LOGIC ALGEBRA, AND COMPUTER SCIENCE
BANACH CENTER PUBLICATIONS, VOLUME 46
INSTITUTE OF MATHEMATICS
POLISH ACADEMY OF SCIENCES
WARSZAWA 1999

BETA-REDUCTION AS UNIFICATION

A. J. KFOURY
Department of Computer Science, Boston University
Boston, MA 02215, U.S.A.
E-mail: kfoury@fiddle.bu.edu

Dedicated to the memory of Professor Helena Rasiowa

Abstract. We define a new unification problem, which we call *β-unification* and which can be used to characterize the β-strong normalization of terms in the λ-calculus. We prove the undecidability of β-unification, its connection with the system of intersection types, and several of its basic properties.

1. Introduction. In the λ-calculus, β-reduction is the basic mechanism for evaluating terms. As a rewrite rule, which we identify as (β), it has a very simple definition:

$$(\beta) \qquad (\lambda x\ M)N \longrightarrow_\beta M[x := N]$$

In words, the term N is substituted for every free occurrence of the variable x in the term M.[1] In sharp contrast to the simplicity of its definition, (β) gives rise to a very rich theory that has been extensively researched over more than half a century. In this paper, we wish to examine one particular aspect of (β) which has not been considered in any depth so far. We start with a little background.

Let $\mathcal{U}$ be a unification problem. The reader may wish to specify $\mathcal{U}$ further, by taking it to be any of the familiar unification problems in the literature, such as first-order, or second-order, or semi-unification, etc., with or without restrictions imposed on it. One is usually interested in such questions as: Is $\mathcal{U}$ decidable? If it is, what are efficient algorithms to solve $\mathcal{U}$? If it is not, what are restrictions under which $\mathcal{U}$ becomes decidable? In either case, does $\mathcal{U}$ satisfy a "principality property" (the existence of appropriately defined "most general solutions")? In the investigation of $\mathcal{U}$, a standard approach is to devise a finite set R of rewrite rules with the property that, given an arbitrary instance Δ

1991 *Mathematics Subject Classification*: Primary 46C20; Secondary 32G81.
Partly supported by NSF grant CCR–9417382.
The paper is in final form and no version of it will be published elsewhere.

[1] Here and throughout, we follow the notation and conventions of Barendregt's book [2].

of $\mathcal{U}$, the process of applying the rules of R to Δ (or to some convenient representation of Δ) will terminate iff Δ has a solution. Sometimes an order is prescribed in which the rules of R are applied, but not always. Let us call such a set R a *rewrite characterization* of $\mathcal{U}$. In general, R is not unique and there are several convenient rewrite characterizations $R', R'', \ldots$ for the same $\mathcal{U}$.

Returning to β-reduction, we consider the set $R = \{(\beta)\}$ with just one rewrite rule, and ask:

(A) Is there a way of taking $R = \{(\beta)\}$ as the rewrite characterization of an appropriately defined unification problem $\mathcal{U}$?

At first blush, the answer to Question (A) is "no". While (β) can be applied to any term M of the λ-calculus, it is not clear how M is itself a unification instance or how M can even represent one (which is typically a finite set of constraints, each between two formal expressions). We thus modify Question (A), and ask instead:

(B) What is a unification problem $\mathcal{U}$ such that, given a term M, we can systematically transform M into an instance Δ of $\mathcal{U}$, in the form of a finite set of constraints, which has a solution iff M is β-strongly normalizable?

The "only-if" part (the left-to-right direction) of Question (B) is relatively easy. In fact, it was considered repeatedly in past investigations and given a satisfactory answer. For example, we can take $\mathcal{U}$ to be standard first-order unification, as there are well-known techniques to transform a term M of the λ-calculus into an instance Δ of first-order unification such that, if Δ has a solution, then M is β-strongly normalizable.[2] In this case, $\mathcal{U}$ has several pleasant properties: It is decidable in low polynomial time, there are several efficient algorithms to solve it, and it satisfies a "principality property".

It is the "if" part of Question (B) which is more difficult and has not been considered in the past. This paper provides an answer to Question (B), for both of its parts simultaneously. Quite apart from the new light it sheds on β-reduction, the required analysis turns out to have several other benefits, as we outline below.

2. Organization and contributions of the paper. Section 4 defines our unification problem which we call *β-unification*, for lack of a better name, and denote $\boldsymbol{\beta}\mathbf{U}$. This is a higher-order unification problem, with many similarities with other well-known forms of unification, in particular with second-order unification. But there are also many differences. The peculiarities of $\boldsymbol{\beta}\mathbf{U}$ are dictated by results of later sections. The reader is invited to pay close attention to *expansion variables* and the way *expansion substitutions* act on them (Definition 9), which are the most distinctive features of $\boldsymbol{\beta}\mathbf{U}$. A good understanding of these will dispel whatever may appear contrived in $\boldsymbol{\beta}\mathbf{U}$.

Section 5 makes a detour through the system of *intersection types*. Our interest in this system is in that it can be used to characterize the β-strong normalization of terms

[2] The usual statement is slightly different: "If Δ has a solution, then M is simply-typable". We use here the additional non-trivial fact that if M is simply-typable, then M is β-strongly normalizable. There are many proofs for this result in the literature; see [2, Appendix A], [9, Chapter 3] and [17, Chapter 8], among other places.

in the λ-calculus; specifically, a term M is β-strong normalizable iff M is typable in a lean fragment of the system of intersection types (where we omit, among other features, a "top" type that can be assigned to every term). In this paper, we define our own lean fragment of the system, which we call $\boldsymbol{\lambda}^{\rightarrow,\wedge}$. The key result of this section is Theorem 17, stating that M is β-strong normalizable iff M is typable in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$. As there are many different proofs in the literature for what is essentially the same result, we omit the details of our own proof and refer the reader instead to our technical report [11]. Section 4 and Section 5 are largely independent of each other.

Section 6 presents the main results of the paper, by connecting the analyses of the two preceding sections. We develop a procedure Γ which, when applied to a term M, returns a instance $\Gamma(M)$ of $\boldsymbol{\beta}\mathbf{U}$ such that M is typable in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ iff $\Gamma(M)$ has a solution (Theorem 22). Put differently, $\boldsymbol{\beta}\mathbf{U}$ is an appropriate unification problem to characterize typability in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ (in just the same sense that first-order unification can be used to characterize typability in the system of simple types). From this result, we draw several consequences, one of which is a direct answer to Question (B) above (Corollary 23). For a better exposition, we found it appropriate to delay the long technical proof of Theorem 22 to the Appendix.

Section 7 concludes the paper by proposing several open problems. This paper is far from giving a complete examination of $\boldsymbol{\beta}\mathbf{U}$. For example, nothing is said anywhere about a "principality property". More pressing are perhaps the possible connections between $\boldsymbol{\beta}\mathbf{U}$ and other forms of unification, which we list in this last section.

3. Nomenclature and conventions. The notational overhead in this paper is relatively heavy. The following lists collect in one place frequently used names and symbols.

- *A list of often used sets*:

  $\boldsymbol{\mathcal{N}}$ = the set of natural numbers.
  $\boldsymbol{\mathcal{P}}$ = the set of positive integers.
  $\boldsymbol{\mathcal{P}}^+ = \boldsymbol{\mathcal{P}}^* - \{\varepsilon\}$ = the set of non-empty strings of positive integers.
  λ-Var = the set of λ-variables.
  Λ = the set of λ-terms over λ-Var.
  TVar = the set of type variables.
  TVar(σ) = the set of type variables in σ.
  EVar = the set of expansion variables.
  EVar(σ) = the set of expansion variables in σ.
  $\mathcal{T}$, $\mathcal{T}$(TVar), and $\mathcal{T}$(TVar, EVar) are various sets of type expressions.
  $\boldsymbol{\mathcal{E}}$ = the set of expansions.

- *A list of often used metavariables*, which can be further decorated (with a subscript, a superscript, a prime, a double prime, a tilde, etc.):

  i, j, k, ℓ, m, n range over $\boldsymbol{\mathcal{N}}$.
  $\mathbf{p}, \mathbf{q}, \mathbf{r}$ range over $\boldsymbol{\mathcal{P}}^+$.
  v, w, x, y, z range over λ-Var.
  L, M, N, P, Q, R range over Λ.

α, β range over TVar.
F, G range over EVar.
$\overline{F}, \overline{G}$ range over $(\text{EVar})^*$.
e ranges over $\boldsymbol{\mathcal{E}}$.
ρ, σ, τ range over types and type schemes.
Δ ranges over constraint sets.

- *A list of dedicated function names*:

  $\langle\ \rangle_j$ and $\langle\!\langle\ \rangle\!\rangle_j$ are renaming functions.
  Γ and $\Gamma^\star$ are functions generating constraint sets.
  (φ, ψ) ranges over solutions (pairs of substitutions) of constraint sets.

- *Conventions for strings of integers*:

  A string of positive integers is for example: 36 2 101, three numbers separated by blanks. We may write instead: 36,2,101. If the entries in the string are single-digit numbers, we may omit both blanks and commas so that, for example, 3623 may be a string consisting of a single 4-digit number or a string consisting of 4 single-digit numbers (the context will make clear which is the case). We do not stick to a single convention and use whichever is convenient, as long as there is no ambiguity.

- *Other conventions*:

  $M \equiv N$ means "M and N are syntactically identical" (up to α-conversion). We write $P \subset M$ to mean "P is a proper subterm *occurrence* in M". This goes against the usual practice of writing "$P \subset M$" to mean "P is a proper subterm in M", without reference to one of its occurrences. We write $P \subseteq M$ for "$P \subset M$ or $P \equiv M$".

Acknowledgements. The final version of this paper is based on detailed comments and criticisms which the author received from Paweł Urzyczyn for an earlier technical report with the same title ([11]). Special thanks are due to Paweł for all his help.

4. A unification problem. We define a unification problem, called *β-unification* (for want of a better name) and denoted $\boldsymbol{\beta}\mathbf{U}$, which gives an appropriate algebraic characterization of β-strong normalization. The result is proved in Section 6. Formally, the set of *basic type variables* or *basic* T-*variables* is $\text{TVar}_1 = \{\alpha_i^j \mid i \in \boldsymbol{\mathcal{P}},\ j \in \boldsymbol{\mathcal{P}} \cup \{\varepsilon\}\ \}$, and the set of *basic expansion variables* or *basic* E-*variables* is $\text{EVar}_1 = \{F_i \mid i \in \boldsymbol{\mathcal{P}}\}$. The set of *type variables* or T-*variables*, properly extending the basic T-variables, is

$$\text{TVar} = \{\alpha_{i,\mathbf{p}}^j \mid i \in \boldsymbol{\mathcal{P}},\ j \in \boldsymbol{\mathcal{P}} \cup \{\varepsilon\},\ \mathbf{p} \in \boldsymbol{\mathcal{P}}^*\}$$

and the set of *expansion variables* or E-*variables*, properly extending the set of basic E-variables, is

$$\text{EVar} = \{F_{i,\mathbf{p}} \mid i \in \boldsymbol{\mathcal{P}},\ \mathbf{p} \in \boldsymbol{\mathcal{P}}^*\}.$$

We abuse notation and use α (appropriately decorated) to denote both actual members of TVar and metavariables ranging over TVar, and similarly for F (appropriately decorated). Instead of $\alpha_{i,\mathbf{p}}^j$ (resp. $F_{i,\mathbf{p}}$), we may write $\alpha_{\mathbf{q}}^j$ (resp. $F_{\mathbf{q}}$) where $\mathbf{q} = i, \mathbf{p}$.

