Complete Types in an Extension of the System AF2

Samir Farkh, Karim Nour

HAL Id: hal-00380721
https://hal.archives-ouvertes.fr/hal-00380721

Complete Types in an Extension of the System $\mathcal{AF}2$

Samir Farkh — **Karim Nour**

LAMA - Équipe de Logique
Université de Chambéry
73376 Le Bourget du Lac
nour@univ-savoie.fr

ABSTRACT. *In this paper, we extend the system $\mathcal{AF}2$ in order to have the subject reduction for the $\beta\eta$-reduction. We prove that the types with positive quantifiers are complete for models that are stable by weak-head expansion.*

KEYWORDS: *system $\mathcal{AF}2$, type with positive quantifier, complete type.*

1. Introduction

The semantics of realisability of the system $\mathcal{F}$, proposed by J.-Y. Girard, consists in interpreting the types by "saturated subsets" of λ-terms. The correction theorem (also called "adequacy lemma") stipulates that: if a λ-term is typable then it belongs to the interpretation of its type. The adequacy lemma allows to show the strong normalization of the system $\mathcal{F}$ when we take an adequate concept of saturation. The power of this notion of semantics comes from the variety of possible interpretations of the second order quantifier. For the system $\mathcal{AF}2$, J.-L. Krivine proposed a more general semantics by defining the concept of λ-models for a second-order language. His semantics is a modification of the traditional concept of a second-order model in which the set of the truth values is not, as usual, $\{0, 1\}$ but an adequate subset of λ-terms (see [KRI 94] and [RAF 98]). The corresponding adequacy lemma allows also to prove the uniqueness of the representation of the data.

Many researchers were interested in finding a general definition of a data type. For example, Böhm and Berarducci gave such a definition, only for term algebras, in the system $\mathcal{F}$ (see [BÖH 85]) and Krivine generalized their definition to system $\mathcal{AF}2$ (see [KRI 90]). We noticed that the class $\mathcal{A}$ of the types thus built has the following feature: a normal λ-term is typable of a type $D \in \mathcal{A}$ iff it is in the interpretation of D for a certain semantics. Then, we decided to take this result as the definition of the data types which we called "complete types", because the considered semantics is complete

for these types. R. Labib-Sami was the first to build a class of complete types: they are the types with positive quantifiers (denoted by $\forall_2^+$) of the system $\mathcal{F}$ compared to a semantics based on the subsets saturated by $\beta\eta$-equivalence (see [LAB 86]).

We generalized in [FAR 98] Labib-Sami's result, by showing that the $\forall_2^+$ types of system $\mathcal{AF}2$ are complete for the semantics based on sets saturated by $\beta\eta$-equivalence. It was natural to imagine a refinement of this result, namely interpretation of the types by sets saturated by weak-head expansion. For this, we considered a more restricted class of the $\forall_2^+$ types which includes the data types of J.-L. Krivine. Then, we showed in [FAR 98] that these types are preserved by η-reduction and are complete for the considered semantics.

We propose in this paper another solution to this problem. We add typing rules to the system $\mathcal{AF}2$ in order to have the conservation of types by $\beta\eta$-reduction. The system, which we propose, is inspired by the works of Mitchell [MIT 88] and the second author [NOU 96]. We show that, in this new system, all $\forall_2^+$ types are complete for the semantics based on sets saturated by weak-head expansion.

2. Notations and definitions

NOTATIONS 1. — *We denote by Λ the set of terms of pure λ-calculus, also called λ-terms. Let $t, u, u_1, \ldots, u_n \in \Lambda$, the application of t to u is denoted by $(t)u$. In the same way we write $(t)u_1 \ldots u_n$ instead of $(\ldots((t)u_1)\ldots)u_n$. The β-reduction (resp. β-equivalence) is denoted by $t \rightarrow_\beta u$ (resp. $t \simeq_\beta u$). The set of free variables of a λ-term t is denoted by $Fv(t)$. Let us recall that a λ-term t either has a **weak-head redex** [i.e. $t = (\lambda xu)vv_1 \ldots v_m$, the weak-head redex being $(\lambda xu)v$], or is in **weak-head normal form** [i.e. $t = (x)vv_1 \ldots v_m$ or $t = \lambda xu$]. The notation $t \succ_f t'$ means that t' is obtained from t by some **weak-head reductions**.*

2.1. The $\mathcal{AF}2$ type system

The types will be formulas of second-order predicate logic over a given language. The logical symbols are $\bot$ (for absurd), $\rightarrow$ and $\forall$ (and no other ones). There are individual variables: $x, y, \ldots$ (also called first-order variables) and n-ary predicate variables ($n = 0, 1, \ldots$): $X, Y, \ldots$ (also called second-order variables). The terms and formulas are built in the usual way.

If X is a unary predicate variable, t and t' two terms, then the formula $\forall X[Xt \rightarrow Xt']$ is denoted by $t = t'$, and is said to be an **equation**. A **particular case** of $t = t'$ is a formula of the form $t[u_1/x_1, \ldots, u_n/x_n] = t'[u_1/x_1, \ldots, u_n/x_n]$ or $t'[u_1/x_1, \ldots, u_n/x_n] = t[u_1/x_1, \ldots, u_n/x_n]$, $u_1, \ldots, u_n$ being terms of the language. Then, we denote by **E** a system of function equations. A context Γ is a set of the form $x_1 : A_1, \ldots, x_n : A_n$ where $x_1, \ldots, x_n$ are distinct variables and $A_1, \ldots, A_n$ are formulas. We are going to describe a system of typed λ-calculus called second-

order functional arithmetic (abbreviated in $\mathcal{AF}2$ for **Arithmétique Fonctionnelle du second ordre**). The typing rules are the following:

(1) $\Gamma, x : A \vdash_{\mathcal{AF}2} x : A$.

(2) If $\Gamma, x : B \vdash_{\mathcal{AF}2} t : C$, then $\Gamma \vdash_{\mathcal{AF}2} \lambda x t : B \to C$.

(3) If $\Gamma \vdash_{\mathcal{AF}2} u : B \to C$, and $\Gamma \vdash_{AF2} v : B$, then $\Gamma \vdash_{\mathcal{AF}2} (u)v : C$.

