Third Order Matching is Decidable

Gilles Dowek
INRIA-Rocquencourt,
B.P. 105, 78153 Le Chesnay CEDEX, France
dowek@margaux.inria.fr
and
School of Computer Science,
Carnegie Mellon University,
Pittsburgh PA15213-3890, U.S.A.
gdowek@cs.cmu.edu

Abstract

The higher order matching *problem is the problem of determining whether a term is an instance of another in the simply typed λ-calculus, i.e. to solve the equation $a = b$ where a and b are simply typed λ-terms and b is ground. The decidability of this problem is still open. We prove the decidability of the particular case in which the variables occurring in the term a are at most third order.*

Introduction

The *higher order matching* problem is the problem of determining whether a term is an instance of another in the simply typed λ-calculus i.e. to solve the equation $a = b$ where a and b are simply typed λ-terms and b is ground.

Pattern matching algorithms are used to check if a proposition can be deduced from another by elimination of universal quantifiers or by introduction of existential quantifiers. In automated theorem proving, elimination of universal quantifiers and introduction of existential quantifiers are mixed and full unification is required, but in proof-checking and semi-automated theorem proving, these rules can be applied separately and thus pattern matching can be used instead of unification.

The higher order matching is conjectured decidable in [6] and the problem is still open. In [5] [6] [7] Huet has given a semi-decision algorithm and shown that in the particular case in which the variables occurring in the term a are at most second order this algorithm terminates, and thus that second order matching is decidable. In [9] Statman has reduced the conjecture to the λ-definability conjecture and in [10] Wolfram has given an always terminating algorithm whose completeness is conjectured.

We prove in this paper that third order matching is decidable i.e. we give an algorithm that decides if a matching problem, in which all the variables are at most third order, has a solution. The main idea is that if the problem $a = b$ has a solution then it also has a solution whose depth is bounded by some integer s depending only on the problem $a = b$, so a simple enumeration of the substitutions whose depth is bounded by s gives a decision algorithm. This result can also be used to bound the depth of the search tree in Huet's semi-decision algorithm and thus turn it into an always terminating algorithm.

At last we discuss the problems that occur when we try to generalize the proof given here to higher order matching.

1 Trees and Terms

1.1 Trees

Definitions 1 (Following [3]) An *occurrence* is a list of strictly positive integers. A *tree domain* D is a non empty finite set of occurrences such that if $\alpha[n] \in D$ then $\alpha \in D$ and if also $n \neq 1$ then $\alpha[n-1] \in D$. A *tree* is a function from a tree domain D to a set L, called the set of labels of the tree.

If T is a tree and D its domain, the occurrence $[\,]$ is called the *root* of T and the occurrence $\alpha[n]$ is called the n^{th} *son* of the occurrence α. The *number of sons* of an occurrence α is the greatest integer n such that $\alpha[n] \in D$. A *leaf* is an occurrence that has no sons.

Let T be a tree and $\alpha = [s_1; ...; s_n]$ an occurrence in this tree, the *path* of α is the set of occurrences $\{[s_1; ...; s_p] \mid p \leq n\}$. The number of elements of this path is the length of α plus one.

The *depth* of the tree T is the length of the longest occurrence in D. This occurrence is, of course, a leaf.

If a is a label and $T_1, ..., T_n$ are trees (of domains $D_1, ..., D_n$) then the *tree of root a and sons $T_1, ..., T_n$* is the tree T of domain $D = \{[\,]\} \cup \bigcup_i \{[i]\alpha \mid \alpha \in D_i\}$ such that

$$T([\,]) = a$$

and

$$T([i]\alpha) = T_i(\alpha)$$

0-8186-2735-2/92 $3.00 © 1992 IEEE

If T is a tree of domain D and α is an occurrence of D, the *subtree* T/α is the tree T' whose domain is $D' = \{\beta \mid \alpha\beta \in D\}$ and such that

$$T'(\beta) = T(\alpha\beta)$$

If T is a tree of domain D, α an occurrence of D and T' a tree of domain D' then the *graft* of T' in T at the occurrence α ($T[\alpha \leftarrow T']$) is the tree T'' of domain $D'' = D - \{\alpha\beta \mid \alpha\beta \in D\} \cup \{\alpha\beta \mid \beta \in D'\}$ and such that

$$T''(\gamma) = T'(\beta) \text{ if } \gamma = \alpha\beta$$

and

$$T''(\gamma) = T(\gamma) \text{ otherwise}$$

Let T and T' be trees, and a a label such that all the occurrences of a in T are leaves $\alpha_1, ..., \alpha_n$ then the *substitution* of T' for a in T ($T[a \leftarrow T']$) is defined as $T[\alpha_1 \leftarrow T']...[\alpha_n \leftarrow T']$. Remark that since $\alpha_1, ..., \alpha_n$ are leaves, the order in which the grafts are performed is insignificant.

1.2 Types

Definition 2 Let us consider a finite set $\mathcal{T}$. The elements of $\mathcal{T}$ are called *atomic types*. A *type* is a tree whose labels are either the elements of $\mathcal{T}$ or $\rightarrow$ and such that the occurrences labeled by an element of $\mathcal{T}$ are leaves and the ones labeled by $\rightarrow$ have two sons.

Let T be a type, if the root of T is an atomic type U then T is written U, if the root of T is $\rightarrow$ and its sons are written T_1 and T_2 then T is written $(T_1 \rightarrow T_2)$. By convention $T_1 \rightarrow T_2 \rightarrow T_3$ is an abbreviation for $(T_1 \rightarrow (T_2 \rightarrow T_3))$.

Definition 3 If T is a type, the *order* of T is defined by

- $o(T) = 1$ if T is atomic,
- $o(T_1 \rightarrow T_2) = max\{1 + o(T_1), o(T_2)\}$.

