Structural properties of intersection types

Emilie Sayag, Michel Mauny

INRIA-Rocquencourt, Projet Cristal
B.P. 105, F-78153 Le Chesnay Cedex, France

Email: {Emilie.Sayag,Michel.Mauny}@inria.fr

Structural properties of intersection types

Emilie Sayag, Michel Mauny*

INRIA-Rocquencourt, Projet Cristal, B.P. 105, F-78153 Le Chesnay Cedex, France

Abstract. We study an intersection type system which is a restriction of the intersection type discipline. This restriction leads to a purely syntactic and completely characterized notion of principal types. Using the equivalence between principal types and normal forms, we define an expansion operation on types which allows us to recover all possible types for any normalizable λ-term. The contribution of this work is a new and simpler definition of the operation of expansion and the description of the structure of principal types.

1 Introduction

In the approach of untyped λ-calculus as a model of programming languages, Curry's type system is the basis of type systems of programming languages like ML [3]. Indeed, Curry's type system has the principal type property *i.e.*, for each typable λ-term there exists a type, the principal type, from which we can find all possible types for this term. Furthermore, typability in this system is decidable. However, this type system has some limitations: polymorphic abstractions are not allowed and types are not preserved under β-conversion. For instance, the λ-term $(\lambda x.x\ x)$ is not typable in this system and the λ-terms $(\lambda x.\lambda y.y)$ and $(\lambda x.\lambda y.\lambda z.x\ z\ (y\ z))(\lambda s.\lambda t.s)$ are β-equivalent, but they have two different principal types.

To supply a type system that does not have these drawbacks, several extensions of Curry's type system have been proposed, the most studied of which being the intersection type discipline. Using intersection types, terms and term variables can have more than one type. This allows polymorphic abstraction, and types are invariant under β-conversion of terms i.e., two λ-terms which are β-equivalent have the same type. Moreover, intersection types characterize normalizable λ-terms: a term is normalizable if and only if it is typable. Intersection type systems are therefore very expressive: this is why many authors have been interested by their theory or their usage.

However, the price of this expressiveness is that type assignment is only semi-decidable. Another drawback is the loss of the principal type property in the classical sense. As a matter of fact, in order to find all possible types of a term from a unique type, we must have more than just substitutions. In [2, 5, 8, 9] a property which is similar to the principal type property is proved by adding new operations on types. Thus, we are convinced that all interesting properties of

* Email: {`Emilie.Sayag,Michel.Mauny`}`@inria.fr`.

intersection type systems have not been highlighted yet. In this paper, in order to fill this gap, we propose a new approach to intersection types.The work presented here is based on the intersection type system introduces in [6]. This type system is a restriction of the one presented in [1] in the sense that intersections occur only in the left hand side of arrow types. In [6], we have defined a new notion of principal type, corresponding exactly to the notion of normal form in the λ-calculus. We now extend this notion to all normalizable λ-terms and using the structural properties of principal types that we proved in [6], we give a simpler definition of the expansion operation than the one proposed in [2, 4, 8, 9], and a simpler proof of the existence of a principal type for all normalizable λ-terms.

The general outline of this paper is as follows: in section 2, we recall the type system of [6] and its main properties. In section 3, we define the operation of expansion and we give some of its properties. The main result of section 4 states the principal type property for normalizable terms and section 5 gives an overview of the related works. Finally, section 6 contains a few concluding remarks.

2 The type theory

For more details about this section, one can see [6]. Here, we only recall the main definitions and properties. The set of types is defined as the following:

$$\begin{array}{lll} \rho \in \mathcal{T} ::= & \alpha & \text{type variable} \\ & \mid [\rho_1, \ldots, \rho_n] \rightarrow \rho & \text{for } n \geq 0 \end{array}$$

We assume a countably infite set TV of type variables.

Definition 1. We define the *positive* and *negative occurrences* of a type variable α in a type ρ by induction on the structure of ρ in the following way:

- if ρ is a type variable, then the possible occurrence of α in ρ is positive
- if $\rho = [\rho_1, \ldots, \rho_n] \rightarrow \rho'$, then the positive (respectively negative) occurrences of α in ρ are the positive (respectively negative) occurrences of α in ρ' and if $n \geq 1$, the negative (respectively positive) occurrences of α in ρ_i for $i = 1, \ldots, n$.

Definition 2. Let ρ be a type in $\mathcal{T}$ and α a type variable. We say that α has a *final occurrence* in ρ if one of the following cases is verified:

- $\rho = \alpha$
- $\rho = [\rho_1, \ldots, \rho_n] \rightarrow \rho'$ and α has a final occurrence in ρ'.

Definition 3. Let $\rho \in \mathcal{T}$, the set $L_0(\rho)$ of *left sub-terms* of ρ is defined by induction on the structure of ρ, in the following way:

- if $\rho = \alpha$, $L(\rho) = \emptyset$
- if $\rho = [\rho_1, \ldots, \rho_n] \rightarrow \rho'$, $L_0(\rho) = \{\rho_1, \ldots, \rho_n\} \cup L_0(\rho')$.

$$\vdash_{s\omega} x : \rho; \{x : [\rho]\} \quad \text{(Var)}$$

$$\frac{\vdash_{s\omega} e_1 : \rho_1; A_1}{\vdash_{s\omega} \lambda x.e_1 : A_1(x) \to \rho_1; A_1 \setminus \{x\}} \quad \text{(Abs)}$$

$$\frac{\vdash_{s\omega} e_1 : [\rho_2^1, \ldots, \rho_2^n] \to \rho_1; A_1 \quad \vdash_{s\omega} e_2 : \rho_2^1; A_2^1 \ldots \vdash_{s\omega} e_2 : \rho_2^n; A_2^n}{\vdash_{s\omega} e_1\ e_2 : \rho_1; A_1 + A_2^1 + \ldots + A_2^n} \ (n \geq 0) \quad \text{(App)}$$

Fig. 1. Inference rules

We also define a mapping *TypeVar* from types to sets of type variables. This function returns the set of type variables which occur in a type.

