INFORMATION AND COMPUTATION **125**, 103–117 (1996)
ARTICLE NO. 0025

FRANCO BARBANERA AND STEFANO BERARDI

Dipartimento di Informatica, Università di Torino, Corso Svizzera, 185, 10149 Torino, Italy
E-mail: {barba,stefano}@di.unito.it

We introduce a λ-calculus with symmetric reduction rules and "classical" types, i.e., types corresponding to formulas of classical propositional logic. The strong normalization property is proved to hold for such a calculus, as well as for its extension to a system equivalent to Peano arithmetic. A theorem on the shape of terms in normal form is also proved, making it possible to get recursive functions out of proofs of Π^0_2 formulas, i.e., those corresponding to program specifications.

1. INTRODUCTION

The possibility of extracting recursive functions out of intuitionistic proofs of formulas expressing their specifications, i.e., in general, the feature of *effectiveness* of constructive mathematics, has had a leading role in the development of computer science. This role, not just limited to foundational aspects, has been played in a wide spectrum of research fields, with the aim of supporting the *working* computer scientist. In particular, the correspondence between logical constructive systems and several λ-calculi, known as the Curry–Howard analogy, has been used to construct prototypes of systems for the design and development of provably correct programs [Con 86, NPS 90, PN 90].

Classical logics were always left out of the investigations on the relationships between logics, type-theories, and programming languages. This, however, was not caused by its alleged lack of effective features. Indeed, as far as the part of logics relevant for computer science is concerned, this is absolutely not the case. Quite old and well-known theoretic results (for instance, [Kre 58]) made sure that it is possible to get, out of a *classical* proof of $\forall x\, \exists y.P(x, y)$ (with P decidable), a *recursive* function f such that, for any x, $P(x, f(x))$ holds. What was still preventing classical logics to have a more relevant role in computer science, was instead the lack of *clear* and *practical* methods to extract their constructive contents from classical proofs, and systems that helped to understand their constructive features.

In the recent years, many efforts have been made in this direction, among which has to be mentioned the interpretation of classical logics into calculi with continuations [Gri 90, Mur 90], originating from Friedman's A-translation [Fri 78]. An investigation of Prawitz's set of reductions for classical logic [Pra 65, Pra 81] was the starting point for a method for extracting the constructive content from classical proofs devised in [BB 91], which has an interpretation in terms of a valuation semantics [BB 92]. All the above-mentioned methods have natural deduction versions of classical logics at their roots. On the sequent calculus-side, research efforts have led to the $\lambda\mu$-calculus of Parigot [Par 92] and to Coquand's *game-theoretical* interpretation [Coq 92].

These preliminary results, at least those on the natural deduction-side, share a common problem: that of a complex syntax. This, of course, represents a serious obstacle to a neat and full understanding of classical logics from a computational point of view.

Our main aim in the present paper is then to define a system for classical program extraction which is simple enough. Our starting point, in Section 2, will be to define a classical simply typed λ-calculus ($\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$), i.e., a λ-calculus which is in a Formulas-as-Types correspondence to propositional classical logics. In this system, negation is not a primitive connective, and we manage to identify a type (formula) A with its double negation $A^{\perp\perp}$. This enables us to get a system where we have a *symmetric* application, such that either component of an application can be looked at virtually indifferently as function or argument. Because of this symmetry, all the reductions of the calculus have a dual version. It is relevant to stress that the reductions we define are simple and natural, and, differently from what was done in other systems for classical program extraction, no ad hoc reduction is introduced. Our system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ is then proved in Section 5 to be strongly normalizable using a non-trivial version of Tait and Girard's computability method: symmetric candidates. In Section 3 system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ is extended with first order features in order to obtain a system corresponding to Peano arithmetic ($\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$), still strongly normalizable, as proved in the Appendix. Moreover, a Shape of Normal Forms Theorem, proved in Section 4, makes it possible to extract the constructive contents of terms corresponding to proofs of formulas of the form $\forall x\, \exists y.P(x, y)$, with P decidable.

0890-5401/96 $18.00

2. $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$: A SYMMETRIC SIMPLY TYPED CLASSICAL λ-CALCULUS

In this section, we will introduce the system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$. In such a system, types correspond to formulas and terms to proofs of propositional classical logic. We shall then often use indifferently the words *type*, *formula*, and *proposition* as well as *term* and *proof*.

The basis for building the types of our system consists of two sets of base types: $\mathscr{A} = \{a, b, ...\}$ (atomic types) and $\mathscr{A}^\perp = \{a^\perp, b^\perp, ...\}$ (negated atomic types). These two sets are used to build, as shown below, *m-types* and *types*.

DEFINITION 2.1. (i) The set of m-types is defined by the grammar:

$$A ::= \alpha \mid \alpha^\perp \mid A \wedge A \mid A \vee A,$$

where α ranges over $\mathscr{A}$ and $\alpha^\perp$ over $\mathscr{A}^\perp$.

(ii) The set of types is defined by the grammar

$$C ::= A \mid \perp.$$

We need to define the m-types first since we wish to have a calculus where formulas do not contain the absurdity proposition as a proper subtype. Such a choice is motivated by technical reasons, which will be made clear in Section 5. It is easy to check, however, that this is no restriction at all (a formula $A \wedge \perp$ can always be identified with $\perp$, and $A \vee \perp$ with A). It is also no restriction to prevent $\perp$ being used as assumption in a derivation, as we do—that is, in the calculus, not to have variables of type $\perp$.

In the following we shall denote m-types by A, B, A_1, A_2, ..., while types will be denoted by C, D, C_1, C_2, ...

By having a set of atomic types and a set of negated atomic types, it is easy to see that we have a propositional calculus where negation is neither primitive nor defined in terms of $\perp$.

DEFINITION 2.2. We define the negation $A^\perp$ of an m-type A as follows:

1. $(\alpha)^\perp = \alpha^\perp$
2. $(\alpha^\perp)^\perp = \alpha$
3. $(A \wedge B)^\perp = A^\perp \vee B^\perp$
4. $(A \vee B)^\perp = A^\perp \wedge B^\perp$.

We then get a calculus with involutive negation.

LEMMA 2.3.

$$A^{\perp\perp} = A.$$

Proof. By induction on A, using Definition 2.2. ∎

DEFINITION 2.4. ($\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$-Rules). The terms of the system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ are defined by the following rules:

$$(\mathrm{var})\ x^A : A$$

$$(\langle\ ,\ \rangle)\ \frac{P_1 : A_1 \quad P_2 : A_2}{\langle P_1, P_2\rangle : A_1 \wedge A_2}$$

$$(\sigma_i)\ \frac{P_i : A_i}{\sigma_i^{A_1, A_2}(P_i) : A_1 \vee A_2} \quad (i = 1, 2)$$

$$\begin{array}{c}[x^A : A] \\ \vdots \\ P : \perp \end{array}$$

$$(\lambda)\ \frac{}{\lambda x^A . P : A^\perp}$$

$$(\star)\ \frac{P_1 : A^\perp \quad P_2 : A}{(P_1 \star P_2) : \perp}$$

In the following the type of a term will often be denoted by superscripts while the superscripts A_1, A_2 in terms such as $\sigma_i^{A_1, A_2}(P_i)$ will often be omitted.

Remark 2.5. The propositional classical logic associated with our system is complete. Rules and connectives not given above can be derived as is usual in classical logic. We show below the (type part) of the derivation of the conjunction–elimination rule and the implication–elimination rule.

$$\frac{A_1 \wedge A_2 \equiv (A_1^\perp \vee A_2^\perp)^\perp \quad \dfrac{[A_i^\perp]}{A_1^\perp \vee A_2^\perp}}{\dfrac{\perp}{A_i}}$$

$$A \to B =_{\mathrm{Def}} A^\perp \vee B$$

$$\frac{A \to B \equiv A^\perp \vee B \quad \dfrac{A \quad [B^\perp]}{(A^\perp \vee B)^\perp \equiv A \wedge B^\perp}}{\dfrac{\perp}{B}}$$

We call the operator "★" *symmetric application* since, given the terms $P^{A^\perp}$ and Q^A, both $P^{A^\perp} \star Q^A$ and $Q^A \star P^{A^\perp}$ are correct $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$-terms. This symmetry is reflected by the (pairwise dual, except for rule (Triv)) reductions rules defined below.

DEFINITION 2.6 ($\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$-Reduction Rules).

$$\begin{cases}(\beta) & \lambda x.P \star Q \to_{\beta} P[Q/x] \\ (\beta^{\perp}) & Q \star \lambda x.P \to_{\beta^{\perp}} P[Q/x]\end{cases}$$

$$\begin{cases}(\eta) & \lambda x.(P \star x) \to_{\eta} P \quad \text{if} \quad x \notin FV(P) \\ (\eta^{\perp}) & \lambda x.(x \star P) \to_{\eta^{\perp}} P \quad \text{if} \quad x \notin FV(P)\end{cases}$$

$$\begin{cases}(\pi) & \langle P_1, P_2\rangle \star \sigma_i(Q_i) \to_{\pi} P_i \star Q_i \qquad (i=1,2) \\ (\pi^{\perp}) & \sigma_i(Q_i) \star \langle P_1, P_2\rangle \to_{\pi^{\perp}} Q_i \star P_i \qquad (i=1,\ 2)\end{cases}$$

(Triv) $E[P] \to_{\mathrm{Triv}} P$ if $E[\,\_\,]$ is a context with type $\perp$ and $\neq [\,\_\,]$, P has type $\perp$ and $E[\,\_\,]$ does not bind any free variables in P.

In the following $\to_1$ will denote the union of the reductions defined above. $\to$ will denote the reflexive, transitive closure of $\to_1$.

Remark 2.7. Rule (π), which is inspired by the sequent calculus, looks as follows in terms of derivations:

$$\dfrac{\dfrac{A_1 \quad A_2}{A_1 \wedge A_2} \quad \dfrac{A_i^{\perp}}{A_1^{\perp} \vee A_2^{\perp}}}{\perp} \rightsquigarrow \dfrac{A_i \quad A_i^{\perp}}{\perp}$$

This rule in our system plays the rôle of the usual reduction rule for removing, in natural deduction, an introduction of a conjuction followed by its elimination. As we have seen in Remark 2.5, the elimination of conjunction can be derived in our system. It is then easy to see that, by using that derived rule, the usual reduction rule can be defined in terms of one (π)-reduction and one (η)-reduction.

Notice that, by defining the (π) rule as we have done above, we manage easily to *dualize* it.

DEFINITION 2.8. Let k be an integer and P a term.

(i) k is a *bound* for P if the reduction tree of P has finite height smaller or equal to k.

(ii) P *strongly normalizes* if it has a bound.

Thus a term strongly normalizes if and only if its reduction tree is finite.

The above definition has been chosen since it is intuitionistically stronger than the usual "each reduction sequence out of P is finite" (classically, they are equivalent through König's Lemma).

Notation. The following notations will be used:

$$\mathrm{Var}_C = \{\text{variables of type } C\}$$
$$\mathrm{Term}_C = \{\text{terms of type } C\}$$
$$\mathrm{SN}_C = \{P \in \mathrm{Term}_C \mid P \text{ strongly normalizes}\}.$$

One of the main properties enjoyed by the system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ that will be essential for its applications is that of strong normalization.

THEOREM 2.9 (Strong Normalization for $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$). *Let C be a type. Then*

$$\mathrm{Term}_C = \mathrm{SN}_C.$$

The proof of this theorem will be the argument of Section 5.

