A self-dependency constraint in the simply typed lambda calculus

Aleksy Schubert[⋆]

Institute of Informatics
Warsaw University
ul. Banacha 2
02–097 Warsaw
Poland

Abstract. We consider terms of simply typed lambda calculus in which copy of a subterm may not be inserted to the argument of itself during the reduction. The terms form wide class which includes linear terms. We show that corresponding variant of the dual interpolation problem i.e. problem in which all expressions can be restricted to terms of this kind. Thus the model for this kind of expressions can admit fully abstract semantics.

Classification: semantics, logic in computer science.

1 Introduction

The higher-order matching problem (HOM) consists in solving certain equations in the simply typed λ-calculus. The equations have the form $M \doteq N$ where the unknowns occur only in the term M. HOM is a restricted version of the higher-order unification problem (HOU) which has several applications in the field of automated theorem proving. Unfortunately, the higher-order unification problem has been proved undecidable [Gol81]. Despite this sad result HOU has attracted a significant attention and is used as a tool in automated theorem proving (e.g. in Isabelle [Pau89]) and higher-order logic programming (e.g. in λ-Prolog [MN88]). As a special case of HOU the higher-order matching problem can be applied in various situations in these areas (see e.g. [Har96]).

The higher-order matching problem has also connections with semantics of the simply typed λ-calculus [Sta82]. It is strongly connected with the problem of λ-definability. The problem of λ-definability is to decide whether a given function from some function space over a finite base can be defined by a λ-term. The problem of λ-definability is undecidable which corresponds to the fact that there is no way to precisely describe a fully-abstract model for the simply typed λ-calculus [Loa01]. The higher-order matching problem defines a stronger notion of definability for which it is not known if it is decidable. Semantical investigations in simply typed lambda calculus have some practical impact as they are strongly connected to the area of program transformations (see e.g. [dMS01]).

[⋆] This work was partly supported by KBN grant 7 T11C 028 20; part of the work was conducted while the author was a winner of the fellowship of The Foundation for Polish Science (FNP).

The higher-order matching problem has a long history. The problem was posed by G. Huet in his PhD thesis [Hue76]. It was considered in some older papers ([Wol89,Bax76]), but it gained more attention in the last years. V. Padovani presented a new approach to higher-order matching. He proved that higher-order matching is decidable without restriction on the order but with a restriction on the term N — it must be a single constant of a base type [Pad95a]. In addition, R. Loader presented [Loa] the undecidability of unrestricted higher-order matching, but for β-equality (the standard formulation requires $\beta\eta$-equality). This was supplemented by results that take into account restrictions on domains of solutions [dG00,DW02,SSS02].

This paper expands significantly the technique used in [Pad95a] and proposes a new interesting class of terms. Informally, a term M is self-independent when in each self-independent context and for each subterm N of the term if the evaluation of the subterm N is executed somewhere in the reduction then all its arguments may not contain a copy of N. For example the term

$$\lambda x : (\alpha \to \alpha) \to \alpha \to \alpha . x(\lambda z_1 . f(z_1, z_1))a$$

is self-independent since the only self-independent terms that can be substituted for x are the identity, and the constant term and these can only result in forgetting of most of the term of in copying a to z_1. Accordingly the term

$$\lambda x : ((\alpha \to \alpha) \to \alpha \to \alpha) \to (\alpha \to \alpha) \to \alpha . x(\lambda z_1 . z_1(z_1(a))f$$

is not self-independent, as in one if possible contexts (when the identity is substituted for x) the term f occurs in the argument of the first z_1. This class extends the class of linear terms. Moreover, presented proof gives a nice syntactical characterisation of this class. We present a proof of decidability for the dual interpolation problem with an additional restriction on solutions and instances — they must be self-independent terms. This result extends to higher-order matching for this kind of expressions.

2 Preliminaries

Assume we are given a set $\mathcal{B}$ of *base types.* Let $\mathcal{T}_{\mathcal{B}}$ be the set of all *simple types over* $\mathcal{B}$ defined as the smallest set containing $\mathcal{B}$ and closed on $\to$. We shall often omit the subscript $\mathcal{B}$ if $\mathcal{B}$ is clear from the context or unimportant.

In this paper we limit ourselves to the case where $\mathcal{B}$ is a singleton $\{\iota\}$.

The set of simply typed terms is defined based on the set of pre-terms. The set $\lambda^*_{\to}$ of *simply typed pre-terms* contains an infinite, countable set of variables V, a countable set of constants C, and a map $T : V \cup C \to \mathcal{T}_{\mathcal{B}}$ that indicates types of symbols. We assume that there exists an infinite number of variables of a type α. This set is also closed on the application and λ-abstraction operations. We will usually write $s : \tau$ to denote the fact that a pre-term s has a type τ. As usual, we deal with α-equivalence. The symbol $[s]_\alpha$ denotes the α-equivalence class of s. The set $\lambda_{\to}$ is defined as a quotient of $\lambda^*_{\to}$ by α-equivalence. The elements of $\lambda_{\to}$ are denoted usually by $M, M_1, \ldots, N, N_1 \ldots$ etc. as well as the elements of $\lambda^*_\rightarrow$ are denoted by $s, s_1, \ldots, t, t_1, \ldots$ etc. The notion of a closed term is defined as usual, similarly the set $\mathrm{FV}(M)$ of free variables in M. The symbol Const_M denotes the set of constants occurring in M.

We denote by $T(\tau, C)$ the set of all closed terms of type τ built-up of constants from the set C. The notion of *sorted set* is of some usefulness here. The family of sets $T^C = \{T(\tau, C)\}_{\tau \in \mathcal{T}_\mathcal{B}}$ is an example of a sorted set. We adopt standard $\in$ notation to sorted sets.

The term Const_M denotes the set of constants that occur in M. This notation is extended to sets (and other structures) of terms.

The *order* of a type τ, denoted by $\mathrm{ord}(\tau)$ is defined inductively as

- $\mathrm{ord}(\iota) = 0$ for $\iota \in \mathcal{B}$;
- $\mathrm{ord}(\tau_1 \rightarrow \tau_2) = \max(\mathrm{ord}(\tau_1) + 1, \mathrm{ord}(\tau_2))$.

The notion of order extends to terms and pre-terms. We define $\mathrm{ord(M)} = \mathrm{ord}(\tau)$ or $\mathrm{ord(t)} = \mathrm{ord}(\tau)$, where τ is the type of M or t respectively.

The notion of β-reduction ($\rightarrow^*_\beta$) is defined as a congruent extension of the relation $(\lambda x.M)N \rightarrow_\beta M[x := N]$ where $M[x := N]$ denotes substitution of N for x with usual renaming of bounded identifiers. We sometimes write substitutions in the prefix mode as in $S(N)$.

The notion of $\beta\eta$-reduction ($\rightarrow^*_{\beta\eta}$) is defined as a congruent extension of the above-mentioned $\rightarrow_\beta$ relation supplied with an additional rule $\lambda x.Mx \rightarrow_\eta M$ where M has no occurrence of x. In this paper, we deal with terms in β-normal, η-long form. The β-normal, η-long form for a term M is denoted by $\mathrm{NFL}(M)$.

We call *an instance of the higher-order matching problem* each pair of simply typed λ-terms $\langle M, N\rangle$. We usually denote them as $M \doteq N$.

A solution of such an instance is a substitution S such that $S(M) =_{\beta\eta} N$.

We often restrict ourselves to the case when N has no free variables. This restriction is not essential.

Definition 1 (the higher-order matching problem)
The higher-order matching problem is a decision problem — given an instance $M \doteq N$ of the higher-order matching problem whether there exists a solution of $M \doteq N$.

An *interpolation equation* is a pair of terms, usually written $xN_1 \cdots N_k \doteq N$, such that x is the only free variable in the left-hand side of the equation.

