

# On the Membership Problem for Non-Linear Abstract Categorical Grammars

Sylvain Salvati

► **To cite this version:**

Sylvain Salvati. On the Membership Problem for Non-Linear Abstract Categorical Grammars. Journal of Logic, Language and Information, Springer Verlag, 2010, 19 (2), pp.163-183. <<http://www.springerlink.com/content/f3g38386p568n7j4/>>. <10.1007/s10849-009-9110-0>. <inria-00446785>

**HAL Id: inria-00446785**

**<https://hal.inria.fr/inria-00446785>**

Submitted on 13 Jan 2010

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

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

# On the membership problem for non-linear Abstract Categorial Grammars

Sylvain Salvati

**Abstract** In this paper we show that the membership problem for second order non-linear Abstract Categorial Grammars is decidable. A consequence of that result is that Montague-like semantics yield to a decidable text generation problem. Furthermore the proof we propose is based on a new tool, Higher Order Intersection Signatures, which grasps statically dynamic properties of  $\lambda$ -terms and presents an interest in its own.

## 1 Introduction

Abstract Categorial Grammars (ACGs) have been introduced in [dG01] as a simple tool to model the interface between syntax and semantics. In those grammars, there is a distinction between the abstract level in which deep structures are represented as proofs in linear logic and the object level which accounts for the surface structures. The *lexicon* is formalized as a homomorphism which performs the translation between those two levels. This lexicon is bound to be described by means of higher order linear  $\lambda$ -terms (each bound variable occurs exactly once). In this paper we are concerned with the consequences on the membership problem when the linearity constraint on the lexicons is dropped. Extending ACGs in that direction is natural since linearity has many shortcomings especially when one is concerned with semantics. But also, many structures can be encoded in the simply typed  $\lambda$ -calculus, such as tuples, enumerated types, conditionals, lists or even relational databases [Hil94]. All those structures can serve both in the modelisation of the syntax and of the semantics of natural languages. We show here that in the particular case of second order ACGs, the membership problem remains decidable while non-linear lexicons are used. This result is of particular importance since it means that, in general, generating texts from meaning representations is decidable when the semantics is Montague-like [Mon74].

---

This work was done while the author was at the National Institute of Informatics as a post-doctoral fellow of the Japan Society for the Promotion of Science, and was supported by the Grant-in-Aid for Scientific Research (18-06739).

---

Sylvain Salvati  
INRIA BORDEAUX SUD-OUEST, LABRI, UNIVERSITÉ DE BORDEAUX, Unité Mixte de Recherche  
CNRS (UMR 5800), 351, cours de la Libération, F-33405 Talence cedex France  
Tel.: +33(0)5 4000 35 23, Fax: +33(0)5 4000 66 69  
E-mail: sylvain.salvati@labri.fr

Modern approaches of Montague semantics are mainly concerned with describing the interface between syntax and semantics by using an intermediate structure, usually called the deep structure of the sentence, and interpreting this structure with a homomorphism so as to obtain a representation of the meaning of that sentence. Usually this meaning is represented as a formula in a given logic. For example we can use the following operators so as to define the deep structure of a toy language:

1. SEEK :  $np \rightarrow np \rightarrow s$ ,
2. AND :  $np \rightarrow np \rightarrow np$ ,
3. JOHN :  $np$ ,
4. MARY :  $np$ ,
5. A :  $n \rightarrow np$ ,
6. UNICORN :  $n$ .

The terms built with these operators can be interpreted with the homomorphism  $\mathcal{L}_{syn}$  such that:

1.  $\mathcal{L}_{syn}(\text{SEEK}) = \lambda xy.x \text{ cherchent } y$ ,
2.  $\mathcal{L}_{syn}(\text{AND}) = \lambda xy.x \text{ et } y$ ,
3.  $\mathcal{L}_{syn}(\text{JOHN}) = \text{Jean}$ ,
4.  $\mathcal{L}_{syn}(\text{MARY}) = \text{Marie}$ ,
5.  $\mathcal{L}_{syn}(\text{A}) = \lambda x.\text{une } x$ ,
6.  $\mathcal{L}_{syn}(\text{UNICORN}) = \text{licorne}$ .

The term  $\text{sentence} = \text{SEEK}(\text{AND JOHN MARY})(\text{A UNICORN})$  is interpreted as the French sentence *Jean et Marie cherchent une licorne* with the homomorphism  $\mathcal{L}_{syn}$ . We may then try to give a semantics to this language by defining a homomorphism  $\mathcal{L}_{sem}$  such that:

1.  $\mathcal{L}_{sem}(\text{SEEK}) = \lambda S O.O(\lambda x.S(\lambda y.\text{seek } y x))$ ,
2.  $\mathcal{L}_{sem}(\text{AND}) = \lambda N_1 N_2 P. \wedge (N_1 P)(N_2 P)$ ,
3.  $\mathcal{L}_{sem}(\text{JOHN}) = \lambda P.P \mathbf{j}$ ,
4.  $\mathcal{L}_{sem}(\text{MARY}) = \lambda P.P \mathbf{m}$ ,
5.  $\mathcal{L}_{sem}(\text{A}) = \lambda P Q.\exists(\lambda x. \wedge (P x)(Q x))$ ,
6.  $\mathcal{L}_{sem}(\text{UNICORN}) = \lambda x.\text{unicorn } x$ .

Then  $\mathcal{L}_{sem}$  maps  $\text{sentence}$  to the term

$$\exists(\lambda x. \wedge (\text{unicorn } x)(\wedge(\text{seek } \mathbf{j} x)(\text{seek } \mathbf{m} x)))$$

Given such a term, one may wonder whether the grammar we use contains a sentence whose meaning representation is precisely that term. We here show that answering such a question is decidable in general and our results can be used so as to generate such a sentence. Note that we do not answer the problem of generation in general which should take into account some problem of logical equivalence, we rather solve the problem of *exact generation* as proposed in [Kan07]. Indeed given the semantic term

$$\exists(\lambda x. \wedge (\text{unicorn } x)(\wedge(\text{seek } \mathbf{j} x)(\text{seek } \mathbf{j} x)))$$

we will generate the sentence *Jean et Jean cherchent une licorne*. Contrary to [Kan07], and besides well-typedness, we do not have any restrictions on the  $\lambda$ -terms that can be used to define the correspondence between syntax and semantics. One may in particular

use duplication and deletion as much as one wishes to describe the translation of syntactic structures into meaning representations.

The approach that leads to this result is similar to one generally used in formal language theory that consists in showing the closure of the class of languages under investigation by intersection with regular sets. ACGs do not define strings or trees languages but languages of  $\lambda$ -terms. Thus the main question we face is related to the definition of a suitable notion of regularity for the simply typed  $\lambda$ -calculus. The tools involved to cope with that problem are based on intersection types [DCGdL98]. Even though we won't discuss the issue of defining a relevant notion of regularity for the simply typed  $\lambda$ -calculus here, all that we do is closely related to such a notion, and the proofs we get are quite similar to the usual syntactic proofs of closure of Context Free Languages under intersection with regular sets. They also resemble to the proofs of [Kan06] where it is showed that Abstract Categorical Languages (languages defined by ACGs) are closed under the inverse of a relabeling. Actually, restricted to strings or trees languages, second-order non-linear ACGs coincide with level- $n$  IO grammars introduced and studied in [Dam82]. Our result somehow extends Damm's result which shows that level- $n$  IO grammars are closed under intersection with regular sets.

The outline of the paper is as follows, in section 2 we introduce the necessary technical notions. In section 3, we define and show the principal properties of the main tool on which our results are based, *Higher Order Intersection Signatures*. We give and prove the main theorem, *the singleton set theorem*, concerning higher-order intersection signatures in section 4. This theorem shows that singleton sets, *i.e.* sets like  $\{N|N =_{\beta\eta} M\}$ , can be characterized by typing properties expressed on Higher Order Intersection Signatures. Section 5 shows that the class of languages of  $\lambda$ -terms defined by second order non-linear ACGs is closed under typing judgements over Higher Order Intersection Signature. Finally section 6 gives some conclusions and future directions of research.

## 2 Simply typed $\lambda$ -calculus and non-linear ACGs

A *higher order signature* (HOS)  $\Sigma$  is a triple  $(\mathcal{A}, \mathcal{C}, \tau)$  such that  $\mathcal{A}$  is a finite set of *atomic types*,  $\mathcal{C}$  is a finite set of *constants*, and  $\tau$  is a function from  $\mathcal{C}$  to  $\mathcal{T}_{\mathcal{A}}$ , the set of *simple types* (or *types*).  $\mathcal{T}_{\mathcal{A}}$  is the smallest set containing  $\mathcal{A}$  and verifying the property that whenever it contains  $\alpha_1$  and  $\alpha_2$  then it contains  $(\alpha_1 \rightarrow \alpha_2)$ ; we may also write  $\mathcal{T}_{\Sigma}$  instead of  $\mathcal{T}_{\mathcal{A}}$ . We will use Greek lowercase letters,  $\alpha, \beta, \gamma, \dots$  to denote types, when necessary we will also use indices and superscripts. We will write  $\alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \beta$  for the type  $(\alpha_1 \rightarrow (\dots \rightarrow (\alpha_n \rightarrow \beta) \dots))$ . Moreover, higher order signatures will be denoted by  $\Sigma$  (sometimes with subscripts),  $\Sigma_1, \Sigma_2, \dots$  and, unless stated otherwise, we will assume that  $\Sigma$  is the triple  $(\mathcal{A}, \mathcal{C}, \tau)$ ,  $\Sigma_{sub}$  is the triple  $(\mathcal{A}_{sub}, \mathcal{C}_{sub}, \tau_{sub})$  with *sub* being either a subscript or an index. In general, we will use the roman lowercase letters  $a, b, c, d, e$  to denote constants. The order  $ord(\alpha)$  of a type  $\alpha$  is defined to be 1 if that type is atomic and  $max(ord(\alpha_1) + 1, ord(\alpha_2))$  if  $\alpha = \alpha_1 \rightarrow \alpha_2$ . The notion of order is extended to higher order signatures, and  $ord(\Sigma) = max\{c \in \mathcal{C} | ord(\tau(c))\}$ .

We then assume that we are given an infinite and countable set  $\mathcal{V}$  of  $\lambda$ -variables; in general,  $x, y$  and  $z$  (possibly with indices) will range over  $\lambda$ -variables. The  $\lambda$ -calculus we will use is typed *à la Church* so that terms have a unique type. In particular  $\lambda$ -variables (or simply variables when it is unambiguous) explicitly convey their types. Given a finite set of atomic types  $\mathcal{A}$ , the association of  $\lambda$ -variables with types is simply

obtained by considering the set  $\mathcal{V} \times \mathcal{T}_{\mathcal{A}}$ . Given  $\alpha \in \mathcal{T}_{\mathcal{A}}$ , we write  $\mathcal{V}^\alpha$  for the set  $\mathcal{V} \times \{\alpha\}$ , the set of  $\lambda$ -variables of type  $\alpha$ ; and the set  $\bigcup_{\alpha \in \mathcal{T}_{\mathcal{A}}} \mathcal{V}^\alpha$  will be denoted by  $\mathcal{V}_{\mathcal{A}}$  or  $\mathcal{V}_{\Sigma}$ . In general the elements of  $\mathcal{V}^\alpha$  will be written  $x^\alpha, y^\alpha$  and  $z^\alpha$  instead of  $(x, \alpha), (y, \alpha)$  or  $(z, \alpha)$ , we may also sometimes omit the type annotation when it is irrelevant.