(4) If $\Gamma \vdash_{\mathcal{AF}2} t : A$, and x does not appear in Γ, then $\Gamma \vdash_{\mathcal{AF}2} t : \forall x A$.

(5) If $\Gamma \vdash_{\mathcal{AF}2} t : \forall x A$, then, for every term u, $\Gamma \vdash_{\mathcal{AF}2} t : A[u/x]$.

(6) If $\Gamma \vdash_{\mathcal{AF}2} t : A$, and X does not appear in Γ, then $\Gamma \vdash_{\mathcal{AF}2} t : \forall X A$.

(7) If $\Gamma \vdash_{\mathcal{AF}2} t : \forall X A$, then, for every formula G,

$\Gamma \vdash_{\mathcal{AF}2} t : A[G/X(x_1, \ldots, x_n)]$.[1]

(8) If $\Gamma \vdash_{\mathcal{AF}2} t : A[u/x]$, then $\Gamma \vdash_{\mathcal{AF}2} t : A[v/x]$, $u = v$ being a particular case of an equation of **E**.

Whenever we obtain the typing $\Gamma \vdash_{\mathcal{AF}2} t : A$ by means of these rules, we say that "the λ-term t is of type A in the context Γ, with respect to the equations of **E**".

THEOREM 2. —

1) If $\Gamma \vdash_{\mathcal{AF}2} t : A$, and $t \to_\beta t'$, then $\Gamma \vdash_{\mathcal{AF}2} t' : A$.

2) If $\Gamma \vdash_{\mathcal{AF}2} t : A$, then t is strongly normalizable.

2.2. *The semantics of $\mathcal{AF}2$*

If $G, G' \in P(\Lambda)$, we define an element of $P(\Lambda)$ by: $G \to G' = \{u \in \Lambda \; / \; (u)t \in G' \text{ for every } t \in G\}$. Let $\mathcal{R}_f$ the set of subsets of Λ stable by weak-head reduction (i.e. $\Xi \in \mathcal{R}_f$ iff for every $v \in \Xi$, if $u \succ_f v$, then $u \in \Xi$). A subset R of $\mathcal{R}_f$ is said **adequate** iff R is closed by $\to$ and $\cap$.

Let L be a second-order language. A Λ_f-**model** is defined by:

– a non empty set $|M|$ called **domain** of M,

– an adequate set R of $\mathcal{R}_f$,

– for every an n-ary function symbol of L, a function $f_M : |M|^n \to |M|$,

– for every n-ary predicate symbol P of L, a function $P_M : |M|^n \to R$.

Let M be a Λ-model of L.

– An **interpretation** I is a function from the set of first (resp. the set of n-ary second) order variables to $|M|$ (resp. to $R^{|M|^n}$).

– Let I be an interpretation, x (resp. X) a first (resp. an n-ary second) order variable, and a (resp. Φ) an element of $|M|$ (resp. of $R^{|M|^n}$). We define an interpretation

---

1. $A[G/X(x_1, \ldots, x_n)]$ is obtained by replacing in A each atomic formula $X(t_1, \ldots, t_n)$ by $G[t_1/x_1, \ldots, t_n/x_n]$. To simplify, we write $A[G/X]$ instead of $A[G/X(x_1, \ldots, x_n)]$.

$J = I[x \leftarrow a]$ (resp. $J = I[X \leftarrow \Phi]$) by taking $J(x) = a$ (resp. $J(X) = \Phi$) and $J(\xi) = I(\xi)$ (resp. $J(\xi') = I(\xi')$) for every variables $\xi \neq x$ (resp. $\xi' \neq X$).

Let I be an interpretation. To every term t of L, we define, by induction, its **value** $t_{M,I} \in |M|$:

– if $t = x$, then $t_{M,I} = I(x)$,
– if $t = f(t^1, \ldots, t^n)$, then $t_{M,I} = f_M(t^1_{M,I}, \ldots, t^n_{M,I})$.

Let A be a formula of L. The **value** of A in a model M and an interpretation I (denoted by $|A|_{M,I}$) is an element of R defined by induction:

– if $A = P(t^1, \ldots, t^n)$, where P is a n-ary predicate symbol (resp. second-order variable) and $t^1, \ldots, t^n$ are terms of L, then $|A|_{M,I} = P_M(t^1_{M,I}, \ldots, t^n_{M,I})$ (resp. $|A|_{M,I} = I(X)(t^1_{M,I}, \ldots, t^n_{M,I})$).
  – if $A = B \rightarrow C$, then $|A|_{M,I} = |B|_{M,I} \rightarrow |C|_{M,I}$,
  – if $A = \forall x B$ where x is a first-order variable, then $|A|_{M,I} = \bigcap\{|B[x]|_{M,I[x \leftarrow a]}; a \in |M|\}$,
  – if $A = \forall X B$ where X is a n-ary second-order variable, then $|A|_{M,I} = \bigcap\{|B[X]|_{M,I[X \leftarrow \Phi]}; \Phi \in R^{|M|^n}\}$.

It is clear that: if A is a closed type, then $|A|_{M,I}$ does not depend on the interpretation I and we write $|A|_M$.

Let M be a Λ-model of L.

– We say that M **satisfies** the equation $u = v$, if for every interpretation I, $u_{M,I} = v_{M,I}$. If E is a set of equations of L, we say that M **satisfies** E, or M is a **model** for E, iff M satisfies all the equations of E.
– If A is a closed formula, we denote by $|A|_f = \bigcap\{|A|_M; M$ is a Λ_f-model which satisfies $E\}$.

