Journal of Computing and Information Technology - CIT 5, 1997, 2, 129–138

Principal Typing for Parallel and non-Deterministic λ-calculus

Ali S. Aoun[1], Franco Barbanera[2], Mariangiola Dezani–Ciancaglini[2] and Şeref Mirasyedioğlu[3]

[1]Faculty of Science, Mathematics Department Hacettepe University, Ankara, Turkey
[2]Dipartimento di Informatica, Universitá di Torino, Torino, Italy
[3]Faculty of Education, Mathematics Department, Beşevler Ankara, Turkey

Parallelism and non-determinism are fundamental concepts in the process algebra theory. Combining them with λ-calculus can enlighten the theory of *higher-order* process algebras. In recent papers an analysis of a λ-calculus containing parallel and non-deterministic operators was carried on by means of a type assignment system with intersection and union types. The present paper answers the problem of determining principal types for this system.

1. Introduction

Type assignment systems are formal systems to derive types (if any) for untyped terms. In the context of programming languages theory, having a type assignment system enables one to check the type (partial specification) of a program *after* its development. This way of dealing with types has several advantages in practice and it is used in powerful functional programming languages, notably ML and MIRANDA. A problem which is worth addressing when we deal with type assignment systems, is definitely that of the principal typing, i.e. the problem of looking for a type (the principal type-scheme), if any, from which all the types that can be inferred for a given term can be derived by means of suitable operations. Principal typings lie at the core of any practical type-synthesis semi-algorithm.

For Curry's basic functionality theory for λ-calculus [9] this problem was addressed by Hindley in [14]. For this system there exists a procedure to find the principal type scheme of a term, if any, and derive all its types by means of a single operation, namely substitution.

In [5] Curry's type assignment system was extended by adding a constant ω as a universal type and a new constructor '$\wedge$' for the intersection of two types. Type assignment systems with intersection types have been widely investigated in literature, since they enable to precisely characterize relevant syntactical property of the λ-calculus, as well as many of its models [5, 7]. In systems with intersection types, the presence of the $\wedge$ type constructor makes the principal typing problem much more difficult to solve than in the basic functionality theory. In [8, 18, 3] the principal type scheme problem for systems with intersection types was solved making an essential use of the notion of approximant of a term. There are three operations devised to derive all possible types from the principal type scheme, that is substitution, expansion and lifting.

In [15] a theory with both intersection and polymorphic types was introduced with the aim of widening the investigation of filter models for the λ-calculus, filter models being defined through systems with intersection types. The principal typing for the system of [15] was presented in [16], where a relation on pairs $\langle$basis;type$\rangle$ is used in order to overcome technical difficulties of the expansion operation.

All the systems mentioned above deal with λ-terms. Intersection types, however, can be of much help also for the investigation of the functional properties, as well as the models, of concurrent functional languages. The parallel and non-deterministic λ-calculus is an extension of the standard λ-calculus with a non-deterministic choice operator $+$ and a parallel operator $||$ [10]. Its theoretical significance in Computer Science is that it allows to carry out a fine tuned analysis of the interaction between functional and parallel primitives in view of their possible integration. The approach taken in [10] is that of considering the non-deterministic choice between two processes as their meet and the parallel of two processes as their join. This implies that both M and N must have a property to assure that also $M+N$ has it, while $M||N$ has a property as soon as either M or N has it.

The parallel and non-deterministic λ-calculus has been thoroughly investigated in the past years, both syntactically and semantically, by using the finitary logical description of its semantics in terms of intersection and union types [11, 12, 2, 13]. Intersection and union types dually reflect the conjunctive and disjunctive interpretations of $||$ and $+$, respectively. This paper can be viewed as part of the above mentioned investigation, since it addresses the issue of principal typings for the parallel and non-deterministic λ-calculus. An essential use will be done of the notion of approximant and its properties, as described in [12]. The approach of [16] will be taken to derive all possible types of a term from the principal one.

2. The parallel and non-deterministic λ-calculus

As in [12], let $\Lambda_{+||}$ be the set of pure λ-terms enriched with the binary operators $+$ and $||$, that is the set of expressions generated by the following grammar:

$$M ::= \mathsf{x} \mid \lambda\mathsf{x}.M \mid MM \mid M+M \mid M||M,$$

where x ranges over a denumerable set of term-variables. As usual, $FV(M)$ is the set of variables which occur free in M. We consider terms modulo α-conversion. To simplify the notation, we assume that the abstraction and the application take precedence over $+$ and $||$.

On $\Lambda_{+||}$ we define a reduction relation, which is an extension of the β-reduction of classical λ-calculus. More precisely, we make $+$ and $||$ asynchronous evaluators of their arguments. Moreover, since every term in $\Lambda_{+||}$ represents a function, the application of M *op* N (*op* being $+$ or $||$) to L reduces to ML *op* NL (rules $(+_{app})$ and $(||_{app})$). This reduction relation was already introduced in [12].

Definition 2.1. *The relation $\rightarrow$ is the least binary relation on $\Lambda_{+||}$ satisfying:*

$$\begin{array}{ll}
(\beta) & (\lambda x.M)N \rightarrow M[N/x] \\
(\mu) & M \rightarrow N \Rightarrow LM \rightarrow LN \\
(\nu) & M \rightarrow N \Rightarrow ML \rightarrow NL \\
(\xi) & M \rightarrow N \Rightarrow \lambda x.M \rightarrow \lambda x.N \\
(+_a) & M \rightarrow M' \Rightarrow \left\{ \begin{array}{l} M+N \rightarrow M'+N \\ N+M \rightarrow N+M' \end{array} \right. \\
(||_a) & M \rightarrow M' \Rightarrow \left\{ \begin{array}{l} M||N \rightarrow M'||N \\ N||M \rightarrow N||M' \end{array} \right. \\
(+_{app}) & (M+N)L \rightarrow ML+NL \\
(||_{app}) & (M||N)L \rightarrow ML||NL.
\end{array}$$

$\twoheadrightarrow$ will denote the reflexive and transitive closure of $\rightarrow$, while $=$ its reflexive, transitive and symmetric closure.

It turns out that $+$ and $||$ behave in the same way with respect to $\rightarrow$. The difference between them is established by the order we will consider between approximate normal forms. Let us first define the set of approximate normal forms by extending the standard notion [4](p.366) with terms containing $+$ and $||$.

