Dinatural terms in system F

Joachim de Lataillade
Institut de Mathématiques de Luminy
Campus de Luminy, Case 907
13288 MARSEILLE Cedex 9
delatail@iml.univ-mrs.fr

Abstract

We provide in this article two characterisation results, describing exactly which terms verify the dinaturality diagram, in Church-style system F and in Curry-style system F.

The proof techniques we use here are purely syntactic, giving in particular a direct construction of the two terms generated by the dinaturality diagram. But the origin of these techniques lies in fact directly on the analysis of system F through game semantics.

Thus, this article provides an example of backward engineering, where powerful syntactic results can be extracted from a semantic analysis.

1 Introduction

Dinaturality. In this article we solve an open problem raised in [10]: the analysis of dinaturality for second-order lambda-calculus.

Dinaturality is a categorical notion that has appeared in logic since [2]. The idea behind this is to consider a type A with a free variables X as a functor, which associates to each type B the type $A[B/X]$. This raises a problem when we apply the functor to a morphism: indeed, if we consider for example $A = X \to X$, it is well-known that the first occurrence of X has to be contravariant, whereas the second one is covariant. Thus, the functors we need have the form

$$F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$$

where $\mathbf{C}$ is the category to be considered (in this article $\mathbf{C}$ will be a syntactic category).

What is then the good notion of morphism between two such functors $F, G : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$? A reasonable answer to this question is the notion of **dinatural transformation** (see [15]) that extends the natural transformations to this new kind of functors. A dinatural transformation θ between F and G, denoted $\theta : F \ddot{\to} G$, is a collection of morphisms $\theta_A : FAA \to GAA$, for A object in $\mathbf{C}$, such that the following diagram commutes:

$$\begin{tikzcd}
 & F[B,B] \arrow[r, "\theta_B"] & G[B,B] \arrow[dr, "{G[B,\mathfrak{u}]}"] & \\
F[C,B] \arrow[ur, "{F[\mathfrak{u},B]}"] \arrow[dr, "{F[C,\mathfrak{u}]}"'] & & & G[B,C] \\
 & F[C,C] \arrow[r, "\theta_C"'] & G[C,C] \arrow[ur, "{G[\mathfrak{u},C]}"'] &
\end{tikzcd}$$

for every morphism $\mathfrak{u} : B \to C$ in $\mathbf{C}$.

Unfortunately, dinatural transformations do not compose in general, so we cannot build a category with this structure. However, there are some special cases where composition works properly: it was the case in [10] and [5], and again in this article we will not get any trouble regarding composition.

Appearances of dinaturality in computer science include in particular the formalisation of parametric polymorphism (see below) and the theory of traced monoidal categories [13], notably their relations with fixpoint operators [11].

Characterisation of dinatural terms. Rather than composition, the problem we are interested in is to check whether, in a given syntactic category, every term t actually defines a dinatural transformation, when considering the family $(t[A/X])_A$ (where A describes the set of all types).

It was shown in [10] that this is true for simply-typed lambda-calculus. This means that every term in this calculus satisfies a general and non-trivial property, stated by the dinaturality diagram. This is the origin of the famous *theorems for free* of Wadler [26].

But the authors of [10] insisted that they did not solve the problem in the more general case where there is a second-order quantification in the grammar of types. The corresponding calculus, called system F, was discovered by Girard [8] and Reynolds [22]. There are two main presentations of system F, that we will both study in this article: Church-style system F, where types appear explicitly in the terms, and Curry-style system F where, on the contrary, the terms contain no type indications, and the language of terms is the pure lambda-calculus. In this article, Church-style sequents are noted $\Gamma \vdash t : A$, whereas Curry-style sequent are noted $\Gamma \vdash_{\mathrm{Cu}} \tau : A$.

It is in fact easy to realise that the dinaturality property is not satisfied any more with system F: as we shall see (and as can be seen directly by checking the diagram), in Church-style system F the term $t = \lambda x^{\forall Y.(Y \to Y)} \lambda y^X.(x)\{X\}y$ is not dinatural in X; that is, the family $(t[A/X])_A$ is not a dinatural transformation. For Curry-style system F, the question is more tricky, but again the dinaturality will not be true in general.

In the present article we provide two **characterisation results**: the first is for Church-style system F, for which we show that a term t is dinatural in X if its normal form contains no type instantiation where X appears as a free variable. The second result is for Curry-style system F: we show that a term τ such that $\Gamma \vdash_{\mathrm{Cu}} \tau : A$ is dinatural in X if and only if there exists a Church-style term t, equal to τ modulo type erasure and such that $\Gamma \vdash t : A$, which contains no type instantiation where X appears as a free variable. Both characterisations are decidable.

Proof techniques. To prove these results, we use purely syntactic techniques. For the Church-style case, we give a construction that we call **simple expansion**, defined only on a certain subset of terms, and we prove that, for these terms, the two terms produced by the dinaturality diagram happen to be both equal to the result of the simple expansion; thus, these terms are dinatural. For the other terms, a very simple argument shows that they are not dinatural.

Curry-style system F is defined here as the Church-style system F modulo type erasure. And to deal with dinaturality in this system, we will need to completely describe the result of the dinaturality diagram for Church-style terms. Hence, we introduce an **upper expansion** and a **lower expansion**, and we prove that they correspond respectively to the upper term (or morphism) and the lower term in the dinaturality diagram. From this we extract a proof of characterisation of dinatural Curry-style terms.

Thus, the article will not only give us these two characterisation results, but also provide a complete description of the terms given by the dinaturality diagram.

But if the techniques used in this article are purely syntactic, their origin lie in fact on a semantic ground: namely, on **game semantics**. Indeed, in games, the above expansions are related to a process that is already well-known by those interested in games for system F: the *copycat expansion*, appeared implicitly in [12] and more explicitly in [19] and [4]. It was proved in [3] that by carefully generalising the copycat expansion process, one can exactly describe the dinaturality diagram in game semantics; in fact, one obtains the processes corresponding in games to the upper and lower expansions described in this article (and simple expansion is just a special case of these two expansions).

Thus, what we do in this article can be considered as a form of **backward engineering**: from a notion already understood in game semantics, we extract a purely syntactic approach of the problem, that we use to prove our characterisation results. Note that the characterisations might be proved directly in game semantics, using full completeness results. But for the sake of simplicity, it is easier to use a syntactic translation of the game mechanisms.