The following theorem is known under the name "adequation lemma" or "the correction theorem":

THEOREM 3. — *Let t be a λ-term and A a closed type of system $\mathcal{AF}2$. If $\vdash_{\mathcal{AF}2} t : A$, then $t \in |A|_f$.*

3. The system $\mathcal{AF}2_\subseteq$

The typing system $\mathcal{AF}2$ does not conserve the types by η-reduction. Indeed, $\vdash_{\mathcal{AF}2} \lambda x \lambda y(x)y : \forall X(X \rightarrow (X \rightarrow X)) \rightarrow (\forall X X \rightarrow \forall X(X \rightarrow X))$ but $\lambda x \lambda y(x)y \rightarrow_\eta \lambda x x$ and $\nvdash_{\mathcal{AF}2} \lambda x x : \forall X(X \rightarrow (X \rightarrow X)) \rightarrow (\forall X X \rightarrow \forall X(X \rightarrow X))$. We

will define an extension of the system $\mathcal{AF}2$ which, while keeping the properties of the system $\mathcal{AF}2$, conserves the types by η-reduction.

DEFINITION 4. — *Let E be an equation system of second-order language L. We define on the formulas of $\mathcal{AF}2$ a binary relation $\subseteq$ by: $A \subseteq B$ iff it is obtained by the following proof rules:*

(ax') $A \subseteq A$

$(dist)$ $\forall\xi(C \to D) \subseteq \forall\xi C \to \forall\xi D$

$(\to)$ *If $C' \subseteq C$ and $D \subseteq D'$, then $C \to D \subseteq C' \to D'$*

$(\forall_e)$ *If $A \subseteq \forall\xi C$, then $A \subseteq C[F/\xi]$*

$(\forall_i)$ *If $A \subseteq D$ and ξ is not free in A, then $A \subseteq \forall\xi D$*

(tr) *If $A \subseteq D$ and $D \subseteq B$, then $A \subseteq B$*

(e) *If $A \subseteq D[u/y]$ and $u = v$ is a particular case of an equation of E, then $A \subseteq D[v/y]$*

DEFINITION 5. — *The system $\mathcal{AF}2_\subseteq$ is the system $\mathcal{AF}2$ where we add the following rule:*

$$\text{If } \Gamma \vdash_{\mathcal{AF}2_\subseteq} t : A \text{ and } A \subseteq B, \text{ then } \Gamma \vdash_{\mathcal{AF}2_\subseteq} t : B \qquad (\subseteq)$$

It is clear that the rules (5), (7) and (8) are particular cases of the rule $(\subseteq)$.

3.1. *Syntactical properties of the system*

NOTATIONS 6. — *Let $\boldsymbol{\xi} = \xi_1, \ldots, \xi_n$ be a sequence of variables. We denote the formula $\forall\xi_1 \ldots \forall\xi_n F$ by $\forall\boldsymbol{\xi}F$. We write "$\boldsymbol{\xi}$ is not free in A" if for every $1 \leq i \leq n$, ξ_i is not free in A. Let A be a formula, $\boldsymbol{F}$ a sequence of formulas $F_1, \ldots, F_n$, $\boldsymbol{u}$ a sequence of terms $u_1, \ldots, u_n$ and $\boldsymbol{x}$ (resp. $\boldsymbol{X}$) a sequence of first (resp second) ordre variables $x_1, \ldots, x_n$ (resp. $X_1, \ldots, X_n$). We denote by $A[\boldsymbol{u}/\boldsymbol{x}]$ the formula $A[u_1/x_1, \ldots, u_n/x_n]$ and by $A[\boldsymbol{F}/\boldsymbol{X}]$ the formula $A[F_1/X_1, \ldots, F_n/X_n]$.*

LEMMA 7. — *In the typing, we may replace the succession of n times $(\subseteq)$ and m times (4) and (6), by the succession of m times (4) and (6), and n times $(\subseteq)$.*

PROOF. — By induction on n and m. ∎

We deduce the following corollary:

COROLLARY 8. — *If $\Gamma \vdash_{\mathcal{AF}2_\subseteq} t : B$ is derived from $\Gamma \vdash_{\mathcal{AF}2_\subseteq} t : A$, then we may assume that we begin by the applications of (4), (6) and next $(\subseteq)$ (i.e. there is $\boldsymbol{\xi}$ not free in Γ such that $\forall\boldsymbol{\xi}A \subseteq B$).*

Then we have the following characterization:

THEOREM 9. —

(i) If $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} x : A$, then there is a type B such that $x : B \in \Gamma$ and $\forall \boldsymbol{\xi} B \subseteq A$, where $\boldsymbol{\xi}$ is not free in Γ.

(ii) If $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} \lambda x u : A$, then there are two types B and C such that $\Gamma, x : B \vdash_{\mathcal{AF}2_{\subseteq}} u : C$ and $\forall \boldsymbol{\xi} (B \rightarrow C) \subseteq A$, where $\boldsymbol{\xi}$ is not free in Γ.

(iii) If $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} (u)v : A$, then there are two types B and C such that $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} u : B \rightarrow C$, $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} v : B$ and $\forall \boldsymbol{\xi} C \subseteq A$, where $\boldsymbol{\xi}$ is not free in Γ.

We will define a typing system equivalent to system $\mathcal{AF}2_{\subseteq}$.

DEFINITION 10. — *The system $\mathcal{AF}2S$ is defined only by the three following rules:*

(S1) *If $x : B \in \Gamma$ and $\forall \boldsymbol{\xi} B \subseteq A$, then $\Gamma \vdash_{\mathcal{AF}2S} x : A$*

(S2) *If $\Gamma, x : B \vdash_{\mathcal{AF}2S} u : C$ and $\forall \boldsymbol{\xi} (B \rightarrow C) \subseteq A$, then $\Gamma \vdash_{\mathcal{AF}2S} \lambda x u : A$*

(S3) *If $\Gamma \vdash_{\mathcal{AF}2S} u : B \rightarrow C$, $\Gamma \vdash_{\mathcal{AF}2S} v : B$ and $\forall \boldsymbol{\xi} C \subseteq A$, then $\Gamma \vdash_{\mathcal{AF}2S} (u)v : A$*

where $\boldsymbol{\xi}$ is not free in Γ.

We have the following result:

THEOREM 11. — $\Gamma \vdash_{\mathcal{AF}2_{\subseteq}} t : A$ *iff* $\Gamma \vdash_{\mathcal{AF}2S} t : A$.

PROOF. — We use Theorem 9.   ∎

In the rest of the paper we often consider the system $\mathcal{AF}2S$.