The operation of substitution is defined as usual: a *substitution* is a mapping from type variables to types, which can be extended in a natural way to a mapping from types to types. The *domain* of a substitution S is the set of type variables which are modified by S. More formally:

$$Dom(S) = \{\alpha \in TV / S(\alpha) \neq \alpha\}$$

We write $[\alpha/\rho]$ the substitution that maps α to ρ and leaves other types variables unchanged. We define the substitution $S_1 + S_2$ where the two substitutions S_1 and S_2 have disjoint domains in the following way:

$$(S_1 + S_2)(\alpha) = \begin{cases} S_i(\alpha) & \text{if } \alpha \in Dom(S_i),\ i = 1 \text{ or } 2 \\ \alpha & \text{if } \alpha \notin Dom(S_1) \cup Dom(S_2) \end{cases}$$

Definition 4. A *constraint environment* A, is a mapping from the set $\mathcal{V}$ of term variables to the multi-sets of types.

We define the *domain* of A, written $Dom(A)$ as: $Dom(A) = \{x \in \mathcal{V} /\ A(x) \neq []\}$. If A is a constraint environment such that $Dom(A) = \{x_1, \ldots, x_n\}$ and $A(x_i) = [\rho_{i1}, \ldots, \rho_{ip_i}]$, for all $i \in \{1, \ldots, n\}$, we use the notation:

$$\{x_1 : [\rho_{11}, \ldots, \rho_{1p_1}], \ldots, x_n : [\rho_{n1}, \ldots, \rho_{np_n}]\}$$

As for all applications, we can restrict the domain of a constraint environment:

$$A \setminus \{x\}(y) = \begin{cases} A(y) & \text{if } y \neq x \\ [] & \text{otherwise} \end{cases}$$

and extend it: $\quad (A_1 + A_2)(x) = A_1(x) \cup A_2(x)$, for all $x \in \mathcal{V}$
where $\cup$ is the union of multi-sets.

Remark. We adopt the usual conventions for omitting parentheses in types and terms and some other syntactic conventions: we use metavariables $x, y, \ldots$ to denote term variables, $\alpha, \beta, \gamma, \ldots$ for type variables, and $A, A_1, \ldots$ for constraint environments.

$Infer(N) =$

- **Case** $N = x$
  - **let** α be a new type variable
  - **return** $(\alpha, \{x : [\alpha]\})$
- **Case** $N = \lambda x.N_1$
  - **let** $(\rho_1, A_1) = Infer(N_1)$
  - **return** $(A_1(x) \rightarrow \rho_1, A_1 \setminus \{x\})$
- **Case** $N = x\ N_1\ \dots\ N_n$
  - **let** $(\rho_1, A_1) = Infer(N_1)$

    $\vdots$

    $(\rho_n, A_n) = Infer(N_n)$

    α be a new type variable
  - **return** $(\alpha, \{x : [[\rho_1] \rightarrow \cdots \rightarrow [\rho_n] \rightarrow \alpha]\} + A_1 + \cdots + A_n)$

Fig. 2. Type inference algorithm

The type assignment relations $\vdash_{sw}$, relating λ-terms, types and constraint environments, are defined in figure 1. We notice that in the rule for applications, if $n = 0$ then there is only one premisse in that inference rule and the argument of the application does not interfere in the derivation. We prove in [7] the following theorem:

Theorem 5. *Let e and e' be λ-terms such that $e =_\beta e'$. $\vdash_{s\omega} e : \rho; A$ if and only if $\vdash_{s\omega} e' : \rho; A$.*

The type inference algorithm is presented in figure 2. This algorithm is *sound* and *complete* [6]. As an example, type inference of $\lambda x.\lambda y.x(y\ x)$ produces the type $([\alpha, [\beta] \rightarrow \gamma] \rightarrow [[\alpha] \rightarrow \beta] \rightarrow \gamma)$.

3 Ground pairs

We now study the structure of ground pairs which are closed under expansions.

We give mutually recursive definitions of $\mathcal{T}_{E_g}$, $\mathcal{T}_g$ and $\mathcal{E}_g$. $\mathcal{T}_g$ is the set of ground types and $\mathcal{E}_g$ is the set of ground constraint environments.

$$\begin{aligned} \nu \in \mathcal{T}_{E_g} ::= &\ \alpha \\ &\mid [\mu_1, \dots, \mu_n] \rightarrow \nu \end{aligned}$$

with $n > 0$, $\forall i \in \{1, \dots, n\}$, $\mu_i \in \mathcal{T}_g$, $TypeVar(\mu_i) \cap TypeVar(\nu) = \emptyset$ and $\forall j \in \{1, \dots, n\}$ such that $j \neq i$, $TypeVar(\mu_i) \cap TypeVar(\mu_j) = \emptyset$

$$\begin{aligned} \mu \in \mathcal{T}_g ::= &\ \alpha \\ &\mid [\nu_1, \dots, \nu_n] \rightarrow \mu \end{aligned}$$

with $n \geq 0$ and $\forall i \in \{1, \dots, n\}$, $\nu_i \in \mathcal{T}_{E_g}$

$$\begin{aligned} A \in \mathcal{E}_g ::= &\ \{\,\} \\ &\mid \{x : [\nu]\} \\ &\mid A_1 + A_2 \end{aligned}$$

From now on, metavariables ν and μ denote elements of $\mathcal{T}_{E_g}$ and $\mathcal{T}_g$ respectively.

In order to link types and constraint environments and to write negatively type constraints, we define *B-types* from ground types and ground constraint environments, in the following way:

$$\begin{aligned} U ::= & [\nu_1, \ldots, \nu_n] \Rightarrow \mu \text{ with } n \geq 0 \\ & \mid [\nu_1, \ldots, \nu_n] \Rightarrow \quad \text{with } n \geq 1 \end{aligned}$$

We can see the types on the left hand side of the double arrow as the other types on the left hand side of arrows. So we extend easily to B-types the notions of sign of an occurrence and left sub-terms.

Definition 6. A B-type U is *closed* if each type variable of $\mathit{TypeVar}(U)$ has exactly one positive occurrence and one negative occurrence in U.

Definition 7. Let $U = [\nu_1, \ldots, \nu_n] \Rightarrow \mu$ be a B-type. U is *finally closed* if the variable α in the final occurrence of μ is also in the final occurrence of a type which is element of $L_0(U)$.

Definition 8. Let U be a B-type. U is *minimally closed* if there is no closed B-type strictly held in U.

The following definition gives a short way to talk about the three previous properties simultaneously.

Definition 9. Let $U = [\nu_1, \ldots, \nu_n] \Rightarrow \mu$ be a B-type. We say that U is *complete* if U is closed, finally closed and minimally closed.