1.3 Typed λ-terms

Definitions 4 For each type T we consider three sets $\mathcal{U}_T$, $\mathcal{L}_T$, $\mathcal{E}_T$. The elements of $\mathcal{U}_T$ are called *universal variables* of type T, those of $\mathcal{L}_T$ *local variables* of type T and those of $\mathcal{E}_T$ *existential variables* of type T. We assume that we have in each atomic type at least a universal variable and that there is a finite number of universal variables i.e. that the set $\bigcup_T \mathcal{U}_T$ is finite. We assume also that we have in each type an infinite number of local and existential variables.

A typed λ-term is a tree whose labels are either *App*, $< Lam, x >$ where x is a local variable or $< Var, x >$ where x is a universal, local or existential variable, such that the occurrences labeled by *App* have two sons, the occurrences labeled $< Lam, x >$ have one son and the occurrences labeled $< Var, x >$ are leaves.

Let t be a term, if the root of t is $< Var, x >$ we write it x, if the root of t is $< Lam, x >$ and its son is written u then we write it $[x : T]u$ where T is the type of x, if the root of t is *App* and its sons are written u and v then we write it $(u\ v)$. By convention $(u\ v\ w)$ is an abbreviation for $((u\ v)\ w)$.

In a term t, an occurrence α labeled by $< Var, x >$ is *bound* if there exists an occurrence β in the path of α labeled by $< Lam, x >$, it is *free* otherwise.

A term is *ground* if no occurrence is labeled by a pair $< Var, x >$ with x existential.

Let t and t' be terms and x be a variable, the *substitution* of t' for x in t ($t[x \leftarrow t']$) is defined as $t[< Var, x > \leftarrow t']$.

Definition 5 Type of a term

A term t is said to have the type T if either:

- t is a variable (universal, local or existential) of type T.
- $t = (u\ v)$ and u has type $U \rightarrow T$ and v type U for some type U,
- $t = [x : U]u$, the term u has type V and $T = U \rightarrow V$.

A term t is said to be *well-typed* if there exists a type T such that t has type T. In this case T is unique and is called *the type of* t.

Definition 6 The $\beta\eta$-reduction is defined as smallest transitive relation, compatible with term structure such that

$$([x : T]t\ u) \rhd t[x \leftarrow u]$$

$$[x : T](t\ x) \rhd t \quad \text{if } x \text{ is not free in } t$$

We adopt the usual convention of considering terms up to α-conversion (i.e. bound variable renaming) and we consider that bound variables are renamed to avoid capture during substitutions. A rigorous presentation would use de Bruijn indices [2].

Proposition 1 The $\beta\eta$-reduction relation is strongly normalizable and confluent on typed terms, and thus each term has a unique normal form.
Proof See, for instance, [4].

Proposition 2 Let t be a normal well-typed term of type $T_1 \rightarrow ... \rightarrow T_n \rightarrow T$ (T atomic), the term t has the form

$$t = [y_1 : T_1]...[y_m : T_m](x\ u_1\ ...\ u_p)$$

where $m \leq n$ and x is a variable.
Proof The term t can be written in a unique way $t = [y_1 : U_1]...[y_m : U_m]u$ where u is not an abstraction. The term u can be written in a unique way $u = (v\ u_1\ ...\ u_p)$ where v is not an application. The term v is not an application by definition, it is not an abstraction (if $p = 0$ because u is not an abstraction and if $p \neq 0$ because t is normal), it is therefore a variable. Then for type reasons $m \leq n$ and for all i, $U_i = T_i$.

Definition 7 If $t = [y_1 : T_1]...[y_m : T_m](x\ u_1\ ...\ u_p)$ is a term of type $T = T_1 \rightarrow ... \rightarrow T_n \rightarrow T$ (T atomic) ($m \leq n$) which is in $\beta\eta$-normal form then we define its *β-normal η-long* form as the term

$$t' = [y_1 : T_1]...[y_m : T_m][y_{m+1} : T_{m+1}]...[y_n : T_n] \\ (x\ u_1'\ ...\ u_p'\ y_{m+1}'\ ...\ y_n')$$

where u_i' is the β-normal η-long form of u_i and y_i' is the β-normal η-long form of y_i.

This definition is by induction on the pair $< c_1, c_2 >$ where c_1 is the number of occurrences in t and c_2 the number of occurrences in T

In the following all the terms are assumed to be on β-normal η-long form.

1.4 Böhm Trees

Definition 8 Böhm Tree

A (finite) *Böhm tree* is a tree whose occurrences are labeled by pairs $< l, x >$ such that l is a list of local variables $[y_1; ...; y_n]$ and x is a variable and the number of sons of an occurrence labeled by $< l, x >$ is the arity of x i.e. the integer p such that the type of x has the form $T_1 \rightarrow ... \rightarrow T_p \rightarrow T$ with T atomic.

Definition 9 Type of a Böhm Tree

Let t be a Böhm tree whose root is labeled by the pair $< [y_1; ...; y_n], x >$ and whose sons are $u_1, ..., u_p$. The Böhm tree t is said to have the type T if the Böhm trees $u_1, ..., u_p$ have type $U_1, ..., U_p$ the variable x has type $U_1 \rightarrow ... \rightarrow U_p \rightarrow U$ and $T = T_1 \rightarrow ... \rightarrow T_n \rightarrow U$ where $T_1, ..., T_n$ are the types of the variables $y_1, ..., y_n$.

A Böhm tree t is said to be *well-typed* if there exists a type T such that t has type T. In this case T is unique and is called *the type of* t.

Definition 10 Let t be a λ-term in normal form. We write $t = [y_1 : T_1]...[y_n : T_n](x\ u_1\ ...\ u_p)$. The *Böhm tree* of t is the tree whose root is the pair $< l, x >$ where $l = [y_1; ...; y_n]$ is the list of the variables bound at the top of this term, x is the head variable of t and sons are the Böhm trees of $u_1, ..., u_p$.