The following corollary will often be used:

COROLLARY 12. — *If $\Gamma, x : A \vdash_{\mathcal{AF}2S} (x)u_1 \ldots u_n : B$, then*
$n = 0$, $\forall \boldsymbol{\xi_0} A \subseteq B$ *and $\boldsymbol{\xi_0}$ does not appear in Γ and A, or*
$n \neq 0$, $\forall \boldsymbol{\xi_0} A \subseteq C_1 \rightarrow B_1$, $\forall \boldsymbol{\xi_i} B_i \subseteq C_{i+1} \rightarrow B_{i+1}$ $(1 \leq i \leq n-1)$, *and* $\forall \boldsymbol{\xi_n} B_n \subseteq B$
where $\boldsymbol{\xi_i}$ $(0 \leq i \leq n)$ are not free in Γ and A, and $\Gamma, x : A \vdash_{\mathcal{AF}2S} u_i : C_i$
$(1 \leq i \leq n)$.

PROOF. — By induction on n and using Theorem 9.   ∎

3.2. *Conservation of type by β-reduction*

LEMMA 13. — *If $A \subseteq B$, then, for any sequence of terms $\boldsymbol{u}$ (resp. of formulas $\boldsymbol{F}$), $A[\boldsymbol{u}/\boldsymbol{x}] \subseteq B[\boldsymbol{u}/\boldsymbol{x}]$ (resp. $A[\boldsymbol{F}/\boldsymbol{X}] \subseteq B[\boldsymbol{F}/\boldsymbol{X}]$), and we use the same proof rules.*

PROOF. — By induction on the derivation $A \subseteq B$. ∎

LEMMA 14. — *If* $\Gamma \vdash_{\mathcal{AF}2S} t : A$, *then, for all sequences of terms* $\boldsymbol{u}$ *(resp. of formulas* $\boldsymbol{F}$*),* $\Gamma[\boldsymbol{u}/\boldsymbol{x}] \vdash_{\mathcal{AF}2S} t : A[\boldsymbol{u}/\boldsymbol{x}]$ *(resp.* $\Gamma[\boldsymbol{F}/\boldsymbol{Y}] \vdash_{\mathcal{AF}2S} t : A[\boldsymbol{F}/\boldsymbol{Y}]$*) and we use the same typing rules.*

PROOF. — By induction on the derivation $\Gamma \vdash_{\mathcal{AF}2S} t : A$. We look at the last rule used and we use Lemma 13. ∎

LEMMA 15. — *If* $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathcal{AF}2S} t : A$, $B_i \subseteq A_i$ $(1 \leq i \leq n)$ *et* $A \subseteq B$, *then* $x_1 : B_1, \ldots, x_n : B_n \vdash_{\mathcal{AF}2S} t : B$.

PROOF. — By induction on the λ-term t. ∎

LEMMA 16. — *If* $\Gamma, x : B \vdash_{\mathcal{AF}2S} u : A$ *et* $\Gamma \vdash_{\mathcal{AF}2S} v : B$, *then* $\Gamma \vdash_{\mathcal{AF}2S} u[v/x] : A$.

PROOF. — By induction on the derivation $\Gamma, x : B \vdash_{\mathcal{AF}2S} u : A$. ∎

LEMMA 17. — *If* $\Gamma, x : C \vdash_{\mathcal{AF}2S} u : D$ *and there is a* $\boldsymbol{\xi}$ *which does not appear in* Γ *and* $\boldsymbol{\xi}'$ *such that* $\forall \boldsymbol{\xi}(C \to D) \subseteq \forall \boldsymbol{\xi}'(A \to B)$*, then* $\Gamma, x : A \vdash_{\mathcal{AF}2S} u : B$.

PROOF. — By induction on the derivation $\forall \boldsymbol{\xi}(C \to D) \subseteq \forall \boldsymbol{\xi}'(A \to B)$. We look at the last rule used. We consider only three cases.

($\to$)   We have $A \subseteq C$ and $D \subseteq B$, then, by Lemma 15, we deduce the result.

(e)   We have $\forall \boldsymbol{\xi}(C \to D) \subseteq E[u/y] = \forall \boldsymbol{\xi}'(F[u/y] \to M[u/y])$. Then $A = F[v/y]$ and $B = M[v/y]$ where $u = v$ is a particular case of an equation of E. By induction hypothesis, we obtain $\Gamma, x : F[u/y] \vdash_{\mathcal{AF}2S} u : M[u/y]$. But $F[v/y] \subseteq F[u/y]$ and $M[u/y] \subseteq M[v/y]$, then, by Lemma 15, $\Gamma, x : A \vdash_{\mathcal{AF}2S} u : B$.

($\forall_e$)   We have $\forall \boldsymbol{\xi}(C \to D) \subseteq \forall s \forall \boldsymbol{\xi}'(E \to F)$ and $A = E[G/s]$, $B = F[G/s]$. By induction hypothesis, we obtain $\Gamma, x : E \vdash_{\mathcal{AF}2S} u : F$. We may assume that s is not free in Γ, then, by Lemma 14, $\Gamma, x : E[G/s] \vdash_{\mathcal{AF}2S} u : F[G/s]$, i.e $\Gamma, x : A \vdash_{\mathcal{AF}2S} u : B$.

∎

LEMMA 18. — *If* $\Gamma \vdash_{\mathcal{AF}2S} \lambda x u : A \to B$, *then* $\Gamma, x : A \vdash_{\mathcal{AF}2S} u : B$.

PROOF. — We have $\Gamma \vdash_{\mathcal{AF}2S} \lambda x u : A \to B$, then $\Gamma, x : C \vdash_{\mathcal{AF}2S} u : D$ and $\forall \boldsymbol{\xi}(C \to D) \subseteq (A \to B)$ where $\boldsymbol{\xi}$ is not free in Γ. Therefore, by Lemma 17, $\Gamma, x : A \vdash_{\mathcal{AF}2S} u : B$. ∎

THEOREM 19. — *If* $\Gamma \vdash_{\mathcal{AF}2S} t : A$ *and* $t \to_\beta t'$, *then* $\Gamma \vdash_{\mathcal{AF}2S} t' : A$.