A similar approach was taken in [14], where a purely syntactic study of exceptions and continuations was proposed, arising from a game semantics analysis. Other, more direct applications of game semantics for dealing with syntactic problems concern model-checking [20] and type isomorphisms [16, 4].

Connections with parametricity. One interesting feature of dinaturality, already noted in [2], is its connections with parametric polymorphism.

In a polymorphic language (or in a model of polymorphism), a function f of type $\forall X.A$ is called **parametric** if its behaviour does not depend, intuitively, on the value of the argument given for X. This notion was introduced by Strachey [25], and analysed by Reynolds with his notion of relational parametricity [23]. Relational parametricity was first a semantic idea, but it was later translated syntactically [1, 21] trough a language refining Church-style system F.

Dinaturality was proposed in [2] as an alternative criterion for parametricity, and the property was analysed in Reynolds' PER model. Then in [21] it was shown that relational parametricity implies dinaturality at a syntactic level. Starting from the fact that Church-style system F itself does not satisfy the dinaturality property, it was argued in [3] that another condition, restricting dinaturality to simple types (i.e. types without second-order quantification), was a more sensible criterion for parametricity.

However, as explained in [3], this last criterion is still a bit too strong for some languages or models that should be considered as parametric. Thus, the present work might be useful in the tedious quest for a perfect parametricity criterion.

2 Church-style system F

In this article we will focus mainly on Church-style system F, the variant of system F where explicit types appear in the terms. In fact, even Curry-style system F will be studied through an analysis of the Church-style case.

The grammar and rules of this system are shown on Figure 1, where *FTV*(Γ) is the set of free type variables in all types appearing in Γ, *FTV*(t) the set of all free type variables in all types appearing in t and *FV*(t) the set of all free term variables in t. For example, for the term $t = \Lambda Y \lambda x^{X \to Y}.(z)\{Z \to Y\}$ we have *FTV*$(t) = \{X, Z\}$ and *FV*$(t) = \{z\}$. The equality in this system, denoted $=_F$, is the congruence relation resulting from the equalities given at the end of Figure 1.

The set of all second-order types is called $\mathtt{Types}$. Note that we do not consider the product type $A \times B$ in the grammar of system F: this is only for the sake of simplicity, to avoid making our proofs more technical. In fact we could do all the proofs of the article, and obtain the same results, in a case where we have a product type. As this construction is not an essential part of system F, we chose to ignore it here. Finally, for a first reading it may be useful to forget about the type indications in the lambda-abstractions λx^A: ignoring them will make many further constructions much simpler.

We introduce two specific notations on system F that will be of great use throughout the article:

- introducing a new atomic type $\perp$ (which will be used only in this definition), we call $\mathcal{V}(A)$ the type variable appearing on top of the type A if this variable is free, $\mathcal{V}(A) = \perp$ otherwise; this means, inductively: $\mathcal{V}(X) = X$, $\mathcal{V}(\perp) = \perp$, $\mathcal{V}(B \to A) = \mathcal{V}(A)$ and $\mathcal{V}(\forall X.A) = \mathcal{V}(A[\perp/X])$

- given a term t, we note $\mathcal{Z}(t)$ the set of all free type variables in all type instantiations $\{D\}$ appearing in t. Note that $\mathcal{Z}(t) \subseteq$ *FTV*(t) but they are not equal in general: indeed, X might appear in t as a lambda-abstraction even if $X \notin \mathcal{Z}(t)$. For example, if $t = \lambda x^X.\lambda y^{\forall Y.Y}.(y)\{Z\}$, then *FTV*$(t) = \{X, Z\}$ whereas $\mathcal{Z}(t) = \{Z\}$.

2.1 Normal forms of system F

Throughout the article, we will consider the **normal forms** of the Church-style system F; by this, we mean η-long, β-normal forms:

- a β-normal form is a term t that contains no redex of the form $(\lambda x^A.u)v$ or $(\Lambda X.u)\{B\}$

- an η-long β-normal form is a β-normal form t with $\Gamma \vdash t : A$ such that, if we apply an η-expansion $u \mapsto \lambda x^B.(u)x$ or an $\eta 2$-expansion $u \mapsto \Lambda X.(u)\{X\}$ anywhere inside t, then for the resulting term t', we have either $\Gamma \nvdash t' : A$ or t' is not β-normal.

It is well-known since [8] that any Church-style term t_0 such that $\Gamma \vdash t_0 : A$ has a unique normal form $t = \mathrm{NF}(t_0)$, which can be written

$$t = \alpha_1 \dots \alpha_P.\, (z)\, T_1 \dots T_n \qquad (1)$$

where:

- each α_i is either of the form $\lambda x_j^{A_j}$ $(1 \leq j \leq p)$ or ΛX_k $(1 \leq k \leq P - p)$
- there exists $B \in \mathtt{Types}$ such that $z : B \in \Gamma \cup \{x_1 : A_1, \dots, x_p : A_p\}$
- if $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash (z)\, T_1 \dots T_i : \forall Y.B_0$ and $i < n$ then T_{i+1} is of the form $\{D\}$ with $D \in \mathtt{Types}$
- if $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash (z)\, T_1 \dots T_i : B_1 \to B_2$ and $i < n$ then T_{i+1} is a Church-style term in normal form such that $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash T_{i+1} : B_1$
- if $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash (z)\, T_1 \dots T_n : B_0$ then B_0 is a type variable.

We will define all our operations on Church-style terms inductively, using this presentation of normal forms. We introduce a few more notations: given a normal form t as in (1), we set $\mathcal{V}_\Gamma(t) = Z$ if $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash z : D$ and $\mathcal{V}(D) = Z$, and we note $z \in \Gamma$ if there exists $B \in \mathtt{Types}$ such that $z : B \in \Gamma$.

2.2 Syntactic dinaturality

In this article we are concerned with dinaturality at the level of the syntax - only Church-style system F for the moment. In this case, the category of interest, called the **syntactic category**, is denoted $\mathcal{T}$ and defined as follows (for a given term variable x):

- the objects are second-order types

- a morphism from A to B is an equivalence class, modulo $=_F$, of terms t such that $x : A \vdash t : B$

- composition is given by substitution: if $x : A \vdash t : B$ and $x : B \vdash u : C$ then $x : A \vdash u[t/x] : C$, so composing the class of u with the class of t gives the class of $u[t/x]$, denoted $t\ ;\ u$

- the identity is the class of the term x.