The set,  $\Lambda_{\Sigma}$ , of  $\lambda$ -terms built on  $\Sigma$ , is the union of the sets of the family  $(\Lambda_{\Sigma}^{\alpha})_{\alpha \in \mathcal{T}_{\mathcal{A}}}$  which is defined as the smallest family verifying:

1. if  $x \in \mathcal{V}$  and  $\alpha \in \mathcal{T}_{\mathcal{A}}$ , then  $x^\alpha \in \Lambda_{\Sigma}^{\alpha}$ ,
2. if  $c \in \mathcal{C}$  then  $c \in \Lambda_{\Sigma}^{\tau(c)}$ ,
3. if  $M_1 \in \Lambda_{\Sigma}^{\alpha \rightarrow \beta}$ ,  $M_2 \in \Lambda_{\Sigma}^{\alpha}$ , then  $(M_1 M_2) \in \Lambda_{\Sigma}^{\beta}$ ,
4. if  $M \in \Lambda_{\Sigma}^{\beta}$ , then  $\lambda x^\alpha. M \in \Lambda_{\Sigma}^{\alpha \rightarrow \beta}$ .

The notation  $FV(M)$  stands for the set of free variables of  $M$  and is defined as usual. Note that the variables  $x^\alpha$  and  $x^\beta$  are considered to be distinct when  $\alpha \neq \beta$ . This in particular has the consequence that  $FV(\lambda x^\alpha. x^{\alpha \rightarrow \alpha}) = \{x^{\alpha \rightarrow \alpha}\}$ . We will use the letters  $M, N, P, Q$ , possibly with some indices, to denote  $\lambda$ -terms. Given  $M_1 \in \Lambda_{\Sigma}^{\alpha_1}, \dots, M_n \in \Lambda_{\Sigma}^{\alpha_n}$ , we define  $M[x_1^{\alpha_1} := M_1; \dots; x_n^{\alpha_n} := M_n]$  to be the result of the simultaneous capture-avoiding substitution of  $x_1^{\alpha_1}$  by  $M_1, \dots$  and of  $x_n^{\alpha_n}$  by  $M_n$ . Given  $\gamma$  in  $\{\beta; \eta; \beta\eta\}$ , we write  $M \rightarrow_{\gamma} M'$  to denote that  $M$   $\gamma$ -contracts to  $M'$ ,  $M \xrightarrow{*}_{\gamma} M'$  denotes the fact that  $M$  is  $\gamma$ -reducible to  $M'$  and  $M =_{\gamma} M'$  stands for  $M$  is  $\gamma$ -convertible to  $M'$ ; we assume that the reader is familiar with these notions and that of normal form; we otherwise refer him/her to [Bar84]. In general we will write  $M_1 M_2 \dots M_n$  instead of  $(\dots (M_1 M_2) \dots M_n)$  and  $\lambda x_1^{\alpha_1} \dots x_n^{\alpha_n}. M$  instead of  $\lambda x_1^{\alpha_1} \dots \lambda x_n^{\alpha_n}. M$ .

A context is a term with a hole, generally written  $C[\ ]$ , the term  $C[N]$  is obtained by grafting the term  $N$  in the hole. Note that *grafting* a term in a context may bind some variable; for example if  $C[\ ] = \lambda x. [\ ]$  and  $N = x$  then  $C[x] = \lambda x. x$ . Given  $M$  in  $\Lambda_{\Sigma}$ ,  $M$  is said in  $\eta$ -long form, or in long form, if for every  $C[\ ]$  and  $N$  such that  $C[N] = M$  and  $N \in \Lambda_{\Sigma}^{\alpha \rightarrow \beta}$ , one of the following properties holds:

1.  $N$  is of the form  $\lambda x^\alpha. N'$ ,
2.  $C[\ ]$  is of the form  $C'[\ ]N'$ .

When a term is in long form, the structure of the types of its sub-terms is made explicit in the syntax of the term itself. If it has a complex type then this sub-term is either a  $\lambda$ -abstraction or it is applied to some argument. The set of terms in long form are closed under  $\beta$ -reduction and also when  $M$  is in long form and  $M' \rightarrow_{\eta} M$  we have that  $M' \rightarrow_{\beta} M$ . For example,  $\lambda y_1. (\lambda y_2. x y_2) y_1 \rightarrow_{\eta} \lambda y_1. x y_1$  and  $\lambda y_1. (\lambda y_2. x y_2) y_1 \rightarrow_{\beta} \lambda y_1. x y_1$ . Working with terms in long form allow us to compare terms modulo  $\beta\eta$ -convertibility only using  $\beta$ -conversion. Indeed, when  $M_1 =_{\beta\eta} M_2$  then if  $M'_1$  and  $M'_2$  are long forms of  $M_1$  and  $M_2$  then  $M'_1 =_{\beta} M'_2$  and the converse also hold. Long forms have been introduced in [Hue76] where their main properties have also been proved. It is quite usual to work with the long normal forms of terms in order to prove properties about the simply typed  $\lambda$ -calculus (*e.g.* [GTL89]) since they are a canonical representation of the proofs denoted by the terms.

A *homomorphism between two higher order signatures*  $\Sigma_1$  and  $\Sigma_2$  is a pair  $(g, h)$  such that  $g$  maps  $\mathcal{T}_{\mathcal{A}_1}$  to  $\mathcal{T}_{\mathcal{A}_2}$ ,  $h$  maps  $\Lambda_{\Sigma_1}$  to  $\Lambda_{\Sigma_2}$  and the following holds:

1.  $g(\alpha \rightarrow \beta) = g(\alpha) \rightarrow g(\beta)$
2.  $h(x^\alpha) = x^{g(\alpha)}$
3. for  $c \in \mathcal{C}_1$ ,  $h(c)$  is a closed element (*i.e.*  $FV(h(c)) = \emptyset$ ) of  $\Lambda_{\Sigma_2}^{\tau_1(c)}$
4.  $h(M_1 M_2) = h(M_1) h(M_2)$

$$5. h(\lambda x^\alpha.M) = \lambda x^{g(\alpha)}.h(M).$$

It is easy to establish that whenever  $M \in \Lambda_{\Sigma_1}^\alpha$  then  $h(M) \in \Lambda_{\Sigma_2}^{g(\alpha)}$ . In general, if  $\mathcal{H} = (g, h)$  is a homomorphism, we will write  $\mathcal{H}(\alpha)$  for  $g(\alpha)$  and  $\mathcal{H}(M)$  for  $h(M)$ .

Non-linear<sup>1</sup> Abstract Categorical Grammars are 4-tuples  $\mathcal{G} = (\Sigma_1, \Sigma_2, \mathcal{L}, S)$  where:

1.  $\Sigma_1$  is a higher order signature, *the abstract vocabulary*,
2.  $\Sigma_2$  is a higher order signature, *the object vocabulary*,
3.  $\mathcal{L}$  is a homomorphism from  $\Sigma_1$  to  $\Sigma_2$ , *the lexicon* and,
4.  $S$  is an element of  $\mathcal{A}_1$ , *the distinguished type*.

Note that contrary to the usual definition we do not require that  $\Sigma_1$  or  $\Sigma_2$  are linear signatures or the lexicon to be linear. The *order of a non-linear ACG* is the order of its abstract vocabulary. A non-linear ACG  $\mathcal{G} = (\Sigma_1, \Sigma_2, \mathcal{L}, S)$  defines two languages:

- *The abstract language*:  $\mathcal{A}(\mathcal{G}) = \{M \in \Lambda_{\Sigma_1}^S \mid M \text{ is closed}\}$ ,
- *The object language*:  $\mathcal{O}(\mathcal{G}) = \{M \in \Lambda_{\Sigma_2}^{\mathcal{L}(S)} \mid \exists N \in \mathcal{A}(\mathcal{G}). \mathcal{L}(N) =_{\beta\eta} M\}$

### 3 Higher order intersection signatures

In this section we define *higher order intersection signatures* (HOIS). This new kind of signatures is used as a second layer of typing over simply typed  $\lambda$ -calculus. The reader must be careful not to get confused between the simple type a  $\lambda$ -term may have and the intersection types higher order intersection signatures may assign to it. HOIS adapt intersection types [DCGdL98], usually dedicated to the untyped  $\lambda$ -calculus, to the simply typed  $\lambda$ -calculus. Intuitively, the reason why we use intersection type comes from the fact that we want to obtain a notion of regularity for the simply typed  $\lambda$ -calculus<sup>2</sup>. Such a notion should define sets that are closed under  $\beta\eta$ -convertibility and expressing invariants of  $\lambda$ -terms under conversion is usually done with types. Thus regular sets of  $\lambda$ -terms should be defined in terms of typing. Furthermore a notion of regularity for the simply typed  $\lambda$ -calculus should extend the usual notions of regularities for strings and trees. If we take a tree automaton, the abstract machine that recognizes regular set of trees, its behavior is defined by rules of the form  $a(q_1, \dots, q_n) \rightarrow q$ , which could intuitively be rephrased with types by the fact that  $a$  has type  $q_1 \rightarrow \dots \rightarrow q_n \rightarrow q$ . But a tree automaton can have several such rules concerning  $a$ , it would then be as if  $a$  had several types. A natural way of giving several types to a constant in type theory is to use intersection types. We stop here concerning the motivations of using intersection types and we now define higher order intersection signatures.

Given a higher order signature  $\Sigma$ , a finite set  $I$  and a function  $\rho$  from  $I$  to  $\mathcal{A}$ , we define the set of intersection types  $\cap_\rho$  as the union of the member of the family  $(\cap_\rho^\alpha)_{\alpha \in \mathcal{T}_\mathcal{A}}$  which is the smallest family verifying:

1.  $\cap_\rho^\alpha = \rho^{-1}(\alpha)$  if  $\alpha \in \mathcal{A}$
2.  $\cap_\rho^{\alpha \rightarrow \beta} = \mathcal{P}(\cap_\rho^\alpha) \times \{\alpha\} \times \cap_\rho^\beta$

where  $\mathcal{P}(A)$  is the powerset of  $A$ .

<sup>1</sup> as opposed to the usual ACGs defined in [dG01].

<sup>2</sup> we will develop the reasons why we can actually define regular sets of  $\lambda$ -terms with HOIS in a future paper.

