : A$ instead of $x_1 : A_1, \ldots, x_k : A_k$, $\mathbf{c} : \forall X(\neg\neg X \to X) \vdash_{\mathscr{E}} \tau : A$.

Intuitively speaking, $\bot$ is the type of executable programs, i.e. the type of programs to which we can apply the process of head reduction. Programs of other types are modules, used to build executable programs, but are not executable themselves.

Remark. The λ-term $\lambda y(\mathbf{c})\lambda x.y$ considered above to represent the instruction exit$(\cdot)$ can be given the type $\forall X(\bot \to X)$ which is the "absurdity rule" of intuitionistic logic. In fact, since we shall substitute to y a program executed at the top level, it is natural to declare $y : \bot$. Then $y : \bot \vdash^{\mathbf{c}} \lambda x.y : (X \to \bot) \to \bot$ and $y : \bot \vdash^{\mathbf{c}} (\mathbf{c})\lambda x.y : X$, so that $\vdash^{\mathbf{c}} \lambda y(\mathbf{c})\lambda x.y : \forall X(\bot \to X)$.

Since $\bot$ is the type of executable programs, it must be preserved by head $\mathbf{c}$-reduction. This is the meaning of the following proposition.

Proposition 2.5. *If* $x_1 : A_1, \ldots, x_k : A_k, \vdash^{\mathbf{c}}_{\mathscr{E}} \tau : \bot$, *and* $\tau \succ_{\mathbf{c}} \tau'$, *then* $x_1 : A_1, \ldots, x_k : A_k, \vdash^{\mathbf{c}}_{\mathscr{E}} \tau' : \bot$.

A formula will be said to be *open* if it does not begin by $\forall$. Every formula F of the system $\mathscr{F}$ can be written, in only one way, in the form: $F \equiv \forall X_1 \ldots \forall X_n F^\circ$, where F° is an open formula, which will be called the *interior of* F. It is clear that, if $\Gamma \vdash t : F$, then $\Gamma \vdash t : F^\circ$. A context Γ will be called a $\mathbf{c}$-context if it contains the declaration $\mathbf{c} : \forall X(\neg\neg X \to X)$. We first prove the following proposition.

Proposition 2.6. *Let* Γ *be a* $\mathbf{c}$*-context, and assume that* $\Gamma \vdash \mathbf{c}ut_1 \ldots t_n : A$ *in the system* $\mathscr{F}$. *Then there is a formula* B *such that* $\Gamma \vdash u : \neg\neg B$, *and* $\Gamma, x : B \vdash xt_1 \ldots t_n : A^\circ$ (x *being a* λ*-variable not appearing in* $\Gamma, t_1, \ldots, t_n$).

Proof. The proof is by induction on the construction, in the system $\mathscr{F}$, of the typed term $\Gamma \vdash \mathbf{c}ut_1 \ldots t_n : A$. Let us consider the last application of rules T1, T2, T3, T6, T7: it cannot be T1 or T2 since $\mathbf{c}ut_1 \ldots t_n$ is not a variable, and does not begin by λ.

If it is T3, then there are two cases:

(i) $n > 0$; then $\Gamma \vdash \mathbf{c}ut_1 \ldots t_{n-1} : F \to A$, $t_n : F$. By induction hypothesis, there is a formula B such that $\Gamma \vdash u : \neg\neg B$, and $\Gamma, x : B \vdash xt_1 \ldots t_{n-1} : F \to A$. So $\Gamma, x : B \vdash xt_1 \ldots t_n : A$, and also $\Gamma, x : B \vdash xt_1 \cdots t_n : A^\circ$.

(ii) $n = 0$; then we have $\Gamma \vdash \mathbf{c} : F \to A$, $u : F$. By Lemma VIII.1 of [7], the formula $F \to A$ is of the form $\neg\neg\Phi \to \Phi$. It follows that $F \equiv \neg\neg A$, and $\Gamma \vdash u : \neg\neg A$. Since it is clear that $\Gamma, x : A \vdash x : A^\circ$, we obtain the result, with $B = A$.

If it is the rule T6, then $A = \Phi[F/X]$, and $\Gamma \vdash \mathbf{c}ut_1 \ldots t_n : \forall X\, \Phi$. We can suppose that the variable X does not appear in Γ. By induction hypothesis, there is a formula D such that $\Gamma \vdash u : \neg\neg D$, and $\Gamma, x : D \vdash xt_1 \ldots t_n : \Phi^\circ$. It follows that $\Gamma[F/X] \vdash u : \neg\neg D[F/X]$, and $\Gamma[F/X], x : D[F/X] \vdash xt_1 \ldots t_n : \Phi^\circ[F/X]$. But

$\Gamma[F/X]$ is Γ itself. Let $\Psi = \Phi^\circ[F/X]$, and $B = D[F/X]$; then $\Gamma \vdash u : \neg\neg B$, and Γ, $x : B \vdash xt_1 \ldots t_n : \psi$. It follows that Γ, $x : B \vdash xt_1 \ldots t_n : \Psi^\circ$, which is the desired result, since $\Psi^\circ \equiv A^\circ$.

If it is the rule T7, then $A = \forall X\, F$, and $\Gamma \vdash \mathbf{c}ut_1 \ldots t_n : F[P/X]$, P being a propositional constant not appearing in Γ. By induction hypothesis, we have $\Gamma \vdash u : \neg\neg B$ and Γ, $x : B \vdash xt_1 \ldots t_n : F^\circ$. The result follows, since $F^\circ \equiv A^\circ$. □

Proof of Proposition 2.5. We can now prove Proposition 2.5 (only for the system $\mathscr{F}$, for sake of brevity): we can suppose that τ' is obtained from τ by one step of head **c**-reduction. If this is a step of head reduction, the result is already known. Otherwise, $\tau = \mathbf{c}u\tau_1 \ldots \tau_k$ and $\tau' = (u)\lambda x(x)\tau_1 \ldots \tau_k$. So, we have $\Gamma \vdash \mathbf{c}ut_1 \ldots t_k : \bot$. By Proposition 2.6 (with $A \equiv \bot$), there is a formula B such that $\Gamma \vdash u : \neg\neg B$, and $\Gamma, x : B \vdash xt_1 \ldots t_k : \bot$. Then $\Gamma \vdash \lambda x(x)t_1 \ldots t_k : \neg B$, so that $\Gamma \vdash (u)\lambda x(x)t_1 \ldots t_k : \bot$. □

3. Realizability

3.1. Λ-models

A subset $\mathscr{X}$ of Λ will be called *saturated* if $u[t/x]t_1 \ldots t_k \in \mathscr{X} \Rightarrow (\lambda x.u)tt_1 \ldots t_k \in \mathscr{X}$ for every $k \geqslant 0$, $t, u, t_1, \ldots, t_k \in \Lambda$. In other words, $\mathscr{X}$ is saturated if $t \in \mathscr{X}$, $t' \succ t \Rightarrow t' \in \mathscr{X}$.

If $\mathscr{X}, \mathscr{Y} \subset \Lambda$, we define $\mathscr{X} \to \mathscr{Y}$ to be $\{t \in \Lambda;\ t\xi \in \mathscr{Y}$ for every $\xi \in \mathscr{X}\}$. It is clear that, if $\mathscr{Y}$ is saturated, the same is true for $\mathscr{X} \to \mathscr{Y}$, for any $\mathscr{X} \subset \Lambda$.

Let $\mathscr{P}_s(\Lambda)$ be the set of all saturated subsets of Λ. A subset $\mathfrak{R}$ of $\mathscr{P}_s(\Lambda)$ will be called *adequate* if $\mathscr{X}, \mathscr{Y} \in \mathfrak{R} \Rightarrow (\mathscr{X} \to \mathscr{Y}) \in \mathfrak{R}$, and for every subset $\mathfrak{S}$ of $\mathfrak{R}$, the intersection of $\mathfrak{S}$ belongs to $\mathfrak{R}$. In particular, $\Lambda \in \mathfrak{R}$ (take $\mathfrak{S} = \emptyset$).

Let $\mathscr{L}$ be a second-order language. We shall now define the notion of Λ-model for $\mathscr{L}$; it is a modification of the classical notion of second-order model, in which the set of truth values is not $\{0, 1\}$ as usual, but an adequate subset $\mathfrak{R}$ of $\mathscr{P}_s(\Lambda)$. (To avoid any confusion, let us notice that this notion of Λ-model has nothing to do with the notion of model of λ-calculus used in denotational semantics).

A *Λ-model* $\mathscr{M}$ for the language $\mathscr{L}$ is composed of the following data:

- a nonempty set $[\mathscr{M}]$, called *the universe* of the Λ-model $\mathscr{M}$,
- an adequate subset $\mathfrak{R}$ of $\mathscr{P}_s(\Lambda)$; $\mathfrak{R}$ will be called the *truth value set* of the Λ-model $\mathscr{M}$,
- for every n-ary functional symbol f of $\mathscr{L}$, a function $f_{\mathscr{M}} : [\mathscr{M}]^n \to [\mathscr{M}]$,
- for every n-ary predicate constant P of $\mathscr{L}$, a function $P_{\mathscr{M}} : [\mathscr{M}]^n \to \mathfrak{R}$. For $n = 0$, this means that for each propositional constant P (for example, $\bot$), we are given an element $P_{\mathscr{M}} \in \mathfrak{R}$.

Let us now define the value of a second-order formula of $\mathscr{L}$ in the Λ-model $\mathscr{M}$. In order to do this, we shall bound the individual variables to $[\mathscr{M}]$, and the n-ary predicate variables to $\mathfrak{R}^{[\mathscr{M}]^n}$.

The formulas of $\mathscr{L}$ with parameters in $\mathscr{M}$ are defined to be the formulas of the language $\mathscr{L}_{\mathscr{M}}$ obtained by adding to $\mathscr{L}$ each element of $[\mathscr{M}]$ as a constant symbol, and each element of $R^{[\mathscr{M}]^n}$ as an n-ary predicate constant. Each new symbol is interpreted in $\mathscr{M}$ as itself, so that $\mathscr{M}$ can be considered, in an obvious manner, as a Λ-model of $\mathscr{L}_{\mathscr{M}}$.

Let F be a closed second-order formula of $\mathscr{L}$ with parameters in $\mathscr{M}$. *The value of F in the Λ-model $\mathscr{M}$*, denoted by $|F|_{\mathscr{M}}$ is an element of $\mathfrak{R}$ inductively defined by the following conditions:

- If F is an atomic formula, then $F \equiv P(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are closed $\mathscr{L}_{\mathscr{M}}$-terms, and P an n-ary predicate constant of $\mathscr{L}_{\mathscr{M}}$. Let $m_1, \ldots, m_n \in [\mathscr{M}]$ be the values of $t_1, \ldots, t_n$ in $\mathscr{M}$; then we define $|F|_{\mathscr{M}}$ as $P_{\mathscr{M}}(m_1, \ldots, m_n)$ which is an element of R.
- If $F \equiv G \to G'$, then $|F|_{\mathscr{M}} = |G|_{\mathscr{M}} \to |G'|_{\mathscr{M}}$,
- If $F \equiv \forall x G$, where x is a first-order variable, then $|F|_{\mathscr{M}} = \bigcap \{|G[m/x]|_{\mathscr{M}}; m \in [\mathscr{M}]\}$,
- If $F \equiv \forall X G$, where X is an n-ary predicate variable, then $|F|_{\mathscr{M}} = \bigcap \{|G[\Phi/X]|_{\mathscr{M}}; \Phi \in \mathscr{R}^{[\mathscr{M}]^n}\}$.