It is no surprise that $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ does not have the Church–Rosser property, even without taking into account rule (Triv). The following is a simple example of the non-confluence of $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ by Schivalocchi [Schi 95]:

$$\lambda x^{\alpha^{\perp} \wedge \alpha}.(\lambda y^{\alpha}.x \star \sigma_1^{\alpha,\alpha^{\perp}}(y)) \star (\lambda z^{\alpha^{\perp}}.x \star \sigma_2^{\alpha,\alpha^{\perp}}(z)).$$

The above closed term can be reduced either by rule (β) or by rule $(\beta^{\perp})$, yielding the two following distinct closed normal forms:

$$\lambda x^{\alpha^{\perp} \wedge \alpha}.(x \star \sigma_1^{\alpha,\alpha^{\perp}}(\lambda z^{\alpha^{\perp}}.x \star \sigma_2^{\alpha,\alpha^{\perp}}(z)))$$

$$\lambda x^{\alpha^{\perp} \wedge \alpha}.(x \star \sigma_2^{\alpha,\alpha^{\perp}}(\lambda y^{\alpha}.x \star \sigma_1^{\alpha,\alpha^{\perp}}(y))).$$

More complex examples of non-confluence are given and discussed in [Schi 95], together with concrete examples of proofs in $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ (the system to be defined in the next section).

3. $\lambda^{\mathrm{SYM}}_{\mathbf{PA}}$: A CALCULUS FOR PEANO ARITHMETIC

In Section 2 we have defined a calculus based on a version of propositional classical logic. This logic, however, even when considered as a possible basis for a (simply) typed λ-calculus with symmetric application and *classical* types, is too poor for our present purposes, i.e., the investigation of the computational content of classical reasoning. In fact, we defined it only as a starting point.

We want to deal with a logic in which it is possible to express and prove meaningful specifications of programs. We choose Peano arithmetic. In the following we shall define a calculus that corresponds to a natural deduction version of Peano arithmetic and is based on system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$. We shall call this calculus $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$.

We begin by defining Peano arithmetic terms (PA-terms) in our context. They denote integers and (possibly higher order) functions, and are built out of numerical and function variables, the constant 0, the successor function **s**, primitive recursion, abstraction, and application.

DEFINITION 3.1 (PA-Terms). (i) PA-terms are all the terms possible to build using the following *term-formation*

rules: Let g be a numerical or function variable, and G, G_1, G_2 types built out of the type constant Int using the arrow constructor.

$$g^G: G \qquad 0: \mathrm{Int}$$

$$[\,g^{G_1}: G_1\,]$$

$$\vdots$$

$$\frac{p: G_2}{\lambda g^{G_1}.p: G_1 \to G_2} \qquad \frac{p_1: G_1 \to G_2 \quad p_2: G_1}{(p_1\, p_2): G_2}$$

$$\frac{u: \mathrm{Int}}{\mathsf{s}u: \mathrm{Int}} \qquad \frac{u: \mathrm{Int} \quad p: G \quad f: \mathrm{Int} \to G \to G}{\mathsf{Rec}(u, p, f): G}$$

(ii) On PA-terms the following well-known notions of reductions are defined:

$$(\beta_{\mathrm{PA}}) \quad (\lambda g.p)\, q \to_{\beta \mathrm{PA}} p[q/g]$$

$$(\mathsf{Rec}_0) \quad \mathsf{Rec}(0, p, f) \to_{\mathrm{Rec}_0} p$$

$$(\mathsf{Rec}_\mathsf{s}) \quad \mathsf{Rec}(\mathsf{s}u, p, f) \to_{\mathrm{Rec}_\mathsf{s}} f(u)\, \mathsf{Rec}(u, p, f)$$

We denote by $\to_{1\mathrm{PA}}$ the union of all the above reductions, and by $\to_{\mathrm{PA}}$ its reflexive and transitive closure.

(iii) We denote by $\simeq$ the least congruence obtained from $\to_{\mathrm{PA}}$.

In what follows, numerical variables (i.e., of type Int) will be denoted by $n, m, \ldots$, while generic PA-terms will be denoted by $u, v, t, \ldots$.

LEMMA 3.2. [Tait 67]. *All PA-terms are strongly normalizable.*

Given a PA-term t, its normal form will be denoted by $\mathsf{nf}(t)$. t will be said to be a *numeral* if it is 0 or $\mathsf{s}^{k+1}0$ for some k, where $\mathsf{s}^0 0 = 0$ and $\mathsf{s}^{k+1}0 = \mathsf{s}(\mathsf{s}^k 0)$.

The types of system $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ will be like the types of system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$ considering as m-types also those corresponding to existential and universal quantification and with the sets of atomic and negated atomic types defined as follows:

$$\mathscr{A} = \{u = v \mid u, v \text{ PA-terms of type Int}\}$$

$$\mathscr{A}^\perp = \{u \neq v \mid u, v \text{ PA-terms of type Int}\}.$$

The definition of negation (Definition 2.2) is naturally extended as follows:

5. $(\exists n.A)^\perp = \forall n.A^\perp$
6. $(\forall n.A)^\perp = \exists n.A^\perp$.

A type is said to be PA-closed if it does not contain free PA-term variables.

DEFINITION 3.3 ($\lambda^{\mathrm{Sym}}_{\mathrm{PA}}$-Rules). Atomic rules:

$$(\mathsf{PA}_1)\ \frac{}{\mathsf{PA}_1: (u = u)} \qquad (\mathsf{PA}_2)\ \frac{P: (u = t)}{\mathsf{PA}_2(P): (t = u)}$$

$$(\mathsf{PA}_3)\ \frac{P: (u = v) \quad Q: (v = t)}{\mathsf{PA}_3(P, Q): (u = t)} \qquad (\mathsf{PA}_4)\ \frac{P: (\mathsf{s}0 = 0)}{\mathsf{PA}_4(P): \perp}$$

$$(\mathsf{PA}_5)\ \frac{P: (u = v)}{\mathsf{PA}_5(P): (\mathsf{s}u = \mathsf{s}v)} \qquad (\mathsf{PA}_6)\ \frac{P: (\mathsf{s}u = \mathsf{s}v)}{\mathsf{PA}_6(P): (u = v)}$$

By "atomic rule" we will always mean a rule with atomic formulas as premises and conclusion.

Logical rules:

$$(\mathrm{var})\ x^A: A$$

$$(\langle\ ,\ \rangle)\ \frac{P_1: A_1 \quad P_2: A_2}{\langle P_1, P_2\rangle: A_1 \wedge A_2}$$

$$(\sigma_i)\ \frac{P_i: A_i}{\sigma_i^{A_1, A_2}(P_i): A_1 \vee A_2} \quad (i = 1, 2)$$

$$[\,x^A: A\,]$$

$$\vdots$$

$$(\lambda)\ \frac{P: \perp}{\lambda x^A.P: A^\perp}$$

$$(\bigstar)\ \frac{P_1: A^\perp \quad P_2: A}{P_1 \bigstar P_2: \perp}$$

$$(\lambda_\forall)\ \frac{P: A}{\lambda_\forall n.P: \forall n.A} \quad \text{for all} \quad x^B \in \mathrm{FV}(P), \quad n \notin \mathrm{FV}(B)$$

$$(\sigma_t)\ \frac{P: A(t)}{\sigma_t(P): \exists n.A(n)}$$

$$[\,n: \mathrm{Int}\,][\,x: A(n)\,]$$

$$\vdots$$

$$(\mathsf{Ind})\ \frac{u: \mathrm{Int} \quad P: A(0) \quad F: A(\mathsf{s}n)}{\mathsf{Ind}_{n,x}(u, P, F): A(u)}$$

n not free in types of assumptions different from $A(n)$.

$$(\mathrm{Conv})\ \frac{P: A(u)}{P: A(u')} \quad \text{if} \quad u \simeq u'$$

Rules $(\langle\ ,\ \rangle)$, (σ_i), (σ_t), and $(\lambda_\forall)$ will be called *introduction* rules in the following.

We shall say that a term P represents the proof of a formula A (is a term of type A) if it is possible to derive $P: A$.

We shall denote by $\Gamma \vdash_{\lambda^{\mathrm{Sym}}_{\mathrm{PA}}} P: A$ the fact that $P: A$ is derivable in $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ from the set of assumptions Γ.

We shall call a term *atomic* if it is formed only by atomic rules.

It is easy to show that all other rules of first order logic natural deduction can be derived from the ones given above.

A term will be called *PA-closed* if its type is so and if it contains no free PA-term variables.

DEFINITION 3.4 ($\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$-Reduction Rules). We add to the reduction rules of Definition 2.6 the following reductions.

$(\beta_\forall)$ $(\lambda_\forall n.P) \star \sigma_t(Q) \to_{\beta_\forall} P[t/n] \star Q$
$(\beta^\perp_\forall)$ $\sigma_t(Q) \star (\lambda_\forall n.P) \to_{\beta^\perp_\forall} Q \star P[t/n]$

(Ind_0) $\mathsf{Ind}(0, P, F) \to_{\mathsf{Ind}_0} P$
(Ind_s) $\mathsf{Ind}_{n,x}(\mathsf{s}^{k+1}0, P, F) \to_{\mathsf{Ind}_s} F[\mathsf{s}^k 0, \mathsf{Ind}_{n,x}(\mathsf{s}^k 0, P, F)]$
$F[\mathsf{s}^k 0, \mathsf{Ind}_{n,x}(\mathsf{s}^k, 0, P, F)]$ is short for
$F[\mathsf{s}^k 0/n, \mathsf{Ind}_{n,x}(\mathsf{s}^k 0, P, F)/x]$

(Comp) $u \to_{\mathrm{cmp}} u'$ if u and u' are PA-terms and $u \to_{\mathrm{PA}} u'$ (We have this rule in order to be able to reduce PA-terms when they are inside $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ terms).

The choice of the rule (Ind_s) instead of the more liberal

$$\mathsf{Ind}_{n,x}(\mathsf{s}u, P, F) \to_{\mathsf{Ind}_s} F[u, \mathsf{Ind}_{n,x}(u, P, F)] \quad \text{for any} \quad u\text{: } \mathit{Int}$$

has been made just to simplify the proof of the strong normalization theorem. Such a proof can easily be modified to take into account the more liberal version of the rule.

By close inspection of the reduction rules and of the term formation rules, it is easy to check that types are preserved by reduction (subject reduction property).

The property of strong normalization holds for $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$-terms.

THEOREM 3.5 (Strong Normalization for $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$). *Terms of $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ are strongly normalizable.*

The proof of this theorem will be given in the Appendix and can be obtained by an extension of the proof of strong normalization for the system $\lambda^{\mathrm{Sym}}_{\mathrm{Prop}}$. This latter proof will be presented separately in Section 5 since the simplicity of the system allows a better understanding of the main ideas.

4. SHAPE OF NORMAL FORMS IN $\lambda^{\mathrm{SYM}}_{\mathbf{PA}}$ AND EXTRACTION OF CONSTRUCTIVE CONTENTS

We introduce now two sets of types. From terms of which the types are in one of these sets, namely Σ^0_1, it will be possible to extract the constructive contents expressed by the types, seen as specifications.

DEFINITION 4.1. (i) The set of Σ^0_1 types (formulas) is composed out of PA-closed m-types containing neither elements of the set $\mathscr{A}^\perp$, nor universal quantifications, i.e., Σ^0_1 is the restriction to PA-closed elements of the set defined by the grammar

$$\mathsf{S} ::= \mathscr{A} \mid \mathsf{S} \wedge \mathsf{S} \mid \mathsf{S} \vee \mathsf{S} \mid \exists \mathsf{n}.\mathsf{S},$$

where n ranges over the category of numerical variables.