In what follows, the reader will notice many similarities between "expansion" variables in *β-unification* and "function" variables in second-order unification; but the two are not quite the same, and the relationship between the two is one of the questions we leave open for future examination.

DEFINITION 1. Let $*$ be a type constant, and $\rightarrow$ and $\wedge$ binary type constructors. The set $\mathcal{T}$ of *types* is defined by induction:

1. $* \in \mathcal{T}$.
2. If $\sigma, \tau \in \mathcal{T}$ then $(\sigma \rightarrow \tau) \in \mathcal{T}$.
3. If $\sigma, \tau \in \mathcal{T}$ then $(\sigma \wedge \tau) \in \mathcal{T}$.

The set $\mathcal{T}$ is the set of usual intersection types over the single type constant $*$. For later purposes, we identify a particular subset $\mathcal{T}^{\rightarrow}$ of $\mathcal{T}$:

$$\mathcal{T}^{\rightarrow} \;=\; \{\ *\ \} \cup \{\ (\sigma \rightarrow \tau) \mid \sigma, \tau \in \mathcal{T}\ \}$$

consisting of all types not of the form $(\sigma \wedge \tau)$, i.e. types where the binary type constructor $\wedge$ is not allowed to appear in outermost position but, of course, it is allowed to appear in σ and τ.

DEFINITION 2. Let $X \subseteq \mathrm{TVar}$ and $Y \subseteq \mathrm{EVar}$. The set $\mathcal{T}(X, Y)$ of *type schemes* over X and Y is defined by induction:

1. $X \subset \mathcal{T}(X, Y)$.
2. If $\sigma, \tau \in \mathcal{T}(X, Y)$ then $(\sigma \rightarrow \tau) \in \mathcal{T}(X, Y)$.
3. If $\sigma, \tau \in \mathcal{T}(X, Y)$ then $(\sigma \wedge \tau) \in \mathcal{T}(X, Y)$.
4. If $\sigma \in \mathcal{T}(X, Y)$ and $F \in Y$ then $F\sigma \in \mathcal{T}(X, Y)$.

A particular case of this definition is when we set $X = \mathrm{TVar}$ and $Y = \mathrm{EVar}$. For $\sigma \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$ we denote the set of T-variables occurring in σ by $\mathrm{TVar}(\sigma)$, and the set of E-variables by $\mathrm{EVar}(\sigma)$. We identify an appropriate subset of $\mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$:

$$\mathcal{T}(\mathrm{TVar}) = \mathcal{T}(\mathrm{TVar}, \emptyset) = \ \{\ \sigma \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar}) \mid \mathrm{EVar}(\sigma) = \emptyset\ \}$$

CONVENTIONS 3.

1. Other studies take the binary constructor $\wedge$ commutative and associative, and sometimes also idempotent (e.g. see [4, 22]). Here, $\wedge$ has none of the 3 properties: It is neither associative, nor commutative, nor idempotent.
2. Parentheses are omitted in formal type expressions whenever convenient, provided no ambiguity is introduced:
   (a) As usual, $\rightarrow$ associates to the right,
   so that $\tau_1 \rightarrow \tau_2 \rightarrow \tau_3$ is shorthand for $(\tau_1 \rightarrow (\tau_2 \rightarrow \tau_3))$.
   (b) We also take $\wedge$ to be right-associative,
   so that $\tau_1 \wedge \tau_2 \wedge \tau_3$ is shorthand for $(\tau_1 \wedge (\tau_2 \wedge \tau_3))$.
   (c) $\wedge$ binds more strongly than $\rightarrow$,
   so that $\tau_1 \wedge \tau_2 \rightarrow \tau_3$ is shorthand for $(\tau_1 \wedge \tau_2) \rightarrow \tau_3$, not $\tau_1 \wedge (\tau_2 \rightarrow \tau_3)$.
3. $\tau_1 \equiv \tau_2$ means "τ_1 and τ_2 are syntactically identical" modulo the preceding conventions.

Definitions 4 and 5 introduce appropriate restrictions satisfied by type schemes derived from λ-terms (Section 6).

DEFINITION 4. The set of *E-paths* is $(\mathrm{EVar})^*$. Every E-path $\overline{G}$ is therefore an arbitrary sequence $F_{\mathbf{p}_1} F_{\mathbf{p}_2} \cdots F_{\mathbf{p}_n}$ of E-variables, for some $n \geq 0$. We define a function E-*path* from $(\mathrm{TVar} \cup \mathrm{EVar}) \times \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$ to finite subsets of $(\mathrm{EVar})^*$. We first define E-*path*(α, τ) for every $\alpha \in \mathrm{TVar}$ and $\tau \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$, by induction:

1. E-*path*$(\alpha, \beta) = \begin{cases} \{\varepsilon\}, \text{ if } \alpha \equiv \beta, \\ \emptyset, \quad \text{if } \alpha \not\equiv \beta. \end{cases}$
2. E-*path*$(\alpha, \sigma \to \tau) =$ E-*path*$(\alpha, \sigma) \ \cup$ E-*path*(α, τ).
3. E-*path*$(\alpha, \sigma \wedge \tau) =$ E-*path*$(\alpha, \sigma) \ \cup$ E-*path*(α, τ).
4. E-*path*$(\alpha, F\sigma) = \{F\overline{G} \mid \overline{G} \in$ E-*path*$(\alpha, \sigma)\}$.

We next define E-*path*(F, τ) for every $F \in \mathrm{EVar}$ and $\tau \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$, by induction:

1. E-*path*$(F, \alpha) = \emptyset$.
2. E-*path*$(F, \sigma \to \tau) =$ E-*path*$(F, \sigma) \ \cup$ E-*path*(F, τ).
3. E-*path*$(F, \sigma \wedge \tau) =$ E-*path*$(F, \sigma) \ \cup$ E-*path*(F, τ).
4. E-*path*$(F, F'\sigma) = \begin{cases} \{F'\overline{G} \mid \overline{G} \in \text{E-}\mathit{path}(F, \sigma)\}, & \text{if } F \not\equiv F', \\ \{\varepsilon\} \ \cup \ \{F'\overline{G} \mid \overline{G} \in \text{E-}\mathit{path}(F, \sigma)\}, & \text{if } F \equiv F'. \end{cases}$

DEFINITION 5. We say that a type scheme $\tau \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$ is *well-behaved* if it satisfies 4 conditions:

1. For every $\alpha \in \mathrm{TVar}(\tau)$, E-*path*$(\alpha, \tau)$ is a singleton set.
2. For every $F \in \mathrm{EVar}(\tau)$, E-*path*$(F, \tau)$ is a singleton set.
3. For all $\alpha^i_{\mathbf{p}}, \alpha^j_{\mathbf{q}} \in \mathrm{TVar}(\tau)$, if $\mathbf{p} \neq \mathbf{q}$ then neither $\mathbf{p}$ nor $\mathbf{q}$ is a prefix of the other.
4. For all $F_{\mathbf{p}}, F_{\mathbf{q}} \in \mathrm{EVar}(\tau)$, if $\mathbf{p} \neq \mathbf{q}$ then neither $\mathbf{p}$ nor $\mathbf{q}$ is a prefix of the other.

In words, conditions 1 and 2 require that the E-path of every T-variable and the E-path of every E-variable be uniquely defined; put differently still, in a well-behaved type scheme, the sequence of E-variables encountered as we go from the root of the type scheme to an occurrence of an E- or T-variable is always the same, for all occurrences of this variable. (A similar property defines the so-called "stratified terms" in second-order unification [16, 19].) None of these 4 conditions is implied by the others. This is clear for conditions 3 and 4. The next example shows the independence of conditions 1 and 2.

EXAMPLE 6. Let $\sigma \equiv FF'\alpha \wedge F'\alpha' \to \beta$. Then E-*path*$(\alpha, \sigma) = \{FF'\}$, E-*path*$(\alpha', \sigma) = \{F'\}$ and E-*path*$(\beta, \sigma) = \{\varepsilon\}$. Thus σ satisfies condition 1 in Definition 5, but does not satisfy condition 2, because E-*path*$(F', \sigma) = \{\varepsilon, F\}$. Now let $\tau \equiv \alpha \wedge F\alpha \wedge FF'\alpha \to \beta$. Then E-*path*$(F, \tau) = \{\varepsilon\}$ and E-*path*$(F', \tau) = \{F\}$, which implies that τ satisfies condition 2 in Definition 5. But τ does not satisfy condition 1, because E-*path*$(\alpha, \tau) = \{\varepsilon, F, FF'\}$.

DEFINITION 7. A *renaming function* has two disjoint parts: an injection from TVar to TVar, and an injection from EVar to EVar, extended by induction to an injection from $\mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$ to $\mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$. A particular kind of renaming functions is now defined, others are considered later. For every $j \in \boldsymbol{\mathcal{P}}$, define the renaming function $\langle\ \rangle_j : \mathcal{T}(\mathrm{TVar}, \mathrm{EVar}) \to \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$, by induction:

1. $\langle \alpha^i_{\mathbf{p}} \rangle_j = \alpha^i_{\mathbf{p},j}$.
2. $\langle \sigma \to \tau \rangle_j = \langle \sigma \rangle_j \to \langle \tau \rangle_j$.
3. $\langle \sigma \wedge \tau \rangle_j = \langle \sigma \rangle_j \wedge \langle \tau \rangle_j$.
4. $\langle F_{\mathbf{p}}\ \sigma \rangle_j = F_{\mathbf{p},j}\ \langle \sigma \rangle_j$.

DEFINITION 8. The symbol $\square$ denotes a "hole". The set $\boldsymbol{\mathcal{E}}$ of *expansions* is defined by induction:

1. $\square \in \boldsymbol{\mathcal{E}}$.
2. If $e, e' \in \boldsymbol{\mathcal{E}}$ then $(e \wedge e') \in \boldsymbol{\mathcal{E}}$.