Definition 2.2. *([12]) Let $\Lambda_{+||\Omega}$ be the language obtained from $\Lambda_{+||}$ by adding the constant Ω. The set of approximate normal forms $\mathcal{A} \subset \Lambda_{+||\Omega}$ is the least one such that:*

1. $\Omega \in \mathcal{A}$;
2. $A_1,\ldots,A_n \in \mathcal{A} \Rightarrow xA_1\ldots A_n \in \mathcal{A}$ $(n \geq 0)$;
3. $A \in \mathcal{A} \Rightarrow \lambda x.A \in \mathcal{A}$;
4. $A_1, A_2 \in \mathcal{A} \Rightarrow A_1+A_2, A_1||A_2 \in \mathcal{A}$.

We define a preorder relation on approximate normal forms which generalizes the classical one taking into account the intended meanings of $+$ as meet and of $||$ as join. Moreover, according to this preorder, an η-redex is always less than its contractum.

Definition 2.3. *([12]) Over the set $\mathcal{A}$ we define $\sqsubseteq$ as the least preorder which makes $\mathcal{A}$ into a distributive lattice with + as meet, $||$ as join and Ω as bottom, and such that:*

1. $\lambda x.\Omega \sqsubseteq \Omega$;
2. $A \sqsubseteq A' \Rightarrow \lambda x.A \sqsubseteq \lambda x.A'$;
3. $A_1 \sqsubseteq A'_1,\ldots,A_n \sqsubseteq A'_n \Rightarrow xA_1..A_n \sqsubseteq xA'_1..A'_n$;
4. $\lambda x.(A||A') \sqsubseteq \lambda x.A||\lambda x.A'$;
5. $\lambda y.xA_1..A_ny \sqsubseteq xA_1..A_n$, *if* $y \notin FV(A_i)$ *for* $1 \leq i \leq n$.

$A \asymp A'$ is short for $A \sqsubseteq A'$ and $A' \sqsubseteq A$. $\sqsubseteq$ is a preorder relation on $\mathcal{A}$ since $A \asymp A$. The set $\mathcal{A}$ becomes a lattice only after taking the quotient by the equivalence relation induced by the preorder $\sqsubseteq$.

Usually [4](p.366), to each pure λ-term M we associate, an approximate normal form $\phi(M)$ obtained by substituting Ω for all subterms which are not head normal forms. The present definition generalizes the standard one simply by making $\phi(\,)$ distribute with respect to + and $||$.

Definition 2.4. *The map $\phi\colon \Lambda_{+||} \to \mathcal{A}$ is defined as follows:*

(i) $\phi(\lambda x_1 \ldots x_n.xM_1 \ldots M_m) = \lambda x_1 \ldots x_n.x\phi(M_1)\ldots\phi(M_m)$;

(ii) $\phi(M+N) = \phi(M) + \phi(N)$;

(iii) $\phi(M||N) = \phi(M)||\phi(N)$;

(iv) $\phi(M) = \Omega$, *otherwise.*

Definition 2.5. *Let $M \in \Lambda_{+||}$, then the set $\mathcal{A}(M)$ of approximants of M is defined by:*

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists M' = M\; A \equiv \phi(M')\}.$$

For example, let us consider the terms $F0$ and $G0$, where

$$F \equiv \Theta(\lambda f x.(x + f\,(\mathsf{Succ}\,x))),$$
$$G \equiv \Theta(\lambda f x.(x||f\,(\mathsf{Succ}\,x))),$$

$\Theta \equiv (\lambda zx.x(zzx))(\lambda zx.x(zzx))$ is the Turing fixed-point combinator, 0 and Succ are the zero and successor of Church numerals, respectively. Let n be the Church numeral for the natural number n, then it is easy to check that, for any n,

$$F0 \twoheadrightarrow 0 + 1 + \ldots + \mathsf{n} + F(\mathsf{Succ}\;\mathsf{n}).$$

So we have $\phi(0+1+\ldots+\mathsf{n}+F(\mathsf{Succ}\;\mathsf{n})) = 0+1+\ldots+\mathsf{n}+\Omega \asymp \Omega$, being + the meet. We get $\mathcal{A}(F0) = \{\Omega\}$.
On the other hand $\mathcal{A}(G0)$ is infinite and it contains all the approximate normal forms of the shape

$$0||1||\ldots||\mathsf{n}||\Omega,$$

for all $n \geq 0$.

The present definition of the sets of approximants simplifies that of [12], but they can be easily shown to be equivalent.

3. A type assignment system for $\Lambda_{+||\Omega}$

Our type assignment system is obtained by extending the standard Curry's system for simple types. We enrich the type syntax by adding the universal type ω, and the intersection and union type constructors.

Definition 3.1. *Let Var be a countable set of type-variables and ω be a type constant.*
The set of types $\mathbf{T}_{\Lambda_{+||}}$ is defined as the smallest set satisfying:

1. $\omega \in \mathbf{T}_{\Lambda+||}$; $Var \subseteq \mathbf{T}_{\Lambda_{+||}}$;
2. $\sigma,\tau \in \mathbf{T}_{\Lambda_{+||}} \Rightarrow \sigma \to \tau,\ \sigma \wedge \tau,\ \sigma \vee \tau \in \mathbf{T}_{\Lambda_{+||}}$.

In the following the symbol ϑ will be used to denote a generic element of Var. Symbols like $\alpha, \beta, \mu, \rho, \sigma$ and τ will range over $\mathbf{T}_{\Lambda_{+||}}$. '$\to$' will be assumed to associate to the right, and '$\wedge$' to bind stronger than '$\to$'. The notation $\bigwedge_{i \in I}\sigma_i$ will be short for $\sigma_{i_1} \wedge \ldots \wedge \sigma_{i_n}$, where $I = \{i_1,\ldots,i_n\}$.

Types are thought of as properties of terms. Accordingly, type inclusion represents logical implication. The system has a universal type ω, the property which trivially holds of everything. Therefore, any type will be less than ω. As usual in type assignment systems for λ-calculi, the arrow type is a "function space" constructor. M has type $\sigma \to \tau$ if, for all N having type σ, MN has type τ. With respect to the order, the arrow is co-variant in the second argument and counter-variant in the first argument. Finally $\sigma \wedge \tau$ and $\sigma \vee \tau$ have a conjunctive and disjunctive meaning, respectively.