(ii) The set of Π^0_1 types (formulas) is composed out of the types D such that $D^\perp \in \Sigma^0_1$, i.e., Π^0_1 is the restriction to PA-closed elements of the set defined by the grammar

$$\mathsf{P} ::= \mathscr{A}^\perp \mid \mathsf{P} \wedge \mathsf{P} \mid \mathsf{P} \vee \mathsf{P} \mid \forall \mathsf{n}.\mathsf{P},$$

where n ranges over the category of numerical variables.

It is worthwhile to outline that all results of the present section, as well as that of Strong Normalization, hold also in case one considers any (consistent) set Δ of atomic rules instead of those for Peano arithmetic. We use $\lambda^{\mathrm{Sym}}_{\Delta}$ for the calculus obtained out of $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ replacing its set of atomic rules by a set of atomic rules Δ.

DEFINITION 4.2. (i) A set Δ of atomic rules is *consistent* if there exists no atomic and closed proof of $\perp$.

(ii) A system $\lambda^{\mathrm{Sym}}_{\Delta}$ is consistent if there exists no closed term of type $\perp$.

LEMMA 4.3. *The set of atomic rules* **PA** *is consistent.*

We state now the main theorem of this section.

THEOREM 4.4 (Normal Form Theorem). *Let C be either $\perp$ or in Σ^0_1 and let P be a closed and PA-closed normal term of type C. Then*

(i) *C is atomic $\Rightarrow$ P is atomic.*

(ii) *C is not atomic $\Rightarrow$ P ends with one of the following rules*: $(\langle\ ,\ \rangle)$, (σ_i), *or* (σ_t).

The statement of the above theorem is clearly analogous to well known properties of simply typed λ-calculus (it says that the results of a computation over a Σ^0_1-type has a concrete meaning). Its proof, however, is not at all trivial, since here we are dealing with classical logics.

It is worth remarking that the restriction to Σ^0_1-types in the Normal Form Theorem is *essential*. For instance, the two closed terms in normal form at the end of Section 2, of type $(\alpha \vee \alpha^\perp)$, are not of the form $\sigma_i(P)$, because we can classically prove $(\alpha \vee \alpha^\perp)$ without proving neither α nor $\alpha^\perp$.

From the Normal Form Theorem, to the proof of which the next subsection will be devoted, it descends easily the consistency of any system $\lambda^{\mathrm{Sym}}_{\Delta}$, in case the set Δ of atomic rules is consistent.

COROLLARY 4.5.

$$\Delta \text{ consistent} \Rightarrow \lambda^{\mathrm{Sym}}_{\Delta} \text{ consistent};$$

in particular, by Lemma 4.3, $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ *is consistent.*

It is easy to see that Theorem 4.4 shows an easy and clear way to get what Kreisel obtained in [Kre 58], i.e., the extraction of computational contents out of classical proofs. By Theorem 4.4, from *normalized* proof of a disjunction we can get a proof of one of the disjuncts, and from a proof of an existentially quantified formula, we can get a witness of it. More generally, given a closed proof of a Σ^0_1 formula in $\lambda^{\mathrm{sym}}_{\mathbf{PA}}$, it is possible to get, by a simple inspection of the normalized proof, the witnesses of all the subformulas of the form $\exists n.A(n)$. This means also that, if we have a formula corresponding to a program specification, i.e., of the form $\forall m.\exists n.A(m, n)$, we can get, out of a proof P of it, a recursive function $f\colon \mathbb{N} \to \mathbb{N}$ such that for all $k \in \mathbb{N}$, $A(k, f(k))$ holds. For a fixed $k \in \mathbb{N}$, to get $f(k)$ we have simply to normalize the proof $\lambda x^{\forall n.A^{\perp}(k,n)}.(\sigma_k(x) \star P)$.

We enphasize once more that the reduction rules of our calculus are quite simple and natural, and not devised ad hoc for constructive content extraction purposes.

4.1. *Proof of the Normal Form Theorem*

This section will be devoted to the proof of Theorem 4.4. First of all we introduce the class of *minimal* proofs. Our proof will then proceed by first showing, after a series of technical lemmas, that the statement of Theorem 4.4 holds for minimal proofs and then that indeed each closed proof of a sentence in Σ^0_1 is minimal.

DEFINITION 4.6. A term (proof) of $\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$ is *minimal* if it is built out only of rules $(\langle\ ,\ \rangle)$, (σ_i), and (σ_t) and of atomic rules.

LEMMA 4.7. *Let u be a PA-term in normal form. Then either u ends with an introduction* $(0, \mathsf{s}$, *or* $\lambda)$, *or it is formed only by eliminations* (Rec *or application*) *followed by a variable.*

Proof. By induction on u.

- u is a variable. Immediate.
- u ends with an introduction rule. Immediate.
- u ends with an elimination rule. The thesis follows by the induction hypothesis, since if the leftmost immediate subterm of u ended with an introduction rule, u would not be in normal form. ▮

LEMMA 4.8. *Let u be a closed PA-term of type Int. Then its normal form is a numeral*; *i.e.*, $\mathsf{nf}(u) \equiv \mathsf{s}^k 0$ *for a* $k \geqslant 0$.

Proof. Immediate from Lemma 4.7. ▮

LEMMA 4.9. (i) *Let P be a PA-closed normal term. Then P does not begin with* Ind (*even if* ind *can occur inside P*).

(ii) *Let $P_1 \star P_2$ be a PA-closed term in normal form. Then either P_1 or P_2 is a variable.*

Proof. (i) Let us assume, towards a contradiction, that P is of the form $\mathsf{Ind}(u, Q, F)$. By Lemma 4.8 it follows that $u \equiv \mathsf{s}^{k+1}0$ for $k \geqslant 0$. We get a contradiction since the reduction $(\mathsf{Ind}_{\mathsf{s}})$ or (Ind_0) could be applied.

(ii) Let us assume, towards a contradiction, that neither P_1, of type, say, $A^{\perp}$, nor P_2, of type A, is a variable. First of all we notice that neither P_1 nor P_2 can be of the form $\lambda x.Q$, $\mathsf{Ind}(u, P, F)$ or be an application. The first case is excluded since, otherwise, $P_1 \star P_2$ would not be normal, the second by (i), while the third case would imply that $A \equiv \perp$, which is impossible. Therefore, both P_1 and P_2 end either with an introduction or with an atomic rule. We can show that even the latter of these cases is to be excluded since, if one of the two terms ends with an atomic rule, the other one would have a *negated* atomic type and hence could end neither with an introduction nor with an atomic rule, contradicting what we have just proved. Thus both P_1 and P_2 end with an introduction. It is easy to check that if one ends with $(\langle\ ,\ \rangle)$ the other has to end with (σ_i) or, alternatively, if one ends with $(\lambda_\forall)$ the other has to end with (σ_t). In both cases, however, we get a contradiction; i.e., $P_1 \star P_2$ would not be normal, since it would then be possible to apply reduction $\pi(\pi^{\perp})$ or $\beta_\forall(\beta^{\perp}_\forall)$, respectively. We then conclude that one of P_1, P_2 is necessarily a variable. ▮

To continue now towards the complete proof of the Normal Form Theorem we need to introduce one more notion: that of Σ^0_1-term.

DEFINITION 4.10. Let P be a term and C its type. P is a *Σ^0_1-term* if:

1. is PA-closed.
2. $C \in \Sigma^0_1$ or $C \equiv \perp$.
3. For all $x \in \mathrm{FV}(P)$, if D is the type of x then $D \in \Pi^0_1$.

LEMMA 4.11. *Let P be a Σ^0_1-term in normal form, and Q a subterm of P. Then*:

1. *Q is a variable x iff it has type in Π^0_1 and*
2. *Q is not a variable iff it is a Σ^0_1-term.*

Moreover, if Q is a variable x then it is on one side of some application $x \star Q'$, or $Q' \star x$, occurring in P.

Proof. By induction over the structure of P.

- $P \equiv x$. This case can never occur, since, by the definition of Σ^0_1-term, P has to have its type in Σ^0_1 and to have the types of its free variables in Π^0_1, and this is impossible.
- $P \equiv \langle P_1, P_2 \rangle$. Then $P\colon A_1 \wedge A_2$ with A_1 and A_2 both in Σ^0_1. Moreover, the free variables of P_1 and P_2 have types in Π^0_1. Thus P_1 and P_2 are Σ^0_1-terms. Since the strict subterms of P are all the subterms of P_1 and P_2, we obtain the thesis by the induction hypothesis.

• $P \equiv \sigma_i(P_i)$. As in the previous case, we deduce that P_i is a Σ^0_1-term and hence we can obtain the thesis by invoking the induction hypothesis.

• $P \equiv \lambda x.P'$. Since $\lambda x.P'$ is a Σ^0_1-term , we get that the type of x is in Π^0_1. Moreover, P' as type $\bot$ and $\mathrm{FV}(P') \subseteq \mathrm{FV}(P) \cup \{x\}$. We then infer that P' is a Σ^0_1-term on which it is possible to apply the induction hypothesis to get the thesis.

• $P \equiv P_1 \star P_2$. By Lemma 4.9 (ii), since P is in normal form and Σ^0_1-terms are PA-closed by definition, it follows that P_1 or P_2, say P_1, is a variable x. Then we have that the type of x is in Π^0_1 and therefore P_2 has type in Σ^0_1 and $\mathrm{FV}(P_2) \subseteq \mathrm{FV}(P)$. This means that P_2 is a Σ^0_1-term. By applying the induction hypothesis on P_2 we get the thesis for P, since the strict subterms of P are x or subterms of P_2.

• $P \equiv \lambda_\forall n.P'$. This can never be the case since the type of P would be of the form $\forall n.A$ and hence P would not be a Σ^0_1-term.

• $P \equiv \sigma_t(P')$. In this case the type of P is of the form $\exists n.A(n)$ with $A(t)$ the type of P' and $\mathrm{FV}(P') \subseteq \mathrm{FV}(P)$. It is then easy to check that P' is a Σ^0_1-term. By applying the induction hypothesis on P' we get the thesis for P since the strict subterms of P are the subterms of P'.

• $P \equiv \mathsf{PA}_i(P_1, ..., P_n)$ $(1 \leqslant i \leqslant 6)$. The thesis follows immediately from the induction hypothesis on $P, ..., P_n$ that have necessarily atomic types and free variables all in $\mathrm{FV}(P)$.

• $P \equiv \mathsf{Ind}(u, Q, F)$. This case can never occur, since otherwise we would have a contradiction with Lemma 4.9 (i), since a Σ^0_1-term is PA-closed by definition. ▮

Corollary 4.12. Let P be a normal Σ^0_1-term. Then

(i) P does not contain $\lambda_\forall$ symbols.

(ii) P does not contain Ind symbols.

Proof. (i) If there were a term of the form $\lambda_\forall n.P$, it would have type of the form $\forall n.A(n)$ which, by not being Σ^0_1, would lead to a contradiction with Lemma 4.11.

(ii) Towards a contradiction, let $\mathsf{Ind}(u, R, F)$ be the maximal subterm of P beginning with Ind. u is in PA-normal form by hypothesis and since P is in normal form, u has necessarily to have at least one free PA-variable n. By the PA-closedness of P, n is necessarily bound by an outer $\lambda_\forall$, contradicting (i). ▮

We can now prove the statement of the Normal Form Theorem, restricted to minimal terms.

LEMMA 4.13. *Let P be a minimal,* PA*-closed term and let α be $\bot$ or an element in $\mathscr{A}$.*

(i) $P: \alpha \Rightarrow$ *P is atomic.*

(ii) $P: A_1 \wedge A_2 \Rightarrow$ *P is of the form $\langle P_1, P_2 \rangle$ with P_1 and P_2 minimal.*

(iii) $P: A_1 \vee A_2 \Rightarrow$ *P is of the form $\sigma_i(P_i)$ with P_i minimal.*

(iv) $P: \exists n.A \Rightarrow$ *P is of the form $\sigma_t(P')$ with P' minimal.*

Proof. (i) By induction on the structure of P. Since P contains only atomic rules and introductions and has an atomic type, it has necessarily the form $\mathsf{PA}_i(P_1, ..., P_n)$ $(1 \leqslant i \leqslant 6)$. Since the P_i's are minimal, we get the thesis by applying the induction hypothesis to them all.