Let $\langle E, E'\rangle$ of interpolation equations such that there exists a variable x which occurs free in the left-hand side of each equation in $E \cup E'$. We call $\langle E, E'\rangle$ *an instance of the dual interpolation problem.* We also call $\langle E, E'\rangle$ a *dual set.*

A solution of such an instance is a term P such that for each equation $[xN_1 \cdots N_k \doteq N] \in E$ we have $PN_1 \cdots N_k =_{\beta\eta} N$, and for each equation $[xN_1 \cdots N_k \doteq N] \in E'$ we have $PN_1 \cdots N_k \neq_{\beta\eta} N$.

Definition 2 (the dual interpolation problem)
The dual interpolation problem is a decision problem — given an instance $\langle E, E'\rangle$ of the dual interpolation problem whether there exists a solution of $\langle E, E'\rangle$.

The dual interpolation problem and the higher-order matching problem are connected in the following way:

Theorem 1. *The problem of higher-order matching reduces to the dual interpolation problem.*

Proof. See [Pad95b,Pad96] or [Sch01].

2.1 Self-independent terms

We define self-independent terms by means of a marked reduction:

$$\begin{array}{l}(\lambda x.M)N \to_{\beta\#} M[x := N] \\ (M)^{\#}N \to_{\beta\#} MN.\end{array}$$

where the substitution $[x := N]$ is defined so that $(P)^{\#}[x := N] = (P[x := N])^{\#}$.

A *self-dependency* is a term of the form $(M)^{\#}N_1 \cdots N_k$ where either M or one of N_i contains the mark $\#$. For each term M we define the set $\mathtt{M}^{\#}(M)$ as

$$\mathtt{M}^{\#}(M) = \{C[x^{\#}N_1 \cdots N_k] \mid M = C[xN_1 \cdots N_k]\}$$

($C[\cdot]$ denotes a term with a single hole usually called context, substitution for $[\cdot]$ is not capture avoiding.)

All terms of a base type are self-independent. We say that a term M is self-independent if for each sequence $N_1, \ldots, N_n$ of self-independent terms the outmost-leftmost reduction of each term $M'N_1 \cdots N_n : \iota$ where ι is a base type and $M' \in \mathtt{M}^{\#}(M)$ does not include a term with a subterm being a self-dependency.

3 Tools and definitions

3.1 Addresses in terms

Let t be a pre-term. We say that a sequence γ of natural numbers is an *address in* t iff

- $\gamma = \varepsilon$ (the empty sequence) or
- $\gamma = i \cdot \gamma'$ where $t = \lambda x_1 \ldots x_m.u_0 u_1 \cdots u_n$ together with $0 \leq i \leq n$ and γ' is an address in u_i.

The set of all addresses in t is written as $\mathrm{Addr}(t)$. The prefix order on addresses is written as $\gamma \preceq \gamma'$. The strict version is denoted as $\gamma \prec \gamma'$.

We say that a subterm u is *pointed out* by an address γ in t iff

- $u = t'$ where $t = \lambda x_1 \ldots x_m.t'$, the term t' does not begin with λ and $\gamma = \varepsilon$,
- or $u = t'$ where $t = \lambda x_1 \ldots x_m.u_0 u_1 \cdots u_n$, and $\gamma = i \cdot \gamma'$ together with $0 \leq i \leq n$ and $\gamma'(u_i) = t'$.

We write $\gamma(t)$ to denote u.

A *graft* of a term u in t at an address γ is a term t' defined as

- u iff $\gamma = \varepsilon$,
- $\lambda x_1 \ldots x_m.u_0 u_1 \cdots u_i' \cdots u_n$ iff $\gamma = i \cdot \gamma'$ and $t = \lambda x_1 \ldots x_n.u_0 u_1 \cdots u_n$ and u_i' is a graft of u in u_i at γ'.

We denote t' as $t[\gamma \leftarrow u]$. We sometimes use the notation $t[c \leftarrow u]$ where c is a constant. This means that the term u is grafted on every occurrence of the constant c.

Let C be a set of fresh constants. We say that a pre-term u is a C-pruning of a pre-term t iff

– $u = t$,
– or $u = u'[\gamma \leftarrow c]$ where $c \in C$ and u' is a C-pruning of t.

Let t' and t'' be C-prunings of a term t. We say that $t' \preceq t''$ iff $\mathrm{Addr}(t') \subseteq \mathrm{Addr}(t'')$. This pre-order may be viewed as an extension of the prefix order on sequences. Thus, we denote this relation by the same symbol. The strict version of the order is denoted by $\prec$. The above-mentioned notions easily extend to λ-terms, sets of λ-terms and sets of pre-terms.

3.2 Matrix notation

Every finite sequence of terms of the same type is called a *column*. We denote columns by symbols like V, W etc. The symbol V^i denotes the i-th element of the column V. The *height* of a column V is the length of the sequence V. It is denoted by $|V|$. As all the elements of a column V have the same type, we may use the notion of a *type of the column* V. When V has a type τ then we denote this fact as $V : \tau$. We say that V is a *constant column* when all its elements are the same.

Every finite sequence of terms is called a *row*. We denote rows by symbols like R, Q etc. The symbol R_i denotes the i-th element of the row R. The *width* of a row R is the length of the sequence R and is denoted as $|R|$. The operation of concatenation of columns W, W' or rows R, R' is denoted as WW' or RR'.

Every finite sequence of columns is called a *matrix*. We denote matrices by symbols like $\overline{\mathtt{M}}, \overline{\mathtt{N}}$.

The symbol $\overline{\mathtt{M}}_i$ denotes the i-th column of $\overline{\mathtt{M}}$. The symbol $\overline{\mathtt{M}}_i^j$ denotes the j-th element of the i-th column of $\overline{\mathtt{M}}$. Similarly, $\overline{\mathtt{M}}^i$ denotes a row defined as $(\overline{\mathtt{M}}^i)_j = \overline{\mathtt{M}}_j^i$.

Let $\overline{\mathtt{M}}$ be a matrix and W a column of a base type. We define a *set of equations with the matrix* $\overline{\mathtt{M}}$ *and results* W as

$$x\overline{\mathtt{M}}_1^1 \cdots \overline{\mathtt{M}}_n^1 \doteq W^1, \quad \cdots, \quad x\overline{\mathtt{M}}_1^m \cdots \overline{\mathtt{M}}_n^m \doteq W^m.$$

We denote the set of equations as $[x\overline{\mathtt{M}} \doteq W]$. We extend the notation $\mathrm{NFL}(M)$ to vectors, rows and matrices: $\mathrm{NFL}(VM_1 \cdots M_n)$, $\mathrm{NFL}(MR)$ and $\mathrm{NFL}(M\overline{\mathtt{M}})$. The notation Const_M extends to matrices and vectors and means the set of constants occurring in the supplied matrix or vector.

3.3 Approximations

We say that a pair W, W' of columns is constant iff W is constant as a column and $W'^i \neq W^1$ for all i.

An approximation of a pair of columns W_1, W_2 in a dual set $\mathcal{E} = \langle [x\overline{\mathtt{M}}_1 \doteq W_1], [x\overline{\mathtt{M}}_2 \doteq W_2] \rangle$ of equations for the solution M is any pair of columns $\widetilde{W_1}, \widetilde{W_2}$ of the heights $|W_1|, |W_2|$ such that

- there exists an $\{a, b, 0, \ldots, l\}$-pruning M' of M such that for each equation $[x\overline{\mathtt{M}}^k_{i,1} \cdots \overline{\mathtt{M}}^k_{i,r} \doteq W^k_i] \in \mathcal{E}$ we have $\mathrm{NFL}(M'\overline{\mathtt{M}}^k_{i,1} \cdots \overline{\mathtt{Mi}}^k_{i,r}) = \widetilde{W^k_i}$,
- for $c \in \{a, b\}$ if c occurs in $\widetilde{W^i_j}$ then $\widetilde{W^i_j} = c$,
- for each constant $c \in \mathbb{N}$ if some $\widetilde{W^i_j} = c$ then there exists $\gamma \succ \varepsilon$ and k, l such that $\gamma(\widetilde{W^k_l}) = c$ (we say c is *guarded*).

