

# Linear Interpolation for the Higher-Order Matching Problem<sup>1</sup>

Aleksy Schubert  
alx@mimuw.edu.pl

January, 1996

**Abstract.** We present here a particular case of the higher order matching problem — the linear interpolation problem. The problem consists in solving a collection of higher order matching equations of the shape  $xM_1 \dots M_k = N$ , where  $x$  is the only unknown quantity. We prove recursive equivalence of the higher order matching problem and the linear interpolation problem. We also investigate decidability of a special case of the fifth order linear interpolation problem. The restriction we consider consists in that arguments of variables from the main abstraction in terms  $M_1, \dots, M_k$  cannot contain variables from the main abstraction.

## 1 Preface

The higher-order matching problem for simply typed  $\lambda$ -calculus has been considered since 1976 ([Hue76]). There were proposed several partial solutions of the problem (second order matching — [GH78]; correct, but without a proof of completeness, algorithm — [Wol89]; third order matching — [Dow93]; fourth order matching — [Pad96]).

In this paper, we present the linear interpolation problem. This problem is interesting since to construct a solution for such a problem we deal with a single object, not a set of objects as in the case of the matching problem in general formulation. Moreover, V. Padovani investigates a similar problem in his paper [Pad96]. The Padovani's problem consists in solving the pair of sets  $\{\Phi, \Psi\}$  of interpolation equations. A solution of such a problem is a concretisation of unknown quantities which satisfies each equation in the set  $\Phi$  and does not satisfy any equation in the set  $\Psi$ . Decidability of the problem implies decidability of the matching problem as proven in [Pad96].

---

<sup>1</sup> This work has been partly supported by ESPRIT BRA 7232 GENTZEN, and KBN 8 T11C 034 10 grants.

In the second part of the paper, we look into decidability of a special case of the fifth order linear interpolation problem. The restriction we consider is that arguments of variables from the main abstraction in terms  $M_1, \dots, M_k$  cannot contain occurrences of variables from the main abstraction.

This issue is interesting, since it gives constructors of proof-checkers and proof-assistants possibility of solving some fifth order matching equations.

This paper is organised as follows — in Section 2 we present some basic definitions and define some useful notation, in Section 3 we prove recursive equivalence of the higher-order matching problem and the interpolation problem, and in Section 4 we prove our decidability result.

The present paper contains only a sketch of the proof. More details can be found in the technical report [Sch96].

*Acknowledgements.* Thanks to prof. J. Tiuryn for encouragement to deal with the higher-order matching problem and for discussions on the topic we had. I also thank prof. P. Urzyczyn, Robert Maron, Grzegorz Grudziński for many prolific debates.

## 2 Basic definitions

### 2.1 Types and terms

We assume the reader is familiar with the notions of  $\lambda$ -term,  $\beta$  and  $\eta$  reduction, type systems. Corresponding definitions can be found in [Bar84] or [Bar92].

Additionally, we understand the simply typed  $\lambda$ -calculus in the formulation, where we have a type indexed set  $\mathcal{E}$  of unknown quantities symbols, and terms may contain these symbols in unbound positions. The set of all simply-typed  $\lambda$ -terms is denoted by  $\Lambda_{\rightarrow}$ . The set of free variables in a term  $t$  is denoted by  $FV(t)$  and the set of constants by  $Const(t)$ .

We assume, except when stated explicitly, all terms are in  $\beta$ -normal,  $\eta$ -long form. When necessary the normal form of a term  $t$  is denoted by  $NF(t)$ .

Moreover, we mean by a *closed term* a term defined as usual but we impose one additional condition — the term cannot contain unknown quantities. We denote by  $Cl(A)$  the restriction of the set  $A$  of terms to closed terms.

Terms are denoted by capital Latin letters (for instance  $A, D, M \dots$ ) and by small Latin letters starting from  $s$  ( $s, t, \dots$ ). Types are denoted by small Greek letters starting from  $\sigma$  ( $\sigma, \tau, \dots$ ). We denote by  $Typ(t)$  the set of all types of subterms of the term  $t$ .  $SubTyp(A)$  denotes the set of all the subtypes of types from  $A$ . Notions of order, path, Böhm tree, occurrence, graft are taken from [Dow93]. From now on, except when stated explicitly, we use the name “term” to refer to the Böhm tree of the term in question.

### 2.2 The matching problem

Now we introduce the definition of the *higher-order matching problem*.

Let  $M : \tau$  and  $N : \tau$  be closed  $\lambda$ -terms where  $N : \tau$  does not contain any unknown quantity. The equation  $M : \alpha = N : \alpha$ , where  $\alpha$  is a base type, is called *higher-order matching equation*.

Any type-respecting function  $\rho : \mathcal{E} \rightarrow \text{Cl}(\text{NF}(\Lambda_{\rightarrow}))$  is called a *concretisation of unknown quantities*. For any  $\lambda$ -term  $M$  the *result of the concretisation of its unknown quantities* is a term  $\rho(M)$  in which every unknown quantity  $x$  is substituted for by the term  $\rho(x)$ .