(ii) (iii) (iv) Since P contains only atomic rules and introductions and has a non-atomic type, we have necessarily that it has the form $\langle P_1, P_2 \rangle$, $\sigma_i(P_i)$, $\sigma_t(P')$, respectively. P_1, P_2, P_i, and P' are minimal because P is. ▮

As last step, we prove now that closed normal Σ^0_1-terms are minimal. The Normal Form Theorem will then follow by Lemma 4.13.

LEMMA 4.14. *Let P be a closed normal Σ^0_1-term*

(i) *If P contains no λ symbol, then it is minimal.*

(ii) *P contains no λ symbol.*

Proof. (i) Let P be a closed Σ^0_1-term with no λ-symbol in it. By definition, to prove that it is minimal, we have to prove that it does not contain free or bound term variables, symmetric applications or Ind or $\lambda_\forall$ symbols. Since it is closed, and variables are bound only by λ's, P cannot contain variables. If P contained an application then, by Lemma 4.9 (ii), it would contain a variable, contradicting what we have just proved. Corollary 4.12 can instead be invoked to make sure that no Ind or $\lambda_\forall$ symbol is present in P.

(ii) We begin by first proving that there exist no subterms of P of the form $\lambda x.Q$ with $x \in \mathrm{FV}(Q)$. In order to do this, let us assume, towards a contradiction, that there exists some subterm of the form $\lambda x.Q$ with $x \in \mathrm{FV}(Q)$, and let us take one of the shortest (w.r.t. the subterm inclusion), say $\lambda x'.Q'$. Since $x' \in \mathrm{FV}(Q')$ and P is a normal Σ^0_1-term, by Lemma 4.11 x' occurs necessarily in a subterm $x' \star Q''$ or $Q'' \star x'$. Let us now consider the shortest among such terms, in such a way that $x \notin \mathrm{FV}(Q'')$. Thus $Q' \equiv C[x' \star Q'']$ (or $C[Q'' \star x']$) for a context $C[\ ]$. Since $\lambda x'.Q'$ is among the shortest subterms with this form and with $x' \in \mathrm{FV}(Q')$, we get that $\mathrm{FV}(x' \star Q'')$ $(\mathrm{FV})(Q'' \star x')) \subseteq \mathrm{FV}(Q')$. It follows that $Q' \equiv x \star Q''$ $(Q'' \star x)$, i.e., $C[\ ] \equiv [\ ]$; otherwise $Q \to_{\mathrm{Triv}} x' \star Q''$ $(Q'' \star x')$, contradicting the hypothesis of normality of P. Even in such a case, however, we get a contradiction since, by the fact that $x' \notin \mathrm{FV}(Q'')$, we could apply an η- ($\eta^\bot$-) reduction on $\lambda x'.x' \star Q''$ $(\lambda x'.Q'' \star x')$. Therefore we can infer that there exist no subterms of P of the form $\lambda x.Q$ with $x \in \mathrm{FV}(Q)$. From this fact then follows that no variables are

discharged in P and hence, since P is closed, all its subterms are.

We can now proceed to prove that there exist no subterms of P of the form $\lambda x.Q$ at all. Towards a contradiction, let us assume that there exist subterms $\lambda x'.Q'$ and take one of the shortest, in such a way that no λ occurs in Q'. Since $Q':\bot$, Q' is not a variable and, by Lemma 4.11, it is a closed normal Σ^0_1-term. By point (i)[1] it follows that Q' is a closed minimal proof of $\bot$, and, by Lemma 4.13(i) an atomic one. This contradicts the consistency of the set of atomic rules. ▮

LEMMA 4.15. *Let P be a closed normal Σ^0_1-term. Then P is minimal.*

Proof. Immediate from Lemma 4.14 (ii) and (i). ▮

We can now give the Proof of Theorem 4.4.

Proof of Theorem 4.4. Immediate from Lemmas 4.13 and 4.15.

4.2. *A Further Improvement*

From the Lemmas proved above it is possible to derive a further lemma which is quite interesting from both a *theoretical* and *an applicative* point of view.

LEMMA 4.16. *Let $P(x_1^{N_1}, ..., x_k^{N_k})$ be a normal Σ^0_1-term. Then either it is minimal (and hence closed) or it contains a subterm $x_i \star Q_i$ (or $Q_i \star x_i$), where $Q_i^{N_i^\perp}$ is minimal.*

Proof. If P is closed ($k=0$), apply Lemma 4.15. Otherwise, any x_i has necessarily to occur, by Lemma 4.11, in subterms of the form $x_i \star Q_i$ (or $Q_i \star x_i$). Let us take one of the shortest among such terms. We get that, for such term, Q_i is closed and, by Lemma 4.11, is a Σ^0_1-term. Lemma 4.15 enables us now to get the thesis. ▮

From a *theoretical* point of view the above lemma states that for any

$$x_1^{N_1}, ..., x_k^{N_k} \vdash^{\mathrm{Sym}}_{\lambda^{\rightarrow}_{\mathcal{A}}} P: A \quad \text{with} \quad A \in \Sigma^0_1 \quad \text{and} \quad N_i \in \Pi^0_1,$$

P contains either an example for A or a counterexample for some of N_i. This agrees with the *Curry–Howard* interpretation of P.

P can be seen as a classical proof of

"If $N_1, ..., N_k$ then P"

and the classical meaning of this is

"Either $N_1^\perp$ or $\cdots$ or $N_k^\perp$ or P".

[1] The need of this result here explains why, in this lemma, the statement of (i) precedes the one of point (ii).

From an *applicative* point of view, Lemma 4.16 can be used to speed up the process of extraction of constructive content. If a Σ^0_1-proof $P[R_1/x_1, ..., R_k/x_k]$ contains closed terms $R_1, ..., R_k$ having Π^0_1-types, *we do not need to consider them during the normalization process.* We can instead replace them by fresh variables, since all normal forms of P are indeed normal forms of $P[R_1/x_1, ..., R_k/x_k]$.

To prove this fact, it sufficies to observe that if the normal form of P is minimal then it is closed, and hence it is also a normal proof of $P[R_1/x_1, ..., R_k/x_k]$. Otherwise, by Lemma 4.16, the normal form of P should have some subterms $x_i \star Q_i$ (or $Q_i \star x_i$) with Q_i minimal. This, however, leads to a contradiction since, by replacing x_i by R_i, we would get a closed proof of $\bot$, which is impossible.

Informally speaking, as G. Kreisel often said, when we have a proof P of a Π^0_1-type N, we only know that N is inhabited, and nothing else. We can use a fresh variable x^N to build an inhabitant of N, being sure that x will disappear during the normalization process.

5. STRONG NORMALIZATION FOR $\lambda^{\mathrm{SYM}}_{\mathrm{PROP}}$

This section will be devoted to the proof of Theorem 2.9. We shall use a non-trivial variant of Tait's well-known computability method. We first assign to each type C a set $[\![C]\!]$ (symmetric candidate) of *computable* terms of type C and show that for all $P \in [\![C]\!]$, P strongly normalizes. Then we prove that for each type C, if P has type C, then $P \in [\![C]\!]$.

The main property the candidate assignment has to enjoy, in order to make the proof work, consists in the fact that the set of computable terms of a m-type A reflects the way these terms are built. In the present case these properties are the ones stated in the following claim.

Claim 5.1. There exists an assignment $[\![\text{—}]\!]: A \mapsto [\![A]\!] \subseteq \mathrm{Term}_A$ such that $[\![\bot]\!] = \mathrm{SN}_\bot$ and

(i) $\mathrm{Var}_A \subseteq [\![A]\!]$.

(ii) $\langle P_1, P_2\rangle \in [\![A_1 \wedge A_2]\!] \Leftrightarrow P_1 \in [\![A_1]\!]$ and $P_2 \in [\![A_2]\!]$.

(iii) $\sigma_i(P_i) \in [\![A_1 \vee A_2]\!] \Leftrightarrow P_i \in [\![A_i]\!] \quad (i=1,2)$.

(iv) $\lambda x.P \in [\![A]\!] \Leftrightarrow \forall Q \in [\![A^\perp]\!].P[Q/x] \in [\![\bot]\!]$.

For simply typed λ-calculi, the properties required for the candidate assignment can be also considered as an inductive definition of the sets $[\![A]\!]$. Unfortunately, this is not the case for our system. The properties stated above cannot be considered as a definition of $[\![A]\!]$, since clause (iv) would cause a circularity ($[\![A^\perp]\!]$ would be defined in terms of $[\![A]\!]$ which, since $A \equiv A^{\perp\perp}$, would be defined in terms of $[\![A^\perp]\!]$ itself). We have therefore to define candidates using a different method. This will be done in Subsection 5.1 where, besides, properties (i)–(iv) will be proved. Until then we shall assume Claim 5.1 to hold, i.e. that the candidates are already well-defined for all m-types, and that properties (i)–(iv) are already proved.

LEMMA 5.2. *Let C be a type. Then*

$$[\![C]\!] \subseteq \mathrm{SN}_C.$$

Proof. By induction on the structure of C, considering the different shapes of $P \in [\![C]\!]$.

- $P \equiv x^C$. Trivially, $x^C \in \mathrm{SN}_C$.
- $P \equiv \langle P_1, P_2 \rangle$. Then $C \equiv C_1 \wedge C_2$. By Claim 5.1 (ii) follows that $P_i \in [\![C_i]\!]$ $(i = 1, 2)$. Now, by the induction hypothesis on $[\![C_i]\!]$, we get $P_i \in \mathrm{SN}_{C_i}$. Since each reduction on $\langle P_1, P_2 \rangle$ is a reduction on P_1 or P_2, we get $\langle P_1, P_2 \rangle \in \mathrm{SN}_{C_1 \wedge C_2}$.
- $P \equiv \sigma_i(P_i)$. Then $C \equiv C_1 \vee C_2$. By Claim 5.1 (iii) follows that $P_i \in [\![C_i]\!]$. Now, by the induction hypothesis and the fact each reduction on $\sigma_i(P_i)$ is indeed a reduction on P_i we get $\sigma_i(P_i) \in \mathrm{SN}_{C_1 \vee C_2}$.
- $P \equiv \lambda x.P'$. Since, by Claim 5.1(i), $x \in [\![C^\perp]\!]$, follows, by Claim 5.1(iv), that $P' \in \mathrm{SN}_\perp$. Each reduction on $\lambda x.P'$ is indeed either a reduction on P' or a $\eta(\eta^\perp)$-reduction and $P' \equiv P'_1 \star x (P' \equiv x \star P_1)$. Therefore $\lambda x.P$ has a bound that is at most $1 + m$, where m is the bound of P.
- $P \equiv P_1 \star P_2$. In such a case $P \in [\![\perp]\!] \equiv \mathrm{SN}_\perp$. ∎

LEMMA 5.3. *Let $P \in \mathrm{Term}_A$. Then*

(i) $\forall Q \in [\![A^\perp]\!].(P \star Q \in SN_\perp) \quad \Rightarrow \quad P \in [\![A]\!].$

(ii) $\forall Q \in [\![A^\perp]\!].(Q \star P \in SN_\perp) \quad \Rightarrow \quad P \in [\![A]\!].$

Proof. Because of the symmetry of application, it is enough to prove just one of (i) and (ii), say (i).