In words, an *expansion* e is a non-empty binary tree, where every internal node is "$\wedge$" and every leaf node is "$\square$". If e has n holes, $n \geq 1$, we may write $\square^{(1)}, \square^{(2)}, \ldots, \square^{(n)}$, to denote these n holes in their order of occurrence in e from left to right. If $\tau_1, \ldots, \tau_n$ are n types (or type schemes, respectively), we write $e[\tau_1, \ldots, \tau_n]$ to denote the result of placing τ_i in hole $\square^{(i)}$ for every $i \in \{1, \ldots, n\}$. The resulting expression $e[\tau_1, \ldots, \tau_n]$ is a type (or a type scheme, respectively).[3]

DEFINITION 9. An *expansion substitution* or, more simply, an E-*substitution* is a map φ : EVar $\to \boldsymbol{\mathcal{E}}$ such that $\varphi(F) = \square$ for almost all $F \in$ EVar. Such a substitution is extended to $\varphi : \mathcal{T}(\text{TVar}, \text{EVar}) \to \mathcal{T}(\text{TVar})$ by induction on $\mathcal{T}(\text{TVar}, \text{EVar})$:

1. $\varphi(\alpha) = \alpha$.
2. $\varphi(\sigma \to \tau) = \varphi(\sigma) \to \varphi(\tau)$.
3. $\varphi(\sigma \wedge \tau) = \varphi(\sigma) \wedge \varphi(\tau)$.
4. $\varphi(F\sigma) = e[\varphi(\langle\sigma\rangle_1), \ldots, \varphi(\langle\sigma\rangle_n)]$ where $e = \varphi(F)$ and e has $n \geq 1$ holes.

There are several subtleties connected with the 4th clause. First, observe that if $\varphi(F) = \square$ and σ does not mention any E-variables, then $\varphi(F\sigma)$ is not σ, but rather $\langle\sigma\rangle_1$ which is σ with all T-variables uniformly renamed. Moreover, observe that φ is applied in "outside-in" fashion. If φ were applied "inside-out", then we would write instead $\varphi(F\sigma) = e[\langle\varphi(\sigma)\rangle_1, \ldots, \langle\varphi(\sigma)\rangle_n)]$ and the result would be very different. For example, let $\sigma \equiv F(G\alpha \to \alpha') \to \alpha''$ and $\varphi(F) = \varphi(G) = \square$ and $\varphi(G_1) = \square \wedge \square$. Applying φ to σ "outside-in" in this case produces $(\alpha_{11} \wedge \alpha_{12} \to \alpha'_1) \to \alpha''$, whereas applying φ to σ "inside-out" produces $(\alpha_{11} \to \alpha'_1) \to \alpha''$.

PROPOSITION 10.

1. *If $\sigma \in \mathcal{T}(\text{TVar}, \text{EVar})$ is well-behaved, then so is $\langle\sigma\rangle_j$, for every $j \in \boldsymbol{\mathcal{P}}$.*
2. *If $\sigma \in \mathcal{T}(\text{TVar}, \text{EVar})$ is well-behaved and φ : EVar $\to \boldsymbol{\mathcal{E}}$, then $\varphi(\sigma)$ is well-behaved.*

PROOF. Part 1 is clear from Definitions 5 and 7. A formal proof is by induction on $\mathcal{T}(\text{TVar}, \text{EVar})$. Part 2 is by induction on $\mathcal{T}(\text{TVar}, \text{EVar})$, using part 1 also. ∎

DEFINITION 11. A *type substitution* or, more simply, a T-*substitution* is a function ψ : TVar $\to \mathcal{T}^{\to}$ such that $\psi(\alpha) = \ast$ for almost all $\alpha \in$ TVar, extended in the usual way to $\psi : \mathcal{T}(\text{TVar}) \to \mathcal{T}$ by induction on $\mathcal{T}(\text{TVar})$:

[3] "Expansions" in this paper are unrelated to "expansions" as defined in various articles by researchers at the University of Turin, see [6] and [7, Definition 2.10] and some of the references cited therein.

1. $\psi(\sigma \to \tau) = \psi(\sigma) \to \psi(\tau)$.
2. $\psi(\sigma \wedge \tau) = \psi(\sigma) \wedge \psi(\tau)$.

Note the basis of the preceding induction is a function $\psi : \mathrm{TVar} \to \mathcal{T}^{\to}$ rather than $\psi : \mathrm{TVar} \to \mathcal{T}$. These are not equivalent choices, as illustrated by Example 13.

DEFINITION 12. We define a *constraint* to be an equation of the form $\sigma = \tau$ where $\sigma, \tau \in \mathcal{T}(\mathrm{TVar}, \mathrm{EVar})$. The constraint $\sigma = \tau$ is *well-behaved* if the type scheme $\sigma \wedge \tau$ is well-behaved. An *instance* Δ of $\beta\mathbf{U}$ is a well-behaved finite set of constraints, i.e.

$$\Delta = \{\sigma_1 = \tau_1,\ \sigma_2 = \tau_2,\ \ldots\ ,\ \sigma_n = \tau_n\}$$

such that the type scheme $\sigma_1 \wedge \tau_1 \wedge \sigma_2 \wedge \tau_2 \wedge \cdots \wedge \sigma_n \wedge \tau_n$ is well-behaved. A *solution* for Δ is a pair (φ, ψ) where φ is an E-substitution and ψ is a T-substitution, such that $\psi(\varphi(\sigma_i)) \equiv \psi(\varphi(\tau_i))$ for $i = 1, \ldots, n$, in which case we write $(\varphi, \psi) \models \Delta$. A particular case is $\Delta = \emptyset$, the empty set of constraints, which always has a solution.

If Δ is the constraint set above, then $\bigwedge\!\!\!\bigwedge \Delta$ is a shorthand notation:

$$\bigwedge\!\!\!\bigwedge \Delta = \sigma_1 \wedge \tau_1 \wedge \sigma_2 \wedge \tau_2 \wedge \cdots \wedge \sigma_n \wedge \tau_n .$$

Notions and functions defined earlier in this section for type schemes are extended to constraint sets in the obvious way. For example, if Δ is a constraint set, then $\mathrm{TVar}(\Delta) = \mathrm{TVar}(\bigwedge\!\!\!\bigwedge \Delta)$. The sets $\mathrm{EVar}(\Delta)$, E-*path*(α, Δ), E-*path*(F, Δ), etc., are defined similarly.

EXAMPLE 13. The constraint set $\Delta = \{\alpha_1 = \alpha_2 \wedge \alpha_3\}$ has no solution, in our sense above. However, there is an extended sense of E-substitution, according to which it is a function ψ from TVar to $\mathcal{T}$ rather than to $\mathcal{T}^{\to}$. In the extended sense, not considered in the rest of the paper, Δ has a solution (φ, ψ) such that: $\varphi(F) = \Box$ for all $F \in \mathrm{EVar}$, $\psi(\alpha_1) = * \wedge *$ and $\psi(\alpha) = *$ for all $\alpha \in \mathrm{TVar} - \{\alpha_1\}$. ∎

We point out a peculiarity of $\beta\mathbf{U}$, which distinguishes it from other forms of unification. Finding a solution (φ, ψ) for a constraint set Δ generally requires that φ and ψ explicitly assign values to, not only (the finitely many) E-variables and T-variables occurring in Δ, but also (finitely many) other E-variables and T-variables not occurring in Δ. Moreover, if a solution must assign a value to, say, E-variable $F_{i,\mathbf{p}}$ then it does not have to assign a value to its "parent" F_i which explicitly appears in Δ; thus, such a variable F_i in Δ can be viewed as a "representative" of a number of its offsets $F_{i,\mathbf{p}_1}, \ldots, F_{i,\mathbf{p}_n}$, where the number $n \geq 1$ is determined by a solution for Δ. This is illustrated by the next example and, again, by Example 20 in Section 6.

EXAMPLE 14. Consider the constraint set $\Delta = \{\alpha = F\beta \to \beta', F\alpha' = F(G\alpha'' \to \beta)\}$. For notational convenience, we use α, α', β, etc., instead of formal T-variables of the form α_i^j, and F and G instead of formal E-variables of the form F_i. It is readily checked that Δ is well-behaved. Moreover, Δ has infinitely many solutions, and here is a particular one (φ, ψ). Let

$$\varphi(F) = \varphi(G_1) = \varphi(G_2) = \Box \wedge \Box,$$
$$\psi(\alpha) = \beta_1 \wedge \beta_2 \to \beta', \quad \psi(\alpha'_1) = \alpha''_{1,1} \wedge \alpha''_{1,2} \to \beta_1, \quad \psi(\alpha'_2) = \alpha''_{2,1} \wedge \alpha''_{2,2} \to \beta_2.$$

Given the above assignment of values to F, G_1 and G_2, all other E-variables (in particular the "parent" G) need not be assigned any values — or they can all be assigned the default value $\square$. Similarly, given the above assignment of values to α, α_1' and α_2', all other T-variables (in particular the "parent" α') need not be assigned any values — or they can all be assigned the default value $\ast$. ∎

We also note that, although some of the restrictions imposed on $\beta\mathbf{U}$ may now seem arbitrary, they were entirely dictated by the needs to prove the equivalence between β-SN and $\beta\mathbf{U}$, as expressed by Corollary 23. Among such restrictions are: applying E-substitutions in "outside-in" fashion (Definition 9), restricting the range of T-substitutions to $\mathcal{T}^{\rightarrow}$ (Definition 11), and requiring that instances of $\beta\mathbf{U}$ be well-behaved (Definition 12).

5. The system of intersection types. The two conditions in the next definition are standard in the literature (see, for example, [17, Chapter 11]), whenever constraints are formulated in relation to typability of λ-terms in a type inference system.

DEFINITION 15. A λ-term $M \in \Lambda$ is *well-named* if it satisfies two conditions:

1. No variable in M has more than one binding occurrence.
2. The bound and free variables in M are disjoint sets.

PROPOSITION 16. *For every λ-term $M \in \Lambda$, we can effectively define a well-named λ-term $N \in \Lambda$ such that $M \equiv N$.*

PROOF. By induction on the definition of M. ∎

System $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ is our lean version of the system of intersection types. The type-inference rules of $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ are collected together in the table at the end of Section 5. (Similar but not quite identical restrictions of the system of intersection types are extensively studied in [22] and [23], and also in earlier studies, e.g. [5].) In the definition of $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ below, A is a *type assignment*, i.e. a partial function from λ-Var to $\mathcal{T}$ with finite domain of definition, written as a finite list of pairs, as in

$$x_1 : \tau_1, \ \ldots \ , x_k : \tau_k$$

for some distinct $x_1, \ldots, x_k \in \lambda$-Var, some $\tau_1, \ldots, \tau_k \in \mathcal{T}$ and some $k \geq 0$. If A and B are type assignments defined over the same finite subset $X \subset \lambda$-Var, then $A \wedge B$ is a new type assignment given by:

$$(A \wedge B)(x) = A(x) \wedge B(x)$$

for every $x \in X$. If A and B have distinct domains of definition, we do not need to define $A \wedge B$. We take $\wedge$ non-associative, non-commutative and non-idempotent (Conventions 3). More generally, if $e \in \boldsymbol{\mathcal{E}}$ is an expansion with n holes and $B_1, \ldots, B_n$ are type assignments all defined over the same finite subset $X \subset \lambda$-Var, then $e[B_1, \ldots, B_n]$ is the type assignment given by

$$e[B_1, \ldots, B_n](x) = e[B_1(x), \ldots, B_n(x)]$$

for every $x \in X$. Observe that, in our system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$, the rule ABS-K is not a special case of the rule ABS-I. This is because there is no "weakening" rule in our system (which allows to add redundant type assumptions to a type assignment) and, moreover, if there is a proof for the sequent $A \vdash M : \tau$ in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ where M is well-named and $x \notin \mathrm{FV}(M)$, then $A(x)$ is not defined.