The traditional notation for the intersection types is  $(\alpha \cap \beta)$ . It is provable that the  $\cap$  connector enjoys properties like idempotence, associativity, commutativity and has a neutral element  $\omega$ .<sup>3</sup> Since, for technical reasons, we want to use types modulo these equivalence relations, we use a different notation for intersection types. The intersection of several types will be denoted by the set containing those types. Intuitively the intersection type  $(S, \alpha, p) \in \cap_{\rho}^{\alpha \rightarrow \beta}$  would be written in the usual syntax as  $p_1 \cap \dots \cap p_n \rightarrow p$  when  $S \neq \emptyset$  and  $S = \{p_1; \dots; p_n\}$  or  $\omega_{\alpha} \rightarrow p$ <sup>4</sup> when  $S = \emptyset$ . The fact that we use  $\mathcal{P}(\cap_{\rho}^{\alpha}) \times \{\alpha\} \times \cap_{\rho}^{\beta}$  instead of  $\mathcal{P}(\cap_{\rho}^{\alpha}) \times \cap_{\rho}^{\beta}$  in the definition  $\cap_{\rho}^{\alpha \rightarrow \beta}$  has the consequence that if  $p$  belongs to  $\cap_{\rho}$  then there is a unique  $\alpha$  such that  $p \in \cap_{\rho}^{\alpha}$ . Thus an intersection type  $p$  that belongs to  $\cap_{\rho}^{\alpha}$  is, as one may expect, only used to type the terms of  $\Lambda_{\Sigma}^{\alpha}$ . In examples we will feel free to use the more traditional notation that uses  $\cap$  and  $\omega_{\alpha}$ .

Note that given  $\alpha \in \mathcal{A}$ , we have  $|\cap_{\rho}^{\alpha}| \leq |I|$  given  $\alpha \rightarrow \beta \in \mathcal{T}_{\mathcal{A}}$  we have  $|\cap_{\rho}^{\alpha \rightarrow \beta}| = 2^{|\cap_{\rho}^{\alpha}|} \times |\cap_{\rho}^{\beta}|$ . Thus  $\cap_{\rho}^{\alpha}$  is finite for every  $\alpha \in \mathcal{T}_{\mathcal{A}}$ .

Then, a *higher order intersection signature*  $\Pi$  is a 4-tuple  $(\Sigma, I, \rho, f)$  where:

1.  $\Sigma$  is a higher order signature,
2.  $I$  is a finite set of atomic types.
3.  $\rho$  is a mapping from  $I$  to  $\mathcal{A}$ ,
4.  $f$  associates to any  $c \in \mathcal{C}$  a subset of  $\cap_{\rho}^{\tau(c)}$ .

Such a higher order intersection signature is also called an *intersection signature over*  $\Sigma$ . In general we shall write  $\cap_{\Pi}^{\alpha}$  instead of  $\cap_{\rho}^{\alpha}$  and  $\cap_{\Pi}$  instead of  $\cap_{\rho}$  for a HOIS  $\Pi = (\Sigma, I, \rho, f)$ . A  *$\Pi$ -type environment* is a function from  $\mathcal{V}_{\Sigma}$  to  $\mathcal{P}(\cap_{\Pi})$  verifying:

1.  $\Gamma(x^{\alpha})$  is a subset of  $\cap_{\Pi}^{\alpha}$ ,
2.  $\Gamma$  associates a non-empty subset of  $\cap_{\Pi}$  only to a finite number of variables.

A  $\Pi$ -type environment  $\Gamma$  may be represented as a finite sequence such as  $x_1^{\alpha_1} : S_1, \dots, x_n^{\alpha_n} : S_n$  where for all  $i$ ,  $\Gamma(x_i^{\alpha_i}) = S_i$  and if  $z^{\beta} \neq x_i^{\alpha_i}$  for all  $i \in [1; n]$  then  $\Gamma(z^{\beta}) = \emptyset$ . Given two  $\Pi$ -type environments  $\Gamma$  and  $\Delta$ , we write  $\Gamma, \Delta$  the  $\Pi$ -type environment such that  $\Gamma, \Delta(x^{\alpha}) = \Gamma(x^{\alpha}) \cup \Delta(x^{\alpha})$ . In particular, when we write  $\Gamma, x^{\alpha} : S$ , we implicitly consider that  $\Gamma(x^{\alpha}) = \emptyset$ .

Given an intersection signature  $\Pi = (\Sigma, I, \rho, f)$ , we define derivations which establish *judgements* of the form:  $\Gamma \vdash_{\Pi} M : p$  where  $\Gamma$  is a  $\Pi$ -type environment,  $M \in \Lambda_{\Sigma}$  and  $p \in \cap_{\Pi}$ . The term  $M$  is called the *subject* of the judgement  $\Gamma \vdash_{\Pi} M : p$ . The judgements are obtained with the following deduction rules:

$$\frac{p \in S}{\Gamma, x^{\alpha} : S \vdash_{\Pi} x^{\alpha} : p} \text{AXIOM} \quad \frac{p \in f(c)}{\Gamma \vdash_{\Pi} c : p} \text{CONST} \quad \frac{\Gamma, x^{\alpha} : S \vdash_{\Pi} M : p}{\Gamma \vdash_{\Pi} \lambda x^{\alpha}. M : (S, \alpha, p)} \text{ABSTRACTION}$$

$$\frac{\Gamma \vdash_{\Pi} M : (S, \alpha, p) \quad N \in \Lambda_{\Sigma}^{\alpha} \quad \forall q \in S. \Gamma \vdash_{\Pi} N : q}{\Gamma \vdash_{\Pi} MN : p} \text{APP}$$

As a shorthand we may write that  $\Gamma \vdash M : S$  is *derivable* to express that for all  $p$  in  $S$ ,  $\Gamma \vdash M : p$  is derivable.

An immediate consequence of the fact that  $\cap_{\Pi}^{\alpha}$  is finite for every  $\alpha \in \mathcal{T}_{\Sigma}$  is that type checking is decidable.

<sup>3</sup>  $\omega$  stands for the universal type, *i.e.* every term can be typed by  $\omega$ , and thus  $\omega$  represents the set of all  $\lambda$ -terms.

<sup>4</sup> intuitively  $\omega_{\alpha}$  stands for the simply typed version of  $\omega$  and it is the universal intersection type of the terms that belong to  $\Lambda_{\Sigma}^{\alpha}$ .

The properties of terms which are typed in an intersection signature are quite similar to the properties one usually obtains with intersection types. The most important one is that judgements are invariant modulo  $\beta$ -convertibility.

Before we come to prove this strong property we formulate some simple technical lemmas. In the following we assume that we are given a HOS  $\Sigma$  and a HOIS over  $\Sigma$ ,  $\Pi$ .

**Lemma 1** *If  $\Gamma \vdash_{\Pi} M : p$  is derivable then there is  $\alpha$  in  $\mathcal{T}_{\mathcal{A}}$  such that  $M \in \Lambda_{\Sigma}^{\alpha}$  and  $p \in \cap_{\Pi}^{\alpha}$ .*

*Proof* Simple induction on the derivation of  $\Gamma \vdash_{\Pi} M : p$ .

**Lemma 2** *If  $\Gamma \vdash_{\Pi} M : p$  is derivable and  $\Delta$  is a  $\Pi$ -type environment then  $\Gamma, \Delta \vdash_{\Pi} M : p$  is derivable.*

*Proof* Simple induction on the derivation of  $\Gamma \vdash_{\Pi} M : p$ .

**Lemma 3** *If  $\Gamma \vdash_{\Pi} M : p$  is derivable then for all  $\Pi$ -type environments  $\Delta$  verifying:*

1. *for all  $x^{\alpha} \in \mathcal{V}$ ,  $\Delta(x^{\alpha}) \subseteq \Gamma(x^{\alpha})$  and*
2. *for all  $x^{\alpha} \in FV(M)$ ,  $\Delta(x^{\alpha}) = \Gamma(x^{\alpha})$ .*

*$\Delta \vdash_{\Pi} M : p$  is derivable.*

*Proof* Simple induction on the derivation of  $\Gamma \vdash_{\Pi} M : p$ .

We now prove that this derivation system enjoys the substitution lemma.

**Lemma 4** *Given  $N \in \Lambda_{\Sigma}^{\alpha}$ , if  $\Gamma, x^{\alpha} : S \vdash_{\Pi} M : p$  and if for all  $q \in S$ ,  $\Gamma_q \vdash_{\Pi} N : q$  then  $\Gamma, \Gamma_S \vdash_{\Pi} M[x^{\alpha} := N] : p$ , where  $\Gamma_S(y^{\beta}) = \bigcup_{q \in S} \Gamma_q(y^{\beta})$ .*

*Proof* We proceed by induction on the derivation of  $\Gamma, x^{\alpha} : S \vdash_{\Pi} M : p$ :

1. If the derivation is of the form

$$\frac{p \in \Gamma(z^{\gamma})}{\Gamma, x^{\alpha} : S \vdash_{\Pi} z^{\gamma} : p}$$

there are two cases: either  $z^{\gamma} = x^{\alpha}$ , then using Lemma 2 we get that  $\Gamma, \Gamma_S \vdash_{\Pi} N : \alpha$  is derivable and the conclusion follows immediately; or  $z^{\gamma} \neq x^{\alpha}$  and in that case the conclusion is also trivial with the use of Lemma 2 and Lemma 3.

2. If the derivation is of the form:

$$\frac{p \in f(c)}{\Gamma, x^{\alpha} : S \vdash_{\Pi} c : p}$$

then the conclusion is trivial.

3. If the derivation is of the form:

$$\frac{\Gamma, x^{\alpha} : S, y^{\gamma} : S' \vdash_{\Pi} M : p}{\Gamma, x^{\alpha} : S \vdash_{\Pi} \lambda y^{\gamma}.M : (S', \gamma, p)}$$

we suppose that  $y^{\gamma} \notin FV(N)$  (this is possible thanks to  $\alpha$ -conversion), thus  $\lambda y^{\gamma}.M[x^{\alpha} := N] = (\lambda y^{\gamma}.M)[x^{\alpha} := N]$ . By induction hypothesis, we get that  $\Gamma, \Gamma_S, y^{\gamma} : S' \vdash_{\Pi} M[x^{\alpha} := N] : p$  is derivable, and thus we obtain the result by abstracting over  $y^{\gamma}$ .



4. Suppose the derivation is of the form:

$$\frac{\Gamma, x^\alpha : S \vdash_{\Pi} M_1 : (S', \gamma, p) \quad M_2 \in \Lambda_{\Sigma}^{\gamma} \quad \forall q \in S'. \Gamma, x^\alpha : S \vdash_{\Pi} M_2 : q}{\Gamma, x^\alpha : S \vdash_{\Pi} M_1 M_2 : p}$$

By induction hypothesis,  $\Gamma, \Gamma_S \vdash_{\Pi} M_1[x^\alpha := N] : (S', \gamma, p)$  is derivable and for all  $q \in S'$ ,  $\Gamma, \Gamma_S \vdash_{\Pi} M_2[x^\alpha := N] : q$  is derivable. The conclusion follows using APP.

