Algebraic Higher-Order Matching

Aleksy Schubert

February, 2002

Abstract

The decidability of algebraic higher-order matching is proved. Algebraic higher-order matching is a matching where the constant term in equation is built of first-order constants and constants of a base type. This result is particularly appealing in the light of the last result of Loader.

1 Introduction

The problem to solve equations $M = N$ in the simply typed $\lambda$-calculus is called higher-order unification. This problem has been proved undecidable [Gol81, Hue73, Luc72]. There is a restricted version of the problem known as higher-order matching. In this problem, the equation may have unknowns only on one side, say in $M$. The decidability of the higher-order matching problem is still open.

The higher-order matching problem has a long-lasting history. The problem was posed by G. Huet in his PhD thesis [Hue76]. One way to attack the problem consisted in providing a modified version of Huet’s general unification procedure from [Hue75]. The most serious attempt to solve it this way was conducted by D. Wolfram in [Wol89]. His algorithm has the termination property, but it is not known to be complete. The second direction consisted in conducting induction on the order of types. This resulted in decidability of second-order matching [Bax76], third-order matching [Dow93], fourth-order matching [Pad00] and a special case of fifth-order matching [Sch97].

The situation evolved greatly over the last years. V. Padovani presented a completely new approach to higher-order matching. He proved that higher-order matching is decidable without restriction on the order but with a further restriction on the term $N$ — it must be a single constant of a base type [Pad95a]. On the other hand R. Loader presented in [Loa] the undecidability of unrestricted higher-order matching, but for $\beta$-equality (the standard formulation requires $\beta\eta$-equality).

This paper expands significantly the technique used in [Pad95a] and increases decidable higher-order matching cases. In fact, we generalize approach so that constants in $N$ are replaced by algebraic terms. This is especially appealing in the light of the Loader’s result — algebraic terms are quite close to general higher-order terms so this paper gives more hope for decidability of higher-order matching.

The higher-order matching problem has several applications. The most well-known one is in the field of automated theorem proving. The initial interest in this problem involved this motivation. Unfortunately, this problem has tremendous complexity (provided that it is decidable) [Vor97]. Applications in this field require a lot of work in providing heuristics so that solutions to this problem are practical. On the other hand the paper by Padovani [Pad95a] makes the problem a good point of focus in the area of semantics. In fact, Padovani’s paper shows a very tangible description of a model

*This work was partly supported by KBN grant 7 T11C 028 20.
for the simply typed $\lambda$-calculus which is fully abstract for free. His model is abridged in several ways, but it is commonly admitted that there is a trade-off between tangibility of a model, full abstraction and completeness. We present an extended version of Padovani’s model here which is as tangible as its predecessor and captures a greater scope of behavior in the simply typed $\lambda$-calculus.

Here is a section by section summary of the paper. Section 2 contains basic definitions and serves to settle notation. Section 3 presents definitions specific for this paper. It contains the crucial definition of transferring terms. Section 4 contains a proof of the central theorem that links algebraic dual interpolation and transferring terms. Section 5 draws final conclusions about decidability of the algebraic dual interpolation problem and algebraic higher-order matching.

2 Preliminaries

This section introduces notions that are defined in external sources. This is to settle notation only.

2.1 Types

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

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

2.2 Simply typed terms

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 $T : V \cup C \to T_B$ that indicates types of symbols. We assume that there exists an infinite number of variables of a type $\alpha$. This set is also closed on the application and $\lambda$-abstraction operations. We will usually write $s : \tau$ to denote the fact that a pre-term $s$ has a type $\tau$. As usual, we deal with $\alpha$-equivalence. The set $\lambda_\alpha$ is defined as a quotient of $\lambda^*$ by $\alpha$-equivalence. Elements of $\lambda_\alpha$ are denoted usually by $M, M_1, ..., N, N_1, ...$ etc. while the elements of $\lambda^*$ are denoted by $s, s_1, ..., t, t_1, ...$ etc. The notion of a closed term is defined as usual, similarly the set $FV(M)$ of free variables in $M$.

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

The symbol $\text{Const}_M$ denotes the set of constants that occur in $M$. This notation is extended to sets (and other structures) of terms.

2.3 Order

Definition 1 (order)

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

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

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

The notion of $\beta$-reduction 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 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 $\beta$-normal, $\eta$-long form. The $\beta$-normal, $\eta$-long form for a term $M$ is denoted by NFL($M$).

Definition 3 (an instance of the higher-order matching problem)
We call an instance of the higher-order matching problem each pair of simply typed $\lambda$-terms $(M, N)$. We usually denote them as $M = N$.

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

Definition 5 (an interpolation equation)
An interpolation equation is a pair of terms usually written as

$$xN_1 \cdots N_k = N$$

such that $x$ is the only free variable in the left-hand side of the equation.

Definition 6 (an instance of the dual interpolation problem)
We call an instance of the dual interpolation problem each pair of sets $(E, E')$ 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 sometimes call $(E, E')$ a dual set.

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

Definition 7 (the dual interpolation problem)
The dual interpolation problem is a decision problem — given an instance $(E, E')$ of the dual interpolation problem whether there exists a solution of $(E, E')$.

We deal with the following restrictions of the above-mentioned problems.
Definition 8 (algebraic problems)
A problem is called algebraic iff the set of constants contains only first-order and base type constants and all the right-hand sides of instances are algebraic.

Note that algebraic terms do not contain the $\lambda$ symbol.

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

Theorem 2.1 (reduction of problems)
The problem of higher-order matching reduces to the dual interpolation problem.

Proof:
See [Pad95b] or [Pad96].