We denote by $\mathbb{N}_{\widetilde{W_1}\widetilde{W_2}}$ the set $\mathrm{Const}_{\widetilde{W_1}\widetilde{W_2}} \cap (\mathbb{N} \cup \{a, b\})$. Note that $\widetilde{W^k_1}$ is a $\{a, b, 0, \ldots, l\}$-pruning of W^k_1 while $\widetilde{W^k_2}$ may not be a $\{a, b, 0, \ldots, l\}$-pruning of W^k_2.

For the sake of notational convenience we assume that for each approximation, the set $\mathbb{N}_{\widetilde{W}\widetilde{W'}} \cap \mathbb{N}$ is an initial connected subset of $\mathbb{N}$, e.g. $\{0, 1, 2\}$ or $\{0, 1\}$.

Let $\widetilde{W_1}, \widetilde{W'_1}$ and $\widetilde{W_2}, \widetilde{W'_2}$ be approximations of a pair of columns W, W'. We write $\widetilde{W_1}, \widetilde{W'_1} \preceq \widetilde{W_2}, \widetilde{W'_2}$ when for each k we have $\widetilde{W^k_1} \preceq \widetilde{W^k_2}$ and $\widetilde{W'^k_2} \preceq \widetilde{W'^k_2}$. It is easily verified that $\trianglelefteq$ is a pre-order. The strict version of the order is denoted as $\triangleleft$.

We say that for an approximation $\widetilde{W}, \widetilde{W'}$ of W, W' the term $\widetilde{W'}^k$ *meets* W'^k iff for each k there exist $s_a, s_b, s_0, \ldots s_l$ such that

$$W'^k = \widetilde{W'}^k[a \leftarrow s_a, b \leftarrow s_b, 0 \leftarrow s_0, \ldots, l \leftarrow, s_l].$$

A pair of columns W, W' is *trivial* iff for each i, j we have $W^i = W'^j = c$, where c is either a or b.

Let $\langle [x\overline{\mathtt{M}} \doteq W], [x\overline{\mathtt{M'}} \doteq W'] \rangle$ be a dual set and M its solution. We say that $\widetilde{W}, \widetilde{W'}$ *is a minimal approximation* of W, W' iff there is no non-trivial $\widetilde{W_1}\widetilde{W'_1} \triangleleft \widetilde{W}, \widetilde{W'}$.

We say that a term M' is a *minimal pruning for minimal approximation* $\widetilde{W}, \widetilde{W'}$ iff $\widetilde{W}, \widetilde{W'}$ is a minimal approximation of W, W' and there is no $M'' \prec M'$ such that $\mathrm{NFL}(M''\overline{\mathtt{M}}) = \widetilde{W}$ and $\mathrm{NFL}(M''\overline{\mathtt{M'}}) = \widetilde{W'}$.

Let W, W' be a pair of columns with a type $\tau_1 \to \cdots \to \tau_n \to \iota$. We say that a pair of columns V, V' is a splitting pair of columns for W, W' iff there exist terms $M_1, \ldots, M_n$ such that the pair of columns $\mathrm{NFL}(VM_1 \cdots M_n)$, $\mathrm{NFL}(V'M_1 \cdots M_n)$ is an approximation of W, W'.

3.4 Accessibility

Let $E = [x\overline{\mathtt{M}} \doteq W]$ be a set of interpolation equations. We say that an address γ is *accessible* in a term M wrt. E iff

- γ is an address in M,
- there is an equation $[x\overline{\mathtt{M}}^i_1 \cdots \overline{\mathtt{M}}^i_n \doteq W^i] \in E$ such that

$$\mathrm{NF}(M[\gamma \leftarrow c]\overline{\mathtt{M}}^i_1 \cdots \overline{\mathtt{M}}^i_n) \tag{1}$$

  has an occurrence of c, where c is a fresh constant of a base type.

We say that an address is *totally accessible* iff for each equation in E the condition (1) holds.

We say that an address is *totally head accessible* iff for each equation in E we have $\mathrm{NF}(M[\gamma \leftarrow c]\overline{\mathtt{M}}^{i}_{1} \cdots \overline{\mathtt{M}}^{i}_{n}) = c$, where c is a fresh constant. Without loss of generality we may assume that c is a constant of the base type.

Let $\langle E, E' \rangle$ be a dual set and M its solution. We say that an occurrence γ is *totally head accessible* wrt. the dual set iff it is totally head accessible wrt. E and E'.

3.5 Observational equivalence

We introduce a notion of observational equivalence. This notion is very closely related to the dual interpolation problem. Roughly speaking, solutions of an instance of the dual interpolation problem form an equivalence class in this relation.

Let $\mathcal{R} = \{\mathcal{R}_\tau\}_{\tau \in \mathcal{T}}$ be an indexed family of sets containing λ-terms, and satisfying conditions

1. all terms in $\mathcal{R}$ are in β-normal, η-long form,
2. for each term $M \in \mathcal{R}$, there exists an α-representant s_M of M such that for each subterm t of the pre-term s_M we have $[t]_\alpha \in \mathcal{R}$.

Such a set is called an *observable.*

The notion of an observable gives rise to a variation of the dual interpolation problem and the higher-order matching problem.

An instance of the dual interpolation problem for an observable $\mathcal{R}$ is a pair of sets $\mathcal{E} = \langle E, E' \rangle$ of interpolation equations such that there exists a variable x which occurs free in the left-hand side of each equation in $E \cup E'$ and all right-hand sides of $E \cup E'$ belong to $\mathcal{R}$.

The dual interpolation problem for an observable $\mathcal{R}$ is to decide whether an instance of the dual interpolation problem for the observable $\mathcal{R}$ has a solution.

Definition 3 (the higher-order matching problem for an observable)
An instance of the higher-order matching problem for an observable $\mathcal{R}$ *is a pair of simply typed* λ*-terms* $\langle M, N \rangle$ *where* $N \in \mathcal{R}$.

The higher-order matching problem for an observable $\mathcal{R}$ *is to decide whether a given instance of the higher-order matching problem for the observable* $\mathcal{R}$ *has a solution.*

The notion of an observable is also a base for a pre-order and an equivalence relation which, in turn, allow us to define a semantic structure for the simply typed λ-calculus.

Definition 4 (observational pre-order)
For each observable $\mathcal{R}$ *we define an* observational pre-order with respect to $\mathcal{R}$ *as the relation* $\sqsubseteq_{\mathcal{R}}$ *on* λ*-terms such that*

$M \sqsubseteq_{\mathcal{R}} M'$ *iff* M *and* M' *have both the type* $\sigma \to \tau$*, and for each sequence of terms* $N_1, \ldots, N_n$ *with appropriate types and* $n \geq 1$ *we have that if* $\mathrm{NFL}(MN_1 \cdots N_n) \in \mathcal{R}$ *then* $MN_1 \cdots N_n =_{\beta\eta} M'N_1 \cdots N_n$ *or*
M *and* M' *are both of a base type and if* $\mathrm{NFL}(M) \in \mathcal{R}$ *then* $M =_{\beta\eta} M'$

Definition 5 (observational equivalence)
For each observable $\mathcal{R}$ we define an observational equivalence with respect to this observable *as the relation on closed terms $\approx_{\mathcal{R}}$ such that*

$$M \approx_{\mathcal{R}} M' \textit{ iff } M \sqsubseteq_{\mathcal{R}} M' \textit{ and } M' \sqsubseteq_{\mathcal{R}} M$$

More details concerning this equivalence are in [Sch01]. In particular, we have the following theorem

Theorem 2. *For each solvable instance $\mathcal{E}$ of dual interpolation for an observable $\mathcal{R}$ there exists an equivalence class of $\approx_{\mathcal{R}}$ such that all its elements are solutions of $\mathcal{E}$.*

For each class A of the relation $\approx_{\mathcal{R}}$ there exists an instance $\mathcal{E}$ of the dual interpolation problem for the observable $\mathcal{R}$ such that terms from A solve $\mathcal{E}$.

There is an algorithm which for a type $\tau = \tau_1 \to \cdots \tau_n \to \iota$ and complete sets of representants of $\approx_{\mathcal{R}}$ classes for types $\tau_1, \ldots, \tau_n$ generates a set $\mathcal{C}$ of instances such that for each $\approx_{\mathcal{R}}$ class A in the type τ there is an instance $\mathcal{E}_A$ such that terms of A are the only solutions of $\mathcal{E}_A$.