Definition 10. We say that U is a *ground B-type* if U is complete and if it is one of the following forms:

- $U = [\rho] \Rightarrow \rho$ with $\rho \in \mathcal{T}_g \cap \mathcal{T}_{E_g}$.
- $U = [\nu_1, \ldots, \nu_n] \Rightarrow \alpha$ and $\exists i \in \{1, \ldots, n\}$ such that ν_i has the following shape

$$[\mu_1^1, \ldots, \mu_1^{n_1}] \to \cdots \to [\mu_p^1, \ldots, \mu_p^{n_p}] \to \alpha$$

  with $p > 0$, and $\exists (E_j^k)_{j=1\ldots p, k=1\ldots n_j}$, a partition of $[\nu_1, \ldots, \nu_{i-1}, \nu_{i+1}, \ldots, \nu_n]$ such that each $E_j^k \Rightarrow \mu_j^k$ is a ground B-type.
- $U = [\nu_1, \ldots, \nu_n] \Rightarrow [\nu_{n+1}, \ldots, \nu_{n+p}] \to \mu'$ with $[\nu_1, \ldots, \nu_{n+p}] \Rightarrow \mu'$ a ground B-type.

Remark. The partition $(E_j^k)_{j=1\ldots p, k=1\ldots n_j}$ is unique. Since U is closed, each type variable has only two occurrences in U and we have not the choice of the definition of each E_j^k.

In order to simplify the definition of expansion, we need to describe several further notions and prove some properties about the structure of ground B-types. In fact, expansion is a complex operation on pairs. As S. van Bakel explains in [8], the expansion of a sub-term ρ of a type ρ' replaces the occurrences of ρ in

$Clos(\mu, U) =$

- **Case** $U = [\rho] \Rightarrow \rho$

  return $[\rho]$
- **Case** $U = [\nu_1, \ldots, \nu_n] \Rightarrow \alpha$

  let $i \in \{1, \ldots, n\}$ such that

  $\nu_i = [\mu_1^1, \ldots, \mu_1^{n_1}] \rightarrow \cdots \rightarrow [\mu_p^1, \ldots, \mu_p^1, \ldots, \mu_p^{n_p}] \Rightarrow \alpha$

  let $(E_j^k)_{j=1,\ldots,p,k=1,\ldots,n_j}$ the partition of $[\nu_1, \ldots, \nu_{i-1}, \nu_{i+1}, \ldots, \nu_n]$ such that $\forall j \in \{1, \ldots, p\}, \forall k \in \{1, \ldots, n_j\}, E_j^k \Rightarrow \mu_j^k$ is a ground B-type

  if $\exists j, k$ such that $\mu = \mu_j^k$

  then return E_j^k

  else if $\exists j, k$ such that$\mu \in \mathcal{L}(E_j^k \Rightarrow \mu_j^k) \cap \mathcal{T}_{E_g}$

  then return $Clos(\mu, E_j^k \Rightarrow \mu_j^k)$

  else fail
- **Case** $U = [\nu_1, \ldots, \nu_n] \Rightarrow [\nu_{n+1}, \ldots, \nu_{n+m}] \rightarrow \mu'$

  return $Clos(\mu, [\nu_1, \ldots, \nu_{n+m}] \Rightarrow \mu')$

Fig. 3. Closure algorithm

ρ' by a number of copies of that sub-term. To be applied an expansion must therefore specify the type to be expanded and the number of necessary copies.

Intuitively, expansion corresponds to the duplication of the sub-derivation of the argument e_2 in the use of the inference rule (App) for a term $e_1\ e_2$. So it is not enough to duplicate one type: we must also copy all the types of this sub-derivation. Until now, this point was the source of the complexity of the definitions of expansion [2, 5, 4, 8]. Even if the need of duplicating more than one type is well understood, the definition of the set of types to be copied, is still a difficult problem. So far, no convincing justification has been given.

The contribution of this section is precisely the definition of this problematic set of types. The justification of this definition is obvious according to the previous results about the structure of principal and ground types.

The notion of left sub-terms does not take into account the full recursive structure of a type. We now define a notion of *generalized left sub-terms*, following the recursive structure of types to consider all possible sub-terms which are to the left of an arrow at any level in the recursive structure of a type.

Definition 11. Let U be a B-type, we define the set $\mathcal{L}(U)$ of *generalized left sub-terms* of U in the following way:

- $L_0(U)$ is defined as for A-types
- $\forall n > 0$, $L_n(U) = \bigcup_{\rho \in L_{n-1}(U)} L_0(\rho)$
- $\mathcal{L}(U) = \bigcup_{n \geq 0} L_n(U)$

We define in figure 3 an algorithm constructing the multi-set of types that we must duplicate when we expand a type.

Lemma 12. *Let U be a ground B-type and $\mu \in \mathcal{L}(U) \cap \mathcal{T}_g$. $Clos(\mu, U)$ is well-defined and verifies the following conditions:*

- *$Clos(\mu, U) \subset \mathcal{L}(U) \cap \mathcal{T}_{E_g}$*
- *$Clos(\mu, U) \Rightarrow \mu$ is a ground B-type*
- *$Clos(\mu, U)$ is the unique sub-multi-set of $\mathcal{L}(U) \cap \mathcal{T}_{E_g}$ which verifies the previous condition.*

The next definition is just a syntactic facility.

Definition 13. Let U be a ground B-type and $\mu \in \mathcal{L}(U) \cap \mathcal{T}_g$ a type, the ground B-type $Clos(\mu, U) \Rightarrow \mu$ is called the *closure* of μ in U.

An expansion makes a number of copies of several types. We want each copy of a type to be disjoint from all others, *i.e.* two copies of the same type have no common type variables. In order to be precise, we define specific substitutions which will make the copies of types exactly as we need.

Definition 14. Let S be a substitution, we say that S is a *renaming substitution* if for all $\alpha \in Dom(S)$, $S(\alpha) = \beta$ where β is a type variable and S is injective. Furthermore, if $Range(S)$ is a set of new type variables, we say that S is a *fresh renaming substitution.*

Thus any renaming substitution S has an inverse, written S^{-1} which is also a renaming substitution, but even if S is a fresh renaming substitution, S^{-1} is not a fresh renaming substitution since type variables in $Range(S^{-1}) = Dom(S)$ are not fresh.