This definition can be given in terms of the notion of *realizability*. We shall say that a λ-term τ *realizes* F in the Λ-model $\mathscr{M}$ (F being a closed formula with parameters in $\mathscr{M}$), if and only if $\tau \in |F|_{\mathscr{M}}$. A notation for this is $\tau \Vdash_{\mathscr{M}} F$, or even $\tau \Vdash F$, if there is no ambiguity about the Λ-model in use. Clearly, we have the following inductive definition of realizability:

- If F is a closed atomic formula $P(t_1, \ldots, t_n)$, then $\tau \Vdash F \Leftrightarrow \tau \in P_{\mathscr{M}}(m_1, \ldots, m_n)$, m_i being the value in $\mathscr{M}$ of the closed $\mathscr{L}_{\mathscr{M}}$-term t_i,
- $\tau \Vdash G \to G' \Leftrightarrow \tau\xi \Vdash G'$ for every $\xi \in \Lambda$, $\xi \Vdash G$,
- If x is an individual variable, then $\tau \Vdash \forall x\, G \Leftrightarrow \tau \Vdash G[m/x]$ for every $m \in [\mathscr{M}]$,
- If X is an n-ary predicate variable, then $\tau \Vdash \forall X\, G \Leftrightarrow \tau \Vdash G[\Phi/X]$ for every $\Phi \in \mathfrak{R}^{[\mathscr{M}]^n}$.

If we restrict ourselves to the system $\mathscr{F}$, a Λ-model $\mathscr{M}$ is only composed of an adequate subset R of $\mathscr{P}_s(\Lambda)$, and, for each propositional constant P, an element $|P|_{\mathscr{M}}$ of $\mathfrak{R}$.

The following important lemma links the notions of typed terms and realizability in a Λ-model. First, we need a definition.

Let $\mathscr{M}$ be a Λ-model for $\mathscr{L}$, and $t = u$ an equation of $\mathscr{L}$. We shall say that $\mathscr{M}$ satisfies $t = u$, if the closure of this formula is true in $\mathscr{M}$; in other words, $x_1, \ldots, x_k$ being the free variables of t, u, $\mathscr{M}$ satisfies $t = u$ if, and only if, for every $a_1, \ldots, a_k \in [\mathscr{M}]$, the terms $t[a_1/x_1, \ldots, a_k/x_k]$ and $u[a_1/x_1, \ldots, a_k/x_k]$ have the same value in $\mathscr{M}$.

If $\mathscr{E}$ is a set of equations of $\mathscr{L}$, we shall say that $\mathscr{M}$ satisfies $\mathscr{E}$ (in symbols $\mathscr{M} \models \mathscr{E}$), or that $\mathscr{M}$ is a model of $\mathscr{E}$, if $\mathscr{M}$ satisfies each equation of $\mathscr{E}$.

Lemma 3.1. *Let $\mathscr{E}$ be a finite set of equations in the language $\mathscr{L}$, and $\mathscr{M}$ a Λ-model of $\mathscr{E}$. Suppose that $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : A$ for some closed formulas $A_1, \ldots, A_k$, A with*

parameters in $\mathcal{M}$. *If* $\xi_1 \Vdash_{\mathcal{M}} A_1, \ldots, \xi_k \Vdash_{\mathcal{M}} A_k$, *then* $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} A$. *In particular, if* $x_1 \in |A_1|_{\mathcal{M}}, \ldots, x_k \in |A_k|_{\mathcal{M}}$ *then* $\tau \in |A|_{\mathcal{M}}$.

Proof. Let $\xi_1, \ldots, \xi_k \in \Lambda$, such that $\xi_1 \Vdash_{\mathcal{M}} A_1, \ldots, \xi_k \Vdash_{\mathcal{M}} A_k$, be fixed throughout the proof. We reason by induction on the length of the proof of $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : A$, by means of the rules T1–T8. Let us consider the last rule used:

If it is T1, then $\tau \equiv x_i$ and $A \equiv A_i$, so that the result is trivial.

If it is T2, then $\tau \equiv \lambda x.\tau'$, $A \equiv B \to C$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k, x : B \vdash_{\mathscr{E}} \tau' : C$. If $\xi \Vdash_{\mathcal{M}} B$, then, by induction hypothesis, we get $\tau'[\xi_1/x_1, \ldots, \xi_k/x_k, \xi/x] \in |C|_{\mathcal{M}}$. But $|C|_{\mathcal{M}}$ is a saturated subset of Λ, so that $(\lambda x \tau'[\xi_1/x_1, \ldots, \xi_k/x_k])\xi \in |C|_{\mathcal{M}}$. Since this is true for every ξ which realizes B, we obtain $\lambda x \tau'[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B \to C|_{\mathcal{M}}$, i.e. $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} B \to C$.

If it is T3, then $\tau \equiv \tau'\tau''$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau' : B \to A$, $\tau'' : B$. By induction hypothesis, we get $\tau'[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} B \to A$, $\tau''[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} B$, and so $(\tau'\tau'')[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} A$.

If it is T4, then $A \equiv B[t/x]$, where t is a closed $\mathscr{L}$-term with parameters in $\mathcal{M}$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : \forall x\, B$. By induction hypothesis, we get $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} \forall x\, B$, and so $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \Vdash_{\mathcal{M}} B[a/x]$ for every $a \in [\mathcal{M}]$. We take for a the value of t in $\mathcal{M}$, and get the desired result by Lemma 3.2 below.

If it is T5, then $A \equiv \forall x\, B$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : B[c/x]$, c being an individual constant which does not appear in $A_1, \ldots, A_k$. If $a \in [\mathcal{M}]$, let $\mathcal{M}'$ be the Λ-model obtained from $\mathcal{M}$ by only changing the value of c, and giving to it the value a. By induction hypothesis, we have $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[c/x]|_{\mathcal{M}'}$ since $|A_i|_{\mathcal{M}'} = |A_i|_{\mathcal{M}}$. This shows that $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[a/x]|_{\mathcal{M}}$. Since this is true for every $a \in [\mathcal{M}]$, we get $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |\forall x\, B|_{\mathcal{M}}$.

If it is T6, then $A \equiv B[F/Xx_1 \ldots x_n]$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : \forall X\, B$. By induction hypothesis, we have $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |\forall X\, B|_{\mathcal{M}}$, so that $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[\Phi/X]|_{\mathcal{M}}$, for every $\Phi \in \mathfrak{R}^{[\mathcal{M}]^n}$. By Lemma 3.3 below, it follows that $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[F/Xx_1 \ldots x_n]|_{\mathcal{M}}$.

If it is T7, then $A \equiv \forall X\, B$, and we previously obtained $x_1 : A_1, \ldots, x_k : A_k \vdash_{\mathscr{E}} \tau : B[P/X]$, P being an n-ary predicate constant which does not appear in $A_1, \ldots, A_k$. Let $\Phi \in \mathfrak{R}^{[\mathcal{M}]^n}$, and $\mathcal{M}'$ the Λ-model obtained from $\mathcal{M}$ by only changing the value of the predicate constant P, and giving to it the value Φ. By induction hypothesis, we have $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[P/X]|_{\mathcal{M}'}$ since $|A_i|_{\mathcal{M}'} = |A_i|_{\mathcal{M}}$ (P does not appear in A_i). This shows that $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[\Phi/X]|_{\mathcal{M}}$. Since this is true for every $\Phi \in \mathfrak{R}^{[\mathcal{M}]^n}$, we get $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |\forall X\, B|_{\mathcal{M}}$.

If it is T8, then $A \equiv B[u/x]$, $t = u$ is an equational consequence of $\mathscr{E}$, and we previously obtained $x_1 : A_1, \ldots, x_k : A \vdash_{\mathscr{E}} \tau : B[t/x]$. By induction hypothesis, we have $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[t/x]|_{\mathcal{M}}$. Since $\mathscr{E}$ is true in $\mathcal{M}$, so that $\mathcal{M}$ satisfies $t = u$, we have $|B[t/x]|_{\mathcal{M}} = |B[u/x]|_{\mathcal{M}}$ and $\tau[\xi_1/x_1, \ldots, \xi_k/x_k] \in |B[u/x]|_{\mathcal{M}}$. □

The following two lemmas have been used in the proof of Lemma 3.1. They are easily proved, by induction on the formula A:

Lemma 3.2. *Let t be a closed $\mathscr{L}$-term and A a formula, both with parameters in the Λ-model $\mathscr{M}$. If x is the unique free variable of A, and a is the value of t in $\mathscr{M}$, then* $|A[t/x]|_{\mathscr{M}} = |A[a/x]|_{\mathscr{M}}$.

Lemma 3.3. *Let F be a formula with parameters in the Λ-model $\mathscr{M}$, whose free variables are $x_1, \ldots, x_n$, and let $\Phi \in \mathfrak{R}^{[\mathscr{M}]^n}$ be defined by $\Phi(a_1, \ldots, a_n) = |F[a_1/x_1, \ldots, a_n/x_n]|_{\mathscr{M}}$ for every $a_1, \ldots, a_n \in [\mathscr{M}]$. If A is a formula whose only free variable is the n-ary predicate variable X, then* $|A[F/Xx_1 \ldots x_n]|_{\mathscr{M}} = |A[\Phi/X]|_{\mathscr{M}}$.

3.2. About data types

We intend to study the properties of some special types, named *data types* (like booleans, integers, lists, trees, etc.). These types are second-order formulas which characterize "concrete mathematical objects". In the present paper, we shall neither give nor use the general definition for these types. Instead, we shall consider the two special cases of booleans and integers, which are generic enough for our purposes. The general case will be treated elsewhere.

In the system $\mathscr{F}$, the type of booleans is the formula $\mathrm{Bool} \equiv \forall X(X, X \to X)$, the type of integers is the formula $\mathrm{Int} \equiv \forall X((X \to X), X \to X)$, where X is a propositional variable. Other simple examples of data types in system $\mathscr{F}$ are: the sum and product of the two data types A and B, which are $\forall X[(A \to X), (B \to X) \to X]$ and $\forall X[(A, B \to X) \to X]$; the type of lists of objects of the data type A, which is $\forall X[(A, X \to X), X \to X]$, etc.

In the full system of second-order types, we define the formulas:

$$\mathrm{Bool}[x] \equiv \forall X(X\underline{1}, X\underline{0} \to Xx)$$

where $\underline{0}, \underline{1}$ are individuals constants of the language $\mathscr{L}$;

$$\mathrm{Int}[x] \equiv \forall X\{\forall y(Xy \to Xsy), X0 \to Xx\},$$

where 0 is an individual constant, and s a unary function symbol of $\mathscr{L}$.

Let $\mathscr{E}$ be a set of equations in the language $\mathscr{L}$. If we want to use some data types, for example $\mathrm{Bool}[x]$ or $\mathrm{Int}[x]$, with $\mathscr{E}$ as logical axioms, it is clear that $\mathscr{E}$ must satisfy some compatibility conditions with the data types in use. Let us state these conditions for the types of booleans and of integers.

We shall say that $\mathscr{E}$ is a *system of equations for booleans*, if $\mathscr{L}$ contains the symbols of constant $\underline{1}$ and $\underline{0}$, and if $\underline{0} = \underline{1}$ *is not* an equational consequence of $\mathscr{E}$.