Assume $\forall Q \in [\![A^\perp]\!].(P \star Q \in \mathrm{SN}_\perp)$. We shall prove $P \in [\![A]\!]$ by induction on the structure of A. We distinguish now different cases according to the shape of P. Note that, since A cannot be $\perp$, the case $P \equiv P_1 \star P_2$ cannot occur.

- $P \equiv x^A$. Then $P \in [\![A]\!]$ by Claim 5.1(i).
- $P \equiv \langle P_1, P_2 \rangle$. Then $A \equiv A_1 \wedge A_2$. By Claim 5.1(ii), it suffices to prove that $P_i \in [\![A_i]\!]$ $(i = 1, 2)$. By the induction hypothesis this can in turn be proved by showing that, for all $Q_i \in [\![A_i^\perp]\!]$, we have that $P_i \star Q_i \in \mathrm{SN}_\perp$. Since $\langle P_1, P_2 \rangle \star \sigma_i(Q_i) \to_1 P_i \star Q_i$, it would be enough to prove $\langle P_1, P_2 \rangle \star \sigma_i(Q_i) \in \mathrm{SN}_\perp$. By the assumption $\forall Q \in [\![A^\perp]\!].(P \star Q \in \mathrm{SN}_\perp)$, this descends from $\sigma_i(Q_i) \in [\![A_1^\perp \vee A_2^\perp]\!]$ which, in turn, is a consequence of Claim 5.1(iii) and of $Q_i \in [\![A_i^\perp]\!]$.
- $P \equiv \sigma_i(P_i)$. This case can be treated similarly to the previous one.
- $P \equiv \lambda x.P_1$. By hypothesis we have that $\forall Q \in [\![A^\perp]\!].((\lambda x.P_1) \star Q \in \mathrm{SN}_\perp)$, from which it immediately follows that $\forall Q \in [\![A^\perp]\!].P_1[Q/x] \in \mathrm{SN}_\perp$. Hence, by Claim 5.1(iv), we get $\lambda x.P_1 \in [\![A]\!]$. ∎

LEMMA 5.4.

$$P \in [\![C]\!], \quad P \to_1 P' \Rightarrow P' \in [\![C]\!].$$

Proof. By induction on C, considering the different forms of $P \in [\![C]\!]$.

- $P \equiv x^C$. It can never be that $x^C \to_1 P'$.
- $P \equiv \langle P_1, P_2 \rangle$. In such a case $P \equiv \langle P_1, P_2 \rangle$, $C \equiv C_1 \wedge C_2$, and either $P_1 \to_1 P'_1$ or $P_2 \to_1 P'_2$. In either case we get the thesis by the induction hypothesis and Claim 5.1(ii).
- $P \equiv \sigma_i(P_i)$. This case is similar to the previous one.
- $P \equiv \lambda x.P_1$. In this case we have two possibilities: either $P' \equiv \lambda x.P'_1$ and $P_1 \to_1 P'_1$ or $P_1 \equiv Q_1 \star x$ $(P_1 \equiv x \star Q_1)$ and $P' \equiv Q_1$ (we performed an $\eta(\eta^\perp)$-reduction).

In the first case, by Claim 5.1(iv), to check that $\lambda x.P'_1 \in [\![C]\!]$ it suffices to show, by Claim 5.1, that $\forall Q \in [\![C]\!].P'_1[Q/x] \in \mathrm{SN}_\perp$. This fact follows immediately from $P'_1[Q/x] \in \mathrm{SN}_\perp$, which in turn follows from the hypothesis and Claim 5.1(iv).

In the second case, by Lemma 5.3, it suffices to show that $\forall S \in [\![C^\perp]\!].Q_1 \star S \in \mathrm{SN}_\perp$ (or equivalently $\forall S \in [\![C^\perp]\!].S \star Q_1 \in \mathrm{SN}_\perp$). This follows from the fact $P \equiv \lambda x.Q_1 \star x$ $(\lambda x.x \star Q_1) \in [\![C]\!]$ and from Claim 5.1(iv), since $x \notin \mathrm{FV}(Q_1)$ and therefore $P_1[S/x] \equiv Q_1 \star S$.

- $P \equiv P_1 \star P_2$. Then $P \in [\![\perp]\!] \equiv \mathrm{SN}_\perp$ and $P \to_1 P'$ implies $P' \in \mathrm{SN}_\perp \equiv [\![\perp]\!]$. ∎

LEMMA 5.5.

$$P \in [\![A]\!], \quad Q \in [\![A^\perp]\!] \quad \Rightarrow \quad P \star Q \in \mathrm{SN}_\perp.$$

Proof. By Lemma 5.2, there exist n and m, bounds for P and Q, respectively. We prove $P \star Q \in \mathrm{SN}_\perp$ by double induction: primary induction on the structure of A and secondary induction on $n + m$.

It is easy to see that proving the thesis is equivalent to proving that

$$\forall S.(P \star Q \to_1 S \Rightarrow S \in \mathrm{SN}_\perp).$$

We have eight cases, pairwise symmetric, to consider.

1. $\begin{cases} P \equiv \lambda x.P_1, & S \equiv P_1[Q/x] \\ Q = \lambda x.Q_1, & S \equiv Q_1[P/x] \end{cases}$
2. $\begin{cases} P \to_1 P', & S \equiv P' \star Q \\ Q \to_1 Q', & S \equiv P \star Q' \end{cases}$
3. $\begin{cases} P \star Q \to_{\mathrm{Triv}} S, & S \text{ subterm of } P \\ P \star Q \to_{\mathrm{Triv}} S, & S \text{ subterm of } Q \end{cases}$
4. $\begin{cases} P \equiv \langle P_1, P_2 \rangle, & Q \equiv \sigma_i(Q_i), & S \equiv P_i \star Q_i \\ P \equiv \sigma_i(P_i), & Q \equiv \langle Q_1, Q_2 \rangle, & S \equiv P_i \star Q_i. \end{cases}$

1. For these cases, the thesis follows immediately from Claim 5.1(iv).

2. By Lemma 5.4, we have that $P' \in [\![A]\!]$ or $Q' \in [\![A]\!]$. We then get $P' \star Q \in \mathrm{SN}_\perp$ or $P \star Q' \in \mathrm{SN}_\perp$ by the induction hypothesis, since in both cases the sum of the two bounds decreased.

3. Since S is a subterm of P or Q, the thesis follows immediately from the hypothesis and Lemma 5.2.

4. Immediate by primary induction. ∎

We now need a last lemma in order to prove strong normalization.

LEMMA 5.6. *Let C be a type and $P \in \mathrm{Term}_C$ with $\mathrm{FV}(P) \subseteq \{x_1^{A_1}, ..., x_n^{A_n}\}$. Then*

$$\forall P_1 \in [\![A_1]\!] \cdots P_n \in [\![A_n]\!].P[P_1/x_1, ..., P_n/x_n] \in [\![C]\!].$$

Proof. By induction on P.

- $P \equiv x$. In such a case $P \equiv x_i$ and $C \equiv A_i$ for some i, by hypothesis. Hence $P[P_1/x_1, ..., P_n/x_n] \equiv P_i \in [\![A_i]\!] \equiv [\![C]\!]$.
- $P \equiv \langle P_1, P_2 \rangle$, $\sigma_i(P_i)$, $P_1 \star P_2$. In all these cases the thesis follows by the induction hypothesis on P_1, P_2, and P_i, and by Claim 5.1(ii), Claim 5.1(iii), and Lemma 5.5. respectively.
- $\lambda x.P'$. Since we can rename bound variables, it is not restrictive to assume that $x \notin \{x_1^{A_1}, ..., x_n^{A_n}\}$. Now, by Claim 5.1(iv), to prove that

$$(\lambda x.P')[P_1/x_1, ..., P_n/x_n] \equiv \lambda x.P'[P_1/x_1, ..., P_n/x_n] \in [\![C]\!].$$

it is enough to prove that, for all $Q \in [\![C^\perp]\!]$, $P'[Q/x, P_1/x_1, ..., P_n/x_n] \in [\![\perp]\!]$. This last fact follows by the induction hypothesis. ∎

We can now present the proof of Theorem 2.9.

Proof of Theorem 2.9. One inclusion is immediate by definition. For the other one, let $P \in \mathrm{Term}_C$ with $\mathrm{FV}(P) = \{x_1^{A_1}, ..., x_n^{A_n}\}$. By Claim 5.1(i), we get $x_i \in [\![A_i]\!]$ $(i = 1, ..., n)$ and therefore, by Lemma 5.6, $P \equiv P[x_1/x_1, ..., x_n/x_n] \in [\![C]\!]$. Lemma 5.2 now makes it possible to infer $P \in \mathrm{SN}_C$. ∎

5.1. *Symmetric Candidates*: *Definition and Main Properties*

This subsection will be devoted to the definition of sets of computable terms, which we shall call *symmetric candidates*. From such a definition the properties of Claim 5.1 will easily descend.

We have spoken before about the impossibility of considering the properties of Claim 5.1 as an inductive definition for the symmetric candidates. So what we do is to define first an operator on sets of terms of given type for each term constructor, except for the symmetric application (Pair(−), Sigma(−), Lambda(−)). Out of these operators we then define an operator for the negation (Neg(−)), reflecting the possible way of obtaining terms whose type can be seen as a negation. Since we wish the involutive property of negation to be reflected in the symmetric candidates, we define a candidate as a fixed point of the composition of Neg with itself. Since this composition turns out to be an increasing operator, the fixed-point exists by Tarski's Theorem. The definition of Neg uses the notion of symmetric candidate. This is why the definitions of Neg and of symmetric candidate will have to be given simultaneously on the structure of the type.

DEFINITION 5.7. Let A, A_1, A_2 be m-types. We define the operators

$$\begin{aligned}
&\mathrm{Pair}_{A_1, A_2}: && \mathscr{P}(\mathrm{Term}_{A_1}) \times \mathscr{P}(\mathrm{Term}_{A_2}) \to \mathscr{P}(\mathrm{Term}_{A_1 \wedge A_2})\\
&\mathrm{Sigma}^i_{A_1, A_2}: && \mathscr{P}(\mathrm{Term}_{A_i}) \to \mathscr{P}(\mathrm{Term}_{A_1 \vee A_2})\\
&\mathrm{Lambda}_A: && \mathscr{P}(\mathrm{Term}_{A^\perp}) \to \mathscr{P}(\mathrm{Term}_A)
\end{aligned}$$

as follows:

$$\begin{aligned}
&\mathrm{Pair}_{A_1, A_2}(X_1, X_2)\\
&\quad =_{\mathrm{Def}} \{\langle P_1, P_2 \rangle : A_1 \wedge A_2 \mid P_1 \in X_1 \text{ and } P_2 \in X_2\}.\\
&\mathrm{Sigma}^i_{A_1, A_2}(X_i)\\
&\quad =_{\mathrm{Def}} \{\sigma_i(P_i) : A_1 \vee A_2 \mid P_i \in X_i\} \quad (i = 1, 2)\\
&\mathrm{Lambda}_A(X)\\
&\quad =_{\mathrm{Def}} \{\lambda x.P : A \mid \forall Q \in X.P[Q/x] \in \mathrm{SN}_\perp\}.
\end{aligned}$$

DEFINITION 5.8. Let A be an m-type. By induction on the structure of A we simultaneously define

(a) The operators

$$\begin{cases} \mathrm{Neg}_A : \mathscr{P}(\mathrm{Terms}_{A^\perp}) \to \mathscr{P}(\mathrm{Terms}_A)\\ \mathrm{Neg}_{A^\perp} : \mathscr{P}(\mathrm{Terms}_A) \to \mathscr{P}(\mathrm{Terms}_{A^\perp}), \end{cases}$$

(b) The sets $[\![A]\!]$ and $[\![A^\perp]\!]$.

as follows:

(a) Let $X \subseteq \mathrm{Term}_A$ and $Y \subseteq \mathrm{Term}_{A^\perp}$. By the involutive property of negation in our system it is not restrictive to consider only the cases for A atomic and A a conjunction.