Definition 15. Let p be an integer. For all types μ in $\mathcal{T}_g$ we define an operation of *expansion* of μ, on the ground B-type U, written $E_{(p,\mu)}$, by:

$$E_{(p,\mu)}(U) = \begin{cases} U & \text{if } \mu \notin \mathcal{L}(U) \\ U' & \text{otherwise} \end{cases}$$

where U' is obtained from U by replacing each occurrence of an element ν of $Clos(\mu, U)$ by $R_1(\nu), \ldots, R_p(\nu)$ and μ by $R_1(\mu), \ldots, R_p(\mu)$, if $R_1, \ldots, R_p$ are p fresh renaming substitutions of domain $TypeVar(Clos(\mu, U))$.

We remark that since the renaming substitutions $R_1, \ldots, R_p$ are not unique, the expansion of μ in U is defined up to a renaming. Moreover, we make no hypothesis on p. If $p = 0$, the expansion $E_{(p,\mu)}$ removes all the occurrences of μ and of the elements of its closure.

Since our work is essentially based on the structure of types, we want to prove that expansions do not change this structure. So we prove that the set of ground pairs is closed under expansion.

Lemma 16. *Let U be a ground B-type, $\mu \in \mathcal{T}_g$, and p an integer. Then $E_{(p,\mu)}(U)$ is a ground B-type.*

4 Principal typing of normalizable λ-terms

This section states the existence of principal types for all normalizable λ-terms in corollary 19. The interest of this section is not the result itself, but its proof which is conceptually much simpler than the proofs in [2, 5, 8, 9]. Here we are only interested in normalizable λ-terms. Thus, thanks to theorem 5, it is enough to use normal forms. We do not need to introduce approximants which significantly simplifies the proofs.

The substitution and expansion operations are both necessary to find a possible pair for a normal form from its principal pair. However these operations must be applied in an appropriate order.

Definition 17. We name *chain* a composition of substitutions, expansions and renaming substitutions, of the form $S_n \circ \ldots \circ S_1 \circ O_m \circ \ldots \circ O_1$ where S_i is a substitution for $i = 1, \ldots, n$ and O_j is either a renaming substitution or an expansion for $j = 1, \ldots, m$.

Theorem 18. *Let N be a term in normal form such that $\vdash_{s\omega} N : \mu; A$. If $Infer(N) = (\mu_p, A_p)$ then there exists a chain C such that $C(\overline{A_p} \Rightarrow \mu_p) = \overline{A} \Rightarrow \mu$.*

Corollary 19. *Let e be a normalizable term such that $\vdash_{s\omega} e : \mu; A$, N its normal form and $(\mu_p, A_p) = Infer(N)$. Then there exists a chain C such that $C(\overline{A_p} \Rightarrow \mu_p) = \overline{A} \Rightarrow \mu$.*

5 Related work

The authors of [2, 5, 8, 9] introduce a notion of *approximants*, also named λ-*Ω-normal forms*, and define principal typing for these extended normal forms before generalizing to λ-terms using an approximation property, i.e. $B \vdash e : \mu$ if and only if there exists an approximant a of e such that $B \vdash a : \mu$. S. Ronchi della Rocca proposed a semi-algorithm for type inference in [4]. These results give important theoretical benefits, but unfortunately, they provide a good understanding neither of the structure of principal types nor of their characteristic properties. Furthermore, the semi-algorithm proposed in [4] is not practical because of its conceptual complexity.

As far as we know, the work of S. van Bakel [8, 9] is the first real advance in the simplication of the presentation of the intersection type discipline since the initial presentations. Furthermore, in [8], S. van Bakel defines an intersection type system close to the one introduced in [1] with the same partial order relation on types. He studies the existence of principal types for this system. He was induced to define several sub-sets of the set of pairs of a type and a basis, ordered by inclusion. The smallest is the *set of principal pairs* which corresponds to set $\mathcal{P}$ in our work. Van Bakel's *set of ground pairs* is equivalent to the set of ground pairs that we define. It is the sub-set of pairs closed under expansion. Because of the partial order relation, van Bakel uses an extra sub-set: the *set of primitive pairs*, closed under *lifting*. (This operation is necessary to take into account the inference rule that deals with the partial order relation.) Finally, his set of pairs is closed under substitution. Here we only distinguish the set of principal pairs and ground pairs. The *relevant* intersection type system, presented in [9], is pretty closed to the one presented here, the main difference being that intersections *are not* types in our system, which results in simpler operations on types. Despite of this, we obtain the same final result: *all normalizable λ-terms have a principal type.*

6 Conclusion

In this article, we take advantage of the structural properties of intersection types providing simple proofs of the principal type property for normalizable λ-terms.

According to the structure of principal types, we define ground types which correspond to S. van Bakel's ground types. But, as we did for principal types, we have studied in detail the structure of these ground types. These structural properties lead us to a simple definition of the expansion operation. Then we prove that in our intersection type system, expansions and substitutions are enough to find all possible types of a normalizable λ-term from a unique type called its principal type.

These results shed a new light on intersection typing , which can be presented through principal types isomorphic to λ-terms in normal form. This presentation of intersection typing can serve as a basis for a better understanding of the relationship between intersection typing of arbitrary λ-terms and β-réduction, which in turn could lead to a more operational presentation of intersection typing. *In fine*, such a presentation could be used as a general framework for solving the problem of dynamic type reconstruction for polymorphic programming languages.

References

1. Henk P. Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic*, 48(4):931–940, 1983.
2. Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Principal type schemes and λ-calculus semantics. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda-calculus and Formalism*, pages 536–560. Academic Press, London, 1980.
3. Robin Milner. A theory of type polymorphism in programming. *Computer and System Sciences*, 17(3):348–375, 1978.
4. Simona Ronchi Della Rocca. Principal type scheme and unification for intersection type discipline. *Theoretical Computer Science*, 59:181–209, 1988.
5. Simona Ronchi Della Rocca and Betti Venneri. Principal type schemes for an extended type theory. *Theoretical Computer Science*, 28:151–169, 1984.

6. Émilie Sayag and Michel Mauny. Characterization of principal types of normal forms in an intersection type system. In V. Chandru and V. Vinay, editors, *Proceedings of Sixteenth Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1180 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.
7. Émilie Sayag and Michel Mauny. A new presentation of the intersection type discipline through principal typings of normal forms. Technical Report RR-2998, INRIA, 1996. available on our web site http://pauillac.inria.fr/~ sayag/rr.html.
8. Steffen van Bakel. Principal type schemes for strict type assignment system. *Logic and Computation*, 3(6):643–670, 1993.
9. Steffen van Bakel. Intersection type assignment systems. *Theoretical Computer Science*, 151:385–435, 1995.