This result may be generalized so that the following special case is covered:

Theorem 2.2 (reduction in algebraic case)
The problem of algebraic higher-order matching reduces to the algebraic dual interpolation problem.

Proof:
See [Sch01].

3 Tools and definitions

This section presents specific definitions used in this paper.

3.1 Addresses in terms

Definition 9 (an address in a term)
Let $t$ be a pre-term. We say that a sequence $\gamma$ of natural numbers is an address in $t$ iff

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

The set of all addresses in $t$ is written as $\text{Addr}(t)$.

The suffix order on addresses is written as $\gamma \preceq \gamma'$. The strict version is denoted as $\gamma < \gamma'$.

Definition 10 (pointing out)
We say that a subterm $u$ is pointed out by an address $\gamma$ in $t$ iff

- $u = t$ and $\gamma = \varepsilon$, or
- $u = t'$ where $t = \lambda x_1 \ldots x_n.u_0 u_1 \ldots u_n$, and $\gamma = \gamma' \cdot i$ together with $0 \leq i \leq n$ and $\gamma'(u_i) = t'$.

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

Definition 11 (graft)
A graft of a term $u$ in $t$ at an address $\gamma$ is a term $t'$ defined as
Warsaw University Institute of Informatics

• $u$ iff $\gamma = \varepsilon$,

• $\lambda x_1 \ldots x_n. u_0 u_1 \ldots u_n$ iff $\gamma = \gamma' \cdot i$ and $t = \lambda x_1 \ldots x_n. u_0 u_1 \ldots u_n$ and $u'_i$ is a graft of $u$ in $u_i$ at $\gamma'$.

We denote $t'$ as $t[\gamma \leftarrow u]$.

**Definition 12 (a $C$-pruning)**
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 $t$.

**Definition 13 (pre-order on prunings)**
Let $t'$ and $t''$ be $C$-prunings of a term $t$. We say that

$$t' \preceq t'' \text{ iff } \text{Addr}(t') \subseteq \text{Addr}(t'').$$

This pre-order may be viewed as an extension of the suffix 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 $\lambda$-terms, sets of $\lambda$-terms and sets of pre-terms.

### 3.2 Matrix notation

**Definition 14 (a column)**
Every finite sequence of terms of the same type is called a column. We denote columns by symbols like $V, W$ etc. $V^i$ 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 $\tau$ then we denote this fact as $V : \tau$. We say that $V$ is a constant column when all its elements are the same.

**Definition 15 (a row)**
Every finite sequence of terms is called a row. We denote rows by symbols like $R, Q$ etc. $R_i$ 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 $RR'$.

**Definition 16 (a matrix)**
Every finite sequence of columns is called a matrix. We denote matrices by symbols like $M, N$ etc. $M_i$ denotes the $i$-th column of the matrix $M$. $M^i_j$ denotes the $j$-th element of the $i$-th column of $M$. $M^i$ denotes a row defined as $(M^i)^j = M^i_j$.

**Definition 17 (a matrix and a set of equations)**
Let $\bar{M}$ be a matrix and $W$ a column of a base type. We define a set of equations with the matrix $\bar{M}$ and results $W$ as

$$xM^1_1 \ldots M^1_n \doteq W^1$$

•

$$xM^m_1 \ldots M^m_n \doteq W^m.$$

We denote the set of equations as $[x\bar{M} \doteq W]$. 

5
3.3 Accessibility

Definition 18 (accessible address)
Let \( E = [x \overline{R}_1 \doteq W_1, \ldots, x \overline{R}_n \doteq W_n] \) be a set of interpolation equations. We say that an address \( \gamma \) is accessible in a term \( M \) wrt. \( E \) iff

- \( \gamma \) is an address in \( M \),
- there is an equation \([x \overline{M}_1 \cdots \overline{M}_m = W_i] \in E\) such that

\[
\text{NF}(M[\gamma \leftarrow c]\overline{M}_1 \cdots \overline{M}_m) = c
\]  

has an occurrence of \( c \) where \( c \) is a fresh constant.

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 \( E \)

\[
\text{NF}(M[\gamma \leftarrow c]\overline{M}_1 \cdots \overline{M}_m) = c
\]

where \( c \) is a fresh constant.

3.4 Approximations

Definition 19 (constant pair of columns)
We say that a pair \( W, W' \) of columns is constant iff \( W \) is constant as a column and \( W' \) is not constant for all \( i \).

Definition 20 (an approximation)
An approximation of a pair of columns \( W_1, W_2 \) in a dual set \( E = ([x \overline{W}_1 = W_1, x \overline{W}_2 = W_2]) \) of equations for the solution \( M \) is any pair of columns \( \tilde{W}_1, \tilde{W}_2 \) 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 \([x \overline{M}'_1 \cdots \overline{M}'_r = W'_k] \in E \) we have \( \text{NFL}(M'[\overline{M}'_1 \cdots \overline{M}'_r] = W'_k) \),
- for each constant \( c \in \mathbb{N} \) if some \( \tilde{W}'_j = c \) then there exists \( \gamma \succ \varepsilon \) and \( k, l \) such that \( \gamma(\tilde{W}'_k) = c \) (we say \( c \) is guarded),

We denote by \( N_{\tilde{W}_1 \tilde{W}_2} \) the set \( \text{Const}_{\tilde{W}_1 \tilde{W}_2} \cap (\mathbb{N} \cup \{a, b\}) \).

Note that \( \tilde{W}'_k \) is a \( \{a, b, 0, \ldots, l\} \)-pruning of \( W'_k \) while \( \tilde{W}'_j \) is not necessarily a pruning of \( W'_j \).