Remark Normal well-typed terms and well-typed Böhm trees are in one-to-one correspondence. Moreover if t is a normal term and $\tilde{t}$ is its Böhm tree then occurrences in t labeled by a variable and occurrences in $\tilde{t}$ are in one-to-one correspondence.

So we will use the following abuse of notation: if α is an occurrence in the Böhm tree of t we write (t/α) for the normal term corresponding to the Böhm tree $(\tilde{t}/\alpha)$ and $t[\alpha \leftarrow u]$ for the term $t[\alpha' \leftarrow u]$ where α' is the variable occurrence in t corresponding to α.

Definition 11 Let t be a term, we write $|t|$ for the depth of the Böhm tree of the normal form of t.

Proposition 3 In each type T there is a ground term t such that $|t| = 0$.

Proof Let $T = U_1 \rightarrow ... \rightarrow U_n \rightarrow U$ with U atomic. Let x be a universal variable of type U. The term $t = [y_1 : U_1]...[y_n : U_n]x$ has type T and its depth is 0.

1.5 Substitution

Definition 12 A *substitution* is a finite set of pairs $< x_i, t_i >$ where x_i is an existential variable and t_i a term of the same type in which no local variable occurs free such that if $< x, t >$ and $< x, t' >$ are both in this set then $t = t'$. The variables x_i are said to be *bound* by the substitution.

Definition 13 If σ is a substitution and t a term then we let

$$\sigma t = t[\alpha_1^1 \leftarrow t_1]...[\alpha_1^{p_1} \leftarrow t_1]...[\alpha_n^1 \leftarrow t_n]...[\alpha_n^{p_n} \leftarrow t_n]$$

where $\alpha_i^1, ..., \alpha_i^{p_i}$ are the occurrences of x_i in t.

Remark that since the α_i^j are leaves, the order in which the grafts are performed is insignificant.

Definition 14 Let σ and τ be two substitutions the substitution $\tau \circ \sigma$ is defined by

$$\tau \circ \sigma = \{< x, \tau t > \ |\ < x, t > \in \sigma\} \cup \{< x, t > \ |\ < x, t > \in \tau \text{ and } x \text{ not bound by } \sigma\}$$

Proposition 4 Let σ and τ be two substitutions and t is a term, we have

$$(\tau \circ \sigma)t = \tau(\sigma t)$$

Proof By decreasing induction on the depth of an occurrence α in t we prove that we have

$$(\tau \circ \sigma)(t/\alpha) = \tau(\sigma(t/\alpha))$$

2 Pattern Matching

Definition 15 Matching Problem

A *matching problem* is a pair of well-typed terms $< a, b >$ where a and b have the same type and b is ground.

Definition 16 Third Order Matching Problem

A *third order matching problem* is a matching problem $< a, b >$ such that the types of the existential variables that occur in a are of order at most three.

Definition 17 Solution

Let $a = b$ be a matching problem. A substitution σ is a *solution* of this problem if and only if the normal forms of the terms σa and b are identical up to α-conversion.

Remark Usual unification terminology distinguishes *variables* (here existential variables) and *constants* (here universal variables). The need for local variables comes from the fact that we want to transform the problem $[y : T]x = [y : T]y$ (where x is an existential variable of type T) into the problem $x = y$ by dropping the common abstraction. The symbol y cannot be an existential variable because it cannot be instanciated by a substitution, it cannot be a universal variable because, if it were, we would have the solution to the second problem $x \leftarrow y$ which is not a solution to the first. So we let y be a local variable and the solution $x \leftarrow y$ is now forbidden in both problems because no local variable can occur free in the terms substituted to variables in a substitution.

In Huet's unification algorithm [5] [6] these local variables are always kept in the head of the terms in common abstractions. In Miller's mixed prefixes terminology [8], these local variables are universal variables declared to the right of all the existential variables.

Definition 18 Ground Solution
Let $a = b$ be a problem and σ a solution to $a = b$. The solution σ is *ground* if for each existential variable that has an occurrence in a, the term σx is ground.

Proposition 5 If a matching problem has a solution then it has a ground solution.
Proof Let $a = b$ be a matching problem and σ a solution. Let $y_1 : T_1, ..., y_n : T_n$ the existential variables occurring in the σx for x existential variable of a. Let $u_1, ..., u_n$ be ground terms of the types $T_1, ..., T_n$. Let $\tau = \{< y_1, u_1 >, ..., < y_n, u_n >\}$, and $\sigma' = \tau \circ \sigma$. Obviously, for each existential variable x of a, the term $\sigma' x$ is ground. And $\sigma' a = \tau\sigma a = \tau b = b$. So the problem $a = b$ has a ground solution.

Definition 19 Complete Set of Solutions
Obviously if σ is a solution to a problem $a = b$ then $\tau \circ \sigma$ is one too. A set S of solutions to a problem $a = b$ is said to be *complete* if for every substitution θ solution to this problem there exists a substitution $\sigma \in S$ and a substitution τ such that $\theta = \tau \circ \sigma$.

Lemma 1 Some problems have no finite complete set of solutions.
Proof (Example 1) Consider an atomic type T and an existential variable $x : T \to (T \to T) \to T$. The problem

$$[a : T](x\ a\ [z : T]z) = [a : T]a$$

has an infinite number of independent minimal solutions

$$x \leftarrow [o : T][s : T \to T](s\ ...\ (s\ o)\ ...\ )$$

So in contrast with second order matching [6] [7] there is no (always terminating) algorithm that enumerates all the independent minimal solutions of a third order matching problem.

We consider now algorithms that take as an input a matching problem and either give *one* solution to it or fail if it does not have any.

3 A Bound on the Depth of Solutions

The main idea in this paper is that when we have a matching problem $a = b$ and x is an existential variable occurring in a and t is the term substituted to x by some solution to the problem then the depth of the Böhm tree of t can be bounded by an integer s depending only on the problem $a = b$. Of course the previous example shows that a matching problem may have solutions of arbitrary depth, but to design a decision algorithm we do not need to prove that *all* the solutions are bounded by s but only that *at least one* is.