Please note that in the definition of the result of concretisation, no variable from concretisation gets bounded during the process of substitution.

The *higher-order matching problem* is a decision problem to ascertain whether for a given higher-order matching equation  $M = N$  exists a concretisation of unknown quantities  $\rho : \mathcal{E} \rightarrow \text{Cl}(\text{NF}(\Lambda_{\rightarrow}))$  such that  $\text{NF}(\rho(M)) = N$ . Such a concretisation is called a *solution of the equation  $M = N$* . The *matching problem of the order  $n$*  is a higher-order matching problem where instances may have unknown quantities symbols of the order at most  $n$ .

Throughout the rest of the text, we use the term *matching problem* to denote the *higher-order matching problem*.

### 2.3 The linear interpolation problem

Now we introduce the problem that is equivalent to the matching problem as we show later.

We say the matching equation  $M = N$  is an interpolation equation iff  $M = xM_1 \dots M_n$  where  $x \in \mathcal{E}$  and for each  $i$  term  $M_i$  is closed (in particular it has no occurrence of an unknown quantity).

Conceptually, it is simpler to solve interpolation equations are much simpler than arbitrary since we look for exactly one term.

The *linear interpolation problem* is a problem to decide whether there exists for a finite set  $E$  of interpolation equations of the shape  $[xM_1 \dots M_n = N]$ , where  $x$  appears in all equations and is the only unknown quantity in  $E$ , a concretisation of unknown quantities  $\rho : \{x\} \rightarrow \text{Cl}(\text{NF}(\Lambda_{\rightarrow}))$  that is a solution for each  $e \in E$ . We call such a concretisation a *solution of the interpolation set  $E$* . As only one value is relevant in such a concretisation, we sometimes use the name *solution of an interpolation set* in order to refer the one value. The maximum in  $E$  of the number of occurrences in terms  $N$  is denoted  $\text{MaxRes}(E)$ . The *interpolation problem of the order  $n$*  is an interpolation problem where instances may have unknown quantities symbols of the order at most  $n$ .

In the next section, we show the relation between the just formulated problems.

## 3 The matching problem and the interpolation problem

We start with the simpler reduction. We show that interpolation problems may be solved using algorithm for the higher-order matching problem.

**Fact 3.1** *Assume that there exists an algorithm  $\mathcal{A}$  that solves the higher-order matching problem. Then, there exists an algorithm  $\mathcal{B}$  that solves the interpolation problem.*

*Proof.* Let  $E = \{e_1, \dots, e_m\}$  be an instance of the interpolation problem that for each  $i$  has  $e_i = [x^i M_1^i \dots M_{n_i}^i = N^i]$ . If we have an algorithm  $\mathcal{A}$  for the higher-order matching problem we can solve such an instance by introducing a new constant  $Z : \tau_1 \rightarrow \dots \rightarrow \tau_m \rightarrow \alpha$ , where  $\tau_i$  is a type of term  $N^i$  in the interpolation equation  $e_i$ , and  $\alpha$  is a type constant, and then applying the algorithm  $\mathcal{A}$  to the following instance of the matching problem  $Z(x^1 M_1^1 \dots M_{n_1}^1) \dots (x^m M_1^m \dots M_{n_m}^m) = ZN^1 \dots N^m$ . It is straightforward that each solution of the instance gives a solution of the collection  $E$  and that each solution of  $E$  is a solution of the instance, too.

It is worth mentioning that in this construction the order of the problem to be solved does not change.

Now we show the reverse reduction.

**Fact 3.2** *Given an algorithm  $\mathcal{A}$  that solves the interpolation problem, we can construct an algorithm  $\mathcal{B}$  that solves the higher-order matching problem.*

*Proof.* Let  $M = N$  be a matching problem instance. We present an interpolation problem instance which has a solution iff the instance  $M = N$  has one.

The constructed set  $E$  of interpolation equations contains elements

$$\begin{aligned} x\lambda y_1 \dots y_m.M' &= N \\ x\lambda y_1 \dots y_m.Z &= Z \end{aligned} \tag{1}$$

where  $Z$  is a fresh constant of a suitable type and  $M' = M[y_1/x_1, \dots, y_m/x_m]$ . Additionally,  $\{x_1, \dots, x_m\} = \text{FV}(M) \cap \mathcal{E}$  and  $\{y_1, \dots, y_m\}$  is a set of fresh local variables of suitable types.

( $\Rightarrow$ ) Given a solution  $\rho : \mathcal{E} \rightarrow \Lambda_{\rightarrow}$  of the equation  $M = N$  such that  $\rho(x_i) = t_i$  the solution  $\rho' : \{x\} \rightarrow \text{Cl}(\text{NF}(\Lambda_{\rightarrow}))$  of the constructed interpolation problem instance is  $\rho'(x) = \lambda y. yt_1 \dots t_m$ . The proof that  $\rho'$  is a solution of the collection (1) is straightforward but tedious.