On the other hand, if M is well-named, $x \in \mathrm{FV}(M)$ and there is a proof in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ for the sequent $A \vdash M : \tau$, then $A(x)$ is necessarily defined. If there are $n \geq 1$ invocations of rule VAR in this proof to derive n types for x, then

$$A(x) = e[\sigma_1, \sigma_2, \ldots, \sigma_n]$$

for some $e \in \boldsymbol{\mathcal{E}}$, an expansion with n holes, and some $\sigma_i \in \mathcal{T}^{\rightarrow}$ for $i = 1, \ldots, n$. In general, if m is the number of occurrences of x in M, then $n \geq m$, i.e. it is not necessarily the case that $n = m$.

To illustrate this last point, consider the subterm $N \equiv (v^{(1)}(v^{(2)}(v^{(3)}x)))$ of M in Example 20 below. There is a proof of the sequent $\emptyset \vdash M : \tau$ in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ for some $\tau \in \mathcal{T}^{\rightarrow}$. This proof contains therefore a subderivation for a sequent $A \vdash N : \sigma$ for some type assignment A and type $\sigma \in \mathcal{T}^{\rightarrow}$. It is not difficult to verify that $A(z)$ is undefined for all $z \in \lambda\text{-Var} - \{v, x\}$, with $A(v) = e[\sigma_1, \ldots, \sigma_7]$ and $A(x) = e'[\sigma'_1, \ldots, \sigma'_8]$ where $\sigma_1, \ldots, \sigma_7, \sigma'_1, \ldots, \sigma'_8 \in \mathcal{T}^{\rightarrow}$.

THEOREM 17. *For every well-named $M \in \Lambda$, the term M is β-SN iff M is typable in the system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$.*

PROOF. Although our system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ is slightly different from other systems of intersection types, this theorem is no surprise for the reader familiar with intersection types. Nevertheless, what is interesting is that there are several very different approaches to prove what is essentially the same result. For some of these approaches, we refer the reader to [13, 15, 18, 21, 22, 23] as well as to some of the references cited therein. Yet another approach is based on the results of [10] after appropriate adjustments (mostly required by the fact that $\wedge$ is an associative binary constructor in [10], which it is not here). Details of this last approach are included in the technical report version of this paper [11]. ∎

In this report we do not pursue the study of system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ further. Let us point out at least one peculiarity of $\boldsymbol{\lambda}^{\rightarrow,\wedge}$: It does not satisfy the property of *subject reduction*, i.e. types in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ are not always preserved by β-reduction and we may have the following situation:

$$\vdash M : \sigma \qquad \text{and} \qquad M \longrightarrow_\beta N \qquad \text{but} \qquad \not\vdash N : \sigma$$

(We take M to be a closed λ-term, for simplicity, so that the type assignment to the left of "$\vdash$" is empty.) For this to happen, for example, we may take $M \equiv (\lambda x.((\lambda y.P)Q))$ where x and y do not occur in P and x occurs twice in Q. In particular, $((\lambda y.P)Q)$ is a K-redex. In this case, if $\vdash M : \sigma$, it must be that $\sigma = \tau_1 \wedge \tau_2 \rightarrow \rho$, where $\tau_1, \tau_2, \rho \in \mathcal{T}^{\rightarrow}$ and τ_1 and τ_2 are the types assigned to the two occurrences of x in Q. We then have:

$$\begin{array}{lll} \vdash (\lambda x.((\lambda y.P)Q)) : \tau_1 \wedge \tau_2 \rightarrow \rho & \quad \text{and} \quad & (\lambda x.((\lambda y.P)Q)) \longrightarrow_\beta (\lambda x.P) \\ \text{but} \qquad \not\vdash (\lambda x.P) : \tau_1 \wedge \tau_2 \rightarrow \rho & & \end{array}$$

because x does not occur in P. On the other hand, $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ does satisfy a weaker form of subject reduction: Typability (as opposed to types) in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ is preserved by β-reduction. This is so because:

1. Typability in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ is equivalent to β-SN.

2. If M is β-SN and $M \longrightarrow_\beta N$ then N is also β-SN.

In the preceding example, even though $\not\vdash (\lambda x.P) : \tau_1 \wedge \tau_2 \rightarrow \rho$, it is still the case that $\vdash (\lambda x.P) : \tau \rightarrow \rho$ for some $\tau \in \mathcal{T}^{\rightarrow}$.

Clearly, the peculiarity described above is due to the presence of K-redexes. In fact, when restricted to λI-terms, system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ does satisfy the subject-reduction property (proof omitted). Let us also point out that $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ can be adjusted in order to satisfy subject-reduction in general.

The peculiarities of system $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ were dictated by the needs of the results of the next section, connecting $\boldsymbol{\beta}\mathbf{U}$ and β-SN. Specifically, $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ was defined and adjusted in order to push the proof of Theorem 22 through.

System $\boldsymbol{\lambda}^{\rightarrow,\wedge}$

VAR $\qquad x : \tau \;\vdash\; x : \tau \qquad \tau \in \mathcal{T}^{\rightarrow}$

ABS-I
$$\frac{A,\ x : \sigma \;\vdash\; M : \tau \qquad x \in \mathrm{FV}(M)}{A \;\vdash\; (\lambda x.M) : (\sigma \rightarrow \tau)}$$

ABS-K
$$\frac{A \;\vdash\; M : \tau \qquad x \notin \mathrm{FV}(M) \qquad \sigma \in \mathcal{T}^{\rightarrow}}{A \;\vdash\; (\lambda x.M) : (\sigma \rightarrow \tau)}$$

APP
$$\frac{A \;\vdash\; M : (e[\sigma_1, \ldots, \sigma_n] \rightarrow \tau) \qquad B_1 \;\vdash\; N : \sigma_1 \;\cdots\; B_n \;\vdash\; N : \sigma_n}{A \wedge e[B_1, \ldots, B_n] \;\vdash\; (MN) : \tau}$$

$\tau, \sigma_1, \ldots, \sigma_n \in \mathcal{T}^{\rightarrow}$ with $n \geq 1$ and $e \in \boldsymbol{\mathcal{E}}$

6. Typability in the system of intersection types. Let $M \in \Lambda$. Formally, the set of λ-variables is λ-Var $= \{x_i \mid i \in \boldsymbol{\mathcal{P}}\}$. For the same variable x_i, the occurrences of x_i (free or bound but not binding) in M are uniquely identified by their occurrence numbers, as

$$x_i^{(j_1)}, x_i^{(j_2)}, \ \ldots\ , x_i^{(j_n)}$$

for some $j_1, j_2, \ldots, j_n \in \boldsymbol{\mathcal{P}}$. Subscripts are part of the variable name, superscripts are not. For simplicity of notation, we often assign occurrence numbers consecutively, starting with 1, as M is scanned from left-to-right, as

$$x_i^{(1)}, x_i^{(2)}, \ \ldots\ , x_i^{(n)}$$

but our analysis is independent of this numbering scheme. All that matters is that an occurrence of x_i in M is uniquely identified by an occurrence number.

We define a procedure which, given a well-named λ-term M, generates a finite set $\Gamma(M)$ of constraints. If Δ is a set of constraints and $F \in \mathrm{EVar}$, we write $F\Delta$ to denote the set of constraints:

$$F\Delta = \{F\sigma = F\tau \mid \sigma = \tau \text{ is a constraint in } \Delta\}$$

The constraints in $\Gamma(M)$ are written over the sets of basic T-variables and basic E-variables:

$$\mathrm{TVar}_1 = \{\alpha_i^j \mid i \in \boldsymbol{\mathcal{P}}, j \in \boldsymbol{\mathcal{P}} \cup \{\varepsilon\}\} \quad \text{and} \quad \mathrm{EVar}_1 = \{F_i \mid i \in \boldsymbol{\mathcal{P}}\}$$

For a fixed $i \in \boldsymbol{\mathcal{P}}$, all type variables of the form α_i^j correspond to λ-variable x_i. For convenience, distinguish a subset TVar_{aux} of TVar_1 whose members do not correspond to λ-variables in M:

$$\mathrm{TVar}_{aux} = \{\alpha_i \mid i \in \boldsymbol{\mathcal{P}},\ x_i \text{ does not occur in } M\}$$

We reserve β (possibly decorated) as a metavariable ranging over TVar_{aux} ("aux" is for "auxiliary").

Definition 18 (Procedure Γ to generate constraints). Let $M \in \Lambda$ be well-named. Simultaneously with a set of constraints $\Gamma(M)$, we define a type scheme $\tau(M)$ and, for every λ-variable x occurring free in M, a type scheme $\rho(M, x)$. By induction on well-named λ-terms:

1. *Variables* $x_i^{(j)}$:
   - $\Gamma(x_i^{(j)}) = \emptyset$.
   - $\tau(x_i^{(j)}) = \alpha_i^j$.
   - $\rho(x_i^{(j)}, x_i) = \alpha_i^j$.
2. *Applications* (NP):
   - $\Gamma(NP) = \Gamma(N) \cup F\Gamma(P) \cup \{\tau(N) = F\tau(P) \to \beta\}$,
     for a fresh $F \in \mathrm{EVar}_1$ and a fresh $\beta \in \mathrm{TVar}_{aux}$.
   - $\tau(NP) = \beta$.
   - $\rho((NP), x) = \begin{cases} \rho(N, x), & \text{if } x \in \mathrm{FV}(N) \text{ and } x \notin \mathrm{FV}(P), \\ F\rho(P, x), & \text{if } x \notin \mathrm{FV}(N) \text{ and } x \in \mathrm{FV}(P), \\ (\rho(N, x) \wedge F\rho(P, x)), & \text{if } x \in \mathrm{FV}(N) \text{ and } x \in \mathrm{FV}(P), \\ \text{undefined}, & \text{if } x \notin \mathrm{FV}(N) \text{ and } x \notin \mathrm{FV}(P). \end{cases}$
3. *Abstractions* $(\lambda x_i\ N)$, where the free occurrences of x_i in N are $x_i^{(1)}, \ldots, x_i^{(m)}$ as N is scanned from left to right, for some $m \geq 0$:
   - $\Gamma(\lambda x_i\ N) = \Gamma(N)$.
   - $\tau(\lambda x_i\ N) = \begin{cases} \rho(N, x_i) \to \tau(N), & \text{if } m \geq 1, \\ \alpha_i \to \tau(N), & \text{if } m = 0. \end{cases}$
   - $\rho((\lambda x_i\ N), x) = \rho(N, x)$ (note that $x \not\equiv x_i$ necessarily).

Given a fixed numbering of the occurrences of the same λ-variable in M, for every λ-variable in M, the process of going from M to $\Gamma(M)$ is uniquely determined up to the choice of auxiliary type variables.