To show this result we take a problem $a = b$ that has a solution σ (by proposition 5, we can consider without loss of generality that this solution is ground) and we build another solution σ' whose depth is bounded by an integer s depending only on the problem $a = b$.

3.1 Key Lemma

Definition 20 Let $c = [z_1 : U_1]...[z_p : U_p]d$ be a normal term and i an integer, $i \leq p$. We say that c is *relevant* in its i^{th} argument if z_i has an occurrence in the term d.

Lemma 2 (Key Lemma) Let us consider a normal term u, a variable y of type T of order at most two and a normal ground term c of type T.

(1) If y has an occurrence in u then $|c| \leq |u[y \leftarrow c]|$.

(2) If α is an occurrence in the Böhm tree of u such that no occurrence in the path of α is labeled by y, then α is also an occurrence in the normal form of $u[y \leftarrow c]$ and has the same label in the Böhm tree of u and in the Böhm tree of the normal form of $u[y \leftarrow c]$.

(3) If $\alpha = [s_1; ...; s_n]$ is an occurrence in the Böhm tree of u such that for each occurrence $\beta = [s_1; ...; s_k]$ in the path of α, $\beta \neq \alpha$, labeled by y, the term c is relevant in its r^{th} argument where r is the position of the son of β in the path of α i.e. $r = s_{k+1}$, then there exists an occurrence α' of the Böhm tree of the normal form of $u[y \leftarrow c]$ such that all the labels occurring in the path of α, except y, occur in the path of α' and the number of times they occur in the path of α' is greater or equal to the number of times they occur in the path of α.

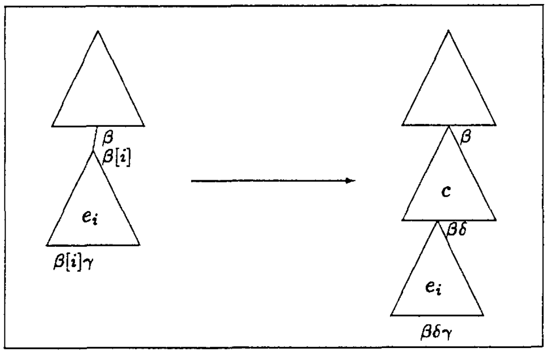
(4) Moreover if $|c| \neq 0$ then the length of α' is greater or equal to the length of α.

Proof By induction on the number of occurrences of y in u. We substitute these occurrences one by one and we normalize the term. Let β be the occurrence in the Böhm tree of u corresponding to the occurrence of y in u we substitute. Let us write

$$c = [z_1 : U_1]...[z_p : U_p]d$$

The term (u/β) has the form $(y\ e_1\ ...\ e_p)$. When we substitute y by the term c in $(y\ e_1\ ...\ e_p)$ we get $(c\ e_1\ ...\ e_p)$ and when we normalize this term we get the term $d[z_1 \leftarrow e_1, ..., z_p \leftarrow e_p]$ which is normal because the type of the e_i are first order.

Let us consider the occurrences in the Böhm tree of u, while substituting the occurrence of y corresponding to β, we have removed all the occurrences $\beta[i]\gamma$ where i is an integer ($i \leq p$) and γ is an occurrence in the Böhm tree of e_i. We have added all the occurrences $\beta\delta$ where δ is an occurrence of the Böhm tree of c labeled by a variable different from $z_1, ..., z_p$ and all the occurrences $\beta\delta\gamma$ where δ is a leaf occurrence in the Böhm tree of c labeled by a z_i and γ is an occurrence of the Böhm tree of e_i.



(1) Let β be an outermost occurrence of y in the Böhm tree of u. For each occurrence δ in the Böhm tree of c, $\beta\delta$ is an occurrence in the Böhm tree of the normal form of $u[y \leftarrow c]$. So $|c| \leq |u[y \leftarrow c]|$.

(2) When an occurrence β of y is substituted by c all the occurrences removed have the form $\beta[i]\gamma$. So if no occurrence in the path of α is labeled by y, the occurrence α remain in the normal form of $u[y \leftarrow c]$.

(3) If the occurrence β is not in the path of α then the occurrence α is still an occurrence in the normal form of $u[y \leftarrow c]$, we take $\alpha' = \alpha$.

If $\beta = \alpha$ then the occurrence β is an occurrence of the Böhm tree of the normal form of $u[y \leftarrow c]$. We take $\alpha' = \beta = \alpha$.

If β is in the path of α and $\beta \neq \alpha$, $\beta = [s_1, ..., s_k]$ then let r be the position of the son of β in the path of α i.e. $r = s_{k+1}$. Let γ such that $\alpha = \beta[r]\gamma$. By hypothesis z_r has an occurrence in d, let δ be such an occurrence. The occurrence $\beta\delta\gamma$ is an occurrence in the Böhm tree of the normal form of $u[y \leftarrow c]$. We take $\alpha' = \beta\delta\gamma$.

In all the cases, all the labels occurring in the path of α, except y, occur in the path of α' and the number of times they occur in the path of α' is greater or equal to the number of times they occur in the path of α.

(4) If $\delta = [\ ]$ then $c = [z_1 : U_1]...[z_p : U_p]z_r$ and $|c| = 0$. So if $|c| \neq 0$ then $\delta \neq [\ ]$ and the length of α' is greater or equal to the length of α.

Corollary Let us consider a normal term u, a variable y of type T of order at most two and a ground term c of type T. If c is relevant in all its arguments and $|c| \neq 0$ then $|u| \leq |u[y \leftarrow c]|$.
Proof We take for α the longest occurrence in the Böhm tree of u. When we substitute one by one the occurrences of y, by part (4) of the key lemma, we get longer occurrences. So there is an occurrence in the Böhm tree of the normal form of $u[y \leftarrow c]$ which is longer than α. So $|u| \leq |u[y \leftarrow c]|$.