In this category, given a type variable X, one can associate to every type A a functor $A[(\_,\_)/X] : \mathcal{T}^{op} \times \mathcal{T} \to \mathcal{T}$.

This functor is constructed as follows. For two types B and C, $A[(B, C)/X]$ (or $A[B, C]$ for short) is obtained from A by replacing every positive occurrence of X by C and every negative occurrence of X by B. Inductively, this means:

- $X[B, C] = C$ and $Y[B, C] = Y$ if $Y \neq X$

- $(A_1 \to A_2)[B, C] = A_1[C, B] \to A_2[B, C]$

Grammars:

$$
\begin{array}{lcl}
A & ::= & X \mid A \to A \mid \forall X.A \\
t & ::= & x \mid \lambda x^A.t \mid (t)t \mid \Lambda X.t \mid (t)\{A\}
\end{array}
$$

Typing rules:

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\ (\text{ax})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A.t : A \to B}\ (\to I)$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash (t)u : B}\ (\to E)$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \Lambda X.t : \forall X.A}\ (\forall I)\ \text{ if } X \notin \mathit{FTV}(\Gamma)$$

$$\frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash (t)\{B\} : A[B/X]}\ (\forall E)$$

Equalities:

$$
\begin{array}{rccl}
(\lambda x^A.t)u & =_\beta & t[u/x] & \\
\lambda x^A.(t)x & =_\eta & t & \text{if } x \notin \mathit{FV}(t) \\
(\Lambda X.t)\{B\} & =_{\beta 2} & t[B/X] & \\
\Lambda X.(t)\{X\} & =_{\eta 2} & t & \text{if } X \notin \mathit{FTV}(t)
\end{array}
$$

Figure 1. Church-style system F

- $(\forall Y.A_0)[B, C] = \forall Y.A_0[B, C]$ with $Y \neq X$.

For two terms u, v with $x : A_1 \vdash u : B_1$ and $x : A_2 \vdash v : B_2$, we first define $f \to g$ for $f = \lambda x^{A_1}.u$ and $g = \lambda x^{A_2}.v$ as:

$$f \to g = \lambda x^{B_1 \to A_2} \lambda y^{A_1}.(g)(x)(f)y$$

so that: $\vdash f \to g : (B_1 \to A_2) \to (A_1 \to B_2)$. Then we can define more generally $A[(f, g)/X]$ (or $A[f, g]$ for short) as follows:

- $X[f, g] = g$ and $Y[f, g] = id_Y$ if $Y \neq X$ (with $id_Y = \lambda x^Y.x$)
- $(A_1 \to A_2)[f, g] = A_1[g, f] \to A_2[f, g]$
- $(\forall Y.A_0)[f, g] = \lambda x^{\forall Y.A_0[B_1, A_2]}.\Lambda Y.(A_0[f, g])(x)\{Y\}$ with $Y \neq X$.

Finally, we set $A[(u, v)/X] = (A[(f, g)/X])x$, so that

$$x : A[(B_1, A_2)/X] \vdash\ A[(u, v)/X]\ : A[(A_1, B_2)/X]$$

Thus we obtain two different definitions for $A[(\_, \_)/X]$, but depending on the context it will be clear in what follows whether we use $A[(f, g)/X]$ or $A[(u, v)/X]$. One can also write $A[(f, B)/X]$ or $A[(B, f)/X]$ (resp. $A[(u, B)/X]$ or $A[(B, u)/X]$): it is just a shorthand for $A[(f, id_B)/X]$ or $A[(id_B, f)/X]$ (resp. $A[(u, x)/X]$ or $A[(x, u)/X]$).

As the variable X will always be fixed in what follows, we sometimes use the notation $A[\gamma, \delta]$, or even $A\gamma\delta$, for $A[(\gamma, \delta)/X]$ (whatever γ and δ might be).

Now it makes sense to talk about dinaturality in this category:

Definition 1 (dinatural term, Church-style) *A term t such that $x : A_1 \vdash t : A_2$ is* ***dinatural in X*** *if the family $(t[A/X])_{A \in \mathit{Types}}$ is a dinatural transformation; that is, if the diagram*

$$
\begin{tikzcd}
 & A_1[B, B] \arrow[r, "{t[B/X]}"] & A_2[B, B] \arrow[dr, "{A_2[B,u]}"] & \\
A_1[C, B] \arrow[ur, "{A_1[u,B]}"] \arrow[dr, "{A_1[C,u]}"'] & & & A_2[B, C] \\
 & A_1[C, C] \arrow[r, "{t[C/X]}"'] & A_2[C, C] \arrow[ur, "{A_2[u,C]}"'] &
\end{tikzcd}
$$

commutes, modulo $=_F$*, for every* u *such that* $x : B \vdash u : C$*.*

In what follows, when the types B and C are considered as fixed we note

$$\overline{A} = A[(B,C)/X] \qquad \widehat{A} = A[(C,B)/X]$$

3 Dinatural terms for Church-style system F

Definition 2 (simple expansion) *Let us consider a type variable* X *and a normal form* t *such that* $\Gamma \vdash t : A$ *and* $X \notin \mathcal{Z}(t)$*. Consider also a term* f *such that* $\vdash f : B \to C$*.*

Then we define the ***simple expansion*** *of* t *along* f *on* X *as the term*

$$t[f/X] = \mathcal{E}_\Gamma(t)$$

where $\mathcal{E}_\Gamma(t)$ *is defined, for* t *written as in (1), by:*

$$\mathcal{E}_\Gamma(t) = \widehat{\alpha_1}\ldots\widehat{\alpha_P}.\begin{cases}(f)\ (z)\ U_1\ldots U_n & \text{if } \mathcal{V}_\Gamma(t) = X\\ (z)\ U_1\ldots U_n & \text{otherwise}\end{cases}$$

with $U_i = \mathcal{E}_{\Gamma\cup\Delta}(T_i)$ *and:*

- $\widehat{\alpha_i} = \Lambda X_k$ *if* $\alpha_i = \Lambda X_k$*, and* $\widehat{\alpha_i} = \lambda x_j^{\widehat{A_j}}$ *if* $\alpha_i = \lambda x_j^{A_j}$
- $\Delta = \{x_1 : A_1, \ldots, x_p : A_p\}$
- $\mathcal{E}_\Gamma(\{D\}) = \{D\}$ *if* $D \in \texttt{Types}$*.*

Intuition. Informally, the construction of the simple expansion can be described as follows:

- we replace each subterm T in t of type X by $(f)\ T$
- we replace each occurrence of $\lambda x_j^{A_j}$ by $\lambda x_j^{\widehat{A_j}}$.