( $\Leftarrow$ ) Let  $\rho'$  be a solution of our instance of the interpolation problem and let  $\rho'(x) = t$ . Assume further,  $t$  is in normal form. The shape of the second equation implies that  $\text{NF}(t\lambda y_1 \dots y_m.Z) = Z$ . Therefore  $t = \lambda y. Z$  or  $t = \lambda y. yu_1 \dots u_m$ . The first case is impossible, because the result of the first interpolation equation cannot be  $Z$  ( $Z$  is fresh).

Now it is easy to see that  $\rho : \mathcal{E} \rightarrow \Lambda_{\rightarrow}$  such that

$$\rho(x_i) = \text{NF}((\lambda y. u_i)\lambda y_1 \dots y_m.M')$$

is a solution of the instance  $M = N$  of the matching problem.

*Remark.* The order raises by two in the previous construction.

At the end we get as a consequence of Fact 3.1 and Fact 3.2

**Theorem 1.** *The problem of linear interpolation is recursively equivalent to the higher-order matching problem.*

## 4 Decidability of a fragment of the fifth order interpolation problem

At the beginning, we introduce a definition that shall help us to present the fragment of the interpolation problem we are dealing with.

**Definition 1** *We say a  $\lambda$ -term  $\lambda x_1 \dots x_m.M$  is unsophisticated iff for each occurrence in  $M$  of a term of the form  $x_i N_1 \dots N_k$  and for each  $j \in \{1, \dots, k\}$  none of  $x_l$ , where  $l \in \{1, \dots, m\}$ , appears in  $N_j$ .*

*Each variable, constant, or application is an unsophisticated term. The linear interpolation problem with unsophisticated arguments is a linear interpolation problem, where we impose additional restriction on the form of an instance — for each  $e \in E$  of the form  $e = [xM_1 \dots M_n = N]$  where  $M_i$  for  $i \in \{1, \dots, n\}$  is an unsophisticated term.*

*A solution of a set  $E$  of such equations is called a solution of the interpolation with unsophisticated arguments set  $E$ .*

We restrict additionally our attention to the instances of the fifth order.

We show further that if for a given set  $E$ , there exists any solution then there exists one in some recursively dependent on  $E$  set of  $\lambda$ -terms.

Our construction consists of two steps. In the first one, we restrict the set of solutions in such a way that we know the set of types of their subterms and the set of constants they are built up of. In the second one, we narrow the already got class so that we know the depth of solutions.

Before we present next results, we introduce some notation. For an interpolation equations set  $E = \{e_1, \dots, e_n\}$  where  $e_i = [xM_1^i \dots M_k^i = N^i]$  we put  $\text{MaxRes}(E) = \max_{i \in \{1, \dots, n\}} |N^i|$  and  $\text{SumRes}(E) = \sum_{i=1}^n |N^i|$ .

### 4.1 Accessible terms

We define here a class of solutions the elements of what intuitively do not have unnecessary subtrees in their Böhm trees. The idea of the notions is taken from [Dow93].

Let  $e = [xM_1 \dots M_n = N]$  is an interpolation equation,  $t$  its solution, and  $\gamma$  an occurrence in the term  $t$ . We say *the occurrence  $\gamma$  in the term  $t$  is accessible wrt. equation  $e$*  iff the term  $\text{NF}((t[Z \rightarrow \gamma])M_1 \dots M_n)$ , where  $Z$  is a fresh constant of a suitable type, has an occurrence of the constant  $Z$ .

The set of all accessible wrt. equation  $e$  occurrences is denoted by  $\text{Acc}(t, e)$ .

Intuitively, an accessible subterm cannot be lost during the reduction of the term  $tM_1 \dots M_n$ .

We say that *an occurrence  $\gamma$  is accessible wrt. the set  $E$  of equations* iff it is accessible wrt. at least one of  $e \in E$ . If  $\gamma$  is not accessible we say it is *inaccessible*. We denote the set of all occurrences accessible wrt. the set  $E$  of equations by  $\text{Acc}(t, E)$ .

*A path  $\gamma$  is accessible* iff the occurrence that corresponds to its end is accessible. If such a path is not accessible we call it *inaccessible*.

The solution  $t$  of a set  $E$  is accessible iff each occurrence  $\gamma$  in the solution either is accessible or is an occurrence of a constant  $Z$  of a suitable type.

We fix a simple types indexed set  $\mathcal{P}$  of fresh constants. Additionally, we assume each  $\mathcal{P}_\sigma$  has exactly one element. On next pages, we are considering the lambda calculus extended by the set. The set we call *the set of fillings* and each its element a *filling*.

We say the solution  $t$  of the set  $E$  of equations has a good filling iff each inaccessible occurrence in  $t$  is an occurrence of a filling.