We shall say that $\mathscr{E}$ is a *system of equations for integers*, if $\mathscr{L}$ contains the symbol of constant 0, the symbol of unary function s, and for any $\mathscr{L}$-term t and any $k \in \mathbb{N}$: $st = 0$ *is not* an equational consequence of $\mathscr{E}$; if $st = s^{k+1}0$ is an equational consequence of $\mathscr{E}$, so is $t = s^k 0$.

Note that, clearly, $\emptyset$ is a system of equations for booleans, or for integers.

In the case of a data type, the normalization theorem (Theorem 2.3) can be made more precise, to indicate the operational behaviour of the λ-term. We shall only

consider the types of booleans and integers, first in the system $\mathscr{F}$, and then in the full second-order type system.

Theorem 3.4. (i) *If* $\vdash \tau$: Bool *in the system* $\mathscr{F}$, *then* $\tau \simeq_\beta \lambda x\,\lambda y\,yx$ *or* $\tau \simeq_\beta \lambda x\,\lambda y\,y$.
(ii) If $\vdash \tau$: Int *in the system* $\mathscr{F}$, *then* $\tau \simeq_\beta \lambda x\,x$ *or* $\tau \simeq_\beta \lambda f\lambda x(f)^n x$ *for some* $n \in \mathbb{N}$.

Theorem 3.5. *Let* $\mathscr{E}$ *be a system of equations for booleans, in the language* $\mathscr{L}$, *and* t *a* $\mathscr{L}$*-term. If* $\vdash_{\mathscr{E}} \tau$: Bool$[t]$, *then either* $\mathscr{E} \vdash t = \underline{1}$ *and* $\tau \simeq_\beta \lambda x\,\lambda y\,yx$, or $\mathscr{E} \vdash t = \underline{0}$ *and* $\tau \simeq_\beta \lambda x\,\lambda y\,y$.

Theorem 3.6. *Let* $\mathscr{E}$ *be a system of equations for integers, in the language* $\mathscr{L}$, *and* t *a* $\mathscr{L}$*-term. If* $\vdash_{\mathscr{E}} \tau$: Int$[t]$, *then there is an* $n \in \mathbb{N}$ *such that* $\mathscr{E} \vdash t = s^n 0$, *and either* $\tau \simeq_\beta \lambda f\lambda x(f)^n x$, *or* $n = 1$ *and* $\tau \simeq_\beta \lambda x\,x$.

Briefly, these theorems tell us that if a λ-term τ is of type Bool or Int, then it behaves in programs like a boolean or an integer. Moreover, if τ is of type Bool$[\underline{1}]$ or Int$[s^5 0]$, then it behaves in programs like the boolean **1** (i.e. $\lambda x\lambda y yx$) or the Church integer 5 (i.e. $\lambda f\lambda x(f)^5 x$).

These theorems indicate also that a context like x: Int(resp. x: Int$[s^5 0]$) modelizes the following declaration of variable (in the *C* language, taken as a prototype of imperative languages): int x; (resp. int $x = 5$;).

We shall prove only the assertions concerning integers (the case of booleans is similar, but easier), i.e. Theorems 3.4(ii) and 3.6. With this aim, we need some definitions and results about models.

Let $\mathscr{E}$ be a system of equations for integers, and $\mathscr{M}_0$ be a model of $\mathscr{E}$ (in the sense of model theory). Let m_n be the value, in $\mathscr{M}_0$, of the term $s^n 0$. The m_n will be called the "integers" of $\mathscr{M}_0$. $\mathscr{M}_0$ will be called a *regular model of* $\mathscr{E}$, if the m_n are distinct ($n \neq k \Rightarrow m_n \neq m_k$), and if every $m \in \mathscr{M}$ such that sm is an integer of $\mathscr{M}_0$ is itself an integer of $\mathscr{M}_0$.

Lemma 3.7. *If* $\mathscr{E}$ *is a system of equations for integers, there exists a regular model of* $\mathscr{E}$.

Proof. Let $\mathscr{T}$ be the set of $\mathscr{L}$-terms, and define on $\mathscr{T}$ an equivalence relation in the following way: $t \simeq u \Leftrightarrow \mathscr{E} \vdash t = u$. The universe $[\mathscr{M}_0]$ of the wanted model is the quotient set $\mathscr{T}/\simeq$. Each symbol of function is defined in the canonical way on this universe. We check that the model obtained in this way is regular: if $n \neq k$ and $\mathscr{M}_0 \models s^n 0 = s^k 0$, then $\mathscr{E} \vdash s^n 0 = s^k 0$, which is impossible, by hypothesis on $\mathscr{E}$. If $\mathscr{M}_0 \models st = s^k 0$ for some $\mathscr{L}$-term t and some $k \in \mathbb{N}$, then $\mathscr{E} \vdash st = s^k 0$. It follows that $k > 0$. Let $k = n + 1$; then, by hypothesis on $\mathscr{E}$, we have $\mathscr{E} \vdash t = s^n 0$, and, finally, $\mathscr{M}_0 \models t = s^n 0$. □

A Λ-model $\mathscr{M}$ will be called *standard* if its truth value set is the maximal one, i.e. $\mathscr{P}_s(\Lambda)$ (the set of all saturated subsets of Λ).

The following theorems mean that, in a "good" Λ-model, the λ-terms which realize a data type are the expected ones.

Theorem 3.8. *Let $\mathcal{M}$ be a standard Λ-model for the system $\mathcal{F}$, and τ a λ-term. Then:*
(i) *If $\tau \Vdash_{\mathcal{M}} \mathrm{Bool}$ in the system $\mathcal{F}$, then $\tau \simeq_\beta \lambda x \lambda y x$ or $\tau \simeq_\beta \lambda x \lambda y y$.*
(ii) *If $\tau \Vdash_{\mathcal{M}} \mathrm{Int}$ in the system $\mathcal{F}$, then $\tau \simeq_\beta \lambda x x$ or $\tau \simeq_\beta \lambda f \lambda x (f)^n x$ for some $n \in \mathbb{N}$.*

Proof. We shall prove only (ii). Let $\Xi = \{t \in \Lambda;\ t \simeq_\beta (f)^n a \text{ for some } n \in \mathbb{N}\}$, where f, a are distinct λ-variables, not in τ. Then Ξ is saturated and it is clear that $a \in \Xi$ and that $f \in \Xi \to \Xi$. But, by hypothesis, $\tau \in |\forall X((X \to X), X \to X)|_{\mathcal{M}}$, and, therefore, $\tau \in |(\Xi \to \Xi), \Xi \to \Xi|_{\mathcal{M}}$.

It follows that $\tau f a \in \Xi$, and there is an integer n such that $\tau f a \simeq_\beta (f)^n a$.

It follows that τ is resoluble; let $\tau' \simeq_\beta \tau$ be its head normal form. If τ' does not begin by λ, we set $\tau' = (z) t_1 \ldots t_k$, and we have $(z) t_1 \ldots t_k f a \simeq_\beta (f)^n a$, which is impossible, for every integer n. If τ' begins by only one λ, we can set $\tau' = \lambda f (z) t_1 \ldots t_k$, and we have $(z) t_1 \ldots t_k a \simeq_\beta (f)^n a$. It follows that $k = 0$, $n = 1$ and $z = f$, so that $\tau' = \lambda f . f$. Finally, if τ' begins by at least two λ's, we can set $\tau' = \lambda f \lambda a \tau''$, and we get $\tau'' \simeq_\beta (f)^n a$, so that $\tau \simeq_\beta \lambda f \lambda a (f)^n a$. □

Theorem 3.4. is a trivial consequence of Theorem 3.8 and Lemma 3.1. In order to prove theorem 3.6 we shall need the following theorem.

Theorem 3.9. *Let $\mathcal{E}$ be a system of equations for integers, in the language $\mathcal{L}$, t a $\mathcal{L}$-term, and $\mathcal{M}$ a standard Λ-model of $\mathcal{E}$, which is regular. If τ is a λ-term such that $\tau \Vdash_{\mathcal{M}} \mathrm{Int}[t]$, then there is an $n \in \mathbb{N}$ such that $\mathcal{M} \models t = s^n 0$, and either $\tau \simeq_\beta \lambda f \lambda x (f)^n x$, or $n = 1$ and $\tau \simeq_\beta \lambda x x$.*

Proof. Let us denote by m_n the value, in $\mathcal{M}$, of the $\mathcal{L}$-term $s^n 0$ (the m_n are the so-called integers of $\mathcal{M}$). By hypothesis, the m_n are distinct, and, if $m \in \mathcal{M}$ is not an integer of $\mathcal{M}$, then neither is $s(m)$. Thus, we can define a function $\Xi : [\mathcal{M}] \to \mathcal{P}_s(\Lambda)$ in the following way: $\Xi(m) = \emptyset$ if m is not an integer of $\mathcal{M}$; $\Xi(m_n) = \{t \in \Lambda;\ t \simeq_\beta (f)^n a\}$, where f, a are distinct λ-variables, not in τ. It is clear that $f \in (\Xi(m_n) \to \Xi(s m_n))$, since $s(m_n) = m_{n+1}$. Also, trivially, $f \in (\Xi(m) \to \Xi(sm))$ if m is not an integer of $\mathcal{M}$, since $\Xi(m) = \Xi(sm) = \emptyset$. It follows that $f \in |\forall y(\Xi(y) \to \Xi(sy))|_{\mathcal{M}}$. It is clear that $a \in |\Xi(0)|_{\mathcal{M}} = \Xi(m_0)$. Since, by hypothesis, $\tau \in |\forall X\{\forall y(Xy \to Xsy), X0 \to Xt\}|_{\mathcal{M}}$, it follows that $\tau f a \in |\Xi(t)|_{\mathcal{M}}$. Thus, there is an integer n such that $\mathcal{M} \models t = s^n 0$, and $\tau f a \simeq_\beta (f)^n a$.

It follows, as in the proof of Theorem 3.8 that $\tau \simeq_\beta \lambda f \lambda a (f)^n a$, or $n = 1$ and $\tau \simeq_\beta \lambda f . f$. □

Proof of Theorem 3.6. By Lemma 3.7, there exists a regular model $\mathcal{M}_0$ of $\mathcal{E}$. Then, we can define a standard Λ-model $\mathcal{M}$, such that $[\mathcal{M}] = [\mathcal{M}_0]$, in which the interpretation of function symbols is the same as in $\mathcal{M}_0$. It then follows from Lemma 3.1 that $\tau \Vdash_{\mathcal{M}} \mathrm{Int}[t]$, and from Theorem 3.9 that $\mathcal{M} \models t = s^n 0$ and $\tau \simeq_\beta \lambda f \lambda x (f)^n x$, or

$\mathscr{M} \models t = s0$ and $\tau \simeq_\beta \lambda x\, x$. Now, we have $\mathscr{M}_0 \models t = s^n 0$. By the very definition of $\mathscr{M}_0$, in the proof of Lemma 3.7, it follows that $\mathscr{E} \vdash t = s^n 0$. □

3.3. Storage operators and Gödel translation

Let T be a closed λ-term, and τ an essentially closed λ-term (i.e. τ is β-equivalent to a closed term). We shall say that T is a *storage operator for* τ, if f being a variable, there exists $\tau_0 \simeq_\beta \tau$, whose free variables are among $x_1, \ldots, x_k, f$, such that, for any term $\theta \simeq_\beta \tau$, we have $Tf\theta \succ f\tau_0[\theta_1/x_1, \ldots \theta_k/x_k]$.

Let E be a set of essentially closed λ-terms. T is called a *storage operator for* E, if it is a storage operator for each $\tau \in E$.