Example. The term $t = \lambda x^{X\to X}\lambda y^X.(x)y$ is in normal form and such that $X \notin \mathcal{Z}(t)$. Its simple expansion along f on X is then

$$t[f/X] = \lambda x^{C\to B}\lambda y^B.(f)(x)(f)y$$

With the hypotheses of the definition, the simple expansion verifies:

$$\Gamma[(C,B)/X] \ \vdash\ t[f/X] \ :\ A[(B,C)/X]$$

In fact, this term is exactly the term resulting from the dinaturality diagram. Indeed:

Theorem 1 *Let* X *be a type variable, and* t_0 *a term such that* $x : A_1 \vdash t_0 : A_2$*. Consider* $t = NF(t_0)$*, and suppose that* $X \notin \mathcal{Z}(t)$*. Then for any term* $f = \lambda x^B.\ u$ *such that* $\vdash f : B \to C$*, we have:*

$$A_1[(u,B)/X]\ ;\ t_0[B/X]\ ;\ A_2[(B,u)/X] =_F t[f/X]$$
$$A_1[(C,u)/X]\ ;\ t_0[C/X];\ A_2[(u,C)/X] =_F t[f/X]$$

which implies that t_0 *is dinatural in* X*.*

The proof of this theorem is given in Appendix A.

Corollary 1 *In the Church-style system F, a term* t_0 *is dinatural in* X *if and only if* $X \notin \mathcal{Z}(NF(t_0))$*.*

PROOF: Theorem 1 tells us that $X \notin \mathcal{Z}(\mathrm{NF}(t_0))$ is sufficient for t_0 to be dinatural. To prove that it is also necessary, consider a normal form t with $x : A_1 \vdash t : A_2$ such that, for some $D \in \texttt{Types}$, $\{D\}$ appears in t and $X \in FTV(D)$. It is easy to see that in this case, the dinaturality diagram will not commute for every u, because $\{D[B/X]\}$ appears in $t[B/X]$ whereas $\{D[C/X]\}$ appears in $t[C/X]$ at the same place.

To be more precise, consider $B = Y$ and $C = Z \to Y$, where $Z \notin FTV(t)$, and set $u = \lambda y^Z.x$, so that $x : B \vdash u : C$. The precompositions by $A_1[(u,B)/X]$ or $A_1[(C,u)/X]$ and the postcompositions by $A_2[(B,u)/X]$ or $A_2[(u,C)/X]$ will not give rise to any $\beta 2$-redex $(\Lambda Y.t)\{D\}$, so at the end if we note

$$t^{up} = A_1[(u,B)/X]\ ;\ t[B/X]\ ;\ A_2[(B,u)/X]$$
$$t^{low} = A_1[(C,u)/X]\ ;\ t[C/X]\ ;\ A_2[(u,C)/X]$$

we get $Z \notin \mathcal{Z}(\mathrm{NF}(t^{up}))$ and $Z \in \mathcal{Z}(\mathrm{NF}(t^{low}))$, so t^{up} and t^{low} are not equal modulo $=_F$ in this case. Hence, t is not dinatural. $\square$

Note that the result in [5] for simply-typed lambda-calculus was apparently slightly more general than our result for system F: it stated that, in *any* cartesian closed category, all the definable morphisms (i.e. those corresponding to simply-typed lambda-terms) are dinatural. Actually, in the proof of Theorem 1 we do not use any induction on the term u, so we can state a similar result here: in any hyperdoctrine[1], the only definable morphisms (i.e. those corresponding to system F terms) that are dinatural are those coming from a term t_0 such that $X \notin \mathcal{Z}(\mathrm{NF}(t_0))$. The proof would consist in replacing every appearance of this syntactic u by a composition with a morphism $\mathfrak{u}$.

We shall not insist on this point, but simply remark that we did not really lose any generality by working directly in the syntactic category.

4 General description of the dinaturality diagram

In the above proof of characterisation of dinatural terms, we only dealt with the dinaturality diagram for a term t_0 in the case where $X \notin \mathcal{Z}(\mathrm{NF}(t_0))$. This lazy approach is enough to do the proof, but a bit unsatisfactory, as we do not describe all the possible differences between the two morphisms of the diagram.

[1]A hyperdoctrine provides a model of Church-style system F, see [17, 24].

This gap is filled in the present section: staying with Church-style system F, we describe the dinaturality diagram in all cases. Thus we give two different constructions on normal forms, $t[u/X]_u$ and $t[u/X]_l$, which correspond to the two morphisms of the diagram, and which coincide only when $X \notin \mathcal{Z}(t)$.

The interest of this is not only to give a full description of the diagram, but also to allow to derive the characterisation of dinatural terms in Curry-style system F: in this case, there is no trivial argument similar to what we used for the reciprocal of Theorem 1, so we do need this complete study of the dinaturality diagram. The characterisation of dinatural terms in Curry-style system F will be given in section 5.

4.1 Partial substitution

Our first task is to distinguish, within a given term t in normal form such that $\Gamma \vdash t : A$, between occurrences of the variable X that come from Γ and A, and those that appear because of the type instantiations in t.

This will be done by first enriching our grammar of types by a new type variable X^*, that we see as a distinct copy of X (alternatively we can just choose an already existing type variable X^* that does not appear in t, in Γ and in A, nor in B and C, and such that $X^* \neq X$).

Then we want to substitute X by X^* in t, but only when X is used during an instantiation: in particular, the substitution does not occur for a lambda-abstraction like λx^X when the X corresponds to an occurrence of type variable in Γ or in A. Thus we need a notion of **partial substitution**, that can be stated in general as $t\langle D/X\rangle$ with $D \in \texttt{Types}$. The term resulting from the partial substitution is not necessarily well-typed; in fact, we shall see in section 5 that dinaturality for Curry-style system F is strongly connected to this question of well-typedness.