Proposition 19. *For every well-named $M \in \Lambda$, the set $\Gamma(M)$ is a well-behaved set of constraints.*

PROOF. Because $\mathrm{TVar}(\Gamma(M)) \subset \mathrm{TVar}_1$ and $\mathrm{EVar}(\Gamma(M)) \subset \mathrm{EVar}_1$, conditions 3 and 4 in Definition 5 are automatically satisfied. We only need to show that:

1. For every $\alpha \in \mathrm{TVar}(\Gamma(M))$, E-*path*$(\alpha, \bigwedge\!\!\!\bigwedge\Gamma(M))$ is uniquely defined.
2. For every $F \in \mathrm{EVar}(\Gamma(M))$, E-*path*$(F, \bigwedge\!\!\!\bigwedge\Gamma(M))$ is uniquely defined.

These two properties are shown simultaneously by a straightforward induction on M. Define

$$\Gamma^\star(M) \;=\; \bigwedge\!\!\!\bigwedge\Gamma(M) \;\wedge\; \bigwedge\!\!\!\bigwedge\{\, \rho(M,x) \mid x \text{ free in } M \,\} \;\wedge\; \tau(M)$$

To push the induction through, we prove a slight generalization of the two preceding properties, where $\Gamma(M)$ is replaced by $\bigwedge\!\!\!\bigwedge\Gamma^\star(M)$ throughout, together with two additional properties (below). Let $\{N_1, \ldots, N_\ell\} = \{N | N \subseteq M\}$. The two additional properties are, for all $i, j \in \{1, \ldots, \ell\}$:

3. If $N_i \cap N_j = \emptyset$ then $\mathrm{TVar}(\Gamma^\star(N_i)) \cap \mathrm{TVar}(\Gamma^\star(N_j)) = \emptyset$.
4. If $N_i \cap N_j = \emptyset$ then $\mathrm{EVar}(\Gamma^\star(N_i)) \cap \mathrm{EVar}(\Gamma^\star(N_j)) = \emptyset$.

The only non-trivial case of the induction is when $N_k \equiv (N_i N_j)$. Properties 1 to 4 for this case are satisfied using the induction hypothesis for N_i and N_j, together with the fact that a fresh $\beta \in \mathrm{TVar}_{aux}$ and a fresh $F \in \mathrm{EVar}_1$ are introduced. We omit the straightforward details. ∎

EXAMPLE 20. Let $M \equiv (\lambda v.\lambda x.v^{(1)}\ (v^{(2)}\ (v^{(3)}x)))\ (\lambda w.\lambda y.w^{(1)}\ (w^{(2)}y))$. (For better readability, instead of the formal λ-variables x_1, x_2, x_3, and x_4, we use x, y, v, and w; instead of the formal T-variables α_1^j, α_2^j, etc., we use α_x^j, α_y^j, etc.; instead of the formal E-variables F_1, F_2, F_3, etc., we use F, G, H, etc.) The set $\Gamma(M)$ contains 6 constraints:

(1) $$lrlllHG\alpha_v^3 = HG(F\alpha_x \to \beta_1)$$

(2) $$H\alpha_v^2 = H(G\beta_1 \to \beta_2)$$

(3) $$\alpha_v^1 = H\beta_2 \to \beta_3$$

(4) $$KJ\alpha_w^2 = KJ(I\alpha_y \to \beta_4)$$

(5) $$K\alpha_w^1 = K(J\beta_4 \to \beta_5)$$

(6) $$\alpha_v^1 \wedge H(\alpha_v^2 \wedge G\alpha_v^3) \to (HGF\alpha_x \to \beta_3) = K(\alpha_w^1 \wedge J\alpha_w^2 \to (JI\alpha_y \to \beta_5)) \to \beta_6$$

These 6 constraints are produced by applying the inductive procedure given in Definition 18 to the term M. We do not give the details, except for the last step, which is an application. Let $N \equiv (\lambda v.\lambda x.v^{(1)}\ (v^{(2)}\ (v^{(3)}x)))$ and $P \equiv (\lambda w.\lambda y.w^{(1)}\ (w^{(2)}y))$. Then $\Gamma(N)$ is the set of constraints $\{(1),(2),(3)\}$ and $\Gamma(P)$ contains 2 constraints:

(4′) $$J\alpha_w^2 = J(I\alpha_y \to \beta_4)$$

(5′) $$\alpha_w^1 = J\beta_4 \to \beta_5$$

Observe that (4′) and (5′) are (4) and (5), respectively, with the outermost E-variable K crossed out. Moreover,

$$\tau(N) \equiv \alpha_v^1 \wedge H(\alpha_v^2 \wedge G\alpha_v^3) \to (HGF\alpha_x \to \beta_3)$$
$$\tau(P) \equiv \alpha_w^1 \wedge J\alpha_w^2 \to (JI\alpha_y \to \beta_5)$$

Hence,

$$\Gamma(M) \;=\; \Gamma(NP) \;=\; \Gamma(N) \;\cup\; K\Gamma(P) \;\cup\; \{\tau(N) = K\tau(P) \to \beta_6\}$$

where K and β_6 are fresh. A solution (not unique) for $\Gamma(M)$ is given by a pair (φ, ψ) where (for convenience we write F_{11} instead of $F_{1,1}$, F_{12} instead of $F_{1,2}$, etc.):

- $\varphi(F') = \Box$ for all $F' \in \mathrm{EVar} - \{F_{11}, F_{12}, F_{21}, F_{22}, G_1, G_2, H, K\}$.
- $\varphi(F_{11}) = \varphi(F_{12}) = \varphi(F_{21}) = \varphi(F_{22}) = \varphi(G_1) = \varphi(G_2) = \varphi(H) = \Box \wedge \Box$.
- $\varphi(K) = \Box \wedge ((\Box \wedge (\Box \wedge \Box)) \wedge (\Box \wedge (\Box \wedge \Box)))$.

We omit the details of ψ, easily obtained by inspecting the constraints in $\varphi(\Gamma(M))$. The existence of a solution (φ, ψ) for $\Gamma(M)$ implies that M is typable in $\boldsymbol{\lambda}^{\to,\wedge}$, by Theorem 22 below. Of course, the typability of M in $\boldsymbol{\lambda}^{\to,\wedge}$ is simple enough to be checked directly. More interesting is the relationship between solutions (φ, ψ) for $\Gamma(M)$ and proofs of a sequent $\emptyset \vdash M : \tau$ in $\boldsymbol{\lambda}^{\to,\wedge}$. We do not investigate this relationship in this paper, but here is a glimpse for the interested reader. For the particular solution (φ, ψ) defined above, the corresponding proof of the sequent $\emptyset \vdash M : \tau$ is such that, if ρ and ρ' are the types of the bindings "λv" and "λx", then $\rho \equiv e[\sigma_1, \ldots, \sigma_7]$ and $\rho' \equiv e'[\sigma'_1, \ldots, \sigma'_8]$, for some $\sigma_1, \ldots, \sigma_7, \sigma'_1, \ldots, \sigma'_8 \in \mathcal{T}^{\to}$, where $e \equiv \varphi(K)$ and

$$e' \equiv \varphi(HGF) = (\varphi(F_{11}) \wedge \varphi(F_{12})) \wedge (\varphi(F_{21}) \wedge \varphi(F_{22}))$$

Complementary comments are in the paragraph preceding Theorem 17. ∎

Example 21. Let $N \equiv (\lambda x.x^{(1)}x^{(2)})\ (\lambda y.y^{(1)}y^{(2)})$. $\Gamma(N)$ is the following set of constraints:

$$\alpha^1_x = F\alpha^2_x \to \beta_1 \tag{1}$$
$$H\alpha^1_y = H(G\alpha^2_y \to \beta_2) \tag{2}$$
$$\alpha^1_x \wedge F\alpha^2_x \to \beta_1 = H(\alpha^1_y \wedge G\alpha^2_y \to \beta_2) \to \beta_3 \tag{3}$$

The set $\Gamma(M)$ does not have a solution, corresponding to the fact that M is not typable in $\boldsymbol{\lambda}^{\to,\wedge}$, by Theorem 22. ∎

Theorem 22. *For every well-named $M \in \Lambda$, M is typable in $\boldsymbol{\lambda}^{\to,\wedge}$ iff $\Gamma(M)$ has a solution.*

Proof. This is a long technical induction, delayed to the Appendix. ∎

The next result is our promised characterization of β-SN via the unification problem $\boldsymbol{\beta}\mathbf{U}$.

Corollary 23. *For every well-named $M \in \Lambda$, the term M is β-SN iff $\Gamma(M)$ has a solution.*

Proof. Immediate from Theorem 17 and Theorem 22. ∎

Corollary 24. *There is no algorithm which, given an arbitrary well-named $M \in \Lambda$, can decide whether the set of constraints $\Gamma(M)$ has a solution.*

Proof. Immediate from Corollary 23, using the fact that it is undecidable whether an arbitrary λ-term is β-SN. This is a "folk" result, which is not explicitly stated in [2], but here is a simple proof for it, based on a similar proof in [15]. The partial recursive functions are λI-definable [2, Theorem 9.2.16], which implies it is undecidable whether an arbitrary λI-term has a β-nf [2, Definition 9.2.3], by the undecidability of the Halting Problem. But a λI-term has a β-nf iff it is β-SN [2, Corollary 11.3.5]. Hence, it is undecidable whether an arbitrary λI-term is β-SN and, more generally, it is undecidable whether an arbitrary λ-term is β-SN. ∎

THEOREM 25. *There is a semi-decision procedure which, given an arbitrary well-named $M \in \Lambda$, terminates iff the set of constraints $\Gamma(M)$ has a solution. Moreover, if and when the procedure terminates, it returns a solution (φ, ψ) for $\Gamma(M)$.*

PROOF. We can effectively generate all E-substitutions $\varphi : \text{EVar} \to \boldsymbol{\mathcal{E}}$ such that $\varphi(F) = \Box$ for almost all $F \in \text{EVar}$, and all T-substitutions $\psi : \text{TVar} \to \mathcal{T}^{\to}$ such that $\psi(\alpha) = *$ for almost all $\alpha \in \text{TVar}$. We systematically generate all such pairs (φ, ψ), and we stop the procedure if and when we find one which is a solution for $\Gamma(M)$. ∎

7. Relationships with other forms of unification. We use the expression **1U** to denote first-order unification over a single binary function symbol "$\to$", a single constant symbol $*$, and a countably infinite set of ground variables TVar.

PROPOSITION 26. **1U** *is a special case of* $\boldsymbol{\beta}$**U**.

PROOF. Let Δ be an instance of **1U**, i.e.

$$\Delta = \{\sigma_1 = \tau_1, \ \ldots \ , \ \sigma_n = \tau_n\}$$

where $\text{EVar}(\Delta) = \emptyset$ and "$\wedge$" appears nowhere in Δ. With no loss of generality, we require that Δ be well-behaved; only condition 3 in Definition 5 applies to Δ here. Clearly,

1. Δ is also an instance of $\boldsymbol{\beta}$**U**.
2. Every solution ψ of Δ in the sense of **1U** induces a solution (φ, ψ) of Δ in the sense of $\boldsymbol{\beta}$**U**, where $\varphi(F) = \Box$ for every $F \in \text{EVar}$.