— $A \equiv \alpha$.

$$\begin{cases} \mathrm{Neg}_\alpha(Y) =_{\mathrm{Def}} \mathrm{Var}_\alpha \cup \mathrm{Lambda}_\alpha(Y)\\ \mathrm{Neg}_{\alpha^\perp}(X) =_{\mathrm{Def}} \mathrm{Var}_{\alpha^\perp} \cup \mathrm{Lambda}_{\alpha^\perp}(X). \end{cases}$$

— $A \equiv A_1 \wedge A_2$ ($A^\perp \equiv A_1^\perp \vee A_2^\perp$):

$$\begin{cases} \mathrm{Neg}_A(Y) =_{\mathrm{Def}} \mathrm{Var}_A \cup \mathrm{Pair}([\![A_1]\!], [\![A_2]\!]) \\ \qquad\qquad \cup \mathrm{Lambda}_A(Y) \\ \mathrm{Neg}_{A^\perp}(X) =_{\mathrm{Def}} \mathrm{Var}_{A^\perp} \cup \left(\bigcup_{i=1}^{2} \mathrm{Sigma}^i([\![A_i^\perp]\!]) \right) \\ \qquad\qquad \cup \mathrm{Lambda}_{A_1^\perp \vee A_2^\perp}(X). \end{cases}$$

(b) For all A, Neg_A is a decreasing operator (w.r.t. set theoretical inclusion), since Lambda_A is so. Then, once one has defined Neg_A for some A, it is possible to get, by Tarski's Fixed-Point Theorem, the smallest fixed-point of the (increasing) operator corresponding to the composition of Neg_A with $\mathrm{Neg}_{A^\perp}$, i.e. $\mathrm{Neg}_A \circ \mathrm{Neg}_{A^\perp}$. Let us call X_0 such a fixed-point. We then define

$$\begin{cases} [\![A]\!] &=_{\mathrm{Def}} X_0 \\ [\![A^\perp]\!] &=_{\mathrm{Def}} \mathrm{Neg}_{A^\perp}(X_0). \end{cases}$$

We extend the definition of computable set of terms to all types by defining $[\![\perp]\!] =_{\mathrm{Def}} \mathrm{SN}_\perp$.

Note that, for our strong normalization proof, the use of the *smallest* fixed point in the above definition is not essential. Any other possible fixed-point would work.

Since $[\![A]\!]$ is a fixed-point of $\mathrm{Neg}_A \circ \mathrm{Neg}_{A^\perp}$, we get what we were looking for, i.e., an operator with the property $[\![A]\!] \equiv \mathrm{Neg}_A([\![A^\perp]\!])$. Given an m-type A, from the fact $[\![A]\!] \equiv \mathrm{Neg}_A([\![A^\perp]\!])$ and the definition of Neg_A, the properties (i)—(iv) of Claim 5.1 descend easily.

Because of the use of the fixed-point theorem, our proof of strong normalization requires the full power of Generalized Inductive Definitions. Such a requirement, however, is not proved to be a minimal one. Therefore it could be the case a proof can be found needing a strictly weaker requirement.

APPENDIX: STRONG NORMALIZATION OF $\lambda_{\mathbf{PA}}^{\mathrm{SYM}}$

In this appendix, we will prove the strong normalization property for terms of $\lambda_{\mathbf{PA}}^{\mathrm{Sym}}$.

We first modify system $\lambda_{\mathbf{PA}}^{\mathrm{Sym}}$ as follows: in the formation of types, instead of the sets

$$\mathscr{A} = \{u = v \mid u, v \text{ PA-terms of type Int}\}$$
$$\mathscr{A}^\perp = \{u \neq v \mid u, v \text{ PA-terms of type Int}\}$$

we take as atomic and negated atomic types the singletons

$$\mathscr{A} = \{\mathsf{U}\} \quad \mathscr{A}^\perp = \{\mathsf{U}^\perp\}.$$

The term formation rules and the reduction rules remain unchanged but for the fact that we replace U for any atomic formula of the form $t = v$, and $\mathsf{U}^\perp$ for any negated atomic formula of the form $t \neq v$.

We call the system so obtained $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$.

It is straightforward to check that, for any term of $\lambda_{\mathbf{PA}}^{\mathrm{Sym}}$, there exists a term of $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$ and these two terms have isomorphic reduction trees. For instance, the term

$$\langle \lambda_\forall n.\mathsf{PA}_1, x^{t=v} \rangle \star y^{\exists n.n \neq n \vee t \neq v}$$

has

$$\langle \lambda_\forall n.\mathsf{PA}_1, x^{\mathsf{U}} \rangle \star y^{\exists n.\mathsf{U}^\perp \vee \mathsf{U}^\perp}$$

as its counterpart in $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$.

We will prove the strong normalization property for terms of $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$, thus immediately obtaining strong normalization for terms of $\lambda_{\mathbf{PA}}^{\mathrm{Sym}}$.

It is worthwhile to stress that the proof given in the present section works also for $\lambda_\Delta^{\mathrm{Sym}}$ for any Δ.

Our proof will consist in an extension of the one of Theorem 2.9. Most of the Lemmas of Section 5 have exactly the same statement when we consider $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$ instead of $\lambda_{\mathbf{PA}}^{\mathrm{Sym}}$ and need simply to have their proofs extended. Even more, the extensions will consist mostly in adding new induction cases. For such lemmas, instead of restating them we will gives only the parts to be added to their proofs.

Of course also Claim 5.1 need to be extended by adding the following clauses to it.

Extension of Claim 5.1.

(v) If v: Int with $\mathsf{nf}(v) \equiv 0$, then

$$\mathsf{Ind}(v, P, F) \in [\![A]\!] \quad \Leftrightarrow \quad P \in [\![A]\!] \cap \mathrm{SN}_A \quad \& \quad F \in \mathrm{SN}_A$$

(vi) If v: Int with $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$, then

$$\mathsf{Ind}(v, P, F) \in [\![A]\!]$$
$$\Leftrightarrow \quad P \in \mathrm{SN}_A \quad \& \quad F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, P, F)] \in [\![A]\!] \cap \mathrm{SN}_A$$

(vii) If v: Int and $\mathsf{nf}(v)$ is not a numeral, then

$$\mathsf{Ind}(v, P, F) \in [\![A]\!] \quad \Leftrightarrow \quad P, F \in \mathrm{SN}_A$$

(viii) $\lambda_\forall n.P \in [\![\forall n.A]\!] \quad \Leftrightarrow \quad \forall t\text{: Int} P[t/n] \in [\![A]\!]$

(ix) $\sigma_t(P) \in [\![\exists n.A]\!] \quad \Leftrightarrow \quad P \in [\![A]\!]$

(x) $\mathsf{PA}_i(P_1, ..., P_{n_i}) \quad \Leftrightarrow \quad \forall j \in \{0, ..., n_i\}.P_j \in \mathrm{SN}_{\mathsf{U}}$
$(1 \leqslant i \leqslant 6)$.

From now on by "Claim 5.1" (or simply "the claim") we will intend its extension for the $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$ case. Also for the lemmas, once we have extended them, any reference to them will be to their versions for system $\lambda_{\mathbf{PA}-\mathsf{U}}^{\mathrm{Sym}}$.

Extension of the proof of Lemma 5.2.

- $P \equiv \mathsf{Ind}(v, Q, F)$. Let a be a bound for v, b a bound for Q, and c a bound for F if $\mathsf{nf}(v) \equiv 0$ or $\mathsf{nf}(v)$ is not a numeral, and a bound for $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, Q, F)]$, too, if $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$. a, b, and c exist by the claim and Lemma 3.2. It is now easy to check that $a+b+2c$ is a bound for $\mathsf{Ind}(v, Q, F)$. In fact, in the worst case, when $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$, one could reduce v, Q and F inside $\mathsf{Ind}(v, Q, F)$, to $\mathsf{s}^{k+1}0$, Q' and F', respectively, and then reduce $F'[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, Q', F')]$.

- $P \equiv \sigma_t(Q)$. Then $C \equiv \exists n.C_1$. By point (ix) of the claim and the induction hypothesis we get $Q \in \mathrm{SN}_C$ and then $\sigma_t(Q) \in \mathrm{SN}_C$.

- $P \equiv \lambda_\forall n.Q$. Then $C \equiv \forall n.C_1$. By point (viii) of the claim and applying the induction hypothesis we get $\forall t\colon \mathrm{Int}.\, Q[t/n] \in \mathrm{SN}_C$ and then $Q \in \mathrm{SN}_C$, from which it follows that $\lambda_\forall n.Q \in \mathrm{SN}_C$.

- $P \equiv \mathsf{PA}_i(P_1, ..., P_{n_i})$ $(1 \leqslant i \leqslant 6)$. Immediate by the claim. ▮

By Lemma 5.2, points (v) and (vi) of the claim can be simplified as follows:

(v) If $v\colon \mathrm{Int}$ with $\mathsf{nf}(v) \equiv 0$, then

$$\mathsf{Ind}(v, P, F) \in [\![A]\!] \quad \Leftrightarrow \quad P \in [\![A]\!] \quad \& \quad F \in \mathrm{SN}_A$$

(vi) If $v\colon \mathrm{Int}$ with $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$, then

$$\mathsf{Ind}(v, P, F) \in [\![A]\!] \quad \Leftrightarrow \quad P \in \mathrm{SN}_A \quad \& \quad F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, P, F)] \in [\![A]\!]$$

Extension of the proof of Lemma 5.3. Because of the presence of terms of the form $\mathsf{Ind}(v, R, F)$, the lemma has to be proved by double induction, the secondary one being on the normalization tree of P.

- $P \equiv \mathsf{Ind}(R, F)$ with $\mathsf{nf}(v) \equiv 0$. By the claim, we have to prove $R \in [\![A]\!]$ and $F \in \mathrm{SN}_A$. $F \in \mathrm{SN}_A$ follows by hypothesis, since F is a subterm of $\mathsf{Ind}(v, R, F) \star Q$. The fact that $\mathsf{Ind}(v, R, F) \star Q \to \mathsf{Ind}(0, R, F) \star Q \to_1 R \star Q$ enables us to infer $\forall Q \in [\![A^\perp]\!].R \star Q \in \mathrm{SN}_\perp$ from the hypothesis and to apply the secondary induction hypothesis, from which $R \in [\![A]\!]$ follows.

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$. By the claim, the thesis follows by checking that $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \in [\![A]\!]$ and $R \in \mathrm{SN}_A$. From the hypothesis and the fact that $\mathsf{Ind}(v, R, F) \to \mathsf{Ind}(\mathsf{s}^{k+1}0, R, F) \to_1 F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)]$, $\forall Q \in [\![A^\perp]\!].F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \star Q \in \mathrm{SN}_\perp$ follows. Then, by the secondary induction hypothesis $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \in [\![A]\!]$, while by hypothesis $R \in \mathrm{SN}_A$.

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v)$ not a numeral. By the hypothesis, we trivially get $R, F \in \mathrm{SN}_A$. The thesis follows by point (vii) of the claim.