For the sake of notational convenience we assume that for each approximation, the set \( N_{\tilde{W}_1 \tilde{W}_2} \cap \mathbb{N} \) is an initial connected subset of \( \mathbb{N} \), e.g. \( \{1, 2, 3\} \) or \( \{1, 2\} \).

Definition 21 (pre-order on approximations)
Let \( \tilde{W}_1, \tilde{W}'_1 \) and \( \tilde{W}_2, \tilde{W}'_2 \) be approximations of a pair of columns \( W, W' \). We write \( \tilde{W}_1, \tilde{W}'_1 \leq \tilde{W}_2, \tilde{W}'_2 \) when

\[
N_{\tilde{W}_1 \tilde{W}_1} \subseteq N_{\tilde{W}_2 \tilde{W}_2}, \text{ and}
\]
if $\gamma(\tilde{\omega}_i) \in N_{\tilde{\omega}_2}$, then there exists $\gamma' \leq \gamma$ such that $\gamma'(\tilde{\omega}'_1) \in N_{\tilde{\omega}_1}$, and

- if $\gamma(\tilde{\omega}_i) \in N_{\tilde{\omega}_2}$, then there exists $\gamma' \leq \gamma$ such that $\gamma'(\tilde{\omega}'_1) \in N_{\tilde{\omega}_1}$.

It is easily verified that $\preceq$ is a pre-order. The strict version of the order is denoted as $\prec$.

**Definition 22 (meet with result)**

We say that for an approximation $\tilde{\omega}, \tilde{\omega}'$ of $\omega, \omega'$ the term $\tilde{\omega}'_k$ meets $\omega'_k$ iff for each occurrence $\gamma$ such that the symbol $\gamma(\tilde{\omega}'_k) = c$ is not in \{a, b\} we have that $\gamma(\omega'_k) = c$, too.

**Definition 23 (minimal approximation)**

Let $\langle [x: \tilde{\omega} \vdash \omega], [x: \tilde{\omega}' \vdash \omega'] \rangle$ be a dual set and $M$ its solution. We say that $\tilde{\omega}, \tilde{\omega}'$ is a minimal approximation of $\omega, \omega'$ iff there exists a pruning $M' \preceq M$ with $NFL(M[M]) = \tilde{\omega}_1$ and $NFL(M'[M']) = \tilde{\omega}'_1$.

We say that a term $M'$ is a minimal pruning for minimal approximation $\tilde{\omega}, \tilde{\omega}'$ iff $\tilde{\omega}, \tilde{\omega}'$ is a minimal approximation of $\omega, \omega'$ and there is no $M'' \preceq M'$ such that $NFL(M''[M]) = \tilde{\omega}_1$ and $NFL(M'[M]) = \tilde{\omega}'_1$.

**Definition 24 (splitting columns)**

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

**Definition 25 (accessibility for dual sets)**

Let $\langle E, E' \rangle$ be a dual set and $M$ its solution. We say that an occurrence $\gamma$ 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 a class of abstraction in this equivalence.

**Definition 26 (observable)**

Let $R = \{R_\tau\}_{\tau \in T}$ be a family of sets indexed by types containing $\lambda$-terms and satisfying conditions

1. all terms in $R$ are in $\beta$-normal, $\eta$-long form,

2. for each term $M \in R$, there exists an $\alpha$-representant $s_M$ of $M$ such that for each subterm $t$ of the pre-term $s_M$ we have $[t]_\alpha \in R$.

Such a set is called an observable.

It is worth noting that for each $n$ the set of algebraic terms with height less or equal to $n$ is an observable.

The notion of an observable is a base for a pre-order and an equivalence relation which, in turn, allow us to define a semantic structure for the simply typed $\lambda$-calculus.
**Definition 27** (observational pre-order)
For each observable $R$ we define an observational pre-order with respect to $R$ as the relation $\sqsubseteq_R$ on $\lambda$-terms such that

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

$$M \text{ and } M' \text{ are both of a base type and if } NFL(M) \in R \text{ then } M =_{\beta\eta} M'.$$

**Definition 28** (observational equivalence)
For each observable $R$ we define an observational equivalence with respect to this observable as the relation on closed terms $\approx_R$ such that

$$M \approx_R M' \iff M \sqsubseteq_R M' \text{ and } M' \sqsubseteq_R M.$$

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

**Theorem 3.1** (dual interpolation and equivalence)
For each solvable instance $E$ of dual interpolation for an observable $R$ there exists a class of abstraction of $\approx_R$ such that all its elements are solutions of $E$.

For each class $A$ of the relation $\approx_R$ there exists an instance $E$ of the dual interpolation problem such that terms from $A$ solve $E$.

**Proof:**
Easy conclusion from [Sch01].

### 3.6 Transferring terms

The proof in this paper is based on the following schema: We define transferring terms for a given finite observable $R$. We show that each solvable dual interpolation instance for the observable 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 in our case.