The details are straightforward and therefore omitted. ∎

Let **2U** refer to second-order unification, and **SU** to semi-unification. If **2U** and **SU** are over a signature containing at least one function symbol of arity ≥ 2, it is well-known that both **2U** and **SU** are, in general, undecidable problems. To date, there is only one rather complicated proof for the undecidability of **SU**, by a reduction from the *mortality problem* for Turing machines [12]. After the initial proof for the undecidabilty of **2U**, by a reduction from the *diophantine problem* (or *Hilbert's tenth problem*) [8], alternative simpler proofs were discovered [3, 14, 20]. The relationship between **2U** and **SU** has been an open problem for several years. A new, hopefully simpler, proof for the undecidability of **SU** will be a welcome addition to the literature.

OPEN PROBLEM 27. *The relationship between* $\boldsymbol{\beta}$**U** *and* **2U**. *In particular, define a direct reduction from* $\boldsymbol{\beta}$**U** *to* **2U** *and/or vice-versa.*

OPEN PROBLEM 28. *The relationship between* $\boldsymbol{\beta}$**U** *and* **SU**. *In particular, define a direct reduction from* $\boldsymbol{\beta}$**U** *to* **SU** *and/or vice-versa.*

There are other forms of higher-order unification besides **2U** and **SU**, such as "second-order unification with partial substitution" [1] and "unification of stratified second-order terms" [19], for which there may exist reductions from and/or to $\beta\mathbf{U}$. These are all very interesting questions for unification theory, and we leave their investigation for others to pursue.

A. Remaining proofs for Section 4. The following lemma is a technical result we need for the proof of Theorem 22.

LEMMA 29. *Let $X \subseteq \mathrm{TVar}_1$ and $Y \subseteq \mathrm{EVar}_1$. Suppose that either (A) or (B) holds. In both cases, n is a fixed positive integer and $\mathbf{p}$ ranges over $\mathcal{P}^*$.*

(A) Let $\varphi_1, \ldots, \varphi_n : \mathrm{EVar} \to \mathcal{E}$ and $\psi_1, \ldots, \psi_n : \mathrm{TVar} \to \mathcal{T}^{\to}$ be given. Define a new E-substitution φ and a new T-substitution ψ:

$$\varphi(\langle F\rangle_{\mathbf{p}}) = \begin{cases} \varphi_k(\langle F\rangle_{\mathbf{q}}), & \text{if } F \in Y \text{ and } \mathbf{p} = k\mathbf{q} \text{ for some } k \in \{1,\ldots,n\}\ , \\ \Box, & \text{otherwise.} \end{cases}$$

$$\psi(\langle \alpha\rangle_{\mathbf{p}}) = \begin{cases} \psi_k(\langle \alpha\rangle_{\mathbf{q}}), & \text{if } \alpha \in X \text{ and } \mathbf{p} = k\mathbf{q} \text{ for some } k \in \{1,\ldots,n\}\ , \\ \ast, & \text{otherwise.} \end{cases}$$

(B) Let $\varphi : \mathrm{EVar} \to \mathcal{E}$ and $\psi : \mathrm{TVar} \to \mathcal{T}^{\to}$ be given. For every $k \in \{1,\ldots,n\}$, define a new E-substitution φ_k and a new T-substitution ψ_k:

$$\varphi_k(\langle F\rangle_{\mathbf{p}}) = \begin{cases} \varphi(\langle F\rangle_{k\mathbf{p}}), & \text{if } F \in Y\ , \\ \Box, & \text{otherwise.} \end{cases}$$

$$\psi_k(\langle \alpha\rangle_{\mathbf{p}}) = \begin{cases} \psi(\langle \alpha\rangle_{k\mathbf{p}}), & \text{if } \alpha \in X\ , \\ \ast, & \text{otherwise.} \end{cases}$$

Conclusion: For every $k \in \{1,\ldots,n\}$ and every type scheme $\sigma \in \mathcal{T}(X,Y)$, we have $\psi\varphi(\langle\sigma\rangle_k) = \psi_k\varphi_k(\sigma)$.[4]

PROOF. Let k be a fixed integer in $\{1,\ldots,n\}$ throughout the proof. We first define a particular renaming function $\langle\!\langle\ \rangle\!\rangle_k : \mathcal{T}(\mathrm{TVar}) \to \mathcal{T}(\mathrm{TVar})$. By induction on $\mathcal{T}(\mathrm{TVar})$, where $\mathbf{p} \in \mathcal{P}^*$ and $i, j, \ell \in \mathcal{P}$:

1. $\langle\!\langle \alpha^j_{i\mathbf{p}} \rangle\!\rangle_k = \alpha^j_{ik\mathbf{p}}$.
2. $\langle\!\langle \sigma \to \tau \rangle\!\rangle_k = \langle\!\langle \sigma \rangle\!\rangle_k \to \langle\!\langle \tau \rangle\!\rangle_k$.
3. $\langle\!\langle \sigma \wedge \tau \rangle\!\rangle_k = \langle\!\langle \sigma \rangle\!\rangle_k \wedge \langle\!\langle \tau \rangle\!\rangle_k$.

Using either hypothesis (A) or hypothesis (B) in the lemma statement, we first prove

$$\varphi(\langle\sigma\rangle_{k\mathbf{q}}) \ = \ \langle\!\langle \varphi_k(\langle\sigma\rangle_{\mathbf{q}}) \rangle\!\rangle_k \tag{1}$$

for every $\sigma \in \mathcal{T}(X,Y)$ and $\mathbf{q} \in \mathcal{P}^*$. To understand what equation (1) says, first note that what φ does to $\langle F\rangle_{k\mathbf{q}}$ is the same as what φ_k does to $\langle F\rangle_{\mathbf{q}}$. But the variables in $\langle\sigma\rangle_{k\mathbf{q}}$ all have the additional "k" which is not present in $\langle\sigma\rangle_{\mathbf{q}}$, and this is why we need the outer $\langle\!\langle\ \rangle\!\rangle_k$. We prove (1) by induction on σ. We omit the easy case when $\sigma \equiv \alpha^j_i$ (T-variable), $\sigma \equiv \tau_1 \to \tau_2$ ($\to$-type scheme) or $\sigma \equiv \tau_1 \wedge \tau_2$ ($\wedge$-type scheme). For the interested reader,

[4] This lemma is actually two distinct lemmas sharing the same conclusion. The hypothesis of one lemma is (A) and the hypothesis of the other is (B). It is possible to condense these two hypotheses, or even to merge them, but at the price of making the formalism more difficult to decipher. Although unusually long, we prefer to leave the statement of Lemma 29 in its present form.

the details are available in the technical report version of this paper [11]. We only consider here the case when $\sigma \equiv F_i \tau$ in the induction. For this case:

$$\begin{aligned}
\varphi(\langle F_i \tau\rangle_{k\mathbf{q}}) &= \varphi(F_{ik\mathbf{q}}\langle\tau\rangle_{k\mathbf{q}}) \\
&= e[\varphi(\langle\tau\rangle_{k\mathbf{q}1}), \ldots, \varphi(\langle\tau\rangle_{k\mathbf{q}\ell})] \\
&\qquad \text{where } e = \varphi(F_{ik\mathbf{q}}) = \varphi_k(F_{i\mathbf{q}}) \text{ has } \ell \text{ holes} \\
&= e[\langle\!\langle \varphi_k(\langle\tau\rangle_{\mathbf{q}1})\rangle\!\rangle_k, \ldots, \langle\!\langle \varphi_k(\langle\tau\rangle_{\mathbf{q}\ell})\rangle\!\rangle_k] \\
&\qquad \text{by the induction hypothesis} \\
&= \langle\!\langle e[\varphi_k(\langle\tau\rangle_{\mathbf{q}1}), \ldots, \varphi_k(\langle\tau\rangle_{\mathbf{q}\ell})]\rangle\!\rangle_k \\
&= \langle\!\langle \varphi_k(F_{i\mathbf{q}}\langle\tau\rangle_{\mathbf{q}})\rangle\!\rangle_k \\
&= \langle\!\langle \varphi_k(\langle F_i\tau\rangle_{\mathbf{q}})\rangle\!\rangle_k
\end{aligned}$$

This concludes the proof of (1). Let $\widetilde{X} = \{\langle\alpha\rangle_{\mathbf{p}} | \alpha \in X,\ \mathbf{p} \in \boldsymbol{\mathcal{P}}^*\}$. Observe that $\varphi(\langle\sigma\rangle_{k\mathbf{q}}) = \langle\!\langle \varphi_k(\langle\sigma\rangle_{\mathbf{q}})\rangle\!\rangle_k \in \mathcal{T}(\widetilde{X}, \emptyset)$ for every $\sigma \in \mathcal{T}(X, Y)$ and $\mathbf{q} \in \boldsymbol{\mathcal{P}}^*$. We next prove

$$\psi(\langle\!\langle\sigma\rangle\!\rangle_k) \;=\; \psi_k(\sigma) \tag{2}$$

for every $\sigma \in \mathcal{T}(\widetilde{X}, \emptyset)$. The proof of (2) is by induction on σ. We include all the details of this easy induction — there are only 3 cases, because σ does not mention E-variables:

1. $\psi(\langle\!\langle\alpha^j_{i\mathbf{q}}\rangle\!\rangle_k) = \psi(\alpha^j_{ik\mathbf{q}}) = \psi(\langle\alpha^j_i\rangle_{k\mathbf{q}}) = \psi_k(\langle\alpha^j_i\rangle_{\mathbf{q}}) = \psi_k(\alpha^j_{i\mathbf{q}})$

2. $$\begin{aligned}
\psi(\langle\!\langle\sigma \to \tau\rangle\!\rangle_k) &= \psi(\langle\!\langle\sigma\rangle\!\rangle_k \to \langle\!\langle\tau\rangle\!\rangle_k) \\
&= \psi(\langle\!\langle\sigma\rangle\!\rangle_k) \to \psi(\langle\!\langle\tau\rangle\!\rangle_k) \\
&= \psi_k(\sigma) \to \psi_k(\tau) \\
&\qquad \text{by the induction hypothesis} \\
&= \psi_k(\sigma \to \tau)
\end{aligned}$$

3. $$\begin{aligned}
\psi(\langle\!\langle\sigma \wedge \tau\rangle\!\rangle_k) &= \psi(\langle\!\langle\sigma\rangle\!\rangle_k \wedge \langle\!\langle\tau\rangle\!\rangle_k) \\
&= \psi(\langle\!\langle\sigma\rangle\!\rangle_k) \wedge \psi(\langle\!\langle\tau\rangle\!\rangle_k) \\
&= \psi_k(\sigma) \wedge \psi_k(\tau) \\
&\qquad \text{by the induction hypothesis} \\
&= \psi_k(\sigma \wedge \tau)
\end{aligned}$$

This concludes the induction for the proof of (2). By (1) and (2) together, we now have:

$$\psi\varphi(\langle\sigma\rangle_{k\mathbf{q}}) = \psi\langle\!\langle\varphi_k(\langle\sigma\rangle_{\mathbf{q}})\rangle\!\rangle_k = \psi_k\varphi_k(\langle\sigma\rangle_{\mathbf{q}})$$

for every $\sigma \in \mathcal{T}(X, Y)$ and $\mathbf{q} \in \boldsymbol{\mathcal{P}}^*$. In particular, $\psi\varphi(\langle\sigma\rangle_k) = \psi_k\varphi_k(\sigma)$, as desired. ∎

Proof of Theorem 22. For the left to right implication of Theorem 22, we prove a more general fact, namely if there is a derivation $\mathcal{D}$ in $\boldsymbol{\lambda}^{\to,\wedge}$ whose last sequent is $A \vdash M : \sigma$, then there is a pair (φ, ψ) where $\varphi : \mathrm{EVar} \to \boldsymbol{\mathcal{E}}$ and $\psi : \mathrm{TVar} \to \mathcal{T}^{\to}$ such that:

1. $(\varphi, \psi) \models \Gamma(M)$.
2. $A(x) = \psi\varphi(\rho(M, x))$ for every λ-variable x occurring free in M.
3. $\sigma = \psi\varphi(\tau(M))$.

