Third-Order Matching in the Presence of Type Constructors

Jan Springintveld*
Department of Philosophy
Utrecht University
P.O. Box 80126, 3508 TC Utrecht, The Netherlands

Abstract

We show that it is decidable whether a third-order matching problem in $\lambda\underline{\omega}$ (an extension of the simply typed lambda calculus with type constructors) has a solution or not. We present an algorithm which, given such a problem, returns a solution for this problem if the problem has a solution and returns *fail* otherwise. We also show that it is undecidable whether a third-order matching problem in $\lambda\underline{\omega}$ has a *closed* solution or not.

1 Introduction

It is well-known that type theory is a good basis for the implementation of proof checkers. Although there are various ways to use type theory for proof checking, they all exploit the fact that type theory provides a uniform way to represent and manipulate proofs, formulas and data types.

The man-machine interaction of proof checking can be considerably improved if some kind of matching algorithm can be implemented for the terms of the underlying type theory. For if one wants to prove $\phi(t)$ for a certain formula ϕ and term t, and one already has a proof H of $\forall x\phi(x)$, one would like it to be sufficient to indicate that H should be used without having to mention t. The proof checker should be able to match $\phi(x)$ with $\phi(t)$. Very often one is confronted with situations where t is not just an object (natural number, boolean, ...) but a function, or a functional, etc. To deal with this, one needs an algorithm for *higher-order* matching. So the question becomes: can we find such an algorithm? In other words: is the higher-order matching problem decidable for the underlying type theory? The starting point of this paper is the Calculus of Constructions (CoC), a type theory which features polymorphism, dependent types and type constructors (see [3]). Unfortunately, while second-order matching is decidable in CoC ([4], [6], [7]), third-order matching is undecidable ([4], [5]). Since CoC forms the starting point for many type theoretical studies as well as for concrete implementations (e.g. [13], [15], [11], [16]), it is important to understand why third-order and hence higher-order matching is undecidable in CoC: is it a specific feature (polymorphism, dependent types, type constructors) that is responsible for the undecidability or is the undecidability caused by the interaction of these features? It is precisely for this

*This work is supported by the Netherlands Computer Science Research Foundation (SION) with financial support of the Netherlands Organisation for Scientific Research (NWO).

kind of questions that Barendregt's λ-cube ([2]) was developed. This cube consists of eight systems, each extending the simply typed lambda calculus and each a subsystem of CoC. Starting from the simply typed lambda calculus the cube is erected by three systems, each supporting one of the abovementioned features: λP supports dependent types, $\lambda 2$ supports polymorphism and $\lambda\underline{\omega}$ supports type constructors. The other systems in the cube combine these features in all possible combinations. It is proved in [5] that in λP third-order matching is undecidable. In $\lambda 2$, higher-order matching is undecidable ([10]); it is (to our knowledge) an open question whether matching of finite order is decidable in $\lambda 2$. In $\lambda\omega$, the combination of $\lambda 2$ and $\lambda\underline{\omega}$, fourth-order matching is undecidable. It is (again to our knowledge) open whether third-order matching is decidable in $\lambda\omega$.

In this paper, we show that third-order matching is decidable in $\lambda\underline{\omega}$. In other words, we show that the mere presence of type constructors is not sufficient to make third-order matching undecidable. At first sight, this is not surprising, since $\lambda\underline{\omega}$ is a weak extension of the simply typed lambda calculus and in [8] it is proved that third-order matching is decidable in the simply typed lambda calculus (recently it is proved ([18], [19]) that fourth order matching is decidable as well). But it becomes more surprising when one realizes that the type structure of $\lambda\underline{\omega}$ is rich enough to encode the second-order unification problem in types of order 3. As is well-known, the second-order unification problem is undecidable (see [14]). And in fact, when one demands that solutions for matching problems are *closed*, in the sense that they do not contain *new* free variables, the third-order matching problem becomes undecidable in $\lambda\underline{\omega}$. It is not at all unnatural to demand this. For often one wants that the matching substitution is valid within the context one works and does not contain new variables that have to be instantiated further. The main argument in the proof of the decidability of general third-order matching is that when solutions are allowed to contain new, existential variables, second-order unification can be avoided by defining and solving matching problems in a suitable order.

This paper is organized as follows. In Section 2, we present the two typed lambda calculi with which we shall be concerned in this paper: $\lambda\tau$ (the simply typed lambda calculus with one base type O) and $\lambda\underline{\omega}$. In Section 3, we present terminology concerning matching problems and solutions. We also give an example which illustrates the difficulties that are involved in the proof of our main result. In Section 4, we show that third-order matching is decidable in $\lambda\underline{\omega}$ in case the terms are types. In Section 5, we prove the decidability of third-order matching for objects in a special case; i.e. the case where there is a certain mild restriction on the order of every variable. In the appendix we show how to reduce the general case to the special case. In Section 6 we give a proof of the fact that it is undecidable whether a third-order matching problem in $\lambda\underline{\omega}$ has a closed solution or not. This proof is a variant of the undecidability proofs in [5].

Acknowledgements. I would like to thank Gilles Dowek for his hospitality during my stay at INRIA Rocquencourt and for the fruitful discussions we had there. I also thank Marc Bezem for carefully reading an earlier version of this paper and Jan Friso Groote, Jaco van de Pol and Alex Sellink for critical remarks and helpful suggestions.

2 The systems $\lambda\tau$ and $\lambda\underline{\omega}$

In this section we introduce two typed lambda calculi: $\lambda\tau$, the simply typed lambda calculus with one base type O, and $\lambda\underline{\omega}$, the simply typed lambda calculus with type variables and type constructors (we consider this system with the Conversion Rule for $\beta\eta$-conversion). For more information on these systems the reader is referred to [2] or [13].

Definition 2.1 *(Terms and reductions).* *Pseudo-terms* are given by the following abstract syntax:

$$\mathcal{T} ::= \mathcal{C} \mid \mathcal{V} \mid \mathcal{T}\mathcal{T} \mid \lambda\mathcal{V}{:}\mathcal{T}.\mathcal{T} \mid \mathcal{T}{\to}\mathcal{T}.$$

Here $\mathcal{C}$ is an infinite set of constants and $\mathcal{V}$ is an infinite set of variables; x, y, y', y_1, ... range over $\mathcal{V}$. Among the constants, three elements are singled out: O, $*$ and $\Box$. Roman letters range over $\mathcal{T}$. We define $\mathcal{K}_\Box ::= * \mid \mathcal{K}_\Box{\to}\mathcal{K}_\Box$ and $\mathcal{K}_* ::= O \mid \mathcal{K}_*{\to}\mathcal{K}_*$. We apply the usual conventions concerning brackets; so ABC means $(AB)C$ and $A{\to}B{\to}C$ means $A{\to}(B{\to}C)$. The set of free variables of A is defined as usual and denoted by $FV(A)$. Also the substitution of A for x in B (denoted by $B[x := A]$) and the relations $\to_\beta$, $\twoheadrightarrow_\beta$, $\to_\eta$, $\twoheadrightarrow_\eta$, $\twoheadrightarrow_{\beta\eta}$, $=_\beta$ and $=_{\beta\eta}$ are defined on pseudo-terms as usual. Syntactic equality (modulo α-conversion) is denoted by $\equiv$. The *length* of a term A is the number of symbols that occur in A.

Definition 2.2 *(Contexts and Judgements).* In this paper Q, Q_1, ... range over $\{\exists, \forall\}$. $Qx : B$ is called a *(quantified) declaration.* A *(quantified) pseudo-context* is a finite ordered sequence of quantified declarations $Q_i x_i : C_i$, where the x_i are pairwise distinct. Pseudo-contexts are denoted by capital Greek letters Δ, Γ, Γ_0, ... The empty context is denoted by $\langle\rangle$.

If $\exists x : C$ occurs in Γ, then x is said to be *existential in* Γ. If $\forall x : C$ occurs in Γ, then x is said to be *universal in* Γ. If every declaration in Γ is of the form $\exists x : C$, then Γ is an *existential context.*

The intuition behind the quantification of variables is that universal variables are considered to be constant, in the sense that solutions to matching problems are not allowed to substitute terms for them; substitutions are only allowed to substitute terms for existential variables.

If $\Gamma \equiv \langle Q_1 x_1 : A_1, \ldots, Q_n x_n : A_n\rangle$, then $FV(\Gamma) := \{x_1, \ldots, x_n\} \cup \bigcup_{1\le i\le n} FV(A_i)$, $dom(\Gamma) = \{x_1, \ldots, x_n\}$ and $\Gamma, Qx : B$ denotes $\langle Q_1 x_1 : A_1, \ldots, Q_n x_n : A_n, Qx : B\rangle$. (In general, we denote the concatenation of Γ and Δ by Γ, Δ.) Furthermore, for $1 \le i \le n$, Γ_{x_i} denotes $\langle Q_1 x_1 : A_1, \ldots, Q_{i-1} x_{i-1} : A_{i-1}\rangle$ and Γ^{x_i} denotes $\langle Q_{i+1} x_{i+1} : A_1, \ldots, Q_n x_n : A_n\rangle$. If the declaration $Qx : C$ occurs in Γ, then $\Gamma(x)$ denotes C. We write $\Gamma \subseteq \Delta$ if each $Qx{:}A$ in Γ also occurs in Δ; we call such Δ an *extension of* Γ.

A *judgement* is of the form $\Gamma \vdash A : B$, where Γ is a quantified pseudo-context and A and B are pseudo-terms. When we want to indicate that a judgement is derived in a system $\lambda\circ$, we write $\Gamma \vdash_{\lambda\circ} A : B$. If $\Gamma \vdash_{\lambda\circ} A : B$, then we call A *closed in* Γ if all free variables x in A are universal in Γ and moreover $\Gamma(x)$ is closed in Γ_x. It is easily seen that if A is closed in Γ and Δ is a legal extension of Γ then A is closed in Δ.

A pseudo-term A is called *legal* when there exist a pseudo-context Γ and a pseudo-term B such that $\Gamma \vdash A : B$ or $\Gamma \vdash B : A$. A pseudo-context Γ is called legal when there exist pseudo-terms A and B such that $\Gamma \vdash A : B$.

Axiom	$\langle\rangle \vdash c : s$	if $c : s \in Ax_{\lambda\circ}$
Start	$\dfrac{\Gamma \vdash B : s}{\Gamma, Qx{:}B \vdash x{:}B}$	if $x \notin FV(\Gamma)$
Weakening	$\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma, Qx{:}B' \vdash A : B}$	if $x \notin FV(\Gamma)$
Product	$\dfrac{\Gamma \vdash A_1 : s \quad \Gamma \vdash A_2 : s}{\Gamma \vdash A_1 \to A_2 : s}$	
Application	$\dfrac{\Gamma \vdash A_1 : B_1 \to B_2 \quad \Gamma \vdash A_2 : B_1}{\Gamma \vdash A_1 A_2 : B_2}$	
Abstraction	$\dfrac{\Gamma, Qx{:}A_1 \vdash A_2 : B_2 \quad \Gamma \vdash A_1 \to B_2 : s}{\Gamma \vdash \lambda x{:}A_1.A_2 : A_1 \to B_2}$	
Conversion	$\dfrac{\Gamma \vdash A : B' \quad \Gamma \vdash B : s}{\Gamma \vdash A : B}$	if $B' =_{\beta\eta} B$

Table 1: The rules

Definition 2.3 *(The systems).* The systems $\lambda\tau$ and $\lambda\underline{\omega}$ are defined using the rules in Table 1. Here $Sort_{\lambda\circ} \subseteq \mathcal{C}$ is a set of *sorts* and s ranges over $Sort_{\lambda\circ}$. Furthermore $Ax_{\lambda\circ}$ is a set of statements of the from $c : s$, where $c \in \mathcal{C}$. The systems $\lambda\tau$ and $\lambda\underline{\omega}$ can be obtained from the rules by taking $Sort_{\lambda\tau} = \{*\}$, $Sort_{\lambda\underline{\omega}} = \{*, \Box\}$, $Ax_{\lambda\tau} = \{O : *\}$ and $Ax_{\lambda\underline{\omega}} = \{* : \Box\}$. In this paper we let $\lambda\circ$ range over $\lambda\tau$ and $\lambda\underline{\omega}$ and (except in Table 1) s over $\{*, \Box\}$.

We fix some more notation.

Notation 2.4. Let Γ be $\langle x_1 : A_1, \ldots, x_n : A_n\rangle$ and let Δ be $\langle x_1 : B_1, \ldots, x_n : B_n\rangle$. Then we write $\Gamma \to_\beta \Delta$ if $A_i \to_\beta B_i$, for some i, $1 \leq i \leq n$, and for all $1 \leq j \leq n$, $j \neq i$, we have $A_j \equiv B_j$. Similarly for $\Gamma \to_\eta \Delta$. From this the notions $\Gamma \twoheadrightarrow_\beta \Delta$, $\Gamma \twoheadrightarrow_\eta \Delta$, $\Gamma \twoheadrightarrow_{\beta\eta} \Delta$, $\Gamma =_\beta \Delta$, $\Gamma =_{\beta\eta} \Delta$ can be defined as usual.

We write $\Gamma \subseteq_\Box \Delta$ if for all $Qx : A$ in Γ such that $A \in \mathcal{K}_\Box$ also $Qx : A$ in Δ. So every *type* that is well-typed in Γ is also well-typed in Δ (provided Γ and Δ are legal). From $\subseteq_\Box$, $=_\Box$ is defined as usual. We use the abbreviation $\Gamma \vdash A : B : C$ for $\Gamma \vdash A : B$ and $\Gamma \vdash B : C$.

The following meta-theoretic properties hold in $\lambda\tau$ and $\lambda\underline{\omega}$. The results are taken from [1], [2], [13], and [12].

Theorem 2.5.

1. *(*Substitutivity*) If $A \twoheadrightarrow_{\beta\eta} A'$, then $A[x := B] \twoheadrightarrow_{\beta\eta} A'[x := B]$.*

2. (Substitution Lemma) *Suppose* $\Gamma, Qx : C, \Delta \vdash_{\lambda\circ} A : B$ *and* $\Gamma \vdash_{\lambda\circ} D : C$. *Then* $\Gamma, \Delta[x := D] \vdash_{\lambda\circ} A[x := D] : B[x := D]$.

3. (Strong Normalization). *If* $\Gamma \vdash_{\lambda\circ} A : B$, *then there are no infinite* $\beta\eta$*-reduction sequences starting from* A *or from* B.

4. (Confluence) *If* $\Gamma \vdash_{\lambda\circ} A_1 : B$ *and* $\Gamma \vdash_{\lambda\circ} A_2 : B$ *and* $A_1 =_{\beta\eta} A_2$, *then there exists a term* A_3 *such that* $A_1 \twoheadrightarrow_{\beta\eta} A_3$ *and* $A_2 \twoheadrightarrow_{\beta\eta} A_3$.

5. (Subject Reduction) *If* $\Gamma \vdash_{\lambda\circ} A : B$ *and* $A \twoheadrightarrow_{\beta\eta} A'$, *then* $\Gamma \vdash_{\lambda\circ} A' : B$. *If* $\Gamma \twoheadrightarrow_{\beta\eta} \Gamma'$, *then* $\Gamma' \vdash_{\lambda\circ} A : B$.

6. (Unicity of Types) *If* $\Gamma \vdash_{\lambda\circ} A : B$ *and* $\Gamma \vdash_{\lambda\circ} A : B'$, *then* $B =_{\beta\eta} B'$.

7. (Thinning) *If* $\Gamma \vdash_{\lambda\circ} A : B$, $\Gamma \subseteq \Delta$ *and* Δ *is legal in* $\lambda\circ$, *then* $\Delta \vdash_{\lambda\circ} A : B$.

8. (Strengthening). *If* $\Gamma, x : C, \Delta \vdash_{\lambda\circ} A : B$ *and* $x \notin FV(\Delta) \cup FV(A) \cup FV(B)$, *then* $\Gamma, \Delta \vdash_{\lambda\circ} A : B$.

9. (Permutation) *If* $\Gamma, x : C, y : D, \Delta \vdash_{\lambda\circ} A : B$ *and* $\Gamma \vdash_{\lambda\circ} D : s$, *then* $\Gamma, y : D, x : C, \Delta \vdash_{\lambda\circ} A : B$.

For $\lambda\tau$ *we can replace* $=_{\beta\eta}$ *in (5) by* $\equiv$. *This implies that in* $\lambda\tau$ *the Conversion Rule is redundant.*

By Confluence and Strong Normalization we know that each legal term A has a unique $\beta\eta$-normal form. We denote it by $\mathrm{nf}(A)$. A term A is called *normal* when $\mathrm{nf}(A) \equiv A$. Next, we give some more basic facts concerning the terms in our systems. The proofs of these facts are standard and omitted.

Lemma 2.6.

1. $\Gamma \vdash_{\lambda\underline{\omega}} A : \Box \Leftrightarrow (\Gamma$ *legal and* $A \in \mathcal{K}_\Box)$. *Such* A *are called* kinds. $*$ *is called an* atomic kind*; all other kinds are called* arrow kinds.

2. $\Gamma \vdash_{\lambda\tau} A : s \Leftrightarrow (\Gamma$ *legal and* $s \equiv *$ *and* $A \in \mathcal{K}_*)$.

3. $\Gamma \vdash_{\lambda\circ} A : s$, *then for all* $x \in dom(\Gamma)$ *such that* $\Gamma_x \vdash_{\lambda\circ} \Gamma(x) : *$ *we have:* $x \notin FV(A)$. *So* A *contains no object variables.*

4. *For no* A *it is true that* $\langle\rangle \vdash_{\lambda\underline{\omega}} A : *$. *So there exist no closed types in* $\lambda\underline{\omega}$.

5. *If* $\Gamma \vdash_{\lambda\circ} A : *$ *and* Γ, A *are normal, then* A *is called a* type *and* A *is of one of the three following forms:*

   - O.
   - $xA_1 \ldots A_n$, *for some* $n \geq 0$, *where (for* $0 \leq i \leq n$*)* A_i *is normal and* $\Gamma \vdash_{\lambda\circ} A_i : B_i : \Box$, *for some term* B_i.
   - $A_1 \to A_2$, *where (for* $i = 1, 2$*)* A_i *is normal and* $\Gamma \vdash_{\lambda\circ} A_i : *$.

In the first two cases, A is called an atomic type. *In the third case, A is called an* arrow type.

6. *If $\Gamma \vdash_{\lambda\circ} A : B : *$ and Γ, A are normal, then A is of one of the two following forms:*

   - *$xA_1 \dots A_n$, for some $n \geq 0$, where (for $1 \leq i \leq n$) A_i is normal and $\Gamma \vdash_{\lambda\circ} A_i : B_i : *$ for some term B_i.*
   - *$\lambda x{:}A_1.A_2$, where A_1 is as described in (5), A_2 is normal and $\Gamma, x : A_1 \vdash_{\lambda\circ} A_2 : B_2 : *$, for some term B_2.*

   In the first case, A is called an atomic term. *In the second case, A is called an* abstraction term.

7. *Suppose $\Gamma \vdash_{\lambda\circ} A : s$, where A is normal. Using the brackets convention we can write A uniquely as $A_1 \to \dots \to A_n \to B$, with $n \geq 0$ and B atomic. Unless stated otherwise, we assume that types and kinds are written in this way.*

8. *Suppose $\Gamma \vdash_{\lambda\circ} A : B$. There exist $\Gamma_1, \Gamma_2 \subseteq \Gamma$ such that $\Gamma_1, \Gamma_2 \vdash_{\lambda\circ} A : B$ and for all declarations $Qx : C$ in Γ we have: $Qx : C \in \Gamma_1 \Leftrightarrow C \in \mathcal{K}_\Box$. Γ_1 and Γ_2 are unique up to the order of the declarations. The variables in Γ_1 are called* constructor variables*; the variables in Γ_2 are called* object variables. From now on we assume that contexts are in this form.

It will be convenient to assume that terms are in η-long-β-normal form. This notion is defined below.

Definition 2.7. Suppose $\Gamma \vdash_{\lambda\circ} A : B$, where all terms are normal. We define $\mathrm{lnf}_\Gamma(A)$ by induction.

- $A \in \mathcal{C}$. Then $\mathrm{lnf}_\Gamma(A) \equiv A$.
- $A \equiv \lambda x{:}A_1.A_2$. Then $\mathrm{lnf}_\Gamma(A) \equiv \lambda x{:}\mathrm{lnf}_\Gamma(A_1).\mathrm{lnf}_{\Gamma,\forall x:A_1}(A_2)$.
- $A \equiv A_1 \to A_2$. Then $\mathrm{lnf}_\Gamma(A) \equiv \mathrm{lnf}_\Gamma(A_1) \to \mathrm{lnf}_\Gamma(A_2)$.
- $A \equiv xA_1 \dots A_n$. Write B as $B_1 \to \dots \to B_m \to C$. Then

  $$\mathrm{lnf}_\Gamma(A) \equiv \lambda y_1{:}B'_1 \dots \lambda y_m{:}B'_m.xA'_1 \dots A'_n y'_1 \dots y'_m,$$

  where $B'_i \equiv \mathrm{lnf}_\Gamma(B_i)$ (for $1 \leq i \leq m$), $A'_i \equiv \mathrm{lnf}_\Gamma(A_i)$ (for $1 \leq i \leq n$) and $y'_i \equiv \mathrm{lnf}_{\Gamma,\forall y_i:B_i}(y_i)$ (for $1 \leq i \leq m$).

(For the well-foundedness of this definition, see [7]). This definition is extended to non-normal terms as follows. Suppose $\Gamma \vdash_{\lambda\circ} A : B$. Then $\mathrm{lnf}_\Gamma(A) := \mathrm{lnf}_{\mathrm{nf}(\Gamma)}(\mathrm{nf}(A))$. This definition is justified by Theorem 2.5. If $\Gamma \vdash_{\lambda\circ} A : B$, then A is said to be *in LNF* if $\mathrm{lnf}_\Gamma(A) \equiv A$. Unless stated otherwise, we assume that terms are in LNF.

Lemma 2.6 remains true when we replace 'normal' by 'in LNF'.

Lemma 2.8. *Suppose* $\Gamma \vdash_{\lambda\circ} A : A_1 \to \cdots \to A_n \to B$, *where all terms are in* LNF. *Then* $A \equiv \lambda x_1{:}A_1 \ldots \lambda x_n{:}A_n.xB_1 \ldots B_m$, *for some variable* x *(possibly among* $\{x_1, \ldots, x_n\}$*) and terms* $B_1, \ldots, B_m$ *(in LNF).*

We need a slightly non-standard notion of head normal form for types. First we say what a *domain* is. If A has a subterm of the form $\lambda x{:}A_1.A_2$, then A_1 is called a *domain in* A.

Definition 2.9. In this definition terms are not assumed to be in LNF.

1. Let $\Gamma \vdash_{\lambda\circ} A : s$. Then A is in *head-normal form* (*HNF*) if $A \equiv O$, $A \equiv *$, $A \equiv xA_1 \ldots A_n$ or $A \equiv A_1 \to A_2$ with A_1, A_2 in HNF.
2. A term A is in *D-HNF* if every domain in A is in HNF.