We now prove that the converse of the substitution lemma, the extraction lemma, holds for this system.

**Lemma 5** *If  $N \in \Lambda_{\Sigma}^{\alpha}$ ,  $\Gamma \vdash_{\Pi} M[x^\alpha := N] : p$  then there is  $S \subseteq \cap_{\Pi}^{\alpha}$  such that:*

1.  $\Gamma, x^\alpha : S \vdash M : p$  is derivable
2. for all  $q \in S$ ,  $\Gamma \vdash_{\Pi} N : q$  is derivable

*Proof* We proceed by induction on the structure of  $M$ :

1. If  $M = z^\gamma$  (resp.  $M = c$ ) with  $z^\gamma \neq x^\alpha$ , then it suffices to take  $S = \emptyset$ .
2. If  $M = x^\alpha$  then we have that  $\Gamma \vdash_{\Pi} N : p$  is derivable and since  $N \in \Lambda_{\Pi}^{\alpha}$ , lemma 1 implies that  $p \in \cap_{\Pi}^{\alpha}$ . Thus,  $\Gamma, x^\alpha : \{p\} \vdash_{\Pi} x^\alpha : p$  is also derivable. The conclusion trivially follows with the choice  $S = \{p\}$ .
3. If  $M = M_1 M_2$ , and the derivation of  $\Gamma \vdash_{\Pi} M[x^\alpha := N] : p$  is of the form:

$$\frac{\Gamma \vdash_{\Pi} M_1[x^\alpha := N] : (S', \beta, p) \quad M_2 \in \Lambda_{\Sigma}^{\beta} \quad \forall q \in S'. \Gamma \vdash_{\Pi} M_2[x^\alpha := N] : q}{\Gamma \vdash_{\Pi} (M_1 M_2)[x^\alpha := N] : p}$$

by induction hypothesis, we have the existence of  $S_1$  such that:

- (a) for all  $q \in S_1$ ,  $\Gamma \vdash_{\Pi} N : q$  is derivable
- (b)  $\Gamma, x^\alpha : S_1 \vdash_{\Pi} M_1 : (S', \beta, p)$  is derivable

The induction hypothesis also implies that for all  $q \in S'$  there is  $S_q$  such that:

- (a) for all  $q' \in S_q$ ,  $\Gamma \vdash_{\Pi} N : q'$  is derivable
- (b)  $\Gamma, x^\alpha : S_q \vdash_{\Pi} M_2 : q$  is derivable

We let  $S = S_1 \cup \bigcup_{q \in S'} S_q$ . Then, with Lemma 2, we obtain the derivability of  $\Gamma, x^\alpha : S \vdash_{\Pi} M_1 : (S', \beta, p)$  and  $\Gamma, x^\alpha : S \vdash_{\Pi} M_2 : q$  for all  $q \in S'$ . This ends the proof in this case.

4. If  $M = \lambda y^\beta. M'$ , then  $p = (S', \beta, p')$  and the derivation of  $\Gamma \vdash_{\Pi} M[x^\alpha := N] : p$  must be of the form:

$$\frac{\Gamma, y^\beta : S' \vdash_{\Pi} M'[x^\alpha := N] : p'}{\Gamma \vdash_{\Pi} (\lambda y^\beta. M')[x^\alpha := N] : (S', \beta, p')}$$

we may suppose without loss of generality that  $y^\beta \notin FV(N)$  which implies that  $(\lambda y^\beta. M')[x^\alpha := N] = \lambda y^\beta. M'[x^\alpha := N]$ . By induction hypothesis, we have the existence of  $S \subseteq \cap_{\Pi}^{\alpha}$ , such that:

- (a) for all  $q \in S$ ,  $\Gamma, y^\beta : S' \vdash_{\Pi} N : q$  is derivable
- (b)  $\Gamma, x^\alpha : S, y^\beta : S' \vdash_{\Pi} M : p$  is derivable

From Lemma 3, and the fact that  $y^\beta \notin FV(N)$  we have that  $\Gamma \vdash_{\Pi} N : q$  is derivable. By using the abstraction rule, we obtain that  $\Gamma, x^\alpha : S \vdash_{\Pi} \lambda y^\beta. M : (S', \beta, p)$  is derivable and this concludes the proof.

We are now in position to prove the subject reduction and the subject expansion theorems.

**Theorem 1** *If  $\Gamma \vdash_{\Pi} M : p$  is derivable and  $M \xrightarrow{*}_{\beta} N$  then  $\Gamma \vdash N : p$  is also derivable.*

*Proof* We first prove that the property holds when  $M \rightarrow_{\beta} N$ . In such a case we have  $M = C[(\lambda x^{\alpha}.M_1)M_2]$  and  $N = C[M_1[x^{\alpha} := M_2]]$ . The proof is done by a trivial induction on the structure of  $C[\ ]$ , the initial case being proved using the substitution lemma for intersection types.

We then iterate this result to obtain the theorem.

**Theorem 2** *If  $M \in \Lambda_{\Sigma}$ ,  $M \xrightarrow{*}_{\beta\eta} N$  and  $\Gamma \vdash_{\Pi} N : p$  is derivable then  $\Gamma \vdash_{\Pi} M : p$  is also derivable.*

*Proof* To prove this theorem, we first prove that:

1. if  $\Gamma \vdash_{\Pi} M : (S, \beta, p)$  is derivable,  $x^{\beta} \notin FV(M)$  and  $\Gamma(x^{\beta}) = \emptyset$  then  $\Gamma \vdash_{\Pi} \lambda x^{\beta}.(Mx^{\beta}) : (S, \beta, p)$  is derivable with the following derivation:

$$\frac{\frac{\frac{\Gamma, x^{\beta} : S \vdash_{\Pi} M : (S, \beta, p) \quad x^{\beta} \in \Lambda_{\Sigma}^{\beta} \quad \forall q \in S. \quad \frac{q \in S}{\Gamma, x^{\beta} : S \vdash_{\Pi} x^{\beta} : q}}{\Gamma, x^{\beta} : S \vdash_{\Pi} (Mx^{\beta}) : p}}{\Gamma \vdash_{\Pi} \lambda x^{\beta}.(Mx^{\beta}) : (S, \beta, p)}}$$

We indeed have that  $\Gamma, x^{\beta} : S \vdash_{\Pi} M : (S, \beta, p)$  (by Lemma 2 and since  $\Gamma \vdash_{\Pi} M : (S, \beta, p)$  is derivable).

2. if  $M_2 \in \Lambda_{\Sigma}^{\alpha}$  and  $\Gamma \vdash_{\Pi} M_1[x^{\alpha} := M_2] : p$  is derivable then  $\Gamma \vdash_{\Pi} (\lambda x^{\alpha}.M_1)M_2 : p$  is derivable. Indeed, from the extraction lemma we know that there exists  $S \subseteq \cap_{\Pi}^{\alpha}$ , such that:

- (a) for all  $q \in S$ ,  $\Gamma \vdash_{\Pi} M_2 : q$  is derivable
- (b)  $\Gamma, x^{\alpha} : S \vdash_{\Pi} M_1 : p$  is derivable

We then get:

$$\frac{\frac{\Gamma, x^{\alpha} : S \vdash_{\Pi} M_1 : p}{\Gamma \vdash_{\Pi} \lambda x^{\alpha}.M_1 : (S, \alpha, p)} \quad M_2 \in \Lambda_{\Sigma}^{\alpha} \quad \forall q \in S. \Gamma \vdash_{\Pi} M_2 : q}{\Gamma \vdash_{\Pi} (\lambda x^{\alpha}.M_1)M_2 : p}}$$

Observe that this also works in the case where  $x^{\alpha}$  is not free in  $M_1$  (in that case,  $S = \emptyset$ ).

Then these results are extended to the case of a one step reduction by induction on the structure on  $M$ . The theorem is finally obtained by iterating the one step reduction case.

Note that this last theorem only holds when  $M \in \Lambda_{\Sigma}$ , that is when  $M$  is simply typed.

As one may remark, Theorem 1 only gives subject reduction for  $\beta$ -reduction. This is actually due to the fact that derivability is not preserved under  $\eta$ -contraction. One can derive the sequent  $x : a \rightarrow c \vdash \lambda y.xy : a \cap b \rightarrow c$  with the following derivation:

$$\frac{\frac{\frac{y : a \cap b, x : a \rightarrow c \vdash x : a \rightarrow c \quad y : a \cap b, x : a \rightarrow c \vdash y : a}{y : a \cap b, x : a \rightarrow c \vdash xy : c}}{x : a \rightarrow c \vdash \lambda y.xy : a \cap b \rightarrow c}}$$

while  $\lambda y.xy \rightarrow_\eta x$  and it is not possible to derive  $x : a \rightarrow c \vdash x : a \cap b \rightarrow c$ .

In order to fix this problem in the typing system, we need define the following subtyping relation on intersection types:

$$\frac{\iota \in \rho^{-1}(\alpha)}{\iota \sqsubseteq^\alpha \iota} \quad \frac{T \subseteq \cap_{\Pi}^\alpha \quad \forall p \in S. \exists q \in T. q \sqsubseteq^\alpha p}{T \preceq^\alpha S} \quad \frac{S \preceq^\alpha T \quad q \sqsubseteq^\beta p}{(T, \alpha, q) \sqsubseteq^{\alpha \rightarrow \beta} (S, \alpha, p)}$$

We may add the following typing rule:

$$\frac{\Gamma \vdash_{\Pi} M : p \quad p \sqsubseteq^\alpha q}{\Gamma \vdash_{\Pi} M : q}$$

When a derivation uses this subtyping rule, we write the judgement  $\Gamma \vdash_{\Pi}^{\sqsubseteq} M : p$ . Actually on terms in long form we will see that any derivation established by  $\vdash_{\Pi}^{\sqsubseteq}$  can also be derived by  $\vdash_{\Pi}$ .

We first give some basic properties concerning  $\sqsubseteq^\alpha$ .

We first state that  $\sqsubseteq^\alpha$  is a preorder on  $\cap_{\Pi}^\alpha$  (we leave the proof to the reader).

**Lemma 6** *The relation  $\sqsubseteq^\alpha$  verifies the following properties:*

1. if  $p \sqsubseteq^\alpha q$  then  $p, q \in \cap_{\Pi}^\alpha$
2. if  $p \in \cap_{\Pi}^\alpha$  then  $p \sqsubseteq^\alpha p$ ,
3. if  $p_1, p_2, p_3 \in \cap_{\Pi}^\alpha$ ,  $p_1 \sqsubseteq^\alpha p_2$  and  $p_2 \sqsubseteq^\alpha p_3$  then  $p_1 \sqsubseteq^\alpha p_3$ .