Proof. Easy conclusion from [Sch01].

3.6 Transferring terms

The proof in this paper is based on the following schema: We define *transferring terms*. We show that each solvable dual interpolation instance has a solution of this form. As the form is quite simple, we are able to enumerate the terms and this gives a proof that the dual interpolation problem is decidable.

This section contains the crucial definition of transferring terms. The definition is a little bit tedious, but we shall explain some of its elements later on in this section.

Definition 6 (transferring terms)
Let $n, m \in \mathbb{N}$ and let C be a set of constants. A closed term $M : \tau_1 \to \cdots \to \tau_p \to \iota$ is an (n, m, C)-transferring term *iff*

1. *$M = \lambda y_1 \ldots y_p.M'$ and M' is a term over C without any occurrence of y_i where $i = 1, \ldots, p$, or*
2. *$M = \lambda y_1 \ldots y_p.f M_1 \cdots M_k$, where*
   - *$f \in C$ has a type $\sigma_1 \to \cdots \to \sigma_k \to \iota$, and*
   - *$M_i = \lambda z_1 \ldots z_r.M'_i$ where M'_i does not begin with λ, and*
   - *$\lambda y_1 \ldots y_p.M'_i$ are (n_i, m, C_i)-transferring terms, and*
   - *$n_i < n$, and*
   - *$C_i = C \cup \{z_1, \ldots, z_r\}$, or*
3. *$M = \lambda y_1 \ldots y_p.y_i M_1 \cdots M_k[a \leftarrow N_a \boldsymbol{y}, b \leftarrow N_b \boldsymbol{y}, 0 \leftarrow N_0 \boldsymbol{y}\boldsymbol{x_0}, \ldots, l \leftarrow N_l \boldsymbol{y}\boldsymbol{x_l}]$, where*
   - *$\boldsymbol{y} = y_1, \ldots, y_p$;*
   - *each M_i is a closed term over $C \cup \{a, b, 0, \ldots, l\}$;*
   - *each constant $0, \ldots, l$ occurs only once in $y_i M_1 \cdots M_k$;*
   - *N_a and N_b are (n_a, m_a, C)-transferring and (n_b, m_b, C)-transferring respectively;*

- *N_j is (n_j, m_j, C)-transferring for each j;*
- *$n_a, n_b \leq n$;*
- *$m_a + m_b \leq m$ and $m_a, m_b \neq m$;*
- *$n_j < n$ and $m_j \leq n$.*

Sometimes when (n, m, C) are unimportant or clearly seen from the context, we shorten the name to transferring*.*

Suppose that case (3) is abridged so that there are no constants from $\mathbb{N}$. The resulting term M may be presented as

$$M = \lambda \boldsymbol{y}.C_{y_k}[y_k M_1 \cdots (\lambda \boldsymbol{z} z_j \boldsymbol{z'}.M_i) \cdots M_r]$$

Consider $\lambda \boldsymbol{z} z_j \boldsymbol{z'}.M_i$. In this term, none of occurrences of z_j in M_i is in a subterm beginning with y_l from $\boldsymbol{y}$.

When this constraint is applied, the number of occurrences of variables from $\boldsymbol{y}$ in an (n, m, C)-transferring term M is bounded by m.

This simple picture is contaminated by the presence of constants from $\mathbb{N}$. We allow these constants to occur in subterms of M beginning with variables y_l from $\boldsymbol{y}$. This adds some flexibility and, consequently, expressive power to our terms. This flexibility is restricted, though. We have to pay for blocks that declare these variables with coins kept in n.

We sometimes treat informally this particular form of λ-terms as a syntactic form. In this syntactic notation, all substitutions from case (3) are actually present in a term. Formal development of this approach would involve a formalism similar to explicit substitutions formalisms.

4 Transferring terms and dual interpolation

Here is a theorem that relates the dual interpolation problem to transferring representatives. This is the central theorem of the paper.

Theorem 3. *Let $\mathcal{E}$ be an instance of the dual interpolation problem. If M is a solution of $\mathcal{E}$ then there exist $n, m \in \mathbb{N}$ together with a set of constants C and an (n, m, C)-transferring term M' such that M' is a solution of $\mathcal{E}$, too.*

Moreover, n, m, C depend recursively on $\mathcal{E}$.

In order to prove this theorem we need two lemmas that enable some cleaning of sometimes excessively complicated terms.

4.1 Skipping unimportant variables

The induction step in the proof of Theorem 3 consists in splitting a dual set and finding transferring solutions for the results of the split. These solutions are based on a term M that solves the whole set at the very beginning. This term is too complicated to be a compact solution of the split sets. During the process of construction of transferring solutions for them, we have to compact M in several different ways. This section is devoted to the major step of the compactification.

Below, we use C-prunings for $C = \{a, b\} \cup \mathbb{N}$. Thus we shorten the name C-pruning to pruning here. We impose a constraint on the shape of prunings performed on solutions. *Constants from $\mathbb{N}$ may occur only once.*

First of all, we have to determine what we want to skip.

Definition 7 (an unimportant occurrence)
Let $M = \lambda \boldsymbol{y}.y_i M_1 \cdots M_k$ be a term that solves a dual set $\mathcal{E} = \langle [x\overline{\mathtt{M}} \doteq W], [x\overline{\mathtt{M}} \doteq W'] \rangle$, and let M' be a pruning of M that gives a minimal approximation of W, W'. An occurrence $\gamma \cdot 0$ of a variable z in M' is unimportant *iff*

- *γ is totally head accessible, and*
- *there is $\gamma' \neq 0$ such that $\gamma \cdot \gamma'$ is an occurrence of a variable bound on γ and $\gamma \cdot \gamma'$ is totally head accessible.*

Note that unimportant occurrence has always the shape $\gamma \cdot 0$. We say that an unimportant occurrence $\gamma \cdot 0$ is maximal if there is no unimportant occurrence $\gamma \cdot \gamma' \cdot 0$ where $\gamma' \succ \varepsilon$.

Lemma 1. *Let $M = \lambda \boldsymbol{y}.z M_1 \cdots M_k$ be a term that solves a set of dual equations $\mathcal{E} = \langle [x\overline{\mathtt{M}} \doteq W], [x\overline{\mathtt{M}'} \doteq W'] \rangle$, where W, W' is a non-constant pair of columns. If $M' = \lambda \boldsymbol{y}.z \widehat{M_1} \cdots \widehat{M_k}$ is a minimal pruning of M that gives minimal approximation of W, W' wrt. $\mathcal{E}$, then*

- *either there exists j such that $\overline{\mathtt{M}}_j, \overline{\mathtt{M}'}_j$ is a pair of splitting columns for W, W',*
- *or there is an address γ in M that points to a constant $f \in \mathrm{Const}_M$ that is totally head accessible.*

Proof. The general idea is that we get rid of unimportant occurrences which can be evaluated. This is the place where the definition of self-independent terms plays crucial role.

Proof of Theorem 3:
The aforementioned lemma allows to prove the main theorem of the paper. This is done by splitting dual interpolation instances. The way the split is done is controlled by the shape of minimal approximation for the hypothetical solution. Details are in appendix. □

5 Decidability

We have to provide yet another characterisation of solutions for the higher-order matching problem in the case of self-independent terms. This characterisation relies strongly on some constructions for observables.

Definition 8 (pruned observable)
Let $\mathcal{R}$ be an observable. The observable $\mathcal{R}_C$ is called a pruning of $\mathcal{R}$ *if it contains all the C-prunings of terms from $\mathcal{R}$.*

Note that $\mathcal{R} \subseteq \mathcal{R}_C$ for each C.

Definition 9 (observable for terms)
Let T be a set of terms. $\mathcal{R}^T$ is a minimal observable containing T.

Note that if $T \subseteq \mathcal{R}$ then $\mathcal{R}^T \subseteq \mathcal{R}$.