Note that a term in LNF is also in (D)-HNF and that we can speak of (legal) contexts that are normal, in HNF or in LNF.
The following technical lemmas are used in Section 5. The first lemma states that if a term A is closed after substitution of terms of atomic type for universal variables, then A is itself closed. The second lemma states that if a term A after substitution of a second-order term $\lambda y_1{:}S_1 \ldots \lambda y_n{:}S_n.B'$ reduces to a closed term C then B' is itself closed.

Lemma 2.10. *Suppose*

1. $\Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n, \Delta \vdash_{\lambda\circ} A : T$, *where* S_i *is atomic, for each* $1 \leq i \leq n$;
2. $\Gamma, \Delta \vdash_{\lambda\circ} B : T$, *where* B *is closed in* Γ, Δ;
3. $\Gamma \vdash_{\lambda\circ} C_i : S_i$ $(1 \leq i \leq n)$;
4. $A[\vec{x} := \vec{C}] \twoheadrightarrow_{\beta\eta} B$.

Then A *is closed in* $\Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n, \Delta$.

Proof. By induction on the LNF-structure of A. ⊠

Lemma 2.11. *Let* B *be* $\lambda y_1{:}S_1 \ldots \lambda y_n{:}S_n.B'$. *Suppose*

1. $\Gamma, \forall x : S_1 \to \cdots \to S_n \to S, \Delta \vdash_{\lambda\circ} A : T$, *where* $n \geq 0$, S_i *(for* $1 \leq i \leq n$*) and* S *are atomic and* $x \in FV(A)$.
2. $\Gamma \vdash_{\lambda\circ} B : S_1 \to \cdots \to S_n \to S : *$;
3. $\Gamma, \Delta \vdash_{\lambda\circ} C : T$, *where* C *is closed in* Γ, Δ;
4. $A[x := B] \twoheadrightarrow_{\beta\eta} C$.

Then B' *is closed in* $\Gamma, \forall y_1 : S_1, \ldots, \forall y_n : S_n$ *and* S *is closed in* Γ. *Moreover, every variable free in* B' *but not among* $\{y_1, \ldots, y_n\}$ *occurs in* C *(hence in every legal* $\beta\eta$*-expansion of* C*).*

Proof. By induction on the LNF-structure of A, using Lemma 2.10. ⊠

3 Matching problems

This section recalls some definitions from [7], [9].

Definition 3.1. Suppose $\Gamma \vdash_{\lambda\circ} A : s$, where A is in HNF. We define $ord_\Gamma(A)$, the *order of A in Γ*, as follows.

$$ord_\Gamma(A) = \begin{cases} 2 & \text{if } A \equiv * \\ 1 & \text{if } A \equiv xA_1 \dots A_n \ \&\ x \text{ universal in } \Gamma \\ \infty & \text{if } A \equiv xA_1 \dots A_n \ \&\ x \text{ existential in } \Gamma \\ 1 & \text{if } A \equiv O \\ \max(\{1 + ord_\Gamma(A_1), ord_\Gamma(A_2)\}) & \text{if } A \equiv A_1 \to A_2 \end{cases}$$

The definition by cases is O.K. by Lemma 2.6. By convention, $\max(\{n, \infty\}) = \infty$ and $n + \infty = \infty$. When Γ is clear from the context, we simply speak about the order of A. Also if $Qx : A$ appear in Γ, then we sometimes speak about the order of x. For instance, if $ord_\Gamma(\Gamma(x)) = \infty$, then we say that x is a variable of order ∞. Note that if A is closed in Γ, then $ord_\Gamma(A)$ is finite.

Definition 3.2.

1. A *substitution* is a finite set of triples $\langle x_i\,;\,\gamma_i\,;\,M_i\rangle$, such that the x_i are pairwise distinct, γ_i is an existential context and $dom(\gamma_i)$ consists of fresh variables, possibly occurring in M_i. We let σ, σ', τ, ... range over substitutions.

2. If $\langle x\,;\,\gamma\,;\,M\rangle \in \sigma$, then we say that σ *binds* x. Put $dom(\sigma) = \{x \mid x \text{ bound by } \sigma\}$. M is called a *substitution term*, γ a *substitution context*. To indicate that the substitution context is an "auxiliary" context, we denote it by a small Greek letter.

3. A substitution σ is extended to a function on pseudo-terms as follows:

$$\begin{aligned} \sigma(c) &= c \quad \text{for } c \in \mathcal{C} \\ \sigma(x) &= \begin{cases} M & \text{if } \langle x\,;\,\gamma\,;\,M\rangle \in \sigma \\ x & \text{otherwise.} \end{cases} \\ \sigma(A_1A_2) &= \sigma(A_1)\sigma(A_2) \\ \sigma(\lambda x{:}A_1.A_2) &= \lambda x{:}\sigma(A_1).\sigma(A_2) \\ \sigma(A_1 \to A_2) &= \sigma(A_1) \to \sigma(A_2) \end{aligned}$$

   If $\sigma = \{\langle x_i\,;\,\gamma_i\,;\,M_i\rangle \mid 1 \le i \le n\}$, then we sometimes write $\sigma(A)$ as $A[x_1 := M_1, \dots, x_n := M_n]$ or $A[\vec{x} := \vec{M}]$.

4. A substitution σ is extended to a function on pseudo-contexts as follows:

$$\begin{aligned} \sigma(\langle\rangle) &= \langle\rangle \\ \sigma(\Gamma, Qx : C) &= \begin{cases} \sigma(\Gamma), \gamma & \text{if } Q = \exists \text{ and } \langle x\,;\,\gamma\,;\,M\rangle \in \sigma \\ \sigma(\Gamma), Qx : \sigma(C) & \text{otherwise.} \end{cases} \end{aligned}$$

5. Let Γ be a legal context in $\lambda\circ$. Then we call σ *well-typed in* Γ when the following holds:

(a) σ binds no variables that are universal in Γ.

(b) $\sigma(\Gamma)$ is legal in $\lambda\circ$.

(c) For all existential variables x in Γ that are bound by σ we have that $\sigma(\Gamma_x), \gamma \vdash_{\lambda\circ} M : \sigma(\Gamma(x))$, where $\langle x\,;\,\gamma\,;\,M\rangle$ is the unique triple in σ that binds x. In other words, we start with $\Gamma_x \vdash_{\lambda\circ} x : \Gamma(x)$ and end up with $\sigma(\Gamma_x) \vdash_{\lambda\circ} \sigma(x) : \sigma(\Gamma(x))$.

Note that the empty substitution, denoted by $\emptyset$, is well-typed in any legal context.

6. Let σ and τ be substitutions. We define

$$\sigma\circ\tau = \{\langle x\,;\,\sigma(\gamma)\,;\,\sigma(t)\rangle \mid \langle x\,;\,\gamma\,;\,t\rangle \in \tau\} \cup \{\langle x\,;\,\gamma\,;\,t\rangle \in \sigma \mid x \text{ not bound by } \tau\}.$$

7. Let σ be well-typed in some context Γ which is legal in $\lambda\circ$ and suppose $dom(\sigma) \subseteq dom(\Gamma)$. We can write σ uniquely as $\sigma = \sigma_\Box \cup \sigma_*$, where σ_s is the set of triples $\langle x, \gamma, M\rangle$ such that $\Gamma_x \vdash_{\lambda\circ} \Gamma(x) : s$. For a fixed sort s, σ is said to be an s-substitution if for every triple $\langle x\,;\,\gamma\,;\,M\rangle$ we have $\Gamma_x \vdash_{\lambda\circ} \Gamma(x) : s$. (Note that $\sigma_\Box$ and σ_* depend on Γ.)

Remark 3.3. Let Γ be legal and σ be well-typed in Γ. Then by inspection of the definition of substitutions and well-typedness one easily verifies that for all triples $\langle x\,;\,\gamma\,;\,t\rangle$ in σ and all y in $dom(\sigma)$, we have: $y \notin FV(t)$. Given that contexts are assumed to be in the form described in Lemma 2.6 (8) and given Lemma 2.6 (3), it is no restriction to assume that a declaration of the form $x : A$ such that $A \in \mathcal{K}_\Box$ only occurs in a substitution context in $\sigma_\Box$ and that the substitution contexts in $\sigma_\Box$ only contain such declarations. From now on we assume that substitutions satisfy this restriction. It is easy to check that the application of such substitutions to contexts of the form described in Lemma 2.6 (8) yields contexts of the same form.

Definition 3.4.

1. A *unification problem in* $\lambda\circ$ is a triple $\langle\Gamma\,;\,A\,;\,B\rangle$, where Γ is a quantified context such that there exists a term C such that $\Gamma \vdash_{\lambda\circ} A : C$ and $\Gamma \vdash_{\lambda\circ} B : C$. We assume that Γ, A and B are in LNF. If $\Gamma \vdash_{\lambda\circ} C : *$, then we say that $\langle\Gamma\,;\,A\,;\,B\rangle$ is a unification problem *for objects*; if $\Gamma \vdash_{\lambda\circ} C : \Box$, then we say that $\langle\Gamma\,;\,A\,;\,B\rangle$ is a unification problem *for types*.

2. A *matching problem in* $\lambda\circ$ is a unification problem $\langle\Gamma\,;\,A\,;\,B\rangle$ in $\lambda\circ$ such that B is closed in Γ. Note that this implies that C is closed in Γ (recall that C is in LNF!).

3. A unification (or matching) problem $\langle\Gamma\,;\,A\,;\,B\rangle$ is *of order* n if the types of the existential variables in Γ have order at most n in Γ. *In the first five sections of this paper we assume that the type of* every *variable in* Γ *is of finite order. In the appendix we show how the proof of Theorem 5.54 can be adapted to the general case. After Lemma 5.9 it is explained where the proof in Section 5 of this paper goes wrong if our restriction on the types of universal variables is relaxed.*

4. A *solution* for a unification problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\circ$ is a substitution σ, well-typed in Γ, such that $\sigma(A) =_{\beta\eta} \sigma(B)$. (So a solution for a *matching* problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\circ$ is a substitution σ, well-typed in Γ, such that $\sigma(A) =_{\beta\eta} B$.) σ is called a solution for a collection of unification (matching) problems $\{P_i \mid i \in I\}$ if σ is a solution for every P_i. By a standard argument, a set of unification (matching) problems $\{\langle \Gamma\,;\, A_i\,;\, B_i\rangle \mid 1 \leq i \leq n\}$ (for some $n \in \mathbb{N}$) can be encoded as a single unification (matching) problem. We call the unification (matching) problems in such a set Γ-*compatible*.

5. A solution σ for a unification problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\circ$ is *closed* when for all triples $\langle x\,;\, \gamma\,;\, M\rangle$ in σ, $\gamma \equiv \langle\rangle$ and M is closed in $\sigma(\Gamma_x)$. Another option would be to say that a solution σ is closed when we all substitution *contexts* in σ are closed, whereas substitution *terms* need not be closed. The undecidability result involving closed solutions in Section 6 holds for both notions of a closed solution.

Now we give two examples of a third-order matching problem for objects in $\lambda\underline{\omega}$. The first example illustrates the use of new, existential variables in substitutions. The second example illustrates the difficulties which have to be overcome in order to prove the decidability of third-order matching in $\lambda\underline{\omega}$.

Example 3.5. We define $\langle \Gamma\,;\, t_1\,;\, t_2\rangle$, a third-order matching problem for objects in $\lambda\underline{\omega}$. Take

- $\Gamma \equiv \langle \forall A : *, \forall B : *, \forall g : B{\to}B{\to}B, \forall b_1 : B, \forall b_2 : B, \exists f : (A{\to}B){\to}B\rangle$;
- $t_1 \equiv g(f(\lambda x{:}A.b_1))(f(\lambda x{:}A.b_2))$;
- $t_2 \equiv g b_1 b_2$.

One easily checks that every solution for this problem is of the form
$\sigma \;=\; \{\langle f\,;\, \langle \exists x : A\rangle\,;\, \lambda y{:}A{\to}B.yx\rangle\}$,
for some existential variable x.

Example 3.6. We define $\langle \Gamma\,;\, t_1\,;\, t_2\rangle$, a third-order matching problem for objects in $\lambda\underline{\omega}$. Take

- $\Gamma \equiv \langle \forall B : *, \forall X : *{\to}*, \exists A_1 : *, \exists A_2 : *, \forall a_1 : (XB), \forall a_2 : (XB),$
  $\forall g : (XB){\to}(XB){\to}(XB){\to}B, \exists y_1 : (XA_1), \exists y_2 : (XA_1),$
  $\exists f : ((XA_2){\to}(XA_2){\to}(XA_2)){\to}(XA_1){\to}(XA_1){\to}(XB)\rangle$
- $t_1 \equiv g$
  $(f(\lambda x_1{:}(XA_2).\lambda x_2{:}(XA_2).x_1)y_1y_2)$
  $(f(\lambda x_1{:}(XA_2).\lambda x_2{:}(XA_2).x_2)y_1y_2)$
  $(f(\lambda x_1{:}(XA_2).\lambda x_2{:}(XA_2).x_1)y_2y_1)$
- $t_2 \equiv g a_1 a_2 a_2$.

The reader is invited to check that this problem has a unique solution:
$\sigma \;=\; \{\langle A_1\,;\, \langle\rangle\,;\, B\rangle, \langle A_2\,;\, \langle\rangle\,;\, B\rangle, \langle y_1\,;\, \langle\rangle\,;\, a_1\rangle, \langle y_2\,;\, \langle\rangle\,;\, a_2\rangle,$
$\langle f\,;\, \langle\rangle\,;\, \lambda z_1{:}(XB){\to}(XB){\to}(XB).\lambda z_2{:}(XB).\lambda z_3{:}(XB).z_1z_2z_3\rangle\}$.

Note that this matching problem has as only solution a substitution which *unifies* the terms XA_1 and XA_2. For the proof of decidability of third-order matching in $\lambda\underline{\omega}$ it is essential that full (third-order) unification of types can be avoided. For it is in general undecidable whether a third-order unification problem for types has a solution or not (see [14], [4]). The idea is to avoid full unification of types by defining and solving matching problems for types in some specific order.

Let us take a closer look at how unification can be avoided in our example. The variable f has three occurrences in t_1 and the term that is substituted for f each time takes a term whose type is initially $(XA_2)\to(XA_2)\to(XA_2)$ and which after application to two arguments (y_1, y_2 of type XA_1) should yield a term of type XB. So XA_2 has to be matched with XB. This constitutes a matching problem because XB is closed. Thereafter we know that the type of the two arguments (i.e. XA_1) has to be matched with XB. This again constitutes a matching problem. Note that if we would first try to match XA_1 with XA_2 we would be faced with a unification problem.

The order is implemented as follows. First we define (and solve) matching problems that arise when we try to find terms that have to be substituted for those existential variables that have type $S_1\to\cdots\to S_n\to S$, *where S is closed.* (In this case we start with f.) Then we apply the solutions for these matching problems to the types of the other existential variables (in this case y_1 and y_2), in the hope that the respective types (in this case XA_1) become closed (in this case the solution changes XA_1 to XB). If this hope is fulfilled then we define (and solve) the matching problems that arise when we try to find terms that have to be substituted for these variables w.r.t. their (new) types. For completeness we show that if the initial problem has a solution then the matching problems have a solution *and* all existential variables are treated (in case substituting a term for such a variable is essential to obtain a solution). We end this section by stating some properties of matching problems and substitutions.

Lemma 3.7. *Let $P = \langle\Gamma\,;\,A\,;\,B\rangle$ be a matching problem in $\lambda\circ$, σ a solution for P. If $x \in FV(A)$ is existential in Γ, then there exists a $y \in FV(A)$, bound by σ, of type $S_1\to\cdots\to S_n\to S$ in Γ (for some $n \geq 0$, S_1, …, S_n, S) such that S is closed in Γ.*

Proof. Induction on the LNF-structure of A. ⊠

Lemma 3.8.

1. *Suppose $\Gamma \vdash_{\lambda\circ} A : B$ and let σ be well-typed in Γ. Then $\sigma(\Gamma) \vdash_{\lambda\circ} \sigma(A) : \sigma(B)$.*
2. *Suppose $\Gamma \vdash_{\lambda\circ} A : s$ and let σ be well-typed in Γ. Then $ord_{\sigma(\Gamma)}(\sigma(A)) \leq ord_\Gamma(A)$.*
3. *Let Γ be legal, τ well-typed in Γ and σ well-typed in $\tau(\Gamma)$. Then $\sigma\circ\tau$ is well-typed in Γ and $(\sigma\circ\tau)(\Gamma) = \sigma(\tau(\Gamma))$.*

Proof. See [7]. ⊠

Lemma 3.9. *Let $P = \langle\Gamma\,;\,A\,;\,B\rangle$ be a matching problem of finite order in $\lambda\underline{\omega}$. If P has solution σ then it has a solution τ such that for every declaration $\exists x : C$ in a substitution context in τ, C is atomic and of order less than or equal to 2.*

Proof. Let $\exists x : C$ appear in a substitution context in σ. Write $C \equiv C_1 \to \cdots \to C_n \to C'$. By Remark A.4, we may assume that the order of C is finite. Hence the order of C' is 1 or (if $C' \equiv *$) 2. Define $c_x \equiv \lambda x_1{:}C_1 \ldots \lambda x_n{:}C_n.y$, where y is a fresh variable of type C'. Let σ' be the result of the following operation. Replace in the substitution context $\exists x : C$ by $\exists y : C'$ and thereafter replace in all substitution terms and contexts x by c_x. By the Substitution Lemma and the fact that x does not occur in Γ we have that σ' is well-typed in Γ. By Substitutivity, $\sigma'(A) \twoheadrightarrow_{\beta\eta} \mathrm{nf}(B)$. So σ is a solution for P. By iterating this procedure, we obtain the required solution τ. ⊠

Lemma 3.10.

1. *Suppose that $\Gamma \vdash_{\lambda\circ} A : B$ and A is closed in Γ. Let σ be a substitution, well-typed in Γ. Then A is closed in $\sigma(\Gamma)$.*
2. *Let Γ be legal in $\lambda\underline{\omega}$ and σ well-typed in Γ. Suppose furthermore that $\sigma(\Gamma) \vdash_{\lambda\underline{\omega}} A : B : \Box$ and that A is closed in $\sigma(\Gamma)$. Then A is closed in Γ.*
3. *Let Γ be legal in $\lambda\circ$ and σ well-typed in Γ. Then $\sigma_\Box$ is well-typed in Γ and σ_* is well-typed in $\sigma_\Box(\Gamma)$.*
4. *Suppose $\Gamma \vdash_{\lambda\circ} A : B$ and σ is well-typed in Γ. Then $\sigma_*(\sigma_\Box(A)) \equiv \sigma(A)$.*
5. *Let Γ be legal in $\lambda\circ$ and σ well-typed in Γ. Suppose $\sigma(\Gamma) \vdash_{\lambda\circ} A : B$ and A is closed in $\sigma(\Gamma)$. Then already $\sigma_\Box(\Gamma) \vdash_{\lambda\circ} A : B$ and A is closed in $\sigma_\Box(\Gamma)$.*

Proof. For (1): By Lemma 3.8 (1) and the fact that $\sigma(A) \equiv A$, we have that $\sigma(\Gamma) \vdash_{\lambda\underline{\omega}} A : \sigma(B)$. By induction on $\mathrm{lh}(\Gamma) + \mathrm{lh}(A)$ one proves that A is closed in $\sigma(\Gamma)$. The proofs of (2), (4), (5) are straightforward inductions on the structure of A. The proof of (3) is an induction on the length of Γ. ⊠

The previous lemma allows the following useful notation. Suppose $\Gamma \vdash_{\lambda\circ} A : B$ and let σ be well-typed in Γ. Write $\sigma_\Box$ as $\{\langle x_i\,;\,\gamma_i\,;\,S_i\rangle \mid 1 \le i \le n\}$ and σ_* as $\{\langle y_i\,;\,\delta_i\,;\,T_i\rangle \mid 1 \le i \le m\}$. Recall that we sometimes write $\sigma(A)$ as

$$A[x_1 := S_1, \ldots, x_n := S_n, y_1 := T_1, \ldots, y_m := T_m].$$

By Lemma 3.10 (3), (4), we can also write $\sigma(A)$ as

$$A[x_1 := S_1, \ldots, x_n := S_n][y_1 := T_1, \ldots, y_m := T_m],$$

where $A[x_1 := S_1, \ldots, x_n := S_n]$ is legal in $\sigma_\Box(\Gamma)$ and hence in $\sigma(\Gamma)$.

Lemma 3.11. *Suppose $\Gamma \vdash_{\lambda\circ} A : B$. Suppose that B is closed in Γ and that for every $x \in FV(A)$, universal in Γ such that $\Gamma(x)$ is non-atomic, $\Gamma(x)$ is closed in Γ. Let σ be well-typed in Γ and suppose that $\sigma(A)$ is closed in $\sigma(\Gamma)$. Suppose $z \in FV(A)$ is existential in Γ and of type $Z_1 \to \cdots \to Z_k \to Z$, where Z is not closed in Γ. For every occurrence of z there exists a u, bound by σ, such that, for some terms $D_1, \ldots, D_l$, the term $uD_1 \ldots D_l$ has an occurrence in A and, for some $1 \le j \le l$, this occurrence of z is an occurrence in D_j.*

Proof. By induction on the LNF-structure of A. Write $A \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$. We distinguish two cases. The first case is where y is universal in Γ or an x_i $(1 \leq i \leq n)$. The result follows easily from the induction hypothesis. The second case is where y is existential in Γ. We can take y for the required variable u. ⊠

4 Third-order matching for types

As explained in the previous section, the proof of decidability of third-order matching for objects hinges on the possibility to avoid having to solve (third-order) unification problems for types. The strategy is to decompose these unification problems into third-order matching problems for types. This of course only makes sense if it is indeed decidable whether a third-order matching problem for types in $\lambda\underline{\omega}$ has a solution or not. In this section we will, without going into any detail, show that this is the case. The proof of this fact uses a translation $()^-$ from [13] which maps types and constructors from $\lambda\underline{\omega}$ to objects of $\lambda\tau$. This translation preserves $\beta\eta$-reduction and (after extension to contexts) judgements. It is easy to extend this translation further to substitutions and to define a left inverse $()^+$ to this translation (up to β-conversion). This enables us to prove the following result.