Definition 3 (partial substitution) *Given a term t in normal form such that $\Gamma \vdash t : A$, given a type variable X and a type D, the **partial substitution** $t\langle D/X\rangle$ of X by D in t is defined as the term $t\langle D/X \mid \Gamma, A\rangle$ which is constructed as follows:*

- $(\lambda x^E.t)\langle D/X \mid \Gamma, A_1 \to A_2\rangle = \lambda x^{A_1}.t\langle D/X \mid \Gamma \cup \{x : A_1\}, A_2\rangle$
- $(\Lambda Y.t)\langle D/X \mid \Gamma, \forall Y.A_0\rangle = \Lambda Y.t\langle D/X \mid \Gamma, A_0\rangle$
- $(z)T_1 \dots T_n\langle D/X \mid \Gamma, A\rangle = (z)T'_1 \dots T'_n$ *with*
  - $T'_i = \{D_i[D/X]\}$ *if* $T_i = \{D_i\}$
  - $T'_i = T_i\langle D/X \mid \Gamma, A_i\rangle$ *if* T_i *is a term*

  where the type A_i for $0 \leq i \leq n$ is again defined by induction:
  - A_0 *is such that* $z : A_0 \in \Gamma$
  - $A_{i+1} = E_1$ *if* $A_i = E_1 \to E_2$
  - $A_{i+1} = E_0[D_i[D/X]/Y]$ *if* $A_i = \forall Y.E_0$ *and* $T_i = \{D_i\}$.

Intuition. In more practical terms, $t\langle D/X\rangle$ is obtained from t by replacing any occurrence of X in any instantiation $\{E\}$ by D, and by propagating this replacement at the level of the lambda-abstractions. This propagation corresponds to the first case in the above induction, where λx^E is replaced by λx^{A_1}.

Example. By applying the partial substitution of X by D to the term $t = \lambda x^{\forall Y.(Y\to Y)}\lambda y^X.(x)\{X\}y$, we get

$$t\langle D/X\rangle = \lambda x^{\forall Y.(Y\to Y)}\lambda y^X.(x)\{D\}y$$

In particular, for $D = X^\star$, we get $t^\star = t\langle X^\star/X\rangle = \lambda x^{\forall Y.(Y\to Y)}\lambda y^X.(x)\{X^\star\}y$. Note that this term is not well-typed in Church-style system F: the subterm $(x)\{X^*\}$ requires an argument of type X^* but is given y of type X.

4.2 Produced types and expected types

The term $t^\star = t\langle X^\star/X\rangle$, for t normal form with $\Gamma \vdash t : A$, can be written as:

$$t^\star = \alpha_1 \dots \alpha_P.\,(z)\,T_1 \dots T_n \tag{2}$$

where each α_i is either of the form $\lambda x_j^{A_j}$ ($1 \leq j \leq p$) or ΛX_k ($1 \leq k \leq P - p$). But this term is not well-typed in general. This typing issue is captured by the following definition:

Definition 4 (Produced types and expected types) *For $t^\star$ as in (2) and $0 \leq i \leq n$, we define the **ith produced type** $\Pi^i_\Gamma(t^\star) \in \texttt{Types}$ and the **ith expected type** $\Sigma^i_\Gamma(t^\star) \in \texttt{Types} \cup \{\dagger\}$ as follows:*

- $\Pi^0_\Gamma(t^\star) = D$ *where D is such that* $\Gamma, x_1 : A_1, \dots, x_p : A_p \vdash z : D$, *and* $\Sigma^0_\Gamma(t^\star) = \dagger$
- *if* $\Pi^i_\Gamma(t^\star) = B_1 \to B_2$ *then* $\Sigma^{i+1}_\Gamma(t^\star) = B_1$ *and* $\Pi^{i+1}_\Gamma(t^\star) = B_2$
- *if* $\Pi^i_\Gamma(t^\star) = \forall Y.B$ *and* $T_{i+1} = \{D\}$ *then* $\Sigma^{i+1}_\Gamma(t^\star) = \dagger$ *and* $\Pi^{i+1}_\Gamma(t^\star) = B[D/Y]$.

Moreover, as the type $\Pi^n_\Gamma(t^\star)$ is always a type variable, we note $\mathcal{V}_\Gamma(t^\star) = \mathcal{V}(\Pi^n_\Gamma(t^\star)) = \Pi^n_\Gamma(t^\star)$.

Intuition. The idea behind these notions of produced and expected types is that the subterm $(z)T_1 \dots T_i$ in $t^\star$ will have the type $\Pi^i_\Gamma(t^\star)$ *provided* that each T_i which is a term has the type $\Sigma^i_\Gamma(t^\star)$ ($\Sigma^i_\Gamma(t^\star)$ being set to $\dagger$ if T_i is not a term). So, $\Pi^i_\Gamma(t^\star)$ is the type *produced* by $(z)T_1 \dots T_i$ (provided well-typedness) and $\Sigma^i_\Gamma(t^\star)$ is the type *expected* for the term T_i (to ensure well-typedness).

Of course, the terms T_i will not always produce the type $\Sigma^i_\Gamma(t^\star)$, as we have set $X \neq X^\star$. And that is why $t^\star$ is not well-typed in general.

Example. Consider again $t = \lambda x^{\forall Y.(Y \to Y)} \lambda y^X.(x)\{X\}y$, for which $t^\star = \lambda x^{\forall Y.(Y \to Y)} \lambda y^X.(x)\{X^\star\}y$. The produced types of $t^\star$ are

$$\Pi^0_\emptyset(t^\star) = \forall Y.Y \qquad \Pi^1_\emptyset(t^\star) = X^\star \to X^\star \qquad \Pi^2_\emptyset(t^\star) = X^\star$$

whereas the expected types are

$$\Sigma^0_\emptyset(t^\star) = \Sigma^1_\emptyset(t^\star) = \dagger \qquad \Sigma^2_\emptyset(t^\star) = X^\star$$

The lack of typing for $t^\star$ comes from the conflict between the expected type $\Sigma^2_\emptyset(t^\star) = X^\star$ and the produced type $\Pi^0_\Gamma(y) = X$ with $\Gamma = \{x : \forall Y.(Y \to Y), y : X\}$.

4.3 The dinaturality diagram

We are now equipped with the tools to express the content of the two terms resulting from the dinaturality diagram.