3.2 Constraints on the Substitution σ'

First we are going to express some equational constraints that the substitution σ' must verify in order to be a solution to the problem $a = b$. Of course the equation $a = b$ answers the problem, but we need simpler ones: our equations will be on the form $(x\ c_1\ ...\ c_n) = b'$ where x is a variable and $c_1, ..., c_n, b'$ are ground terms.

Definition 21 Let $a = b$ be a problem and σ a (ground) solution to this problem. By induction on the number of occurrences of a we construct a set of equations $\Xi(a = b, \sigma)$.

• If $a = [x : T]d$ then since σ is a solution to the problem $a = b$ we have $b = [x : T]e$ and σ is a solution to the problem $d = e$. We let

$$\Xi(a = b, \sigma) = \Xi(d = e, \sigma)$$

• If $a = (f\ d_1\ ...\ d_n)$ with f universal or local then since σ is a solution to $a = b$ we have $b = (f\ e_1\ ...\ e_n)$ and σ is a solution to the problems $d_i = e_i$. We let

$$\Xi(a = b, \sigma) = \bigcup_i \Xi(d_i = e_i, \sigma)$$

• If $a = (x\ d_1\ ...\ d_n)$ with x existential then for all i such that z has an occurrence in the normal form of the term $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z\ \sigma d_{i+1}\ ...\ \sigma d_n)$ we let $c_i = \sigma d_i$ and $H_i = \Xi(d_i = \sigma d_i, \sigma)$ (obviously σ is a solution to $d_i = \sigma d_i$). Otherwise we let $c_i = z_i$ where z_i is a new local variable and $H_i = \emptyset$. We let

$$\Xi(a = b, \sigma) = \{(x\ c_1\ ...\ c_n) = b\} \cup \bigcup_i H_i$$

Proposition 6 Let $t = (x\ d_1\ ...\ d_n)$ be a term and σ be a substitution. Let $c_i = \sigma d_i$ if z has an occurrence in $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z\ \sigma d_{i+1}\ ...\ \sigma d_n)$ and $c_i = z_i$ new local variable of the same type as d_i otherwise. The variables z_i do not occur in the normal form of $(\sigma x\ c_1\ ...\ c_n)$.
Proof Let us assume that some of these variables have an occurrence in the normal form of $(\sigma x\ c_1\ ...\ c_n)$ and consider outermost occurrence of such a variable z_i in the Böhm tree of the normal form of $(\sigma x\ c_1\ ...\ c_n)$. By part (2) of the key lemma, the variable z_i has also an occurrence in the normal form of term $(\sigma x\ c_1\ ...\ c_n)[z_j \leftarrow \sigma d_j \mid j \neq i]$ i.e. in the normal form of the term $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z_i\ \sigma d_{i+1}\ ...\ \sigma d_n)$, which is contradictory.

Proposition 7 Let $a = b$ be an equation and σ a solution to this equation,

• the substitution σ is a solution to the equations of the set $\Xi(a = b, \sigma)$,

• conversely if σ' is a solution to the equations of $\Xi(a = b, \sigma)$ then σ' is also a solution to the problem $a = b$.

Proof

• By induction on the number of occurrences of a. When a is an abstraction (resp. an atomic term whose head is universal or local) then by induction hypothesis σ is a solution to all the equations of the set $\Xi(d = e, \sigma)$ (resp. $\Xi(d_i = e_i, \sigma)$), so it is a solution to all the equations of $\Xi(a = b, \sigma)$.

When $a = (x\ d_1\ ...\ d_n)$ then by induction hypothesis σ is a solution to all the equations of the $H_i's$ and using the previous proposition the variables z_i have no occurrences in the term $(\sigma x\ c_1\ ...\ c_n)$ so we have

$$(\sigma x\ c_1\ ...\ c_n) = (\sigma x\ c_1\ ...\ c_n)[z_i \leftarrow \sigma d_i]$$

$$(\sigma x\ c_1\ ...\ c_n) = (\sigma x\ \sigma d_1\ ...\ \sigma d_n) = b$$

So σ is a solution to the equation $(x\ c_1\ ...\ c_n) = b$.

• By induction on the number of occurrences of a. Let $a = b$ be a problem and σ a solution to $a = b$. Let σ' be a substitution solution to the equations of $\Xi(a = b, \sigma)$. If a is an abstraction (resp. an atomic term whose head is universal or local) by induction we have $\sigma' d = e$ (resp. $\sigma' d_i = e_i$) and so $\sigma' a = b$.

If $a = (x\ d_1\ ...\ d_n)$ then we have

$$(\sigma' x\ c_1\ ...\ c_n) = b$$

and for all i such that z has an occurrence in $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z\ \sigma d_{i+1}\ ...\ \sigma d_n)$ we have by induction hypothesis $\sigma' d_i = \sigma d_i$, so $c_i = \sigma' d_i$. Therefore

$$(\sigma' x\ c_1\ ...\ c_n)[z_i \leftarrow \sigma' d_i] = b[z_i \leftarrow \sigma' d_i]$$

$$(\sigma' x\ c_1\ ...\ c_n)[z_i \leftarrow \sigma' d_i] = b$$

$$(\sigma' x\ \sigma' d_1\ ...\ \sigma' d_n) = b$$

$$\sigma' a = b$$

Proposition 8 Let $a = b$ be an equation and σ a solution to this equation, if $a' = b'$ is an equation of $\Xi(a = b, \sigma)$ then $|b'| \leq |b|$.

Proof By induction on the number of occurrences of a. When a is an abstraction (reps. an atomic term whose head is universal or local) then by induction hypothesis $|b'| \leq |e|$ (resp. $|b'| \leq |e_i|$) so $|b'| \leq |b|$.