Definition 10 (pseudo transferring terms)
*Let $\mathcal{R}$ be a finite observable, $n, m \in \mathbb{N}$, and C be a set of constants. A term $M : \tau_1 \to \cdots \to \tau_p \to \iota$ is a pseudo (n, m, C)-*transferring term *for the observable $\mathcal{R}$ iff*

1. *$M = \lambda y_1 \ldots y_p.M'$ and M' is a term over constants from C without any occurrence of y_i,*
2. *$M = \lambda y_1 \ldots y_p.f M_1 \cdots M_k$, where $f \in C$ has a type $\sigma_1 \to \cdots \to \sigma_k \to \iota$ and $\lambda y_1 \ldots y_p.M_i$ are pseudo (n_i, m, C_i)-transferring terms for the observable $\mathcal{R}$ with $n_i < n$ and $C_i = C \cup \{z_1, \ldots, z_r\}$ where $M_i = \lambda z_1 \ldots z_r.M'_i$ and M'_i does not begin with λ.*
3. *$M = \lambda y_1 \ldots y_p.y_i M_1 \cdots M_k[a \leftarrow N_a\boldsymbol{y}, b \leftarrow N_b\boldsymbol{y}, z_0 := N_0\boldsymbol{y}, \ldots, z_l := N_l\boldsymbol{y}]$, where*
   - *$\boldsymbol{y} = y_1, \ldots, y_p$;*
   - *$z_0, \ldots, z_l$ have order less than the order of y_i;*
   - *$\lambda z_{i_1} \ldots z_{i_k}.M_i$ are pseudo $(n_i, m_i, C \cup \{a, b\})$-transferring terms for the observable $\mathcal{R}_{\{a,b\}}$, where n_i is the maximal number of equations in a dual set $\mathcal{E}$characterising a single equivalence class as in Theorem 2, and m_i is the maximal total size of right-hand sides in such an $\mathcal{E}$*
   - *N_a and N_b are pseudo (n_a, m_a, C)-transferring and pseudo (n_b, m_b, C)-transferring respectively for the observable $\mathcal{R}$;*
   - *T is a set of terms such that $\mathcal{R}^T \subsetneq \mathcal{R}$;*
   - *for each j the term N_j is pseudo (n_j, m_j, C)-transferring for the observable $\mathcal{R}^T$.*
   - *$n_a, n_b \leq n$;*
   - *$m_a + m_b \leq m$ and $m_a, m_b \neq m$;*
   - *n_j is the number of equations in a dual set characterising an equivalence class of observational equivalence for $\mathcal{R}^T$ and m_j is the total size of right-hand sides in the aforementioned dual set.*

Sometimes when (n, m, C) are unimportant or clearly seen from the context, we shorten the name to pseudo transferring.

We have the following theorem:

Theorem 4. *If a dual interpolation set $\mathcal{E}$ for an observable $\mathcal{R}$ has an (n,m,C)-transferring solution M then it has a pseudo (n,m,C)-transferring solution, too.*

We are now ready to conclude with the following theorem.

Theorem 5. *The dual interpolation problem for self-independent terms is decidable.*

Proof. The decision procedure consists in enumerating all possible pseudo transferring terms for the type of the head variable in an instance of the dual interpolation problem. This may be done recursively according to the definition of pseudo transferring terms.

As an easy corollary we obtain.

Theorem 6. *Higher-order matching for self-independent terms is decidable.*

References

[Bax76] Lewis D. Baxter. *The Complexity of Unification*. PhD thesis, University of Waterloo, 1976.

[dG00] Philippe de Groote. Linear Higher-Order Matching Is NP-Complete. In Leo Bachmair, editor, *Proceedings of 11th International Conference on Rewriting Techniques and Applications*, number 1833 in LNCS, pages 127–140. Springer-Verlag, 2000.

[dMS01] Oege de Moor and Ganesh Sittampalam. Higher-order matching for program transformation. *Theoretical Computer Science*, 269(1–2):135–162, 2001.

[DW02] Daniel J. Dougherty and Tomasz Wierzbicki. A decidable variant of higher order matching. In Sophie Tison, editor, *Proc. Thirteenth Intl. Conf. on Rewriting Techniques and Applications (RTA)*, pages 340–351, 2002.

[Gol81] Warren D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 3:225–230, 1981.

[Har96] John Harrison. HOL light: a tutorial introduction. In Mandayam Srivas and Albert Camilleri, editors, *Proceedings of the First International Conference on Formal Methods in Computer-Aided Design*, pages 265–269, 1996.

[Hue76] G. Huet. *Résolution d'Équations dans les Langages d'Ordre* $1, 2, \ldots, \omega$. PhD thesis, Université de Paris VII, 1976.

[Loa] Ralph Loader. Higher-order β matching is undecidable. Can be found at `http://homepages.ihug.co.nz/~suckfish/match/`.

[Loa01] Ralph Loader. Finitary PCF is not decidable. *Theoretical Computer Science*, 1-2(266):341–364, 2001.

[MN88] D. Miller and G. Nadathur. An overview of λ-Prolog. In R. Bowen and K. Kowalski, editors, *Proceedings of the Fifth International Logic Programming Conference/ Fifth Symposium on Logic Programming*. MIT Press, 1988.

[Pad95a] Vincent Padovani. Decidability of all minimal models. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs, International Workshop TYPES'95*, number 1158 in LNCS, pages 201–215. Springer Verlag, 1995.

[Pad95b] Vincent Padovani. On equivalence classes of interpolation equations. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, number 902 in LNCS, pages 335–349, 1995.

[Pad96] Vincent Padovani. *Filtrage d'Ordre superieur*. PhD thesis, Université Paris VII, January 1996.

[Pau89] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1989.

[Sch01] Aleksy Schubert. A note on observational equivalence in the simply typed lambda calculus. Technical Report TR 02-01 (266), Institute of Informatics, Warsaw University, 2001. Can be found at `http://www.mimuw.edu.pl/~alx/ftp-public/domains.ps.gz`.

[SSS02] Manfred Schmidt-Schauß and Klaus U. Schulz. Decidability of bounded higher-order unification. In *CSL 2002*, LNCS 2471, pages 522–536. Springer-Verlag, 2002.

[Sta82] Richard Statman. Completness, invariance and λ-definability. *Journal of Symbolic Logic*, 1(47), 1982.

[Wol89] David A. Wolfram. *The Clausal Theory of Types*. PhD thesis, University of Cambridge, 1989.

A Proofs

The proofs presented here are only to support the validity of the presented material. They will not be submitted as a part of the paper.

The following lemma allows us to state that absence of unimportant variables guarantees that we are able to construct a splitting column.

Lemma 2. *Let $M = \lambda \boldsymbol{y}.zM_1 \cdots M_k$ be a term such that M solves a dual set $\mathcal{E} = \langle [x\overline{\mathtt{M}} \doteq W], [x\overline{\mathtt{M}'} \doteq W'] \rangle$, where W, W' is a non-constant pair of columns. If $M' = \lambda \boldsymbol{y}.z\widehat{M_1} \cdots \widehat{M_k}$ is a minimal pruning of M that gives a minimal approximation of W, W' wrt. $\mathcal{E}$ and M' does not contain unimportant occurrences of any variable from $\boldsymbol{y}$, then terms $\widehat{M_i}$ do not contain occurrences of variables from $\boldsymbol{y}$.*

Proof. Existence of such a variable means that 0 is unimportant. Details follow.

Suppose, contrary to our claim, that $\gamma = p \cdot \gamma'$ is an address in M' of a term that starts with a variable $y_j \in \boldsymbol{y}$. W.l.o.g. we may assume that γ is a $\preceq$-minimal address of this kind. As this address is not unimportant,

1. either γ is not totally head accessible,
2. or the term pointed out by γ is totally head accessible and there is no $\gamma \cdot \gamma'$ totally head accessible.

Case (1). We replace the term at γ with a, b or one of constants from $\mathbb{N}$ so that the constraints on pruning and approximation are fulfilled. We obtain this way either an approximation which is less than the original one or a pruning which is less than M'. This contradicts our assumption that M' is a minimal pruning of M that gives a minimal approximation.