Let then φ be any λ-term, and θ a λ-term which is β-equivalent to $\tau \in E$. During the computation (i.e. the head reduction) of $\varphi\theta$, θ is likely to be computed many times (for example, every time it comes in head position). Instead of computing $\varphi\theta$, let us look at the head reduction of $T\varphi\theta$. Since it is $(Tf\theta)[\varphi/f]$, by Lemma 2.1, we shall first reduce $Tf\theta$ to its head normal form, which is $f\tau_0[\theta_1/x_1, \ldots, \theta_k/x_k]$, and then compute $\varphi\tau_0[\theta_1/x_1, \ldots, \theta_k/x_k, \varphi/f]$. The computation has been decomposed into two parts, the first being independent of φ. This first part is essentially a computation of θ, the result being τ_0, which is a kind of normal form of θ, because it only depends on the β-equivalence class of θ; the substitutions made in τ_0 have no computational importance, since τ_0 is essentially closed.

So, in the computation of $T\varphi\theta$, θ is computed first, and the result is passed to φ as an argument. T behaves as if it stored the result, so as to be able to supply it, as many times as needed, to any function.

Let us give some examples for the sets of booleans and integers.

If $E = \{\mathbf{0}, \mathbf{1}\}$ (remember that $\mathbf{0}$ is $\lambda x \lambda y\, y$, and $\mathbf{1}$ is $\lambda x \lambda y\, x$), then $T_{\text{Bool}} = \lambda f \lambda x((x)(f)\mathbf{1})(f)\mathbf{0}$ and $T'_{\text{Bool}} = \lambda f \lambda x(((x)\lambda g(g)\mathbf{1})\lambda g(g)\mathbf{0})f$ are storage operators for booleans. In fact, it is easily checked that, if $\theta \simeq_\beta \mathbf{0}$ (resp. $\mathbf{1}$), then $T_{\text{Bool}} f\theta$ and $T'_{\text{Bool}} f\theta \succ f\mathbf{0}$ (resp. $f\mathbf{1}$).

If E is the set of Church integers, let s be a λ-term for the successor (for example, $s = \lambda k \lambda f \lambda x(f)(k)fx)$, and $F = \lambda h \lambda y(h)(s)y = \lambda h.h \circ s$. Then $T_{\text{Int}} = \lambda f \lambda x(x)Ff0$ is a storage operator for integers. Another one is $T'_{\text{Int}} = \lambda f \lambda x(((x)G)\lambda g(g)0)f$, where $G = \lambda h \lambda g(h)\lambda x(g)(s)x$. In fact, it is proved in [8] that, if $\theta \simeq_\beta \lambda f \lambda x(f)^n x$, then $T_{\text{Int}} f\theta$ and $T'_{\text{Int}} f\theta \succ (f)(s)^n 0$.

In order to assign types to storage operators, we shall use Gödel translation. The Gödel translation of a formula A is a formula A^* obtained as follows: if A is atomic, then $A^* = \neg A$; $(A \to B)^* = A^* \to B^*$; $(\forall x\, A)^* = \forall x\, A^*$ (x being any first order variable); $(\forall X\, A)^* = \forall X\, A^*$ (X being any second-order variable).

So, we get A^* by putting a $\neg$ in front of each atomic formula of A. The following theorem is proved in [8].

Theorem 3.10. *Let $\mathscr{E}$ be a system of equations for integers, and T a λ-term such that $\vdash_{\mathscr{E}} T: \forall x(\neg \text{Int}[x] \to \neg \text{Int}^*[x])$. Then T is a storage operator for Church integers.*

This theorem gives a way to obtain storage operators for integers: you only have to find some intuitionistic proof of $\vdash_{\mathscr{E}} \forall x(\neg \mathrm{Int}[x] \to \neg \mathrm{Int}^*[x])$. The two examples T_{Int} and T'_{Int} can be obtained in this way, with $\mathscr{E} = \emptyset$ (this is easy for T'_{Int}, but not trivial for T_{Int}).

The same theorem holds for booleans, and, in fact, for any data type. It also holds for other Gödel translations. But we shall not be concerned here with these generalizations.

4. Head c-reduction and the axiom $\forall X(\neg\neg X \to X)$

From now on, we shall denote by Λ the set of λ-terms written with a new constant $\mathbf{c}$. A subset $\mathscr{X}$ of Λ will be said to be $\mathbf{c}$-*saturated* if it is saturated, and if $(t)\lambda x(x)t_1 \ldots t_k \in \mathscr{X} \Rightarrow \mathbf{c}tt_1 \ldots t_k \in \mathscr{X}$, for every $k \in \mathbb{N}$ and $t, t_1, \ldots, t_k \in \Lambda$, x being a λ-variable not appearing in $t_1, \ldots, t_k$.

Thus, a subset $\mathscr{X}$ of Λ is $\mathbf{c}$-saturated if and only if, for every $t, t' \in \Lambda$, $t \in \mathscr{X}$ and $t' \succ_{\mathbf{c}} t \Rightarrow t' \in \mathscr{X}$.

Let O be a $\mathbf{c}$-saturated subset of Λ. We shall denote by $\mathfrak{R}_O$ the set of all intersections of subsets of Λ of the form $\{t_1\}, \ldots, \{t_k\} \to O$, for $k \in \mathbb{N}$, and $t_1, \ldots, t_k \in \Lambda$. For $k = 0$, this means that $O \in \mathfrak{R}_O$. Also $\Lambda \in \mathfrak{R}_O$, since Λ is the void intersection.

It is clear that $\mathfrak{R}_O$ is also the set of all intersections of subsets of Λ of the form $\mathscr{X}_1, \ldots, \mathscr{X}_k \to O$, for $k \in \mathbb{N}$ and $\mathscr{X}_1, \ldots, \mathscr{X}_k \subset \Lambda$. It follows that $\mathfrak{R}_O$ is an adequate set.

A Λ-model $\mathscr{N}$ will be called $\perp$-*standard* if there is a $\mathbf{c}$-saturated subset O of Λ such that $|\perp|_{\mathscr{N}} = O$, and $\mathfrak{R}_O$ is the truth value set of $\mathscr{N}$.

Lemma 4.1. *If $\mathscr{N}$ is a Λ-model which is $\perp$-standard, then $\mathbf{c} \in |\forall X(\neg\neg X \to X)|_{\mathscr{N}}$.*

Proof. By hypothesis, we have $|\perp|_{\mathscr{N}} = O$, a $\mathbf{c}$-saturated set. Since $\mathfrak{R}_O$ is the truth value set of $\mathscr{N}$, we have to prove that $\mathbf{c} \in |\neg\neg \mathscr{X} \to \mathscr{X}|_{\mathscr{N}}$ for every $\mathscr{X} \in \mathfrak{R}_O$.

Let $t_1, \ldots, t_k \in \Lambda$, and $\mathscr{X} = \{t_1\}, \ldots, \{t_k\} \to O$. We show that $\mathbf{c} \in |\neg\neg \mathscr{X} \to \mathscr{X}|_{\mathscr{N}}$. We note first that $\lambda x(x)t_1 \ldots t_k \in |\neg \mathscr{X}|_{\mathscr{N}}$ (x being not free in $t_1, \ldots, t_k$): in fact, for every $t \in \mathscr{X}$, we have $tt_1 \ldots t_k \in O$. But O is saturated, so that $(\lambda x(x)t_1 \ldots t_k)t \in O$.

Now let $t \in |\neg\neg \mathscr{X}|_{\mathscr{N}}$; from what we have just proved, it follows that $(t)\lambda x(x)t_1 \ldots t_k \in O$. Since O is $\mathbf{c}$-saturated, we get $\mathbf{c}tt_1 \ldots t_k \in O$, that is, $\mathbf{c}t \in \mathscr{X}$. It follows that $\mathbf{c} \in |\neg\neg \mathscr{X} \to \mathscr{X}|_{\mathscr{N}}$. Let now $\mathscr{X} \in \mathfrak{R}_O$; by definition of $\mathfrak{R}_O$, we have $\mathscr{X} = \bigcap_{i \in I} \mathscr{X}_i$, each $\mathscr{X}_i$ being of the form $\{t_1\}, \ldots, \{t_k\} \to O$. Now, we know that $\mathbf{c} \in |\neg\neg \mathscr{X}_i \to \mathscr{X}_i|_{\mathscr{N}}$. If $t \in |\neg\neg \mathscr{X}|_{\mathscr{N}}$, then $t \in |\neg\neg \mathscr{X}_i|_{\mathscr{N}}$ for each $i \in I$, since $\mathscr{X} \subset \mathscr{X}_i$; therefore, $\mathbf{c}t \in \mathscr{X}_i$ for each $i \in I$, and it follows that $\mathbf{c}t \in \mathscr{X}$, which is the desired result. □

Let θ be an "intuitionistic integer", i.e. a λ-term such that $\vdash \theta : \mathrm{Int}$ (resp. $\vdash_{\mathscr{E}} \theta : \mathrm{Int}[s^n 0]$ for some $n \in \mathbb{N}$). Then, by Theorem 3.4(ii) (resp. 3.6) we know that θ is

β-equivalent to a Church integer (resp. to the Church integer n), and, as such, will behave as expected when it will occur in some program.

What happens if θ is a "classical integer", i.e. $\vdash^{c} \theta$: Int, or $\vdash^{c}_{\mathscr{E}} \theta$: Int$[s^n 0]$? Then, it is no longer true that θ is β-equivalent to a Church integer. In fact, in order to compute θ, one has to make use of storage operators. This was first noticed by Parigot [13].

Theorem 4.2. *Let T and θ be λ-terms such that $\vdash T$:$\neg$ Int $\to \neg$ Int* and $\vdash^{c} \theta$: Int in the system $\mathscr{F}$, and let f be any λ-variable. Then $Tf\theta \succ_{c} f\alpha$ where α is β-equivalent to a Church integer or to $\lambda x\, x$.*

Theorem 4.3. (Parigot [14]). *Let T and θ be λ-terms such that $\vdash_{\mathscr{E}} T$: $\forall x(\neg \text{Int}[x] \to \neg \text{Int}^*[x])$ and $\vdash^{c}_{\mathscr{E}} \theta$: Int$[s^n 0]$, $\mathscr{E}$ being a system of equations for integers. If f is any λ-variable, then $Tf\theta \succ_{c} f\alpha$ where either $\alpha \simeq_{\beta} \lambda f \lambda x (f)^n x$, or $n = 1$ and $\alpha \simeq_{\beta} \lambda x\, x$.*

These two theorems are in a certain sense, "classical" counterparts of Theorems 3.4(ii) and 3.6. But the role played by the storage operator T is rather unexpected.

Theorem 4.3 will be used in the proof of Theorem 4.4, which is the main result of the present paper. In fact, Theorem 4.4 is strictly stronger than Theorem 4.3, but its proof is much more complicated.

Proof of Theorem 4.2. We set $O = \{t \in \Lambda;\ t \succ_{c} f\alpha$ for some α which is β-equivalent to $\lambda x\, x$ or to a Church integer$\}$. It is clear that O is **c**-saturated. Let $\mathscr{M}$ (resp. $\mathscr{N}$) be the standard (resp. $\perp$-standard) Λ-model of system $\mathscr{F}$, such that $|\perp|_{\mathscr{M}} = |\perp|_{\mathscr{N}} = O$. By hypothesis, we have **c**:$\forall X(\neg\neg X \to X) \vdash \theta$: Int. By Lemma 3.1, applied to $\mathscr{N}$, we get $\theta \in |\text{Int}|_{\mathscr{N}}$, since, by Lemma 4.1, we know that $\mathbf{c} \in |\forall X(\neg\neg X \to X)|_{\mathscr{N}}$.