Annex

Ground pairs

The next lemma precises the structure of ρ in a B-type $U = [\rho]\Rightarrow\rho$.

Lemma . *Let $U = [\rho]\Rightarrow\rho$ be a ground B-type and ρ' a sub-term of U then ρ' has two occurrences in U, one such that $\rho' \in \mathcal{T}_g$ and one such that $\rho' \in \mathcal{T}_{E_g}$.*

Proof. by case on ρ'.

• If $\rho' = \rho$ then ρ verifies the property by definition of a B-type.

• Otherwise $\rho = [\rho_1, \ldots, \rho_n] \rightarrow \rho''$ where by definition of a ground B-type, ρ_i has no common type variable neither with the other ρ_j nor with ρ''. ρ' is a sub-term of $\rho_1, \ldots, \rho_n$ or ρ. Moreover, since $\rho \in \mathcal{T}_g \cap \mathcal{T}_{E_g}$, we have for $i = 1, \ldots, n$, $\rho_i \in \mathcal{T}_g \cap \mathcal{T}_{E_g}$ and $\rho'' \in \mathcal{T}_g \cap \mathcal{T}_{E_g}$ and each of them has two occurrences in U. If we consider ρ as an element of $\mathcal{T}_{E_g}$ (respectively $\mathcal{T}_g$), $\rho_1, \ldots, \rho_n$ are elements of $\mathcal{T}_g$ (respectively $\mathcal{T}_{E_g}$) and ρ'' of $\mathcal{T}_{E_g}$ (respectively $\mathcal{T}_g$). So ρ' has two occurrences in U such that one is element of $\mathcal{T}_{E_g}$ and the other of $\mathcal{T}_g$.

Lemma 12. *Let U be a ground B-type and $\mu \in \mathcal{L}(U) \cap \mathcal{T}_g$. $Clos(\mu, U)$ is well-defined and verifies the following conditions:*

- *$Clos(\mu, U) \subset \mathcal{L}(U) \cap \mathcal{T}_{E_g}$*
- *$Clos(\mu, U)\Rightarrow\mu$ is a ground B-type*
- *$Clos(\mu, U)$ is the unique sub-multi-set of $\mathcal{L}(U) \cap \mathcal{T}_{E_g}$ which verifies the previous condition.*

Proof. by induction on the structure of U.

• If $U = [\rho]\Rightarrow\rho$ then we notice that $\mu \neq \rho$, otherwise $\mu \notin \mathcal{L}(U) \cap \mathcal{T}_g$. By the lemma 6, we know that every sub-term of ρ has two occurrences, one belonging to $\mathcal{T}_{E_g}$ and the other belonging to $\mathcal{T}_g$. Thus, there exists an occurrence of μ in U which belongs to $\mathcal{L}(U) \cap \mathcal{T}_{E_g}$. Moreover $[\mu]\Rightarrow\mu$ is a ground B-type since $\mu \in \mathcal{T}_{E_g} \cap \mathcal{T}_g$ and $[\mu]$ is the unique sub-multi-set of $\mathcal{L}(U) \cap \mathcal{T}_{E_g}$ since μ as only two occurrences in U.

• If $U = [\nu_1, \ldots, \nu_n]\Rightarrow\alpha$ then we distinguish several cases:

- if $\exists(j,k)/\mu = \mu_j^k$, then by definition of a ground B-type E_j^k is well defined and unique. Moreover, $E_j^k \subset \mathcal{L}(U) \cap \mathcal{T}_{E_g}$ and $E_j^k \Rightarrow \mu$ is a ground B-type.
- if $\exists(j,k)/\mu \in \mathcal{L}(E_j^k \Rightarrow \mu_j^k) \cap \mathcal{T}_{E_g}$ then by the induction hypothesis on $E_j^k \Rightarrow \mu_j^k$, $Clos(\mu, E_j^k \Rightarrow \mu_j^k)$ is well defined and verifies the conditions of the lemma. Since $\mathcal{L}(E_j^k \Rightarrow \mu_j^k) \subset \mathcal{L}(U)$, we only must show that $Clos(\mu, E_j^k \Rightarrow \mu_j^k)$ is the unique sub-multi-set of $\mathcal{L}(U) \cap \mathcal{T}_{E_g}$. Since each $E_j^k \Rightarrow \mu_j^k$ is a ground B-type, therefore closed, they are disjoint from each other and we deduce the unicity.
- otherwise it is impossible since we suppose that $\mu \in \mathcal{L}(U) \cap \mathcal{T}_g$.

• If $U = [\nu_1, \ldots, \nu_n] \Rightarrow [\nu_{n+1}, \ldots, \nu_{n+m}] \to \mu'$ then the induction hypothesis gives the result.

Definition. We say that two types ρ_1 et ρ_2 *have the same structure* in one of the following cases:

- if $\rho_1 = \alpha$ and $\rho_2 = \beta$
- if $\rho_1 = [\mu_1, \ldots, \mu_n] \to \nu$, $\rho_2 = [\mu'_1, \ldots, \mu'_m] \to \nu'$ and for all $j \in \{1, \ldots, m\}$, there exists $i_j \in \{1, \ldots, n\}$ such that μ_{i_j} and μ_j have the same structure, and if ν and ν' have the same structure.
- if ρ_1 and ρ_2 belong to $\mathcal{T}_g$ and are such that $\rho_1 = [\nu_1, \ldots, \nu_n] \to \mu$ and $\rho_2 = [\nu'_1, \ldots, \nu'_n] \to \mu'$ for all $i \in \{1, \ldots, n\}$, ν_i and ν'_i have the same structure and μ and μ' have the same structure.

Remark. Let ρ be a type, $\mu \in \mathcal{T}_g \cap \mathcal{L}(\rho)$ and p an integer. If we replace in ρ the occurrence of μ by $R_1(\mu), \ldots, R_p(\mu)$ to obtain ρ', where $R_1, \ldots, R_p$ are fresh renaming substitutions then ρ' have the same structure as ρ. The proof by induction on the structure of ρ is immediate.