PROOF. — It suffices to do the proof for one step of reduction. We proceed by induction on t et we use Lemmas 16 and 18. ∎

3.3. *Conservation of type by η-reduction*

THEOREM 20. — *If $\Gamma \vdash_{\mathcal{AF}2S} t : A$ and $t \rightarrow_\eta t'$, then $\Gamma \vdash_{\mathcal{AF}2S} t' : A$.*

PROOF. — It suffices to do the proof for one step of η-reduction denoted η_0. We do the proof by induction on t. The only difficult case is $t = \lambda x u$, then two cases can arise:

1) $t' = \lambda x u'$ where $u \rightarrow_{\eta_0} u'$: We have $\Gamma \vdash_{\mathcal{AF}2S} \lambda x u : A$, then $\Gamma, x : B \vdash_{\mathcal{AF}2S} u : C$ and $\forall \boldsymbol{\xi}(B \rightarrow C) \subseteq A$ where $\boldsymbol{\xi}$ is not free in Γ. By induction hypothesis, we have $\Gamma, x : B \vdash_{\mathcal{AF}2S} u' : C$, and, by the rule $(S2)$, $\Gamma \vdash_{\mathcal{AF}2S} \lambda x u' : A$, i.e $\Gamma \vdash_{\mathcal{AF}2S} t' : A$.

2) $u = (t')x$ where x is not free in t': We have $\Gamma, x : B \vdash_{\mathcal{AF}2S} (t')x : C$, and $\forall \boldsymbol{\xi}(B \rightarrow C) \subseteq A$ where $\boldsymbol{\xi}$ is not free in Γ. Then $\Gamma, x : B \vdash_{\mathcal{AF}2S} t' : E \rightarrow F$, $\Gamma, x : B \vdash_{\mathcal{AF}2S} x : E$ and $\forall \boldsymbol{\xi}'F \subseteq C$ where $\boldsymbol{\xi}'$ is not free in Γ and B. By Corollary 12, we obtain $\forall \boldsymbol{\xi}''B \subseteq E$ where $\boldsymbol{\xi}''$ is not free in Γ and B. We have $B \subseteq B$, then $B \subseteq \forall \boldsymbol{\xi}''B \subseteq E$, and $B \subseteq \forall \boldsymbol{\xi}'E$. Using the rules $(dist)$ and $(\rightarrow)$, we deduce $\forall \boldsymbol{\xi}'(E \rightarrow F) \subseteq \forall \boldsymbol{\xi}'E \rightarrow \forall \boldsymbol{\xi}'F \subseteq B \rightarrow C$ and $\forall \boldsymbol{\xi} \forall \boldsymbol{\xi}'(E \rightarrow F) \subseteq \forall \boldsymbol{\xi}(B \rightarrow C)$. Finally , we have $\Gamma \vdash_{\mathcal{AF}2S} t' : E \rightarrow F$, then $\Gamma \vdash_{\mathcal{AF}2S} t' : \forall \boldsymbol{\xi} \forall \boldsymbol{\xi}'(E \rightarrow F)$, and, by the rule (tr), we obtain $\Gamma \vdash_{\mathcal{AF}2S} t' : A$.

∎

We will see that the system $\mathcal{AF}2S$ is exactly $\mathcal{AF}2$ in which one adds the conservation of the type by η-reduction as a typing rule.

DEFINITION 21. — *The typing system $\mathcal{AF}2\eta$ is the system $\mathcal{AF}2$, in which we add the following typing rule:*

$$\text{If } \Gamma \vdash_{\mathcal{AF}2\eta} t : A \text{ and } t \rightarrow_\eta t', \text{ then } \Gamma \vdash_{\mathcal{AF}2\eta} t' : A \qquad (\eta)$$

The typing rule $(\subseteq)$ is derivable in the system $\mathcal{AF}2\eta$.

THEOREM 22. — *If $\Gamma \vdash_{\mathcal{AF}2\eta} t : A$ and $A \subseteq B$, then $\Gamma \vdash_{\mathcal{AF}2\eta} t : B$.*

PROOF. — By induction on the proof of $A \subseteq B$. We consider the last rule used. The only difficult case is $(dist)$. We have $A = \forall \xi(C \rightarrow D)$ and $B = \forall \xi C \rightarrow \forall \xi D$. If $\Gamma, x : \forall \boldsymbol{\xi} C \vdash_{\mathcal{AF}2} t : \forall \xi(C \rightarrow D)$, then $\Gamma, x : \forall \boldsymbol{\xi} C \vdash_{\mathcal{AF}2} t : C \rightarrow D$ and $\Gamma, x : \forall \boldsymbol{\xi} C \vdash_{\mathcal{AF}2} (t)x : D$. Since $\boldsymbol{\xi}$ is not free in Γ, we obtain $\Gamma, x : \forall \boldsymbol{\xi} C \vdash_{\mathcal{AF}2} (t)x : \forall \boldsymbol{\xi} D$ and $\Gamma \vdash_{\mathcal{AF}2} \lambda x(t)x : \forall \boldsymbol{\xi} C \rightarrow \forall \boldsymbol{\xi} D$. Since $\lambda x(t)x \rightarrow_\eta t$, we deduce $\Gamma \vdash_{\mathcal{AF}2} t : B$. ∎

We can then deduce the following result:

THEOREM 23. — $\Gamma \vdash_{\mathcal{AF}2S} t : A$ *iff* $\Gamma \vdash_{\mathcal{AF}2\eta} t : A$.