We prove that $\theta \in |\text{Int}^*|_{\mathscr{M}} = |\forall X((\neg X \to \neg X), \neg X \to \neg X)|_{\mathscr{M}}$: let $\mathscr{X}$ be a saturated subset of Λ, and $\mathscr{X}' = |\neg \mathscr{X}|_{\mathscr{M}} = \mathscr{X} \to O$. We have to prove that $\theta \in |(\mathscr{X}' \to \mathscr{X}'), \mathscr{X}' \to \mathscr{X}'|$. But it is clear that $\mathscr{X}' \in \mathfrak{R}_O$ which is the truth value set of $\mathscr{N}$. Since $\theta \in |\text{Int}|_{\mathscr{N}} = |\forall X(X \to X), X \to X)|_{\mathscr{N}}$, we get $\theta \in |(\mathscr{X}' \to \mathscr{X}'), \mathscr{X}' \to \mathscr{X}'|$, as desired.

Let $\alpha \in |\text{Int}|_{\mathscr{M}}$. Then, by Theorem 3.8(ii), α is β-equivalent to a Church integer or to $\lambda x\, x$. By definition of O, it follows that $f\alpha \in O$. Since this is true for every $\alpha \in |\text{Int}|_{\mathscr{M}}$, we have shown that $f \in |\neg \text{Int}|_{\mathscr{M}}$.

Now, since $\vdash T$:$\neg$ Int $\to \neg$ Int*, by Lemma 3.1, we have $T \in |\neg \text{Int} \to \neg \text{Int}^*|_{\mathscr{M}}$. We deduce that $Tf \in |\neg \text{Int}^*|_{\mathscr{M}}$, and so $Tf\theta \in O$. This is exactly the conclusion of the theorem. □

Proof of Theorem 4.3. We set $O = \{t \in \Lambda;\ t \succ_{c} f\alpha$ for some $\alpha \simeq_{\beta} \lambda f \lambda x (f)^n x\}$ if $n \neq 1$, and $O = \{t \in \Lambda;\ t \succ_{c} f\alpha$ for some $\alpha \simeq_{\beta} \lambda f \lambda x f x$ or $\simeq_{\beta} \lambda x\, x\}$ if $n = 1$. It is clear that O is **c**-saturated. By Lemma 3.7, there exists a regular model $\mathscr{M}_0$ of $\mathscr{E}$. We define $\mathscr{M}$ (resp. $\mathscr{N}$). to be the standard (resp. $\perp$-standard) Λ-model, such that $[\mathscr{M}] = [\mathscr{N}] = [\mathscr{M}_0]$, $|\perp|_{\mathscr{M}} = |\perp|_{\mathscr{N}} = O$, and with the same interpretation of function symbols as in $\mathscr{M}_0$.

If $\alpha \in |\mathrm{Int}[s^n 0]|_{\mathscr{M}}$, then, by Theorem 3.9, we know that $\alpha \simeq_\beta \lambda f \lambda x (f)^n x$, or possibly, if $n = 1$, $\alpha \simeq_\beta \lambda x\, x$. By definition of O, it follows that $f\alpha \in O$. Since this is true for every $\alpha \in |\mathrm{Int}[s^n 0]|_{\mathscr{M}}$, we have shown that $f \in |\neg\, \mathrm{Int}[s^n 0]|_{\mathscr{M}}$.

Moreover, by hypothesis, we have $\mathbf{c}: \forall X (\neg\neg X \to X) \vdash \theta: \mathrm{Int}[s^n 0]$. By Lemma 3.1 applied to $\mathscr{N}$, we get $\theta \in |\mathrm{Int}[s^n 0]|_{\mathscr{N}}$, since, by Lemma 4.1, we know that $\mathbf{c} \in |\forall X (\neg\neg X \to X)|_{\mathscr{N}}$.

We prove that $\theta \in |\mathrm{Int}^*[s^n 0]|_{\mathscr{M}} = |\forall X \{\forall y (\neg Xy \to \neg Xsy), \neg X0 \to \neg Xs^n 0)|_{\mathscr{M}}$. Let Ξ be any function from $[\mathscr{M}]$ into $\mathscr{P}_S(\Lambda)$, and define $\Xi' : [\mathscr{N}] \to \mathfrak{R}_O$ by $\Xi'(m) = (\Xi(m) \to O) = |\neg\, \Xi(m)|_{\mathscr{M}}$ for every $m \in [\mathscr{M}] = [\mathscr{N}]$. Since $\mathscr{R}_O$ is the truth value set of $\mathscr{N}$, and we have shown that $\theta \in |\forall X \{\forall y (Xy \to Xsy), X0 \to Xs^n 0)|_{\mathscr{N}}$, it follows that $\theta \in |\forall y (\Xi'(y) \to \Xi'(sy)), \Xi'(0) \to \Xi'(s^n 0)|_{\mathscr{N}} = |\forall y (\neg\, \Xi(y) \to \neg\, \Xi(sy)), \neg\, \Xi(0) \to \neg \Xi(s^n 0)|_{\mathscr{M}}$ which is what we need.

Now, since $\vdash T : \neg\, \mathrm{Int}[s^n 0] \to \neg\, \mathrm{Int}^*[s^n 0]$, by Lemma 3.1, we have $T \in |\neg\, \mathrm{Int}[s^n 0] \to \neg\, \mathrm{Int}^*[s^n 0]|_{\mathscr{M}}$. We deduce that $Tf \in |\neg\, \mathrm{Int}^*[s^n 0]|_{\mathscr{M}}$, and so $Tf\theta \in O$. This is exactly the conclusion of the theorem. □

We come now to the main result of this paper, which is the extension of Theorem 3.10, about storage operators, to classical integers. It is a refinement of Theorem 4.3.

Theorem 4.4. *Let $\mathscr{E}$ be a system of equations for integers, and f be any λ-variable. Suppose that $\vdash_{\mathscr{E}} T : \forall x (\neg\, \mathrm{Int}[x] \to \neg\, \mathrm{Int}^*[x])$. Then, for any $n \in \mathbb{N}$, there exists $\alpha_n \simeq_\beta \lambda f \lambda x (f)^n x$ (or, possibly, if $n = 1$, $\alpha_n \simeq_\beta \lambda x\, x$) with the following property: if $\vdash^{\mathbf{c}}_{\mathscr{E}} \theta : \mathrm{Int}[s^n 0]$, then $Tf\theta \succ_{\mathbf{c}} f\alpha$ where α is obtained from α_n by some substitution $(\alpha = \alpha_n[t_1/x_1, \ldots, t_k/x_k])$.*

By Theorem 4.3, we know that $Tf\theta \succ_{\mathbf{c}} f\alpha$, where $\alpha \simeq_\beta \lambda f \lambda x (f)^n x$, α depending on θ. We have to prove that all these λ-terms α are obtained from the same term α_n by some substitution (which will depend on θ).

The proof uses two independent lemmas: the first one (Lemma 4.5) expresses a property of "classical integers", i.e. of λ-terms θ such that $\vdash^{\mathbf{c}} \theta : \mathrm{Int}[s^n 0]$, and the second one (Lemma 4.7) a property of storage operators, mores precisely, of λ-terms T such that $\vdash T : \neg\, \mathrm{Int}[s^n 0] \to \neg\, \mathrm{Int}^*[s^n 0]$.

The integer n will be kept fixed in all what follows. We shall first suppose that $n > 0$. The case $n = 0$, which, in fact, is much simpler, will be treated separately afterwards.

Let $\mathscr{D}$ be a denumerable set of λ-variables, and $\chi : \mathscr{D} \to \{0, \ldots, n\}$ be a map such that $\chi^{-1}(\{k\})$ is infinite for each $k \in \{0, \ldots, n\}$.

Lemma 4.5. *If $\vdash^{\mathbf{c}}_{\mathscr{E}} \theta : \mathrm{Int}[s^n 0]$, $(n > 0)$, then there exists $N > 0$, and distinct variables $g, a, d_0, d_1, \ldots, d_N$, with $d_p \in \mathscr{D}$ for $0 \leqslant p \leqslant N$, such that $\theta g a d_0 \succ_{\mathbf{c}} g\theta_1 d_{r_0}$; $\theta_1 d_1 \succ_{\mathbf{c}} g\theta_2 d_{r_1}$; $\ldots$; $\theta_p d_p \succ_{\mathbf{c}} g\theta_{p+1} d_{r_p}$; $\ldots$; $\theta_N d_N \succ_{\mathbf{c}} a d_{r_N}$. Moreover, $\chi(d_0) = n$, $\chi(d_{r_N}) = 0$, and $\chi(d_{p+1}) = \chi(d_{r_p}) - 1$ for $0 \leqslant p < N$.*

Let us note that θ is a closed term, because $\vdash^{c} \theta : \mathrm{Int}[s^n 0]$; therefore, since no new variable can appear in a head c-reduction, it is clear that d_p is not in $\theta, \theta_1, \ldots, \theta_p$, and that $0 \leqslant r_p \leqslant p$ for $0 \leqslant p \leqslant N$. In particular, $r_0 = 0$.

We have $\chi(d_{r_p}) = 0 \Leftrightarrow p = N$. Indeed, if $p < N$, then $\chi(d_{p+1}) = \chi(d_{r_p}) - 1 \geqslant 0$. We also have $\chi(d_p) = n \Leftrightarrow p = 0$. In fact, $\chi(d_{p+1}) < \chi(d_{r_p}) \leqslant n$.

Proof of Lemma 4.5. Let g, a be two variables, not in $\mathscr{D}$. We define a sequence O_m ($m \in \mathbb{N}$) of subsets of Λ in the following way:

$$O_0 = \{t \in \Lambda;\ t \succ_c ad' \text{ with } d' \in \mathscr{D},\ \chi(d') = 0\}$$

$$O_{m+1} = \{t \in \Lambda;\ \text{there exists } t_1 \in \Lambda,\ d, d' \in \mathscr{D},\ d \text{ not in } t_1,\ \chi(d') = \chi(d) + 1, \text{ such that } t \succ_c gt_1 d' \text{ and } t_1 d \in O_m\}.$$

Let $O = \bigcup_m O_m$. Then O is clearly c-saturated. Let $\mathscr{M}_0$ be a regular model of $\mathscr{E}$, given by Lemma 3.7. We define $\mathscr{N}$ to be the $\perp$-standard Λ-model such that $[\mathscr{N}] = [\mathscr{M}_0]$, $|\perp|_{\mathscr{N}} = O$, and in which the interpretation of function symbols of $\mathscr{L}$ is the same as in $\mathscr{M}_0$.

Let m_k be the kth integer of $\mathscr{N}$. We define a function $\Xi : [\mathscr{N}] \to \mathfrak{R}_O$, in the following way:

– If $k \in \mathbb{N}$, $0 \leqslant k \leqslant n$, then $\Xi(m_k) = \bigcap_{\chi(d) = k} (\{d\} \to O)$.

– If $m \in [\mathscr{N}]$ is not an integer of $\mathscr{N}$, or if $m = m_k$ with $k > n$, then $\Xi(m) = \Lambda$.