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 29** (transferring terms)
A term $M : \tau_1 \to \cdots \to \tau_p \to \iota$ is an $(n, m)$-transferring term where $n, m \in \mathbb{N}$ iff

1. $M = \lambda y_1 \ldots y_p. M'$ and $M'$ is a term without any occurrence of $y_i$,
2. $M = \lambda y_1 \ldots y_p. fM_1 \cdots M_k$, where $f$ is a constant of a type $\sigma_1 \to \cdots \to \sigma_k \to \iota$ and $\lambda y_1 \ldots y_p. M_i$ are $(n_i, m)$-transferring terms with $n_i < n$.
3. $M = \lambda y_1 \ldots y_p. y_1M_1 \cdots M_k\{a := N_0\vec{y}, \quad b := N_0\vec{y}, \quad 0 := N_0\vec{y}\vec{x}_0, \quad \ldots, \quad l := N_0\vec{y}\vec{x}_l\}$,

where
\( \bar{y} = y_1, \ldots, y_p; \)
\( M_i \) are closed terms over constants extended with a set \( \text{Const}_i; \)
\( \bigcup_{i=1}^{k} \text{Const}_i = \{a, b, 0, \ldots, l\}; \)
each constant \( 0, \ldots, l \) occurs only once in \( y_i M_1 \cdots M_k; \)
\( N_a \) and \( N_b \) are \((n_a,m_a)\)-transferring and \((n_b,m_b)\)-transferring respectively;
\( N_j \) is \((n_j,m_j)\)-transferring for each \( j; \)
n\( n_a, n_b \leq n; \)
m\( a + m_b \leq m \) and \( m_a, m_b \neq m; \)
n\( j < n \) and \( m_j \leq n. \)

Sometimes when \((n,m)\) are unimportant or clearly seen from the context, we shorten the name of a term to transferring.

Suppose that the point 3 was abridged so that no constants from \( \mathbb{N} \) occur in the presented shape. Note that in the obtained term \( M \) for all \( \lambda \)-bindings occurring in the context
\[
M = \lambda \bar{y}. C[y_k M_1 \cdots (\lambda \bar{z}_j z_j^2 . M_i) \cdots M_n]
\]
one of occurrences of \( z_j \) in \( M_i \) is in a subterm beginning with \( y_l \) from \( \bar{y}. \)

When this constraint is applied, the number of occurrences of variables from \( \bar{y} \) in an \((n,m)\)-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 \( \bar{y}. \) This adds some flexibility and, consequently, expressive power to our terms. We pay a price for this comfort, though. We cannot pay for blocks that declare these variables more than \( n. \)

We sometimes treat informally this particular form of \( \lambda \)-terms as a syntactic form as if all substitutions in case (3) were 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 4.1 (transferring terms and dual interpolation)**

Let \( \mathcal{E} \) be an instance of the algebraic dual interpolation problem. If \( M \) is a solution of \( \mathcal{E} \) then there exist \( n, m \in \mathbb{N} \) and an \((n,m)\)-transferring term \( M' \) such that \( M' \) is a solution of \( \mathcal{E}, \) too.

Moreover, \( n, m \) depend recursively on \( \mathcal{E}. \)

The proof of this theorem is deferred to Subsection 4.2. We have to do some preparations for this proof to be conducted.
4.1 Skipping unimportant variables

The induction step in the proof of Theorem 4.1 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 splitted sets. During the process of construction of transferring solutions for them, we have to compactify \( M \) in several different ways. This section is devoted to the major step of the compactification.

We use \( C \)-prunings for \( C = \{a, b\} \cup \mathbb{N} \) in this and the next sections. Thus we shorten the name \( C \)-pruning to pruning here. We impose an additional 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 30 (an unimportant occurrence)**

Let \( M = \lambda \vec{y}.y_1M_1 \cdots M_k \) be a term such that

- \( M \) solves a dual set \( E = \langle [x\overrightarrow{M} \equiv W], [x\overrightarrow{M} \equiv W'] \rangle \),
- \( M' \) is 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

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

In the following proofs natural numbers from prunings and from the definition of transferring terms mix. This is not harmful. These two uses may be differentiated by the context of use. A more formal approach would require an explicit distinction between these two kinds of constants.

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

**Lemma 4.2 (absence of variables)**

If \( M = \lambda \vec{y}.zM_1 \cdots M_k \) is a term such that

- \( M \) solves a dual set \( E = \langle [x\overrightarrow{M} \equiv W], [x\overrightarrow{M'} \equiv W'] \rangle \),
- \( W, W' \) is a non-constant pair of columns,
- \( M' = \lambda \vec{y}.zM_1 \cdots \hat{M}_i \cdots \hat{M}_k \) is a minimal pruning of \( M \) that gives a minimal approximation of \( W, W' \) wrt. \( E \),
- \( M' \) does not contain unimportant occurrence of a variable from \( \vec{y} \),

then terms \( \hat{M}_i \) do not contain occurrences of variables from \( \vec{y} \).

**Proof:**

The proof is by contradiction. Suppose that \( \gamma \cdot 0 = p \cdot \gamma' \) is an address in \( M' \) of a variable \( y_j \in \vec{y} \). W.l.o.g. we may assume that \( \gamma \) is the \( \preceq \)-minimal address of this kind. As this address is not unimportant,

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

Case (1). We replace the term at $\gamma$ with $a, b$ or one of constants $0, \ldots, l$ depending on the result. We obtain this way either an approximation which is less than the original one or a pruning which is less than $M'$. This contradicts with the choice of $M'$.

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

Lemma 4.3 (skipping unimportant occurrences)

If $M = \lambda \bar{y} \cdot z M_1 \cdots M_k$ is a term such that

- $M$ solves a set of dual equations $E = \langle [x\overline{\mathbf{F}} \triangleq W], [x\overline{\mathbf{F}'} \triangleq W'] \rangle$,
- $W, W'$ is a non-constant pair of columns,
- $M' = \lambda \bar{y} \cdot \bar{z} M_1 \cdots \bar{M}_k$ is a minimal pruning of $M$ that gives minimal approximation of $W, W'$ wrt. $E$,

then

- either there exists $j$ such that $\overline{\mathbf{F}}_j, \overline{\mathbf{F}'}_j$ is a pair of splitting columns for $W, W'$,
- or there is an address $\gamma$ that points out a first-order constant $f$ that is totally head accessible.