*Remark.* Obviously, solutions with a good filling are accessible.

**Theorem 2.** *If there exists a solution of a set  $E$  of interpolation equations then there exists a solution of the set which has a good filling.*

*Proof.* Let  $E$  be a set of interpolation equations and  $t$  its solution. We construct the term  $\hat{t}$  that has a good filling.

Let  $\gamma$  be an inaccessible wrt. the set  $E$  occurrence in  $t$ . We graft the filling  $Z$  of a suitable type at  $\gamma$ . The resulting term  $t'$  is a solution. Using this argumentation, we eliminate one by one inaccessible occurrences of non-trivial terms. At last, we get a solution that has a good filling.

**Fact 4.1** *The set of constants occurring in the solution with a good filling is contained in the set  $C = \bigcup_{i=1}^n \text{Const}(N^i) \cup \mathcal{P}$ .*

*Proof.* We have two possibilities — an occurrence  $\gamma$  of a constant  $D$  is accessible wrt. some equation  $e_i$ , an occurrence  $\gamma$  of  $D$  is inaccessible. In the first case, we show by the induction on the sum of the length of all the reductions from the term  $t[Z \rightarrow \gamma]M_1^i \dots M_k^i$  to its normal form that occurrences of the constant  $Z$  correspond to suitable occurrences of the symbol  $D$ . As  $\gamma$  is an accessible path,  $\text{NF}(t[Z \rightarrow \gamma]M_1^i \dots M_k^i)$  contains an occurrence of  $Z$ , so  $\text{NF}(tM_1^i \dots M_k^i)$  contains  $D$  and then  $D$  is a constant in  $N^i$ .

In the second case, we get by the definition of an accessible solution that  $D \in \mathcal{P}$ .

**Fact 4.2** *For each interpolation set  $E$ , there exists a closed on subtypes, finite and recursively dependent on  $E$  set of types  $\mathcal{T}$  such that for each accessible solution  $t$  of  $E$  the set of types of symbols occurring in  $t$  is included in  $\mathcal{T}$ .*

*Proof.* The set of types is  $\mathcal{T} = \bigcup_{i=1}^n \text{SubTyp}(\text{Typ}(N^i)) \cup \text{SubTyp}(\{\sigma\})$  This set is finite, recursively dependent on  $E$  and closed on subtypes. We show our fact by the induction wrt. depth of occurrence of  $v$  that the type of each symbol  $v$  occurring in  $t$  is in  $\mathcal{T}$ . When the depth is zero the symbol  $v$  cannot occur. When the depth is greater than zero — the symbol  $v$  is either a variable or a constant. If it is a variable then it must be declared in an argument of active variable. If it is a constant it must occur in one of  $N^i$ .

*Remark.* The above reasoning concerns arbitrary interpolation problem.

In the following sections, we achieve the third needed property — boundedness of the length of a path in a solution. We make the assumption that if we have a solution in hand then the solution is accessible.

## 4.2 Active and passive symbols

Here we introduce the notion of active and passive symbols and show a bound on the number of occurrences of the latter ones. The next section is concerned with active symbols.

**Definition 2** Let  $t = \lambda \mathbf{x}.t'$  be a  $\lambda$ -term. We define the set  $\text{Act}(t)$  of the active occurrences — each occurrence of  $x_i$  from  $\mathbf{x}$  is in  $\text{Act}(t)$ ; for each subterm  $zt_1 \dots t_k$  such that  $z$  is in  $\text{Act}(t)$ , if  $t_i = \lambda \mathbf{y}.t'_i$  then each occurrence of  $y_j$  from  $\mathbf{y}$  in  $t'_i$  belongs to  $\text{Act}(t)$ . We say an occurrence  $\gamma$  in a term  $t$  is passive, when it does not belong to  $\text{Act}(t)$ . The set of all passive occurrences is denoted by  $\text{Pas}(t)$ . If an occurrence of a symbol  $z$  is in  $\text{Act}(t)$  then we say the symbol is active else we say the symbol is passive.

**Fact 4.3** For each interpolation equations set  $E$  and its accessible solution  $t$ , if an occurrence  $\gamma \in \text{Pas}(t)$  is an occurrence of a symbol  $v$  then for any equation  $e = [xM_1 \dots M_k = N] \in E$  during normalisation of  $tM_1 \dots M_k$  the variable  $v$  is not substituted for.

*Proof.* Simple induction on the depth of an occurrence of passive variable on the path  $\gamma$ .

**Fact 4.4** For each interpolation equations set  $E$ , its accessible solution  $t$ , and equation  $e \in E$  of the form  $e = [xM_1 \dots M_k = N]$ , there exists one-to-one map  $f_e : \text{Pas}(t) \cap \text{Acc}(t, e) \rightarrow \text{Occur}(N)$ .