Theorem 4.1. *Let $P_1 = \langle \Gamma\,;\, A\,;\, B\rangle$ be a matching problem of order n for types in $\lambda\underline{\omega}$. Then $P_2 = \langle (\Gamma)^-\,;\, (A)^-\,;\, (B)^-\rangle$ is a matching problem of order $n{-}1$ in $\lambda\tau$ and σ is a solution for P_1 iff $(\sigma)^-$ is a solution for P_2.*

Corollary 4.2. *It is decidable whether a third-order matching problem for types in $\lambda\underline{\omega}$ has a solution or not.*

Proof. It is decidable whether a second-order matching problem in $\lambda\tau$ has a solution; see [7]. ⊠

5 Third-order matching for objects

In this section we will show that it is decidable whether a third-order matching problem for objects in $\lambda\underline{\omega}$ has a solution. First we define a translation that maps such a problem $P = \langle \Gamma\,;\, A\,;\, B\rangle$ to a third-order matching problem $|P|$ in $\lambda\tau$ and solutions σ for P to solutions $|\sigma|$ for $|P|$. Then we will prove that a substitution σ is a solution for P iff σ is well-typed in Γ and $|\sigma|$ is a solution for $|P|$. This divides the task of finding solutions for P in two parts: find solutions τ for $|P|$ and see if we can "lift" such solutions to substitutions τ' that are well-typed in Γ and such that $|\tau'| = \tau$. Dowek has shown (in [8]) that to find solutions for $|P|$ it does no harm to restrict one's attention to a search space whose cardinality is bounded by a function value depending only on the size of $|P|$. Given such a solution τ, we will try to lift τ in two stages. First we decorate τ in a straightforward way: given an existential variable x of type $S_1 \to \cdots \to S_n \to S$ in Γ and a triple $\langle x\,;\, \gamma\,;\, t\rangle$ in τ, where $t \equiv \lambda x_1{:}|S_1| \ldots \lambda x_n{:}|S_n|.yt_1 \ldots t_m$, we decorate t to $\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt'_1 \ldots t'_m$, where t'_j is the decorated version of t_j. This procedure need not yield terms that are well-typed in $\lambda\underline{\omega}$. In order to change these terms to

well-typed ones, we define (starting from Γ and τ) a third-order matching problem $Match$ for types in $\lambda\underline{\omega}$ such that if this problem has a solution ρ then the composition of ρ with the decorated substitution is a substitution θ that is well-typed in Γ and such that $|\theta| = \tau$ (hence $|\theta|$ is a solution for $|P|$ and θ is a solution for P).

5.1 Flattening types

We define a map, $|\cdot|$, that replaces all atomic subtypes by O. This map is extended to contexts and substitutions. We show that it preserves judgements, order, $\beta\eta$-reduction, the property of being a matching problem of finite order and the property of being a solution for such a problem. When we say below that a term is in (D-) HNF, then this term is not assumed to be normal or in LNF.

Definition 5.1.

1. Suppose $\Gamma \vdash_{\lambda\circ} A : s$, where A is in HNF. We define $|A|_T$ by induction on the structure of A.

$$\begin{array}{rcl} |*|_T & = & O \\ |O|_T & = & O \\ |xA_1 \cdots A_n|_T & = & O \\ |A_1 \to A_2|_T & = & |A_1|_T \to |A_2|_T \end{array}$$

   By inspecting Definition 2.9, one easily verifies that the case distinction is O.K.

2. Let Γ be legal in $\lambda\circ$ (and in HNF) and suppose that for every domain D in A we have $\Gamma \vdash_{\lambda\underline{\omega}} D : s$ (and D is in HNF). We define $|A|$ by induction on the structure of A.

$$\begin{array}{rcl} |c| & = & c \quad \text{for } c \in \mathcal{C} \\ |x| & = & x \\ |A_1 A_2| & = & |A_1||A_2| \\ |\lambda x{:}A_1.A_2| & = & \lambda x{:}|A_1|_T.|A_2| \\ |A_1 \to A_2| & = & |A_1| \to |A_2| \end{array}$$

3. We extend the definition to contexts in HNF that are legal in $\lambda\circ$.

$$\begin{array}{rcl} |\langle\rangle| & = & \langle\rangle \\ |\Gamma, Qx : A| & = & |\Gamma|, Qx : |A|_T \text{ if } \Gamma \vdash_{\lambda\circ} A : * \\ |\Gamma, Qx : A| & = & |\Gamma| \text{ if } \Gamma \vdash_{\lambda\circ} A : \Box \end{array}$$

4. Let Γ be legal in $\lambda\circ$, σ well-typed in Γ and $dom(\sigma) \subseteq dom(\Gamma)$.

   Then $|\sigma| = \{\langle x\,;\, |\gamma|\,;\, |S|\rangle \mid \langle x\,;\, \gamma\,;\, S\rangle \in \sigma_*\}$. (We need to have $dom(\sigma) \subseteq dom(\Gamma)$, because otherwise σ_* is not defined.)

It is easily seen that if $\Gamma \vdash_{\lambda\tau} A : B$, σ is well-typed in Γ and $dom(\sigma) \subseteq dom(\Gamma)$, then $|\Gamma| \equiv \Gamma$, $|A| \equiv A$, $|B|_T \equiv B$ and $|\sigma| \equiv \sigma$. Because of this, all results listed below are trivially true when we replace $\lambda\underline{\omega}$ by $\lambda\tau$.

Lemma 5.2. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} A : B$, *where* Γ *and* A *are in* (D-) HNF.

1. $|A|$ *is in* D-HNF.
2. *If* A *is normal (resp. in* LNF*), then* $|A|$ *is normal (resp. in* LNF*).*
3. *If* $x \in FV(|A|)$, *then* $x \in FV(A)$.

The same holds for $|A|_T$ *(in case* $B \equiv s$*).*

Proof. (1), (2), (3): by induction on the structure of A. ⊠

Lemma 5.3.

1. *Suppose that* $\Gamma \vdash_{\lambda\underline{\omega}} A : s$ *and that* Γ *and* A *are in* HNF. *Then* $|\Gamma| \vdash_{\lambda\tau} |A|_T : *$.
2. *Suppose that* $\Gamma \vdash_{\lambda\underline{\omega}} A : B : *$ *and* Γ, A *and* B *are in* (D-) HNF. *Then* $|\Gamma| \vdash_{\lambda\tau} |A| : |B|_T : *$.

Proof.

1. By induction, first on the length of Γ and second on the HNF-structure of A and using Lemma 2.6, one proves that $|\Gamma|$ is legal in $\lambda\tau$ and $|A| \in \mathcal{K}_*$. The result then follows from Lemma 2.6 (2).
2. By induction on the structure of A, using (1).

⊠

Lemma 5.4. *Suppose that* $\Gamma \vdash_{\lambda\underline{\omega}} A : *$, *where* Γ *and* A *are in* HNF *and* $ord_\Gamma(A)$ *is finite. Then* $ord_\Gamma(A) = ord_{|\Gamma|}(|A|_T)$.

Proof. By Lemma 5.2 and Lemma 5.3 (1), $ord_{|\Gamma|}(|A|_T)$ is defined. The proof of the lemma proceeds by induction on the HNF-structure of A. ⊠

Corollary 5.5. *Let* $P = \langle \Gamma\,;\, A\,;\, B\rangle$ *be a matching problem of order* n *for objects in* $\lambda\underline{\omega}$. *Then* $\langle |\Gamma|\,;\, |A|\,;\, |B|\rangle$ *is a matching problem of order* n *in* $\lambda\tau$. *We denote it by* $|P|$.

Proof. By Lemma 5.2 (2), Lemma 5.3 (2) and Lemma 5.4. ⊠

Lemma 5.6. *Assume that all terms mentioned in the premises of (1) and (2) are in* (D-) HNF.

1. *Suppose* $\Gamma, \exists x : C, \Delta \vdash_{\lambda\underline{\omega}} A : *$. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} D : C$ *and* $ord_\Gamma(A)$ *is finite. Then* $A[x := D]$ *is in* HNF.
2. *Suppose* $\Gamma, Qx : C, \Delta \vdash_{\lambda\underline{\omega}} A : B : *$ *and* $\Gamma \vdash_{\lambda\underline{\omega}} D : C : *$. *Then* $A[x := D]$ *is in* D-HNF.

Proof. (1), (2): by induction on the (HNF-) structure of A. The intuition for (1) is as follows. A is composed of arrows and atomic types $yA_1 \ldots A_n$. Since the order of A is finite, y will not be existential in Γ, hence will not be x.
In other words, $(yA_1 \ldots A_n)[x := D] \equiv y(A_1[x := D]) \ldots (A_n[x := D])$. The intuition for (2) is that by Lemma 2.6 (3), x does not occur in domains in A. Hence the substitution of D for x does not affect the domains in A. ⊠

Lemma 5.7. *Assume that all terms mentioned in the premises of (1) and (2) are in* (D-) HNF.

1. *Suppose* $\Gamma, Qx : C, \Delta \vdash_{\lambda\underline{\omega}} A : B : *$ *and* $\Gamma \vdash_{\lambda\underline{\omega}} D : C : *$. *Then* $|A[x := D]| \equiv |A|[x := |D|]$.
2. *Suppose* $\Gamma, \exists x : C, \Delta \vdash_{\lambda\underline{\omega}} A : *$, *where* $ord_\Gamma(A)$ *is finite. Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} B : C$. *Then* $|A[x := B]|_T \equiv |A|_T$.

Proof.

1. By induction on the structure of A. The same intuition as for Lemma 5.6 (2) applies.
2. By Lemma 5.6 (1), $|A[x := B]|_T$ is defined, since $A[x := B]$ is in HNF. The proof proceeds by induction on the HNF-structure of A. The same intuition as for Lemma 5.6 (1) applies.

⊠

Lemma 5.8. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} A : B : *$, *where* Γ *and* A *are in* (D-) HNF. *Then*

$(A \twoheadrightarrow_{\beta\eta} A') \Rightarrow |A| \twoheadrightarrow_{\beta\eta} |A'|$.

Proof. By induction on the generation of $\twoheadrightarrow_{\beta\eta}$, using Lemma 5.7. ⊠

The following lemma states that the application of substitutions to terms, types and contexts under certain conditions commutes with the $||$-function. Its validity hinges on our restriction w.r.t. the types of universal variables and the fact that substitutions are (w.l.o.g.) assumed to satisfy the condition in Remark 3.3.

Lemma 5.9. *Suppose* $\Gamma \vdash_{\lambda\circ} A : B : *$, *where* Γ *and* A *are in* (D-) HNF. *Suppose that for all* $y \in dom(\Gamma)$, $\Gamma(y)$ *is of finite order and that every domain in* A *has finite order w.r.t.* Γ. *Let* σ *be well-typed in* Γ, $dom(\sigma) \subseteq dom(\Gamma)$. *Then* $|\sigma(\Gamma)| \equiv |\sigma|(|\Gamma|)$, *and* $|\sigma(A)| \equiv |\sigma|(|A|)$. *If* B *is of finite order, then* $|\sigma(B)| \equiv |\sigma|(|B|) \equiv |B|$.

Proof. By induction on the length of Γ, the structure of A, and the HNF-structure of B respectively. Use Lemma 5.7. ⊠

In the presence of universal variables of order ∞ there may occur situations where the condition on the domains in the previous lemma is not met, thereby preventing the application of that lemma. Let us see why. Let $\Gamma \equiv \langle \forall B : *, \exists X : * \to *, \forall x : (XB \to XB) \to B \rangle$ and let A consist of the variable x. In LNF, $A \equiv \lambda y{:}XB \to XB.x(\lambda z{:}XB.yz)$. Note that x is a variable of order ∞ and XB is a domain of order ∞ in A. Let σ be $\{\langle X\,;\,\langle\rangle\,;\,\lambda\beta{:}*.\beta \to \beta\rangle\}$. We have that $|\sigma| = \emptyset$ and $|\sigma|(|A|) \equiv |A| \equiv \lambda y{:}O \to O.x(\lambda z{:}O.yz)$. Furthermore we have that (in LNF) $\sigma(A) \equiv \lambda y{:}(B \to B) \to B \to B.x(\lambda z{:}B \to B.y(\lambda u{:}B.zu))$ and $|\sigma(A)| \equiv \lambda y{:}(O \to O) \to O \to O.x(\lambda z{:}O \to O.y(\lambda u{:}O.zu))$. So $|\sigma|(|A|) \not\equiv |\sigma(A)|$.

Lemma 5.10. *Suppose that* $\Gamma \vdash_{\lambda\underline{\omega}} \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.xt_1 \ldots t_m : S$, *where* S_i $(1 \leq i \leq n)$ *and* $\Gamma(y)$ *(for all* $y \in FV(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.xt_1 \ldots t_m)$*) are of finite order* w.r.t. Γ. *Then every domain in* $\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.xt_1 \ldots t_m$ *is of finite order* w.r.t. Γ.

Proof. Induction on the LNF-structure of $\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.xt_1 \ldots t_m$. ⊠

Proposition 5.11. *Let* $P_1 = \langle \Gamma\,;\,A\,;\,B\rangle$ *be a matching problem of order* n *for objects in* $\lambda\underline{\omega}$, σ *a solution for* P_1 *and suppose* $dom(\sigma) \subseteq dom(\Gamma)$. *Then* $|\sigma|$ *is a solution for* $P_2 = \langle |\Gamma|\,;\,|A|\,;\,|B|\rangle$.

Proof. Write $\sigma_\Box = \{\langle x_i\,;\,\delta_i\,;\,S_i\rangle \mid 1 \leq i \leq n\}$ and $\sigma_* = \{\langle y_i\,;\,\gamma_i\,;\,T_i\rangle \mid 1 \leq i \leq m\}$. Then $A[\vec{x} := \vec{S}][\vec{y} := \vec{T}] =_{\beta\eta} B$ and so by Confluence there exists a term C such that

$$A[\vec{x} := \vec{S}][\vec{y} := \vec{T}] \twoheadrightarrow_{\beta\eta} C \text{ and } B \twoheadrightarrow_{\beta\eta} C.$$

We need to show that $|A|[\vec{y} := |\vec{T}|] =_{\beta\eta} |B|$. By Lemma 5.8 we have that $|B| \twoheadrightarrow_{\beta\eta} |C|$. So we are done if we can show that $|A|[\vec{y} := |\vec{T}|] \twoheadrightarrow_{\beta\eta} |C|$. We calculate:

$A[\vec{x} := \vec{S}][\vec{y} := \vec{T}] \twoheadrightarrow_{\beta\eta} C$	hence by Lemma 5.10, Lemma 5.8 and Lemma 5.6
$\lvert A[\vec{x} := \vec{S}][\vec{y} := \vec{T}]\rvert \twoheadrightarrow_{\beta\eta} \lvert C\rvert$	hence by Lemma 5.7 (1) and well-typedness of σ
$\lvert A[\vec{x} := \vec{S}]\rvert[\vec{y} := \lvert\vec{T}\rvert] \twoheadrightarrow_{\beta\eta} \lvert C\rvert$	hence by Lemma 5.7 (2)
$\lvert A\rvert[\vec{y} := \lvert\vec{T}\rvert] \twoheadrightarrow_{\beta\eta} \lvert C\rvert$.	

Note that Lemma 5.10 is applicable because the type of A is closed and hence of finite order in Γ. Next we show that $|\sigma|$ is well-typed in $|\Gamma|$. First note that $|\sigma|(|\Gamma|)$ is legal in $\lambda\tau$ by Lemma 5.9. Suppose x is bound by $|\sigma|$ (hence by σ). Let Γ be $\Delta, \exists x : D, \Delta'$. Then $|\Gamma| \equiv |\Delta|, \exists x : |D|, |\Delta'|$. Say that $\langle x\,;\,\gamma\,;\,S\rangle \in \sigma$. We have $\sigma(\Delta), \gamma \vdash_{\lambda\underline{\omega}} S : \sigma(D)$ and so (by Lemma 5.6 (1), Lemma 5.3 (2), Lemma 5.7 (2) and Lemma 5.9), $|\sigma|(|\Delta|), |\gamma| \vdash_{\lambda\tau} |S| : |D|_T$. Thus $|\sigma|$ is well-typed in $|\Gamma|$. We conclude that $|\sigma|$ is a solution for P_2. ⊠

Lemma 5.12. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} A_1 : B : *$ *and* $\Gamma \vdash_{\lambda\underline{\omega}} A_2 : B : *$. *If* Γ, A_1, A_2 *and* B *are normal or in* LNF *and* $|A_1| \equiv |A_2|$, *then* $A_1 \equiv A_2$.

Proof. By induction on the LNF-structure of A_1. Use the fact that domains are syntactically equal since A_1 and A_2 have the same type B and B is normal (or in LNF). ⊠

The following lemma is a key step in the main proof. It allows us to cut up the problem of finding solutions for third-order matching problems in $\lambda\underline{\omega}$ into two relatively easy subproblems.

Lemma 5.13. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. Let σ be a substitution, well-typed in Γ, $dom(\sigma) \subseteq dom(\Gamma)$. Suppose that $|\sigma|$ is a solution for $|P|$. Then σ is a solution for P.*

Proof. It suffices to show that $\sigma(A) \twoheadrightarrow_{\beta\eta} \mathrm{nf}(B)$. Since σ is well-typed in Γ, $\mathrm{nf}(\sigma(A))$ exists. We have $\sigma(A) \twoheadrightarrow_{\beta\eta} \mathrm{nf}(\sigma(A))$. As in the proof of Proposition 5.11, we have $|\sigma(A)| \twoheadrightarrow_{\beta\eta} |\mathrm{nf}(\sigma(A))|$ and by Lemma 5.2 we know that $|\mathrm{nf}(\sigma(A))|$ is normal. By Lemma 5.9 we know $|\sigma|(|A|) \twoheadrightarrow_{\beta\eta} |\mathrm{nf}(\sigma(A))|$. Since $|\sigma|$ is a solution for $|P|$, we have that $|\sigma|(|A|) \twoheadrightarrow_{\beta\eta} \mathrm{nf}(|B|)$. As before, $\mathrm{nf}(|B|) \equiv |\mathrm{nf}(B)|$. We obtain that $|\mathrm{nf}(\sigma(A))| \equiv |\mathrm{nf}(B)|$. Furthermore, B and $\sigma(A)$ have the same type, hence, by Subject Reduction, $\mathrm{nf}(\sigma(A))$ and $\mathrm{nf}(B)$ have the same type. By Lemma 5.12, we infer that $\mathrm{nf}(\sigma(A)) \equiv \mathrm{nf}(B)$. We conclude that σ is a solution for P. ⊠

5.2 Decompositions and standard solutions

In [8], Dowek defines for each third-order matching problem P in $\lambda\tau$ and each solution σ for P, a set $\Phi(P,\sigma)$ of third-order matching problems in $\lambda\tau$. This set is such that for each problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\Phi(P,\sigma)$, A is of the form $xA_1 \ldots A_n$, where x is existential in Γ (we say that *x is a* head *in $\Phi(P,\sigma)$*). σ is a solution for $\Phi(P,\sigma)$ and each solution for $\Phi(P,\sigma)$ is also a solution for P. Using this set, σ is transformed to σ', a substitution which is also a solution for P but more efficient in two ways. First, the substitution terms in σ' are stripped of irrelevant parts and second, σ' does not contain irrelevant triples. Using this efficiency, Dowek shows that it is decidable whether a third-order matching problem in $\lambda\tau$ has a solution or not.

In this section we present a slight generalisation of this "decomposition" to the setting of $\lambda\underline{\omega}$. The generalisation is slight in the sense that we do not claim that a solution for the decomposition is a solution for the initial problem. It would not be difficult to adapt the definition to meet this additional requirement, but since this property is not used in the sequel we have chosen not to do so. The generalisation is instrumental in proving the completeness of the algorithm we define to establish decidability of third-order matching for objects in $\lambda\underline{\omega}$.

From this point on we use the following convention. If $\Gamma \vdash_{\lambda\circ} A : B$ and σ is well-typed in Γ, then we write $\sigma(A)$ for $\mathrm{lnf}_{\sigma(\Gamma)}(\sigma(A))$. Next we state an important technical lemma.

Lemma 5.14. *Suppose $P = \langle \Gamma\,;\, xA_1 \ldots A_n\,;\, B\rangle$ is a third-order matching problem for objects in $\lambda\circ$, where x is existential in Γ; say $\exists x : S_1 \to \cdots \to S_n \to S$ occurs in Γ. Fix $1 \leq i \leq n$. Write $S_i \equiv R_1 \to \cdots \to R_m \to R$. Suppose σ is a solution for P. Write $A_i \equiv \lambda y_1{:}R_1 \ldots \lambda y_m{:}R_m.A_i'$ and put $\Delta \equiv \sigma(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m)$. Consider the normal form of $\sigma(x)\sigma(A_1)\ldots\sigma(A_{i-1})z_i\sigma(A_{i+1})\ldots\sigma(A_n)$, where z_i is a fresh variable of type $\sigma_\Box(S_i)$. Suppose z_i occurs in this normal form. Then $P' = \langle \Delta\,;\, \sigma_\Box(A_i')\,;\, \sigma(A_i')\rangle$ is a third-order matching problem for objects in $\lambda\circ$. Moreover σ_* is a solution for P'.*

Proof. The first point follows from the following observations.

1. By Lemma 3.9, every variable added to Γ by $\sigma_\Box$ has order 2 and by Lemma 3.8, the order of every existential variable in $\sigma_\Box(\Gamma)$ has order less than or equal to 3 in $\sigma_\Box(\Gamma)$.

2. $\Gamma \vdash_{\lambda\circ} A_i : S_i \Rightarrow \Gamma, \forall y_1 : R_1, \ldots, \forall y_m : R_m \vdash_{\lambda\circ} A'_i : R \Rightarrow$ (by Lemma 3.8 (1) and Lemma 3.10 (3)) $\sigma_\Box(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m) \vdash_{\lambda\circ} \sigma_\Box(A'_i) : \sigma_\Box(R)$.

3. $\Gamma, \forall y_1 : R_1, \ldots, \forall y_m : R_m \vdash_{\lambda\circ} A'_i : R \Rightarrow$ (by Lemma 3.8 (1)) $\sigma(\Gamma, \forall y_1 : R_1, \ldots, \forall y_m : R_m) \vdash_{\lambda\circ} \sigma(A'_i) : \sigma(R) \Leftrightarrow \sigma(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m) \vdash_{\lambda\circ} \sigma(A'_i) : \sigma_\Box(R)$.

4. By Lemma 2.11, $\sigma(A'_i)$ is closed in $\sigma(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m)$. (Apply this lemma with $\sigma(\Gamma)$ for Γ, z_i for x, $\langle\rangle$ for Δ, $\sigma(x)\sigma(A_1)\ldots\sigma(A_{i-1})z_i\sigma(A_{i+1})\ldots\sigma(A_n)$ for A, $\sigma(A_i)$ for B, and the normal form of B for C.)

5. $\sigma(A'_i)$ closed in $\sigma(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m) \Rightarrow$ (by Lemma 3.10 (5))
   $\sigma_\Box(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m) \vdash_{\lambda\circ} \sigma(A'_i) : \sigma_\Box(R)$.
   Moreover $\sigma(A'_i)$ is closed in $\sigma_\Box(\Gamma), \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m)$.

The second point follows from Lemma 3.10 (3), (4). ⊠

Definition 5.15. Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem in $\lambda\circ$ and let σ be a solution for P. By induction on the length of $\sigma_\Box(A)$ we define $\Phi(P,\sigma)$, a tree in which some of the nodes are labeled with triples (these triples can be proved to be third-order matching problems). It suffices to distinguish the following cases.