Definition 5 (Upper and lower expansions) *Let t be a normal form such that $\Gamma \vdash t : A$, and set $t^* = t\langle X^\star/X\rangle$. Consider also a term f such that $\vdash f : B \to C$.*

*Then we define the **upper expansion** of t along f on X, denoted $t[f/X]_u$, as*

$$t[f/X]_u = \mathcal{J}_{\Gamma,A}(t^*)$$

where $\mathcal{J}_{\Gamma,A}(t^*)$ is defined, for t^* given in (2), by:

$$\mathcal{J}_{\Gamma,A}(t^*) = \vec{\alpha}_1 \dots \vec{\alpha}_n. \begin{cases} (f)\ (z)\ V_1 \dots V_n & \text{if } \mathcal{V}(A) = X \\ (z)\ V_1 \dots V_n & \text{otherwise} \end{cases}$$

with $V_i = \mathcal{J}_{\Gamma \cup \Delta, \Sigma_i}(T_i)$ and:

- $\vec{\alpha}_i = \Lambda X_k$ *if* $\alpha_i = \Lambda X_k$, *and* $\vec{\alpha}_i = \lambda x_j^{D_j}$ *with* $D_j = A_j[(C,B)/X][B/X^*]$ *if* $\alpha_i = \lambda x_j^{A_j}$
- *for* $1 \leq i \leq n$, $\Sigma_i = \Sigma^i_\Gamma(t^*)$
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{J}_{\Gamma,A}(\{D\}) = \{D[B/X^\star]\}$ *if* $D \in$ `Types`.

*We also define the **lower expansion** of t along f on X, denoted $t[f/X]_l$, as*

$$t[f/X]_l = \mathcal{K}_{\Gamma,A}(t^*)$$

where $\mathcal{K}_{\Gamma,A}(t^*)$ is defined, for t^* given in (2), by:

$$\mathcal{K}_{\Gamma,A}(t^*) = \vec{\alpha}_1 \dots \vec{\alpha}_n. \begin{cases} (f)\ (z)\ W_1 \dots W_n & \text{if } \mathcal{V}_\Gamma(t^*) = X \\ (z)\ W_1 \dots W_n & \text{otherwise} \end{cases}$$

with $W_i = \mathcal{K}_{\Gamma \cup \Delta, \Sigma_i}(T_i)$ and:

- $\vec{\alpha}_i = \Lambda X_k$ *if* $\alpha_i = \Lambda X_k$, *and* $\vec{\alpha}_i = \lambda x_j^{D_j}$ *with* $D_j = A_j[(C,B)/X][C/X^*]$ *if* $\alpha_i = \lambda x_j^{A_j}$
- *for* $1 \leq i \leq n$, $\Sigma_i = \Sigma^i_\Gamma(t^*)$
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{K}_{\Gamma,A}(\{D\}) = \{D[C/X^\star]\}$ *if* $D \in$ `Types`.

Intuition. Informally, the construction of the upper expansion (resp. the lower expansion) can be described as follows:

- we consider the term $t^\star = t\langle X^\star/X\rangle$
- we replace each subterm T in $t^\star$ of expected type X (resp. of produced type X) by $(f)\ T$
- we replace each occurrence of $\lambda x_j^{A_j}$ by $\lambda x_j^{D_j}$ where $D_j = A_j[(C,B)/X][B/X^*]$ (resp. where $D_j = A_j[(C,B)/X][C/X^*]$)
- we replace each type instantiation $\{D\}$ by $\{D[B/X^\star]\}$ (resp. by $\{D[C/X^\star]\}$).

Example. Consider once again the term $t = \lambda x^{\forall Y.(Y \to Y)} \lambda y^X.(x)\{X\}y$, with $t^\star = \lambda x^{\forall Y.(Y \to Y)} \lambda y^X.(x)\{X^\star\}y$. The result of the upper expansion of t along f on X is

$$t[f/X]_u = \lambda x^{\forall Y.(Y \to Y)} \lambda y^B.(f)(x)\{B\}y$$

whereas the lower expansion of t along f on X gives

$$t[f/X]_l = \lambda x^{\forall Y.(Y \to Y)} \lambda y^B.(x)\{C\}(f)y$$

These two terms are not equal in general. As these two expansions will correspond to the two terms resulting from the dinaturality diagram applied to t, this is consistent with the fact that t is not dinatural in X, which we already knew from Corollary 1.

The symmetry between the upper and lower expansions lies in the fact that $\mathcal{J}_{\Gamma,A}$ deals with the type expected for t^* whereas $\mathcal{K}_{\Gamma,A}$ deals with the type produced by t^*. Note by the way that in $\mathcal{J}_{\Gamma,A}$ the argument Γ is not used, whereas in $\mathcal{K}_{\Gamma,A}$ the argument A is not used.

The two expansions verify:

$$\Gamma[(C,B)/X] \vdash t[f/X]_u : A[(B,C)/X]$$
$$\Gamma[(C,B)/X] \vdash t[f/X]_l : A[(B,C)/X]$$

In fact, these two terms give us a complete description of the dinaturality diagram. Indeed:

Theorem 2 *Let X be a type variable, and t_0 a term such that $x : A_1 \vdash t_0 : A_2$. If we note $t = NF(t_0)$, then for any term $f = \lambda x^B.\ u$ such that $\vdash f : B \to C$, we have:*

$$A_1[(u,B)/X]\ ;\ t_0[B/X]\ ;\ A_2[(B,u)/X] =_F t[f/X]_u$$
$$A_1[(C,u)/X]\ ;\ t_0[C/X]\ ;\ A_2[(u,C)/X] =_F t[f/X]_l$$

The proof of this theorem is given in Appendix B.

5 Dinatural terms for Curry-style system F

5.1 Presentation of the calculus

Curry-style system F is a variant of system F where there is no type indication in the terms. Otherwise said, the grammar of terms is the grammar of **pure lambda-terms**:

$$\tau\ ::=\ x\ |\ \lambda x.\tau\ |\ (\tau)\tau$$

The straightforward presentation of the calculus consists in saying that the typing rules for the second-order quantifier become

$$\frac{\Gamma \vdash_{\text{Cu}} \tau : A}{\Gamma \vdash_{\text{Cu}} \tau : \forall X.A}\ \text{if } X \notin \mathit{FTV}(\Gamma)$$
$$\frac{\Gamma \vdash_{\text{Cu}} \tau : \forall X.A}{\Gamma \vdash_{\text{Cu}} \tau : A[B/X]}$$

and that we consider terms modulo untyped β- and η-equalities. However this naive approach encounters many troubles regarding the subject reduction of the calculus. These questions have been analysed in [27, 18] for η-reduction, and in [7] for η-expansion.