*Proof.* Let  $\gamma \in \text{Pas}(t) \cap \text{Acc}(t, e)$  be a path. As  $\gamma \in \text{Pas}(t)$  during normalisation of  $tM_1 \dots M_k$ , the occurring at  $\gamma$  symbol  $v$  is not substituted for. As  $v \in \text{Acc}(t, e)$ , there exists nonempty set  $A_\gamma$  of occurrences in  $\text{NF}(tM_1 \dots M_k)$  that appeared due to  $\gamma$ . Let us put  $f_e(\gamma)$  so that  $f_e(\gamma) \in A_\gamma$ . Each such function is one-to-one.

**Corollary 4.5** For any interpolation equation set  $E$  with an accessible solution  $t$ , we have  $|\text{Pas}(t) \cap \text{Acc}(t, E)| \leq \text{SumRes}(E)$ .

*Proof.* Let  $E = \{e_1, \dots, e_n\}$  where  $e_i = [xM_1^i \dots M_k^i = N^i]$ . We have

$$|\text{Pas}(t) \cap \text{Acc}(t, E)| \leq \sum_{i=1}^n |\text{Pas}(t) \cap \text{Acc}(t, e_i)|$$

By Fact 4.4, we get that the last number is less or equal

$$\sum_{i=1}^n |\mathbf{f}_{e_i}(\text{Pas}(t) \cap \text{Acc}(t, e_i))| \leq \sum_{i=1}^n |N^i| \leq \text{SumRes}(E)$$

This completes the proof.

Now we show a bound on the number of occurrences of active variables. First, for the so called variables from the main abstraction then for the variables from side abstractions.

## 4.3 Variables from the main abstraction

Now we show that if there exists a solution of a set  $E$  then there exists a solution  $\hat{t}$  such that paths in  $\hat{t}$  have the number of occurrences of variables from the main abstraction bounded.

We begin with the definition of *variables from the main abstraction*.

**Definition 3** Let  $t = \lambda \mathbf{y}.t'$  be a solution of a set  $E$ . Variables from  $\mathbf{y}$  are called variables from the main abstraction. The rest of active variables are called variables from side abstractions.

We prove a fact that will help us to understand what happens for deep paths.

**Fact 4.6** Let  $E = \{e_1, \dots, e_n\}$  be a set of equations with unsophisticated arguments and  $t$  its solution. Additionally, let  $e_i = [xM_1^i \dots M_k^i = N^i]$  for each  $i$ . Then for each  $i$  there exists  $p$  such that if for a certain path  $\gamma$  in  $t$ , which is accessible wrt. one of  $e_j$ , the variable  $y_i$  from the main abstraction has on  $\gamma$  more than  $\text{MaxRes}(E)$  occurrences then

- the term  $M_i^j$  has the form  $\lambda \mathbf{z}.z_p C_1 \dots C_r$  where none of  $z_i$  in  $\mathbf{z}$  occurs in  $C_l$  for  $l \in \{1, \dots, r\}$ ,
- for each occurrence  $\delta$  of the variable  $y_i$ , where the term at  $\delta$  has the form  $y_i D_1 \dots D_m$ , the next occurrence on  $\gamma$  is an occurrence of  $D_p$ .

*Proof.* The term  $M_i^j$  has the form  $\lambda \mathbf{z}.v C_1 \dots C_r$  where  $v$  is a constant or a local variable. The only local variables that can occur here (as  $M_i^j$  is closed) are variables from  $\mathbf{z}$  so the only bad case is the case when  $v$  is a constant. However,  $v$  cannot be a constant. Indeed, if  $v$  is a constant then observation of the normalisation of  $tM_1^j \dots M_k^j$  leads to the conclusion that  $\text{NF}(tM_1^j \dots M_k^j) (= N^j)$  has more than  $\text{MaxRes}(E)$  occurrences (which is impossible). The last claim is true since the term  $M_i^j$  is substituted on  $y_i$  and none of accessible occurrences of  $y_i$  (there are more than  $\text{MaxRes}(E)$  such occurrences) can disappear.

**Definition 4** For a given  $E$  and  $t$  in the situation from the previous fact, the number  $p$  from the previous fact is denoted by  $\text{Dir}(y_i)$ .

We introduce the notion of a compact wrt. variables from the main abstraction solution. For such solutions we can give a bound on the number of occurrences of variables from the main abstraction.

**Definition 5** Let  $E$  be a set of interpolation equations. We say that an accessible solution  $t$  is compact wrt. variables from the main abstraction iff for each variable  $y$  the variable can occur at most  $\text{MaxRes}(E)$  times on any path in  $t$ .

**Theorem 3.** If the set  $E$  of interpolation equations with unsophisticated arguments has an accessible solution  $t$  then there exists a compact wrt. variables from the main abstraction solution  $\hat{t}$ .

Moreover, we can assume the set of symbols occurring in  $\hat{t}$  is contained in the set of symbols occurring in  $t$ .