Proof:

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 behavior by replacing a corresponding variable with an appropriately modified solution of $E$. 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 $p$ being the number of unimportant occurrences of variables from $\bar{y}$ in $M'$.

Case $p = 0$. Lemma 4.2 implies that the head symbol $z$ is either a first order constant $f$ or a variable $y_j \in \bar{y}$. The latter case implies immediately that $\overline{\mathbf{F}}$ is a splitting column with arguments $\bar{M}_1, \ldots, \bar{M}_k$.

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

$$\gamma(\lambda \bar{y} \cdot \bar{M}_k) = z M'_1 \cdots M'_r[c := N_c \bar{y} \bar{x} \bar{c}] \quad (2)$$

where

- $M'_i = \lambda z_1 \ldots z_r \cdot \bar{M}_i$ are closed terms over constants Const, and $\bar{M}_i$ does not begin with $\lambda$;
- $\varepsilon(N_c)$ begins with a symbol from $\bar{y} \bar{x}$, and $c$ is totally head accessible.

We have several cases according to the form of $N_c$.  

11 TR 02-02 (267)
1. The head symbol of \( N_c \) is a constant. The form of (2) implies that

\[
M = C_M[z\overline{M}_1' \cdots \overline{M}_r'] \ c := \overline{N_c}y\overline{\varepsilon}_c]
\]

where \( M_1', \ldots, M_r', N_c \) are suitable prunings of \( M_1', \ldots, M_r', N_c \). The term \( \overline{N_c} \) begins with a constant. This term cannot be a constant term, since \( W, W' \) is not a constant pair. This means that \( \overline{N_c} \) begins with a function symbol. Similarly, the function symbol starts \( N_c \). This means that for the address \( \gamma' \) of \( N_c \) in \( M' \) the address \( \gamma' \cdot 0 \) is a path to a totally head accessible function symbol.

2. The head symbol of \( N_c \) is \( x \in y\overline{\varepsilon}_c \). We define a context \( C_\gamma[\bullet] \) as \( C_\gamma[\gamma(M)] = M \). The term \( M_{\text{new}} \) is defined as

\[
M_{\text{new}} = C_\gamma[\overline{N_c}y\overline{\varepsilon}_cR_1 \cdots R_p]
\]

Let \( x_i \in \overline{x}_c \) (with \( x_i : \tau_1 \rightarrow \cdots \rightarrow \tau_r \rightarrow \iota \)), we define

\[
R_i = \lambda z_1 \ldots z_r. C_\gamma[zM_1' \cdots M_r'[c := x_i \cdot z_1 \cdots z_r]].
\]

Note that for such \( M_{\text{new}} \) the term \( M_{\text{new}}' \) is a pruning of \( M_{\text{new}} \) defined as

\[
M_{\text{new}}' = C_\gamma[N_c y\overline{\varepsilon}_c(\lambda z_1. c_1) \cdots (\lambda z_r. c_p)]
\]

where \( c_1, \ldots, c_p \in \{a, b\} \cup \mathbb{N} \) and we can always choose the constants \( c_1, \ldots, c_p \) so that conditions for pruning and approximation are met.

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

It is easily verified that \( M_{\text{new}} \) is a solution of \( \mathcal{E} \). As \( M_{\text{new}}' \) is a pruning of \( M_{\text{new}} \), we have a minimal pruning \( M_{\text{new}}'' \) of \( M_{\text{new}} \) wrt. \( W, W' \). The term \( M_{\text{new}}'' \) has strictly less occurrences of unimportant variables than \( M' \) thus we can apply the induction hypothesis and obtain that either some \( \overline{R}' \) is a splitting column or that there exists an address \( \gamma' \) in \( M_{\text{new}}'' \) which is an occurrence of a totally head accessible constant symbol \( f \). If \( \gamma' = k \cdot \gamma \cdot \hat{\gamma} \cdot 0 \) then we obtain an address \( \gamma'' \) of \( f \) in \( M' \) as

\[
\gamma'' = k \cdot \gamma_c \cdot \hat{\gamma} \cdot 0
\]

where \( \gamma_c(zM_1' \cdots M_r') = c \). If \( \gamma' \neq k \cdot \gamma \cdot \hat{\gamma} \cdot 0 \) then this \( \gamma' \) is also an occurrence of \( f \) in \( M' \).

\[
\square
\]

4.2 Proof of the main theorem

Proof of Theorem 4.1:

Let \( \mathcal{E} = \langle E, E' \rangle = \langle [x\overline{M} = W], [x\overline{M}' = W'] \rangle \). We define the value \( n \) as the number of symbols in \( W, W' \), the value \( m \) as the number of equations in \( \mathcal{E} \) and the value \( h \) as the height of \( M \).

The proof is by induction wrt. lexicographic order on triples \( (n, m, h) \).

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

If \( m = 1 \), then \( M' = \lambda \overline{y}. W^1 \) is a transferring solution provided that \( [x\overline{M} = W] \neq \emptyset \). If \( [x\overline{M} = W] = \emptyset \) then \( M' = \lambda \overline{y}. c \) where \( c \neq W^1 \) is a transferring solution.

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

If \( n = 1 \), then this reduces to the case when \( m = 1 \).

If \( (n, m, h) > (1, 1, 1) \), then we have two subcases

1. \( M = \lambda \overline{y}. f M_1 \cdots M_k \);
2. \( M = \lambda \vec{y}.y_1 M_1 \cdots M_k \) for some \( y_i \in \vec{y} \).