**Lemma 7** *If  $\Gamma \vdash_{\Pi}^{\sqsubseteq} M : p$  is derivable and  $M \xrightarrow{*}_\eta N$  then  $\Gamma \vdash_{\Pi}^{\sqsubseteq} N : p$  is derivable.*

*Proof* We use the same method as for subject reduction for the type system  $\vdash_{\Pi}$ . We show that if  $M = \lambda x^\alpha. N x^\alpha$  so that  $x^\alpha \notin FV(N)$ , then whenever  $\Gamma \vdash_{\Pi}^{\sqsubseteq} M : p$  is derivable we have that  $\Gamma \vdash_{\Pi}^{\sqsubseteq} N : p$  is also derivable. We extend this property to the case where  $M \rightarrow_\eta N$  by an easy induction on the structure of  $M$ . Finally we obtain the result by iterating this last property.

**Lemma 8** *Suppose that  $\Gamma \vdash_{\Pi} M : p$  is derivable and that  $p \sqsubseteq^\alpha q$  then there is  $M' \xrightarrow{*}_\eta M$  such that  $\Gamma \vdash_{\Pi} M' : q$  is derivable.*

*Proof* By Lemma 6 we know that  $p \in \cap_{\Pi}^\alpha$ , thus, with Lemma 1 we obtain that  $M \in \Lambda_{\Sigma}^\alpha$ . We then prove this lemma by induction on the structure of  $\alpha$ :

1. if  $\alpha$  is atomic then obviously  $p = q$  and we may conclude by choosing  $M' = M$ .
2. if  $\alpha = \alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \alpha_0$  with  $\alpha_0$  being atomic then we have that:
  - (a)  $p = (S_1, \alpha_1, (\dots (S_n, \alpha_n, \iota) \dots))$
  - (b)  $q = (T_1, \alpha_1, (\dots (T_n, \alpha_n, \iota) \dots))$
  - (c) for all  $k \in [1, n]$ ,  $T_k \preceq^{\alpha_k} S_k$  so that for all  $p' \in S_k$  there is  $q' \in T_k$  such that  $q' \sqsubseteq p'$ ,
  - (d) and  $\iota \in \rho(\alpha_0)$ .

Let's take fresh variables,  $y_1^{\alpha_1}, \dots, y_n^{\alpha_n}$ , which are not free in  $M$  and are associated to the empty set by  $\Gamma$ . We let  $\Delta = y_1^\alpha : T_1, \dots, y_n^\alpha : T_n$  then for all  $k \in [1, n]$  and all  $q' \in T_k$ ,  $\Gamma, \Delta \vdash_{\Pi} y_k^{\alpha_k} : q'$  is derivable. Thus given  $p' \in S_k$ , the induction hypothesis gives us the existence of  $M_k^{p'}$  such that  $M_k^{p'} \xrightarrow{*}_\eta y_k^{\alpha_k}$  and  $\Gamma, \Delta \vdash_{\Pi} M_k^{p'} : p'$  is derivable. Since derivability is preserved by  $\eta$ -expansion and that  $\eta$ -expansion is

confluent, there is  $M_k$  such that  $M_k \xrightarrow{*}_\eta y_k^{\alpha_k}$  and for all  $p' \in S_k$ ,  $\Gamma, \Delta \vdash_{\Pi} M_k : p'$ . Thus we have that  $\Gamma, \Delta \vdash_{\Pi} MM_1 \dots M_n : \iota$  is derivable and then

$$\Gamma \vdash_{\Pi} \lambda y_1^{\alpha_1} \dots y_n^{\alpha_n}. MM_1 \dots M_n : (T_1, \alpha_1, (\dots (T_n, \alpha_n, \iota) \dots))$$

is derivable. The proof is finished since we have:

$$\begin{aligned} \lambda y_1^{\alpha_1} \dots y_n^{\alpha_n}. MM_1 \dots M_n &\xrightarrow{*}_\eta \lambda y_1^{\alpha_1} \dots y_n^{\alpha_n}. My_1^{\alpha_1} \dots y_n^{\alpha_n} \\ &\xrightarrow{*}_\eta M \end{aligned}$$

Together with Lemma 7, this last lemma shows that the subject reduction and the subject expansion theorems can easily be transferred from the type system  $\vdash_{\Pi}$  to the system  $\vdash_{\overline{\Pi}}$ .

**Theorem 3** *If  $M$  is in long form,  $\Gamma \vdash_{\Pi} M : p$  is derivable and  $p \sqsubseteq^\alpha q$ . Then  $\Gamma \vdash_{\Pi} M : q$  is derivable.*

*Proof* From the previous lemma we have the existence of  $M'$  such that  $M' \xrightarrow{*}_\eta M$  and  $\Gamma \vdash_{\Pi} M' : q$  is derivable. But since  $M$  is in long form,  $M' \xrightarrow{*}_\beta M$  and from the subject reduction theorem for  $\beta$ -reduction we get that  $\Gamma \vdash_{\Pi} M : q$  is derivable.

**Theorem 4** *Suppose that  $M'$  is the long form of  $M$  then  $\Gamma \vdash_{\overline{\Pi}} M : p$  is derivable if and only if  $\Gamma \vdash_{\Pi} M' : p$  is derivable.*

*Proof* The first direction of the implication is an immediate consequence of the previous theorem. For the second direction it suffices to use Lemma 7.

So the judgments of the derivation system  $\vdash_{\overline{\Pi}}$  are closed under  $\beta\eta$ -convertibility when the subjects are elements of  $\Lambda_{\Sigma}$ , but derivability in  $\vdash_{\overline{\Pi}}$  is equivalent to derivability in  $\vdash_{\Pi}$  on long forms. This gives us a nice tool to avoid the subsumption rule in inductions.

HOIS may express very strong properties about the terms. In particular the theorem we prove in the next section shows that they can characterize  $\beta\eta$ -convertible classes of  $\lambda$ -terms.

#### 4 Defining singleton sets with intersection types

The main interest for this paper of higher order intersection signatures is that they can be used to describe with static typing any *singleton set*, i.e sets of the form  $\{N \in \Lambda_{\Sigma} \mid N =_{\beta\eta} M\}$ . Characterizing singleton sets with simple types has already been explored. The first result we are aware of is [BS82] which proves that simple typing enables to define any singleton sets of *affine* terms. This result has been extended in several directions by [AO94], [Aot99], [Tat99] and [HT00]. But it is well known that simple types cannot sufficiently constrain the set of terms they type so as to define any singleton set. The simplest example is given by Church numerals which, when greater than 2, have  $(o \rightarrow o) \rightarrow o \rightarrow o$  as most general type. Therefore, a singleton set containing any church numeral greater than 2 cannot be defined by means simple types. This is why a stronger notion of typing is necessary. In this section we show that this can be done with HOIS. As an example, we can illustrate how intersection types can discriminate Church numerals. Indeed, one can see that  $M =$

$\lambda f^{o \rightarrow o} x^o . \underbrace{f^{o \rightarrow o}(\dots (f^{o \rightarrow o} x^o) \dots)}_{n \times}$ , the church numeral  $n$ , is the only  $\lambda$ -term, modulo

$\beta\eta$ -convertibility, that can be typed with  $(0 \rightarrow 1 \cap \dots \cap (n-1) \rightarrow n) \rightarrow 0 \rightarrow n$  in the HOIS  $(\Sigma, \{0; \dots; n\}, \rho, f)$  where  $\Sigma$  is a signature that does not declare any constant and only uses one atomic type, namely  $o$ ,  $\rho$  is a function that maps every  $k$  in  $\{0; \dots; n\}$  to  $o$ , and  $f$  is the empty function.

As usual in what follows, unless stated otherwise, we assume that we are given a HOS  $\Sigma$  and an intersection signature over  $\Sigma$  noted  $\Pi$ .

Our theorem, the singleton set theorem, relies on a precise account of the occurrences of atomic formulae.

**Definition 1** Given  $\iota \in I$  and  $p \in \cap \Pi$ , the number of occurrences of  $\iota$  in  $p$ ,  $|p|_\iota$  is defined as:

1. if  $p \in I$  then  $|p|_\iota = \begin{cases} 0 & \text{if } p \neq \iota \\ 1 & \text{if } p = \iota \end{cases}$
2. if  $p = (S, \alpha, p')$  then  $|p|_\iota = |p'|_\iota + \sum_{q \in S} |q|_\iota$

Given  $\Gamma$ , a  $\Pi$ -type environment, the number of occurrences of  $\iota$  in  $\Gamma$ , written  $|\Gamma|_\iota$ , is equal to  $\sum_{x^\alpha \in \mathcal{V}_\Sigma} \sum_{q \in \Gamma(x^\alpha)} |q|_\iota$ . The number of occurrences of  $\iota$  in the types of the constants of  $\Pi$  is defined as  $|\Pi|_\iota = \sum_{c \in \mathcal{C}} \sum_{q \in f(c)} |q|_\iota$ .

We also need the usual notion of polarity which in a formula distinguishes, for any derivation, the subformulae that are bound to be used as hypothesis (*i.e.* used as the type of a variable) and the subformulae that will have to be proved (*i.e.* used as the type of a  $\lambda$ -term).

**Definition 2** Given a higher order intersection signature  $\Pi$  and  $p \in \cap \Pi$ , we say that  $q \in \cap \Pi$  (*resp.*  $S \subseteq \cap \Pi$ ) occurs positively in  $p$  if and only if:

1.  $q = p$  or,
2.  $p = (T, \alpha, p')$  and either  $q$  (*resp.*  $S$ ) occurs positively in  $p'$  or  $q$  occurs negatively in  $p'' \in T$  (*resp.* or  $S$  occurs negatively in  $p'' \in T$ ).

We say that  $q \in \cap \Pi$  (*resp.*  $S \subseteq \cap \Pi$ ) occurs negatively in  $p$  if and only if  $p = (T, \alpha, p')$  and either  $q$  (*resp.*  $S$ ) occurs negatively in  $p'$  or  $q$  occurs positively in  $p'' \in T$  (*resp.*  $S = T$  or  $S$  occurs positively in  $p'' \in T$ ).

We say that  $q$  (*resp.*  $S$ ) occurs positively or negatively in a  $\Pi$ -type environment  $\Gamma$  whenever there is  $x^\alpha \in \mathcal{V}_\Sigma$  such that  $q$  (*resp.*  $S$ ) occurs positively or negatively in  $p \in \Gamma(x^\alpha)$ .

The difference between constants and variables is rather small. It stems from the fact that variables are declared directly in  $\Pi$ -type environments whereas constants are declared in  $\Pi$ , the HOIS. The proof of the singleton set theorem requires a precise management of the  $\Pi$ -type environments for the variables. Concerning constants, such a management is performed by operations on higher order intersection signatures. So we give here an operation that allows to discard a type assignment in a  $\Pi$ -type environment and its counter-part at the level of higher order intersection signatures.