- $A \equiv xA_1\ldots A_n$, where x is universal in Γ. Because σ is a solution for P, $B \equiv xB_1\ldots B_n$ and σ is a solution for $\langle \Gamma\,;\, A_i\,;\, B_i\rangle$, for every $1 \le i \le n$. We let $\Phi(P,\sigma)$ consist of an unlabeled root and subtrees $\Phi(\langle \Gamma\,;\, A_1\,;\, B_1\rangle, \sigma)$, …, $\Phi(\langle \Gamma\,;\, A_n\,;\, B_n\rangle, \sigma)$ (ordered from left to right).

- $A \equiv \lambda x{:}A_1.A_2$. Again because σ is a solution for P, $B \equiv \lambda x{:}A_1.B_2$ and σ is a solution for $\langle \Gamma, \forall x{:}A_1\,;\, A_2\,;\, B_2\rangle$. Put $\Phi(P,\sigma) = \Phi(\langle \Gamma, \forall x{:}A_1\,;\, A_2\,;\, B_2\rangle, \sigma)$.

- $A \equiv xA_1\ldots A_n$, where x is existential in Γ. Suppose $\exists x : S_1{\to}\cdots{\to}S_n{\to}S$ occurs in Γ. Put $\Delta := \sigma_\Box(\Gamma)$. For all i, $1 \le i \le n$, write $A_i \equiv \lambda y_1{:}R_1\ldots\lambda y_m{:}R_m.A'_i$ and consider the normal form of $\sigma(x)\sigma(A_1)\ldots\sigma(A_{i-1})z_i\sigma(A_{i+1})\ldots\sigma(A_n)$, where z_i is a fresh variable of type $\sigma_\Box(S_i)$. If z_i occurs in this normal form, put $C_i = \sigma(A_i)$, $C'_i = \sigma(A'_i)$ and $H_i = \Phi(\langle \Delta, \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m)\,;\, \sigma_\Box(A'_i)\,;\, C'_i\rangle, \sigma)$. ($H_i$ is defined by Lemma 5.14 and the fact that $\sigma_\Box(\sigma_\Box(A)) \equiv \sigma_\Box(A)$). Otherwise put $C_i := z_i$ and H_i a tree consisting of an unlabeled node (we call z_i a *dummy variable*). Let $\{z_{i_1}, \ldots, z_{i_k}\}$ be the set of new variables thus obtained. Now the tree $\Phi(P,\sigma)$ consists of a root labeled with
  $\langle \Delta, \forall z_{i_1} : \sigma_\Box(S_{i_1}), \ldots, \forall z_{i_k} : \sigma_\Box(S_{i_k})\,;\, xC_1\ldots C_n\,;\, B\rangle$ and subtrees H_1, …, H_n (ordered from left to right).

Compare the form of the triples in $\Phi(P,\sigma)$ with the $\forall\exists\forall$ format in [17]. Note that each label of a node in $\Phi(P,\sigma)$ is of the form $\langle \Delta\,;\, xC_1\ldots C_n\,;\, D\rangle$. The phrase *a triple P' in $\Phi(P,\sigma)$* will mean a node p' in $\Phi(P,\sigma)$ labeled with P'. The *depth* of P' in $\Phi(P,\sigma)$ is the number of labeled

nodes in the path from the root to p', not counting p'. The depth of $\Phi(P,\sigma)$ is the maximal depth of a triple in $\Phi(P,\sigma)$. Let $Q = \langle \Delta\,;\, xC_1 \ldots C_n\,;\, D\rangle$ and $Q' = \langle \Delta'\,;\, x'C'_1 \ldots C'_n\,;\, D'\rangle$ be triples in $\Phi(P,\sigma)$. Then we say that *the head x is below the head x'* if Q is below Q'. The tree structure of $\Phi(P,\sigma)$ is only occasionally used; mostly we identify $\Phi(P,\sigma)$ with the set of its labels.

Lemma 5.16. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\circ$ and let σ be a solution for P. Then $\Phi(P,\sigma)$ is a collection of third-order matching problems for objects in $\lambda\circ$.*

Proof. By induction on the length of $\sigma_\Box(A)$. The reasoning is similar to the reasoning in the proof of Lemma 5.14. ⊠

Lemma 5.17.

1. *Let P be an third-order matching problem in $\lambda\circ$ and σ a solution for P. Then σ is a solution for $\Phi(P,\sigma)$.*
2. *Let P be a third-order matching problem in $\lambda\tau$ and σ a solution for P. If τ is a solution for $\Phi(P,\sigma)$, then τ is a solution for P.*

Proof. cf. [8]. ⊠

Definition 5.18. Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$ and let σ be a solution for P. Define $|\Phi(P,\sigma)|$ as the underlying tree of $\Phi(P,\sigma)$ but with each label $\langle \Delta\,;\, C\,;\, D\rangle$ replaced by $\langle |\Delta|\,;\, |C|\,;\, |D|\rangle$. We call $\langle \Delta\,;\, C\,;\, D\rangle$ and $\langle |\Delta|\,;\, |C|\,;\, |D|\rangle$ *corresponding*. Note that the use of the $||$-function is justified by Lemma 5.16.

Lemma 5.19. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem in $\lambda\underline{\omega}$ and let σ be a solution for P, $dom(\sigma) \subseteq dom(\Gamma)$. Then for a suitable choice of variables we have $|\Phi(P,\sigma)| = \Phi(|P|,|\sigma|)$.*

Proof. By induction on the length of $\sigma_\Box(A)$, using the results of Subsection 5.1. ⊠

The phrase "suitable" means the following. When we shift in the construction of $\Phi(P,\sigma)$ from $\Phi(\langle \Gamma\,;\, \lambda x{:}A.B\,;\, \lambda x{:}A.C\rangle,\sigma)$ to $\Phi(\langle \Gamma, \forall x{:}A\,;\, B\,;\, C\rangle,\sigma)$ and we shift in the construction of $\Phi(|P|,|\sigma|)$ from $\Phi(\langle |\Gamma|\,;\, \lambda y{:}|A|.|B|\,;\, \lambda y{:}|A|.|C|\rangle,|\sigma|)$ to $\Phi(\langle |\Gamma|, \forall y{:}|A|\,;\, |B|\,;\, |C|\rangle,|\sigma|)$ *at the same place in the underlying tree*, then the variable x and y should be identical. Likewise, dummy variables added at corresponding places should be chosen identical. Below we will assume that the choice is made "suitably".

If $P = \langle \Gamma\,;\, A\,;\, B\rangle$ is a third-order matching problem for objects in $\lambda\underline{\omega}$ and we (only) have a solution τ for $|P|$ we cannot speak of correspondence in the sense of the previous definition. Still, we need to be and are able to relate matching problems in $\Phi(|P|,\tau)$ to extensions of Γ and subterms of A. Below we develop some terminology for this goal.

Definition 5.20. Suppose $\Gamma \vdash_{\lambda\circ} A : C : *$. By induction on the LNF-structure of A we define a tree, $witn(\Gamma, A)$ (*witn* for *witness*). The nodes of $witn(\Gamma, A)$ are labeled with pairs of contexts and subterms of A. Write $A \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.xA_1 \ldots A_m$. Then $witn(\Gamma, A)$ consists of a root labeled with the pair consisting of $\Gamma' \equiv \Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n$ and $xA_1 \ldots A_m$ and subtrees $witn(\Gamma', A_1)$, ..., $witn(\Gamma', A_m)$ (ordered from left to right).

Lemma 5.21. *Suppose $\Gamma \vdash_{\lambda\circ} A : C : *$. Let the pair Δ, D be the label of a node in $witn(\Gamma, A)$. Then $\Delta \vdash_{\lambda\circ} D : E$ for some term E. Moreover $\Delta =_\square \Gamma$.*

Proof. By induction on the LNF-structure of A. ⊠

Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\circ$ and σ a solution for P. Write T_1 for the underlying tree of $witn(\Gamma, A)$ and write T_2 for the underlying tree of $\Phi(P, \sigma)$. Then it easy to see that T_2 can be viewed as the result of replacing some subtrees in T_1 by leaves. Thus every path starting from the root in T_2 corresponds to a path starting from the root of T_1. This gives a correspondence between triples in $\Phi(P, \sigma)$ and nodes in T_1. Let Δ', C' be the label of the node in T_1 corresponding to a triple $P' = \langle \Delta\,;\, C\,;\, D\rangle$ in $\Phi(P, \sigma)$. The pair Δ', C' is called the *witnessing pair for* P'; Δ' is called the *witnessing context for* P' and C' the *witnessing term* for P'.

Lemma 5.22. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\circ$. Let $P' = \langle \Delta\,;\, C\,;\, D\rangle$ be a triple in $\Phi(P, \sigma)$. We can write $\Delta \equiv \Delta_1, \Delta_2$, where Δ_2 consists of dummy variables added in the construction of $\Phi(P, \sigma)$. Let Δ', C' be the witnessing pair for P'. Then σ is well-typed in Δ' and $\sigma(C') =_{\beta\eta} D$. If $\lambda\circ$ is $\lambda\tau$ then $\Delta_1 \equiv \Delta'$; if $\lambda\circ$ is $\lambda\underline{\omega}$ then $\Delta_1 \equiv \sigma_\square(\Delta')$.*

Proof. By inspection of the definition of $\Phi(P, \sigma)$ and $witn(\Gamma, A)$. ⊠

Lemma 5.23. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. Let σ be a solution for P. Let $\langle \Delta\,;\, C\,;\, D\rangle$ in $\Phi(P, \sigma)$ and $\langle |\Delta|\,;\, |C|\,;\, |D|\rangle$ in $\Phi(|P|, |\sigma|)$ be corresponding. If Δ', C' is the witnessing pair for $\langle \Delta\,;\, C\,;\, D\rangle$ in $\Phi(P, \sigma)$, then $|\Delta'|, |C'|$ is the witnessing pair for $\langle |\Delta|\,;\, |C|\,;\, |D|\rangle$ in $\Phi(|P|, |\sigma|)$.*

Proof. By induction on the length of $\sigma_\square(A)$. ⊠

Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. Write T_1 for the underlying tree of $witn(\Gamma, A)$. When we write T_2 for the underlying tree of $witn(|\Gamma|, |A|)$ then we see that T_2 equals T_1. Let τ be a solution for $|P|$. Let T_3 be the underlying tree of $\Phi(|P|, \tau)$. Since, as we have seen above, each node in T_3 corresponds to a node in T_2, each such node also corresponds to a node in T_1. Now for each triple $\langle \Delta\,;\, C\,;\, D\rangle$ in $\Phi(|P|, \tau)$ we define the *$\lambda\underline{\omega}$-companion to* Δ as the context part of the label of the corresponding node in T_1.

Lemma 5.24. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. Suppose σ is a solution for $|P|$, $dom(\sigma) \subseteq dom(|\Gamma|)$. Let $Q = \langle \Delta\,;\, C\,;\, D\rangle$ be a triple in $\Phi(|P|, \sigma)$ and let Ξ be the $\lambda\underline{\omega}$-companion to Δ. Then $|\Xi| \subseteq \Delta$. If x occurs free in D and is of non-atomic type in Δ then x is universal in Ξ and of closed type in Ξ.*

Proof. By inspection of the definition of $\Phi(|P|, \sigma)$ and $witn(\Gamma, A)$. ⊠

As mentioned before, Dowek used the sets defined above to transform solutions into more efficient solutions. By slightly changing this transformation, these efficient solutions may be assumed to have a certain desirable "standard" form. We will describe this form in several steps.

Definition 5.25. Let Γ be a context legal in $\lambda\circ$. We say that a pseudo-term t is *pre-well-typed in* Γ if

- for some term S we have $|\Gamma| \vdash_{\lambda\tau} |t| : S$ and $|t|$ is in LNF (w.r.t. $|\Gamma|$).
- For every domain D in t which is not inside another domain in t we have $\Gamma \vdash_{\lambda\circ} D : *$.
- For every variable $y \in FV(t)$, not occurring in a domain in t, $\Gamma(y)$ is defined and $\Gamma \vdash_{\lambda\circ} \Gamma(y) : *$.

For example in the context $\langle \forall B : *, \forall X : * \rightarrow *, \exists A_1 : *, \forall g : XB \rightarrow XB \rightarrow XB, \exists x : XB \rangle$, which is legal in $\lambda\underline{\omega}$, the term $\lambda y{:}XA_1.gxy$ is pre-well-typed but not typable. The term becomes typable when we substitute B for A_1.

If $\lambda x_1{:}S_1 . \ldots . \lambda x_n{:}S_n . y t_1 \ldots t_m$ is pre-well-typed in Γ then $\lambda x_2{:}S_2 . \ldots . \lambda x_n{:}S_n . y t_1 \ldots t_m$ is pre-well-typed in $\Gamma, \forall x_1 : S_1$ and, for all $1 \leq i \leq m$, t_i is pre-well-typed in $\Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n$.

The following definition is perhaps a bit hard to swallow. Therefore we first give an informal version. A substitution σ will be callled *standard in* Γ if for all substitution terms t the following holds. All free object variables are either universal and their types are closed (so they are "constants"), or they are existential, but then they function only as dummy arguments: they are not themselves applied to arguments. So their type is atomic (but not necessarily closed). Furthermore we want each such existential variable to have a unique occurrence in σ. If moreover σ is a solution for $P = \langle \Gamma \,;\, A \,;\, B \rangle$ then we call σ a standard solution for P if every universal variable, free in t, occurs in B (otherwise this variable only occurs in dummy arguments which should be replaced by existential variables). In this way, these parts of substitution terms that could be replaced by other terms are of a very simple form. This is essential for us, because, precisely by the "replaceability" of such parts, reasoning about them is in general very hard. (The last clause in the definition below is added for technical reasons.)

Definition 5.26. Let Γ be legal in $\lambda\circ$. Let $t = \lambda x_1{:}S_1 . \ldots . \lambda x_n{:}S_n . y t_1 \ldots t_m$ be pre-well-typed in Γ. Let Z be a set of variables in Γ. Then t is called *standard in* Γ *and* Z if for every $x \in FV(t)$ such that $x \notin Z$ and $\Gamma(x) \notin \mathcal{K}_\Box$ we have that either x is universal in Γ and $\Gamma(x)$ is closed in Γ, or x is existential in Γ, $\Gamma(x)$ is atomic and x has a unique occurrence in t.

We extend the definition to substitutions. Suppose $dom(\sigma) \subseteq dom(\Gamma)$. Then we call σ *standard in* Γ if, first, for all triples $\langle x \,;\, \gamma \,;\, \lambda x_1{:}S_1 . \ldots . \lambda x_n{:}S_n . y t_1 \ldots t_m \rangle$ in σ such that $\Gamma(x) \notin \mathcal{K}_\Box$, $y t_1 \ldots t_m$ is standard in $\sigma(\Gamma_x), \gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n$ and $\{x_1, \ldots, x_n\}$. Moreover we demand that for each such triple $\langle x \,;\, \gamma \,;\, t \rangle$ in σ, we have that γ consists of declarations of the form $\exists z : C$ with C atomic and $C \notin \mathcal{K}_\Box$ and $dom(\gamma) \subseteq FV(t)$. Conversely, if $x \in FV(t)$ is existential in $\sigma(\Gamma_x), \gamma$ and its type is not in $\mathcal{K}_\Box$, then $x \in dom(\gamma)$.

Finally, let $P = \langle \Gamma \,;\, A \,;\, B \rangle$ be a matching problem for objects in $\lambda\circ$ and suppose σ is a solution for P. Then σ is called *a standard solution for* P if σ is standard in Γ and

moreover every triple $\langle x\,;\,\gamma\,;\,\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\rangle \in \sigma$ such that $\Gamma(x) \notin \mathcal{K}_\Box$ satisfies the following two requirements. For all z in $FV(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m)$ such that z is universal in $\sigma(\Gamma_x)$ and $\Gamma(z) \notin \mathcal{K}_\Box$ we have $z \in FV(B)$. For every subterm of the form $x_i s_1 \ldots s_l$, where x_i $(1 \leq i \leq n)$ is of type $S_i \equiv R_1 \to \cdots \to R_l \to R$ and for every $1 \leq j \leq l$ we have that either there is some triple $\langle \Delta\,;\,xC_1 \ldots C_n\,;\,D\rangle$ in $\Phi(P,\sigma)$ such that C_i is relevant in its j^{th} argument and is not a dummy variable or s_j is of the form $\lambda y_1{:}\sigma(R_1) \ldots \lambda y_l{:}\sigma(R_l).e$ where e is of atomic type $\sigma(R)$ in γ.

If P is a matching problem for *types* in $\lambda\underline{\omega}$ then a standard solution for P is defined as a $\Box$-substitution satisfying the criteria for a standard solution given above but with clauses demanding pre-well-typedness, '$\Gamma(z) \notin \mathcal{K}_\Box$', '$\Gamma(x) \notin \mathcal{K}_\Box$' removed from each of the three stages.

Lemma 5.27. *Suppose Γ is legal in $\lambda\underline{\omega}$ and $\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ is standard in Γ and Z. Suppose that for $1 \leq i \leq n$, S_i is closed in Γ. Then t_i is standard in $\Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n$.*

Proof. Immediate. ⊠

Theorem 5.28 *(Dowek).* *Let $P = \langle \Gamma\,;\,A\,;\,B\rangle$ be a third-order matching problem in $\lambda\tau$ or a third-order matching problem for types in $\lambda\underline{\omega}$. From a solution σ for P one can construct a substitution $\sigma^\star$ for P such that*

1. *$\sigma^\star$ is a standard solution for P;*
2. *$x \in dom(\sigma^\star)$ iff x is a head in $\Phi(P,\sigma)$.*

(Note that $\sigma^\star$ depends not only on σ but also on P.) Moreover there exists a set, $Sol(P)$, containing standard solutions for P, such that if P has a solution σ then $\sigma^\star \in Sol(P)$ and such that for some $n \in \mathbb{N}$, computable from P only, $Sol(P)$ can be enumerated in time bounded by n.

Proof. By a slight modification of Dowek's proof. ⊠

Proposition 5.29. *Let $P = \langle \Gamma\,;\,A\,;\,B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. Let σ be a solution for P. The construction of $|\sigma|^\star$ from $|\sigma|$ w.r.t. $|P|$ can be mimicked in σ and yields a substitution $\sigma^\star$ which is well-typed and standard in Γ. Moreover $|\sigma^\star| = |\sigma|^\star$. Note that from this follows by Lemma 5.13 that $\sigma^\star$ is also a standard solution for P and (by Lemma 5.19) that $|\Phi(P,\sigma^\star)| = \Phi(|P|, |\sigma|^\star)$.*

Proof. By inspection of the construction of $|\sigma|^\star$. ⊠

This is the main point where use is being made of *new* existential variables in substitution terms. Dowek's construction of an efficient solution from an arbitrary solution makes use of a variable of type O. From this variable, canonical terms are constructed for every type. In the context of $\lambda\underline{\omega}$ this would mean that we would have for every atomic type a variable of that

type. But the assumption of having such a variable for each type *in a given context* cannot be maintained. For say we have a variable of type XA with X universal and A existential. Then it may be the case that a solution substitutes a term for A and that the construction would need another variable of the resulting type. Such a variable need not be there initially; so we need the possibility to introduce new variables.

5.3 Lifting solutions from $\lambda\tau$ to $\lambda\underline{\omega}$

In this section we describe a way to lift solutions for matching problems in $\lambda\tau$ to solutions for matching problems in $\lambda\underline{\omega}$. We first decorate $\lambda\tau$ terms in a naive way. This procedure need not yield terms that are well-typed in $\lambda\underline{\omega}$ (they will be pre-well-typed). In order to change them into well-typed ones, we define a third-order matching problem for types such that solutions for this matching problem, when applied to the decorated terms, yield terms of the desired type.

Definition 5.30. Suppose $\Gamma \vdash_{\lambda\circ} S : *$ and $\Gamma \vdash_{\lambda\circ} T : *$, S and T in HNF. Then S and T are called *twins* if $|S|_T \equiv |T|_T$. If we write $S \equiv S_1 \to \cdots \to S_n \to S'$ and $T \equiv T_1 \to \cdots \to T_n \to T'$, then the twinness of S and T implies the twinness of S_i and T_i $(1 \leq i \leq n)$. So we can say that S_i *is the twin of* T_i *in* T.

Definition 5.31. Let Γ be legal in $\lambda\underline{\omega}$. Suppose $\Delta \vdash_{\lambda\tau} \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m : S$ and $\Gamma(z)$ and $\Delta(z)$ are twins, for all $z \in FV(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m) \cap dom(\Gamma)$. (Note that $\Gamma \vdash_{\lambda\underline{\omega}} \Gamma(z) : *$.) Suppose $\Gamma \vdash_{\lambda\underline{\omega}} T_1 \to \cdots \to T_n \to T : *$ and that S_i and T_i are twins, for every $1 \leq i \leq n$. Then we define:
$Deco(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_\Gamma = \lambda x_1{:}T_1 \ldots \lambda x_n{:}T_n.t$ and
$Cont(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_\Gamma = \Xi$,
where (putting $\Gamma' \equiv \Gamma, \forall x_1 : T_1, \ldots, \forall x_n : T_n$)

- If for some $R_1, \ldots, R_m, R$ we have $Qy : R_1 \to \cdots \to R_m \to R \in \Gamma'$, then $t \equiv ys_1 \ldots s_m$ and $\Xi \equiv \Xi_1, \ldots, \Xi_m$. Here $s_i \equiv Deco(t_i\,;\, R_i)_{\Gamma'}$ and $\Xi_i = Cont(t_i\,;\, R_i)_{\Gamma'}$, for $1 \leq i \leq m$.
- If for no $R_1, \ldots, R_m, R$ we have that $Qy : R_1 \to \cdots \to R_m \to R \in \Gamma'$, then $t \equiv y$ and $\Xi \equiv \langle \exists y : T \rangle$.

We extend *Deco* to substitutions as follows. Let Γ be a context legal in $\lambda\underline{\omega}$, σ well-typed in $|\Gamma|$ and $dom(\sigma) \subseteq dom(|\Gamma|)$. Let $\langle x\,;\, \gamma\,;\, t \rangle$ be a triple in σ. Then $Deco(\langle x\,;\, \gamma\,;\, t \rangle)_\Gamma = \langle x\,;\, Cont(t\,;\, \Gamma(x))_{\Gamma_x}\,;\, Deco(t\,;\, \Gamma(x))_{\Gamma_x} \rangle$ and $Deco(\sigma)_\Gamma = \{Deco(\langle x\,;\, \gamma\,;\, t \rangle)_\Gamma \mid \langle x\,;\, \gamma\,;\, t \rangle \in \sigma\}$.

We give an example. Let Γ be $\langle \forall B : *, \forall X : * \to *, \exists A_1 : *, \exists A_2 : *, \forall g : XB \to XB \to XB, \exists f : XA_1 \to XA_2 \rangle$. Then Γ is legal in $\lambda\underline{\omega}$. Let σ be $\{\langle f\,;\, \langle \exists x : O \rangle\,;\, \lambda y{:}O.gxy \rangle\}$. Then $Deco(\sigma)_\Gamma = \{\langle f\,;\, \langle \exists x : XB \rangle\,;\, \lambda y{:}XA_1.gxy \rangle\}$. Note that $\lambda y{:}XA_1.gxy$ is not typable in the context $\langle \forall B : *, \forall X : * \to *, \exists A_1 : *, \exists A_2 : *, \forall g : XB \to XB \to XB, \exists x : XB \rangle$, but it is pre-well-typed in this context. The term becomes typable when we substitute B for A_1. Compare this to the example below Definition 5.25. Observe that $\{\langle A_1\,;\, \langle\rangle\,;\, B \rangle, \langle A_2\,;\, \langle\rangle\,;\, B \rangle\} \circ Deco(\sigma)_\Gamma$ is a substitution that is well-typed in Γ.