**Case** \( M = \lambda \vec{y}.f M_1 \cdots M_k \). In this case, all terms in \( W \) begin with \( f \):

\[
W = \langle f B_1^1 \cdots B_k^1, \ldots, f B_1^p \cdots B_k^p \rangle
\]

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

\[
W'' = \langle f B_1^{n_1} \cdots B_k^{n_1}, f B_1^{n''} \cdots B_k^{n''} \rangle.
\]

We define columns \( W_1, \ldots, W_k \) and \( W'_1, \ldots, W'_{k'} \) as \( W_i = \langle B_1^i, \ldots, B_k^i \rangle \), and \( W'_i = \langle B_1^{n_i}, \ldots, B_k^{n''} \rangle \). We can define sets of equations \( E_i = [x \vec{W} \models W_i] \) and \( E'_i = [x \vec{W}' \models W'_i] \) (\( \vec{W}' \) is a sub-matrix of \( \vec{W} \) corresponding to \( W'' \)). It is easily verified that \( \langle E_i, E'_i \rangle \) has a solution \( \lambda \vec{y}.M_i \). These terms have less height than \( M \) so by the induction hypothesis we have \( (n_i, m_i) \) -transferring solutions \( \lambda \vec{y}.M'_i \) of sets \( \langle E_i, E'_i \rangle \) where \( n_i \) is the number of symbols in \( W_i W'_i \) and \( m_i \) is the number of equations in \( E_i \cup E'_i \). The number of symbols in \( W_i W'_i \) is strictly less than the number of symbols in \( W \) so \( n_i < n \). Thus, \( \lambda \vec{y}.f M'_1 \cdots M'_k \) is a \((n, m)\) -transferring solution of \( \vec{E} \).

**Case** \( M = \lambda \vec{y}.y_1 M_1 \cdots M_k \) for some \( y_i \in \vec{y} \).

Let \( \vec{W}, \vec{W}' \) be a minimal approximation of the pair \( W, W' \). Let \( M' \) be a minimal pruning for minimal approximation of \( W, W' \). Lemma 4.3 implies that either \( \vec{W}_j, \vec{W}'_j \) are splitting columns for \( W, W' \) or there exists an occurrence of a first order constant \( f \) in \( M \) which is totally head accessible.

**Case when certain \( \vec{W}_j, \vec{W}'_j \) are splitting columns for \( W, W' \).** Definition 24 says that there exist terms \( P_1, \ldots, P_r \) such that \( \text{NFL}(\vec{W}_j P_1 \cdots P_r) = \vec{W} \) and \( \text{NFL}(\vec{W}'_j P_1 \cdots P_r) = \vec{W}' \). We produce on the basis of \( \vec{E} \) new dual sets of equations \( E_a, E_b, E_0, \ldots, E_{l} \). These sets are defined as \( \langle E_c \rangle = \langle E_{c'}, E_{c''} \rangle \) where

1. \( E_a = \{ [x \vec{W} \models W^k] | \text{NFL}(\vec{W}_j P_1, \ldots, P_r) = a \} \), and \( E'_a = \{ [x \vec{W}' \models W'^k] | \text{NFL}(\vec{W}_j P_1, \ldots, P_r) = a \} \),
2. \( E_b = \{ [x \vec{W} \models W^k] | \text{NFL}(\vec{W}_j P_1, \ldots, P_r) = b \} \), and \( E'_b = \{ [x \vec{W}' \models W'^k] | \text{NFL}(\vec{W}_j P_1, \ldots, P_r) = b \} \),
3. the definition of \( E_{c}, E'_{c} \) for \( i \in \mathbb{N} \) is more complex and presented hereafter.

Let \( [x \vec{W} \models \vec{W}] = E_a \cup E_b \) and \( [x \vec{W}' \models \vec{W}'] = E'_a \cup E'_b \). We have

\[
\begin{align*}
\text{NFL}(\vec{W}_j P_1 \cdots P_r) & = \vec{W}^k = C^k[n_1, \ldots, n_q] \\
\text{NFL}(\vec{W}_j P_1 \cdots P_r) & = \vec{W}'^k = C'^k[n_1, \ldots, n_q]
\end{align*}
\]

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

\[
\begin{align*}
\Pi_{n_k}^k & = \{ \gamma | \gamma \text{ points out on a constant } n_s^k \text{ in } \vec{W}^k \} \\
\Pi_{n_k}^k & = \{ \gamma | \gamma \text{ points out on a constant } n_s^k \text{ in } \vec{W}'^k \}
\end{align*}
\]

We settle sets of equations

\[
\begin{align*}
E_n &= \{ [x^n \vec{W}_1^k \cdots \vec{W}_m^k O_1^k \cdots O_{k'}^k \models \gamma(W^k)] | \gamma \in \Pi_{n_k}^k \text{ and } k \in \mathbb{N} \} \\
E'_n &= \{ [x^n \vec{W}_1^k \cdots \vec{W}_m^k O_1^k \cdots O_{k'}^k \models \gamma(W'^k)] | \gamma \in \Pi_{n_k}^k \text{ and } k \in \mathbb{N} \} \quad \text{(3)}
\end{align*}
\]

and \( \vec{W}^k \) meets with \( \vec{W}'^k \).
where \( O^k_i \) are defined as follows. Let \( \gamma \) be an occurrence of a constant \( n^k_i \) in \( M' \) that results in \( n^k_i \) in \( W^k \). Replace the constant with a subterm

\[
Zx_1 \cdots x_{l'}
\]

where \( Z \) is a fresh constant of a suitable type and \( \{x_1, \ldots, x_{l'}\} = \text{FV}(\gamma(M))/\vec{y} \). Let the obtained term be \( M'' \). We have

\[
\begin{align*}
\text{NFL}(M'' \vec{w}^k_1 \cdots \vec{w}^k_m) &= C^k[n^k_1 \cdots (ZO^k_1 \cdots O^k_1) \cdots n^k_i], \\
\text{NFL}(M'' \vec{w}^k_1 \cdots \vec{w}^k_m) &= C^k[n^k_1 \cdots (ZO^k_1 \cdots O^k_1) \cdots n^k_i].
\end{align*}
\]