**Definition 3** Given  $\Gamma$  a  $\Pi$ -type environment, we define  $\Gamma \setminus x^\alpha : q$  to be the  $\Pi$ -context such that:

$$(\Gamma \setminus x^\alpha : q)(y^\beta) = \begin{cases} \Gamma(x^\alpha) \setminus \{q\} & \text{when } y^\beta = x^\alpha \\ \Gamma(y^\beta) & \text{otherwise} \end{cases}$$

**Definition 4** Given a higher order intersection signature  $\Pi = (\Sigma, I, \rho, f)$  we define the signature obtained from  $\Pi$  by suppressing the declaration  $c : p$ , written  $\Pi \setminus c : p = (\Sigma, I, \rho, g)$ , where:

$$g(d) = \begin{cases} f(d) & \text{if } d \neq c \\ f(d) \setminus \{p\} & \text{if } d = c \end{cases}$$

**Definition 5** Given  $\Gamma$  a  $\Pi$ -type environment and  $p \in \cap_{\Pi}$ , we say that the pair  $(\Gamma, p)$  is balanced relatively to  $\Pi$  if it fulfills the following properties:

1. Each element  $\iota \in I$  has at most two occurrences in  $\Gamma, p$  and  $\Pi$ , i.e.  $|\Gamma|_{\iota} + |p|_{\iota} + |\Pi|_{\iota} \leq 2$ .
2. There is no negative occurrence of  $\emptyset$  in  $\Gamma$  and  $\bigcup_{c \in C} f(c)$ .
3. There is no positive occurrence of  $\emptyset$  in  $p$ .

A sequent  $\Gamma \vdash_{\Pi}^{\square} M : p$  is said to be balanced whenever the pair  $(\Gamma, p)$  is balanced relatively to  $\Pi$ .

When  $\Pi$  is clear from the context we will simply write that a pair  $(\Gamma, p)$  is balanced.

We now show that if  $(\Gamma, p)$  is balanced then all the terms such that  $\Gamma \vdash_{\Pi} M : p$  is derivable are pairwise  $\beta\eta$ -convertible. The proof of this property relies on the following lemma:

**Lemma 9** Given  $q = (S_1, \alpha_1, (\dots (S_n, \alpha_n, \iota) \dots)) \in \cap_{\Pi}$  such that  $|q|_{\iota} = 1$  we have:

1. If  $\Gamma \vdash_{\Pi}^{\square} M : p$  is derivable,  $q \in \Gamma(x^{\alpha})$  and  $|\Gamma \setminus x^{\alpha} : q|_{\iota} + |p|_{\iota} + |\Pi|_{\iota} = 0$  then  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} M : p$  is derivable.
2. If  $\Gamma \vdash_{\Pi}^{\square} M : p$  is derivable,  $q \in f(c)$  for some  $c \in C$ ,  $|\Gamma|_{\iota} + |p|_{\iota} + |\Pi \setminus c : q|_{\iota} = 0$  then  $\Gamma \vdash_{\Pi \setminus c : q}^{\square} M : p$  is derivable.

*Proof* Using the subject reduction and the subject expansion theorems, we may assume without loss of generality that  $M$  is in long normal form; furthermore, the use of the Theorem 4 allows us to consider that the derivation does not use the subsumption rule. Thus, in general, either  $M = y^{\beta} M_1 \dots M_m$ , or  $M = b M_1 \dots M_m$ , or  $M = \lambda y^{\beta}. M'$ . As the second statement of this lemma can be proved in the same way as the first one, we only give the proof of the first statement.

To prove the first statement of the lemma we proceed by induction on the structure of  $M$ :

1. if  $M = y^{\beta} M_1 \dots M_m$  (resp.  $M = b M_1 \dots M_m$ ) then there is

$$q' = (T_1, \beta_1, (\dots (T_m, \beta_m, p) \dots))$$

such that:

- (a)  $q' \in \Gamma(y^{\beta})$  (resp.  $q' \in f(b)$ )
- (b) Since  $|p|_{\iota} = 0$ ,  $q \neq q'$  and  $q' \in (\Gamma \setminus x^{\alpha} : q)(y^{\beta})$
- (c)  $M_i$  is in long normal form and  $M_i \in A_{\Sigma}^{\beta_i}$
- (d) Since  $|\Gamma \setminus x^{\alpha} : q|_{\iota} = 0$ , for all  $q'' \in T_i$ ,  $|q''|_{\iota} = 0$
- (e) For all  $q'' \in T_i$ ,  $\Gamma \vdash_{\Pi} M_i : q''$  is derivable

Then by induction hypothesis, we get that for all  $i \in [1, m]$  and all  $q'' \in T_i$ ,  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} M_i : q''$  is derivable. Since  $q' \in (\Gamma \setminus x^{\alpha} : q)(y^{\beta})$  and  $\Gamma \setminus x^{\alpha} : q$  is  $\Pi$ -type environment, we obtain that  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} y^{\beta} : q'$  (resp.  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} b : q'$ ) is derivable. Thus by a repeated use of the application rule we obtain that  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} y^{\beta} M_1 \dots M_m : p$  (resp.  $\Gamma \setminus x^{\alpha} : q \vdash_{\Pi} b M_1 \dots M_m : p$ ) is derivable.

2. if  $M = \lambda y^\beta . M'$  then  $p = (S, \beta, p')$  and the derivation of  $\Gamma \vdash_{\Pi} M : p$  must be like:

$$\frac{\Gamma, y^\beta : S \vdash_{\Pi} M' : p'}{\Gamma \vdash_{\Pi} \lambda y^\beta . M' : (S, \beta, p')}$$

as there is no occurrence of  $\iota$  in the elements of  $S$  or in  $p'$ , we can apply the induction hypothesis and obtain that  $(\Gamma \setminus x^\alpha : q), y^\beta : S \vdash_{\Pi} M' : p'$  is derivable and then that  $\Gamma \setminus x^\alpha : q \vdash_{\Pi} M : p$  is also derivable.

We are now in a position to prove the property we wanted for balanced judgements.

**Theorem 5** *If  $\Gamma \vdash_{\Pi}^{\square} M : p$  and  $\Gamma \vdash_{\Pi}^{\square} N : p$  are balanced and derivable then  $M =_{\beta\eta} N$ .*

*Proof* Because we have the subject reduction and the subject expansion theorems for derivations in higher-order intersection signatures with Theorem 4, proving this theorem boils down to proving that given a balanced pair  $(\Gamma, p)$  there is at most one long normal term  $M$  such that  $\Gamma \vdash_{\Pi} M : p$  is derivable.

So let us suppose we are given a balanced pair  $(\Gamma, p)$  and a long normal term  $M$  such that  $\Gamma \vdash_{\Pi} M : p$  is derivable. We prove that  $M$  is unique. The uniqueness of  $M$  is shown by induction on its structure:

1. if  $M = x^\alpha M_1 \dots M_n$  then  $p \in I$  and:
  - (a) there is  $q = (S_1, \alpha_1, (\dots (S_n, \alpha_n, p) \dots))$  in  $\Gamma(x^\alpha)$ .
  - (b) For all  $i \in [1, n]$  and  $q' \in S_i$ ,  $\Gamma \vdash M_i : q'$  is derivable.
  - (c) Since  $(\Gamma, p)$  is balanced,  $|\Gamma \setminus x^\alpha : q|_p + |\Pi|_p = 0$  and for all  $i \in [1, n]$ ,  $S_i \neq \emptyset$ . Let  $\Gamma' = \Gamma \setminus x^\alpha : q$ , from the previous lemma, we have that  $\Gamma' \vdash_{\Pi} M_i : q'$  is derivable for all  $q' \in S_i$ . We also have that for  $\iota \in I$ ,  $|\Gamma'|_\iota + |\Pi|_\iota + |q'|_\iota \leq |\Gamma|_\iota + |\Pi|_\iota + |p|_\iota \leq 2$  which implies that  $(\Gamma', q')$  is balanced. We can then apply the induction hypothesis to  $\Gamma' \vdash_{\Pi} M_i : q'$  which gives that any  $N$  in long normal form such that  $\Gamma' \vdash_{\Pi} N : q'$  is equal to  $M_i$ . Now let us suppose that there is  $N$ , a term in long normal form, such that  $\Gamma \vdash_{\Pi} N : q'$  is derivable. The use of the previous lemma gives that  $\Gamma' \vdash_{\Pi} N : q'$  is also derivable and finally that  $N = M_i$ . The fact that there is no negative occurrence of  $\emptyset$  implies that no  $S_i$  is empty and, then, that each of the  $M_i$  is uniquely defined. Since there is no other possible choice concerning the head of  $M$  we can conclude that  $M$  is unique.
2. if  $M = cM_1 \dots M_n$  then we proceed similarly to the previous case.
3. if  $M = \lambda x^\alpha . M'$  then the derivation must be of the form:

$$\frac{\Gamma, x^\alpha : S \vdash_{\Pi} M' : p'}{\Gamma \vdash_{\Pi} \lambda x^\alpha . M' : (S, \alpha, p')}$$

Obviously the pair  $(\Gamma, x^\alpha : S; p')$  is balanced, then by induction hypothesis,  $M'$  is the unique long normal term such that  $\Gamma, x^\alpha : S \vdash_{\Pi} M' : p'$  is derivable ; the unicity of  $M$  follows.

This theorem proves that balanced sequents of higher-order intersection signatures define singleton sets. The question is now whether all the  $\beta\eta$ -equivalence classes of terms built on a higher order signature  $\Sigma$  can be captured that way. The answer is positive and it leads to the singleton set theorem.

**Definition 6** Given two intersection signatures over  $\Sigma$ ,  $\Pi_1 = (\Sigma, I_1, \rho_1, f_1)$  and  $\Pi_2 = (\Sigma, I_2, \rho_2, f_2)$  such that for all  $\iota \in I_1 \cap I_2$ ,  $\rho_1(\iota) = \rho_2(\iota)$  then  $\Pi_1 \cup \Pi_2$ , the union of  $\Pi_1$  and  $\Pi_2$ , is defined as the intersection signature  $(\Sigma, I_1 \cup I_2, \rho, f)$  where:

1.  $\rho(\iota) = \begin{cases} \rho_1(\iota) & \text{if } \iota \in I_1 \\ \rho_2(\iota) & \text{if } \iota \in I_2 \end{cases}$
2.  $f(c) = f_1(c) \cup f_2(c)$

**Definition 7** Two signatures over  $\Sigma$ ,  $\Pi_1 = (\Sigma, I_1, \rho_1, f_1)$  and  $\Pi_2 = (\Sigma, I_2, \rho_2, f_2)$ , are independent if  $I_1 \cap I_2 = \emptyset$ .

The union of two independent intersection signatures is always defined. Remark that the union operation between signatures is associative and commutative.