As usual, we extend the notion of having the same structure to B-types without difficulty. We say that two B-types U_1 and U_2 have the same structure if one of the following cases is verified:

- $U_1 = [\rho_1] \Rightarrow \rho_1$, $U_2 = [\rho_2] \Rightarrow \rho_2$ and ρ_1 and ρ_2 have the same structure.
- $U_1 = [\nu_1, \ldots, \nu_n] \Rightarrow \alpha$, $U_2 = [\nu'_1, \ldots, \nu'_m] \Rightarrow \beta$ and for all $j \in \{1, \ldots, m\}$, there exists $i_j \in \{1, \ldots, n\}$ such that ν_{i_j} and ν'_j have the same structure.
- $U_1 = [\nu_1, \ldots, \nu_n] \Rightarrow [\nu_{n+1}, \ldots, \nu_{n+p}] \to \mu$, $U_2 = [\nu'_1, \ldots, \nu'_m] \Rightarrow [\nu'_{m+1}, \ldots, \nu'_{m+q}] \to \mu'$, μ and μ' have the same structure and for all $j \in \{1, \ldots, m\}$ (respectively $\{m+1, \ldots, m+q\}$), there exists $i_j \in \{1, \ldots, n\}$ (respectively $\{n+1, \ldots, n+p\}$) such that ν_{i_j} and ν'_j have the same structure.

We need to prove a lemma stronger than the lemma 16.

Lemma. *Let U be a ground B-type, $\mu' \in \mathcal{T}_g$ and p an integer. $E_{(p,\mu')}(U)$ is a ground B-type which has the same structure as U.*

Proof. first by cases on μ' then by induction on the structure of U.

• If $\mu' \notin \mathcal{L}(U)$ then $E_{(p,\mu')}(U) = U$ is a ground B-type which has the same structure as U.

• Otherwise we reason by induction on the structure of U and three cases can arise:

- if $U = [\rho]\Rightarrow\rho$ then μ' is a sub-term of ρ and $Clos(\mu', U) = [\mu']$. So we have $E_{(p,\mu')}(U) = [\rho']\Rightarrow\rho'$ where we have replaced μ' in ρ by the p copies of μ': $R_1(\mu'), \dots R_p(\mu')$, to obtain ρ'. Thus according to the previous remark, μ and μ' have the same structure. We can deduce that U and $E_{(p,\mu')}(U)$ have also the same structure.
- if $U = [\nu_1, \dots, \nu_n]\Rightarrow\alpha$ then since U is a ground B-type, there exists $i \in \{1, \dots, n\}$ such that $\nu_i = [\mu_1^1, \dots, \mu_1^{n_1}] \to \dots \to [\mu_m^1, \dots, \mu_m^{n_m}] \to \alpha$ and there exists a partition $(E_j^k)_{j=1\dots m, k=1,\dots n_j}$ of $[\nu_1, \dots, \nu_{i-1}, \nu_{i+1}, \dots, \nu_n]$ such that for all $j \in \{1, \dots, m\}$ and all $k \in \{1, \dots, n_j\}$, $E_j^k\Rightarrow\mu_j^k$ is a ground B-type.
  - if there exist j and k such that $\mu' = \mu_j^k$ then $Clos(\mu', U) = E_j^k$ and so by definition of an expansion, we have:
    $E_{(p,\mu')}(U) =$
    $E_1^1 \cup \dots \cup E_1^{n_1} \cup \dots \cup E_j^1 \cup \dots \cup E_j^{k-1} \cup E_j^{k_1} \cup \dots \cup E_j^{k_p} \cup$
    $E_j^{k+1} \cup \dots \cup E_j^{n_j} \cup \dots \cup E_p^1 \cup \dots \cup E_p^{n_p} \cup$
    $[[\mu_1^1, \dots, \mu_1^{n_1}] \to \dots \to [\mu_j^1, \dots, \mu_j^{k-1}, \mu_j^{k_1}, \dots, \mu_j^{k_p}, \mu_j^{k+1} \dots, \mu_j^{n_j}]$
    $\to \dots \to [\mu_m^1, \dots, \mu_m^{n_m}] \to \alpha]\Rightarrow\alpha$
    where for all $l \in \{1, \dots, p\}$, $E_j^{k_l} = R_l(E_j^k)$ and $\mu_j^{k_l} = R_l(\mu_j^k)$. Then $R_l(E_j^k\Rightarrow\mu_j^k)$ is a ground B-type, since $E_j^k\Rightarrow\mu_j^k$ is a ground B-type and a renaming substitution does not change the structure of a B-type. We deduce that $E_{(p,\mu')}(U)$ is a ground B-type which has the same structure as U.
  - otherwise there exist j and k such that $\mu' \in \mathcal{L}(E_j^k\Rightarrow\mu_j^k) \cap \mathcal{T}_g$. We write $U_j^k = E_j^k\Rightarrow\mu_j^k$. Since U_j^k is a ground B-type, we can apply the expansion $E_{(p,\mu')}$. Let $U_j^{k'} = E_j^{k'}\Rightarrow\mu_j^{k'}$ be the result of this application. By induction, $U_j^{k'}$ is a ground B-type which has the same structure as U_j^k. So by definition of an expansion, we have:
    $E_{(p,\mu')}(U) =$
    $E_1^1 \cup \dots \cup E_1^{n_1} \cup \dots \cup E_j^1 \cup \dots \cup E_j^{k-1} \cup E_j^{k'} \cup$
    $E_j^{k+1} \cup \dots \cup E_j^{n_j} \cup \dots \cup E_p^1 \cup \dots \cup E_p^{n_p} \cup$
    $[[\mu_1^1, \dots, \mu_1^{n_1}] \to \dots \to [\mu_j^1, \dots, \mu_j^{k-1}, \mu_j^{k'}, \mu_j^{k+1}, \dots, \mu_j^{n_j}] \to \dots \to$
    $[\mu_m^1, \dots, \mu_m^{n_m}]]\Rightarrow\alpha$
    which is a ground B-type with the same structure as U.
- if $U = [\nu_1, \dots, \nu_n]\Rightarrow[\nu_{n+1}, \dots, \nu_{n+m}] \to \mu$ then let $U' = [\nu_1, \dots, \nu_{n+m}]\Rightarrow\mu$. By definition of a ground B-type, U' is immediately a ground B-type. So we can apply the expansion $E_{(p,\mu')}$. By induction, the result of this application is a ground B-type with the same structure as U'.