Case (2). In this case 0 is an unimportant occurrence. Details are left for the reader.

Proof of Lemma 1:
The proof is by induction on the number of variables in M' that do not play an essential role in constructing the minimal approximation of W, W'. These variables do not serve as places where a decision is made or places where some symbols are arranged for right-hand sides of equations. They may only produce some terms that are used later on. We can simulate this behaviour by replacing a corresponding variable with an appropriately modified solution of $\mathcal{E}$. This solution contains complete information about the term that will be used later on. A more detailed presentation goes hereafter.

The proof is by induction on the number p of unimportant occurrences of variables from $\boldsymbol{y}$ in M'.

Case $p = 0$. Lemma 2 implies that the head symbol z is either a constant $f \in \mathrm{Const}_M$ or a variable $y_j \in \boldsymbol{y}$. The last case implies immediately that $\overline{\mathtt{M}}^j$ is a splitting column with arguments $\widehat{M_1}, \ldots, \widehat{M_k}$.

Case $p > 0$. Let $k \cdot \gamma \cdot 0$ be a maximal address of an unimportant variable $z \in \boldsymbol{yx}$ in M' ($\boldsymbol{x}$ are declared on $k \cdot \gamma$). We have

$$\gamma(\lambda \boldsymbol{y}.\widehat{M_k}) = zM'_1 \cdots M'_r[\,\delta \leftarrow N_\delta \boldsymbol{yxx_\delta}] \tag{2}$$

where

- $M'_i = \lambda z_1 \dots z_{r_i}.\widetilde{M_i}$ are closed terms over constants Const_i, and $\widetilde{M_i}$ does not begin with λ;
- $\varepsilon(N_\delta)$ begins with a symbol from $\boldsymbol{yx}$ or with a constant, and δ is totally head accessible;
- $\boldsymbol{x_\delta}$ are declared in $zM'_1 \cdots M'_r$ on the path between γ and δ.

We have several cases according to the form of N_δ.

1. *The head symbol of N_δ is a constant.* The form of (2) implies that M can be presented as

$$M = C_M[z\overline{M'_1} \cdots \overline{M'_r}[\,\delta \leftarrow \overline{N_\delta}\boldsymbol{yxx_\delta}]]$$

where $M'_1, \dots, M'_r, N_\delta$ are corresponding prunings of $\overline{M'_1}, \dots, \overline{M'_r}, \overline{N_\delta}$ and $C_M[\bullet]$ is a suitable context. The term $\overline{N_\delta}$ begins with a constant. This term cannot be a constant term (with no occurrences of variables from $\boldsymbol{yxx_\delta}$), since W, W' is not a constant pair. This means that $\overline{N_\delta}$ begins with a function symbol. Similarly, the function symbol starts N_δ. This means that the address $\delta \cdot 0$ is a path to a totally head accessible constant function symbol.

2. *The head symbol of N_δ is $x \in \boldsymbol{yx}$.* We define a context $C_\gamma[\bullet]$ as follows: $C_\gamma[\gamma(M)] = M$. The term M_{new} is defined as

$$M_{\mathrm{new}} = C_\gamma[\overline{N_\delta}\boldsymbol{yx}R_1 \cdots R_p].$$

Let $x_i \in \boldsymbol{x_\delta}$ (with $x_i : \tau_1 \to \cdots \to \tau_r \to \iota$), we define

$$R_i = \lambda z_1 \dots z_r.C_\gamma[zM'_1 \cdots M'_r[c \leftarrow x_i z_1 \cdots z_r]]\boldsymbol{y}.$$

It is easily verified that M_{new} is a solution of $\mathcal{E}$. It is so because M is self-independent. Let

$$M'_{\mathrm{new}} = C_\gamma[N_\delta \boldsymbol{yx}(\lambda \boldsymbol{z_1}.c_1) \cdots (\lambda \boldsymbol{z_1}.c_p)]$$

with $c_1, \dots, c_p \in \{a, b\} \cup \mathbb{N}$ and the constants $c_1, \dots, c_p$ inserted so that conditions for pruning and approximation are met. Note that M'_{new} is a pruning of M_{new}.

It is not possible that only one constant in $\{c_1, \dots, c_p\}$ is totally head accessible, since otherwise γ is not the maximal occurrence of an unimportant symbol.

The term M'_{new} is a non-trivial pruning. Triviality of M'_{new} implies either triviality of M' or that $k \cdot \gamma$ is not a maximal address of an unimportant variable.

As M'_{new} is a non-trivial pruning, we have a minimal non-trivial pruning M''_{new} of M_{new} wrt. W, W'. The term M''_{new} has strictly less occurrences of unimportant variables than M'. Thus, we can apply the induction hypothesis and obtain that either some $\overline{\mathtt{M}}^j$ is a splitting column or that there exists an address γ' in M''_{new} which is an occurrence of a totally head accessible constant symbol f. If $\gamma' = k \cdot \gamma \cdot \widehat{\gamma} \cdot 0$ then we obtain an address γ'' of f in M' as $\gamma'' = k \cdot \gamma \cdot \gamma_c \cdot \widehat{\gamma} \cdot 0$ where $\gamma_c(zM'_1 \cdots M'_r) = c$. If $\gamma' \neq k \cdot \gamma \cdot \widehat{\gamma} \cdot 0$ then this γ' is also an occurrence of f in M'. $\square$

Proof of Theorem 3:
Let $\mathcal{E} = \langle E, E' \rangle = \langle [x\overline{\texttt{M}} \doteq W], [x\overline{\texttt{M}'} \doteq W'] \rangle$. We define the value n as the number of symbols in W, W', the value m as the number of equations in $\mathcal{E}$, the value h as the height of M, and the set C as the set of constants in $\mathcal{E}$.

The proof is by induction wrt. lexicographic order on triples $\langle n, m, h \rangle$.

If W, W' is a constant pair of columns, then $M' = \lambda \boldsymbol{y}.W^1$ is a transferring solution.

If $m = 1$, then $M' = \lambda \boldsymbol{y}.W^1$ is a transferring solution provided that $E \neq \emptyset$. If $E = \emptyset$ then $M' = \lambda \boldsymbol{y}.\mathrm{NFL}(M \overline{\texttt{M}'}^1_1 \cdots \overline{\texttt{M}'}^1_m)$ is a transferring solution.

If $h = 1$, then $M = \lambda \boldsymbol{y}.c$ for some constant c of a base type or $c \in \boldsymbol{y}$. In this case, $M' = M$.

If $n = 1$, then this reduces to the case when $m = 1$.

If $\langle n, m, h \rangle > \langle 1, 1, 1 \rangle$, then we have two subcases

1. $M = \lambda \boldsymbol{y}.f M_1 \cdots M_k$ where $f \in C$;
2. $M = \lambda \boldsymbol{y}.y_i M_1 \cdots M_k$ for some $y_i \in \boldsymbol{y}$.

Case $M = \lambda \boldsymbol{y}.f M_1 \cdots M_k$. Let $M_i = \lambda z^i_1 \ldots z^i_{r_i}.\widetilde{M_i}$ where $\widetilde{M_i}$ does not begin with λ. In this case, all terms in W begin with f:

$$W = \langle f B^1_1 \cdots B^1_k, \ldots, f B^p_1 \cdots B^p_k \rangle$$

and some terms of W', say in sub-column W'':

$$W'' = \langle f B''^1_1 \cdots B''^1_k, \ldots, f B''^{p'}_1 \cdots B''^{p'}_k \rangle.$$

We define columns $W_1, \ldots, W_k$ and $W'_1, \ldots, W'_k$ as $W_i = \langle B^1_i, \ldots, B^p_i \rangle$, and $W'_i = \langle B''^1_i, \ldots, B''^p_i \rangle$. We can define sets of equations $E_i = [x\overline{\texttt{M}} \doteq W_i]$ and $E'_i = [x\overline{\texttt{M}''} \doteq W'_i]$ ($\overline{\texttt{M}''}$ is a sub-matrix of $\overline{\texttt{M}'}$ corresponding to W''). It is easily verified that $\langle E_i, E'_i \rangle$ has a solution $\lambda \boldsymbol{y}.\widetilde{M_i}$. Let the value n_i be the number of symbols in $W_i W'_i$, the value m_i be the number of equations in $E_i \cup E'_i$ and $C_i = C \cup \{z^i_1, \ldots, z^i_{r_i}\}$. The number n_i is strictly less than the number of symbols n so by the induction hypothesis we have (n_i, m_i, C_i)-transferring solutions $\lambda \boldsymbol{y}.\widetilde{M'_i}$ of sets $\langle E_i, E'_i \rangle$. Thus, $\lambda \boldsymbol{y}.f M'_1 \cdots M'_k$ is a (n, m, C)-transferring solution of $\mathcal{E}$ where $M'_i = \lambda z^i_1 \ldots z^i_{r_i}.\widetilde{M'_i}$.