We will choose here an alternative approach, by considering that Curry-style system F is just **Church-style system F modulo type erasure**. If we had only the β-equality in the system, this would be equivalent to the usual presentation of Curry-style system F. As we also consider η-equality, this approach will allow us to avoid the technical problems related to subject reduction.

Concretely, we consider the operation of erasure from Curry-style terms to pure lambda-terms:

$$\begin{aligned}
\text{erase}(x) &= x\\
\text{erase}(\lambda x^A.t) &= \lambda x.\text{erase}(t)\\
\text{erase}((t)u) &= (\text{erase}(t))\text{erase}(u)\\
\text{erase}(\Lambda X.t) &= \text{erase}(t)\\
\text{erase}((t)\{A\}) &= \text{erase}(t)
\end{aligned}$$

A Curry-style sequent is then of the form $\Gamma \vdash_{\text{Cu}} \tau : A$, and it is valid if there exists t such that $\text{erase}(t) = \tau$ and $\Gamma \vdash t : A$. We introduce a new congruence $t =_{\text{Cu}} u$ between Church-style terms by adding to the usual Church-style equalities the new equality:

$$t =_\rho u \qquad \Leftrightarrow \qquad \text{erase}(t) = \text{erase}(u)$$

This leads to the following definition for dinaturality:

Definition 6 (dinatural term, Curry-style) *A term τ such that $x : A_1 \vdash_{Cu} \tau : A_2$ is* ***dinatural in X*** *if, for every t such that erase$(t) = \tau$ and $x : A_1 \vdash t : A_2$, the diagram*

$$\begin{tikzcd}
 & A_1[B,B] \arrow[r, "t[B/X]"] & A_2[B,B] \arrow[dr, "A_2[B,u]"] & \\
A_1[C,B] \arrow[ur, "A_1[u,B]"] \arrow[dr, "A_1[C,u]"'] & & & A_2[B,C] \\
 & A_1[C,C] \arrow[r, "t[C/X]"'] & A_2[C,C] \arrow[ur, "A_2[u,C]"'] &
\end{tikzcd}$$

commutes modulo $=_{Cu}$ for every u such that $x : B \vdash u : C$.

Of course dinaturality for a Curry-style term only makes sense in a given typing context: saying that τ is dinatural in X has no absolute meaning, we have to give a typing sequent $x : A_1 \vdash_{\text{Cu}} \tau : A_2$ for the question to become meaningful.

One can define a composition on Curry-style terms, again by substitution: given $x : A \vdash_{\text{Cu}} \sigma : B$ and $x : B \vdash_{\text{Cu}} \tau : C$, we define $\sigma;\tau = \tau[\sigma/x]$, and it is easy to check that $x : A \vdash_{\text{Cu}} \sigma;\tau : C$. Moreover, for two composable Church-style terms u and v, we have

$$\text{erase}(u;v) = \text{erase}(u)\ ;\ \text{erase}(v)$$

Thus, given for two terms t_1, t_2 such that $x : A_1 \vdash t_1 : A_2$, $x : A_1 \vdash t_2 : A_2$ and $\text{erase}(t_1) = \text{erase}(t_2)$, and given u such that $x : B \vdash u : C$, if we note $t_1^{up} = A_1uB\ ;\ t_1[B/X]\ ;\ A_2Bu$ and $t_2^{up} = A_1uB\ ;\ t_2[B/X]\ ;\ A_2Bu$ then we have:

$$\begin{aligned}
\text{erase}(t_1^{up}) &= \text{erase}(A_1uB);\text{erase}(t_1);\text{erase}(A_2Bu)\\
&= \text{erase}(A_1uB);\text{erase}(t_2);\text{erase}(A_2Bu)\\
&= \text{erase}(t_2^{up})
\end{aligned}$$

and similarly for the lower term of the diagram. This simple remark gives us a powerful lemma:

Lemma 1 *A term τ such that $x : A_1 \vdash_{Cu} \tau : A_2$ is dinatural in X if and only if there exists a Church-style term t such that erase$(t) = \tau$, $x : A_1 \vdash t : A_2$ and the diagram*

$$\begin{tikzcd}
 & A_1[B,B] \arrow[r, "t[B/X]"] & A_2[B,B] \arrow[dr, "A_2[B,u]"] & \\
A_1[C,B] \arrow[ur, "A_1[u,B]"] \arrow[dr, "A_1[C,u]"'] & & & A_2[B,C] \\
 & A_1[C,C] \arrow[r, "t[C/X]"'] & A_2[C,C] \arrow[ur, "A_2[u,C]"'] &
\end{tikzcd}$$

commutes modulo $=_{Cu}$ *for every* u *such that* $x : B \vdash u : C$.

Otherwise said, the dinaturality diagram needs to be checked only on one Church-style term t in order to ensure the dinaturality of τ.

5.2 Characterisation of dinatural terms

Theorem 3 *Let* $x : A_1 \vdash_{Cu} \tau : A_2$ *be a valid Curry-style sequent. The following propositions are equivalent:*

(i) *τ is dinatural in X*

(ii) *for every Church-style term t_0 such that erase$(t_0) = \tau$ and $x : A_1 \vdash t_0 : A_2$, and for every $D \in$ `Types`, we have, for $t = NF(t_0)$:*

$$x : A_1 \vdash t\langle D/X\rangle : A_2$$

(iii) *there exists a Church-style term t_0 such that erase$(t_0) = \tau$, $x : A_1 \vdash t_0 : A_2$ and $X \notin \mathcal{Z}(t_0)$.*

PROOF: We first prove (ii)⇒(iii): as $x : A_1 \vdash_{\mathrm{Cu}} \tau : A_2$, we necessarily have a Church-style term t_0 such that erase$(t_0) = \tau$ and $x : A_1 \vdash t_0 : A_2$. Then we choose $D \in$ `Types` such that none of the type variables in D appear in A_1, A_2 or t_0, and we have $x : A_1 \vdash t\langle D/X\rangle : A_2$ for $t = \mathrm{NF}(t_0)$. By pulling back from $t\langle D/X\rangle$ all the equalities that go from t_0 to t, we obtain a term t_0' such that erase$(t_0') = \tau$, $x : A_1 \vdash t_0' : A_2$ and $X \notin \mathcal{Z}(t_0')$.