Definition 3.2.

1. *Let $\leq$ be the smallest preorder over types such that $\langle \mathbf{T}_{\Lambda_{+||}}, \leq \rangle$ is a distributive lattice (taking the quotient), in which $\wedge$ is the meet, $\vee$ is the join and ω is the top, and moreover the arrow satisfies:*
   (a) $\omega \leq \omega \to \omega$;
   (b) $(\sigma \to \mu) \wedge (\sigma \to \tau) \leq \sigma \to \mu \wedge \tau$;
   (c) $\sigma' \leq \sigma, \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$.
2. *$\sigma = \tau$ is short for $\sigma \leq \tau \leq \sigma$.*

It is easy to verify that the relation $\leq$ on $\mathbf{T}_{\Lambda_{+||}}$ is preserved under type-substitution.

Property 3.3.

$$\sigma \leq \tau \Rightarrow \sigma[\vartheta := \mu] \leq \tau[\vartheta := \mu].$$

We now turn to the typing rules for non-deterministic and parallel operators. Since $+$ is interpreted as the meet of its arguments, we type $M + N$ with σ if both M and N can be typed with σ. This is also the choice of [1].
Conversely, $M||N$ represents the join of M and N. It follows that one is entitled to type $M||N$ with σ as soon as M or N (or both) can be typed with σ. See [6] for further motivations.

This suggests the following typing rules

$$\frac{B \vdash M:\sigma \quad B \vdash N:\sigma}{B \vdash M+N:\sigma} \quad \frac{B \vdash M:\sigma}{B \vdash M||N:\sigma} \quad \frac{B \vdash N:\sigma}{B \vdash M||N:\sigma}.$$

The inclusion relation $\leq$ among types turns $\wedge$ into the meet and $\vee$ into the join, and we have both a subtyping and an intersection introduction rule, namely

$$\frac{B \vdash M:\sigma \quad \sigma \leq \tau}{B \vdash M:\tau} \qquad \frac{B \vdash M:\sigma \quad B \vdash M:\tau}{B \vdash M:\sigma \wedge \tau}.$$

Therefore the rules for $+$ and $||$ above are equivalent to

$$\frac{B \vdash M:\sigma \; B \vdash N:\tau}{B \vdash M+N:\sigma \vee \tau} \qquad \frac{B \vdash M:\sigma \; B \vdash N:\tau}{B \vdash M||N:\sigma \wedge \tau}.$$

We have the usual rules dealing with the arrow type constructor. We add a rule (ω) which takes into account the universal character of ω, and a standard rule of introduction of $\wedge$. Moreover we use the preorder on types in a subsumption rule.

Definition 3.4. *([12])*

1. *A statement is an expression of the form $M : \sigma$, where M (the subject) belongs to $\Lambda_{+||\Omega}$ and σ (the predicate) is an element of $\mathbf{T}_{\Lambda_{+||}}$. A basis B is a set of statements such that subjects are pairwise distinct variables.*
2. *The type assignment system is defined by the following natural deduction axioms and rules.*

$$(\omega) \quad B \vdash M:\omega$$

$$(Ax) \quad B, x:\sigma \vdash x:\sigma$$

$$(\to I) \quad \frac{B, x:\sigma \vdash M:\tau}{B \vdash \lambda x.M:\sigma \to \tau}$$

$$(\to E) \quad \frac{B \vdash M:\sigma \to \tau \quad B \vdash N:\sigma}{B \vdash MN:\tau}$$

$$(+) \quad \frac{B \vdash M:\sigma \, B \vdash N:\sigma}{B \vdash M+N:\sigma}$$

$$(||) \quad \frac{B \vdash M:\sigma}{B \vdash M||N:\sigma} \quad \frac{B \vdash N:\sigma}{B \vdash M||N:\sigma}$$

$$(\wedge I) \quad \frac{B \vdash M:\sigma \; B \vdash M:\tau}{B \vdash M:\sigma \wedge \tau}$$

$$(\leq) \quad \frac{B \vdash M:\sigma \; \sigma \leq \tau}{B \vdash M:\tau},$$

where $B, x : \sigma$ is short for $B \cup \{x : \sigma\}$, when x does not occur in B.

We shall write $B \vdash M : \sigma$ if $B \vdash M : \sigma$ is derivable in the above system.

Remark 3.5. *It is easy to verify that rules* $(\wedge E)$ *and* $(\vee I)$ *defined by:*

$$(\wedge E)\quad \frac{B \vdash M : \sigma \wedge \tau}{B \vdash M : \sigma} \quad \frac{B \vdash M : \sigma \wedge \tau}{B \vdash M : \tau}$$

$$(\vee I)\quad \frac{B \vdash M : \sigma}{B \vdash M : \sigma \vee \tau} \quad \frac{B \vdash M : \tau}{B \vdash M : \sigma \vee \tau}$$

are derivable. The following rules are admissible:

$$(+\vee)\quad \frac{B \vdash M : \sigma \quad B \vdash N : \tau}{B \vdash M + N : \sigma \vee \tau}$$

$$(||\wedge)\quad \frac{B \vdash M : \sigma \quad B \vdash N : \tau}{B \vdash M||N : \sigma \wedge \tau}$$

$$(\leq L)\quad \frac{B, x : \sigma \vdash M : \tau \quad \sigma' \leq \sigma}{B, x : \sigma' \vdash M : \tau}$$

as proved in [12]. Also the weakening rule is admissible, since we consider terms modulo α-conversion.

We define $Dom(B) = \{x \mid x{:}\sigma \in B \;\&\; \sigma \neq \omega\}$ and we assume $x{:}\omega \in B$ whenever $x \notin Dom(B)$. This is sound in view of rule (ω). Moreover, we consider equality of bases modulo type equivalence, i.e. when we write $B = B'$ we mean that $x : \sigma \in B$ iff $x : \sigma' \in B'$ with $\sigma = \sigma'$ and vice-versa. This is justified by rule $(\leq L)$.