When $a = (x\ d_1\ ...\ d_n)$ and the considered equation is $(x\ c_1\ ...\ c_n) = b$ then we have $b' = b$ so $|b'| \leq |b|$. When the considered equation is in one of the H_i's, the set H_i is non empty so z has an occurrence in the normal form of the term $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z\ \sigma d_{i+1}\ ...\ \sigma d_n)$ and $(\sigma x\ \sigma d_1\ ...\ \sigma d_{i-1}\ z\ \sigma d_{i+1}\ ...\ \sigma d_n)[z \leftarrow \sigma d_i] = b$ so using part (1) of the key lemma we have $|\sigma d_i| \leq |b|$ and by induction hypothesis $|b'| \leq |\sigma d_i|$ so $|b'| \leq |b|$.

3.3 Computing the Substitution σ'

Let us consider an equation $(x\ c_1\ ...\ c_n) = b'$ and

$$t = \sigma x = [y_1 : T_1]...[y_n : T_n]u$$

The normal form of the term $(t\ c_1\ ...\ c_n)$ is the normal form of $u[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]$. If all the c_i are relevant in their arguments and $|c_i| \neq 0$ then using the corollary of the key lemma we have $|t| \leq |(t\ c_1\ ...\ c_n)|$, so $|t| \leq |b'| \leq |b|$ and this gives a bound on the depth of t. But the depth of t may decrease when applied to the c_i's and normalized in two cases:

• if one of the c_i's is not relevant in one of its arguments,
• if one of the c_i's has a null depth.

So solutions may have an arbitrary depth. When this happens, we compute another solution to the problem whose depth is bounded by an integer s depending only on the equation $a = b$. This new substitution is constructed in two steps. In the first we deal with irrelevant terms and in the second with terms of depth 0.

Example 2 Let x be an existential variable of type $T \rightarrow (T \rightarrow T) \rightarrow T$. Consider the problem

$$(x\ a\ [z : T]b) = b$$

The variable z has no occurrence in b so this problem has solutions of arbitrary depth

$$x \leftarrow [o : T][s : T \rightarrow T](s\ t)$$

where t is an arbitrary term of type T. In this example we will compute the substitution

$$x \leftarrow [o : T][s : T \rightarrow T](s\ c)$$

where c is a universal variable.

Example 1 (continued) The term $[z : T]z$ has a depth equal to 0 so we have solutions of an arbitrary depth. In this example we will compute the substitution

$$x \leftarrow [o : T][s : T \rightarrow T](s\ o)$$

Definition 22 Occurrence Accessible with Respect to an Equation of the Form $(x\ c_1\ ...\ c_n) = b'$

Let us consider an equation

$$(x\ c_1\ ...\ c_n) = b'$$

and the term

$$t = \sigma x = [y_1 : T_1]...[y_n : T_n]u$$

Let us consider the Böhm tree of t. The set of the occurrences of the Böhm tree of t *accessible* with respect to the equation $(x\ c_1\ ...\ c_n) = b'$ is inductively defined as:

• the root of the Böhm tree of t is accessible,
• if α is an accessible occurrence labeled by y_i and c_i is relevant in its j^{th} argument then the occurrence $\alpha[j]$ (the j^{th} son of α) is accessible,
• if α is an accessible occurrence labeled by a symbol different from all the y_i then all the sons of α are accessible.

Definition 23 Occurrence Accessible with Respect to a Set of Equations of the Form $(x\ c_1\ ...\ c_n) = b'$

An occurrence is *accessible* with respect to a set of equations of the form $(x\ c_1\ ...\ c_n) = b'$ if it is accessible with respect to one of the equations of the set.

Definition 24 Term Accessible with Respect to a Set of Equations of the Form $(x\ c_1\ ...\ c_n) = b'$

A term is *accessible* with respect to a set of equations of the form $(x\ c_1\ ...\ c_n) = b'$ if all the occurrences of its Böhm tree which are not leaves are accessible with respect to this set of equations.

Definition 25 Accessible Solution Built from a Solution

Let $a = b$ be an equation and σ a solution of this equation. For each existential variable x of a we consider the term $t = \sigma x$. In the Böhm tree of t, we prune all the occurrences non accessible with respect to the equations of $\Xi(a = b, \sigma)$ in which x has an occurrence and put Böhm trees of ground terms of depth 0 of the expected type as leaves. The tree obtained that way is the Böhm tree of some term t'. We let $\hat{\sigma} x = t'$.

Example 2 (continued) From the solution

$$x \leftarrow [o : T][s : T \rightarrow T](s\ t)$$

where t is an arbitrary term, we compute the substitution

$$x \leftarrow [o : T][s : T \rightarrow T](s\ c)$$

where c is a universal variable.

Proposition 9 Let $a = b$ be an equation and σ a solution to this equation, the accessible solution $\hat{\sigma}$ built from σ is a solution to the equation $a = b$.

Proof Let us consider an equation $(x\ c_1\ ...\ c_n) = b'$ of the set $\Xi(a = b, \sigma)$, and the terms

$$\sigma x = t = [y_1 : T_1]...[y_n : T_n]u$$

and

$$\hat{\sigma} x = t' = [y_1 : T_1]...[y_n : T_n]u'$$

We prove by decreasing induction on the depth of the occurrence α of the Böhm tree of u that if α is accessible with respect to the equation $(x\ c_1\ ...\ c_n) = b'$ then α is also an occurrence of the Böhm tree of u' and

$$(u'/\alpha)[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n] = (u/\alpha)[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]$$

and then since the root of u is accessible with respect to this equation we have

$$u'[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n] = u[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]$$

i.e.

$$((\hat{\sigma} x)\ c_1\ ...\ c_n) = b'$$

So $\hat{\sigma}$ is a solution to all the equations of $\Xi(a = b, \sigma)$ and then to the problem $a = b$.

Proposition 10 Let $a = b$ be an equation, σ a solution to this equation, and $\hat{\sigma}$ the accessible solution built from σ. Let

$$t = \hat{\sigma} x = [y_1 : T_1]...[y_n : T_n]u$$

There are at most $h+2$ occurrences of symbols not in $\{y_1, ..., y_n\}$ on a path of the Böhm tree of t, where h is the depth of b.