**Theorem 6** Given  $M$  a term in long normal form then there is an intersection signature  $\Pi$  and a pair  $(\Gamma, p)$  such that  $\Gamma \vdash_{\Pi}^{\square} M : p$  is derivable and balanced.

*Proof* We proceed by induction on the structure of  $M$ :

1. if  $M = x^\beta M_1 \dots M_n$  then by induction hypothesis there is a family of intersection signatures  $(\Pi_k)_{k \in [1, n]}$  and a family of pairs  $((\Gamma_k, p_k))_{k \in [1, n]}$  such that for all  $k \in [1, n]$ ,  $\Gamma_k \vdash_{\Pi_k}^{\square} M_k : p_k$  is derivable and balanced.

We may suppose that the  $\Pi_k$  are pairwise independent. Let  $\iota'$  be a new atomic intersection type such that for all  $k \in [1, n]$ ,  $\iota' \notin I_k$ . If  $\beta = \beta_1 \rightarrow \dots \rightarrow \beta_n \rightarrow \alpha$ , we let  $\Pi' = (\Sigma, \{\iota'\}, \rho', f')$  with  $\rho'(\iota') = \alpha$  and  $f'$  is the empty function. We then let  $\Pi = \Pi' \cup \bigcup_{k \in [1, n]} \Pi_k$ ,  $p = (\{p_1\}, \beta_1, (\dots (\{p_n\}, \beta_n, \iota') \dots))$  and  $\Gamma$  be the  $\Pi$ -context verifying:

$$\Gamma(y^\alpha) = \begin{cases} \{p\} \cup \bigcup_{k \in [1, n]} \Gamma_k(x^\alpha) & \text{if } y^\alpha = x^\beta \\ \bigcup_{k \in [1, n]} \Gamma_k(y^\alpha) & \text{otherwise} \end{cases}$$

we have that for all  $\iota \in I_k$  (for some  $k \in [1, n]$ ),  $|\Gamma|_\iota + |\Pi|_\iota = |\Gamma_k|_\iota + |\Pi_k|_\iota + |p_k|_\iota$  and obviously  $\Gamma \vdash_{\Pi}^{\square} M : \iota'$  is derivable. To prove that  $(\Gamma, \iota')$  is balanced relatively to  $\Pi$  we just have to note that  $|\Gamma|_{\iota'} + |\Pi|_{\iota'} + |\iota'|_{\iota'} = 2$  and use the fact that  $(\Gamma_k, p_k)$  is balanced relatively to  $\Pi_k$ .

2. if  $M = cM_1 \dots M_n$  then the conclusion is proved using the same principle as in the previous case.
3. if  $M = \lambda x^\beta . M'$  the conclusion is a direct consequence of the induction hypothesis.

Finally we get the singleton set theorem as a corollary.

**Theorem 7** Given  $M \in \Lambda_\Sigma$ , there is  $\Pi$ ,  $\Gamma$  and  $p$  such that  $\Gamma \vdash_{\Pi}^{\square} M : p$  is derivable if and only if  $M =_{\beta\eta} N$ .

## 5 Decidability of the membership problem for second order ACGs

To obtain the decidability result we aim at, we start by showing a closure property of the languages of  $\lambda$ -terms defined by second order non-linear ACGs. We indeed prove that the class of languages defined by second order non-linear ACGs is closed under intersection with the set of terms typable on a higher-order intersection signature with all the types of a given set  $P$ . Then the use of the singleton set theorem reduces the membership problem to the problem of the emptiness of the language of a second order ACG, which is trivially decidable (it amounts to checking whether a local tree language is empty or not).



Before establishing the result, we settle some useful notations. For  $P \subseteq \cap_{\Pi}^{\alpha}$ , we will write  $\Gamma \vdash M : P$  is derivable as a shorthand for  $\Gamma \vdash M : p$  is derivable for all  $p$  in  $P$ .  $\Pi$  a higher order intersection signature,  $T_1 \subseteq \cap_{\Pi}^{\alpha}$  and  $T_2 \subseteq \cap_{\Pi}^{\beta}$  we define  $T_1 \rightarrow_{\alpha} T_2$  to be the set  $\{(T_1, \alpha, p) | p \in T_2\}$ . When taking  $T_1 \subseteq \cap_{\Pi}^{\alpha}$  and  $T_2 \subseteq \cap_{\Pi}^{\beta}$ , it is obvious that  $T_1 \rightarrow_{\alpha} T_2 \subseteq \cap_{\Pi}^{\alpha \rightarrow \beta}$ .

**Theorem 8** *Given a second order non-linear ACG  $\mathcal{G} = (\Sigma_0, \Sigma_1, \mathcal{L}, S)$ , an intersection signature  $\Pi = (\Sigma_1, I, \rho, f)$  and  $P \subseteq \cap_{\Pi}^{\mathcal{L}(S)}$ , there is a second order non-linear ACG  $\mathcal{G}'$  such that:*

$$\mathcal{O}(\mathcal{G}') = \{M \mid M \in \mathcal{O}(\mathcal{G}) \wedge \vdash_{\Pi}^{\square} M : P\}$$

*Proof* We define  $\mathcal{G}' = (\Sigma'_0, \Sigma_1, \mathcal{L}', \langle S, P \rangle)$  in the following way:

1.  $A'_0 = \{\langle \alpha, Q \rangle | \alpha \in A_0 \wedge Q \subseteq \cap_{\Pi}^{\mathcal{L}(\alpha)}\}$ ,
2.  $C'_0 = \{\langle c, \langle S_1, \dots, S_n \rangle, S_0 \rangle | \tau_0(c) = \alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \alpha_0 \wedge$   
 $\forall i \in [0, n]. S_i \subseteq \cap_{\Pi}^{\mathcal{L}(\alpha_i)} \wedge$   
 $\vdash_{\Pi}^{\square} \mathcal{L}(c) : S_1 \rightarrow_{\mathcal{L}(\alpha_1)} \dots \rightarrow_{\mathcal{L}(\alpha_{n-1})} S_n \rightarrow_{\mathcal{L}(\alpha_n)} S_0\}$ ,
3.  $\tau'_0(\langle c, \langle S_1, \dots, S_n \rangle, S_0 \rangle) = \langle \alpha_1, S_1 \rangle \rightarrow \dots \rightarrow \langle \alpha_n, S_n \rangle \rightarrow \langle \alpha_0, S_0 \rangle$  when  $\tau_0(c) = \alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \alpha_0$ ,
4.  $\mathcal{L}'(\langle \alpha, Q \rangle) = \mathcal{L}(\alpha)$ ,
5.  $\mathcal{L}'(\langle c, \langle S_1, \dots, S_n \rangle, S_0 \rangle) = \mathcal{L}(c)$ .

We claim that given  $\alpha \in A_0$ ,  $M \in A_{\Sigma_1}^{\mathcal{L}(\alpha)}$  and  $Q \subseteq \cap_{\Pi}^{\mathcal{L}(\alpha)}$ , the two following properties are equivalent:

1. there is a closed term  $N \in A_{\Sigma_0}^{\alpha}$  such that  $\mathcal{L}(N) =_{\beta\eta} M$ , and  $\vdash_{\Pi}^{\square} M : Q$  is derivable,
2. there is a closed term  $N' \in A_{\Sigma'_0}^{\langle \alpha, Q \rangle}$  such that  $\mathcal{L}'(N') =_{\beta\eta} M$ .

We first prove that 1 implies 2. We proceed by induction on the structure of  $N$ . Since  $\Sigma_0$  is a second order signature and  $\alpha$  is an atomic type, we have that  $N$  is of the form  $cN_1 \dots N_n$  (we assume in what follow that  $c$  has type  $\alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \alpha$ ). Thus,  $M =_{\beta\eta} \mathcal{L}(c)(\mathcal{L}(N_1)) \dots (\mathcal{L}(N_n))$  and from the Theorems 1, 2 and 4, and the fact that  $\vdash_{\Pi}^{\square} M : Q$  is derivable we have that the judgement:

$$\vdash_{\Pi}^{\square} \mathcal{L}(c)(\mathcal{L}(N_1)) \dots (\mathcal{L}(N_n)) : q$$

is derivable for all  $q \in Q$ . Therefore there are  $S_1^q, \dots, S_n^q$  such that:

1.  $\vdash_{\Pi}^{\square} \mathcal{L}(c) : p^q$  is derivable with  $p^q = (S_1^q, \mathcal{L}(\alpha_1), \dots (S_n^q, \mathcal{L}(\alpha_n), q) \dots)$ ,
2.  $\vdash_{\Pi}^{\square} \mathcal{L}(N_i) : S_i^q$  is derivable for all  $i$  in  $[1, n]$ .

Let  $S_i = \bigcup_{q \in Q} S_i^q$ , we get that for all  $i$  from  $[1, n]$ ,  $\vdash_{\Pi}^{\square} \mathcal{L}(N_i) : S_i$  is derivable; then by induction hypothesis we get that there is  $N'_i$  in  $A_{\Sigma'_0}^{\langle \alpha_i, S_i \rangle}$  such that  $\mathcal{L}'(N'_i) =_{\beta\eta} \mathcal{L}(N_i)$ . Moreover, if we let  $r^q = (S_1, \mathcal{L}(\alpha_1), \dots (S_n, \mathcal{L}(\alpha_n), q) \dots)$  we have that  $r^q \sqsubseteq p^q$  because for all  $i$  in  $[1, n]$ ,  $S_i^q \subseteq S_i$ . It follows that  $\vdash_{\Pi}^{\square} \mathcal{L}(c) : P$  is derivable with  $P = \bigcup_{q \in Q} r^q$ . We observe that actually:

$$P = S_1 \rightarrow_{\mathcal{L}(\alpha_1)} \dots \rightarrow_{\mathcal{L}(\alpha_{n-1})} S_n \rightarrow_{\mathcal{L}(\alpha_n)} Q$$

it follows that  $\langle c, \langle S_1, \dots, S_n \rangle, Q \rangle$  is an element of  $C'_0$  and its type is  $\langle \alpha_1, S_1 \rangle \rightarrow \dots \rightarrow \langle \alpha_n, S_n \rangle \rightarrow \langle \alpha, Q \rangle$ . So if we let  $N' = \langle c, \langle S_1, \dots, S_n \rangle, Q \rangle N'_1 \dots N'_n$  we get an element of  $\Lambda_{\Sigma'_0}^{\langle \alpha, Q \rangle}$  such that  $\mathcal{L}'(N') =_{\beta\eta} M$  because:

$$\begin{aligned} \mathcal{L}'(N') &= \mathcal{L}'(\langle c, \langle S_1, \dots, S_n \rangle, Q \rangle N'_1 \dots N'_n) \\ &= \mathcal{L}'(\langle c, \langle S_1, \dots, S_n \rangle, Q \rangle)(\mathcal{L}'(N'_1)) \dots (\mathcal{L}'(N'_n)) \\ &= \mathcal{L}(c)(\mathcal{L}'(N'_1)) \dots (\mathcal{L}'(N'_n)) \\ &=_{\beta\eta} \mathcal{L}(c)(\mathcal{L}(N_1)) \dots (\mathcal{L}(N_n)) \\ &=_{\beta\eta} M \end{aligned}$$

This ends the proof of the first part of the claim.