Notation. In the following we shall sometimes refer to the stronger basis which can be formed out of two given bases. This is done by taking the intersection of the types which are predicates of the same variable:

$$B \uplus B' = \{x{:}\, \sigma \wedge \tau \mid x{:}\, \sigma \in B \text{ and } x{:}\, \tau \in B'\}.$$

Accordingly we define:

$$B \Subset B' \;\Leftrightarrow\; \exists B''.\, B \uplus B'' = B'.$$

Notice that $B \Subset B'$ iff $x : \sigma \in B$ implies $x : \sigma' \in B'$ for some $\sigma' \leq \sigma$. In particular we have $B \uplus B \Subset B$ and $B \uplus \{x : \omega\} \Subset B$.

As expected, derivability is preserved under type substitution, i.e.

Lemma 3.6.

$$B \vdash M : \sigma \Rightarrow B[\vartheta := \tau] \vdash M : \sigma[\vartheta := \tau].$$

Using a simple reduction on derivations one can show that all typing rules can be reversed, i.e. that our type assignment system enjoys the following structural properties. All points are proved in [12], but the first one which can be easily proved by induction on deductions.

Lemma 3.7. (Generation Lemma)

(i) $B \vdash \Omega : \sigma \Rightarrow \sigma = \omega$.

(ii) $B \vdash x : \tau \;\Leftrightarrow\; \exists \sigma.\, x : \sigma \in B \;\&\; \sigma \leq \tau$;

(iii) $B \vdash \lambda x.M : \tau \Leftrightarrow \exists \mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n.\, B \vdash \lambda x.M : \bigwedge_{i=1}^{n} (\mu_i \to \nu_i) \;\&\; \bigwedge_{i=1}^{n} (\mu_i \to \nu_i) \leq \tau$;

(iv) $B \vdash \lambda x.M : \sigma \to \tau \Leftrightarrow B, x : \sigma \vdash M : \tau$;

(v) $B \vdash MN : \tau \;\Leftrightarrow\; \exists \sigma.\, B \vdash M : \sigma \to \tau \;\&\; B \vdash N : \sigma$;

(vi) $B \vdash M + N : \sigma \Leftrightarrow B \vdash M : \sigma \;\&\; B \vdash N : \sigma$:

(vii) $B \vdash M||N : \tau \Leftrightarrow \exists \sigma, \sigma'.\, \sigma \wedge \sigma' \leq \tau \;\&\; B \vdash M : \sigma \;\&\; B \vdash N : \sigma'$.

The invariance of types under subject conversion with respect to $=$ is now an easy consequence of the previous Lemmas.

Theorem 3.8. (Subject Conversion [12]) *For any terms M, N, basis B and type σ:*

$$B \vdash M : \sigma,\; M = N \Rightarrow B \vdash N : \sigma.$$

Using a variant of Tait's computability technique, [12] proves that the set of types which can be deduced for any term coincides with the union of the sets of types deducible for its approximants.

Theorem 3.9. (Approximation Theorem [12]) *For any term M, basis B and type σ:*

$$B \vdash M : \sigma \Leftrightarrow \exists A \in \mathcal{A}(M).\, B \vdash A : \sigma.$$

4. Principal pairs for approximate normal forms

Following [16], we associate to each approximate normal form the set of types which can be deduced for it. Since we also consider open terms, we need to define sets of pairs $\langle$basis; type$\rangle$. The correctness and completeness of this definition are proved in Lemma 4.2(ii).

Definition 4.1. *Let* $\mathbf{P}$ *be set of all pairs* $\langle B; \sigma\rangle$ *where* B *is basis and* $\sigma \in \mathbf{T}_{\Lambda_{+||}}$. *Given* $A \in \mathcal{A}$, *we define the set* $\mathsf{AP}(A)$ *of pairs admissible for* A *as the subset of* $\mathbf{P}$ *inductively defined on the structure of* A *as follows:*

a) $A \equiv \Omega$
$\mathsf{AP}(A) =_{Def} \{\langle B; \gamma\rangle | B \text{ is any basis}, \gamma = \omega\}$.

b) $A \equiv x$
$\mathsf{AP}(A) =_{Def} \{\langle B; \gamma\rangle | B \text{ is any basis}, \gamma = \omega\} \cup \{\langle B, x : \sigma; \tau\rangle | B \text{ is any basis and } \sigma \leq \tau\}$.

c) $A \equiv \lambda x.A_1$.
$\mathsf{AP}(A) =_{Def} \{\langle B; \gamma\rangle | B \text{ is any basis}, \gamma = \omega\} \cup \{\langle B; \tau\rangle | \exists \mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n. \langle B, x : \mu_i; \nu_i\rangle \in \mathsf{AP}(A_1) \text{ and } \tau \geq \bigwedge_{i=1}^{n}(\mu_i \to \nu_i)\}$

d) $A \equiv xA_1 \ldots A_n$
$\mathsf{AP}(A) =_{Def} \{\langle B; \gamma\rangle | B \text{ is any basis}, \gamma = \omega\} \cup \{\langle B; \tau\rangle | \exists \sigma_1, \ldots, \sigma_n.([\forall 1 \leq i \leq n. \langle B; \sigma_i\rangle \in \mathsf{AP}(A_i)] \text{ and } [\langle B; \sigma_1 \to \ldots \to \sigma_n \to \tau\rangle \in \mathsf{AP}(x)])\}$.

e) $A \equiv A_1 + A_2$
$\mathsf{AP}(A_1 + A_2) =_{Def} \{\langle B; \tau\rangle | \langle B; \tau\rangle \in \mathsf{AP}(A_1) \cap \mathsf{AP}(A_2)\}$.

f) $A \equiv A_1 || A_2$
$\mathsf{AP}(A_1 || A_2) =_{Def} \{\langle B; \tau\rangle | \exists \sigma_1, \sigma_2.\ \tau \geq \sigma_1 \wedge \sigma_2, \langle B; \sigma_1\rangle \in \mathsf{AP}(A_1), \langle B; \sigma_2\rangle \in \mathsf{AP}(A_2)\}$.

Lemma 4.2.

(i) $\langle B; \tau\rangle \in \mathsf{AP}(A), B \Subset B' \Rightarrow \langle B'; \tau\rangle \in \mathsf{AP}(A)$.

(ii) $\langle B; \tau\rangle \in \mathsf{AP}(A) \Leftrightarrow B \vdash A : \tau$.

Proof. (i) Easy, by induction on the definition of $\mathsf{AP}(A)$.
(ii) $\Rightarrow$) Easy, by induction on the structure of A.
$\Leftarrow$) By induction on the structure of A.