We list a number of elementary properties of *Deco*.

Lemma 5.32. *Let Γ be legal in $\lambda\underline{\omega}$. Suppose $\Delta \vdash_{\lambda\tau} \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m : S$ and $\Gamma(z)$ and $\Delta(z)$ are twins, for all $z \in FV(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m) \cap dom(\Gamma)$. Suppose $\Gamma \vdash_{\lambda\underline{\omega}} T_1 \to \cdots \to T_n \to T' : *$ and that S_i and T_i are twins, for every $1 \leq i \leq n$. Write $T \equiv T_1 \to \cdots \to T_n \to T'$ and $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$. Now we have:*

1. *If every $x \in FV(t) \backslash dom(\Gamma)$ has a unique occurrence in t then $Deco(t\,;T)_\Gamma$ is pre-well-typed in Γ, $Cont(t\,;T)_\Gamma$. Every variable in $Cont(t\,;T)_\Gamma$ has an atomic type.*

2. *Let $Z \subseteq dom(\Gamma)$. If moreover for all $x \in FV(t) \cap dom(\Gamma)$, $x \notin Z$, we have that x is universal in Γ and $\Gamma(x)$ is closed in Γ_x or x is existential in Γ, $\Gamma(x)$ is atomic and x has a unique occurrence in t, then $Deco(t\,;T)_\Gamma$ is standard in Γ, $Cont(t\,;T)_\Gamma$ and Z.*

3. *If furthermore T is of finite order (w.r.t. Γ) and $\Gamma(z)$ is of finite order (w.r.t. Γ) for all $z \in Z$, then every domain in $Deco(t\,;T)_\Gamma$ is of finite order and every variable in $Cont(t\,;T)_\Gamma$ has a type of order 1 (w.r.t. every legal Ξ ($\Gamma \subseteq_\Box \Xi$)).*

Proof. By induction on the LNF-structure of t. ⊠

Lemma 5.33. *Let $P = \langle \Gamma\,;A\,;B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$. Let Δ be a legal extension of Γ such that all existential variables in Δ are of order at most n.*

1. *$Deco(\sigma)_\Gamma(\Delta)$ is legal in $\lambda\underline{\omega}$ and $Deco(\sigma)_\Gamma(\Delta) =_\Box \Delta$. Moreover every existential variable in $Deco(\sigma)_\Gamma(\Delta)$ is of order at most n.*

2. *$Deco(\sigma)_\Gamma$ is standard in Γ. Furthermore $|Deco(\sigma)_\Gamma| = \sigma$ (up to the order of declarations in the substitution contexts).*

3. *Let τ be a $\Box$-substitution, well-typed in Δ. Then τ is well-typed in $Deco(\sigma)_\Gamma(\Delta)$.*

Proof. (1), (2): by Lemma 5.32. For (2) note that since σ is a standard solution for $|P|$, every universal variable y, free in $\sigma(x)$, (x bound by σ) also occurs free in $|B|$ and hence in B. So $\Gamma(y)$ is closed in Γ. (3): by induction on the length of Γ, using Strengthening and Thinning. ⊠

Lemma 5.34. *Let $P = \langle \Gamma\,;A\,;B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$. Let τ be a $\Box$-substitution, well-typed in Γ. Then $|\tau \circ Deco(\sigma)_\Gamma| = \sigma$ (up to a permutation of declarations in the substitution contexts).*

Proof. By Lemma 5.33 (2), Lemma 5.7 (2) and Lemma 5.32 (3). ⊠

Lemma 5.35. *Suppose $\Gamma \vdash_{\lambda\underline{\omega}} t : S : *$. Then $Deco(|t|\,;S)_\Gamma \equiv t$.*

Proof. By induction on the LNF-structure of t. ⊠

Lemma 5.36. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} T_1 \to \cdots \to T_n \to T : *$ *and* $\Delta \vdash_{\lambda\tau} \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m : S$. *Write* $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ *and* $T' \equiv T_1 \to \cdots \to T_n \to T$. *Suppose that* $\Gamma(y)$ *and* $\Delta(y)$ *are twins, for all* $y \in FV(t) \cap dom(\Gamma)$ *and that* S_i *and* T_i *are twins, for every* $1 \leq i \leq n$. *Let* τ *be a* $\Box$*-substitution, well-typed in* Γ. *Suppose that* T' *and* $\Gamma(y)$ *(for all* $y \in FV(t) \cap dom(\Gamma)$*) are of finite order w.r.t.* Γ. *Then* $\tau(Deco(t\,;\, T')_\Gamma) \equiv Deco(t\,;\, \tau(T'))_{\tau(\Gamma)}$ *and* $\tau(\,Cont(t\,;\, T')_\Gamma) \equiv Cont(t\,;\, \tau(T'))_{\tau(\Gamma)}$. *Moreover* $Deco(t\,;\, T_1 \to \cdots \to T_n \to \tau(T))_\Gamma \equiv Deco(t\,;\, T_1 \to \cdots \to T_n \to T)_\Gamma$.

Proof. By induction on the LNF-structure of t. ⊠

Lemma 5.37. *Write* $\Xi = \Gamma_1, \Delta, \Gamma_2$. *Suppose* $\Xi \vdash_{\lambda\underline{\omega}} t : S : *$, *where* Δ *contains precisely all variables* $x \in FV(t)$ *such that* x *is existential in* Ξ *and* $\Xi(x) \notin \mathcal{K}_\Box$. *Suppose every variable in* $dom(\Delta)$ *is of atomic type and each such variable has a unique occurrence in* t. *Then* $Deco(|t|\,;\, S)_\Xi \equiv Deco(|t|\,;\, S)_{\Gamma_1,\Gamma_2}$ *and* $Cont(|t|\,;\, S)_\Xi \equiv Cont(|t|\,;\, S)_{\Gamma_1,\Gamma_2}$ *(up to the order of declarations).*

Proof. By induction on the LNF-structure of t. ⊠

Lemma 5.38. *Let* Γ *be legal in* $\lambda\underline{\omega}$ *and* σ *standard in* Γ. *Suppose* σ *binds only variables such that their types are of finite order in* Γ. *Then* $\sigma_\Box \circ Deco(|\sigma|)_\Gamma = \sigma$ *(up to the order of declarations in the substitution contexts).*

Proof. It suffices to prove that for all $\langle x\,;\, \gamma\,;\, t\rangle \in \sigma_*$ we have $t \equiv \sigma_\Box(Deco(|t|\,;\, \Gamma(x))_{\Gamma_x})$ and $\gamma \equiv \sigma_\Box(Cont(|t|\,;\, \Gamma(x))_{\Gamma_x})$ (up to the order of declarations). We prove the first point; the second point is completely similar. Let $\langle x\,;\, \gamma\,;\, t\rangle \in \sigma_*$ be given. Then $\langle x\,;\, |\gamma|\,;\, |t|\rangle \in |\sigma|$. By Lemma 3.10 (3), $\sigma_\Box$ is well-typed in Γ. Thus we have:
$\sigma_\Box(Deco(|t|\,;\, \Gamma(x))_{\Gamma_x}) \overset{5.36}{\equiv} Deco(|t|\,;\, \sigma_\Box(\Gamma(x)))_{\sigma_\Box(\Gamma_x)} \overset{5.37}{\equiv} Deco(|t|\,;\, \sigma_\Box(\Gamma(x)))_{\sigma_\Box(\Gamma_x),\gamma} \overset{5.35}{\equiv} t$.
That the lemmas used in the last line are indeed applicable follows easily from standardness of σ. ⊠

Next, we define a function, *Targets*, which given a pre-well-typed term t, a type T and a variable x of some unspecified but atomic type, returns the set of types that x must have if t is to be of type T. Of course, we eventually want this set to be a singleton, but it is convenient not to demand that at this moment. The *Targets* function is used in Definition 5.43.

Definition 5.39. Let Γ be a context legal in $\lambda\underline{\omega}$ and $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ be pre-well-typed in Γ. Suppose that $\Gamma \vdash_{\lambda\underline{\omega}} T_1 \to \cdots \to T_n \to T : *$, where for each i, $1 \leq i \leq n$, S_i and T_i are twins. Let $x \in FV(t)$ be of atomic type in Γ, $\Gamma(x) \notin \mathcal{K}_\Box$. Put $I = \{i \mid x \in FV(t_i), 1 \leq i \leq n\}$ and $\Gamma' = \Gamma, \forall x_1 : S_1, \ldots, \forall x_n : S_n$. Note that by, pre-well-typedness, either y is one of the x_i and $S_i \equiv R_1 \to \cdots \to R_m \to R$ (for some $R_1, \ldots, R_m$, R), or Γ contains a declaration $Qy : R_1 \to \cdots \to R_m \to R$. Thus we can define

$$Targets(x\,;\, \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_\Gamma =$$
$$\begin{cases} \{T\} & \text{if } y \equiv x \\ \bigcup_{i \in I} Targets(x\,;\, t_i\,;\, R_i)_{\Gamma'} & \text{otherwise} \end{cases}$$

It follows from pre-well-typedness that $Targets(x\,;\,t_i\,;\,R_i)_{\Gamma'}$ is defined.

Lemma 5.40. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} t : S : *$, $x \in FV(t)$, $\Gamma(x)$ *atomic*, $\Gamma(x) \notin \mathcal{K}_\Box$.
Then $Targets(x\,;\,t\,;\,S)_\Gamma = \{\Gamma(x)\}$.

Proof. By induction on the LNF-structure of t. ⊠

The next lemma states that in a crucial case $Targets(x\,;\,t\,;\,T_1{\to}\cdots{\to}T_n{\to}T)_\Gamma$ contains only closed types.

Lemma 5.41. *Let* Γ *be legal in* $\lambda\underline{\omega}$ *and suppose* $\Delta \vdash_{\lambda\tau} \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m : S$. *Write* $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$. *Suppose that for all* $z \in FV(t)$ *we have that* z *is universal in* Γ, $\Gamma(z)$ *is a twin of* $\Delta(z)$ *and* $\Gamma(z)$ *is atomic or closed in* Γ. *Write* $T' \equiv T_1{\to}\cdots{\to}T_n{\to}T$. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} T' : *$, *that* T' *is closed in* Γ *and that for all* $1 \le i \le n$, S_i *and* T_i *are twins. (Note that by Lemma 5.32,* $Deco(t\,;\,T')_\Gamma$ *is pre-well-typed in* Γ*). Finally, let* $x \in FV(Deco(t\,;\,T')_\Gamma)$ *be of atomic type in* Γ, $\Gamma(x) \notin \mathcal{K}_\Box$. *Then we have:*

1. *For all* U *in* $Targets(x\,;\,Deco(t\,;\,T')_\Gamma\,;\,T')_\Gamma$ *we have that* $\Gamma \vdash_{\lambda\underline{\omega}} U : *$, U *is closed and atomic in* Γ.
2. *If moreover* Δ *satisfies the same properties w.r.t.* t *as* Γ *and for all* z, *free in* t *and universal in* Γ *such that* $\Gamma(z)$ *is not atomic we have* $\Gamma(z) \equiv \Delta(z)$, *then*
   $Targets(x\,;\,Deco(t\,;\,T')_\Gamma\,;\,T')_\Gamma = Targets(x\,;\,Deco(t\,;\,T')_\Gamma\,;\,T')_\Delta$.

Proof. By induction on the LNF-structure of t.
Write $Deco(t\,;\,T')_\Gamma \equiv \lambda x_1{:}T_1 \ldots \lambda x_n{:}T_n.yt'_1 \ldots t'_m$. If $x \equiv y$, then the result is obviously true. If not then for some i, $1 \le i \le m$, $x \in FV(t'_i)$. Now either $y \equiv x_j$, for some j, $1 \le j \le n$, where $T_j \equiv R_1{\to}\cdots{\to}R_m{\to}R$ or a declaration $\forall y : R_1{\to}\cdots{\to}R_m{\to}R$ ($m \ge 1$) appears in Γ (hence in Δ). In both cases R_k ($1 \le k \le m$) is closed in Γ (resp. Δ) and hence in Γ', where $\Gamma' \equiv \Gamma, \forall x_1{:}T_1 \ldots \forall x_n{:}T_n$ (resp. $\Delta' \equiv \Delta, \forall x_1{:}T_1 \ldots \forall x_n{:}T_n$). Hence by induction hypothesis every U in $Targets(x\,;\,t'_i\,;\,R_i)_{\Gamma'}$ is closed and atomic in Γ' and (since $\forall k, 1 \le k \le n$, $x_k \notin FV(R_i)$) closed and atomic in Γ.
Moreover $Targets(x\,;\,t'_i\,;\,R_i)_{\Gamma'} = Targets(x\,;\,t'_i\,;\,R_i)_{\Delta'}$. ⊠

Next we will sketch the matching problem we need for the proof of decidability of third-order matching for objects in $\lambda\underline{\omega}$; this sketch is made precise in Definition 5.43. First we define an auxiliary notion.

Definition 5.42. Let $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ be legal in $\lambda\circ$. Let $1 \le i \le n$. We call t *relevant in its* i^{th} *argument* if $x_i \in FV(yt_1 \ldots t_m)$.

Let $P = \langle \Gamma\,;\,A\,;\,B\rangle$ be a matching problem for objects in $\lambda\underline{\omega}$ and σ a standard solution for $|P|$. Let x be existential in Γ and suppose x is bound by σ. Let $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ be a subterm of the decorated version of $\sigma(x)$ in Γ (w.r.t. some type) and $T \equiv T_1{\to}\cdots{\to}T_n{\to}T'$ a type, closed in Γ. We assume that t is standard in some extension of Γ and $\{z_1, \ldots, z_k\}$ and

we assume the decorated version of $\sigma(x)$ to be of the form $\lambda z_1{:}Z_1 \ldots \lambda z_k{:}Z_k.s$. We want to define by induction on the length of t a matching problem such that every solution τ for this problem is such that $\tau(t)$ is of type T. Obviously we have to match each S_i with T_i. Now if y is free in t and not in $\{z_1, \ldots, z_k\}$ then, by standardness of t, there are two possibilities for y. Either y is universal in Γ and of closed type $R_1 \to \cdots \to R_m \to R$. Hence we have to match R with T' and by induction we know how to deal with t_j and R_j, for all $1 \leq j \leq m$. Or y is existential in Γ and of atomic type R, hence $m = 0$ and we are done if we can match R with T'. If y is a bound variable x_i, then S_i is of the form $R_1 \to \cdots \to R_m \to R$ and since we have matched S_i with T_i (so we can write $T_i \equiv R'_1 \to \cdots \to R'_m \to R'$) we have by induction a matching problem for t_j and R'_j and we have to match R' with T'. If $y \in \{z_1, \ldots, z_k\}$, say $y \equiv z_i$, then it is not evident which *closed* type R_j we should use as input for the matching problem for t_j and R_j, which is by induction defined. The idea is to look at every matching problem $\langle \Delta\,;\, xD_1 \ldots D_n\,;\, C\rangle$ in $\Phi(|P|, \sigma)$ such that D_i is relevant in its j^{th} argument and not a dummy variable. (In Remark 5.46 we discuss the case where such a matching problem does not exist.) Because σ is a solution for $\Phi(|P|, \sigma)$, D_i will be substituted for z_i and applied to t_1, ..., t_m. From D_i we try to read off the type t_j should have in order that D_i can be applied to t_1, ..., t_m. D_i is a term whose type is of order ≤ 2. So either D_i is of the form $\lambda y_1{:}O \ldots y_m{:}O.y_j$ and we take $R_j \equiv T'$ or D_i is of the form $\lambda y_1{:}O \ldots y_m{:}O.es_1 \ldots s_l$ ($e \not\equiv y_j$, for $1 \leq j \leq m$) and we obtain R_j by the *Targets* function applied to y_j, the decorated version of $es_1 \ldots s_l$ and T'.

Definition 5.43. Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$ and σ a solution for $|P|$. Let x be existential in Γ, $\Gamma(x) \notin \mathcal{K}_\Box$. Write $\Phi = \Phi(|P|, \sigma)$. Let Δ be legal, $\Delta =_\Box \Gamma$. Let $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ be pre-well-typed in Δ. Suppose that $\Delta \vdash_{\lambda\underline{\omega}} T_1 \to \cdots \to T_n \to T : *$, where T_i is a twin of S_i, for every $1 \leq i \leq n$. Put $\Delta' = \Delta, \forall x_1 : T_1, \ldots, \forall x_n : T_n$. Let Z be a set of variables that occur in Δ such that the types of the variables in Z are of order at most 2.

We define $SubMatch(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_{\Delta,Z}$. As a consequence of pre-well-typedness, y is one of the x_i and $S_i \equiv R_1 \to \cdots \to R_m \to R$ (for some R_1,..., R_m, R) or Δ contains a declaration $Qy : R_1 \to \cdots \to R_m \to R$. We distinguish two cases.

1. $y \notin Z$. We define $SubMatch(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_{\Delta,Z} = \{\langle \Gamma\,;\, S_i\,;\, T_i\rangle \mid 1 \leq i \leq n\} \cup \{\langle \Gamma\,;\, R\,;\, T\rangle\} \cup \bigcup_{1 \leq i \leq m} SubMatch(t_i\,;\, R'_i)_{\Delta',Z}$.

   Here R'_i is R_i if $y \in dom(\Delta)$ and R'_i is the twin of R_i in T_j if y is x_j (for some $1 \leq j \leq n$).

2. $y \in Z$, say $y \equiv z_i$.

   Write $\Phi_{x,i,j}$ for
   $\{\langle \Psi\,;\, xD_1 \ldots D_p\,;\, C\rangle \in \Phi \mid D_i$ relevant in its j^{th} argument, not a dummy variable$\}$.
   Suppose $E = \langle \Psi\,;\, xD_1 \ldots D_p\,;\, C\rangle$ is an element of $\Phi_{x,i,j}$. Let Ψ_1 be the $\lambda\underline{\omega}$-companion to Ψ.

   Because of typing reasons it is sufficient to distinguish the following two cases.

   (a) $D_i \equiv \lambda y_1{:}O \ldots \lambda y_k{:}O.y_j$.
   We define $add_{j,E} = \{\langle \Gamma\,;\, R_j\,;\, T\rangle\} \cup SubMatch(t_j\,;\, T)_{\Delta',Z}$.

(b) $D_i \equiv \lambda y_1{:}O \ldots \lambda y_k{:}O.es_1 \ldots s_l$ ($e \not\equiv y_j$, for $1 \leq j \leq k$). Put $\Psi_2 = \Psi_1, \forall y_1 : R_1, \ldots, \forall y_k : R_k$ and $\Theta = \mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, T)_{\Psi_2}\,;\, T)_{\Psi_2}$. It is not difficult to check that Θ is defined. Note that because of the relevance of D_i in its j^{th} argument, Θ is non-empty. Also note that since $\Gamma =_\Box \Psi_2 =_\Box \Delta'$ we have $\Gamma \vdash_{\lambda\underline{\omega}} U : *$ and $\Delta' \vdash_{\lambda\underline{\omega}} U : *$, for all $U \in \Theta$. We define $add_{j,E} = \bigcup_{U\in\Theta}(\mathit{SubMatch}(t_j\,;\, U)_{\Delta',Z} \cup \{\langle \Gamma\,;\, R_j\,;\, U\rangle\})$.

Now define $\mathit{Add}_j = \bigcup_{E\in\Phi_{x,i,j}} add_{j,E}$. In Remark 5.46, the case where $\Phi_{x,i,j}$ is empty is discussed.
Finally, define $\mathit{SubMatch}(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m\,;\, T_1 \to \cdots \to T_n \to T)_{\Delta,Z} =$
$\{\langle \Gamma\,;\, S_i\,;\, T_i\rangle \mid 1 \leq i \leq n\} \cup \{\langle \Gamma\,;\, R\,;\, T\rangle\} \cup \bigcup_{1\leq j\leq m} \mathit{Add}_j$.

Lemma 5.44. *Let P, σ, x, Δ, Z, $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$ and $T \equiv T_1 \to \cdots \to T_n \to T'$ be as in Definition 5.43. Suppose that $T_1 \to \cdots \to T_n \to T'$ is closed in Γ and that the order of existential variables in Δ is at most 3. Suppose furthermore that t is standard in Δ and Z. Then $\mathit{SubMatch}(t\,;\, T)_{\Delta,Z}$ is a finite collection of Γ-compatible third-order matching problems for types.*

Proof. By induction on the length of t. Use Lemma 5.27 w.r.t. $\lambda x_1{:}T_1 \ldots \lambda x_n{:}T_n.yt_1 \ldots t_m$ to verify that the induction hypothesis is applicable (note that this term is standard in Γ!). We treat one (crucial) detail. We have to check that in the definition U is closed in Δ', for all $U \in \mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, T)_{\Psi_2}\,;\, T)_{\Psi_2}$. Since $\Delta' =_\Box \Psi_2$, it suffices to show that U is closed in Ψ_2. This follows from Lemma 5.41 (1). We show that this lemma is applicable. Define $\Psi' \equiv \Psi, \forall y_1{:}O, \ldots, \forall y_k{:}O$. Then $\Psi' \vdash_{\lambda\tau} es_1 \ldots s_l : O$. It is easy to see that Ψ_2 is legal in $\lambda\underline{\omega}$, that T is closed in Ψ_2, that y_j is free in $es_1 \ldots s_l$ and of atomic type in Ψ_2 and that its type is not in $\mathcal{K}_\Box$. Next we check the conditions on the free variables of $es_1 \ldots s_l$. A free variable of $es_1 \ldots s_l$ is either among $\{y_1, \ldots, y_k\}$ or not. The variables in $\{y_1, \ldots, y_k\}$ are universal in Ψ_2 and have atomic type in Ψ_2. From Lemma 2.11 we can deduce that every variable z that occurs free in $es_1 \ldots s_l$ but is not among $\{y_1, \ldots, y_k\}$ occurs in C. (Apply this lemma with, for A, the normal form of $\sigma(x)D_1 \ldots D_{i-1}z_iD_{i+1} \ldots D_p$ (z_i a fresh variable of the appropriate type); Ψ for Γ; z_i for x; $\langle\rangle$ for Δ; D_i for B and the normal form of C for C). So by Lemma 5.24, z is universal in Ψ_1 and $\Gamma(z)$ is closed in Ψ_1, hence in Ψ_2. Finally, we need to show that for all $z \in FV(es_1 \ldots s_l)$ we have that $\Psi_2(z)$ and $\Psi'(z)$ are twins. For the variables in $\{y_1, \ldots, y_k\}$ this is obvious. The remaining free variables are declared in Ψ_1. By Lemma 5.24 we know that $|\Psi_1| \subseteq \Psi$. We are done. ⊠

Next we define *Match*. This definition is a straightforward application of *SubMatch*, except that we want to use previously obtained substitutions in the definition of *Match* and we want to restrict the use of *Submatch* to heads above a fixed depth in $\Phi(|P|, \sigma)$. Hence *Match* gets a substitution and a natural number as extra parameters.