Case $M = \lambda \boldsymbol{y}.y_i M_1 \cdots M_k$ for some $y_i \in \boldsymbol{y}$.

The pair of columns W, W' has a minimal approximation. Let M' be a minimal pruning for the minimal approximation of W, W'. Lemma 1 implies that either $\overline{\texttt{M}}_j, \overline{\texttt{M}'}_j$ are splitting columns for W, W' or there exists a totally head accessible occurrence in M of a constant $f \in \mathrm{Const}_M$.

Case when certain $\overline{\texttt{M}}_j, \overline{\texttt{M}'}_j$ are splitting columns for W, W'. The definition of splitting columns says that there exist terms $P_1, \ldots, P_r$ such that $\mathrm{NFL}(\overline{\texttt{M}}_j P_1 \cdots P_r) = \widetilde{W}$ and $\mathrm{NFL}(\overline{\texttt{M}'}_j P_1 \cdots P_r) = \widetilde{W'}$ where $\widetilde{W}, \widetilde{W'}$ is an approximation of W, W'. We produce on the basis of $\mathcal{E}$ new dual sets of equations $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_0, \ldots, \mathcal{E}_l$. These sets are defined as $\mathcal{E}_c = \langle E_c, E'_c \rangle$ where

1. $E_a = \{[x\overline{\mathtt{M}}^k_1 \cdots \overline{\mathtt{M}}^k_m = W^k] \mid \mathrm{NFL}(\overline{\mathtt{M}}^k_j P_1, \ldots, P_r) = a\}$, and
   $E'_a = \{[x\overline{\mathtt{M}'}^k_1 \cdots \overline{\mathtt{M}'}^k_m = W'^k] \mid \mathrm{NFL}(\overline{\mathtt{M}'}^k_j P_1, \ldots, P_r) = a\}$,
2. $E_b = \{[x\overline{\mathtt{M}}^k_1 \cdots \overline{\mathtt{M}}^k_m = W^k] \mid \mathrm{NFL}(\overline{\mathtt{M}}^k_j P_1, \ldots, P_r) = b\}$, and
   $E'_b = \{[x\overline{\mathtt{M}'}^k_1 \cdots \overline{\mathtt{M}'}^k_m = W'^k] \mid \mathrm{NFL}(\overline{\mathtt{M}'}^k_j P_1, \ldots, P_r) = b\}$,
3. the definition of E_i, E'_i for $i \in \mathbb{N}$ is more complex and presented hereafter.

Let $[x\overline{\mathtt{N}} \doteq \widehat{W}] = E \backslash (E_a \cup E_b)$ and $[x\overline{\mathtt{N}'} \doteq \widehat{W'}] = E' \backslash (E'_a \cup E'_b)$. We have

$$\begin{aligned} \mathrm{NFL}(\overline{\mathtt{M}}^k_j P_1 \cdots P_r) &= \widehat{W}^k = C^k[n^k_1, \ldots, n^k_q] \\ \mathrm{NFL}(\overline{\mathtt{M}'}^k_j P_1 \cdots P_r) &= \widehat{W}'^k = C'^k[n^k_1, \ldots, n^k_{q'}] \end{aligned}$$

where C^k and C'^k are contexts (note that some C^k or C'^k may be simply $[\bullet]$). Define

$$\begin{aligned} \mathit{\Pi}^k(n^k_s) &= \{\gamma \mid \gamma \text{ points out on a constant } n^k_s \text{ in } \widehat{W}^k\}, \\ \mathit{\Pi}'^k(n^k_s) &= \{\gamma \mid \gamma \text{ points out on a constant } n^k_s \text{ in } \widehat{W'}^k\}, \end{aligned}$$

We establish sets of equations

$$\begin{aligned} E_n &= \{[x^n \overline{\mathtt{N}}^k_1 \cdots \overline{\mathtt{N}}^k_m O^k_1 \cdots O^k_{l'} \doteq \gamma(W^k)] \quad \mid \gamma \in \mathit{\Pi}^k(n) \text{ and } k \in \mathbb{N}\}. \\ E'_n &= \{[x^n \overline{\mathtt{N}'}^k_1 \cdots \overline{\mathtt{N}'}^k_m O'^k_1 \cdots O'^k_{l'} \doteq \gamma(W'^k)] \mid \gamma \in \mathit{\Pi}^k(n) \text{ and } k \in \mathbb{N} \\ &\qquad \text{and } \widetilde{W'}^k \text{ meets with } W'^k\}. \end{aligned} \tag{3}$$

where terms $O^k_{k'}$ for $k' \in \{1, \ldots, l'\}$ are defined as follows. Let γ_k be an occurrence of a constant n^k_s in M' that results in n^k_s in $\widehat{W}^k$ (or $\widehat{W}'^k$). Let

$$M'' = M[\gamma_k \leftarrow Z x_1 \cdots x_{l'}] \tag{4}$$

where Z is a fresh constant of a suitable type and $\{x_1, \ldots, x_{l'}\} = \mathrm{FV}(\gamma_k(M)) \backslash \boldsymbol{y}$. We have

$$\begin{aligned} \mathrm{NFL}(M'' \overline{\mathtt{N}}^k_1 \cdots \overline{\mathtt{N}}^k_m) &= C^k[B_1, \ldots, (Z\overline{O}^k_1 \cdots \overline{O}^k_{l'}), \ldots, B_q], \\ \mathrm{NFL}(M'' \overline{\mathtt{N}'}^k_1 \cdots \overline{\mathtt{N}'}^k_m) &= C'^k[B'_1 \ldots, (Z\overline{O}'^k_1 \cdots \overline{O}'^k_{l'}), \ldots, B'_{q'}], \end{aligned} \tag{5}$$

We can also mimic reduction steps as above in

$$\begin{aligned} \mathrm{NFL}(M \overline{\mathtt{N}}^k_1 \cdots \overline{\mathtt{N}}^k_m) &\to^*_{\beta\eta} C^k[B_1, \ldots, (M^{\#} \widehat{O}^k_1 \cdots \widehat{O}^k_{l'}), \ldots, B_q], \\ \mathrm{NFL}(M \overline{\mathtt{N}'}^k_1 \cdots \overline{\mathtt{N}'}^k_m) &\to^*_{\beta\eta} C'^k[B'_1 \ldots, (M^{\#} \widehat{O}'^k_1 \cdots \widehat{O}'^k_{l'}), \ldots, B'_{q'}]. \end{aligned} \tag{6}$$

At last we define $O^k_{k'}$ (and $O'^k_{k'}$) as

$$\begin{aligned} O^k_{k'} &= \overline{O}^k_{k'}[Z := M^{\#}], \\ O'^k_{k'} &= \overline{O}'^k_{k'}[Z := M^{\#}]. \end{aligned}$$

The dual sets $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_0, \ldots, \mathcal{E}_s$ are solvable by the terms $M, M, \widetilde{M}_0, \ldots \widetilde{M}_s$ respectively. We define $\widetilde{M}_i$ as

$$\lambda \boldsymbol{y} x_1 \ldots x_{l'}.\gamma_k(M).$$

The sets $\mathcal{E}_a, \mathcal{E}_b$ have both the number of equations and the number of symbols occurrences less than $\mathcal{E}$. In the sets $\mathcal{E}_0, \ldots, \mathcal{E}_l$, the size of right-hand sides is less than the size of WW' (constants from $\mathbb{N}$ are guarded). This means that we can apply the induction hypothesis and obtain transferring solutions $\dot{M}_a, \dot{M}_b, \dot{M}_0, \ldots \dot{M}_l$ of $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_0, \ldots, \mathcal{E}_l$. We define a term

$$\begin{aligned}\dot{M} = \lambda \boldsymbol{y}.y_j P_1 \cdots P_r[\ & a \leftarrow \dot{M}_a \boldsymbol{y}, \\ & b \leftarrow \dot{M}_b \boldsymbol{y}, \\ & 0 \leftarrow \dot{M}_0 \boldsymbol{y} \boldsymbol{x_0}, \\ & \ldots, \\ & l \leftarrow \dot{M}_l \boldsymbol{y} \boldsymbol{x_l}]\end{aligned} \tag{7}$$

where $\boldsymbol{x_i}$ are variables bound in the P_k in which i occurs as a constant.