- $A \equiv \Omega$
  Immediate by definition of admissible pair and Lemma 3.7(i).

- $A \equiv x$
  By definition of admissible pair and Lemma 3.7(ii).

- $A \equiv \lambda x.A_1$
  By Lemma 3.7(iii):
  (a) $\exists \mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n.\ B \vdash \lambda x.A_1 : \bigwedge_{i=1}^{n}(\mu_i \to \nu_i)$
  (b) $\bigwedge_{i=1}^{n}(\mu_i \to \nu_i) \leq \tau$.
  Hence, by $(\wedge E)$, from (a) it follows that, for all $1 \leq i \leq n$, $B \vdash \lambda x.A_1 : \mu_i \to \nu_i$ and, by Lemma 3.7(iv), $B, x : \mu_i \vdash A_1 : \nu_i$. By the induction hypothesis, for all $1 \leq i \leq n$, $\langle B, x : \mu_i; \nu_i\rangle \in \mathsf{AP}(A_1)$ and hence, by (b) and the definition of admissible pair, $\langle B; \tau\rangle \in \mathsf{AP}(A)$.

- $A \equiv xA_1 \ldots A_n$
  By Lemma 3.7(v), $\exists \sigma_1, \ldots, \sigma_n.\ B \vdash x : \sigma_1 \to \ldots \to \sigma_n \to \tau, B \vdash A_i : \sigma_i$ $(1 \leq i \leq n)$. By the induction hypothesis, $\langle B; \sigma_1 \to \ldots \to \sigma_n \to \tau\rangle \in \mathsf{AP}(x)$, $\langle B; \sigma_i\rangle \in \mathsf{AP}(A_i)$ $(1 \leq i \leq n)$. $\langle B; \tau\rangle \in \mathsf{AP}(A)$ is now an immediate consequence of the definition of admissible pair.

- $A \equiv A_1 + A_2$
  By Lemma 3.7(vi), $B \vdash A_1 : \tau$, $B \vdash A_2 : \tau$. By the induction hypothesis $\langle B; \tau\rangle \in \mathsf{AP}(A_1)$ and $\langle B; \tau\rangle \in \mathsf{AP}(A_2)$. It follows that, by definition of admissible pair, $\langle B; \tau\rangle \in \mathsf{AP}(A_1 + A_2)$.

- $A \equiv A_1 || A_2$
  By Lemma 3.7(vii), $\exists \sigma_1, \sigma_2.\ \sigma_1 \wedge \sigma_2 \leq \tau, B \vdash A_i : \sigma_i, i = 1, 2$. By the induction hypothesis $\langle B; \sigma_1\rangle \in \mathsf{AP}(A_1)$ and $\langle B; \sigma_2\rangle \in \mathsf{AP}(A_2)$. By definition of admissible pair we get $\langle B; \tau\rangle \in \mathsf{AP}(A_1 || A_2)$.

□

We define a relation $\preceq_A$ on **P** such that a pair $\langle B;\tau\rangle$ relates to itself iff it is admissible for A, i.e. iff $B \vdash A : \tau$ is derivable. This is an immediate consequence of the following Lemma 4.4.

In Definition 4.3, the first point takes into account rule (ω). The second point is justified by Lemma 3.6. Rules $(\leq)$ and $(\leq L)$ give us the third point. Point (4) makes $\preceq_A$ transitive. Point (5) is due to rule $(\wedge I)$. Notice that all these points do not change the subject of derivation, which is here represented by the subscript of the order relation (i.e. by A in $\preceq_A$). Instead, the other points change the subscript of the order relation, and correspond respectively to clauses (c), (d), and (e) (f) of Definition 4.1.

Definition 4.3. *For every $A \in \mathcal{A}$ we define on* **P** *the $\preceq_A$ relation as the least one satisfying the following conditions:*

1. $\langle B;\tau\rangle \in \mathsf{AP}(A) \;\Rightarrow\; \langle B;\tau\rangle \preceq_A \langle\emptyset;\omega\rangle$.
2. $\langle B;\tau\rangle \in \mathsf{AP}(A) \Rightarrow \langle B;\tau\rangle \preceq_A \langle B[\vartheta := \sigma];\tau[\vartheta := \sigma]\rangle$.
3. $\langle B_1;\tau_1\rangle \in \mathsf{AP}(A),\ B_1 \Subset B_2,\ \tau_1 \leq \tau_2 \;\Rightarrow\; \langle B_1;\tau_1\rangle \preceq_A \langle B_2;\tau_2\rangle$.
4. $\langle B_1;\tau_1\rangle \preceq_A \langle B_2;\tau_2\rangle \preceq_A \langle B_3;\tau_3\rangle \;\Rightarrow\; \langle B_1;\tau_1\rangle \preceq_A \langle B_3;\tau_3\rangle$.
5. $\langle B_1;\tau_1\rangle \preceq_A \langle B_2;\tau_2\rangle, \langle B_1';\tau_1'\rangle \preceq_A \langle B_2';\tau_2'\rangle \;\Rightarrow\; \langle B_1 \uplus B_1';\tau_1\wedge\tau_1'\rangle \preceq_A \langle B_2 \uplus B_2';\tau_2\wedge\tau_2'\rangle$.
6. $\langle B_1, x:\sigma_1;\tau_1\rangle \preceq_A \langle B_2, x:\sigma_2;\tau_2\rangle \;\Rightarrow\; \langle B_1;\sigma_1 \to \tau_1\rangle \preceq_{\lambda x.A} \langle B_2;\sigma_2 \to \tau_2\rangle$.
7. $\langle B_i;\tau_i\rangle \preceq_{A_i} \langle B_i';\tau_i'\rangle\ (i = 1,...,n) \;\Rightarrow\; \langle(\biguplus_{i=1}^n B_i) \uplus \{x : \tau_1 \to \ldots \to \tau_n \to \sigma\};\sigma\rangle \preceq_{xA_1...A_n} \langle(\biguplus_{i=1}^n B_i') \uplus \{x : \tau_1' \to \ldots \to \tau_n' \to \sigma\};\sigma\rangle$, *for any* σ.
8. $\langle B_1;\tau_1\rangle \preceq_{A_1} \langle B_1';\tau_1'\rangle$ & $\langle B_2;\tau_2\rangle \preceq_{A_2} \langle B_2';\tau_2'\rangle \;\Rightarrow\; \langle B_1 \uplus B_2;\tau_1 \vee \tau_2\rangle \preceq_{A_1+A_2} \langle B_1' \uplus B_2';\tau_1' \vee \tau_2'\rangle$ & $\langle B_1 \uplus B_2;\tau_1 \wedge \tau_2\rangle \preceq_{A_1||A_2} \langle B_1' \uplus B_2';\tau_1' \wedge \tau_2'\rangle$.