Definition 5.45. Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$ and σ a standard solution for $|P|$. Let ρ be a $\Box$-substitution, well-typed in Γ. Suppose the declaration $\exists x : S_1 \to \cdots \to S_n \to S : *$ occurs in Γ, where $\rho(S)$ is closed in Γ and x is bound by σ and of depth at most d in $\Phi(|P|, \sigma)$. Write $S' \equiv S_1 \to \cdots \to S_n \to \rho(S)$. Let $\mathit{Deco}(\sigma(x)\,;\, S')_\Gamma$ be

$\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$. Put $Z = \{x_1, \ldots, x_n\}$. Put $\Delta = Deco(\sigma)_\Gamma(\Gamma), \forall x_1 : S_1, \ldots, \forall x_n : S_n$. By Lemma 5.33 (1), Δ is legal in $\lambda\underline{\omega}$ and $\Delta =_\Box \Gamma$. By Lemma 5.32 (1), $yt_1 \ldots t_m$ is pre-well-typed in Δ. So we can define $Match(x, \rho, d)_\Gamma = SubMatch(yt_1 \ldots t_m\,;\, \rho(S))_{\Delta,Z} \cup \{\langle \Gamma\,;\, S\,;\, \rho(S)\rangle\}$. Let X be the set of variables x_i of depth at most d in $\Phi(|P|, \sigma)$ such that x_i is bound by σ and $\exists x_i : T_1 \to \cdots \to T_{n_i} \to T_i$ occurs in Γ, where $\rho(T_i)$ is closed in Γ. Put $Match(\sigma, \rho, d)_\Gamma = \bigcup_{x \in X} Match(x, \rho, d)_\Gamma$.

Remark 5.46. There is one subtlety involved in the definition above. Recall that we want a solution τ of $Match(x, \rho, d)_\Gamma$ to be such that $\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(Deco(\sigma(x)\,;\, S')_\Gamma) : \tau(S')$. Suppose that in some recursive call of *SubMatch*, we are in the second case, where the head variable under consideration is an element of Z, say it is z_i and z_i is in this case the head variable of $z_i t'_1 \ldots t'_{m'}$. Write $S_i = V_1 \to \cdots \to V_{m'} \to V$. Suppose now that $\Phi_{x,i,j} = \emptyset$ (for some $1 \leq j \leq m'$). Then the further recursive calls of *Submatch* in this case do not extend *Match* with matching problems to ensure that $\tau(Deco(\sigma)_\Gamma(\Gamma)') \vdash_{\lambda\underline{\omega}} \tau(t'_j) : \tau(V_j)$, where $Deco(\sigma)_\Gamma(\Gamma)'$ is the current context extending $Deco(\sigma)_\Gamma(\Gamma)$. We explain why there is no need to extend it. Write $V_j \equiv W_1 \to \cdots \to W_l \to W$. By (the modification of) Dowek's construction of the standard term $\sigma(x)$, t'_j is in this case of the form $\lambda z_1{:}W_1 \ldots \lambda z_k{:}W_l.u$, where $\exists u : W$ occurs in $Deco(\sigma)_\Gamma(\Gamma)'$. So *every* substitution θ, well-typed in $Deco(\sigma)_\Gamma(\Gamma)'$, satisfies $\theta(Deco(\sigma)_\Gamma(\Gamma)') \vdash_{\lambda\underline{\omega}} \theta(t'_j) : \theta(V_j)$.

Lemma 5.47. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$ and ρ a $\Box$-substitution, well-typed in Γ. Let d be less or equal to the depth of $\Phi(|P|, \sigma)$. Then $Match(\sigma, \rho, d)_\Gamma$ is a finite collection of Γ-compatible third-order matching problems for types.*

Proof. We show that for each $x \in dom(\sigma)$ such that $\exists x : S_1 \to \cdots \to S_n \to S$ occurs in Γ and $\rho(S)$ is closed in Γ and the depth of x in $\Phi(|P|, \sigma)$ is at most d, it is true that $Match(x, \rho, d)_\Gamma$ is a finite collection of Γ-compatible third-order matching problems for types. Since S is legal in Γ and $\rho(S)$ closed in Γ, $\langle \Gamma\,;\, S\,;\, \rho(S)\rangle$ is a third-order matching problem for types in $\lambda\underline{\omega}$. Next, let $Deco(\sigma(x)\,;\, S_1 \to \cdots \to S_n \to \rho(S))_\Gamma$ be $t = \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$. Put $Z = \{x_1, \ldots, x_n\}$. Put $\Delta = Deco(\sigma)_\Gamma(\Gamma), \forall x_1 : S_1, \ldots, \forall x_n : S_n$. We show that $SubMatch(yt_1 \ldots t_m\,;\, \rho(S))_{\Delta,Z}$ is a collection of Γ-compatible third-order matching problems for types in $\lambda\underline{\omega}$.
By Lemma 5.33 (1), we have that Δ is legal in $\lambda\underline{\omega}$ and $\Delta =_\Box \Gamma$ and Δ contains only existential variables of order at most 3. By Lemma 5.33 (2) we have that $yt_1 \ldots t_m$ is standard in Δ and Z. So the required fact follows from Lemma 5.44. ⊠

Lemma 5.48. *Let P, σ, ρ, x, d, S', $t \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1 \ldots t_m$, Z and Δ be as in Definition 5.45. Then for all $u \in FV(yt_1 \ldots t_m) \cap \{x_1, \ldots, x_n\}$, of type $R_1 \to \cdots \to R_k \to R$, there exists a type R' such that $\langle \Gamma\,;\, R\,;\, R'\rangle \in Match(x, \rho, d)_\Gamma$.*

Proof. Induction on the length of t. Here the last clause in the definition of "standard solution" is used. ⊠

We proceed by stating two properties of solutions τ for $Match(\sigma,\rho,d)_\Gamma$. First, τ is well-typed in Γ and $Deco(\sigma)_\Gamma(\Gamma)$. Secondly, the application of τ to decorated terms originating from σ yields terms of the desired type.

Lemma 5.49. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$ and ρ a $\Box$-substitution, well-typed in Γ. Let d be less or equal to the depth of $\Phi(|P|,\sigma)$. Suppose there exists some x, bound by σ, such that $\exists x : S_1{\to}\cdots{\to}S_n{\to}S$ occurs in Γ and $\rho(S)$ is closed in Γ. Let τ be a solution for $Match(\sigma,\rho,d)_\Gamma$. Then τ is well-typed in Γ and hence (by Lemma 5.33 (3)) in $Deco(\sigma)_\Gamma(\Gamma)$.*

Proof. Immediate by the definition of *Match*. ⊠

Lemma 5.50. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$ and ρ a $\Box$-substitution, well-typed in Γ. Let $\langle x\,;\,\gamma\,;\,t\rangle \in \sigma$, where $\exists x : S_1{\to}\cdots{\to}S_n{\to}S$ occurs in Γ and $\rho(S)$ is closed in Γ. Let τ be a solution for $Match(\sigma,\rho,d)_\Gamma$.*
Then $\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}S)_\Gamma) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.

Proof. Below we prove
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}\rho(S))_\Gamma) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$ (†).
Since τ is by assumption a solution for $\langle\Gamma\,;\,S\,;\,\rho(S)\rangle$, we can write this as
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}\tau(S))_\Gamma) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.
From this follows by Lemma 5.36
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} Deco(t\,;\, \tau(S_1{\to}\cdots{\to}S_n{\to}\tau(S)))_{\tau(\Gamma)} : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.
This is easily seen to be equivalent to
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} Deco(t\,;\, \tau(S_1{\to}\cdots{\to}S_n{\to}S))_{\tau(\Gamma)}) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.
Using Lemma 5.36 in the reverse direction we get
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}S)_\Gamma) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.

As to (†). Write $Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}\rho(S))_\Gamma \equiv \lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1\ldots t_m$. Let Z be $\{x_1,\ldots,x_n\}$. Let s be a subterm of $yt_1\ldots t_m$ (not a domain), such that $|s|$ is in LNF and s is pre-well-typed in some extension Δ of $Deco(\sigma)_\Gamma(\Gamma)$. Suppose that T is closed in Δ. Suppose furthermore that s is standard in Δ and Z. Let θ be a solution for $SubMatch(s\,;\,T)_{\Delta,Z}$. By induction on the length of s one proves: $\theta(\Delta) \vdash_{\lambda\underline{\omega}} \theta(s) : T$. This induction is straightforward, when one keeps Remark 5.46 in mind. Note that since $\rho(S)$ is closed in Γ, $\rho(S)$ is closed in $Deco(\sigma)_\Gamma(\Gamma)$. By taking $yt_1\ldots t_m$ for s, $\rho(S)$ for T and $Deco(\sigma)_\Gamma(\Gamma), \forall x_1 : S_1, \ldots, \forall x_n : S_n$ for Δ we get:
$\tau(Deco(\sigma)_\Gamma(\Gamma), \forall x_1 : S_1, \ldots, \forall x_n : S_n) \vdash_{\lambda\underline{\omega}} \tau(yt_1\ldots t_m) : \rho(S)$.
Since τ is also a solution for $\langle\Gamma\,;\,S\,;\,\rho(S)\rangle$, we can conclude
$\tau(Deco(\sigma)_\Gamma(\Gamma)) \vdash_{\lambda\underline{\omega}} \tau(\lambda x_1{:}S_1 \ldots \lambda x_n{:}S_n.yt_1\ldots t_m) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$. ⊠

Corollary 5.51. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a standard solution for $|P|$ and ρ a $\Box$-substitution, well-typed in Γ. Let $\langle x\,;\,\gamma\,;\,t\rangle \in \sigma$, where $\exists x : S_1{\to}\cdots{\to}S_n{\to}S$ occurs in Γ and $\rho(S)$ is closed in Γ. Let τ be a solution for $Match(\sigma,\rho,d)_\Gamma$. Put $\Delta \equiv \tau(Deco(\sigma)_\Gamma(\Gamma_x), Cont(t\,;\, S_1{\to}\cdots{\to}S_n{\to}S)_{\Gamma_x})$.*
Then $\Delta \vdash_{\lambda\underline{\omega}} \tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}S)_\Gamma) : \tau(S_1{\to}\cdots{\to}S_n{\to}S)$.

Proof. We can write $\tau(Deco(\sigma)_\Gamma(\Gamma)) \equiv \Delta, \Delta'$, for some context Δ'. Using standardness of σ, it is easy to check that for all z in $dom(\Delta')$, z does not occur in $\tau(Deco(t\,;\, S_1{\to}\cdots{\to}S_n{\to}S)_\Gamma)$ or in $\tau(S_1{\to}\cdots{\to}S_n{\to}S)$. By our assumption on the form of contexts (see Lemma 2.6 (8)), we know that z does not occur in a declaration $u : U$ to the right of z in Δ'. So the result follows from Lemma 5.50 and Strengthening. ⊠

So $Match(\sigma, \rho, d)_\Gamma$ does what it should do. Our next task is to check that $Match(|\sigma|^\star, \rho, d)_\Gamma$ has a solution in case the initial problem $\langle \Gamma\,;\, A\,;\, B\rangle$ has solution σ. This is important for the completeness of the algorithm that we will use to decide whether $\langle \Gamma\,;\, A\,;\, B\rangle$ has a solution.

Lemma 5.52. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$, σ a solution for P. Let $\langle x\,;\, \gamma\,;\, t\rangle \in |\sigma|^\star$, where $\exists x : S_1{\to}\cdots{\to}S_n{\to}S$ occurs in Γ and x is of depth at most d in $\Phi(|P|, |\sigma|^\star)$. Suppose $\sigma_\Box(S)$ is closed in Γ. Then $\sigma_\Box$ is a solution for $Match(x, \sigma_\Box, d)_\Gamma$.*

Proof. First we write the available information in a convenient way. By Proposition 5.29, $\sigma^\star$ is well-typed and standard in Γ. Let $\langle x\,;\, \gamma'\,;\, t'\rangle \in \sigma^\star$. Write $S' \equiv S_1{\to}\cdots{\to}S_n{\to}S$. Write $t \equiv \lambda x_1{:}|S_1|\ldots\lambda x_n{:}|S_n|.yt_1\ldots t_m$ and $t' \equiv \lambda x_1{:}\sigma_\Box(S_1)\ldots\lambda x_n{:}\sigma_\Box(S_n).yt'_1\ldots t'_m$. Then by well-typedness of $\sigma^\star$ we have:

$$\sigma^\star(\Gamma) \vdash_{\lambda\underline{\omega}} t' : \sigma_\Box(S').$$

Write $\Gamma' \equiv \sigma^\star(\Gamma), \forall x_1 : \sigma_\Box(S_1), \ldots, \forall x_n : \sigma_\Box(S_n)$. Then:

$$\Gamma' \vdash_{\lambda\underline{\omega}} yt'_1\ldots t'_m : \sigma_\Box(S).$$

By Lemma 5.35 we can write the previous judgement as

$$\Gamma' \vdash_{\lambda\underline{\omega}} Deco(yt_1\ldots t_m\,;\, \sigma_\Box(S))_{\Gamma'} : \sigma_\Box(S).$$

Write $\Gamma'' \equiv Deco(|\sigma^\star|)_\Gamma(\Gamma), \forall x_1 : S_1, \ldots, \forall x_n : S_n$. Then by Lemma 5.38, we can write the previous judgement as:

$$\sigma_\Box(\Gamma'') \vdash_{\lambda\underline{\omega}} Deco(yt_1\ldots t_m\,;\, \sigma_\Box(S))_{\sigma_\Box(\Gamma'')} : \sigma_\Box(S).$$

By Lemma 5.36, this is the same as:

$$\sigma_\Box(\Gamma'') \vdash_{\lambda\underline{\omega}} \sigma_\Box(Deco(yt_1\ldots t_m\,;\, S)_{\Gamma''}) : \sigma_\Box(S) \qquad \text{(I)}.$$

That this lemma is applicable follows from Lemma 5.32 and Lemma 5.33.

Claim. Let Z be $\{x_1, \ldots, x_n\}$. Let W be $W_1{\to}\cdots{\to}W_r{\to}W'$ be of finite order in Γ and let Δ be a legal extension of $Deco(|\sigma|^\star)_\Gamma(\Gamma)$ such that $\Gamma =_\Box \Delta$ and s a subterm of t (not a domain) such that $|s|$ is in LNF and such that $Deco(s\,;\, W)_\Delta$ is defined and standard in Δ and Z. Suppose that W_1, …, W_r and $\sigma_\Box(W')$ are closed in Γ. Finally, suppose that $\sigma_\Box(\Delta) \vdash_{\lambda\underline{\omega}} \sigma_\Box(Deco(s\,;\, W)_\Delta) : \sigma_\Box(W)$. Then $\sigma_\Box$ is a solution for $SubMatch(Deco(s\,;\, W)_\Delta\,;\, \sigma_\Box(W))_{\Delta,Z}$ [1].

[1] It is not difficult to check that $SubMatch(Deco(s\,;\, W)_\Delta\,;\, \sigma_\Box(W))_{\Delta,Z}$ is defined.

By taking $r = 0$, $yt_1 \dots t_m$ for s, S for W' and $Deco(|\sigma|^\star)_\Gamma(\Gamma), \forall x_1 : S_1, \dots, \forall x_n : S_n$ for Δ and applying (I) we get: $\sigma_\Box$ is a solution for $SubMatch(Deco(yt_1 \dots t_m\,;\, S)_\Delta\,;\, \sigma_\Box(S))_{\Delta,Z}$. By Lemma 5.36, we get $\sigma_\Box$ is a solution for $SubMatch(Deco(yt_1 \dots t_m\,;\, \sigma_\Box(S))_\Delta\,;\, \sigma_\Box(S))_{\Delta,Z}$. It remains to show that $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, S\,;\, \sigma_\Box(S)\rangle$. This is trivial.

Proof of claim. By induction on the length of $Deco(s\,;\, W)_\Delta$. Write $Deco(s\,;\, W)_\Delta \equiv \lambda w_1{:}W_1 \dots \lambda w_r{:}W_r.uv_1 \dots v_p$. We distinguish two cases.

The first case is where $u \notin Z$. By standardness we know that for some R_1, ..., R_p, R, either $u \in dom(\Delta)$ and $\Delta \vdash_{\lambda\underline{\omega}} u : R_1 \to \dots \to R_p \to R$ or $u \equiv w_i$ (for some $1 \leq i \leq r$) and $W_i \equiv R_1 \to \dots \to R_p \to R$. That $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, W_i\,;\, \sigma_\Box(W_i)\rangle$ (for each $1 \leq i \leq r$) and for $\langle \Gamma\,;\, R\,;\, \sigma_\Box(W')\rangle$ is evident. Next, suppose $p > 0$. Put $\Delta' \equiv \Delta, \forall w_1 : W_1, \dots, \forall w_r : W_r$. Then we can write v_j ($1 \leq j \leq p$) as $Deco(v'_j\,;\, R_j)_{\Delta'}$, for some $\lambda\tau$ term v'_j. We have to show that $\sigma_\Box$ is a solution for $SubMatch(v_j\,;\, R'_j)_{\Delta',Z}$, where R'_j is either R_j (if $u \in dom(\Delta)$) or the twin of R_j in $\sigma_\Box(W_i)$ (if $u \equiv w_i$). Since W_i is closed in Γ, we see that also in the second case $R'_j \equiv R_j$. So in both cases $R'_j \equiv R_j \equiv \sigma_\Box(R_j)$. Thus it suffices to show hat $\sigma_\Box$ is a solution for $SubMatch(v_j\,;\, \sigma_\Box(R_j))_{\Delta',Z}$. By assumption: $\sigma_\Box(\Delta) \vdash_{\lambda\underline{\omega}} \lambda w_1{:}\sigma_\Box(W_1) \dots \lambda w_r{:}\sigma_\Box(W_r).u\sigma_\Box(v_1 \dots v_p) : \sigma_\Box(W)$. From this it follows that $\sigma_\Box(\Delta') \vdash_{\lambda\underline{\omega}} u\sigma_\Box(v_1) \dots \sigma_\Box(v_p) : \sigma_\Box(W')$. In particular: $\sigma_\Box(\Delta') \vdash_{\lambda\underline{\omega}} \sigma_\Box(v_j) : \sigma_\Box(R_j)$, for every $1 \leq j \leq p$. To be able to apply the induction hypothesis we have yet to verify that v_j is standard in Δ' and Z, that $|v_j|$ is in LNF and that R_j is of the form $R_j^1 \to \dots \to R_j^{p'} \to R'_j$, (for some $p' \geq 0$) where every R_j^l and $\sigma_\Box(R'_j)$ are closed in Γ. The first point follows from Lemma 5.27. As to the second point. If $u \in dom(\Delta)$ then by standardness, R_j is closed in Δ and hence in Γ. Next, suppose that $u \equiv x_i$. This case follows easily since R_j is a subterm of W_i and W_i is closed in Γ. So we can apply the induction hypothesis: $\sigma_\Box$ is a solution for $SubMatch(v_j\,;\, \sigma_\Box(R_j))_{\Delta',Z}$.

The second case is where $u \in Z$, say $u \equiv x_i$. Write $S_i \equiv R_1 \to \dots \to R_m \to R$ (note that $p = m$). The new thing is that for each $1 \leq j \leq m$ and each E in $\Phi_{x,i,j}$, we have to check that $\sigma_\Box$ is a solution for $add_{j,E}$. Write $E = \langle \Psi\,;\, xD_1 \dots D_n\,;\, B'\rangle$ and let $E' = \langle \Psi'\,;\, xD'_1 \dots D'_n\,;\, B''\rangle$ be the triple in $\Phi(P, \sigma^\star)$ that corresponds to E. Then $\Psi' \vdash_{\lambda\underline{\omega}} D'_i : \sigma_\Box(R_1) \to \dots \to \sigma_\Box(R_m) \to \sigma_\Box(R)$.

Again, we distinguish two cases. First suppose that $D_i \equiv \lambda y_1{:}O \dots \lambda y_m{:}O.y_j$. Since $|D'_i| \equiv D_i$, we have that $D'_i \equiv \lambda y_1{:}\sigma_\Box(R_1) \dots \lambda y_m{:}\sigma_\Box(R_m).y_j$. So obviously $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, R_j\,;\, \sigma_\Box(R)\rangle$ and since by assumption $\sigma_\Box(R) \equiv \sigma_\Box(W')$, $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, R_j\,;\, \sigma_\Box(W')\rangle$.

We have to show that $\sigma_\Box$ is a solution for $SubMatch(v_j\,;\, \sigma_\Box(W'))_{\Delta',Z}$. First observe that $\sigma_\Box(W') \equiv \sigma_\Box(R_j)$. So this case is similar to the one treated above. The verification of the fact that the induction hypothesis is applicable is actually easier than before because R_j is atomic.

Next suppose that $D_i \equiv \lambda y_1{:}O \dots \lambda y_m{:}O.es_1 \dots s_l$ ($e \not\equiv y_j$, for $1 \leq j \leq m$). We can identify some types. It must hold that $\Psi' \vdash_{\lambda\underline{\omega}} e : Q_1 \to \dots \to Q_l \to Q$, for some Q_1, ..., Q_l, Q such that $\sigma_\Box(R) \equiv Q$. Recall that by assumption $\sigma_\Box(W') \equiv \sigma_\Box(R)$. Let Ψ_1 be the $\lambda\underline{\omega}$-companion to Ψ and Ψ_2 be $\Psi_1, \forall y_1 : R_1, \dots, \forall y_m : R_m$. By Lemma 5.22, we have that $\sigma_\Box(\Psi_1)$ is an initial segment of Ψ'. We have to show that $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, R_j\,;\, U\rangle$ where $U \in Targets(y_j\,;\, Deco(es_1 \dots s_l\,;\, \sigma_\Box(W'))_{\Psi_2}\,;\, \sigma_\Box(W'))_{\Psi_2}$.

Since $\sigma_\Box(W') \equiv \sigma_\Box(R)$ it suffices to show that $\sigma_\Box$ is a solution for $\langle \Gamma\,;\, R_j\,;\, U\rangle$ where $U \in \mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, \sigma_\Box(R))_{\Psi_2}\,;\, \sigma_\Box(R))_{\Psi_2}$.