The term $\dot{M}$ is a solution, because when we reduce it with left-most, outmost strategy applied in equations of $\mathcal{E}$, then we reach either $\dot{M}_a \overline{\mathtt{M}}^k$, $\dot{M}_b \overline{\mathtt{M}}^k$ or $C^k[(\dot{M}_{n_1^k} \cdots) \cdots (\dot{M}_{n_q^k} \cdots)]$. As $\dot{M}_a, \dot{M}_b, \dot{M}_0, \ldots \dot{M}_l$ are solutions of equations $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_0, \ldots, \mathcal{E}_l$, we immediately see that reached terms reduce further to appropriate subterms of right-hand sides W, W'. Note that so defined $\dot{M}$ is (n, m, C)-transferring solution of $\mathcal{E}$.

Case when there exists an occurrence of a constant f in M which is totally head accessible. This case is technically similar to the case when $M = \lambda \boldsymbol{y}.f M_1 \cdots M_k$ The only difficulty concerns free variables that occur in $\gamma(M)$ where γ is the intervening occurrence of f in M. We can get rid of them using the trick employed in (4). Details are left for the reader.

□

Proof of Theorem 4:
Induction on lexicographically ordered pairs $\langle o, h, m\rangle$ where o is the order of the type of M, h is the height of M and m is the number of terms in $\mathcal{R}$.

If $o = 0$ then $\mathcal{E} = \langle E, E'\rangle$ may have only one equation in E and the equation has the form $x \doteq M$. In this case, M is pseudo transferring.

If $h = 1$ then $M = \lambda \boldsymbol{y}.c$ where c is a base type constant. In this case, M is pseudo transferring, too.

If $m = 1$ then $\mathcal{R}$ is a single term, so there can only be one equation characterising M and thus M is pseudo transferring.

Let $\langle o, h, m\rangle > \langle 0, 1, 1\rangle$. We have three subcases here according to the transferring structure of M. The only non-trivial case is when M has the form (3). As N_a, N_b have depth less then M, we have by the induction hypothesis pseudo transferring terms $\overline{N_a}, \overline{N_b}$ with n_a, m_a, n_b, m_b as in M.

Let $M'_i = M_i[i_1 \leftarrow z_{i_1} \boldsymbol{x_{i_1}}, \ldots, i_k \leftarrow z_{i_k} \boldsymbol{x_{i_k}}]$ where $\{i_1, \ldots, i_k\}$ are constants from $\mathbb{N}$ that occur in M_i. By Theorem 2 and Theorem 3, we can generate transferring equivalents $\lambda \boldsymbol{z_1}.\widehat{M_1}, \ldots, \lambda \boldsymbol{z_k} \widehat{M_k}$ of corresponding terms $\lambda \boldsymbol{z_1}.M'_1, \ldots, \lambda \boldsymbol{z_1}.M'_k$. This equivalents are taken for the observable $\mathcal{R}_{\{a,b\}}$. This ensures that terms

$\mathtt{M}_i\widehat{M_1}\cdots\widehat{M_k}$ give the same approximations as $\mathtt{M}_iM_1\cdots M_k$ and terms

$$\begin{array}{ll}\mathtt{M}_i\widehat{M_1}\cdots\widehat{M_k}[& a \leftarrow N_a\boldsymbol{y},\\ & b \leftarrow N_b\boldsymbol{y},\\ & z_0 := N_0\boldsymbol{y},\\ & \ldots,\\ & z_l := N_l\boldsymbol{y}]\end{array}$$

give the same results as

$$\begin{array}{ll}\mathtt{M}_iM_1\cdots M_k[& a \leftarrow N_a\boldsymbol{y},\\ & b \leftarrow N_b\boldsymbol{y},\\ & 0 \leftarrow N_0\boldsymbol{y}\boldsymbol{x},\\ & \ldots,\\ & l \leftarrow N_l\boldsymbol{y}\boldsymbol{x}].\end{array}$$

As terms $\lambda\boldsymbol{z_1}.\widehat{M_1},\ldots,\lambda\boldsymbol{z_k}\widehat{M_k}$ have order less than o, we can obtain by the induction hypothesis terms $\lambda\boldsymbol{z_1}.\overline{M_1},\ldots,\lambda\boldsymbol{z_k}\overline{M_k}$ which are pseudo $(n',m',C\cup\mathrm{Const}_i)$-transferring with $(n',m',C\cup\mathrm{Const}_i)$ as in the corresponding $\widehat{M_i}$ terms.

The last step is to obtain pseudo transferring equivalents of $N_0,\ldots,N_l$. This can be done as these terms are used in

$$\begin{array}{ll}\mathtt{M}_i\widehat{M_1}\cdots\widehat{M_k}[& a \leftarrow N_a\boldsymbol{y},\\ & b \leftarrow N_a\boldsymbol{y},\\ & z_0 := N_0\boldsymbol{y},\\ & \ldots,\\ & z_l := N_l\boldsymbol{y}]\end{array}$$

only in contexts resulting in terms from $\mathcal{R}^T$ where T is the set of right-hand sides of equations defined as in (3) in the proof of Theorem 3. As $\mathcal{R}^T\subset\mathcal{R}$, we obtain by the induction hypothesis $\mathcal{R}^T$ equivalents $\overline{N_0},\ldots,\overline{N_l}$ of terms $N_0,\ldots,N_l$.

Now, we can conclude with the construction of a pseudo (n,m,C)-transferring term

$$\begin{array}{ll}\lambda y_1\ldots y_p.y_i\overline{M_1}\cdots\overline{M_k}[& a \leftarrow \overline{N_a}\boldsymbol{y},\\ & b \leftarrow \overline{N_a}\boldsymbol{y},\\ & z_0 := \overline{N_0}\boldsymbol{y},\\ & \ldots,\\ & z_l := \overline{N_l}\boldsymbol{y}].\end{array}$$

□

Proof of Theorem 5:

The decision procedure consists in enumerating all possible pseudo transferring terms for the type of the head variable in an instance of the dual interpolation problem. This may be done recursively according to the definition of pseudo transferring terms. The recursion is parametrised with

- the type order o of head variable,

- the number of terms r in an observable,
- the number of symbols n in right-hand sides of the given instance of the dual interpolation problem,
- the number of equations m in the instance.

These numbers constitute a lexicographic measure for the recursion.

In the case (1) of Definition 10, the algorithm stops.

In the case (2) of Definition 10, n decreases while other parameters stay intact.

In the case (3) of Definition 10, we have the following subcases.

- For N_a and N_b the number m decreases while n does not increase and other parameters stay intact.
- For $N_0, \ldots, N_l$ the number of terms in observable decreases since $\mathcal{R}^T \subset \mathcal{R}$. The order stays intact.
- For $M_1, \ldots, M_l$ the order decreases.

The correctness is based on the knowledge that solutions for the dual interpolation problems are representants for all equivalence classes in the observational equivalence $\approx_{\mathcal{R}}$ where $\mathcal{R}$ is an observable (see Theorem 2). □