We now prove the second direction of the equivalence, that is 2 implies 1. We proceed by induction on the structure of  $N'$ . Since  $\langle \alpha, Q \rangle$  is an atomic type and since  $\Sigma'_0$  is a second order signature,  $N'$  must be of the form  $\langle c, \langle S_1, \dots, S_n \rangle, Q \rangle N'_1 \dots N'_n$  and  $N'_i$  is a closed element of  $\Lambda_{\Sigma'_0}^{\langle \alpha_i, S_i \rangle}$  (assuming that  $\tau_0(c) = \alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \alpha$ ). The induction hypothesis gives the existence of  $N_i$  in  $\Lambda_{\Sigma_0}^{\alpha_i}$  such that  $\mathcal{L}'(N'_i) =_{\beta\eta} \mathcal{L}(N_i)$ ; it also gives the fact that  $\vdash_{\Pi}^{\square} \mathcal{L}'(N'_i) : S_i$  is derivable. Since  $\mathcal{L}'(\langle c, \langle S_1, \dots, S_n \rangle, Q \rangle) =_{\beta\eta} \mathcal{L}(c)$  it follows that  $\mathcal{L}'(N') =_{\beta\eta} \mathcal{L}(cN_1 \dots N_n)$ . Furthermore, by definition of  $C'_0$  we have that  $\vdash_{\Pi}^{\square} \mathcal{L}(c) : S_1 \rightarrow_{\mathcal{L}(\alpha_1)} \dots \rightarrow_{\mathcal{L}(\alpha_{n-1})} S_n \rightarrow_{\mathcal{L}(\alpha_n)} Q$  is derivable and therefore we obtain that

$$\vdash_{\Pi}^{\square} \mathcal{L}'(N') : Q$$

is derivable. The conclusion easily follows the use of the subject reduction and subject expansion theorems.

Finally the theorem is a simple consequence of the claim.

Note that the construction of  $\mathcal{G}'$  is effective.

**Theorem 9** *Given  $\mathcal{G} = (\Sigma_1, \Sigma_2, \mathcal{L}, S)$  a second order non-linear ACG, and  $M$ , it is decidable whether  $M \in \mathcal{O}(\mathcal{G})$ .*

*Proof* The singleton set theorem gives the existence of  $\Pi$  and  $p$  such that for all  $N$ , the derivability of  $\vdash_{\Pi}^{\square} N : p$  implies that  $M =_{\beta\eta} N$ . The previous theorem allows us to effectively build an ACG  $\mathcal{G}'$  such that  $\mathcal{O}(\mathcal{G}') = \{N \mid N \in \mathcal{O}(\mathcal{G}) \wedge \vdash_{\Pi}^{\square} N : p\}$ . Thus  $\mathcal{O}(\mathcal{G}')$  is either empty if and only  $M \notin \mathcal{O}(\mathcal{G})$ . Thus checking the emptiness of  $\mathcal{G}'$  amounts to check  $M \notin \mathcal{O}(\mathcal{G})$  which shows our decidability result.

We conclude this section with some remarks. Theorem 8 shows that the languages of second order non-linear ACGs are closed under intersection with typing properties in HOISs. We may wonder whether this is particular to second order non-linear ACGs or whether it can be extended to other kinds of ACGs. We can show that this closure property does not hold in general for non-linear ACGs. We will not get into the details of the proof but sketch the general idea. Basically, this is based on the fact that the emptiness problem for non-linear ACGs is decidable whereas the emptiness of the set of terms being typable with intersection types and belonging to the language of a non-linear ACG is undecidable. The decidability of the emptiness of non-linear ACGs can be easily reduced to proof-search in minimal logic which is known to be decidable. The undecidability of the emptiness of sets of the form  $\{M \in \mathcal{O}(\mathcal{G}) \mid \vdash_{\Pi} M : P\}$  (for some

non-linear ACG  $\mathcal{G}$ ) can easily be reduced to the problem of  $\lambda$ -definability which is known to be undecidable [Loa01]<sup>5</sup>. This negative result lets us suppose that in general the membership problem is undecidable for non-linear ACGs or that if it is decidable, the proof of decidability must be quite evolved. If we are interested in linear ACGs, we can also show that the emptiness of the sets of the form  $\{M \in \mathcal{O}(\mathcal{G}) \mid \vdash_{\Pi} M : P\}$  is undecidable (for  $\mathcal{G}$  being a linear ACG). But since it is not known whether linear ACGs have a decidable emptiness problem we cannot conclude as we did in the non-linear case. We nevertheless conjecture that the languages of linear ACGs are also not closed under this kind of intersection property.

## 6 Conclusion and perspectives

We have shown that the membership problem for second order non-linear ACGs is decidable. The proof we propose to solve this problem uses the definition of a new tool *higher order intersection signatures* which was designed to extend the notion of regularity to simply typed  $\lambda$ -terms. We will give a precise account of such a notion in a future paper.

Since  $\lambda$ -terms both generalize string and trees, this decidability results can be used for a wide range of formalisms. But, if one limits oneself to strings and trees, this result does not bring anything new when compared to Damm’s decidability result for level- $n$  IO grammar [Dam82]. The real novelty comes from the fact our decidability result also applies to languages of  $\lambda$ -terms. A side-effect of this extension of Damm’s result, is the use of intersection types so as to characterize precise syntactic properties of  $\lambda$ -terms. Such languages may be seen as the set of semantic representations of a Montague-like semantics associated to a context-free formalism. Here we mean context-free formalisms in a broad sense, which encompasses tree adjoining grammars [AKJ75] and multiple context free grammars [SMFK91], that is formalisms whose derivation structures can be represented by local sets of trees. Montague-like semantics are much more powerful than what can be found in the literature. Indeed, for tree adjoining grammars, such an approach has been used in [Pog04] to model elegantly some difficult semantics phenomenon. And, contrary to what it was commonly admitted in the literature, [dG07] shows that it is possible to extend Montague’s ideas so as to implement DRT. This work relies on the modelisation of context of the discourse. This can be achieved in the simply typed  $\lambda$ -calculus as soon as the maximal number of individuals that are stored in memory is fixed since databases can be represented in simply typed  $\lambda$ -calculus [Hil94].

Even though we here give a general answer to the problem of text generation in a Montagovian framework, the algorithm we could extract from the proof is non-elementary, *i.e.* highly intractable. This is not surprising since it is known that even normalisation of simply typed  $\lambda$ -terms is non-elementary [Sta79]. On the other hand, restricted variants of this problem have already been showed to be polynomial as in [Pog01], [Sal05] and [Kan07]. The framework we have settled within this paper should give new ways of finding interesting and tractable restrictions of the text generation problem, and it should also allow to define heuristics to partially solve this problem in the general case.

---

<sup>5</sup> Even though the reduction is simple, it has some technical aspects that would not fit in the exposition of the present paper. Furthermore this reduction presents some points of interest that deserve a development on their own.

## References

- [AKJ75] Masako Takahashi Aravind K. Joshi, Leon S. Levy. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163, 1975.
- [AO94] Takahito Aoto and Hiroakira Ono. Non-uniqueness of normal proofs for minimal formulas in implication-conjunction fragment of bck. *Bulletin of the Section of Logic*, 23(3):104–112, 1994.
- [Aot99] Takahito Aoto. Uniqueness of normal proofs in implicational intuitionistic logic. *Journal of Logic, Language and Information*, 8(2):217–242, 1999.
- [Bar84] Henk P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103. Studies in Logic and the Foundations of Mathematics, North-Holland Amsterdam, 1984. revised edition.
- [BS82] A. A. Babaev and S. V. Soloviev. Coherence theorem for canonical maps in cartesian closed categories. *Journal of Soviet Mathematics*, 20, 1982.
- [Dam82] Werner Damm. The IO- and OI-hierarchies. *Theoretical Computer Science*, 20:95–207, 1982.
- [DCGdL98] Mariangiola Dezani-Ciancaglini, Elio Giovannetti, and Ugo de’ Liguoro. Intersection Types, Lambda-models and Böhm Trees. In *MSJ-Memoir Vol. 2 “Theories of Types and Proofs”*, volume 2, pages 45–97. Mathematical Society of Japan, 1998.
- [dG01] Philippe de Groote. Towards abstract categorial grammars. In Association for Computational Linguistic, editor, *Proceedings 39th Annual Meeting and 10th Conference of the European Chapter*, pages 148–155. Morgan Kaufmann Publishers, 2001.
- [dG07] Philippe de Groote. Towards a montagovian account of dynamics. In *Proceedings of Semantics in Linguistic Theory XVI*. CLC Publications, 2007.
- [GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, 1989.
- [Hil94] Gerd G. Hillebrand. *Finite Model Theory in the Simply Typed Lambda Calculus*. PhD thesis, Department of Computer Science, Brown University, Providence, Rhode Island 02912, May 1994.
- [HT00] Sachio Hirokawa and Makoto Tatsuta. Long D-normal form yields uniqueness of proofs. In *Proceedings of Logic Colloquium*, 2000.
- [Hue76] Gérard Huet. *Résolution d’équations dans des langages d’ordre 1,2,..., $\omega$* . Thèse de doctorat es sciences mathématiques, Université Paris VII, 1976.
- [Kan06] Makoto Kanazawa. Abstract families of abstract categorial languages. *Electr. Notes Theor. Comput. Sci.*, 165:65–80, 2006.
- [Kan07] Makoto Kanazawa. Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 176–183. Association for Computational Linguistics, june 2007.
- [Loa01] Ralph Loader. The undecidability of  $\lambda$ -definability. In C. Anthony Anderson and M. Zeleny, editors, *Logic, Meaning and Computation: Essays in memory of Alonzo Church*, pages 331–342. Kluwer, 2001.
- [Mon74] Richard Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT, 1974.
- [Pog01] Sylvain Pogodalla. *Réseaux de preuve et génération pour les grammaires de types logiques*. PhD thesis, Institut National Polytechnique de Lorraine, 2001.
- [Pog04] Sylvain Pogodalla. Computing semantic representation: Towards ACG abstract terms as derivation trees. In *Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+7)*, pages 64–71, May 2004.
- [Sal05] Sylvain Salvati. *Problèmes de filtrage et problèmes d’analyse pour les grammaires catégorielles abstraites*. PhD thesis, Institut National Polytechnique de Lorraine, 2005.
- [SMFK91] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.
- [Sta79] Richard Statman. The typed lambda-calculus is not elementary recursive. *Theoretical Computer Science*, 9:73–81, 1979.
- [Tat99] Makoto Tatsuta. Uniqueness of d-normal proofs. In *Proceedings of 7th Asian Logic Conference*, pages 41–42, 1999.