*Proof.* Given  $t$ , we construct  $\hat{t}$ . Induction on the sum of lengths of paths  $\gamma$  such that there exists a variable  $y$  from the main abstraction with more than  $\text{MaxRes}(E)$  occurrences on  $\gamma$ . If the sum is equal to zero then we put  $\hat{t} = t$ . Obviously, such  $\hat{t}$  meets our requirements. If the sum is greater than zero then we construct an accessible solution  $t'$  which is a copy of  $t$  except one path that



has cut off one occurrence of some variable from the main abstraction. This implies the sum of lengths of described above  $\gamma$ 's is lower for  $t'$  so the induction hypothesis applied to the new solution gives desired  $\hat{t}$ .

Now we construct  $t'$ . First we introduce some notation. Let  $E = \{e_1, \dots, e_n\}$  and for each  $i$  equation  $e_i = [xM_1^i \dots M_k^i = N^i]$ . Let  $y$  be a variable from the main abstraction,  $\gamma$  be a path that is  $\text{MaxRes}(E) + 1$ st occurrence of  $y$  and let  $\delta$  be a  $\text{MaxRes}(E)$ th occurrence of  $y$  on  $\delta$ . The term at  $\gamma$  has the form  $yD_1 \dots D_m$  and the term at  $\delta$  has the form  $yD'_1 \dots D'_m$ . Moreover,  $D'_{\text{Dir}(y)} = \lambda z_1 \dots z_r. D''$ . At last, we can put

$$t' = t[\text{NF}(D_{\text{Dir}(y)}z_1 \dots z_r) \rightarrow \gamma]$$

The term  $t'$  is a copy of  $t$  except for one path that has cut off one occurrence of the variable  $y$  from the main abstraction. In order to prove  $t'$  is an accessible solution it is enough to prove that  $\text{NF}(\hat{t}M_1^i \dots M_k^i) = N^i$  for each  $i \in \{1, \dots, n\}$  and that  $t'$  is accessible.

If  $\gamma$  is inaccessible wrt.  $e_i \in E$  in  $t'$  then conclusion is obvious. If  $\gamma$  is accessible wrt.  $e_i \in E$  then by Fact 4.6 reductions in the term at  $\gamma$  may look as follows

$$yD_1 \dots D_k \rightarrow_{\beta} M_j^i D_1 \dots D_k \rightarrow_{\beta}^* D_{\text{Dir}(y)} C_1 \dots C_r \rightarrow_{\beta}^* D_{\text{Dir}(y)}^{\bullet} [C_1/v_1 \dots C_r/v_r] \quad (2)$$

where  $M_j^i = \lambda z. z_{\text{Dir}(y)} C_1 \dots C_r$  and  $D_{\text{Dir}(y)} = \lambda v_1 \dots v_r. D_{\text{Dir}(y)}^{\bullet}$ . As  $M_j^i$  is unsophisticated, terms  $C_1, \dots, C_r$  are substituted for corresponding variables in  $D'_{\text{Dir}(y)}$  (see the term at the occurrence  $\delta$ ). So the given by the rules

1. substitute arguments  $M_1^i, \dots, M_k^i$
2. do reduction (2)
3. do for  $\delta$  a reduction analogous to (2)

reduction strategy for  $tM_1^i \dots M_k^i$  leads to the same result as the given by the rules

1. substitute arguments  $M_1^i, \dots, M_k^i$
2. do analogous to (2) reduction for  $\delta$

strategy for  $t'M_1^i \dots M_k^i$ . This is enough to get  $\text{NF}(t'M_1^i \dots M_k^i) = N^i$ .

We get accessibility since paths in  $t'$  have they counterparts in  $t$ .

*Remark.* The proof for variables from the main abstraction involves only the assumption about unsophisticated arguments.

Now we modify our solution so that we can estimate the number of variables from side abstractions.

#### 4.4 Variables from side abstractions

We show that we can assume the number of variables from side abstractions is dependent on the number of occurrences of variables from the main abstraction.

**Definition 6** *We say that a compact wrt. variables from the main abstraction solution  $t$  is compact wrt. variables from side abstractions iff on each accessible path  $\gamma$  of the solution and for each variable  $y$  from side abstraction declared in the term  $\lambda x. \lambda y. \lambda z. D$  variable  $y$  occurs on  $\gamma$  in  $D$  at most  $\text{MaxRes}(E)$  times.*

We can prove the constructibility of such solutions now.

**Theorem 4.** *If the set  $E$  of fifth order interpolation equations has a compact wrt. variables from the main abstraction solution  $t$  then there exists a compact wrt. variables from side abstractions solution  $\hat{t}$ .*

*Moreover, we can assume the set of symbols occurring in  $\hat{t}$  is contained in the set of symbols contained in  $t$ .*

*Proof.* Induction on the number of occurrences of variables  $y$  from the side abstractions that are declared on a certain path  $\gamma$  in a subterm of the form

$$\lambda x. \lambda y. \lambda z. D \tag{3}$$

such that  $y$  on  $\gamma$  and in  $D$  occurs more than  $\text{MaxRes}(E)$  times.