Thus $E_{(p,\mu')}(U') = [\nu'_1, \ldots, \nu'_q]\Rightarrow\mu'$ where μ and μ' have the same structure and for all $j \in \{1, \ldots, q\}$, there exists $i_j \in \{1, \ldots, n+m\}$ such that ν_{i_j} and ν_j have the same structure.
Let $[\nu'_{j_1}, \ldots, \nu'_{j_{q-r}}]$ and $[\nu'_{j_{q-r+1}}, \ldots, \nu'_{j_q}]$ be two sub-multi-sets of $[\nu'_1, \ldots, \nu'_q]$ constituted by the types having the same structure as the types of $[\nu_1, \ldots, \nu_n]$ and $[\nu_{n+1}, \ldots, \nu_{n+m}]$ respectively. Then $E_{(p,\mu')}(U) = [\nu'_{j_1}, \ldots, \nu'_{j_{q-r}}]\Rightarrow [\nu'_{j_{q-r+1}}, \ldots, \nu'_{j_q}] \rightarrow \mu'$ and $E_{(p,\mu')}(U)$ is a ground B-type with the same structure as U.

The next lemma specifies some results about expansions' behavior according to the structure of B-types to which they are applied.

Lemma 20. *Let E be an expansion.*

- *If $E(\overline{A}\Rightarrow\mu) = \overline{A'}\Rightarrow\mu'$, then $E(\overline{A \setminus \{x\}}\Rightarrow A(x) \rightarrow \mu) = \overline{A' \setminus \{x\}}\Rightarrow A'(x) \rightarrow \mu'$*
- *If for $i = 1, \ldots, n$, $E(E_i\Rightarrow\mu_i) = E'_i\Rightarrow\mu'_i$, then $E(E_1 + \cdots + E_n + [[\mu_1] \rightarrow \cdots \rightarrow [\mu_n] \rightarrow \alpha]\Rightarrow\alpha) = E'_1 + \cdots + E'_n + [[\mu'_1] \rightarrow \cdots \rightarrow [\mu'_n] \rightarrow \alpha]\Rightarrow\alpha$*

Proof. By definition of expansion and application of lemma 16.

Principal typing of normalizable λ-terms

From now on, we will not need to distinguish the different S_i (respectively O_j) from each other. So it will be enough to write a chain C simply by $\mathcal{C} \circ \mathcal{O}$ where $\mathcal{C}$ will be a composition of substitutions and $\mathcal{O}$ a composition of renaming substitutions and expansions.

Theorem 18. *Let N be a term in normal form such that $\vdash_{s\omega} N : \mu; A$. If Infer$(N) = (\mu_p, A_p)$ then there exists a chain C such that $C(\overline{A_p}\Rightarrow\mu_p) = \overline{A}\Rightarrow\mu$.*

Proof. by induction on the structure of N.

• If $N = x$ then we have derived $\vdash_{s\omega} x : \mu; \{x : [\mu]\}$ and *Infer*$(N) = (\alpha, \{x : [\alpha]\})$ where α is fresh type variable. If we define C by $C = [\alpha/\mu]$, we have $C([\alpha]\Rightarrow\alpha) = [\mu]\Rightarrow\mu$.

• If $N = \lambda x.N_1$ then we have derived:

$$\frac{\vdash_{s\omega} N_1 : \mu_1; A_1}{\vdash_{s\omega} \lambda x.N_1 : A_1(x) \rightarrow \mu_1; A_1 \setminus \{x\}}$$

with $\mu = A_1(x) \rightarrow \mu_1$ and $A = A_1 \setminus \{x\}$.

On the other hand, if *Infer*$(N_1) = (\mu_{1p}, A_{1p})$ then *Infer*$(\lambda x.N_1) = (A_{1p}(x) \rightarrow \mu_{1p}, A_{1p} \setminus \{x\})$ with $\mu_p = A_{1p}(x) \rightarrow \mu_{1p}$ and $A_p = A_{1p} \setminus \{x\}$.

By the induction hypothesis, there exists a chain C such that $C(\overline{A_{1p}}\Rightarrow\mu_{1p}) = \overline{A_1}\Rightarrow\mu_1$. Moreover, $\overline{A_{1p}} = \overline{A_{1p} \setminus \{x\}} \cup A_{1p}(x)$ and a chain respects the structure of B-types (cf. lemma 20). So we have:

$$C(\overline{A_{1p} \setminus \{x\}}\Rightarrow A_{1p}(x) \rightarrow \mu_{1p}) = \overline{A_1 \setminus \{x\}}\Rightarrow A_1(x) \rightarrow \mu_1$$

i.e., $C(\overline{A_p \Rightarrow} \rightarrow \mu_p) = \overline{A} \Rightarrow \mu$.

- If $N = x\ N_1 \dots N_n$ then we have derived:

$$\dfrac{\dfrac{\vdash_{s\omega} x : \mu_1; A \qquad \vdash_{s\omega} N_1 : \mu_1^j; A_1^j}{\vdash_{s\omega} x\ N_1 : \mu_2; B_1} \qquad \ddots \qquad \ddots}{} \quad \dfrac{\vdash_{s\omega} x\ N_1 \dots N_{n-1} : \mu_n; B_{n-1} \qquad \vdash_{s\omega} N_n : \mu_n^j; A_n^j}{\vdash_{s\omega} x\ N_1 \dots N_n : \mu; B_n}$$

where for $k = 1, \dots, n$, $\mu_k = [\mu_k^1, \dots, \mu_k^{m_k}] \rightarrow \dots \rightarrow [\mu_n^1, \dots, \mu_n^{m_n}] \rightarrow \mu$ and $B_k = A + A_1^1 + \dots + A_1^{m_1} + \dots + A_k^1 + \dots + A_k^{m_k}$, and $A = \{x : [[\mu_1^1, \dots, \mu_1^{m_1}] \rightarrow \dots \rightarrow [\mu_n^1, \dots, \mu_n^{m_n}] \rightarrow \mu]\}$. Notice that we can have $m_i = 0$.

On the other hand, if α is a fresh type variable and if for $i = 1, \dots, n$, $\mathit{Infer}(N_i) = (\mu_{ip}, A_{ip})$, then

$$\mathit{Infer}(x\ N_1 \dots N_n) = (\alpha, A_p + A_{1p} + \dots + A_{np})$$

where $A_p = \{x : [[\mu_{1p}] \rightarrow \dots \rightarrow [\mu_{np}] \rightarrow \alpha]\}$.