Proof Let us consider α be an occurrence in the Böhm tree of t such that there are more than $h + 2$ occurrences of symbols not in $\{y_1, ..., y_n\}$ in the path of α.

Let β be the $h + 2^{th}$ occurrence of such a symbol. Since there are more that $h+2$ occurrences of symbols not in $\{y_1, ..., y_n\}$ in the path of α, the occurrence β is not a leaf so it is accessible with respect to some equation $(x\ c_1\ ...\ c_n) = b'$.

For each occurrence $\gamma = [s_1; ...; s_k]$ in the path of β labeled by y_i, let r be the position of the son of this occurrence in this path (i.e. $r = s_{k+1}$), since the occurrence β is accessible with respect to the equation $(x\ c_1\ ...\ c_n) = b'$, the term c_i is relevant in its r^{th} argument. So using part (3) of the key lemma the normal form of the term $b' = (\hat{\sigma} x\ c_1\ ...\ c_n)$ would have a depth greater or equal than $h + 1$ which is contradictory.

Definition 26 Compact Term

A term $t = [y_1 : T_1]...[y_n : T_n]u$ is *compact* with respect to an equation $a = b$ if no variable y_i has more than $h + 1$ occurrences in a path of its Böhm tree, where h is the depth of b.

Proposition 11 Let us consider an equation $a = b$ and a substitution σ solution to this equation. Let $\hat{\sigma}$ be the accessible solution build from σ. Let $h = |b|$.

Let us consider an existential variable x and

$$t = \hat{\sigma} x = [y_1 : T_1]...[y_n : T_n]u$$

Let us consider a variable y_i and an occurrence α of the Böhm tree of t such that there are more that $h+1$ occurrences on the path of α labeled by the variable y_i.

We consider all the equations $(x\ c_1\ ...\ c_n) = b'$ of $\Xi(a = b, \sigma)$ such that the $h + 2^{nd}$ occurrence of y_i is accessible with respect to this equation. Then there exists an integer j such that for every such equation we have

$$c_i = [z_1 : U_1]...[z_p : U_p]z_j$$

Proof Let β be the first occurrence of y_i in the path of α. Let j be the integer such that $\alpha = \beta[j]\beta'$.

Let $(x\ c_1\ ...\ c_n) = b'$ be an equation of $\Xi(a = b, \sigma)$ such that the $h+2^{th}$ occurrence of y_i on the considered path is accessible with respect to this equation.

If the head of c_i is a variable different from a z_k then $|c_i| \neq 0$ and using part (3) of the key lemma when we substitute $c_1, ..., c_{i-1}, c_{i+1}, ..., c_n$ we have an occurrence α' that has more than $h+1$ occurrences of y_i on its path. Then using part (4) of the key lemma, when we substitute c_i we have an occurrence α'' whose length is greater or equal that $h + 1$ so

$$h + 1 \leq |u[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]|$$

i.e. $h + 1 \leq |b'|$ which is contradictory. So we have

$$c_i = [z_1 : U_1]...[z_p : U_p]z_k$$

Since the occurrence β is labeled by y_i and the occurrence $\beta[j]$ is accessible with respect to this equation, the term c_i is relevant in its j^{th} argument. Thus $k = j$ and

$$c_i = [z_1 : U_1]...[z_p : U_p]z_j$$

Definition 27 Compact Accessible Solution Built from an Accessible Solution

Let $a = b$ be an equation, and $h = |b|$. Let $\hat{\sigma}$ be an accessible solution to this equation. We let

$$\hat{\sigma}x = t = [y_1 : T_1]...[y_n : T_n]u$$

For each α, occurrence in t labeled by y_i such that the corresponding occurrence α' in the Böhm tree of t has $h+1$ or more occurrences labeled by y_i in its path, we have $c_i = [z_1 : U_1]...[z_p : U_p]z_j$ in all the equations $(x\ c_1\ ...\ c_n) = b'$ of $\Xi(a = b, \sigma)$ such that α' is accessible with respect to this equation. We substitute the occurrence α by the term $[z_1 : U_1]...[z_p : U_p]z_j$. We get that way a term t'. We let $\sigma'x = t'$.

Example 1 (continued) We build the substitution

$$x \leftarrow [o : T][s : T \rightarrow T](s\ o)$$

Example 3 Consider an existential variable x of type $(T \rightarrow T \rightarrow T) \rightarrow T$. And the equations

$$(x\ [y : T][z : T]y) = a$$

$$(x\ [y : T][z : T]z) = b$$

We have the solution

$$x \leftarrow [f : T \rightarrow T \rightarrow T](f\ a\ (f\ c\ (f\ d\ b)))$$

This solution is accessible but not compact. The first occurrence of f is accessible with respect to both equations, but the second and third occurrences are accessible only with respect to the second one. So we substitute the second and third occurrences of f by the term $[y : T][z : T]z$ and we get the substitution

$$x \leftarrow [f : T \rightarrow T \rightarrow T](f\ a\ b)$$

Remark that we must not substitute the first occurrence of f by $[y : T][z : T]z$, because we would get the substitution $x \leftarrow [f : T \rightarrow T \rightarrow T]b$ which is not a solution to the first equation.

Proposition 12 Let $a = b$ be an equation, σ a solution to this equation, $\hat{\sigma}$ the accessible solution built from σ and σ' the compact accessible solution built from $\hat{\sigma}$. Then σ' is a solution to the equation $a = b$.
Proof We consider an equation $(x\ c_1\ ...\ c_n) = b'$ and we let

$$\hat{\sigma}x = t = [y_1 : T_1]...[y_n : T_n]u$$

and

$$\sigma'x = t = [y_1 : T_1]...[y_n : T_n]u'$$

The term u' is obtained by substituting in the term u some occurrences (say $\beta_1, ..., \beta_k$) by some terms (say $e_1, ..., e_k$). If α is an occurrence of u then we define u'_α as the term obtained by substituting in the term u/α the occurrence γ_i by the term e_i if $\beta_i = \alpha\gamma_i$.