By the definition of O (of O_0, in fact), we immediately have: $a \in \Xi(m_0) = |\Xi(0)|_{\mathscr{N}}$. Let us show that $g \in (\Xi(m_k) \to \Xi(m_{k+1}))$ for $0 \leqslant k < n$: indeed, let $t \in \Xi(m_k)$, and $d' \in \mathscr{D}$ be such that $\chi(d') = k + 1$. We have to prove that $gt \in \Xi(m_{k+1})$, or, equivalently, that $gtd' \in O$. Let us choose $d \in \mathscr{D}$ such that $\chi(d) = k$, with d not appearing in t. Then $td \in O$, since $t \in \Xi(m_k)$, so that $td \in O_m$ for some $m \in \mathbb{N}$. Therefore, $gtd' \in O_{m+1}$, by definition of O_{m+1}.

If k is an integer $\geqslant n$, then, clearly, $g \in (\Xi(m_k) \to \Xi(m_{k+1}))$, since $\Xi(m_{k+1}) = \Lambda$. If $m \in [\mathscr{N}]$ is not an integer of $\mathscr{N}$, we have $g \in (\Xi(m) \to \Xi(sm))$, also because $\Xi(sm) = \Lambda$ (we know that, if m is not an integer of $\mathscr{M}_0$, neither is sm, because $\mathscr{M}_0$ is regular). Finally, we have $g \in (\Xi(m) \to \Xi(sm))$ for every $m \in [\mathscr{N}]$, in other words, $g \in |\forall y(\Xi(y) \to \Xi(sy))|_{\mathscr{N}}$.

Now, since O is c-saturated, we have $\mathbf{c} \in |\forall X(\neg\neg X \to X)|_{\mathscr{N}}$, by Lemma 4.1. From Lemma 3.1, it follows that $\theta \in |\mathrm{Int}[s^n 0]|_{\mathscr{N}}$, and therefore $\theta \in |\forall y(\Xi(y) \to \Xi(sy)),\ \Xi(0) \to \Xi(s^n 0)|_{\mathscr{N}}$. It follows that $\theta ga \in |\Xi(s^n 0)|_{\mathscr{N}} = \Xi(m_n)$.

Let us now choose $d_0 \in \mathscr{D}$, not in θga, such that $\chi(d_0) = n$. By definition of $\Xi(m_n)$, we have $\theta gad_0 \in O$, and so $\theta gad_0 \in O_N$ for some $N \in \mathbb{N}$.

If $N = 0$, then $\theta gad_0 \succ_c ad'_0$ with $d'_0 \in \mathscr{D}$, $\chi(d'_0) = 0$. But, since d_0 is the only element of $\mathscr{D}$ in θgad_0, we have necessarily $d_0 = d'_0$, and so $n = \chi(d_0) = 0$. This is impossible, because we have assumed that $n > 0$. Thus, we can set $N = m + 1$ and we have shown that $\theta gad_0 \in O_{m+1}$.

Let σ be an automorphism of $\mathscr{D}$, i.e. a permutation of $\mathscr{D}$ such that $\chi \circ \sigma = \chi$. Then σ can be extended in a natural way into a permutation of Λ (σ is a substitution on

λ-terms). It is easily seen, by induction on m, that O_m is globally invariant under σ, for every $m \in \mathbb{N}$. It follows that, if $t_1 \in \Lambda$, and if there exists $d \in \mathscr{D}$, d not in t_1, $\chi(d) = k$, such that $t_1 d \in O_m$, then $t_1 d \in O_m$ *for every* $d \in \mathscr{D}$, d not in t_1, $\chi(d) = k$. Thus, O_{m+1} can be defined in an equivalent way as follows:

$$t \in O_{m+1} \Leftrightarrow \text{there exists } t_1 \in \Lambda,\ d' \in \mathscr{D},\ \chi(d') > 0, \text{ such that } t \succ_c g t_1 d' \text{ and } t_1 d \in O_m \text{ for every } d \in \mathscr{D},\ d \text{ not in } t_1,\ \chi(d) = \chi(d') - 1\}.$$

It follows that, if $t \in O_{m+1}$, then we have $t \succ_c g t_1 d'_0$; $t_1 d_1 \succ_c g t_2 d'_1$; ...; $t_p d_p \succ_c g t_{p+1} d'_p$; ...; $t_{m+1} d_{m+1} \succ_c a d'_{m+1}$, with $d'_0, d_1, \ldots, d_{m+1} \in \mathscr{D}$, distinct, $\chi(d_{p+1}) = \chi(d'_p) - 1$ for $0 \leqslant p \leqslant m$, and $\chi(d'_{m+1}) = 0$.

Therefore, we have $\theta g a d_0 \succ_c g \theta_1 d'_0$; $\theta_1 d_1 \succ_c g \theta_2 d'_1$; ...; $\theta_p d_p \succ_c g \theta_{p+1} d'_p$; ...; $\theta_N d_N \succ_c a d'_N$, with $d'_0, d_1, \ldots, d_N \in \mathscr{D}$, distinct, $\chi(d_{p+1}) = \chi(d'_p) - 1$ for $0 \leqslant p < N$, and $\chi(d'_N) = 0$.

Since d_0 is the only element of $\mathscr{D}$ which is in $\theta g a d_0$, we see that $d'_0 = d_0$. Finally, since $d_0, \ldots, d_N$ are distinct and not in $\theta g a$, we have necessarily $d'_p = d_{r_p}$, with $0 \leqslant r_p \leqslant p$. □

Let us consider now a denumerable set $\mathscr{B}$ of λ-variables and a map $\psi : \mathscr{B} \to \{0, \ldots, n-1\} \times \Lambda^2$ such that:

(i) $\{b \in \mathscr{B};\ \psi(b) = (k, t, u)\}$ is denumerable, for every $k \in \{0, \ldots, n-1\}$ and $t, u \in \Lambda$,

(ii) if $b \in \mathscr{B}$ and $\psi(b) = (k, t, u)$, then the variable b does not appear in t, u.

It is very easy to build such a map ψ, on any denumerable set $\mathscr{B}$ of λ-variables. Moreover, let us fix a λ-variable $v \notin \mathscr{B}$. Then, we define the notion R-*reduction* on Λ in the following way: it is the least binary relation, denoted by $\succ_R$, which is reflexive and transitive, and such that:

(R0) $(\lambda x.u) t t_1 \ldots t_p \succ_R (u[t/x]) t_1 \ldots t_p$, for every $p \in \mathbb{N}$, and $t, u, t_1, \ldots, t_p \in \Lambda$.

(R1) $vtuv \succ_R tbv$, with $t, u, v \in \Lambda$, $b \in \mathscr{B}$, not appearing in t, u, v and $\psi(b) = (n-1, t, u)$.

(R2) $bv \succ_R tb'v$ if $t, v \in \Lambda$, $b, b' \in \mathscr{B}$, $\psi(b) = (k, t, u)$ with $1 \leqslant k \leqslant n-1$, $\psi(b') = (k-1, t, u)$, b' not appearing in t, v.

(R3) $bv \succ_R uv$ if $u, v \in \Lambda$, $b \in \mathscr{B}$ and $\psi(b) = (0, t, u)$.

A subset $\mathscr{X}$ of Λ will be called R-saturated if $t \in \mathscr{X}$, $t' \succ_R t \Rightarrow t' \in \mathscr{X}$. Every R-saturated set is saturated (by the rule R0).

Lemma 4.6. *If O is an* R-*saturated subset of Λ, then $v \in |\mathrm{Int}^*[s^n 0]|_{\mathscr{M}}$, for any Λ-model $\mathscr{M}$ such that $|\perp|_{\mathscr{M}} = O$.*

Proof. Let $\mathfrak{R}$ be the truth value set of $\mathscr{M}$, and $\Xi : [\mathscr{M}] \to \mathfrak{R}$. The formula $\mathrm{Int}^*[s^n 0]$ is $\forall X \{\forall y(\neg Xy \to \neg Xsy),\ \neg X0 \to \neg Xs^n 0\}$; so, we must show that $v \in |\forall y(\neg \Xi(y) \to \neg \Xi\ (sy)), \neg \Xi(0) \to \neg \Xi(s^n 0)|_{\mathscr{M}}$. Thus, let $t \in |\forall y(\neg \Xi(y) \to \neg \Xi(sy))|_{\mathscr{M}}$, $u \in |\neg \Xi(0)|_{\mathscr{M}}$, $v \in |\Xi(s^n 0)|_{\mathscr{M}}$. We must show that $vtuv \in O$. We show, by induction on k $(0 \leqslant k \leqslant n-1)$, that if $b_k \in \mathscr{B}$, and $\psi(b_k) = (k, t, u)$, then $b_k \in |\neg \Xi(s^k 0)|_{\mathscr{M}}$.

Let $b_0 \in \mathscr{B}$, $\psi(b_0) = (0, t, u)$. If $v_0 \in |\Xi(0)|_{\mathscr{M}}$, then $uv_0 \in O$. Since O is R-saturated, it follows that $b_0 v_0 \in O$. This proves that $b_0 \in |\neg\, \Xi(0)|_{\mathscr{M}}$.

Now, suppose $0 \leqslant k \leqslant n-2$, and let $b_{k+1} \in \mathscr{B}$ such that $\psi(b_{k+1}) = (k+1, t, u)$, and $v_{k+1} \in |\Xi(s^{k+1}0)|_{\mathscr{M}}$. We must show that $b_{k+1} v_{k+1} \in O$. Choose $b_k \in \mathscr{B}$, not appearing in t, v_{k+1}, and such that $\psi(b_k) = (k, t, u)$. By the induction hypothesis, we have $b_k \in |\neg\, \Xi(s^k 0)|_{\mathscr{M}}$ and, therefore, $tb_k \in |\neg \Xi(s^{k+1}0)|_{\mathscr{M}}$. It follows that $tb_k v_{k+1} \in O$. But, since O is R-saturated, we get $b_{k+1} v_{k+1} \in O$, as desired.

Now, we choose $b_{n-1} \in \mathscr{B}$, not appearing in t, u, v, such that $\psi(b_{n-1}) = (n-1, t, u)$. We have just shown that $b_{n-1} \in |\neg\, \Xi(s^{n-1}0)|_{\mathscr{M}}$. Therefore, $tb_{n-1} \in |\neg\, \Xi(s^n 0)|_{\mathscr{M}}$, and $tb_{n-1} v \in O$. Since O is R-saturated, it follows that $vtuv \in O$. □

Lemma 4.7. *Suppose that* $\vdash_{\mathscr{E}} T: \neg\, \mathrm{Int}[s^n 0] \to \neg\, \mathrm{Int}^*[s^n 0]$, *and* f, v *are distinct variables. Then there exists* $\alpha_0 \simeq_\beta \lambda f \lambda x (f)^n x$ (*or, possibly, if* $n = 1$, $\alpha_0 \simeq_\beta \lambda x\, x$) *such that* $Tfv \succ_{\mathrm{R}} f\alpha_0$.

Proof. We set $O = \{t \in \Lambda;\ t \succ_{\mathrm{R}} f\alpha$ for some $\alpha \simeq_\beta \lambda f \lambda x (f)^n x\}$ if $n \neq 1$, and $O = \{t \in \Lambda;\ t \succ_{\mathrm{R}} f\alpha$ for some $\alpha \simeq_\beta \lambda f \lambda x f x$ or $\simeq_\beta \lambda x\, x\}$ if $n = 1$. Then O is R-saturated.