$\langle B_1;\tau_1\rangle \approx_A \langle B_2;\tau_2\rangle$ is short for $\langle B_1;\tau_1\rangle \preceq_A \langle B_2;\tau_2\rangle$ and $\langle B_2;\tau_2\rangle \preceq_A \langle B_1;\tau_1\rangle$.

Notice that by (3) of the previous definition, the relation $\preceq_A$ is reflexive.

Lemma 4.4. $\langle B_1;\tau_1\rangle \preceq_A \langle B_2;\tau_2\rangle \Rightarrow \langle B_1;\tau_1\rangle \in \mathsf{AP}(A)$ & $\langle B_2;\tau_2\rangle \in \mathsf{AP}(A)$.

Proof. The thesis follows immediately from Lemma 4.2 once we show that $B_1 \vdash A : \tau_1$ and $B_2 \vdash A : \tau_2$. This can be easily proved by induction on the definition of $\preceq_A$. □

For every $A \in \mathcal{A}$, $\mathsf{AP}(A)$ can be considered modulo $\approx_A$, then $\preceq_A$ becomes a partial order on $\mathsf{AP}(A)$.

The following definition of principal pair is a generalization to our calculus of the one given in [8], [18], and [3], where it was used to prove the principal type property for various intersection type disciplines.

Definition 4.5. *Let $A \in \mathcal{A}$. We define $\mathsf{pp}(A)$, the principal pair of A, by structural induction on A as follows:*

a) $A \equiv \Omega$
$\mathsf{pp}(A) =_{Def} \langle\emptyset;\omega\rangle$;

b) $A \equiv x$
$\mathsf{pp}(A) =_{Def} \langle\{x : \vartheta\};\vartheta\rangle$;

c) $A \equiv \lambda x.A_1$
Let $\mathsf{pp}(A_1) = \langle\mathcal{B}_1;\pi_1\rangle$. We distinguish two cases:
- *If $x \in FV(A_1)$ and $\mathcal{B}_1 \equiv \mathcal{B}, x : \mu_1$, then $\mathsf{pp}(\lambda x.A_1) =_{Def} \langle\mathcal{B};\mu_1 \to \pi_1\rangle$;*
- *If $x \notin FV(A_1)$ then $\mathsf{pp}(\lambda x.A_1) = \langle\mathcal{B}_1;\omega \to \pi_1\rangle$.*

d) $A \equiv xA_1...A_n$
Let $\mathsf{pp}(A_i) = \langle\mathcal{B}_i;\pi_i\rangle$, $(i = 1,\ldots,n)$ (it is possible to assume, w.l.o.g., that such pairs do not share type-variables.)
Then $\mathsf{pp}(A) =_{Def} \langle(\biguplus_{i=1}^n \mathcal{B}_i) \uplus \{x : \pi_1 \to \ldots \to \pi_n \to \vartheta\};\vartheta\rangle$, where ϑ is fresh.

e) $A \equiv A_1 + A_2$
Let $\mathsf{pp}(A_i) = \langle\mathcal{B}_i;\pi_i\rangle$, $i = 1,2$ (it is possible to assume, w.l.o.g., that such pairs do not share type-variables.)
Then $\mathsf{pp}(A_1+A_2) =_{Def} \langle\mathcal{B}_1 \uplus \mathcal{B}_2;\pi_1 \vee \pi_2\rangle$.

f) $A \equiv A_1||A_2$
Let $\mathsf{pp}(A_i) = \langle\mathcal{B}_i;\pi_i\rangle$, $i = 1,2$ (it is possible to assume, w.l.o.g., that such pairs do not share type-variables.)
Then $\mathsf{pp}(A_1||A_2) =_{Def} \langle\mathcal{B}_1 \uplus \mathcal{B}_2;\pi_1 \wedge \pi_2\rangle$.

We assume that principal pairs are taken up to renaming of their type variables, so that we may have a unique principal pair for each approximate normal form.

Clearly the principal pair of A is admissible for A.

Lemma 4.6. *For any* $A \in \mathcal{A}$*:* $\mathsf{pp}(A) \in \mathsf{AP}(A)$.

Proof. By an easy induction on the structure of A, using Lemma 4.2(i) when $A \equiv xA_1 \ldots A_n$, $A \equiv A_1 + A_2$ and $A \equiv A_1 || A_2$.. □

Now we can prove that the principal pair of A is the bottom of $\mathsf{AP}(A)$ with respect to the relation $\preceq_A$.

Theorem 4.7. *Let* $\langle B; \tau\rangle$ *be a pair admissible for a given* $A \in \mathcal{A}$ *and* $\langle \mathcal{B}; \pi\rangle$ *be the principal pair of* A*. Then* $\langle \mathcal{B}; \pi\rangle \preceq_A \langle B; \tau\rangle$.

Proof. By structural induction on A. If $\tau = \omega$ the thesis holds by Lemma 4.6 and Definition 4.3(1),(3).

- $A \equiv \Omega$
  Easy from Lemma 3.7(i) and Definition 4.3(3).