We prove by decreasing induction on the depth of the occurrence α of the Böhm tree of u that if α is accessible with respect to the equation $(x\ c_1 ... c_n) = b'$ then

$$(u'_\alpha)[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n] = (u/\alpha)[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]$$

Thus for the root we get

$$u'[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n] = u[y_1 \leftarrow c_1, ..., y_n \leftarrow c_n]$$

i.e.

$$((\sigma'x)\ c_1\ ...\ c_n) = b'$$

So σ' is a solution to all the equations of $\Xi(a = b, \sigma)$ and then to the problem $a = b$.

Proposition 13 Let $a = b$ be an equation, σ a solution to this equation, $\hat{\sigma}$ the accessible solution built from σ and σ' the compact accessible solution built from $\hat{\sigma}$. For every variable x, $\sigma'x$ has a depth lower than $(n+1)(h+1)$, where h is the depth of b and n the arity of x.
Proof In a path of the Böhm tree of $\sigma'x$ each y_i has at most $h+1$ occurrences and there are at most $h+2$ occurrences of other symbols, so there are at most $(n+1)(h+1)+1$ occurrences. Therefore the depth of $\sigma'x$ is bounded by $(n+1)(h+1)$.

Lemma 3 If a problem $a = b$ has a solution σ then it also has a solution σ' such that for every variable x, σx has a depth lower than $(n+1)(h+1)$, where h is the depth of b and n the arity of x.
Proof The compact accessible solution σ' built from the accessible solution built from the solution σ is a solution and for every variable x, $\sigma'x$ has a depth lower than $(n+1)(h+1)$.

4 A Decision Procedure

Theorem Third Order Matching is Decidable
Proof A decision procedure may be obtained by considering the problem $a = b$ and enumerating all the ground substitutions such that the term substituted to x has a depth lower than $(n+1)(h+1)$, where h the depth of b and n is the arity of x. If one of these substitutions is a solution then succeede else fail. This decision procedure is obviously sound. By lemma 3, it is complete.

Remark This bound on the depth of Böhm trees can also be used to control Huet's semi-decision algorithm [5] [6], and forbid it to consider more than $(n+1)(h+1)$ times the same equation.

Remark This decidabililty result can be compared with the decidability of the equations on the form $P(x_1, ..., x_n) = b$ where P is a polynomial whose coefficients are natural numbers and b is a natural number.

If this equation has a solution $< a_1, ..., a_n >$ then it has a solution $< a'_1, ..., a'_n >$ such that $a'_1 \leq b$. Indeed either $Q(X) = P(X, a_2, ..., a_n)$ is not a constant polynomial and for all n, $Q(n) \geq n$, so $a_1 \leq b$, or the polynomial Q is identically equal to b and $< 0, a_2, ..., a_n >$ is also a solution. So a simple induction on n proves that if the equation has a solution then it also has a solution in $\{0, ..., b\}^n$ and an enumeration of this set gives a decision procedure.

Conclusion: Toward Higher Order Matching

The proof given here is based on the fact that if t is a third order term then when we reduce the term $(t\ c_1\ \ldots\ c_n)$ then *in the general case* we get a term deeper than t (or, at least, if it is not, the depth loss can bounded). This gives a bound (in function of b) on the depth of the solutions of the equation $(x\ c_1\ \ldots\ c_n) = b$. In *the particular cases* in which the depth loss is greater than the bound, some part of the term t is superfluous and that we can construct a smaller term t' such that $(t'\ c_1\ \ldots\ c_n) = (t\ c_1\ \ldots\ c_n)$.

Generalizing this property of reduction to the full λ-calculus would give the decidability of higher order matching. To get the normal form of the term $(t\ c_1\ \ldots\ c_n)$ we have followed the strategy hinted by the weak normalization theorem and reduced first all the second order redexes, then all the first order redexes. So a generalization of this proof to higher order should require an induction on the maximal order of a redex. In the proof for the third order case, we quickly get the normal form of the term $(t\ c_1\ \ldots\ c_n)$ and we do not need to define the depth of a non-normal term. It seems that the generalization of this result to higher order requires such a definition.

Acknowledgments

The author would like to thank Gérard Huet and Richard Statman for many very helpful discussions on this problem. This research was partly supported by ESPRIT Basic Research Action "Logical Frameworks".

References

[1] H. Barendregt, The Lambda Calculus, its Syntax and Semantics, *North Holland*, 1981, 1984.

[2] N.G. de Bruijn, Lambda Calculus Notation with Nameless Dummies, a Tool for Automatic Formula Manipulation, with Application to the Church-Rosser Theorem, *Indagationes Mathematicae*, 34, 5, 1972, pp. 381-392.

[3] S. Gorn, Explicit Definitions and Linguistic Dominoes, *University of Totonto*, 1967.

[4] J.R. Hindley, J.P. Seldin, Introduction to Combinators and λ-Calculus, *Cambridge University Press*, 1986.

[5] G. Huet, A Unification Algorithm for Typed λ-calculus, *Theoretical Computer Science*, 1, 1975, pp. 27-57.

[6] G. Huet, Résolution d'Équations dans les Langages d'Ordre 1,2, ..., ω, *Thèse de Doctorat d'État*, Université de Paris VII, 1976.

[7] G. Huet, B. Lang, Proving and Applying Program Transformations Expressed with Second Order Patterns, *Acta Informatica*, 11, 1978, pp. 31-55.

[8] D. A. Miller, Unification Under a Mixed Prefix, To appear in *Journal of Symbolic Computation.*

[9] R. Statman, Completeness, Invariance and λ-definability, *Journal of Symbolic Logic*, 47, 1, 1982.

[10] D.A. Wolfram, The Clausal Theory of Types, *PhD Thesis*, University of Cambridge, 1989.