PROOF. — By Theorems 20 et 22. ∎

We can also state the following proposition:

PROPOSITION 24. — *If $\Gamma \vdash_{\mathcal{AF}2\eta} t : A$, then there is a λ-term u such that $u \to_\eta t$ and $\Gamma \vdash_{\mathcal{AF}2} u : A$.*

PROOF. — By induction on the typing $\Gamma \vdash_{\mathcal{AF}2\eta} t : A$. ∎

3.4. *The strong normalization*

NOTATION 25. — *We write $u \to_{\beta^+} v$ if v is obtained from u by at least one step of β-reduction denoted β_0.*

LEMMA 26. — *Let u, t, v be λ-terms such that $u \to_\eta t$ and $t \to_{\beta_0} v$. Then there is a λ-term w such that $u \to_{\beta^+} w$ and $w \to_\eta v$.*

PROOF. — See [BAR 84]. ∎

LEMMA 27. — *Let u, t be λ-terms. If u is strongly normalizable, and $u \to_\eta t$, then t is also strongly normalizable.*

PROOF. — If t is not strongly normalizable, then there is an infinite sequense of β_0-reductions starting with t. Since $u \to_\eta t$, then, by Lemma 26, we construct an infinite sequence of β_0-reductions starting with u. ∎

THEOREM 28. — *If $\Gamma \vdash_{\mathcal{AF}2S} t : A$, then t is strongly normalizable.*

PROOF. — By Proposition 24, Theorem 2 and Lemma 27. ∎

4. The complete types

DEFINITION 29. — *We say that a closed type A is **complete** in $\mathcal{AF}2S$ iff $|A|_f = \{t \in \Lambda \,/\, t \to_\beta t' \text{ and } \vdash_{\mathcal{AF}2S} t' : A\}$.*

We will give a class of complete types. We start by extending the correction theorem to system $\mathcal{AF}2S$.

LEMMA 30. — *Let M be a Λ_f-model of E and I an interpretation of E. If $A \subseteq B$, then $|A|_{M,I} \subseteq |B|_{M,I}$.*

PROOF. — By induction on the derivation $A \subseteq B$. ∎

THEOREM 31 (THE GENERALIZED CORRECTION). — *Let M be a Λ_f-model of E and I an interpretation. If $\Gamma = x_1 : B_1, \ldots, x_n : B_n \vdash_{\mathcal{AF}2S} t' : A$, $t \simeq_\beta t'$, and $u_i \in |B_i|_{M,I}$ $(1 \le i \le n)$, then $t[u_1/x_1, \ldots, u_n/x_n]_{M,I} \in |A|_{M,I}$.*

PROOF. — We may assume that t' is normal. The proof is done by induction on the typing of t'. We look at the last rule used.

(S1) Then $t' = x_i$ $(1 \le i \le n)$ and $\forall \boldsymbol{\xi} B_i \subseteq A$ where $\boldsymbol{\xi}$ is not free in B_i $(1 \le i \le n)$. Since $t \simeq_\beta x_i$, then $t \succ_f x_i$ and $t[u_1/x_1, \ldots, u_n/x_n] \succ_f u_i$. But $u_i \in |B_i|_{M,I}$, then $t[u_1/x_1, \ldots, u_n/x_n] \in |B_i|_{M,I}$. Since $\boldsymbol{\xi}$ is not free in

B_i, we deduce $t[u_1/x_1, \ldots, u_n/x_n] \in \forall \boldsymbol{\xi} B_i$ and, by Lemma 30, we obtain $t[u_1/x_1, \ldots, u_n/x_n] \in |A|_{M,I}$.

(S2) Then $t' = \lambda x u'$, $\Gamma, x : B \vdash_{\mathcal{AF}2S} u' : C$ and $\forall \boldsymbol{\xi}(B \to C) \subseteq A$ where $\boldsymbol{\xi}$ is not free in B_i $(1 \leq i \leq n)$. Since $t \simeq_\beta \lambda x u'$, then $t \succ_f \lambda x u$ where $u \simeq_\beta u'$ and $t[u_1/x_1, \ldots, u_n/x_n] \succ_f \lambda x u[u_1/x_1, \ldots, u_n/x_n]$. Therefore, by induction hypothesis,
$u[u_1/x_1, \ldots, u_n/x_n, v/x] \in |C|_{M,I}$ for all $v \in |B|_{M,I}$. We have
$(\lambda x u[u_1/x_1, \ldots, u_n/x_n])v \succ_f u[u_1/x_1, \ldots, u_n/x_n, v/x]$, then
$\lambda x u[u_1/x_1, \ldots, u_n/x_n] \in |B \to C|_{M,I}$, and $t[u_1/x_1, \ldots, u_n/x_n] \in |\forall \boldsymbol{\xi}(B \to C)|_{M,I}$. By Lemma 30, we deduce $t[u_1/x_1, \ldots, u_n/x_n] \in |A|_{M,I}$.

(S3) Then $t' = (u')v'$, $\Gamma \vdash_{\mathcal{AF}2S} u' : B \to C$, $\Gamma \vdash_{\mathcal{AF}2S} v' : B$ and $\forall \boldsymbol{\xi} C \subseteq A$ where $\boldsymbol{\xi}$ is not free in B_i $(1 \leq i \leq n)$. But $t \simeq_\beta (x_r)v_1' \ldots v_m'$, then $t \succ_f (x_r)v_1 \ldots v_m$ where $v_i \simeq_\beta v_i'$ $(1 \leq i \leq m)$, and, by induction hypothesis,
$(u_r)v_1[u_1/x_1, \ldots, u_n/x_n] \ldots v_{m-1}[u_1/x_1, \ldots, u_n/x_n] \in |B \to C|_{M,I}$ and
$v_m[u_1/x_1, \ldots, u_n/x_n] \in |B|_{M,I}$. Therefore
$(u_r)v_1[u_1/x_1, \ldots, u_n/x_n] \ldots v_m[u_1/x_1, \ldots, u_n/x_n] \in |C|_{M,I}$ and
$t[u_1/x_1, \ldots, u_n/x_n] \in |A|_{M,I}$.

∎

DEFINITION 32. — *We define the types with **positive quantifier** (resp. **negative quantifier**) denoted $\forall_2^+$ (resp. $\forall_2^-$) by:*

– *An atomic formula is $\forall_2^+$ and $\forall_2^-$;*

– *If A is $\forall_2^+$ (resp. $\forall_2^-$) and B is $\forall_2^-$ (resp. $\forall_2^+$), then $B \to A$ is $\forall_2^+$ (resp. $\forall_2^-$);*

– *If A is $\forall_2^+$ and x (resp. X) is a first order (resp. n-ary second-order) variable, then $\forall x A$ (resp. $\forall X A$) is $\forall_2^+$;*

– *If A is $\forall_2^-$ and x is a first-order variable, then $\forall x A$ is $\forall_2^-$.*

We will prove that the $\forall_2^+$ types are complete in $\mathcal{AF}2S$.