- $P \equiv \lambda_\forall n.P_1$. Then $A \equiv \forall n.A_1$. By point (viii) of the claim we need to prove $P_1[t/n] \in [\![A_1]\!]$ for all t: Int. By the hypothesis and point (ix) of the claim we can infer that, for all t: Int, $\forall Q' \in [\![A_1^\perp]\!].(\lambda_\forall n.P_1) \star \sigma_t(Q') \in \mathrm{SN}_\perp$, and hence for all t: Int, $\forall Q' \in [\![A_1^\perp]\!].P_1[t/n] \star Q' \in \mathrm{SN}_\perp$. We can now apply the induction hypothesis to get $P_1[t/n] \in [\![A_1]\!]$ for all t: Int.

- $P \equiv \sigma_t(P_1)$.Then $A \equiv \exists n.A_1$. By point (ix) of the claim we have to prove $P_1 \in [\![A_1]\!]$. By the hypothesis and point (ix) of the claim we can infer $\forall Q' \in [\![A_1^\perp]\!].\sigma_t(P_1) \star (\lambda_\forall m.Q') \in \mathrm{SN}_\perp$ for $m \notin \mathrm{FV}(Q')$, since $Q'[t/m] \equiv Q'$ for all t: Int. Therefore we have also $\forall Q' \in [\![A_1^\perp]\!].P_1 \star Q' \in \mathrm{SN}_\perp.P_1 \in [\![A_1]\!]$ now descends from the induction hypothesis.

- $P \equiv \mathsf{PA}_i(P_1, ..., P_{n_i})$ $(1 \leqslant i \leqslant 6)$. Immediate by the claim. ▮

Extension of the proof of Lemma 5.4. Because of the presence of terms of the form $\mathsf{Ind}(v, R, F)$, the lemma has to be proved by double induction, the secondary one being on the normalization tree of P, which is finite by Lemma 5.2

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v) \equiv 0$. We have four different cases to take into account: $\mathsf{Ind}(v, R, F)$ can reduce either to $\mathsf{Ind}(v, R, F')$, or to $\mathsf{Ind}(v, R', F)$ or to $\mathsf{Ind}(v', R, F)$ or to R (when $v \equiv 0$). In the first case $F \to_1 F'$ and, by the hypothesis and the claim, $R \in [\![C]\!]$ and $F \in \mathrm{SN}_C$. Then, since also $F' \in \mathrm{SN}_C$, the claim allows us to infer $\mathsf{Ind}(v, R, F') \in [\![C]\!]$. In the second case $R \to_1 R'$ and $R \in [\![C]\!]$ by the claim. Since $\mathsf{Ind}(v, R, F) \to \mathsf{Ind}(0, R, F) \to_1 R$ we get, by the secondary induction hypothesis, that $R' \in [\![C]\!]$. The thesis now can be inferred by the claim since, by the hypothesis and the claim, $F \in \mathrm{SN}_C$. In the fourth case the thesis easily descends from the hypothesis and the claim. Also in the third case, when $v \to_1 v'$, the thesis easily descends from the hypothesis and the claim.

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$. We have four cases to take into account: $\mathsf{Ind}(v, R, F)$ can reduce either to $\mathsf{Ind}(v, R, F')$ or to $\mathsf{Ind}(v, R', F)$ or $\mathsf{Ind}(v', R, F)$ or to $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)]$ (in case $v \equiv \mathsf{s}^{k+1}0$). In the first case, by the hypothesis and the claim, we have that $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \in [\![C]\!]$ and $R \in \mathrm{SN}_C$. Since $\mathsf{Ind}(v, R, F) \to \mathsf{Ind}(\mathsf{s}^{k+1}0, R, F) \to_1 F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F')]$, we get, by the secondary induction hypothesis, $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F')] \in [\![C]\!]$. With the same argument we get $F'[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F')] \in [\![C]\!]$. The thesis now is a consequence of the claim. For the second case, by the hypothesis and the claim, we get $F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \in [\![C]\!]$ and $R \in \mathrm{SN}_C$. Since $\mathsf{Ind}(v, R, F) \to \mathsf{Ind}(\mathsf{s}^{k+1}0, R, F) \to_1 F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R, F)] \to_1 F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R', F)]$, we can apply the secondary induction hypothesis and get

$F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, R', F)] \in [\![C]\!]$. The thesis is now a consequence of the claim when we observe that $R \in \mathrm{SN}_C$ implies $R' \in \mathrm{SN}_C$. In the third and fourth case we get the thesis as immediate consequence of the hypothesis and the claim.

- $P \equiv \mathsf{Ind}(v, R, F)$, with $\mathsf{nf}(v)$ not a numeral. We have to consider three cases according to which one among v, R, and F reduces. In the first case we get the thesis immediately by the claim. For the second case we have to consider that $R' \in \mathrm{SN}_C$, by the fact that, by the claim, $R \in \mathrm{SN}_C$. This fact enables us to infer the thesis using the claim. The third case can be proven similarly.

- $P \equiv \lambda_\forall n.P_1, \sigma_t(Q)$. In both such cases the thesis is an immediate consequence of the hypothesis and the claim.

- $P \equiv \mathsf{Ind}_i(P_1, ..., P_{n_i})$ $(1 \leqslant i \leqslant 6)$. Immediate by the claim. ▮

Extension of the proof of Lemma 5.5. We have to consider two more, pairwise symmetric, cases.

$$5.\ \begin{cases} P \equiv \lambda_\forall n.P_1,\ Q \equiv \sigma_t(Q_1),\ S \equiv P_1[t/n] \star Q \\ P \equiv \sigma_t(P_1),\ Q \equiv \lambda_\forall n.Q_1,\ S \equiv P_1 \star Q_1[t/n]. \end{cases}$$

We prove the first case, the second one is similar. By the claim we have that $P_1[t/n] \in [\![A_1]\!]$ for all t: Int, and $Q_1 \in [\![A_1^\perp]\!]$. By the induction hypothesis, $P_1[t/n] \star Q$ for all t: Int in particular for the t considered. ▮

For the proof of strong normalization for $\lambda^{\mathrm{Sym}}_{\mathbf{PA}-\mathsf{U}}$ we need to have a stronger version of Lemma 5.6.

Extended Lemma 5.6. *Let C be a type and $P \in \mathrm{Term}_C$ with $\mathrm{FV}(P) \subseteq \{x_1^{A_1}, ..., x_n^{A_n}\} \cup \{g_1, ..., g_m\}$, where x_i $(1 \leqslant i \leqslant n)$ is a term variable of type A_i and g_j $(1 \leqslant j \leqslant m)$ is a* PA*-terms variable of* PA*-type G_j. Then*

$$\forall t_1 : G_1 \cdots t_m : G_m . \forall P_1 \in [\![A_1]\!] \cdots P_n \in [\![A_n]\!].$$
$$P[t_1/g_1, ..., t_m/g_m][P_1/x_1, ..., P_n/x_n] \in [\![C]\!].$$

Proof. By double induction: primary on the structure of P and secondary on the sum of the values of numerals in P (by numerals we mean closed PA-terms of type Int). To simplify notation, we denote $[Q_1/y_1, ..., Q_k/y_k]$ by $[\mathbf{Q}/\mathbf{y}]$. We recall that, in $\lambda^{\mathrm{Sym}}_{\mathbf{PA}-\mathsf{U}}$, $C \equiv C[t/g]$ for any type C, PA-term variable g, and PA-term t.

- $P \equiv x, \langle P_1, P_2 \rangle, \sigma_i(P_i), P_1 \star P_2$. As in the proof of Lemma 5.6.

- $\lambda x.P'$. We have to prove $(\lambda x.P')[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in [\![C]\!]$. We have that $(\lambda x.P')[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \equiv \lambda x.P'[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}]$ since we can assume, by the possibility of renaming bound variables, that $x^{C^\perp} \notin \{x_1^{A_1[\mathbf{t},\mathbf{g}]}, ..., x_n^{A_n[\mathbf{t}/\mathbf{g}]}\}$. By the claim, what we have to prove is then that, for all $Q \in [\![C^\perp]\!]$, $P'[\mathbf{t}/\mathbf{g}][Q/x, \mathbf{P}/\mathbf{x}] \in [\![\perp]\!]$, a fact that follows by the induction hypothesis.

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v) \equiv 0$. By the induction hypothesis we get $R[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in [\![C]\!]$. Moreover, by Lemma 5.2, $F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in \mathrm{SN}_C$. The thesis is now a consequence of the claim, since $\mathsf{Ind}(v, R, F)[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \equiv \mathsf{Ind}(v[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], R[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}])$ and since, by the hypothesis, $\mathsf{nf}(v) \equiv \mathsf{nf}(v[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}])$.

- $P \equiv \mathsf{Ind}_{n,x}(v, R, F)$ with $\mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0$. By the induction hypothesis we get $R[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in [\![C]\!]$. By the claim, what we have to prove is $F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}][\mathsf{s}^k 0, \mathsf{Ind}_{n,x}(\mathsf{s}^k 0, R, F)[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}]] \in [\![C]\!]$. Since, by our notational convention, $F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}][\mathsf{s}^k 0, \mathsf{Ind}_{n,x}(\mathsf{s}^k 0, R, F)[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}]] \equiv F[\mathbf{t}/\mathbf{g}, \mathsf{s}^k 0/n][\mathbf{P}/\mathbf{x}, \mathsf{Ind}_{n,x}(\mathsf{s}^k 0, R, F)[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}]/x]$ and n and x are bound in $\mathsf{Ind}_{n,x}(v, R, F)$, we have that the thesis follows by the induction hypothesis once we have shown that $\mathsf{Ind}_{n,x}(\mathsf{s}^k 0, R, F)[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in [\![C]\!]$. This fact follows by the secondary induction hypothesis.

- $P \equiv \mathsf{Ind}(v, R, F)$ with $\mathsf{nf}(v)$ not a numeral. By the induction hypothesis we get $R[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in [\![C]\!]$. Hence by Lemma 5.2, $R[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}], F[\mathbf{t}/\mathbf{g}][\mathbf{P}/\mathbf{x}] \in \mathrm{SN}_C$. The thesis now is a consequence of the claim.

- $P \equiv \mathsf{PA}_i(P_1, ..., P_{n_i})$ $(1 \leqslant i \leqslant 6)$. In all these cases the thesis follows by the claim and the induction hypothesis. ▮

Proof of Theorem 3.5. Let $P : C$ be a term with $\mathrm{FV}(P) = \{x_1^{A_1}, ..., x_n^{A_n}\} \cup \{g_1, ..., g_m\}$. By Claim 5.1(i) we get $x_i \in [\![A_i]\!]$ $(i = 1, ..., n)$ and therefore, by Lemma 5.6 $P \equiv P[x_1/x_1, ..., x_n/x_n][g_1/g_1, ..., g_m/g_m] \in [\![C]\!]$. Lemma 5.2 allows now to infer $P \in \mathrm{SN}_C$.

Symmetric Candidates for $\lambda^{\mathrm{Sym}}_{\mathbf{PA}-\mathsf{U}}$. We extend now the notion of symmetric candidate given in Subsection 5.1, in such a way it can satisfy the additional clauses of the claim.