– If there are no such occurrences then the term  $t$  is compact wrt. variables from the side abstraction.

– If there exists such  $y$  then let us take  $\text{MaxRes}(E) + 1$ st occurrence  $\delta$  of  $y$  in  $D$  and on  $\gamma$ . The term at  $\delta$  has the form  $yD_1 \dots D_m$  and at most one of  $D_i$  for  $i \in \{1, \dots, m\}$  is accessible. The last claim follows from the fact that, as there are more than  $\text{MaxRes}(E)$  occurrences of  $y$ , the only term that may be substituted for the variable in the reduction of the left hand side of the equation wrt. which  $\gamma$  is accessible is a second order term that reduces to  $\lambda z_1 \dots z_m. z_i$ , otherwise the normal form would have more than  $\text{MaxRes}(E)$  occurrences.

Of course, one of  $D_i$  is accessible (otherwise  $\delta$  is inaccessible). Now we can put  $t' = t[D_k \rightarrow \delta]$  where  $D_k$  is the only accessible term. One can see  $t'$  is a term such that if it is a solution then we can apply the induction hypothesis to it.

Now we show that  $t'$  is a solution. If  $\delta$  is not accessible wrt.  $e_i \in E$  then  $\delta$  does not affect the normal form and  $t'$  is a solution of such an equation, too. If  $\delta$  is accessible wrt.  $e_i \in E$  then we reduce left-hand side according to the strategy consisting in following exactly one residuum of a term  $D$  and holding all the reductions inside the term  $D$ . By accessibility of  $D_k$ , there exists among terminal terms of such strategies at least one term  $t''$  for which, after replacing one of copies of  $D_k$  by a suitable constant and further normalising, we get occurrences of the constant in the resulting normal form.

In  $t''$ , we have a term substituted for  $y$ . The normal form of the term may have one of the two shapes

$$\lambda z_1 \dots z_m. v P_1 \dots P_r \tag{4}$$

where  $v$  is a symbol that is not substituted for (if something was substituted then  $t''$  would not be a terminal term of one of our strategies), or

$$\lambda z_1 \dots z_m. z_i. \tag{5}$$

The case (4) cannot happen, because then on the path  $\delta$ , it would occur more than  $\text{MaxRes}(E)$  occurrences of  $v$  and the occurrences could not disappear during the reduction due to accessibility of  $\delta$ . So the only possibility is (5). Moreover, accessibility of  $D_k$  implies that  $\lambda z_1 \dots z_m. z_k$ . Because this reasoning applies always to the situation when we use  $D_k$ , we get the same result by replacing the term at  $\delta$  by  $D_k$ . This completes the proof.

*Remark.* The assumption that the equations are of the fifth order was used only in the proof of Theorem 4.

To formulate and prove next facts, we have to introduce some notation. We let  $\text{Act}_{\text{main}}(t, \gamma)$  denote the set of occurrences of variables from the main abstraction on the path  $\gamma$  and let  $\text{Act}_{\text{side}}(t, \gamma)$  denote the set of occurrences of variables from side abstractions on the path  $\gamma$ .

Let  $\gamma$  be a path and  $\delta$  an occurrence on it. Moreover, let  $\delta$  be an occurrence of a variable  $x$  from the main abstraction. Let the term grafted on  $\delta$  has the form  $xt_1 \dots t_k$  and let the next occurrence on  $\gamma$  be an occurrence of a term  $t_i$  of the form  $t_i = \lambda \mathbf{y}.t'_i$ . In such a situation we let  $\text{Decl}(t, \gamma, \delta)$  denote the set of occurrences of variables that are declared in  $\mathbf{y}$  and let  $\text{Decl}(t, \gamma, \delta, y_1)$ , where  $y_1$  is one of variables from  $\mathbf{y}$ , denote the set of occurrences of the variable  $y_1$  in  $t'_i$ .

A last, we let  $\text{VarDec}(t, \delta)$  denote the set of variables contained in the list  $l$ , where  $l$  is the first component of the last label  $(l, v)$  on  $\delta$ . Intuitively, it is the set of variables declared at the beginning of the subterm of  $t$  that occurs at the end of  $\delta$ .

The maximal number of arguments in types of subterms of the term  $t$  is denoted by  $\text{MaxSub}(t)$ .