Let Ψ'' be $\Psi', \forall y_1 : \sigma_\Box(R_1), \ldots, \forall y_m : \sigma_\Box(R_m)$. Since $|D_i'| \equiv D_i$, we have that $D_i' \equiv \lambda y_1{:}\sigma_\Box(R_1) \ldots \lambda y_m{:}\sigma_\Box(R_m).g$ for some term g such that $|g| \equiv es_1 \ldots s_l$. So we have $\Psi'' \vdash_{\lambda\underline{\omega}} g : \sigma_\Box(R)$ and by Lemma 5.35, $\Psi'' \vdash_{\lambda\underline{\omega}} \mathit{Deco}(es_1 \ldots s_l\,;\, \sigma_\Box(R))_{\Psi''} : \sigma_\Box(R)$.
So by Lemma 5.40, $\mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, \sigma_\Box(R))_{\Psi''}\,;\, \sigma_\Box(R))_{\Psi''} = \{\Psi''(y_j)\} = \sigma_\Box(R_j)$.
Hence we are done if we can show that $\mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, \sigma_\Box(R))_{\Psi_2}\,;\, \sigma_\Box(R))_{\Psi_2} = \mathit{Targets}(y_j\,;\, \mathit{Deco}(es_1 \ldots s_l\,;\, \sigma_\Box(R))_{\Psi''}\,;\, \sigma_\Box(R))_{\Psi''}$.

By Lemma 5.41 (2), it suffices to show that for all z, z free in $es_1 \ldots s_l$ such that $\Psi_2(z)$ is not atomic we have: $\Psi_2(z) \equiv \Psi''(z)$ (it is easy to check that the other conditions mentioned in that lemma are satisfied). Consider such a z. Since its type is not atomic, z is not among $\{y_1, \ldots, y_m\}$. Thus, by a similar application of Lemma 2.11 as in the proof of Lemma 5.14, we have that z occurs in B', hence (by Lemma 5.24) is of closed type in Ψ_1. So $\Psi_2(z) \equiv \Psi_1(z) \equiv \sigma_\Box(\Psi_1(z)) \equiv \Psi'(z) \equiv \Psi''(z)$. That $\sigma_\Box$ is a solution for the remaining part of $\mathit{add}_{j,E}$ is verified as before. ⊠

⊠

The completeness of the algorithm presented in the proof of Theorem 5.54 is obtained by showing that if P in $\lambda\underline{\omega}$ has solution σ then $\sigma_\Box$ is a solution for *Match* for *all* heads x in $\Phi(|P|, |\sigma|^\star)$. The proof of this fact proceeds by induction on the depth of heads in $\Phi(|P|, |\sigma|^\star)$. The following lemma is needed for the induction step.

Lemma 5.53. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$ and σ a solution for P. Let x be a head in $\Phi(|P|, |\sigma|^\star)$ of depth n and of type $T_1 \to \cdots \to T_n \to T$ in Γ. Let y be a head below x of depth $n+1$ and of type $S_1 \to \cdots \to S_m \to S$ in Γ, where S is* not *closed in Γ. Suppose that θ is a $\Box$-substitution, well-typed in Γ, such that $\theta(T) \equiv \sigma_\Box(T)$ and $\theta(T)$ is closed in Γ. Then $\mathit{Match}(x, \theta, n)_\Gamma$ contains a matching problem $\langle \Gamma\,;\, S\,;\, \sigma_\Box(S)\rangle$.*

Proof. Let $E = \langle \Psi\,;\, xD_1 \ldots D_n\,;\, C\rangle \in \Phi(|P|, |\sigma|^\star)$. We can write Ψ as Ψ_1, Ψ_2, where Ψ_2 consists of dummy variables added in the construction of $\Phi(|P|, |\sigma|^\star)$. Let $xA_1 \ldots A_n$ be the witnessing term of E in $\Phi(|P|, |\sigma|^\star)$. By Lemma 5.22, $\Psi_1 \vdash_{\lambda\tau} xA_1 \ldots A_n : O$ and $|\sigma|^\star(xA_1 \ldots A_n) =_{\beta\eta} C$. Moreover each D_i $(1 \le i \le n)$ is either a dummy variable or $|\sigma|^\star(A_i)$. In the second case we know that z_i occurs in the normal form of $|\sigma|^\star(x)|\sigma|^\star(A_1) \ldots |\sigma|^\star(A_{i-1})z_i|\sigma|^\star(A_{i+1}) \ldots |\sigma|^\star(A_n)$ (z_i a fresh variable of the appropriate type). By assumption, we know that for some $1 \le i \le n$, the second case applies, $D_i \equiv |\sigma|^\star(A_i)$ and $y \in FV(A_i)$. So y is a head in $\Phi(\langle \Psi'\,;\, zt_1 \ldots t_p\,;\, D_i'\rangle, |\sigma|^\star)$, where we write $A_i \equiv \lambda u_1{:}U_1 \ldots \lambda u_l{:}U_l.zt_1 \ldots t_p$, $D_i \equiv \lambda u_1{:}U_1 \ldots \lambda u_l{:}U_l.D_i'$ and $\Psi' \equiv \Psi_1, \forall u_1 : U_1, \ldots, \forall u_l : U_l$. We distinguish four cases, three of which turn out to be impossible.

1. $z \equiv u_j$, for some $1 \le j \le l$. Because of typing reasons $p = 0$, contradicting the assumption that $y \in FV(A_i)$.

2. z is existential in Ψ, $z \not\equiv y$. Then y is not of depth $n+1$ and the result follows.

3. z is universal in Ψ. We show that y cannot be of depth $n+1$, contradicting the assumption. To E corresponds by Proposition 5.29, a triple $F = \langle \Psi'' \,;\, xD_1'' \dots D_n'' \,;\, C'' \rangle$ in $\Phi(P, \sigma^\star)$. Let $xA_1'' \dots A_n''$ be the witnessing term of F in $\Phi(P, \sigma^\star)$. By Lemma 5.23 we know that $|xA_1'' \dots A_n''| \equiv xA_1 \dots A_n$. By Lemma 5.22, we have that $\sigma^\star(xA_1'' \dots A_n'') =_{\beta\eta} C''$ and we have that z_i'' (z_i'' a fresh variable of the appropriate type) occurs in the normal form of $\sigma^\star(x)\sigma^\star(A_1'') \dots \sigma^\star(A_{i-1}'')z_i''\sigma^\star(A_{i+1}'') \dots \sigma^\star(A_n'')$.

   Write $A_i'' \equiv \lambda u_1{:}U_1'' \dots \lambda u_l{:}U_l''.zt_1'' \dots t_p''$. Let Ξ be the $\lambda\underline{\omega}$-companion to Ψ and let Ξ' be $\Xi, \forall u_1 : U_1'', \dots, \forall u_l : U_l''$. Using Lemma 5.22, we see that $\Xi' \vdash_{\lambda\underline{\omega}} zt_1'' \dots t_p'' : Z$ for some term Z. By Lemma 2.11, $\sigma^\star(zt_1'' \dots t_p'')$ is closed in $\sigma^\star(\Xi')$. (Apply this lemma with $\sigma^\star(\Xi)$ for Γ, $\langle\rangle$ for Δ, z_i'' for x, $\sigma^\star(x)\sigma^\star(A_1'') \dots \sigma^\star(A_{i-1}'')z_i''\sigma^\star(A_{i+1}'') \dots \sigma^\star(A_n'')$ for A, $\sigma^\star(A_i'')$ for B and the normal form of C'' for C). Moreover every variable, free in $\sigma^\star(zt_1'' \dots t_p'')$ and of non-atomic type in Ξ' (or in $\sigma^\star(\Xi')$), which is therefore not among $\{u_1, \dots, u_l\}$, occurs in C''. By Lemma 5.24, such a variable has closed type in Ξ and thus in Ξ'. By Lemma 3.11, taking $zt_1'' \dots t_p''$ for A, Ξ' for Γ and y for z, every occurrence of y must be in the scope of another existential variable bound by $\sigma^\star$ and hence by $|\sigma|^\star$. We may conclude that y cannot be of depth $n+1$.

4. $z \equiv y$. Let $|\sigma|^\star(x)$ be $\lambda v_1{:}V_1 \dots \lambda v_n{:}V_n.ws_1 \dots s_r$. By construction of $\Phi(P, |\sigma|^\star)$, $v_i \in FV(ws_1 \dots s_r)$ (otherwise y would not be a head in $\Phi(\langle \Psi' \,;\, zt_1 \dots t_p \,;\, D_i' \rangle)$). Let $Deco(|\sigma|^\star(x) \,;\, T_1 \to \dots \to T_n \to \theta(T))_\Gamma$ be $\lambda v_1{:}T_1 \dots \lambda v_n{:}T_n.ws_1' \dots s_r'$. Again we have $v_i \in FV(ws_1' \dots s_r')$. Write $T_i \equiv R_1 \to \dots \to R_l \to R$. One easily verifies that $S \equiv R$ (and hence $\sigma_\Box(R) \equiv \sigma_\Box(S)$). By Lemma 5.48 we have that $Match(x, \theta, n)_\Gamma$ contains a matching problem $\langle \Gamma \,;\, R \,;\, R' \rangle$ for some term R'. Now recall that $\theta(T) \equiv \sigma_\Box(T)$, closed in Γ. So $Match(x, \theta, n)_\Gamma = Match(x, \sigma_\Box, n)_\Gamma$. By Lemma 5.52, $\sigma_\Box$ is a solution for $Match(x, \sigma_\Box, n)_\Gamma$. So $R' \equiv \sigma_\Box(R)$. We are done.

⊠

At last we have reached the point where we can state and prove our main result.

Theorem 5.54. *It is decidable whether a third-order matching problem for objects in $\lambda\underline{\omega}$ has a solution or not.*

Proof. Let $P = \langle \Gamma \,;\, A \,;\, B \rangle$ be a third-order matching problem for objects in $\lambda\underline{\omega}$. We present an algorithm which, given P as input, returns a solution for P if P has a solution and *fail* otherwise. The algorithm is as follows.

- Translate P to $|P|$.
- Enumerate the solutions in $Sol(|P|)$ as $\{\sigma_i \mid 1 \leq i \leq n\}$. If $Sol(|P|)$ is empty, return *fail* and stop. Else do the following.
- Let α be a meta variable ranging over substitutions and *dummy* some dummy substitution. Put $i := 1$ and $\alpha := dummy$. While $i \leq n$ do

- Delete all triples $\langle x\,;\,\gamma\,;\,t\rangle$ in σ_i such that x is not a head in $\Phi(|P|,\sigma_i)$. (By Lemma 5.17, the resulting substitution, which we keep denoting by σ_i, is still a solution for $|P|$ and $\Phi(|P|,\sigma_i)$). If the result is the substitution $\emptyset$, put $i := n+1$ and $\alpha := \emptyset$. Else do the following.
- Define $appr(0) := \{\emptyset\}$; $appr(k+1) := \bigcup_{\theta\in appr(k)} Sol(Match(\sigma_i,\theta,k)_\Gamma)$
- Let d be the depth of $\Phi(|P|,\sigma_i)$. If there exist a θ_1 in $appr(d)$ such that for all x, head in $\Phi(|P|,\sigma_i)$ (with $\Gamma(x) \equiv S_1\to\cdots\to S_n\to S$), $\theta_1(S)$ is closed in Γ and if there exists a θ_2 in $Sol(Match(\sigma_i,\theta_1,d)_\Gamma)$, put $i := n+1$ and $\alpha := \theta_2\circ Deco(\sigma_i)_\Gamma$, else put $i := i+1$.

- If $\alpha \equiv dummy$, return *fail*, else return α. Stop.

We have to check that this algorithm is sound and complete and that it always terminates.

Soundness. Let σ be a standard solution for $|P|$ in $Sol(|P|)$ and d the depth of $\Phi(|P|,\sigma)$. If, after deletion of superfluous parts, $\sigma = \emptyset$, then we must have that $|A| \equiv |B|$, and $\emptyset$ is by Lemma 5.12 (2) indeed a solution for P. Next, suppose that $\sigma \neq \emptyset$. Then by Lemma 3.7, there must be an $x \in FV(A)$, bound by σ and of type $S_1\to\cdots\to S_n\to S$ in Γ, with S is closed in Γ. Let τ_1 be an element of $appr(d)$ such that $\tau_1(S)$ is closed in Γ for all x, head in $\Phi(|P|,\sigma)$ (with $\Gamma(x) \equiv S_1\to\cdots\to S_n\to S$). Let τ_2 be an element of $Sol(Match(\sigma,\tau_1,d)_\Gamma)$. We show that $\tau_2\circ Deco(\sigma)_\Gamma$ is a solution for P. By construction there exist substitutions ρ_1, ..., ρ_{d-1} such that ρ_1 is a standard solution for $Match(\sigma,\emptyset,0)_\Gamma$, each ρ_{i+1} is a standard solution for $Match(\sigma,\rho_i,i)_\Gamma$ and τ_1 is a standard solution for $Match(\sigma,\rho_{d-1},d-1)_\Gamma$. By Lemma 5.49 each ρ_i is well-typed in Γ. The same holds for τ_1 and τ_2. By Lemma 5.49 and Corollary 5.51, $\tau_2\circ Deco(\sigma)_\Gamma$ is well-typed in Γ. By Lemma 5.34, $|\tau_2\circ Deco(\sigma)_\Gamma| = \sigma$ and hence $|\tau_2\circ Deco(\sigma)_\Gamma|$ is a solution for $|P|$. By Lemma 5.13, $\tau_2\circ Deco(\sigma)_\Gamma$ is a solution for P.

Completeness. Suppose P has a solution σ. We have to prove that the algorithm returns a substitution. It is no restriction to assume that $dom(\sigma) \subseteq dom(\Gamma)$. By Proposition 5.11, $|\sigma|$ is a solution for $|P|$. So $|\sigma|^\star$ is also a solution for $|P|$ and occurs in the first enumeration given by the algorithm. The interesting case is where the deleting of superfluous parts from $|\sigma|^\star$ does not yield the empty substitution. Let $d > 0$ be the depth of $\Phi(|P|,|\sigma|^\star)$. Completeness follows from the next claim, which implies that $\alpha := \theta_d\circ Deco(|\sigma|^\star)_\Gamma$ is a solution for P.

Claim. For each k, $0 \le k \le d$, there exist a $\zeta_k \in appr(k)$ and a $\theta_k \in appr(k+1)$ such that $\theta_k \in Sol(Match(|\sigma|^\star,\zeta_k,k)_\Gamma)$ and for all heads x of depth $l \le k$ in $\Phi(|P|,|\sigma|^\star)$ and of type $S_1\to\cdots\to S_n\to S$ in Γ, $\zeta_k(S) \equiv \theta_k(S) \equiv \sigma_\Box(S)$ (and these types are closed in Γ).

Proof of claim. By induction on k.

- $k = 0$. Note that if x is a variable of depth 0 and of type $S_1\to\cdots\to S_n\to S$ in Γ then S is closed in Γ. Take $\zeta_0 = \emptyset$. By Lemma 5.52, $\sigma_\Box$ is a solution for $Match(x,\emptyset,0)_\Gamma$ and hence $(\sigma_\Box)^\star \in appr(1)$ is also a solution for $Match(x,\emptyset,0)_\Gamma$. Since S is closed in Γ, $\emptyset(S) \equiv (\sigma_\Box)^\star(S) \equiv \sigma_\Box(S) \equiv S$.

- $k = k'+1$. We put $\zeta_{k'+1} := \theta_{k'}$.

  First we show that $Match(|\sigma|^\star,\theta_{k'},k'+1)_\Gamma \subseteq Match(|\sigma|^\star,\sigma_\Box,k'+1)_\Gamma$. The interesting case is where there exists a head y of depth at most $k'+1$ in $\Phi(|P|,|\sigma|^\star)$ and of type

$T_1 \to \cdots \to T_m \to T$ in Γ, where T is not closed in Γ. There exists a head z above y in $\Phi(|P|, |\sigma|^\star)$ whose depth is k'. Let $Z_1 \to \cdots \to Z_l \to Z$ be the type of z in Γ. The induction hypothesis yields that $\zeta_{k'}(Z) \equiv \sigma_\Box(Z)$ is closed in Γ. Hence $\theta_{k'}$ is a solution for $Match(z, \zeta_{k'}, k')_\Gamma$. By Lemma 5.53 there exists a matching problem $\langle \Gamma\,;\, T\,;\, \sigma_\Box(T)\rangle \in Match(z, \zeta_{k'}, k')_\Gamma$. So $\theta_{k'}(T) \equiv \sigma_\Box(T)$, by construction closed in Γ.

Thus $Match(|\sigma|^\star, \theta_{k'}, k'+1)_\Gamma \subseteq Match(|\sigma|^\star, \sigma_\Box, k'+1)_\Gamma$. So by Lemma 5.52, $\sigma_\Box$ is a solution for $Match(|\sigma|^\star, \theta_{k'}, k'+1)_\Gamma$, hence there is a $\theta_{k'+1}$ in $Sol(Match(|\sigma|^\star, \theta_{k'}, k'+1)_\Gamma)$.

We still have to check that for all heads x of depth $l \leq k'+1$ in $\Phi(|P|, |\sigma|^\star)$ and of type $S_1 \to \cdots \to S_n \to S$ in Γ, $\theta_{k'}(S) \equiv \theta_{k'+1}(S) \equiv \sigma_\Box(S)$, closed in Γ. For all heads of depth $l \leq k'$ we know by induction hypothesis that $\theta_{k'}(S) \equiv \sigma_\Box(S)$, closed in Γ. Furthermore, we have already proved that if x is a head of depth $k'+1$, then $\theta_{k'}(S) \equiv \sigma_\Box(S)$, closed in Γ. So for all heads x of depth $l \leq k'+1$ we have $\theta_{k'}(S) \equiv \sigma_\Box(S)$, closed in Γ. Since $\theta_{k'+1} \in Sol(Match(|\sigma|^\star, \theta_{k'}, k'+1)_\Gamma)$, $\theta_{k'+1}$ is a solution for $\langle \Gamma\,;\, S\,;\, \theta_{k'}(S)\rangle$, hence $\theta_{k'+1}(S) \equiv \theta_{k'}(S)$, closed in Γ.

⊠

Termination. Evidently, $|P|$ can be computed in time linear in the length of P. By Theorem 5.28, $Sol(|P|)$ can be enumerated in time bounded by some function depending only on $|P|$. The decoration procedure for each solution σ_i can be performed in time linear in the number of symbols in P and σ_i. For all standard solutions σ_i for $|P|$ and all substitutions ρ, well-typed in Γ, the number of matching problems and the length of each matching problem in $Match(\sigma_i, \rho, d)_\Gamma$ is linear in the number of symbols in σ_i. Since moreover $Match(\sigma_i, \rho, d)_\Gamma$ is a collection of third-order matching problem for types in $\lambda\underline{\omega}$, $Sol(Match(\sigma_i, \rho, d)_\Gamma)$ can again be enumerated in time bounded by some function depending only on $Match(\sigma_i, d, \rho)_\Gamma$. So at each stage there are finitely many new matching problems. Because the depth of $\Phi(P, \sigma)$ is finite there are finitely many such stages. ⊠

6 Third-order closed matching is undecidable in $\lambda\underline{\omega}$

In this section we show that it is undecidable whether a third-order matching problem in $\lambda\underline{\omega}$ has a closed solution or not. The proof is a slightly more complicated variant of the proofs in [5]. It is noteworthy that the proof does not depend on the undecidability of the problem whether there is a term of a given type (in a given context). Indeed, this problem is decidable (a proof of this fact will appear in the author's forthcoming thesis).

Definition 6.1. A unification problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\underline{\omega}$ is called *elementary at the level of types* (or *elementary*, for short) if:

- $\Gamma \equiv \langle \forall Z : *\rangle, \Gamma'$, for some variable Z.
- for all declarations $Qx : T$ in Γ it holds that $T \in \{*, * \to *, * \to * \to *, * \to * \to * \to *\}$. (Note that the order of these kinds is at most three.)

- $\Gamma \vdash_{\lambda\underline{\omega}} A : *$ and $\Gamma \vdash_{\lambda\underline{\omega}} B : *$.

Theorem 6.2 *(Goldfarb, Dowek). There exists no algorithm that decides whether a elementary unification problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\underline{\omega}$ has a (closed) solution or not.*

Proof. See [4], [14]. The assumption that Γ starts with the declaration $\forall Z : *$ is justified by the fact that Goldfarb assumes that the language for which the unification problem is stated contains two individual constants (although he does not need this assumption). ⊠

Remark 6.3. It is no restriction to assume that if an elementary unification problem $\langle \Gamma\,;\, A\,;\, B\rangle$ in $\lambda\underline{\omega}$ has a solution then $\sigma(A)$ (and hence $\sigma(B)$) is closed in $\sigma(\Gamma)$. For, let σ be a solution and let e be an existential variable occurring in $\sigma(A)$. In case e has type $*$, we can extend σ with $\{\langle e\,;\, \langle\rangle\,;\, Z\rangle\}$; in case e has type $*\to*$, we can extend σ with $\{\langle e\,;\, \langle\rangle\,;\, \lambda x{:}*.Z\rangle\}$, etc.

Theorem 6.4. *There exists no algorithm that decides whether a third-order matching problem in $\lambda\underline{\omega}$ has a closed solution or not.*

Proof. For every elementary unification problem $\langle \Gamma_1\,;\, A_1\,;\, B_1\rangle$ in $\lambda\underline{\omega}$ we construct a matching problem $\langle \Gamma_2\,;\, A_2\,;\, B_2\rangle$ for objects in $\lambda\underline{\omega}$ such that the order of the existential variables in Γ_2 is at most 3 and such that $\langle \Gamma_1\,;\, A_1\,;\, B_1\rangle$ has a closed solution iff $\langle \Gamma_2\,;\, A_2\,;\, B_2\rangle$ has a closed solution. The result then follows from Theorem 6.2.

Let $\langle \Gamma_1\,;\, A_1\,;\, B_1\rangle$ be given. We define:

- $\Gamma_2 \equiv \Gamma_1, \forall X : *\to*, \forall c : Z, \forall d : Z, \forall g : Z\to Z\to Z, \forall b : XB_1,$
  $\exists f : ((XA_1)\to Z)\to(XB_1)\to Z.$
  (Here the new variables are fresh.)
- $A_2 \equiv g(f(\lambda z{:}XA_1.c)b)(f(\lambda z{:}XA_1.d)b).$
- $B_2 \equiv gcd.$

This defines a third-order matching problem for objects in $\lambda\underline{\omega}$. We show that closed solutions for $\langle \Gamma_1\,;\, A_1\,;\, B_1\rangle$ are in bijective correspondence with closed solutions for $\langle \Gamma_2\,;\, A_2\,;\, B_2\rangle$.

Suppose σ is a solution for $\langle \Gamma_1\,;\, A_1\,;\, B_1\rangle$, i.e. $\sigma(A_1) =_{\beta\eta} \sigma(B_1)$. From σ we construct a solution τ for $\langle \Gamma_2\,;\, A_2\,;\, B_2\rangle$:

$$\tau = \sigma \cup \{\langle f\,;\, \langle\rangle\,;\, \lambda x{:}(X\sigma(A_1))\to Z.\lambda y{:}X\sigma(B_1).xy\rangle\}.$$

We have to verify three things.