Moreover, from the induction hypothesis, we deduce that for all $i \in \{1, \dots, n\}$, there exist m_i chains $C_i^1, \dots, C_i^{m_i}$ such that $\forall j \in \{1, \dots, m_i\}$, $C_i^j(\overline{A_{ip}} \Rightarrow \mu_{ip}) = \overline{A_i^j} \Rightarrow \mu_i^j$.

We write for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, m_i\}$, $C_i^j = \mathcal{S}_i^j \circ \mathcal{O}_i^j$ where $\mathcal{S}_i^j$ names a composition of substitutions and $\mathcal{O}_i^j$ a composition of expansions and renaming substitutions and $\overline{A_i^{j'}} \Rightarrow \mu_i^{j'} = \mathcal{O}_i^j(\overline{A_{ip}} \Rightarrow \mu_{ip})$.

Let $E_{(m_i, \mu_{ip})}$ be an expansion for all $i \in \{1, \dots, n\}$ and $R_i^1, \dots, R_i^{m_i}$ the associated renaming substitutions. If we define the chain C by:

$$\begin{array}{l} C = \\ [\alpha/\mu] o \mathcal{S}_n^{n_m} \circ \dots \circ \mathcal{S}_n^1 \circ \dots \circ \mathcal{S}_1^{m_1} \circ \dots \circ \mathcal{S}_1^1 \circ \\ \mathcal{O}_n^{m_n} \circ (R_n^{m_n})^{-1} \circ \dots \circ \mathcal{O}_n^1 \circ (R_n^1)^{-1} \circ \dots \circ \mathcal{O}_1^{m_1} \circ (R_1^{m_1})^{-1} \circ \dots \circ \mathcal{O}_1^1 \circ (R_1^1)^{-1} \circ \\ E_{(m_n, \mu_{np})} \circ \dots \circ E_{(m_1, \mu_{1p})} \end{array}$$

then $C(\overline{A_p + A_{1p} + \dots + A_{np}} \Rightarrow \alpha) = \overline{B_n} \Rightarrow \mu$. If $m_i = 0$, we keep $E_{(m_i, \mu_{ip})}$ in C but we remove $\mathcal{O}_i^{m_i} \circ (R_i^{m_i})^{-1} \circ \dots \circ \mathcal{O}_i^1 \circ (R_i^1)^{-1}$ and the previous equality is still true.

In fact, we have:

$$\begin{array}{rl} U' = E_{(m_n, \mu_{np})} \circ \dots \circ E_{(m_1, \mu_{1p})}(\overline{A_p + A_{1p} + \dots + A_{np}} \Rightarrow \alpha) = & \\ \quad \overline{[[R_1^1(\mu_{1p}), \dots, R_1^{m_1}(\mu_{1p})] \rightarrow \dots \rightarrow [R_n^1(\mu_{np}), \dots, R_n^{m_n}(\mu_{np})] \rightarrow \alpha]} + & \\ \quad \overline{R_1^1(A_{1p})} + \dots + \overline{R_1^{m_1}(A_{1p})} + \dots + \overline{R_n^1(A_{np})} + \dots + \overline{R_n^{m_n}(A_{np})} \Rightarrow \alpha & \end{array}$$

Then for all $i \in \{1, \ldots, n\}$ and all $j \in \{1, \ldots, m_i\}$,

$$\mathcal{O}_i^j \circ (R_i^j)^{-1}(\overline{R_i^j(A_{ip})} \Rightarrow R_i^j(\mu_{ip})) = \overline{A_i^{j'}} \Rightarrow \mu_i^{j'}$$

so we have:

$$\begin{aligned} U'' = \mathcal{O}_n^{m_n} \circ (R_n^{m_n})^{-1} \circ \ldots \circ \mathcal{O}_n^1 \circ (R_n^1)^{-1} \circ \ldots \circ \\ \mathcal{O}_1^{m_1} \circ (R_1^{m_1})^{-1} \circ \ldots \circ \mathcal{O}_1^1 \circ (R_1^1)^{-1}(U') = \\ \overline{[[\mu_1^{1'}, \ldots, \mu_1^{m'_1}] \to \cdots \to [\mu_n^{1'}, \ldots, \mu_n^{m'_n}] \to \alpha] +} \\ \overline{A_1^{1'} + \cdots + A_1^{m'_1} + \cdots + A_n^{1'} + \cdots + A_n^{m'_n}} \Rightarrow \alpha \end{aligned}$$

and since for all $i \in \{1, \ldots, n\}$ and all $j \in \{1, \ldots, m_i\}$, $S_i^j(\mu_i^{j'}) = \mu_i^j$ and $S_i^j(\overline{A_i^{j'}}) = \overline{A_i^j}$, we have:

$$\begin{aligned} U''' = \mathcal{S}_n^{n_m} \circ \ldots \circ \mathcal{S}_n^1 \circ \ldots \circ \mathcal{S}_1^{m_1} \circ \ldots \circ \mathcal{S}_1^1(U'') = \\ \overline{[[\mu_1^1, \ldots, \mu_1^{m_1}] \to \cdots \to [\mu_n^1, \ldots, \mu_n^{m_n}] \to \alpha] +} \\ \overline{A_1^1 + \cdots + A_1^{m_1} + \cdots + A_n^1 + \cdots + A_n^{m_n}} \Rightarrow \alpha \end{aligned}$$

and finally:

$$\begin{aligned} [\alpha/\mu](U''') = \overline{[[\mu_1^1, \ldots, \mu_1^{m_1}] \to \cdots \to [\mu_n^1, \ldots, \mu_n^{m_n}] \to \mu] +} \\ \overline{A_1^1 + \cdots + A_1^{m_1} + \cdots + A_n^1 + \cdots + A_n^{m_n}} \Rightarrow \mu \end{aligned}$$

So $[\alpha/\mu](U''') = \overline{B_n} \Rightarrow \mu$.

Corollary 19. *Let e be a normalizable term such that $\vdash_{s\omega} e : \mu; A$, N its normal form and $(\mu_p, A_p) = Infer(N)$. Then there exists a chain C such that $C(\overline{A_p} \Rightarrow \mu_p) = \overline{A} \Rightarrow \mu$.*

Proof. Since $e =_\beta N$ by theorem 5 we have $\vdash_{s\omega} N : \mu; A$ and we deduce the result directly from theorem 18.

This article was processed using the LaTeX macro package with LLNCS style