By Lemma 3.7, there exists a regular model $\mathscr{M}_0$ of $\mathscr{E}$. Let $\mathscr{M}$ be the standard Λ-model, such that $[\mathscr{M}] = [\mathscr{M}_0]$, $|\bot|_{\mathscr{M}} = O$, and with the same interpretation of function symbols as in $\mathscr{M}_0$. If $\alpha \in |\mathrm{Int}[s^n 0]|_{\mathscr{M}}$, then, by Theorem 3.9, we know that $\alpha \simeq_\beta \lambda f \lambda x (f)^n x$, or, possibly, if $n = 1$, $\alpha \simeq_\beta \lambda x\, x$. It follows that $f\alpha \in O$. Since this is true for every $\alpha \in |\mathrm{Int}[s^n 0]|_{\mathscr{M}}$, we have shown that $f \in |\neg\, \mathrm{Int}[s^n 0]|_{\mathscr{M}}$.

Now, by Lemma 3.1, we have $T \in |\neg\, \mathrm{Int}[s^n 0] \to \neg\, \mathrm{Int}^*[s^n 0]|_{\mathscr{M}}$, and therefore $Tf \in |\neg\, \mathrm{Int}^*[s^n 0]|_{\mathscr{M}}$. Then, from Lemma 4.6, it follows that $Tfv \in O$. □

Remark. It is clear that the λ-terms α_0 such that $Tfv \succ_{\mathrm{R}} f\alpha_0$ are all identical, up to a permutation of the variables of $\mathscr{B}$. Indeed, there is essentially only one R-reduction of Tfv.

Let us recall that a *substitution* S is, by definition, a map from the set $\mathscr{V}$ of λ-variables into Λ. It is extended, in a natural way, into a map from Λ into Λ, which will be also denoted by S. By the way, it is sufficient to have a map S defined on a subset of $\mathscr{V}$ (we extend it by the identity on the remaining of $\mathscr{V}$).

In particular, the substitution S defined on $\{x_1, \ldots, x_k\}$, such that $S(x_i) = t_i$, is denoted by $[t_1/x_1, \ldots, t_k/x_k]$. If u is any λ-term, $S(u)$ is also denoted by $u[t_1/x_1, \ldots, t_k/x_k]$. If $S' = [t'_1/x'_1, \ldots, t'_l/x'_l]$, $S' \circ S(u)$ is also denoted by $u[t_1/x_1, \ldots, t_k/x_k][t'_1/x'_1, \ldots, t'_l/x'_l]$.

Let us fix, for all the sequel, a λ-term θ such that $\vdash^{\mathrm{c}} \theta: \mathrm{Int}[s^n 0]$; then θ is closed. By Lemma 4.5, we associate with θ an integer $N > 0$, variables $d_0, \ldots, d_N \in \mathscr{D}$, and λ-terms $\theta_1, \ldots, \theta_N$.

Let p be an integer, $p \in \{1, \ldots, N\}$, and t, u be two λ-terms; then, by definition, a *(p, t, u)-term* is a λ-term of the form $\theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$, such that $Tfv \succ_{\mathrm{R}} vtuv_0$ and $Tfv \succ_{\mathrm{R}} b_i v_i$ for each $i \in \{1, \ldots, p-1\}$, with $b_i \in \mathscr{B}$ and $\psi(b_i) = (\chi(d_i), t, u)$.

Lemma 4.8. *There exists a substitution S and a map $U:\mathscr{B} \to \Lambda$ such that:*

(i) $S(x) = x$ *for each variable* $x \notin \mathscr{B}$, $x \neq v$; $S(v) = S(\theta) = \theta$.

(ii) $S(b) = S(U(b))$ *for every* $b \in \mathscr{B}$.

(iii) *For every* $b \in \mathscr{B}$, *either* $S(b) = U(b) = b$, *or* $U(b)$ *is a* (p,t,u)*-term and* $\psi(b) = (\chi(d_p), t, u)$.

(iv) *For each* (p,t,u)*-term* η, *there exists infinitely many* $b \in \mathscr{B}$ *such that* $U(b) = \eta$.

Proof. Let $\eta_0, \eta_1, \ldots, \eta_k, \ldots$ be an enumeration of all (p,t,u)-terms, for every $p \in \{1, \ldots, N\}$ and $t, u \in \Lambda$, each (p,t,u)-term appearing infinitely many times. We define $b_0, b_1, \ldots, b_k, \ldots \in \mathscr{B}$, by induction on k, in the following way: we have $\eta_k = \theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$; then b_k is defined as the first element of $\mathscr{B}$ (for any fixed enumeration of $\mathscr{B}$) which is unequal to $b_0, \ldots, b_{k-1}$, does not appear in $\eta_0, \ldots, \eta_k$, and is such that $\psi(b_k) = (\chi(d_p), t, u)$.

Then, we define $U:\mathscr{B} \to \Lambda$, by setting $U(b_k) = \eta_k$ for every $k \in \mathbb{N}$, and $U(b) = b$ if $b \neq b_k$ for all $k \in \mathbb{N}$. Finally, we define the substitution $S:\mathscr{V} \to \Lambda$, by setting $S(v) = \theta$ (note that $\theta = S(\theta)$, because θ is closed), $S(x) = x$ for every variable $x \neq v$, $b_0, \ldots, b_k, \ldots$; $S(b_k)$ is defined by induction on k, by $S(b_k) = S(U(b_k))$ (note that the only variables b_i which appear in $U(b_k)$ are $b_0, \ldots, b_{k-1}$). Then, the conditions (i)–(iv) of the lemma are trivially satisfied. □

We now choose, once and for all, a substitution S and a map $U:\mathscr{B} \to \Lambda$, which satisfy the conditions (i)–(iv) of Lemma 4.8.

Lemma 4.9. *Let ξ be a λ-term which does not begin by f, and $\tau = S(\xi)$, such that $Tfv \succ_R \xi$, $Tf\theta \succ_c \tau$. Then, there exists ξ' and $\tau' = S(\xi')$, such that $Tfv \succ_R \xi'$, $\tau \succ_c \tau'$, and, either $\tau \neq \tau'$, or $\xi \succ_R \xi'$ and $\xi \neq \xi'$.*

Proof. Since $Tfv \succ_R \xi$ and, by Lemma 4.7, we know that $Tfv \succ_R f\alpha_0$, it follows that $\xi \succ_R f\alpha_0$. Therefore, ξ does not begin by a λ. If ξ is not in head normal form, we get ξ' from ξ by one step of head reduction. Then, clearly, we have $\xi \succ_R \xi'$ and $\xi \neq \xi'$ (and also, by the way, $\tau \succ_c \tau'$ and $\tau \neq \tau'$).

Remark. Since we know, by Theorem 4.3, that the head **c**-reduction of τ is finite, if τ' is obtained from τ by a finite number (> 0) of steps of head **c**-reduction, then, we have necessarily $\tau \neq \tau'$.

Therefore we shall now assume that ξ is in head normal form. Now, since $\xi \succ_R f\alpha_0$ and $\xi \neq f\alpha_0$ (ξ does not begin by f), by definition of R-reduction, there are only two possibilities: (1) $\xi = vtuv$, (2) $\xi = bv$ with $b \in \mathscr{B}$. We now examine each of these cases:

(1) If $\xi = vtuv$, then $\tau = S(vtuv) = S(\theta tuv) = S((\theta g a d_0)[t/g, u/a, v/d_0])$. Now, by Lemma 4.5, $\theta g a d_0 \succ_c g\theta_1 d_{r_0} = g\theta_1 d_0$ since $r_0 = 0$. Therefore, we have:

$$\tau \succ_c S((g\theta_1 d_0)[t/g, u/a, v/d_0]) = S(t\eta v), \text{ with } \eta = \theta_1[t/g, u/a, v/d_0].$$

But η is a $(1,t,u)$-term since $Tfv \succ_R vtuv = \xi$. Therefore, by Lemma 4.8(iv), there exists some $b \in \mathscr{B}$, b not in t,u,v, such that $U(b) = \eta$; by Lemma 4.8(iii), we have $\psi(b) = (\chi(d_1), t, u) = (n-1, t, u)$ since $\chi(d_1) = \chi(d_{r_0}) - 1 = \chi(d_0) - 1 = n - 1$ (Lemma 4.5). Moreover, $S(t\eta v) = S(tbv)$, since $S(b) = S(U(b))$, so that $\tau \succ_c S(tbv)$.

But $\xi = vtuv$ and therefore, we have $\xi \succ_R tbv$. Now, let $\xi' = tbv$, $\tau' = S(\xi')$; we get $\tau \succ_c \tau'$ and $\xi \succ_R \xi'$. Moreover, we have $\xi \neq \xi'$, i.e. $vtuv \neq tbv$, because $vt \neq t$.

(2) If $\xi = bv$ with $b \in \mathscr{B}$, then we have $\tau = S(bv)$; but we cannot have $S(b) = b$, because $Tf\theta \succ_c \tau$, and b is not in $Tf\theta$, and so, cannot appear in τ. Therefore, by Lemma 4.8(iii), $U(b) = \theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$, and $\psi(b) = (\chi(d_p), t, u)$. It follows that

$$\tau = S((U(b))v) = S(\theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]v)$$

$$= S((\theta_p d_p)[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}, v/d_p])$$

(because d_p does not appear in θ_p).

In a natural way, we set $v_p = v$, and we consider the following two cases:

(i) If $p < N$, then, by Lemma 4.5, $\theta_p d_p \succ_c g\theta_{p+1} d_{r_p}$, so that

$$\tau \succ_c S(g\theta_{p+1} d_{r_p})[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}, v_p/d_p]) = S(t\eta v_{r_p})$$

with

$$\eta = \theta_{p+1}[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}, v_p/d_p].$$

Let us check that η is a $(p+1,t,u)$-term: indeed, since $\theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$ is a (p,t,u)-term, we have $Tfv \succ_R vtuv_0$ and $Tfv \succ_R b_i v_i$ for $1 \leqslant i \leqslant p-1$, with $\psi(b_i) = (\chi(d_i), t, u)$. It remains to check this equality when $i = p$ (remember that $v_p = v$). But, by hypothesis, $Tfv \succ_R \xi = bv$, and $\psi(b) = (\chi(d_p), t, u)$, which is exactly what we want. By Lemma 4.8(iv), it follows that there exists $b'' \in \mathscr{B}$, which does not appear in t,u,v,v_{r_p}, such that $U(b'') = \eta$, and therefore, by Lemma 4.8(iii), $\psi(b'') = (\chi(d_{p+1}), t, u)$. Then $\tau \succ_c S(tU(b'')v_{r_p}) = S(tb''v_{r_p})$, since $S(b'') = S(U(b''))$.

If we set $\xi' = tb''v_{r_p}$, $\tau' = S(\xi')$, we have $\tau \succ_c \tau'$. There are now three cases:

- If $0 < r_p < p$, since $\theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$ is a (p,t,u)-term, we have $Tfv \succ_R b'v_{r_p}$ with $\psi(b') = (\chi(d_{r_p}), t, u) = (\chi(d_{p+1}) + 1, t, u)$. It follows, by definition of R-reduction, that $Tfv \succ_R tb''v_{r_p}$, and therefore, $Tfv \succ_R \xi'$.
- If $r_p = 0$, since $\theta_p[t/g, u/a, v_0/d_0, \ldots, v_{p-1}/d_{p-1}]$ is a (p,t,u)-term, we have $Tfv \succ_R vtuv_0$, and $\psi(b'') = (\chi(d_{p+1}), t, u) = (\chi(d_{r_p}) - 1, t, u) = (\chi(d_0) - 1, t, u) = (n-1, t, u)$. Therefore, by definition of R-reduction, we have $Tfv \succ_R tb''v_0 = tb''v_{r_p}$, i.e. $Tfv \succ_R \xi'$.
- If $r_p = p$, then $\xi = bv = bv_p = bv_{r_p}$; but $\psi(b) = (\chi(d_p), t, u) = (\chi(d_{r_p}), t, u) = (\chi(d_{p+1}) + 1, t, u)$ by Lemma 4.5. It follows that $bv \succ_R tb''v$, and so $\xi \succ_R \xi'$ (therefore $Tfv \succ_R \xi'$). Now, suppose that $\tau = \tau'$. Then, the head c-reduction: $\theta_p d_p \succ_c g\theta_{p+1} d_{r_p}$ must be an identity; in other words, $\theta_p d_p = g\theta_{p+1} d_{r_p}$, and therefore

$p = r_p$. We have already seen that, in this case, we have $\xi \succ_R \xi'$; moreover, $\xi \neq \xi'$, i.e. $bv \neq tb''v_{r_p}$, because $b \neq tb''$.