The dual sets \( E_a, E_b, E_0, \ldots, E_s \) are solvable by the terms \( M, M, \tilde{M}_0, \ldots, \tilde{M}_s \) respectively. We define \( \tilde{M}_i \) as

\[
\lambda \vec{y}_i y_{m+1} \cdots y_{m+l'} . \tilde{M}'_i
\]

where \( \tilde{M}'_i \) is \( M \) with variables \( x_1, \ldots, x_{l'} \) defined in (4) replaced by variables \( y_{m+1}, \ldots, y_{m+l'} \) respectively.

The sets \( E_a, E_b \) have the size less than \( E \). In the sets \( E_0, \ldots, E_i \), 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 of \( E_a, E_b, E_0, \ldots, E_i \) being \( M_a, M_b, M_0, \ldots, M_i \). We define a term

\[
\tilde{M} = \lambda \vec{y}_i y_{j} P_l \cdots P_r [ \begin{array}{c} a := M_a y_{i}, \\ b := M_b y_{j}, \\ 0 := M_0 y_{i} x_0, \\ \ldots \\ s := \tilde{M} y_{i} x_s \end{array} ]
\]

where \( x_i \) are variables bound in the \( P_k \) in which \( i \) occurs as a constant.

The term \( \tilde{M} \) is a solution, because when we reduce it with left-most, out-most strategy applied in equations of \( E \), then we reach either \( \tilde{M}_a \vec{w}^k_1 \cdots \vec{w}^k_m \) or \( C^k(M^k_1 \cdots \cdots \cdots (M^k_1 \cdots \cdots \cdots \tilde{M}_k \cdots)) \). As \( M_a, M_b, M_0, \ldots, M_i \) are solutions of equations \( E_a, E_b, E_0, \ldots, E_i \), we immediately see that reached terms reduce further to appropriate right-hand sides defined in \( W, W' \). Note that so defined \( \tilde{M} \) is \( (n,m) \)-transferring solution of \( E \).

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

\[\Box\]

5 Decidability

We have to provide yet another characterisation of solutions for the algebraic higher order matching problem. This characterisation relies strongly on some constructions for observables.

**Definition 31 (pruned observable)**

Let \( R \) be an observable. The observable \( R_C \) is called a pruning of \( R \) if it contains all the \( C \)-prunings of terms from \( R \).

TR 02-02 (267) 14
Note that \( R \subseteq R_C \) for each \( C \).

**Definition 32 (observable for terms)**
Let \( T \) be a set of terms. \( R^T \) is a minimal observable containing \( T \).

Note that if \( T \subseteq R \) then \( R^T \subseteq R \).

**Definition 33 (weakly transferring terms)**
Let \( R \) be a finite observable. A term \( M : \tau_1 \rightarrow \cdots \rightarrow \tau_p \rightarrow \iota \) is a weakly \((n, m)\)-transferring term for the observable \( R \) where \( n, m \in \mathbb{N} \) iff

1. \( M = \lambda y_1 \ldots y_p. M' \) and \( M' \) is a term without any occurrence of \( y_i \),

2. \( M = \lambda y_1 \ldots y_p. f M_1 \cdots M_k \), where \( f \) is a constant of a type \( \sigma_1 \rightarrow \cdots \rightarrow \sigma_k \rightarrow \iota \) and \( \lambda y_1 \ldots y_p. M_i \) are weakly \((n_i, m_i)\)-transferring terms for the observable \( R \) with \( n_i < n \).

3. \( M = \lambda y_1 \ldots y_p. y_i M_1 \cdots M_k \)

where

- \( \bar{y} = y_1, \ldots, y_p \);
- \( \lambda z_{i_1} \ldots z_{i_k}. M_i \) are weakly \((n_i, m_i)\)-transferring terms for the observable \( R_{\{a, b, 0, \ldots, l\}} \) over constants extended with a set \( \text{Const}_i \), where \( n_i \) is the maximal number of equations in a dual set \( \mathcal{E} \) characterising a single class of abstraction as in Theorem 3.1, and \( m_i \) is the maximal total size of right-hand sides in such an \( \mathcal{E} \);
- \( \bigcup_{i=1}^k \text{Const}_i = \{a, b, 0, \ldots, l\} \);
- \( N_a \) and \( N_b \) are weakly \((n_a, m_a)\)-transferring and weakly \((n_b, m_b)\)-transferring respectively for the observable \( R \);
- \( T \) is a strict subset of \( R \);
- for each \( j \) the term \( N_j \) is weakly \((n_j, m_j)\)-transferring for the observable \( R^T \) where \( T \) is the set of right-hand sides of equations analogous to equations (3) in the proof of Theorem 4.1.

- \( n_a, n_b \leq n \);
- \( m_a + m_b \leq m \) and \( m_a, m_b \neq m \);
- \( n_j \) is the number of equations in a dual set characterising a class of abstraction of observational equivalence for \( R^T \) and \( m_j \) is the total size of right-hand sides in the aforementioned dual set.