The proof is by induction on derivations $\mathcal{D}$ in $\boldsymbol{\lambda}^{\to,\wedge}$. The *induction hypothesis for* $\mathcal{D}$, denoted IH($\mathcal{D}$), asserts the existence of (φ, ψ) satisfying the preceding 3 points.

For the base case, the derivation $\mathcal{D}$ consists of a single sequent, $x^{(j)} : \sigma \vdash x^{(j)} : \sigma$ where $\sigma \in \mathcal{T}^{\rightarrow}$. Define $\varphi(F) = \square$ for every $F \in$ EVar, $\psi(\alpha^j_x) = \sigma$ and $\psi(\alpha) = \ast$ for every $\alpha \in$ TVar $- \{\alpha^j_x\}$. For this case, $\Gamma(x^{(j)}) = \emptyset$, and $\tau(x^{(j)}) = \rho(x^{(j)}, x) = \alpha^j_x$. It immediately follows that (φ, ψ) satisfies IH($\mathcal{D}$).

Proceeding inductively, suppose there is a derivation $\mathcal{D}_0$ whose last sequent is $A, y : \sigma_1 \vdash N : \sigma_2$ and a derivation $\mathcal{D}$ obtained from $\mathcal{D}_0$ according to:

$$\frac{A, y : \sigma_1 \;\vdash\; N : \sigma_2}{A \;\vdash\; (\lambda y.N) : \sigma_1 \rightarrow \sigma_2}$$

where $\sigma_1 \in \mathcal{T}$ and $\sigma_2 \in \mathcal{T}^{\rightarrow}$. Here, y occurs $m \geq 1$ times free in N. We assume there is (φ_0, ψ_0) satisfying the 3 parts of IH($\mathcal{D}_0$). If we define $(\varphi, \psi) = (\varphi_0, \psi_0)$, it is easy to check that (φ, ψ) satisfies the 3 parts of IH($\mathcal{D}$).

Suppose there is a derivation $\mathcal{D}_0$ whose last sequent is $A \vdash N : \sigma_2$ and a derivation $\mathcal{D}$ obtained from $\mathcal{D}_0$ according to:

$$\frac{A \;\vdash\; N : \sigma_2}{A \;\vdash\; (\lambda y.N) : \sigma_1 \rightarrow \sigma_2}$$

where $\sigma_1, \sigma_2 \in \mathcal{T}^{\rightarrow}$. Here, y occurs $m = 0$ times free in N. Let (φ_0, ψ_0) satisfy the 3 parts of IH($\mathcal{D}_0$). We define $\varphi = \varphi_0$ and

$$\psi(\alpha) = \begin{cases} \sigma_1, & \text{if } \alpha = \alpha_y \;, \\ \psi_0(\alpha), & \text{if } \alpha \neq \alpha_y \;. \end{cases}$$

It is again easy to check that (φ, ψ) satisfies the 3 parts of IH($\mathcal{D}$).

For the last case of the induction, suppose there is a derivation $\mathcal{D}$ which ends with

$$\frac{A_0 \;\vdash\; N : (e[\sigma_1, \ldots, \sigma_n] \rightarrow \sigma_{n+1}) \qquad A_1 \;\vdash\; P : \sigma_1 \;\cdots\; A_n \;\vdash\; P : \sigma_n}{A_0 \wedge e[A_1, \ldots, A_n] \;\vdash\; (NP) : \sigma_{n+1}}$$

The subderivation corresponding to N is $\mathcal{D}_0$, and the $n \geq 1$ subderivations corresponding to P are $\mathcal{D}_1, \ldots, \mathcal{D}_n$ from left to right. We assume there is (φ_0, ψ_0) satisfying IH($\mathcal{D}_0$) and (φ_k, ψ_k) satisfying IH($\mathcal{D}_k$) for $k = 1, \ldots, n$. We have

$$\Gamma(NP) = \Gamma(N) \;\cup\; F\Gamma(P) \;\cup\; \{\tau(N) = F\tau(P) \rightarrow \beta\}$$

for some fresh $F \in \mathrm{EVar}_1$ and fresh $\beta \in \mathrm{TVar}_{aux}$. For every $G \in \mathrm{EVar}_1$ and $\mathbf{p} \in \mathcal{P}^*$, define:

$$\varphi(\langle G\rangle_{\mathbf{p}}) = \begin{cases} e, & \text{if } G = F \text{ and } \mathbf{p} = \varepsilon \;, \\ \varphi_0(\langle G\rangle_{\mathbf{p}}), & \text{if } G \in \mathrm{EVar}(\Gamma(N)) \;, \\ \varphi_k(\langle G\rangle_{\mathbf{q}}), & \text{if } G \in \mathrm{EVar}(\Gamma(P)) \\ & \text{and } \mathbf{p} = k\mathbf{q} \text{ for some } k \in \{1, \ldots, n\} \;, \\ \square, & \text{otherwise.} \end{cases}$$

The function φ is well-defined because $\mathrm{EVar}(\Gamma(N)) \cap \mathrm{EVar}(\Gamma(P)) = \emptyset$. For every $\alpha \in \mathrm{TVar}_1$ and $\mathbf{p} \in \mathcal{P}^*$, define:

$$\psi(\langle \alpha\rangle_{\mathbf{p}}) = \begin{cases} \sigma_{n+1}, & \text{if } \alpha = \beta \text{ and } \mathbf{p} = \varepsilon \;, \\ \psi_0(\langle \alpha\rangle_{\mathbf{p}}), & \text{if } \alpha \in \mathrm{TVar}(\Gamma(N)) \;, \\ \psi_k(\langle \alpha\rangle_{\mathbf{q}}), & \text{if } \alpha \in \mathrm{TVar}(\Gamma(P)) \\ & \text{and } \mathbf{p} = k\mathbf{q} \text{ for some } k \in \{1, \ldots, n\} \;, \\ \ast, & \text{otherwise.} \end{cases}$$

The function ψ is well-defined because $\mathrm{TVar}(\Gamma(N)) \cap \mathrm{TVar}(\Gamma(P)) = \emptyset$. It remains to check that (φ, ψ) satisfies the 3 parts of $\mathrm{IH}(\mathcal{D})$. We omit parts 1 and 3 of $\mathrm{IH}(\mathcal{D})$, which are simpler to prove, and consider part 2 only. Posing $A = A_0 \wedge e[A_1, \ldots, A_n]$, we have to show that $A(x) = \psi\varphi(\rho(NP, x))$ for every λ-variable x occurring free in (NP). With no loss of generality, assume x occurs free in both N and P, so that $\rho(NP, x) = (\rho(N, x) \wedge F\rho(NP, x))$. By $\mathrm{IH}(\mathcal{D}_0)$, we have $A_0(x) = \psi_0\varphi_0(\rho(N, x))$. By $\mathrm{IH}(\mathcal{D}_k)$, we have $A_k(x) = \psi_k\varphi_k(\rho(P, x))$ for every $k \in \{1, \ldots, n\}$. Hence,

$$\begin{aligned}
A(x) &= \psi_0\varphi_0(\rho(N,x)) \wedge e[\psi_1\varphi_1(\rho(P,x)), \ldots, \psi_n\varphi_n(\rho(P,x))] \\
&= \psi\varphi(\rho(N,x)) \wedge e[\psi\varphi(\langle\rho(P,x)\rangle_1), \ldots, \psi\varphi(\langle\rho(P,x)\rangle_n)] \\
&\qquad \text{by Lemma 29 with hypothesis (1)} \\
&= \psi(\varphi(\rho(N,x)) \wedge e[\varphi(\langle\rho(P,x)\rangle_1), \ldots, \varphi(\langle\rho(P,x)\rangle_n)]) \\
&= \psi(\varphi(\rho(N,x)) \wedge \varphi(F\rho(P,x))) \\
&\qquad \text{because } \varphi(F) = e \\
&= \psi\varphi(\rho(N,x) \wedge F\rho(P,x))
\end{aligned}$$

This concludes the induction for the proof of the left to right implication of the theorem.

For the right to left implication of Theorem 22, we prove if there is a solution (φ, ψ) for $\Gamma(M)$, then there is a derivation in $\boldsymbol{\lambda}^{\rightarrow,\wedge}$ whose last sequent is $A \vdash M : \sigma$ such that:

1. $A(x) = \psi\varphi(\rho(M, x))$ for every λ-variable x occurring free in M.
2. $\sigma = \psi\varphi(\tau(M))$.

The proof is by induction on M. The *induction hypothesis for* M, denoted $\widetilde{\mathrm{IH}}(M)$, asserts that if $(\varphi, \psi) \models \Gamma(M)$ then there is a derivation whose last sequent is $A \vdash M : \sigma$ such that the preceding 2 points hold.

For the base case, $M \equiv x^{(j)}$, $\Gamma(M) = \emptyset$, and $\tau(M) = \rho(M, x) = \alpha_x^j$. Suppose $(\varphi, \psi) \models \Gamma(M)$. The desired derivation consists of a single sequent $x^{(j)} : \sigma \vdash x^{(j)} : \sigma$ where $\sigma = \psi\varphi(\alpha_x^j) = \psi(\alpha_x^j)$. It is clear that $\widetilde{\mathrm{IH}}(x^{(j)})$ is true.

Proceeding inductively, $M \equiv (\lambda y.N)$ where y occurs $m \geq 0$ times free in N, and there is $(\varphi, \psi) \models \Gamma(M)$. In this case, $\Gamma(M) = \Gamma(N)$, so that $(\varphi, \psi) \models \Gamma(N)$. By $\widetilde{\mathrm{IH}}(N)$, there is a derivation whose last sequent is:

- $A, y : \sigma_1 \vdash N : \sigma_2$ where $\sigma_1 = \psi\varphi(\rho(N, y))$ and $\sigma_2 = \psi\varphi(\tau(N))$, if $m \geq 1$.
- $A \vdash N : \sigma_2$ where $\sigma_2 = \psi\varphi(\tau(N))$, if $m = 0$.