DEFINITIONS 33. — *Let $\Omega = \{x_i \, ; \, i \in \boldsymbol{N}\}$ be an enumeration of an infinite set of variables of λ-calculus and $\{A_i \, ; \, i \in \boldsymbol{N}\}$ be an enumeration of $\forall_2^-$ types of $\mathcal{AF}2S$, where every $\forall_2^-$ type occurs an infinite number of times. We define the set $\Gamma^- = \{x_i : A_i \, ; \, i \in \boldsymbol{N}\}$. Let u be a λ-term such that $Fv(u) \subseteq \Omega$, we define the contexte Γ_u^- as the restriction of Γ^- on the set $Fv(u)$. The expression $\Gamma^- \vdash_{\mathcal{AF}2S} u : B$ means that $\Gamma_u^- \vdash_{\mathcal{AF}2S} u : B$. We put $\Gamma^- \vdash_{\mathcal{AF}2S}^\beta u : B$ iff there is a λ-term u' such that $u \to_\beta u'$ and $\Gamma^- \vdash_{\mathcal{AF}2S} u' : B$.*

Let L be a second-order language and E an equation system of L. We define on the set of terms of L an equivalence relation denoted $\approx_E$ by: $a \approx_E b$ iff we can obtain it by the following rules:

(i) if $a = b$ is a particular case of an equation of E, then $a \approx_E b$;

(ii) for every terms a, b, c of L, we have: $a \approx_E a$; and if $a \approx_E b$ and $b \approx_E c$, then $a \approx_E c$;

(iii) If f is n-ary function symbol of L, and if $a_i \approx_E b_i$ ($1 \leq i \leq n$), then $f(a_1, \ldots, a_n) \approx_E f(b_1, \ldots, b_n)$.

The following lemma allows to generalize the rule (8).

LEMMA 34. — *If $\Gamma \vdash_{\mathcal{AF}2} u : B[a/x]$ and $a \approx_E b$, then $\Gamma \vdash_{\mathcal{AF}2} u : B[b/x]$.*

PROOF. — By induction in the definition of $\approx_E$.   ∎

DEFINITION 35. — *We consider M_0 the set of all closed terms of L. We define a particular Λ_f-model $\mathcal{M}$ by:*

– The domain $|\mathcal{M}| = M_0/\approx_E$ (the set of equivalence classes modulo $\approx_E$);

– The adequate set $\mathcal{R}_f$;

– To every n-ary symbol function f, we associate a function $f_\mathcal{M} : |\mathcal{M}|^n \to |\mathcal{M}|$ defined by $f_\mathcal{M}(\overline{a_1}, \ldots, \overline{a_n}) = \overline{f(a_1, \ldots, a_n)}$;

– To every n-ary predicate symbol P, we associate a function $P_\mathcal{M} : |\mathcal{M}|^n \to \mathcal{R}_f$ defined by $P_\mathcal{M}(\overline{a_1}, \ldots, \overline{a_n}) = \{\tau \in \Lambda \,;\, \Gamma^- \vdash^\beta_{\mathcal{AF}2S} \tau : P(a_1, \ldots, a_n)\}$.

It is easy to see that $f_\mathcal{M}$ and $P_\mathcal{M}$ are well defined.

DEFINITION 36. — *We define a particular interpretation $\mathcal{I}$ on the variables by: $\mathcal{I}(x) = \overline{x}$ and $\mathcal{I}(X) = \Phi$, where $\Phi : |\mathcal{M}|^n \to \mathcal{R}_f$ defined by $\Phi(\overline{a_1}, \ldots, \overline{a_n}) = \{\tau \in \Lambda \,;\, \Gamma^- \vdash^\beta_{\mathcal{AF}2S} \tau : X(a_1, \ldots, a_n)\}$.*

We have the following lemma.

LEMMA 37. — *Let S be a formula of L and τ a λ-term.*

(i) If S is $\forall_2^+$ and $\tau \in | S |_{\mathcal{M}, \mathcal{I}}$, then $\Gamma^- \vdash^\beta_{\mathcal{AF}2S} \tau : S$.

(ii) If S is $\forall_2^-$ and $\Gamma^- \vdash^\beta_{\mathcal{AF}2S} \tau : S$, then $\tau \in | S |_{\mathcal{M}, \mathcal{I}}$.

PROOF. — By simultanous induction on the $\forall_2^+$ and $\forall_2^-$ types.

SUBPROOF (OF *(i)*). —

1) S is atomic: The result is trivial.

2) $S = \forall X B$ where B is $\forall_2^+$: Let Y be an n-ary predicate variable which does not appear in Γ_τ^- and B. If $\tau \in |\forall X B|_{\mathcal{M},\mathcal{I}}$, then $\tau \in |B[X]|_{\mathcal{M},\mathcal{I}[X \leftarrow |Y|_{\mathcal{M},\mathcal{I}}]} = |B[Y/X]|_{\mathcal{M},\mathcal{I}}$. By induction hypothesis, we have $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : B[Y]$, and there is a λ-term τ' such that $\tau \to_\beta \tau'$ and $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : B[Y]$. Since $Fv(\tau') \subseteq Fv(\tau)$, we deduce $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : \forall Y B[Y] = \forall X B$, and $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : S$.

3) $S = B \to C$ where B is $\forall_2^-$ and C is $\forall_2^+$: let $\tau \in | B \to C |_{\mathcal{M},\mathcal{I}}$. We put an i such that $B = A_i$ and x_i is not free in τ. We have $x_i : B \vdash_{\mathcal{AF}_2} x_i : B$, then, by (ii), $x_i \in | B |_{\mathcal{M},\mathcal{I}}$, therefore $(\tau)x_i \in | C |_{\mathcal{M},\mathcal{I}}$, and, by induction hypothesis, $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta (\tau)x_i : C$. Thus $(\tau)x_i \to_\beta \tau'$ and $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : C$. We deduce that $(\tau)x_i$ is normalizable, then τ is also normalizable. Since $(\tau)x_i \to_\beta \tau'$, we obtain $\lambda x_i(\tau)x_i \to_\beta \lambda x_i \tau'$.