**Corollary 4.7** *Let  $E$  be an arbitrary set of interpolation equations and  $t$  its solution that is compact wrt. variables from side abstractions. If the number of occurrences on an accessible path  $\gamma$  of variables from the main abstraction is bounded by  $K$  then the number of occurrences of variables from side abstractions on the path is bounded by  $K \text{MaxSub}(t) \text{MaxRes}(E)$ .*

*Proof.* We estimate the number of occurrences of variables from side abstractions

$$|\text{Act}_{\text{side}}(t, \gamma)| = |\bigcup_{\delta \in \text{Act}_{\text{main}}(t, \gamma)} \text{Decl}(t, \gamma, \delta)| \leq \sum_{\delta \in \text{Act}_{\text{main}}(t, \gamma)} \sum_{y \in \text{VarDec}(t, \delta)} |\text{Decl}(t, \gamma, \delta, y)|$$

According to the definition of a compact wrt. variables from side abstractions solution, for each variable from side abstractions the number of its occurrences on an accessible path is not greater than  $|N^j|$ , where  $j$  is the number of the equation wrt. which  $\gamma$  is accessible. This implies estimation

$$\sum_{\delta \in \text{Act}_{\text{main}}(t, \gamma)} \sum_{y \in \text{VarDec}(t, \delta)} |\text{Decl}(t, \gamma, \delta, y)| \leq \sum_{\delta \in \text{Act}_{\text{main}}(t, \gamma)} \sum_{y \in \text{VarDec}(t, \delta)} |N^j|$$

Further, we get easily

$$\sum_{\delta \in \text{Act}_{\text{main}}(t, \gamma)} \sum_{y \in \text{VarDec}(t, \delta)} |N^j| \leq K \text{MaxSub}(t) \text{MaxRes}(E)$$

and this is our result.

Now we are ready to draw final conclusions of the section.

**Corollary 4.8** *Let  $E$  be a set of interpolation equations with unsophisticated arguments. There exist recursively dependent on  $E$  — a simple types indexed set of constants  $C$  such that for each type the set of constants of the type is finite, a finite set of types  $\mathcal{T}$ , a number  $g$  — such that if there exists a solution of  $E$  then there exists a solution of  $E$  such that all the types of subterms belong to  $\mathcal{T}$ , all the constants are from  $C$ , and the length of paths is less or equal  $g$ .*

*Proof.* Let  $E = \{e_1, \dots, e_n\}$  be a set of interpolation equations with unsophisticated arguments of the form  $e_i = [xM_1^i \dots M_k^i = N^i]$  and let  $t$  be a solution of  $E$ . We construct  $C = \bigcup_{i=1}^n \text{Const}(N^i) \cup \mathcal{P}, \mathcal{T}$  as in Fact 4.2,

$$g = k \text{MaxRes}(E) + k (\text{MaxRes}(E))^2 \text{MaxSub}(t) + \text{SumRes}(E) + 1$$

( $\text{MaxSub}(t)$  is by Fact 4.2 recursively dependent on  $E$ .)

Combining Fact 2, Theorem 3 and Theorem 4 we get a solution  $\hat{t}$  that meets our requirements. This last is assured by Fact 4.1, Fact 4.2 and Corollary 4.7, Corollary 4.5.

## 4.5 Decidability

Now we are ready to draw the conclusion of the decidability.

**Theorem 5.** *The fifth order interpolation problem with unsophisticated arguments is decidable.*

*Proof.* We get the algorithm using the algorithm in Fact 4.8 generating on the results of the latter the set of  $\lambda$ -terms built of the data in the results and then extensively searching the space of terms. The algorithm *stops*, because the procedures used stop. The *correctness* of the algorithm is obvious as when we end successfully we have a solution in hand. When the algorithm stops saying that the solution does not exist it means that there is no solution since existence of a solution would contradict the Corollary 4.8. This completes the proof.

*Remark.* The algorithm constructs a solution that may contain fillings which were absent from the original problem. Since the fillings are used to replace other terms, each used filling has at least one corresponding term. When necessary we can replace a filling by such a term.

## References

- [Bar84] H.P. Barendregt, *The lambda calculus, its syntax and semantics*, 2 ed., North Holland, 1984.
- [Bar92] H.P. Barendregt, *Lambda calculi with types*, Handbook of Logic in Computer Science (S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, eds.), vol. 2, Oxford Science Publications, 1992, pp. 117–309.
- [Dow93] G. Dowek, *Third order matching is decidable*, Annals of Pure and Applied Logic (1993).
- [GH78] B. Lang G. Huet, *Proving and applying program transformations expressed with second order patterns*, Acta Informatica (1978), no. 11, 31–55.
- [Hue76] G. Huet, *Résolution d'équations dans les langages d'ordre 1, 2, ...,  $\omega$* , Ph.D. thesis, Université Paris VII, 1976.
- [Pad96] Vincent Padovani, *Filtrage d'ordre supérieur*, Ph.D. thesis, Université Paris VII, January 1996.
- [Sch96] Aleksy Schubert, *Linear interpolation for the higher-order matching problem*, Tech. Report TR 96-16 (237), Institute of Informatics Warsaw University, December 1996.
- [Wol89] D.A. Wolfram, *The clausal theory of types*, Ph.D. thesis, University of Cambridge, 1989.