We can form a new derivation whose last sequent is $A \vdash (\lambda y.N) : \sigma_1 \rightarrow \sigma_2$, where:

- $\psi\varphi(\tau(M)) = \psi\varphi(\rho(N, y)) \rightarrow \psi\varphi(\tau(N)) = \sigma_1 \rightarrow \sigma_2$, if $m \geq 1$.
- $\psi\varphi(\tau(M)) = \psi\varphi(\alpha_y \rightarrow \tau(N)) = \psi\varphi(\alpha_y) \rightarrow \psi\varphi(\tau(N)) = \sigma_1 \rightarrow \sigma_2$, if $m \geq 0$, where we pose $\sigma_1 = \psi\varphi(\alpha_y)$.

Moreover, $\psi\varphi(\rho(M, x)) = \psi\varphi(\rho(N, x))$ if $x \not\equiv y$. We have thus verified that $\widetilde{\mathrm{IH}}(M)$ is true for the case $M \equiv (\lambda y.N)$.

For the last case of the induction, suppose $M \equiv (NP)$ and there is $(\varphi, \psi) \models \Gamma(M)$. In this case:

$$\begin{array}{ll}
(*) & (\varphi, \psi) \models \Gamma(N) \\
(**) & (\varphi, \psi) \models F\Gamma(P) \\
(***) & (\varphi, \psi) \models \{\tau(N) = F\tau(P) \rightarrow \beta\}
\end{array}$$

for some fresh $F \in \mathrm{EVar}_1$ and $\beta \in \mathrm{TVar}_{aux}$. Suppose $\varphi(F) = e$ and c is a type context with $n \geq 1$ holes. For every $G \in \mathrm{EVar}_1$, $\mathbf{p} \in \mathcal{P}^*$ and $k \in \{1, \ldots, n\}$, define:

$$\varphi_k(\langle G \rangle_{\mathbf{p}}) = \begin{cases} \varphi(\langle G \rangle_{k\mathbf{p}}), & \text{if } G \in \mathrm{EVar}(\Gamma(P)) \ , \\ \Box, & \text{otherwise.} \end{cases}$$

For every $\alpha \in \mathrm{TVar}_1$, $\mathbf{p} \in \mathcal{P}^*$ and $k \in \{1, \ldots, n\}$, define:

$$\psi_k(\langle \alpha \rangle_{\mathbf{p}}) = \begin{cases} \psi(\langle \alpha \rangle_{k\mathbf{p}}), & \text{if } \alpha \in \mathrm{TVar}(\Gamma(P)) \ , \\ \ast, & \text{otherwise.} \end{cases}$$

For an arbitrary $\sigma \in \mathcal{T}(X, Y)$ where $X = \mathrm{TVar}(\Gamma(P))$ and $Y = \mathrm{EVar}(\Gamma(P))$, we have

$$\begin{aligned} \psi\varphi(F\sigma) &= \psi(e[\varphi(\langle \sigma \rangle_1), \ldots, \varphi(\langle \sigma \rangle_n)]) \\ &= e[\psi\varphi(\langle \sigma \rangle_1), \ldots, \psi\varphi(\langle \sigma \rangle_n)] \\ &= e[\psi_1\varphi_1(\sigma), \ldots, \psi_n\varphi_n(\sigma)] \\ &\qquad \text{by Lemma 29 with hypothesis (2)} \end{aligned}$$

Hence, by $(**)$, we have that $(\varphi_k, \psi_k) \models \Gamma(P)$ for every $k \in \{1, \ldots, n\}$. Hence, by $\widetilde{\mathrm{IH}}(P)$, for every $k \in \{1, \ldots, n\}$ there is a derivation $\mathcal{D}_k$ whose last sequent is $A_k \vdash P : \sigma_k$ where $\sigma_k = \psi_k\varphi_k(\tau(P))$ and $A_k(x) = \psi_k\varphi_k(\rho(P, x))$ for every λ-variable x occurring free in P. Now, for every $G \in \mathrm{EVar}_1$ and $\mathbf{p} \in \mathcal{P}^*$, define:

$$\varphi_0(\langle G \rangle_{\mathbf{p}}) = \begin{cases} \varphi(\langle G \rangle_{\mathbf{p}}), & \text{if } G \in \mathrm{EVar}(\Gamma(N)) \ , \\ \Box, & \text{otherwise.} \end{cases}$$

For every $\alpha \in \mathrm{TVar}_1$ and $\mathbf{p} \in \mathcal{P}^*$, define:

$$\psi_0(\langle \alpha \rangle_{\mathbf{p}}) = \begin{cases} \psi(\langle \alpha \rangle_{\mathbf{p}}), & \text{if } \alpha \in \mathrm{TVar}(\Gamma(N)) \ , \\ \ast, & \text{otherwise.} \end{cases}$$

Hence, by $(*)$, we have that $(\varphi_0, \psi_0) \models \Gamma(N)$. Hence, by $\widetilde{\mathrm{IH}}(N)$, there is a derivation $\mathcal{D}_0$ whose last sequent is $A_0 \vdash N : \sigma_0$ where $\sigma_0 = \psi_0\varphi_0(\tau(N))$ and $A_0(x) = \psi_0\varphi_0(\rho(N, x))$ for every λ-variable x occurring free in N. We also have

$$\begin{aligned} \sigma_0 = \psi_0\varphi_0(\tau(N)) &= \psi\varphi(\tau(N)) \\ &= \psi\varphi(F\tau(P) \to \beta) \\ &\qquad \text{by } (***) \\ &= \psi\varphi(F\tau(P)) \to \psi\varphi(\beta) \\ &= e[\psi\varphi(\langle \tau(P) \rangle_1), \ldots, \psi\varphi(\langle \tau(P) \rangle_n)] \to \psi\varphi(\beta) \\ &\qquad \text{because } \varphi(F) = e \\ &= e[\psi_1\varphi_1(\tau(P)), \ldots, \psi_n\varphi_n(\tau(P))] \to \psi\varphi(\beta) \\ &\qquad \text{by Lemma 29 with (2)} \\ &= e[\sigma_1, \ldots, \sigma_n] \to \sigma_{n+1} \end{aligned}$$

where we pose $\sigma_{n+1} = \psi\varphi(\beta)$.

Putting all the pieces together, concerning the derivations $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_n$, we can construct a derivation $\mathcal{D}$ ending with

$$\frac{A_0 \ \vdash \ N : (e[\sigma_1, \ldots, \sigma_n] \to \sigma_{n+1}) \qquad A_1 \ \vdash \ P : \sigma_1 \ \cdots \ A_n \ \vdash \ P : \sigma_n}{A_0 \wedge e[A_1 \wedge \cdots \wedge A_n] \ \vdash \ (NP) : \sigma_{n+1}}$$

such that the 2 parts of $\widetilde{\mathrm{IH}}(NP)$ hold. This concludes the induction for the right to left implication of the theorem. ∎

References

[1] S. R. Bass, "The undecidability of k-provability", *Annals of Pure and Applied Logic*, Vol 53, pp 75-102, 1991.

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

[3] W. M. Farmer, "Simple second-order languages for which unification is undecidable", *Theoretical Computer Science*, Vol 87, pp 25-41, 1991.

[4] F. Cardone and M. Coppo, "Two extensions of Curry's type inference system", in *Logic and Computer Science*, ed. P. Odifreddi, APIC series no. 31, pp 19-75, Academic Press, 1990.

[5] M. Coppo and M. Dezani-Ciancaglini, "An extension of basic functionality theory for λ-calculus", *Notre Dame Journal of Formal Logic*, Vol 21, no. 4, pp 685-693, 1980.

[6] M. Coppo and P. Giannini, "A complete type inference algorithm for simple intersection types", in *Proceedings of CAAP '92, 17th Colloquium on Trees in Algebras and Programming*, ed. J.-C. Raoult, Rennes, France. LNCS Vol 581, pp 102-123, Springer-Verlag, 1992.

[7] M. Coppo and P. Giannini, "Principal Types and Unification For a Simple Intersection Type System", *Information and Computation*, Vol 122, pp 70-96, 1995.

[8] W. D. Goldfarb, "The undecidability of the second-order unification problem". *Theoretical Computer Science*, Vol 13, pp 225-230, 1981.

[9] J. R. Hindley, *Basic Simple Type Theory*, Cambridge Tracts in Theoretical Computer Science 42, Cambridge University Press, 1997.

[10] A. J. Kfoury, "A linearization of the λ-calculus and an application". Submitted for publication.

[11] A. J. Kfoury, "Beta-reduction as unification". Technical report, Boston University, 96-019, June 1997. Also available at URL: http://www.cs.bu.edu/faculty/kfoury/research.html

[12] A. J. Kfoury, J. Tiuryn and P. Urzyczyn, "The undecidability of the semi-unification problem". *Information and Computation*, Vol 102, no. 1, pp 83-101, 1993.

[13] A. J. Kfoury and J. B. Wells, "New Notions of Reduction and Non-Semantic Proofs of Beta Strong Normalization in Typed Lambda Calculi", Proceedings of *Logic in Computer Science*, June 1995.

[14] J. Krajíček and P. Pudlák, "The number of proof lines and the size of proofs in first order logic". *Archive for Mathematical Logic*, Vol 27, pp 69-84, 1988.

[15] D. Leivant, "Polymorphic Type Inference", Proceedings of *Tenth Annual ACM Symposium on Principles of Programming Languages*, pp 88-98, 1983.

[16] J. Levy, "Linear second-order unification". In Proceedings of *RTA*, July 1996. Also available at URL: http://www-lsi.upc.es/˜jlevy

[17] J. C. Mitchell, *Foundations for Programming Languages*, MIT Press, 1996.

[18] G. Pottinger, "A Type Assignment for the Strongly Normalizable λ-terms", in *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, J.P. Seldin and J.R. Hindley, eds., pp 561-577, Academic Press, 1980.

[19] M. Schmidt-Schauss, "Unification of stratified second-order terms". Technical Report, Fachbereich Informatik, Johann Wolfgang-Goethe-Universität Frankfurt, 1996. Author's e-mail: schauss@informatik.uni-frankfurt.de

[20] A. Schubert, "Second-order unification and type inference for Church-style polymorphism". Technical Report, Institute of Informatics, Warsaw University, January 1997. Available from FTP site: zls.mimuw.edu.pl in files: pub/alx/simple.dvi.tar.gz and pub/alx/simple.ps.gz. Shorter version in Proceedings of *26th Annual ACM Symposium on Principles of Programming Languages*, 1998.

[21] C. Retoré, "A Note on Intersection Types", INRIA report RR-2431, 1995, postscript available at: ftp://ftp.inria.fr/INRIA/publication/publi-ps-gz/RR/RR-2431.ps.gz

[22] S. van Bakel, "Complete restrictions of the intersection type discipline". *Theo. Comp. Sc.*, Vol 102, pp 135-163, 1992.

[23] S. van Bakel, *Intersection Type Disciplines in Lambda Calculus and Applicative Term Rewriting Systems.* Doctoral dissertation, Catholic University of Nijmegen, also issued by the Mathematisch Centrum, Amsterdam, 1993.