- If the normal form of τ is $\lambda x u$, then $\lambda x_i(\tau)x_i \to_\beta \lambda x_i(\lambda x u)x_i \to_\beta \lambda x u$ and $\lambda x_i \tau' \to_\beta \lambda x u$. But $\Gamma^- \vdash_{\mathcal{AF}_2S} \lambda x_i \tau' : S$ and $Fv(\lambda x u) \subseteq Fv(\lambda x_i \tau')$, then, by Theorem 19, we obtain $\Gamma^- \vdash_{\mathcal{AF}_2S} \lambda x u : S$, and $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : S$.

- If not, let v the normal form of τ. We have $\lambda x_i(\tau)x_i \to_\beta \lambda x_i(v)x_i$ and $\lambda x_i \tau' \to_\beta \lambda x_i(v)x_i$. Since $Fv(\lambda x_i(v)x_i) \subseteq Fv(\lambda x_i \tau')$, we deduce that $\Gamma^- \vdash_{\mathcal{AF}_2S} \lambda x_i(v)x_i : S$. Then, by Theorem 20 and $Fv(\lambda x_i(v)x_i) = Fv(v)$, we obtain $\Gamma^- \vdash_{\mathcal{AF}_2S} v : S$. Therefore $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : S$. □

SUBPROOF (OF *(ii)*). —

1) S is atomic: The result is trivial.

2) $S = B \to C$ where B is $\forall_2^+$ and C is $\forall_2^-$: If $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : B \to C$, then, there is a λ-term τ' such that $\tau \to_\beta \tau'$ and $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : B \to C$. If $u \in |B|_{\mathcal{M},\mathcal{I}}$, then, by (i), $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta u : B$, and there is a λ-term u' such that $u \to_\beta u'$ and $\Gamma_{u'}^- \vdash_{\mathcal{AF}_2S} u' : B$. Therefore $\Gamma_{(\tau')u'}^- \vdash_{\mathcal{AF}_2S} (\tau')u' : C$, and, since $(\tau)u \to_\beta (\tau')u'$, we obtain $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta (\tau)u : C$. By induction hypothesis, we deduce $(\tau)u \in |C|_{\mathcal{M},\mathcal{I}}$.

3) $S = \forall x B$ where B is $\forall_2^-$: Let $a \in |\mathcal{M}|$; we have $a = \overline{b}$ where b is a term of L. If $\Gamma^- \vdash_{\mathcal{AF}_2S}^\beta \tau : \forall x B$, then there is a λ-term τ' such that $\tau \to_\beta \tau'$ and $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : \forall x B$, therefore $\Gamma_{\tau'}^- \vdash_{\mathcal{AF}_2S} \tau' : B[b/x]$. But $B[b/x]$ is $\forall_2^-$, then, by induction hypothesis, $\tau' \in |B[b/x]|_{\mathcal{M},\mathcal{I}} = |B|_{\mathcal{M},\mathcal{I}[x \leftarrow \overline{b}]} = |B|_{\mathcal{M},\mathcal{I}[x \leftarrow a]}$. Thus $\tau \in |B|_{\mathcal{M},\mathcal{I}[x \leftarrow a]}$ for every $a \in |\mathcal{M}|$. □ ■

THEOREM 38. — *The closed $\forall_2^+$ types are complete.*

PROOF. — Let A be a closed $\forall_2^+$ type. We will prove that: $t \in |A|_f$ iff there is a λ-term t' such that $t \to_\beta t'$ and $\vdash_{\mathcal{AF}_2S} t' : A$.

– That the condition is sufficient is a simple consequence of Theorem 31.

– The condition is necessary: Indeed, let t be a λ-term such that $t \in |A|_f$, then $t \in |A|_{\mathcal{M}}$. We may assume that $\Gamma_t^- = \emptyset$. By (i) of Lemma 37, we obtain $\Gamma^- \vdash^{\beta}_{\mathcal{AF}_2S} t : A$, then there is a λ-term t' such that $t \rightarrow_\beta t'$ and $\Gamma_{t'}^- \vdash_{\mathcal{AF}_2S} t' : A$. Since $Fv(t') \subseteq Fv(t)$, we deduce $\Gamma_{t'}^- = \emptyset$.

■

COROLLARY 39. — *Let A be a closed $\forall_2^+$ type and t a λ-term. If $t \in |A|_f$, then t is normalizable and β-equivalent to a closed λ-term.*

PROOF. — By Theorem 38.   ■

5. References

[BAR 84] BARENDREGT H., *The Lambda Calculus : Its Syntax and Semantics*, North Holland, 1984.

[BÖH 85] BÖHM C., "Automatic synthesis of typed Λ-programs on term algebra", *Theoretical Computer Science*, vol. 39, 1985, p. 135–154.

[FAR 98] FARKH S., NOUR K., "Résultats de complétude pour des classes de types du système AF2", *Informatique Théorique et Application*, vol. 31, num. 6, 1998, p. 513–537.

[KRI 90] KRIVINE J.-L., "Opérateurs de mise en mémoire et traduction de Gödel", *Archive for Mathematical Logic*, vol. 30, 1990, p. 241–267.

[KRI 94] KRIVINE J.-L., "Classical Logic, Storage Operators and Second Order Lambda-Calculs", *Annals of Pure and Applied Logic*, vol. 68, 1994, p. 53–78.

[LAB 86] LABIB-SAMI R., "Typer avec (ou sans) types auxilières", *Manuscrit*, , 1986.

[MIT 88] MITCHELL J.-C., "Polymorphic type inference and containment", *Inf. Comput*, vol. 76, num. 2–3, 1988, p. 211–249.

[NOU 96] NOUR K., "Storage operators and $\forall$-positive types of system TTR", *Mathematical Logic Quarterly*, vol. 42, 1996, p. 349–368.

[RAF 98] RAFFALLI C., "A Semantic Storage Operator Theorem for All Types", *Annals of Pure and Applied Logic*, vol. 91, 1998, p. 17–31.