- $A \equiv x$
  Then $\mathsf{pp}(A) = \langle \{x : \vartheta\}; \vartheta\rangle$, where ϑ is a type variable, and $\langle B; \tau\rangle \equiv \langle B, x : \sigma; \tau\rangle$ with $\sigma \leq \tau$. Hence

$$\begin{aligned}\langle \{x : \vartheta\}; \vartheta\rangle &\preceq_x \langle \{x : \sigma\}; \sigma\rangle \\ &\qquad \text{by Definition 4.3(2)} \\ &\preceq_x \langle B, x : \sigma; \tau\rangle \\ &\qquad \text{by Definition 4.3(3).}\end{aligned}$$

- $A \equiv \lambda x.A_1$
  Let $\mathsf{pp}(A_1) = \langle \mathcal{B}_1; \pi_1\rangle$. We distinguish two cases:

  - $x \in FV(A_1)$.
    Then $\mathcal{B}_1 \equiv \mathcal{B}, x : \mu_1$ for some μ_1 and $\pi \equiv \mu_1 \to \pi_1$.
    By Lemmas 3.7(iii) and 4.2(ii), $\tau \geq \bigwedge_{j\in J}(\sigma_j \to \rho_j)$ and $\langle B, x : \sigma_j; \rho_j\rangle \in \mathsf{AP}(A_1)$ for all $j \in J$. We have now
    (a) $\langle \mathcal{B}_1, \pi_1\rangle \preceq_{A_1} \langle B, x : \sigma_j; \rho_j\rangle$ for all $j \in J$, by the induction hypothesis, and hence

$$\begin{aligned}\langle \mathcal{B}; \pi\rangle &\equiv \langle \mathcal{B}; \mu_1 \to \pi_1\rangle \\ &\preceq_A \langle B; \sigma_j \to \rho_j\rangle \\ &\qquad \text{for all } j \in J, \text{ by (a)} \\ &\qquad \text{and Definition 4.3(6)} \\ &\preceq_A \langle B; \bigwedge_{j\in J}(\sigma_j \to \rho_j)\rangle \\ &\qquad \text{by Definition 4.3(5)} \\ &\preceq_A \langle B; \tau\rangle \\ &\qquad \text{by Definition 4.3(3).}\end{aligned}$$

  - $x \notin FV(A_1)$.
    Then $\pi \equiv \omega \to \pi_1$.
    By Lemmas 3.7(iii) and 4.2(ii), $\tau \geq \bigwedge_{j\in J}(\sigma_j \to \rho_j)$ and $\langle B, x : \sigma_j; \rho_j\rangle \in \mathsf{AP}(A_1)$ for all $j \in J$. We have now $\langle \mathcal{B}_1; \pi_1\rangle \preceq_{A_1} \langle B, x : \sigma_j; \rho_j\rangle$, for all $j \in J$, by the induction hypothesis. This implies
    (a) $\langle \mathcal{B}_1, x : \omega; \pi_1\rangle \preceq_{A_1} \langle B, x : \sigma_j; \rho_j\rangle$, for all $j \in J$, by Definition 4.3(3),(4), since $\mathcal{B}_1, x : \omega \Subset \mathcal{B}_1$, and hence

$$\begin{aligned}\langle \mathcal{B}; \pi\rangle &\equiv \langle \mathcal{B}_1; \omega \to \pi_1\rangle \\ &\preceq_A \langle B; \sigma_j \to \rho_j\rangle \\ &\qquad \text{for all } j \in J, \\ &\qquad \text{by (a) and Def. 4.3(6)} \\ &\preceq_A \langle B; \bigwedge_{j\in J}(\sigma_j \to \rho_j)\rangle \\ &\qquad \text{by Definition 4.3(5)} \\ &\preceq_A \langle B; \tau\rangle \\ &\qquad \text{by Definition 4.3(3).}\end{aligned}$$

- $A \equiv xA_1 \ldots A_n$
  Let $\mathsf{pp}(A_i) = \langle \mathcal{B}_i; \pi_i\rangle$ $(i = 1, \ldots, n)$.

  Then $\langle \mathcal{B}; \pi\rangle \equiv \langle (\biguplus_{i=1}^{n} \mathcal{B}_i) \uplus \{x : \pi_1 \to \ldots \to \pi_n \to \vartheta\}; \vartheta\rangle$, where ϑ is fresh. Note that the pairs $\langle \mathcal{B}_i; \pi_i\rangle$, $(i = 1, \ldots, n)$ do not share type-variables.
  By definition of admissible pair, $\langle B; \tau\rangle \in \mathsf{AP}(A)$ implies that there exist $\sigma_1, \ldots, \sigma_n$ such that $\langle B; \sigma_i\rangle \in \mathsf{AP}(A_i)$ and $\langle B; \sigma_1 \to \ldots \to \sigma_n \to \tau\rangle \in \mathsf{AP}(x)$. Now
  (a) $\langle \mathcal{B}_i; \pi_i\rangle \preceq_{A_i} \langle B; \sigma_i\rangle$, $(i = 1, \ldots, n)$ by

the induction hypothesis and hence

$$\begin{aligned}\langle \mathcal{B};\pi\rangle \equiv & \langle(\biguplus_{i=1}^{n}\mathcal{B}_i)\uplus\{x{:}\pi_1\rightarrow\ldots\rightarrow\pi_n\rightarrow\\ & \rightarrow\vartheta\};\vartheta\rangle\\ \preceq_A & \langle B\uplus\{x{:}\sigma_1\rightarrow\ldots\rightarrow\sigma_n\rightarrow\vartheta\};\vartheta\rangle\\ & \text{by (a) and Def. 4.3(7)}\\ \preceq_A & \langle B\uplus\{x{:}\sigma_1\rightarrow\ldots\rightarrow\sigma_n\rightarrow\tau\};\tau\rangle\\ & \text{by Def. 4.3(2).}\end{aligned}$$

- $A \equiv A_1 + A_2$
  Then $\langle \mathcal{B};\pi\rangle \equiv \langle \mathcal{B}_1\uplus\mathcal{B}_2;\pi_1\vee\pi_2\rangle$ where $\langle \mathcal{B}_i;\pi_i\rangle \equiv \mathsf{pp}(A_i)$, $(i = 1,2)$. By definition of admissible pair, $\langle B;\tau\rangle \in \mathsf{AP}(A_i)$ for $i = 1,2$.
  By the induction hypothesis, for i=1,2: $\langle \mathcal{B}_i;\pi_i\rangle \preceq_{A_i} \langle B;\tau\rangle$. And hence

$$\begin{aligned}\langle \mathcal{B};\pi\rangle \equiv & \langle \mathcal{B}_1\uplus\mathcal{B}_2;\pi_1\vee\pi_2\rangle\\ \preceq_A & \langle B;\tau\rangle\\ & \text{by Definition 4.3(8), (3).}\end{aligned}$$