Definition 5.7 is extended with the additional operators

$$\begin{aligned}
&\mathrm{Ind}_{0,A}, \mathrm{Ind}_{\mathsf{s},A}, \\
&\mathrm{Ind}_{\bullet,A}, \mathrm{Ind}_A, \mathrm{IND}_A : && \mathscr{P}(\mathrm{Term}_A) \to \mathscr{P}(\mathrm{Term}_A) \\
&\mathrm{ForAll}_{n,A} : && \mathscr{P}(\mathrm{Term}_A) \to \mathscr{P}(\mathrm{Term}_{\forall n \cdot A}) \\
&\mathrm{Exists}_{n,A} : && \mathscr{P}(\mathrm{Term}_A) \to \mathscr{P}(\mathrm{Term}_{\exists n \cdot A}) \\
&\mathrm{Ax}_{\mathsf{PA}_i} : && \mathscr{P}(\mathrm{Term}_{\mathsf{U}}) \times \cdots \times \mathscr{P}(\mathrm{Term}_{\mathsf{U}}) \\
& && \quad \to \mathscr{P}(\mathrm{Term}_{\mathsf{U}}) \quad (1 \leqslant i \leqslant 6),
\end{aligned}$$

which are defined as follows:

$$\begin{aligned}
&\mathrm{Ind}_{0,A}(X) \\
&\quad =_{\mathrm{Def}} \{\mathsf{Ind}(v, P, F) : A \mid \mathsf{nf}(v) \equiv 0, \\
&\qquad P \in X \cap \mathrm{SN}_A, F \in \mathrm{SN}_A\}
\end{aligned}$$

$$\mathrm{Ind}_{\mathsf{s},A}(X)$$
$$=_{\mathrm{Def}} \{\mathsf{Ind}(v,P,F)\colon A \mid \mathsf{nf}(v) \equiv \mathsf{s}^{k+1}0,\ P \in \mathrm{SN}_A,$$
$$F[\mathsf{s}^k 0, \mathsf{Ind}(\mathsf{s}^k 0, P, F)] \in X \cap \mathrm{SN}_A\}$$
$$\mathrm{Ind}_{\bullet,A}(X)$$
$$=_{\mathrm{Def}} \{\mathsf{Ind}(v,P,F)\colon A \mid \mathsf{nf}(v) \text{ not a numeral},$$
$$P, F \in \mathrm{SN}_A\}$$
$$\mathrm{Ind}_A(X)$$
$$=_{\mathrm{Def}} \mathrm{Ind}_{0,A}(X) \cup \mathrm{Ind}_{\mathsf{s},A}(X) \cup \mathrm{Ind}_{\bullet,A}(X)$$
$$\mathrm{IND}_A(X)$$
$$=_{\mathrm{Def}} \text{least fixed-point of the increasing}$$
$$\text{map } Y \mapsto X \cup \mathrm{Ind}_A(Y)$$
$$\mathrm{ForAll}_{n,A}(X)$$
$$=_{\mathrm{Def}} \{\lambda_\forall n.P\colon \forall n.A \mid \forall t\colon \mathrm{Int}.P[t/n] \in X\}$$
$$\mathrm{Exists}_{n,A}(X)$$
$$=_{\mathrm{Def}} \{\sigma_t(P)\colon \exists n.A \mid t\colon \mathrm{Int}, P \in X\}$$
$$\mathrm{Ax}_{\mathsf{PA}_i}(X_1, ..., X_{n_i})$$
$$=_{\mathrm{Def}} \{\mathsf{PA}_i(P_1, ..., P_{n_i}) \mid P_1 \in X_1, ..., P_{n_i} \in X_{n_i}\}$$
$$(1 \leqslant i \leqslant 6)$$

Because of the presence of terms we define, besides the operator Neg, whose definition is also extended, the operator NEG. A candidate is now a fixed-point of the composition of NEG with itself.

DEFINITION (Symmetric Candidates for $\lambda^{\mathrm{Sym}}_{\mathbf{PA}-\mathsf{U}}$). Let A be an m-type. By induction on the structure of A we simultaneously define

(a) The operators

$$\begin{cases}\mathrm{Neg}_A\colon \mathscr{P}(\mathrm{Terms}_{A^\perp}) \to \mathscr{P}(\mathrm{Terms}_A)\\ \mathrm{Neg}_{A^\perp}\colon \mathscr{P}(\mathrm{Terms}_A) \to \mathscr{P}(\mathrm{Terms}_{A^\perp})\end{cases}$$

(b) The operator

$$\mathrm{NEG}\colon \mathscr{P}(\mathrm{Terms}_A) \to \mathscr{P}(\mathrm{Terms}_{A^\perp})$$

(c) The sets $[\![A]\!]$ and $[\![A^\perp]\!]$

as follows:

(a) Let $X \subseteq \mathrm{Term}_A$ and $Y \subseteq \mathrm{Term}_{A^\perp}$. By the involutive property of negation in our system it is not restrictive to consider only the cases A atomic, A conjunction, and A universally quantified formula:

— $A \equiv \mathsf{U}$.

$$\begin{cases}\mathrm{Neg}_{\mathsf{U}}(Y) =_{\mathrm{Def}} \mathrm{Var}_{\mathsf{U}} \cup \left(\bigcup_{i=1}^{6} \mathrm{Ax}_{\mathsf{PA}_i}(\mathrm{SN}_{\mathsf{U}}, ..., \mathrm{SN}_{\mathsf{U}})\right)\\ \qquad\qquad \cup \mathrm{Lambda}_{\mathsf{U}}(Y)\\ \mathrm{Neg}_{\mathsf{U}^\perp}(X) =_{\mathrm{Def}} \mathrm{Var}_{\mathsf{U}^\perp} \cup \mathrm{Lambda}_{\mathsf{U}^\perp}(X).\end{cases}$$

— $A \equiv A_1 \wedge A_2$ ($A^\perp \equiv A_1^\perp \vee A_2^\perp$).

$$\begin{cases}\mathrm{Neg}_A(Y) =_{\mathrm{Def}} \mathrm{Var}_A \cup \mathrm{Pair}_A([\![A_1]\!], [\![A_2]\!])\\ \qquad\qquad \cup \mathrm{Lambda}_A(Y)\\ \mathrm{Neg}_{A^\perp}(X) =_{\mathrm{Def}} \mathrm{Var}_{A^\perp} \cup \left(\bigcup_{i=1}^{2} \mathrm{Sigma}^i([\![A_i^\perp]\!])\right)\\ \qquad\qquad \cup \mathrm{Lambda}_{A_1^\perp \vee A_2^\perp}(X).\end{cases}$$

— $A \equiv \forall n.A_1$ ($A^\perp \equiv \exists n.A_1^\perp$).

$$\begin{cases}\mathrm{Neg}_A(Y) =_{\mathrm{Def}} \mathrm{Var}_A \cup \mathrm{ForAll}_{n,A_1}([\![A_1]\!]) \cup \mathrm{Lambda}_A(Y)\\ \mathrm{Neg}_{A^\perp}(X) =_{\mathrm{Def}} \mathrm{Var}_{A^\perp} \cup \mathrm{Exists}_{n,A^\perp}([\![A_1^\perp]\!])\\ \qquad\qquad \cup \mathrm{Lambda}_{A_1^\perp \vee A_2^\perp}(X).\end{cases}$$

(b) For all A, once we have defined Neg_A, we can define NEG_A as follows: Let $X \subseteq \mathrm{Term}_{A^\perp}$;

$$\mathrm{NEG}_A(X) =_{\mathrm{Def}} \mathrm{IND}_A(\mathrm{Neg}_A(X)).$$

(c) For all A, Neg_A is a decreasing operator (w.r.t. set theoretical inclusion), since Lambda_A is so. Moreover, since IND_A is increasing, NEG_A is decreasing as well. Then, once one has defined NEG_A for some A, it is possible to get, by Tarsky's Fixed-Point Theorem, the smallest fixed-point of the (increasing) operator $\mathrm{NEG}_A \circ \mathrm{NEG}_{A^\perp}$. Let us call it X_0. We then define

$$\begin{cases}[\![A]\!] =_{\mathrm{Def}} X_0\\ [\![A^\perp]\!] =_{\mathrm{Def}} \mathrm{NEG}_{A^\perp}(X_0).\end{cases}$$

We extend the definition of a computable set of terms to all types by defining $[\![\perp]\!] =_{\mathrm{Def}} \mathrm{SN}_\perp$.

The clauses of the claim are now an immediate consequence of the definition of $[\![A]\!]$, because of the definition of NEG_A and the fact that, being a fixed-point, we have that $[\![A]\!] \equiv \mathrm{NEG}_A([\![A^\perp]\!])$.

We recall, once again, that the proof provided works also for $\lambda^{\mathrm{Sym}}_{\Delta}$ for any Δ.

ACKNOWLEDGMENTS

We thank professor Mariangiola Dezani and professor Mario Coppo for their constant scientific support, Steffen van Bakel for his careful reading, and the anonymous referees for helpful comments. The first author expresses his gratitude also to Elena and Uliana Barbanera for their steadfast encouragement.

Received September 1994; final manuscript received January 4, 1994

REFERENCES

[BB 91] Barbanera, F., and Berardi, S. (1993), Witness extraction in classical logic through normalization, *in* "Logical Environments" (G. Huet and G. Plotkin, Eds.), Cambridge Univ. Press, London/New York.

[BB 92] Barbanera, F., and Berardi, S. (1992), A constructive valuation interpretation for classical logic and its use in witness extraction, *in* "Proceedings of Colloquium on Trees in Algebra and Programming (CAAP)," Lecture Notes in Computer Science, Vol. 581, Springer-Verlag, Berlin/New York.

[BB 93] Barbanera, F., and Berardi, S. (1993), Extracting constructive content from classical proofs via control like reductions, *in* "Proceedings of the International Conference on Typed Lambda Calculus and Applications (TLCA) 93," Lecture Notes in Computer Science, Springer-Verlag, Berlin/New York.

[Con 86] Constable, R. L., Allen, S., Bromely, H., Cleaveland, W. *et al.* (1986), "Implementing Mathematics with the Nuprl Proof Development System," Prentice–Hall, Englewood Cliffs, NJ.

[Coq 92] Coquand, T. (1995), A semantics of evidence for classical arithmetic, *J. Symbolic Logic* **60**, 325–337.

[Fri 78] Friedman, H. (1978), Classically and intuitionistically provably recursive functions, *in* "Higher Set Theory" (D. S. Scott and G. H. Muller, Eds.), pp. 21–28, Lecture Notes in Mathematics, Vol. 699, Springer-Verlag, Berlin/New York.

[Gir 72] Girard, J.-Y. (1972), "Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérièur," Thése de Doctorat d'Etat, University of Paris.

[Gri 90] Griffin, T. G. (1990), A formulae-as-types notion of control, *in* "Proceedings of the Seventeenth Annual ACM Symposium on Principles of Programming Languages."

[Kre 58] Kreisel, G. (1958), Mathematical significance of consistency proofs, *J. Symbolic Logic* **23**, 155–182.

[Mur 90] Murthy, C. (1990), "Extracting Constructive Content from Classical Proofs," Ph.D. Thesis, 90-1151, Department of Computer Science, Cornell University.

[NPS 90] Nordström, B., Petersson, K., and Smith, J. (1990), "Programming in Martin–Löf's Type Theory," Oxford Univ. Press, London.

[Par 92] Parigot, M., $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction, *in* "Proceedings LPAR'92," pp. 190–201, Lecture Notes in Computer Science, Vol. 624, Springer-Verlag, Berlin/New York.

[PN 90] Paulson L. C., and Nipkow, T. (1990), "Isabelle Tutorial and User's Manual," Technical Report 189, Univ. of Cambridge.

[Pra 65] Prawitz, D. (1965), "Natural Deduction, A Proof Theoretical Study," Almqvist–Winskell, Stockholm.

[Pra 81] Prawitz, D. (1981), Validity and normalizability of proofs in 1st and 2nd order classical and intuitionistic logic, *in* "Atti del I Congresso Italiano di Logica," pp. 11–36, Bibliopolis, Naples.

[Schi 95] Schivalocchi, M. (1996), "$\lambda^{\mathrm{Sym}}_{\mathbf{PA}}$: Non confluenza per la logica classica," Master's thesis, Università di Trento.

[Tait 67] Tait, W. W. (1967), Intensional interpretation of functional of finite types, *J. Symbolic Logic* **32**, 198–212.