1. $\tau(A_2) =_{\beta\eta} B_2$. This is easy.
2. τ is closed. We know that σ is closed. By Remark 6.3, we may assume that $\sigma(A_1)$ and $\sigma(B_1)$ are closed in $\sigma(\Gamma_1)$ and hence in $\tau(\Gamma')$. So $\lambda x{:}(X\sigma(A_1))\to Z.\lambda y{:}X\sigma(B_1).xy$ is closed in $\tau(\Gamma')$ from which we may conclude that τ is closed.
3. τ is well-typed in Γ_2. This consists of three things:

(a) We must show that τ binds no variable that are universal in Γ_2. This is easy, since σ has this property w.r.t. Γ_1.

(b) We must show that $\tau(\Gamma_2)$ is legal. This is proved by induction on the length of Γ_2, using that $\sigma(\Gamma_1)$ is legal and that $\sigma(\Gamma_1) \vdash_{\lambda\underline{\omega}} \sigma(B_1) : *$.

(c) For all existential variables x in Γ_2 bound by τ (say $\langle x\,;\,\gamma\,;\,M\rangle \in \tau$), we must show that $\tau((\Gamma_2)_x), \gamma \vdash_{\lambda\circ} M : \tau((\Gamma_2)_x)$. We treat the only new case, the existential variable f. We have to show that
$\tau((\Gamma_2)_f) \vdash_{\lambda\underline{\omega}} \lambda x{:}(X\sigma(A_1))\to Z.\lambda y{:}X\sigma(B_1).xy : ((X\tau(A_1))\to Z)\to(X\tau(B_1))\to Z$.
But this is trivial, given the fact that $\tau(A_1) =_{\beta\eta} \sigma(A_1) =_{\beta\eta} \sigma(B_1) =_{\beta\eta} \tau(B_1)$.

Conversely, suppose that $\langle \Gamma_2\,;\,A_2\,;\,B_2\rangle$ has a closed solution τ. Then τ is well-typed in Γ_2 and (since Γ_1 is an initial segment of Γ_2) τ is also well-typed in Γ_1. Next we show (using $\tau(A_2) =_{\beta\eta} B_2$) that $\tau(A_1) =_{\beta\eta} \tau(B_1)$.

Put $\Delta \equiv (\tau(\Gamma_2)), \forall y_1 : (X\tau(A_1))\to Z, \forall y_2 : X\tau(B_1)$ and define v as the $\beta\eta$-normal form of $(\tau(f))y_1y_2$. We have that

$$\Delta \vdash_{\lambda\underline{\omega}} (\tau(f))y_1y_2 : Z \tag{1}$$

and so by Subject Reduction:

$$\Delta \vdash_{\lambda\underline{\omega}} v : Z. \tag{2}$$

Below we show that $v \equiv y_1y_2$ or $v \equiv y_1b$. For typing reasons this is only possible when $X\tau(A_1) =_{\beta\eta} X\tau(B_1)$, which is equivalent to $\tau(A_1) =_{\beta\eta} \tau(B_1)$.

The equation $\tau(A_2) =_{\beta\eta} B_2$ is by Substitutivity equivalent to the system consisting of the following two equations:

$$v[y_1 := \lambda z{:}X\tau(A_1).c, y_2 := b] =_{\beta\eta} c. \tag{3}$$

$$v[y_1 := \lambda z{:}X\tau(A_1).d, y_2 := b] =_{\beta\eta} d. \tag{4}$$

Since Z, the type of v, is atomic, v is not an abstraction term. Since Z is not ($\beta\eta$-convertible to) a sort, v is not a product term. So v is an atomic term, say $v \equiv \alpha M_1 \ldots M_n$, for some $n \geq 0$. Here α is a sort or variable. Given that $v[y_1 := \lambda z{:}X\tau(A_1).c, y_2 := b] =_{\beta\eta} c$, we have that $\alpha \in \{y_1, y_2, c\}$. Given that $v[y_1 := \lambda z{:}X\tau(A_1).d, y_2 := b] =_{\beta\eta} d$, we have that $\alpha \in \{y_1, y_2, d\}$. So $\alpha \in \{y_1, y_2\}$. Towards a contradiction, suppose that $\alpha \equiv y_2$. Then $n = 0$ and $v[y_1 := \lambda z{:}X\tau(A_1).c, y_2 := b] =_{\beta\eta} b \neq_{\beta\eta} c$. Contradiction. So $\alpha \equiv y_1$. Recall that $\Delta \vdash_{\lambda\underline{\omega}} y_1 : (X\tau(A_1))\to Z$. Hence $n = 1$ and we have that

$$\Delta \vdash_{\lambda\underline{\omega}} \alpha : (X\tau(A_1))\to Z. \tag{5}$$

Let us now take a look at M_1. Since τ is well-typed, the type of M_1 is $\beta\eta$-convertible to $X\tau(A_1)$. So the type of M_1 is atomic and as before we show that M_1 is an atomic term, say $M_1 \equiv \beta N_1 \ldots N_k$, for some $k \geq 0$. Here β is a sort or a variable. For a contradiction, suppose that β is a sort. Then $k = 0$ and the type of M_1 would have to be $\beta\eta$-convertible to a sort, which is not the case. Contradiction. So β is a variable. Since τ is closed, β has to appear in a declaration $\beta : T$ in

$\tau(\Gamma_1), \forall X : *\to *, \forall c : Z, \forall d : Z, \forall g : Z\to Z\to Z, \forall b : X\tau(B_1),$
$\exists f : ((X\tau(A_1))\to Z)\to(X\tau(B_1))\to Z, \forall y_1 : (X\tau(A_1))\to Z, \forall y_2 : X\tau(B_1).$

The declaration $\beta : T$ can't appear in $\tau(\Gamma_1)$, because then the type of M_1 would be $*$, which is not $\beta\eta$-convertible to $X\tau(A_1)$. For the same reason, β can't be X or Z. Suppose $\beta \in \{c, d, g, f, y_1\}$. The types of these variables can all be written as $C_1\to\cdots\to C_m\to Z$, for some m depending on the variable. So the type of M_1 would be Z, which is not $\beta\eta$-convertible to $X\tau(A_1)$. So $\beta \notin \{c, d, g, f, y_1\}$. Hence $\beta \equiv b$ or $\beta \equiv y_2$. Both b and y_2 have type $X\tau(B_1)$. So we have $k = 0$ and $M_1 \equiv \beta$. Recall that the type of M_1 is $\beta\eta$-convertible to $X\tau(A_1)$. So we have that $X\tau(B_1) =_{\beta\eta} X\tau(A_1)$, which implies $\tau(A_1) =_{\beta\eta} \tau(B_1)$, as required. ⊠

Remark 6.5. Recall (see Definition 3.4) that we spoke about an alternative definition of closed solution, in which one only demands that substitution contexts are closed, whereas substitution terms need not be closed. The proof given above can be adapted to this definition. The step from $\langle\Gamma_1\,;\,A_1\,;\,B_1\rangle$ to $\langle\Gamma_2\,;\,A_2\,;\,B_2\rangle$ is in fact easier: one does not have to show that $\lambda x{:}(X\sigma(A_1))\to Z.\lambda y{:}X\sigma(B_1).xy$ is closed in $\tau(\Gamma_2)$. The step from $\langle\Gamma_2\,;\,A_2\,;\,B_2\rangle$ to $\langle\Gamma_1\,;\,A_1\,;\,B_1\rangle$ need not be changed.

A Appendix

In section 5, we have proved the decidability of third-order matching under a certain assumption, namely that universal variables of order ∞ do not occur. In this section we show that this assumption imposes no restriction. We present a translation that maps a matching problem P of finite order involving universal variables of order ∞ to a matching problem P' without such variables. This translation has the property that P has a solution iff P' has a solution. The crucial point here is that (writing $P = \langle\Gamma\,;\,A\,;\,B\rangle$), because B is closed in Γ, variables of order ∞ do not occur in B. Also note that if y is of order ∞ in Γ, then $\Gamma(y) \notin \mathcal{K}_\Box$. Hence *types* do not contain variables of order ∞.

We introduce some terminology. Let Γ be legal in $\lambda\underline{\omega}$, y a variable of order ∞ in Γ. Write $\Gamma(y) \equiv Y_1\to\cdots\to Y_k\to Y$. If Y is of finite order in Γ, then we call y *safe w.r.t.* Γ. When Γ is clear from the context, we simply say that y is safe. Suppose $\Gamma \vdash_{\lambda\underline{\omega}} t : T$ and $\mathcal{L} \subseteq FV(t)$. Let $x \in \mathcal{L}$. If, for some occurrence of x in t, there is no occurrence of a subterm of the form $yt_1 \ldots t_m$ such that $y \in \mathcal{L}$ and this occurrence of x is an occurrence in t_i, for some $1 \le i \le m$, then we call this occurrence of x *outermost w.r.t.* $\mathcal{L}$ *(in* t*).*

Lemma A.1. *Suppose* $\Gamma \vdash_{\lambda\underline{\omega}} t : T$*, where* T *is of finite order w.r.t.* Γ*. Write* $\mathcal{L} = \{y \in FV(t) \mid$ *order of* y *in* Γ *is* $\infty\}$*. Let* $y \in \mathcal{L}$*. If* y *is not safe, then for every occurrence of* y *there is an occurrence of a subterm in* t *of the form* $zs_1 \ldots s_l$ *such that* z *is safe, this occurrence of* z *is outermost w.r.t.* $\mathcal{L}$ *and this occurrence of* y *is an occurrence in* s_i*, for some* $1 \le i \le l$*.*

Proof. By induction on the LNF-structure of t. Write $t \equiv \lambda x_1{:}T_1 \ldots \lambda x_n{:}T_n.xt_1 \ldots t_m$. Let y be given. If $x \equiv y$, then y is safe. Otherwise $y \in FV(t_1) \cup \cdots \cup FV(t_m)$ (remember that $y \notin \mathcal{K}_\Box$) and there are two possible cases. If x is itself of order ∞, then we can take x for z. If not, then x is either an x_j $(1 \le j \le n)$ or a free variable of finite order. Say that

the type of x is $S_1 \to \cdots \to S_m \to T'$. In both cases each S_i $(1 \leq i \leq n)$ is of finite order in Γ. We also know that $\Gamma, \forall x_1 : T_1, \ldots, \forall x_n : T_n \vdash_{\lambda\underline{\omega}} t_i : S_i$, for each $1 \leq i \leq n$. Write $J = \{1 \leq j \leq n \mid y \in FV(t_j)\}$. For all $j \in J$, we have by induction hypothesis: for every occurrence of y in t_j there is an occurrence of a subterm in t_j of the form $z s_1 \ldots s_l$ such that z is safe w.r.t. $\Gamma, \forall x_1 : T_1, \ldots, \forall x_n : T_n$ and this occurrence of z is outermost w.r.t. $\mathcal{L} \cup \{u \in \{x_1, \ldots, x_n\} \mid \text{u of order } \infty \text{ in } \Gamma, \forall x_1 : T_1, \ldots, \forall x_n : T_n\}$ in t_j and this occurrence of y is an occurrence in s_k, for some $1 \leq k \leq l$. Now note that none of the variables in $\{x_1, \ldots, x_n\}$ are of order ∞. So z is not in this set and hence is safe w.r.t. Γ; the occurrence of z is easily seen to be outermost w.r.t. $\mathcal{L}$ in t. This finishes the proof. $\boxtimes$

Definition A.2. Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a matching problem of finite order n in $\lambda\underline{\omega}$. Suppose that $\Gamma \vdash_{\lambda\underline{\omega}} A : C$. We define $P' = \langle \Gamma'\,;\, A'\,;\, B\rangle$ as follows. Write $\mathcal{L} = \{y \in FV(A) \mid \text{order of } y \text{ in } \Gamma \text{ is } \infty\}$ and $\mathcal{L}' = \{y \in \mathcal{L} \mid y \text{ safe in } \Gamma\}$. Note that very variable in $\mathcal{L}$ is universal in Γ. Let Γ_1 be the result of inserting immediately before each declaration $\forall y : Y_1 \to \cdots \to Y_n \to Y$ in Γ with $y \in \mathcal{L}'$ a declaration $\forall y' : Y$, where y' is fresh. For each $y \in \mathcal{L}'$ of type $Y_1 \to \cdots \to Y_n \to Y$, write $c_y \equiv \lambda z_1{:}Y_1 \ldots \lambda z_1{:}Y_n.y'$, where y' is as above and $y' \not\equiv z_i$, for all $1 \leq i \leq n$. Let A_1 be the result of replacing in A each y in $\mathcal{L}'$ by c_y. By the Substitution Lemma and Weakening, $\Gamma_1 \vdash_{\lambda\underline{\omega}} A_1 : C$. So $\mathrm{lnf}_{\Gamma_1}(A_1)$ exists; we write $A' \equiv \mathrm{lnf}_{\Gamma_1}(A_1)$. Using Lemma A.1, we know that no variable in $\mathcal{L}$ occurs in A' (this lemma is applicable because C is closed in Γ_1, hence of finite order in Γ_1). Let Γ' be the result of removing all variables of order ∞ from Γ_1. Then by Thinning we have $\Gamma' \vdash_{\lambda\underline{\omega}} A' : C$ and $\Gamma' \vdash_{\lambda\underline{\omega}} B : C$ (recall that B and C do not contain variables of order ∞). So $\langle \Gamma'\,;\, A'\,;\, B\rangle$ is a matching problem of order n and by construction Γ' and A' do not contain variables of order ∞.

Lemma A.3. *Let $P = \langle \Gamma\,;\, A\,;\, B\rangle$ be a matching problem of finite order in $\lambda\underline{\omega}$. Let $P' = \langle \Gamma'\,;\, A'\,;\, B\rangle$ be as in Definition A.2. Then P has a solution $\Leftrightarrow$ P' has a solution.*

Proof. Let $\mathcal{L}$ and $\mathcal{L}' = \{y_1, \ldots, y_l\}$ and c_y (for every $y \in \mathcal{L}'$) be as in Definition A.2. Let B' be the normal form of B. Recall that no variable of order ∞ or a variable added in the construction of Γ' occurs in B or B'.

"$\Rightarrow$" Suppose P has solution σ. For each $y \in \mathcal{L}'$ of type $Y_1 \to \cdots \to Y_n \to Y$, write $c'_y \equiv \sigma(c_y) \equiv \sigma_\Box(c_y) \equiv \lambda z_1{:}\sigma(Y_1) \ldots \lambda z_1{:}\sigma(Y_n).y'$. Write $\sigma_\Box = \{\langle z_j\,;\, \gamma_j\,;\, Z_j\rangle \mid 1 \leq j \leq m_1\}$ and $\sigma_* = \{\langle x_j\,;\, \delta_j\,;\, S_j\rangle \mid 1 \leq j \leq m_2\}$. For each $1 \leq j \leq m_2$, let S'_j be the result of replacing each $y \in FV(S_j) \cap \mathcal{L}'$ by c'_y and let S''_j be $\mathrm{lnf}_{\sigma(\Gamma_1)}(S'_j)$. (One can check that this is well-defined.) Put $\sigma' = \sigma_\Box \cup \{\langle x_j\,;\, \delta_j\,;\, S''_j\rangle \mid 1 \leq j \leq m_2\}$. We show that σ' is a solution for P'. Using Lemma A.1, σ' is easily seen to be well-typed in Γ'. Below we calculate that $A[\vec{y} := \vec{c_y}][\vec{z} := \vec{Z}][\vec{x} := \vec{S''}] =_{\beta\eta} B'$. Since $A[\vec{y} := \vec{c_y}] =_{\beta\eta} A'$, we have that $A'[\vec{z} := \vec{Z}][\vec{x} := \vec{S''}] =_{\beta\eta} B'$. We may conclude that σ' is a solution for P'. As to the calculation. We know $A[\vec{z} := \vec{Z}][\vec{x} := \vec{S}] \twoheadrightarrow_{\beta\eta} B'$. Now

$A[\vec{y} := \vec{c_y}][\vec{z} := \vec{Z}][\vec{x} := \vec{S''}] \equiv$
$A[\vec{z} := \vec{Z}][\vec{y} := \vec{c'_y}][\vec{x} := \vec{S''}] =_{\beta\eta}$
$A[\vec{z} := \vec{Z}][\vec{y} := \vec{c'_y}][\vec{x} := \vec{S'}] \equiv$
$A[\vec{z} := \vec{Z}][\vec{x} := \vec{S}][\vec{y} := \vec{c'_y}] \twoheadrightarrow_{\beta\eta}$
$B'[\vec{y} := \vec{c'_y}] \equiv B'$.

We have to verify the validity of the steps in the calculation. Note that all terms involved are typable in $\sigma(\Gamma_1)$. As to the first step. Observe that $\{z_1,\ldots,z_{m_1}\}\cap\{y_1,\ldots,y_l\}=\emptyset$, because the variables in the leftmost set are existentially quantified in Γ_1, whereas the variables in the rightmost set (i.e. $\mathcal{L}'$) are universally quantified in Γ_1. Moreover the variables in $\mathcal{L}'$ are of order ∞, hence do not occur in $\sigma_\Box$. The second step is easy. As to the third step. Again we have that $\{x_1,\ldots,x_{m_2}\}\cap\{y_1,\ldots,y_l\}=\emptyset$. Moreover no variable in $\{x_1,\ldots,x_{m_2}\}$ occurs in a term c'_y. The fourth step follows from Substitutivity and the fact that $A[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}]\twoheadrightarrow_{\beta\eta} B'$.

"$\Leftarrow$" Suppose P' has solution τ. By Proposition 5.29, we may assume that the substitution terms in τ do not contain any of the new variables y' or any variable of order ∞. In particular, τ is well-typed in Γ. For each $y\in\mathcal{L}'$ of type $Y_1\to\cdots\to Y_n\to Y$, write $c''_y\equiv\tau(c_y)\equiv\lambda z_1{:}\tau(Y_1)\ldots\lambda z_1{:}\tau(Y_n).y'$. Let A'' be the normal form of $\tau(A)$ (A'' exists because $\Gamma\vdash_{\lambda\underline{\omega}} A:C$ and τ is well-typed in Γ). We prove that $A''\equiv B'$. First, we claim that A'' does not contain variables from the set $\mathcal{L}$. Towards a contradiction, assume that A'' does contain such variables. Then it contains an outermost occurrence of such a variable u. There are two possible cases: the first case is where u is safe in Γ, the second case is where u is not safe in Γ. Let A''' be the result of replacing in A'' each y in $\mathcal{L}'$ by c''_y (this term is well-typed in $\tau(\Gamma_1)$). In the first case, the normal form of A''' contains an occurrence of u'. In the second case the normal form of A''' contains an occurrence of u. But the normal form of A''' is B'. This can be seen as follows. Write $\tau_\Box=\{\langle z_j\,;\,\gamma_j\,;\,Z_j\rangle\mid 1\le j\le m_1\}$ and $\tau_*=\{\langle x_j\,;\,\delta_j\,;\,S_j\rangle\mid 1\le j\le m_2\}$. We know $A[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}]\twoheadrightarrow_{\beta\eta} A''$. So $A[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}][\vec{y}:=\vec{c''_y}]\twoheadrightarrow_{\beta\eta} A''[\vec{y}:=\vec{c''_y}]\equiv A'''$. Now

$A[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}][\vec{y}:=\vec{c''_y}]\equiv$ (no $y\in\mathcal{L}'$ free in $\vec{S}$)

$A[\vec{z}:=\vec{Z}][\vec{y}:=\vec{c''_y}][\vec{x}:=\vec{S}]\equiv$

$A[\vec{y}:=\vec{c_y}][\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}]=_{\beta\eta}$

$A'[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}]$.

Recall that B' is the normal form of $A'[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}]$. Thus we have that B' is also the normal form of $A[\vec{z}:=\vec{Z}][\vec{x}:=\vec{S}][\vec{y}:=\vec{c''_y}]$ and hence of A'''. And B' does not contain u or u'. Contradiction. We conclude that A'' does not contain variables from the set $\mathcal{L}$. So $A''[\vec{y}:=\vec{c''_y}]\equiv A''$. From this we quickly obtain that $A''\equiv B'$, because B' is the normal form of $A''[\vec{y}:=\vec{c''_y}]\equiv A''$ and A'' is itself normal. ⊠

Remark A.4. With a slightly more complicated proof, also removing variables of order ∞ from substitution terms, one obtains the following strengthening. Let P and P' be as in Lemma A.3. Then P has a solution $\Leftrightarrow$ P' has a solution not containing variables of order ∞.

References

[1] H.P. Barendregt. *The Lambda Calculus. Its Syntax and Semantics.* North-Holland, Amsterdam, second, revised edition, 1984.

[2] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Oxford Science Publications, 1992.

[3] T. Coquand and G. Huet. The calculus of constructions. *Information and Control*, 76:95–120, 1988.

[4] G. Dowek. *Démonstration Automatique dans le Calcul des Constructions.* PhD thesis, l'université Paris VII, 1991.

[5] G. Dowek. L'indécidabilité du filtrage du troisième ordre dans les calculs avec types dépendants ou constructeurs de types. *Compte Rendu à l'Académie des Sciences*, 312, Série I:951–956, 1991.

[6] G. Dowek. A second-order pattern matching algorithm for the cube of typed λ-calculi. In *Mathematical Foundations of Computer Science 91*, volume 520 of *Lecture Notes in Computer Science*, pages 151–160. Springer-Verlag, 1991.

[7] G. Dowek. A second-order pattern matching algorithm for the cube of typed λ-calculi. Technical report, INRIA – Rocquencourt, 1991.

[8] G. Dowek. Third order matching is decidable. In *Proceedings* 7^{th} *Annual Symposium on Logic in Computer Science,* Santa Cruz, California, 1992.

[9] G. Dowek. A complete proof synthesis method for the cube of type systems. To be published, 1993.

[10] G. Dowek. The undecidability of pattern matching in calculi where primitive recursive functions are representable. *Theoretical Computer Science*, 107:349–356, 1993.

[11] G. Dowek, A. Felty, H. Herbelin, G. Huet, C. Murthy, C. Parent, C. Paulin-Mohring, and B. Werner. The Coq proof assistant user's guide. Version 5.8. Technical report, INRIA – Rocquencourt, May 1993.

[12] J.H. Geuvers. The Church-Rosser property for $\beta\eta$-reduction in typed λ-calculi. In *Proceedings* 7^{th} *Annual Symposium on Logic in Computer Science,* Santa Cruz, California, 1992.

[13] J.H. Geuvers and M.-J. Nederhof. Modular proof of strong normalisation for the calculus of constructions. *Journal of Functional Programming*, 1:155–189, 1989.

[14] W.D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[15] Z. Luo. ECC, the Extended Calculus of Constructions. In *Proceedings* 4^{th} *Annual Symposium on Logic in Computer Science,* Asilomar, California, pages 386–395. IEEE, 1989.

[16] Z. Luo and R. Pollack. LEGO proof development system: User's manual. Technical report, Department of Computer Science, University of Edinburgh, May 1992.

[17] D. Miller. Unification under a mixed prefix. *Journal of Symbolic Computation*, 14(4):321–359, October 1992.

[18] V. Padovani. Fourth order dual interpolation is decidable. Manuscript, Université Paris VII - C.N.R.S, 1994.

[19] V. Padovani. On equivalence classes of interpolation equations. Manuscript, Université Paris VII - C.N.R.S, 1994.