Sometimes when \((n, m)\) are unimportant or clearly seen from the context, we shorten the name of a term to weakly transferring.

We have the following theorem:

**Theorem 5.1 (weakly transferring terms)**
If a dual interpolation set \( \mathcal{E} \) has an \((n, m)\)-transferring solution \( M \) for an observable \( R \) then it has a weakly \((n, m)\)-transferring solution for the observable, too.
Proof: 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 \( R \).

If \( o = 0 \) then \( E = \langle E, E' \rangle \) may have only one equation in \( E \) and the equation has the form \( x = M \). In this case, \( M \) is weakly transferring.

If \( h = 1 \) then \( M = \lambda \vec{y}.c \) where \( c \) is a base type constant. In this case, \( M \) is weakly transferring, too.

If \( m = 1 \) then \( R \) is a single term, so there can only be one equation characterising \( M \) and thus \( M \) is weakly 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 than \( M \), we have by the induction hypothesis weakly transferring terms \( \overline{N_a}, \overline{N_b} \) with \( n_a, m_a, n_b, m_b \) as in \( M \).

Let \( M'_i = M_i[i_1 := z_i x'_i, \ldots, i_k := z_i x'_k] \) where \( \{i_1, \ldots, i_k\} \) are constants from \( \mathbb{N} \) that occur in \( M_i \). By Theorem 3.1 and Theorem 4.1, we can generate transferring equivalents \( \lambda \vec{z}_i \overline{M_1}, \ldots, \lambda \vec{z}_i \overline{M_k} \) of terms \( \lambda \vec{z}_i.M'_1, \ldots, \lambda \vec{z}_i.M'_k \). This equivalents are taken for the observable \( R_{\{a, b, 0, \ldots l\}} \). This ensures that terms \( \vec{m}_i \overline{M_1} \cdots \overline{M_k} \) give the same approximations as \( \vec{m}_i M_1 \cdots M_k \) and terms

\[
\begin{align*}
\vec{m}_i \overline{M_1} \cdots \overline{M_k} &\mid a := N_a \vec{y}, \\
b &:= N_b \vec{y}, \\
z_0 &:= N_0 \vec{y}, \\
\ldots, \\
z_l &:= N_l \vec{y}
\end{align*}
\]

give the same results as

\[
\begin{align*}
\vec{m}_i M_1 \cdots M_k &\mid a := N_a \vec{y}, \\
b &:= N_b \vec{y}, \\
0 &:= N_0 \vec{y}\vec{x}, \\
\ldots, \\
l &:= N_l \vec{y}\vec{x}.
\end{align*}
\]

As terms \( \lambda \vec{z}_i \overline{M_1}, \ldots, \lambda \vec{z}_i \overline{M_k} \) have order less than \( o \), we can obtain by the induction hypothesis terms \( \lambda \vec{z}_i \overline{M_1}, \ldots, \lambda \vec{z}_i \overline{M_k} \) which are weakly \((n, m)\)-transferring with \((n, m)\) as in the corresponding \( \overline{M_i} \) terms.

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

\[
\begin{align*}
\vec{m}_i \overline{M_1} \cdots \overline{M_k} &\mid a := N_a \vec{y}, \\
b &:= N_a \vec{y}, \\
z_0 &:= N_0 \vec{y}, \\
\ldots, \\
z_l &:= N_l \vec{y}
\end{align*}
\]

only in contexts resulting in terms from \( R^T \) where \( T \) is the set of right-hand sides of equations defined as in (3) in the proof of Theorem 4.1. As \( R^T \subset R \), we obtain by the induction hypothesis \( R^T \) equivalents \( \overline{N_0}, \ldots, \overline{N_l} \) of terms \( N_0, \ldots, N_l \).
Now, we can conclude with the construction of a weakly \((n,m)\)-transferring term
\[
\lambda y_1 \ldots y_p. y_i M_1 \cdots M_k \left| a := \overrightarrow{N_a y}, \right.
\]
\[
\left. b := \overrightarrow{N_a y}, \right.
\]
\[
z_0 := \overrightarrow{N_0 y}, \quad \ldots, \quad z_l := \overrightarrow{N_l y}.
\]

We are now ready to conclude with the following theorem.

**Theorem 5.2 (decidability of the dual interpolation problem)**

The algebraic dual interpolation problem is decidable.

**Proof:**

The decision procedure consists in enumerating all possible weakly transferring terms for the type of the head variable in an instance of the dual algebraic interpolation problem. This may be done recursively according to the definition of weakly transferring terms. The recursion is based on the knowledge that solutions for the dual interpolation problems are representants for all classes of abstraction in the observational equivalence \(\approx_R\) where \(R\) is an observable of algebraic terms (see [Sch01]).

As an easy corollary we obtain.

**Theorem 5.3 (decidability of algebraic higher-order matching)**

Algebraic higher-order matching is decidable.

### 6 Acknowledgments

I would like to thank Prof. Pawel Urzyczyn and Prof. Jerzy Tiuryn who urged me to deal with the higher-order matching problems. Gilles Dowek and Vincent Padovani took a while to discuss these problems with me. Vincent Padovani’s work is the most influential work on the higher-order matching problem I know. Jurek Marcinkowski is another person I want to thank to, especially for so much more than being not interested in higher-order matching problems. ToMasz Wierzbicki is another helpful and even slightly interested person. Prof. Leszek Pacholski and Marcin Benke were so kind enough to ask me about higher-order matching several times. Last but not least I thank my wife Joanna for her help during the preparation of the paper.

### References


Institute of Informatics
Warwal University