(ii) If $p = N$, then $\theta_p d_p = \theta_N d_N \succ_c a d_{r_N}$, by Lemma 4.5, and thus: $\tau \succ_c S((ad_{r_N})[t/g, u/a, v_0/d_0, \ldots, v_{N-1}/d_{N-1}, v/d_N]) = S(uv_{r_N})$ (remember that we set $v_N = v$). Therefore, $\tau \succ_c \tau'$, if we set $\xi' = uv_{r_N}$, and $\tau' = S(\xi')$. Now, we have $r_N \leqslant N$, and $\chi(d_{r_N}) = 0$. We cannot have $r_N = 0$, because this would imply $0 = \chi(d_{r_N}) = \chi(d_0) = n$, and we assumed $n > 0$. Thus, there are two cases:

– If $0 < r_N < N$, since $\theta_N[t/g, u/a, v_0/d_0, \ldots, v_{N-1}/d_{N-1}]$ is a (N, t, u)-term, we see that $Tfv \succ_R b' v_{r_N}$, with $\psi(b') = (\chi(d_{r_N}), t, u) = (0, t, u)$. Therefore, $Tfv \succ_R uv_{r_N}$.

– If $r_N = N$, then $v_{r_N} = v_N = v$. But $d_p = d_N = d_{r_N}$, so that $\chi(d_p) = 0$, and $\psi(b) = (0, t, u)$. Therefore, $bv \succ_R uv$, in other words $\xi \succ_R \xi'$ (and, therefore, $Tfv \succ_R \xi'$).

Thus, in both cases, we have $Tfv \succ_R \xi'$. If $\tau = \tau'$, the head c-reduction: $\theta_N d_N \succ_c a d_{r_N}$ must be an identity, in other words $\theta_N d_N = a d_{r_N}$; therefore, $r_N = N$. We have already seen that, in this case, we have $\xi \succ_R \xi'$. Moreover, we have $\xi \neq \xi'$, i.e. $bv \neq uv_{r_N}$: indeed, since $\psi(b) = (\chi(d_p), t, u)$, we have $b \neq u$, because b does not appear in u, by definition of ψ. □

Proof of Theorem 4.4. Let $\tau_0 = Tf\theta$; then $\tau_0 = S(\xi_0)$, with $\xi_0 = Tfv$. By Lemma 4.9 applied to ξ_0, we obtain a sequence $(\xi_0, \tau_0), (\xi_1, \tau_1), \ldots, (\xi_i, \tau_i), \ldots$, of pairs of λ-terms, such that $\tau_i = S(\xi_i)$, $Tfv \succ_R \xi_i$, $\tau_i \succ_c \tau_{i+1}$, and, if $\tau_i = \tau_{i+1}$, then $\xi_i \succ_R \xi_{i+1}$ and $\xi_i \neq \xi_{i+1}$. This sequence is necessarily finite: indeed, the head c-reduction of $Tf\theta$ is finite, by Theorem 4.3, and all R-reductions of Tfv are finite and of the same length L, by Lemma 4.7 (any λ-term has, essentially, only one R-reduction). It follows that, if $\tau_i = \tau_{i+1} = \cdots = \tau_j$, then the sequence $\xi_i, \xi_{i+1}, \ldots, \xi_j$ is a part of the R-reduction of Tfv, and is therefore of length $\leqslant L$. It follows that the sequence $(\xi_0, \tau_0), \ldots, (\xi_i, \tau_i), \ldots$ has a finite length K. By Lemma 4.9, this means that ξ_K begins by f. But $Tfv \succ_R \xi_K$, and, by Lemma 4.7, it follows that $\xi_k = f\alpha_0$. Thus, $\tau_K = S(f\alpha_0) = (f)S(\alpha_0)$. But $Tf\theta = \tau_0 \succ_c \tau_K$, and, finally, $Tf\theta \succ_c (f)S(\alpha_0)$. □

We consider now the case $n = 0$. In this particular case, the R-reduction is the least binary relation, still denoted by $\succ_R$, which is reflexive and transitive, and such that:

(R0) $(\lambda x\, u)tt_1 \ldots t_p \succ_R (u[t/x])t_1 \ldots t_p$, for every $p \in \mathbb{N}$, and $t, u, t_1, \ldots, t_p \in \Lambda$.

(R1) $vtuv \succ_R uv$, for every $t, u, v \in \Lambda$.

The sets $\mathscr{B}$, $\mathscr{D}$ of λ-variables and the maps ψ, χ are useless.

Lemma 4.7 is still valid for $n = 0$, with the same proof. Lemma 4.5 is replaced by the following.

Lemma 4.10. *If* $\vdash_{\mathscr{E}}^{c} \theta$: Int[0] *then* $\theta gad \succ_c ad$, *for any distinct* λ*-variables* g, a, d.

Proof. Let $O = \{t \in \Lambda; t \succ_c ad\}$. Then O is clearly a c-saturated subset of Λ. Let $\mathscr{M}_0$ be a regular model of $\mathscr{E}$, given by Lemma 3.7. We define $\mathscr{N}$ to be the $\perp$-standard

Λ-model such that $[\mathcal{N}] = [\mathcal{M}_0]$, $|\bot|_{\mathcal{N}} = O$, and in which the interpretation of function symbols of $\mathcal{L}$ is the same as in $\mathcal{M}_0$.

Let m_0 be the interpretation, in $\mathcal{N}$, of the constant symbol 0 (m_0 is "the integer 0" of $\mathcal{N}$). We define a function $\Xi : [\mathcal{N}] \to \mathcal{R}_O$, in the following way: $\Xi(m_0) = (\{d\} \to O)$; $\Xi(m) = \Lambda$, if $m \in [\mathcal{N}]$, $m \neq m_0$.

By the definition of O, we immediately have $a \in \Xi(m_0) = |\Xi(0)|_{\mathcal{N}}$. It is also clear that $g \in (\Xi(m) \to \Xi(sm))$ for every $m \in [\mathcal{N}]$, because $\Xi(sm) = \Lambda$ (we know that $sm \neq m_0$, since $\mathcal{M}_0$ is regular). Therefore, $g \in |\forall y(\Xi(y) \to \Xi(sy))|_{\mathcal{N}}$. Now, since O is **c**-saturated, we have $\mathbf{c} \in |\forall X(\neg\neg X \to X)|_{\mathcal{N}}$, by Lemma 4.1. From Lemma 3.1, we have $\theta \in |\mathrm{Int}[0]|_{\mathcal{N}}$, and therefore $\theta \in |\forall y(\Xi(y) \to \Xi(sy)), \Xi(0) \to \Xi(0)|_{\mathcal{N}}$. It follows that $\theta g a \in |\Xi(0)|_{\mathcal{N}} = \Xi(m_0)$. Thus, $\theta g a d \in O$, and it is exactly what we want. □

Lemma 4.11. *If* $Tfv \succ_{\mathbf{R}} \tau$, *then* $Tf\theta \succ_{\mathbf{c}} \tau[\theta/v]$.

Proof (*by induction on the length of the R-reduction* $Tfv \succ_{\mathbf{R}} \tau$). The result is trivial if $\tau = Tfv$. Now, let τ' be obtained from τ by one step of R-reduction. If this step is given by the rule R0, we have $\tau \succ \tau'$; therefore, $\tau[\theta/v] \succ \tau'[\theta/v]$, and, by induction hypothesis, we get $Tf\theta \succ_{\mathbf{c}} \tau'[\theta/v]$.

If this step is given by the rule R1, then we have $\tau = vtuv$, and $\tau' = uv$. Then $\tau[\theta/v] = (\theta tuv)[\theta/v]$ (remind that θ is closed) $= (\theta g a d)[t/g, u/a, v/d][\theta/v]$.

Now, by Lemma 4.10, $\theta g a d \succ_{\mathbf{c}} ad$, and therefore

$$\tau[\theta/v] \succ_{\mathbf{c}} (ad)[t/g, u/a, v/d][\theta/v] = (uv)[\theta/v] = \tau'[\theta/v].$$

Then, from the induction hypothesis, it follows that $Tf\theta \succ_{\mathbf{c}} \tau'[\theta/v]$. □

Proof of Theorem 4.4. The case $n = 0$ of Theorem 4.4 follows easily: by Lemma 4.7, we have $Tfv \succ_{\mathbf{R}} f\alpha_0$, with $\alpha_0 \simeq_\beta \lambda x \lambda y\, y$. Then, by Lemma 4.11, we get $Tf\theta \succ_{\mathbf{c}} f\alpha_0[\theta/v]$. □

References

[1] H. Barendregt, The Lambda Calculus (North-Holland, Amsterdam, 1984).

[2] M. Felleisen, The calculi of λ_v-CS conversion: a syntactic theory of control and state in imperative higher order programming, Ph.D. Thesis, Indiana Univ. (1987).

[3] J.Y. Girard, Une extension de l'interprétation de Gödel à l'analyse, in: Proc. 2nd Scand. Logic Symp. (North-Holland, Amsterdam, 1971) 63–92.

[4] J.Y. Girard, A new constructive logic: classical logic, preprint (1991).

[5] J.Y. Girard, On the unity of logic, Ann. Pure Appl. Logic 59 (1993) 201–217.

[6] T. Griffin, A formulae-as-type notion of control, in: Conf. Record of the 17th ACM Symp. on Principles of Programming Languages (1990).

[7] J.L. Krivine, Lambda-calcul, types et modèles (Masson, Paris, 1990).

[8] J.L. Krivine, Opérateurs de mise en mémoire et traduction de Gödel, Arch. Math. Logic 30 (1990) 241–267.

[9] J.L. Krivine and M. Parigot, Programming with proofs, J. Inform. Process. Cybernet. EIK 26 (3) (1990) 149–167.

[10] D. Leivant, Reasoning about functional programs and complexity classes associated with type disciplines, in: 24th Ann. Symp. on Found. of Comp. Sci. (1983) 460–469.
[11] C. Murthy, Extracting constructive content from classical proofs, Ph.D. Thesis, Cornell Univ. (1990).
[12] M. Parigot, Free deduction: an analysis of computations in classical logic, in: Proc. of Logic Progr. and Automatic Reasoning, St. Petersbourg, Lecture Notes in Computer Science 592, (Springer, Berlin, 1991) 361–380.
[13] M. Parigot, $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction, in: Proc. of Logic Progr. and Automatic Reasoning, St Petersbourg, Lecture Notes in Computer Science 624, (Springer, Berlin, 1992) 190–201.
[14] M. Parigot, Séminaire de Logique, Université Paris VII (1992).
[15] G. Plotkin, Call-by-name, call-by-value, and the λ-calculus, Theoret. Comp. Sci. 1 (1975) 125–159.