- $A = A_1||A_2$
  Then $\langle \mathcal{B};\pi\rangle \equiv \langle \mathcal{B}_1\uplus\mathcal{B}_2;\pi_1\wedge\pi_2\rangle$, where $\langle \mathcal{B}_i;\pi_i\rangle = \mathsf{pp}(A_i)$, $(i = 1,2)$. By definition of admissible pair, $\tau \geq \tau_1\wedge\tau_2$, and $\langle B;\tau_i\rangle \in \mathsf{AP}(A_i)$ for some τ_i and $i = 1,2$.
  By the induction hypothesis, for i=1,2: $\langle \mathcal{B}_i;\pi_i\rangle \preceq_{A_i} \langle B;\tau_i\rangle$, and hence

$$\begin{aligned}\langle \mathcal{B};\pi\rangle \equiv & \langle \mathcal{B}_1\uplus\mathcal{B}_2;\pi_1\wedge\pi_2\rangle\\ \preceq_A & \langle B\uplus B;\tau_1\wedge\tau_2\rangle\\ & \text{by Definition 4.3(8)}\\ \preceq_A & \langle B;\tau\rangle\\ & \text{by Definition 4.3(3).}\end{aligned}$$

□

Now we can obtain the desired result, i.e. an axiomatic characterization of the types which can be deduced for terms in $\Lambda_{+||}$.

Theorem 4.8.

(i) Let M be a term in $\Lambda_{+||}$ such that $\mathcal{A}(M)$ is finite. Moreover, let $A \equiv \bigsqcup\mathcal{A}(M)$.
Then $B \vdash M : \tau \;\Leftrightarrow\; \mathsf{pp}(A) \preceq_A \langle B;\tau\rangle$.

(ii) Let M be a term in $\Lambda_{+||}$ such that $\mathcal{A}(M)$ is infinite.
Then $B \vdash M : \tau \;\Leftrightarrow\; \exists A \in \mathcal{A}(M).\ \mathsf{pp}(A) \preceq_A \langle B;\tau\rangle$.

Proof. Immediate from Theorems 3.9 and 4.7.
□

5. Conclusions

The present paper answers the problem of determining principal types for the parallel and non-deterministic λ-calculus. This is achieved by extending existing techniques for the standard λ-calculus.

A natural development of the present research is to find effective operations for deriving types from principal pairs following [8, 18, 3]. This would allow the construction of a typing semi-algorithms for the parallel and non-deterministic λ-calculus, which generalizes that given for the standard λ-calculus [17].

Acknowledgements. The present version of this paper has strongly benefited from comments and remarks by Anonymous Referees.
The second author wishes to thank Silvia Puglisi for many interesting discussions about higher-order process algebras.

References

[1] M. Abadi, "A Semantics for Static Type Inference in a Non-Deterministic Language", *Info. and Comp.* 109, 1994, 300–306.

[2] F. Alessi, M. Dezani–Ciancaglini, U. de'Liguoro, "May and Must Convergency in Concurrent λ-calculus", *LNCS* 841, Springer–Verlag, Berlin, 1994, 211–220.

[3] S. van Bakel, "Principal Type Schemes for the Strict Type Assignment System", *J. Logic Comp.* 3(6), 1993, 643–670.

[4] H. P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, 2nd ed., North–Holland, Amsterdam, 1984.

[5] H. Barendregt, M. Coppo, M. Dezani–Ciancaglini, "A Filter Lambda Model and the Completeness of Type Assignment", *J. of Symbolic Logic* 48, 1983, 931–940.

[6] G. Boudol, "A Lambda Calculus for (Strict) Parallel Functions", *Info. and Comp.* 108, 1994, 51–127.

[7] M. Coppo, M. Dezani–Ciancaglini, F. Honsell, G. Longo, "Extended Type Structures and Filter Lambda Models", *Logic Colloquium 82*, North–Holland, Amsterdam, 1984, 241–262.

[8] M. Coppo, M. Dezani–Ciancaglini, B. Venneri, "Principal Type Schemes and λ-calculus Semantics", *To H. B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, New York, 1980, 535–560.

[9] H. B. Curry, R. Feys, *Combinatory Logic*, Vol.1, North–Holland, Amsterdam.

[10] M. Dezani–Ciancaglini, U. de'Liguoro, A. Piperno, "Filter Models for a Parallel and Non-Deterministic λ-calculus", *LNCS* 711, Springer–Verlag, Berlin, 1993, 403–412.

[11] M. Dezani–Ciancaglini, U. de'Liguoro, A. Piperno, "Fully Abstract Semantics for Concurrent λ-calculus", *LNCS* 789, Springer–Verlag, Berlin, 1994, 16–35.

[12] M. Dezani–Ciancaglini, U. de'Liguoro, A. Piperno, "Filter Models for Conjunctive-Disjunctive λ-calculi", *Theor. Comp. Sci.* 170(1–2), 83–128, 1996.

[13] M. Dezani–Ciancaglini, U. de'Liguoro, A. Piperno, "A Filter Model for Concurrent λ-calculi", to appear in *SIAM J. of Comp.*.

[14] R. Hindley, "The Principal Type Scheme of an Object in Combinatory Logic", *Trans. Amer. Math. Soc.* 146, 1969, 29–60.

[15] B. Jacobs, I. Margaria, M. Zacchi, "Filter Models with Polymorphic Types", *Theor.Comp.Sci.* 95, 1992, 143–158.

[16] I. Margaria, M. Zacchi, "Principal Typing in a ∀∧-Discipline", *J.Logic Comp.*, 5(3), 1995, 367–381.

[17] S. Ronchi della Rocca, "Principal Type Scheme and Unification for Intersection Type Discipline", *Theor. Comp. Sci.* 59, 1988, 181–209.

[18] S. Ronchi della Rocca, B. Venneri, "Principal Type Schemes for an Extended Type Theory", *Theor. Comp. Sci.* 28, 1984, 151–169.

Received: October, 1996
Accepted: July, 1997

Contact address:

Ali S. Aoun
Faculty of Science
Mathematics Department
Hacettepe University
06532 Beytepe, Ankara
Turkey

Franco Barbanera
Mariangiola Dezani–Ciancaglini
Dipartimento di Informatica
Universitá di Torino
Corso Svizzera 185
10149 Torino
Italy

Şeref Mirasyedioğlu
Faculty of Education
Mathematics Department
Gazi University
Beşevler, Ankara
Turkey