To prove (iii)⇒(i), consider t_0 satisfying erase$(t_0) = \tau$, $x : A_1 \vdash t_0 : A_2$ and $X \notin \mathcal{Z}(t_0)$, and take $t = \mathrm{NF}(t_0)$. By Theorem 1, as we have $X \notin \mathcal{Z}(t)$, t is dinatural in Church-style system F, so for any Church-style term u, the two terms $t^{up} = A_1uB\ ;\ t[B/X]\ ;\ A_2Bu$ and $t^{low} = A_1Cu\ ;\ t[C/X]\ ;\ A_2uC$ are equal modulo $=_F$. Thus they are also equal modulo $=_{\mathrm{Cu}}$ and, thanks to Lemma 1, τ is dinatural in Curry-style system F.

Finally, the proof that (i)⇒(ii) will make use of Theorem 2. For this proof, we set $A = A_2$ and $\Gamma = \{x : A_1\}$, so that we can reuse the names A_j in a different context.

Consider τ dinatural, t_0 such that erase$(t_0) = \tau$ and $\Gamma \vdash t_0 : A$, $t = \mathrm{NF}(t_0)$ and $t^\star = t\langle X^\star/X\rangle$. We want to prove that $\Gamma \vdash t^\star : A$ by induction on $t^\star$. Thus, we write $t^\star$ as in (2):

$$t^\star = \alpha_1 \ldots \alpha_P.\,(z)\,T_1 \ldots T_n$$

where each α_i is either of the form $\lambda x_j^{A_j}$ ($1 \leq j \leq p$) or ΛX_k ($1 \leq k \leq P - p$) and we suppose as an induction hypothesis[2] that for $1 \leq i \leq n$, $\Gamma \cup \Delta \vdash T_i : \Sigma^i_\Gamma(t^\star)$ with $\Delta = \{x_1 : A_1, \ldots, x_p : A_p\}$.This ensures that

$$\Gamma \cup \Delta \vdash (z)T_1 \ldots T_n : \mathcal{V}_\Gamma(t^\star) \tag{3}$$

[2]The type A has to be seen as the type expected for $t^\star$, as well as $\Sigma^i_\Gamma(t^\star)$ is the typed expected for T_i.

As in the proof of Corollary 1, we consider $B = Y$, $C = Z \to Y$ and $u = \lambda y^Z.x$, so that $x : B \vdash u : C$. If $f = \lambda x^B.u$, then we cannot have $(f)v =_{\mathrm{Cu}} v$ for a term v, because $(f)v$ has an extra lambda-abstraction. But we need to have $t[f/X]_u =_{\mathrm{Cu}} t[f/X]_l$. So, by comparing the expressions of $\mathcal{J}_{\Gamma,A}$ and $\mathcal{K}_{\Gamma,A}$ in Definition 5 we see that we must have:

$$\mathcal{V}(A) = X \Leftrightarrow \mathcal{V}_\Gamma(t^\star) = X$$

As $\mathcal{V}(A)$ and $\mathcal{V}_\Gamma(t^\star)$ can only differ by occurrences of $X^\star$ and X, this means that $\mathcal{V}(A) = \mathcal{V}_\Gamma(t^\star)$.

Together with (3), this implies that $\Gamma \cup \Delta \vdash (z)T_1 \ldots T_n : \mathcal{V}(A)$. By construction of the partial substitution, the α_i's wear the proper type indication, so that $\Gamma \vdash t^\star : A$, and we are done with the induction.

Thus, the term $t^\star$ is well-typed in this case! But then, as $X^\star$ does not appear in Γ or in A, we have $\Gamma \vdash t^\star[D/X^\star] : A$ for any D, which means $\Gamma \vdash t\langle D/X\rangle : A$. □

From Theorem 3 one can extract an **algorithm** to check whether the term τ such that $x : A_1 \vdash_{\mathrm{Cu}} \tau : A_2$ is dinatural. Indeed, suppose that we have at disposal a term t_0 such that erase$(t_0) = \tau$ and $x : A_1 \vdash t_0 : A_2$. Then we consider a type D such that $X \notin FTV(D)$, we construct $t = \mathrm{NF}(t_0)$ and we check whether $x : A_1 \vdash t\langle D/X\rangle : A_2$ (this is decidable). If the term τ is dinatural, then (i)⇒(ii) ensures that $x : A_1 \vdash t\langle D/X\rangle : A_2$. Reciprocally, if $x : A_1 \vdash t\langle D/X\rangle : A_2$ then (iii)⇒(i) ensures that τ is dinatural.

Now, if we do not have this term t_0, we can find it by enumerating all the Church-style terms whose erasure is τ, until we find one such that $x : A_1 \vdash t_0 : A_2$. As we know that $x : A_1 \vdash_{\mathrm{Cu}} \tau : A_2$, this process will terminate. Thus the dinaturality of a Curry-style term τ such that $x : A_1 \vdash_{\mathrm{Cu}} \tau : A_2$ is a decidable question.

Example 1. Consider the Curry-style term $\tau = \lambda x \lambda y.(x)y$, such that $\vdash \tau : (\forall Y.Y \to Y) \to X \to X$. The Church-style term $t_1 = \lambda x^{\forall Y.Y \to Y} \lambda y^X.(x)\{X\}y$ is such that erase$(t_1) = \tau$ and $\vdash t_1 : (\forall Y.Y \to Y) \to X \to X$. Now, given $D \in$ `Types`, we have $t_1\langle D/X\rangle = \lambda x^{\forall Y.Y \to Y} \lambda y^X.(x)\{D\}y$. So, in general (actually, as soon as $D \neq X$) we have:

$$\nvdash t_1\langle D/X\rangle : (\forall Y.Y \to Y) \to X \to X$$

Thus, τ is not dinatural in X in this typing context.

Example 2. Consider now the same Curry-style term $\tau = \lambda x \lambda y.(x)y$, but typed as $\vdash \tau : (\forall Y.Y \to X) \to (\forall Z.Z) \to X$. The Church-style term $t_2 = \lambda x^{\forall Y.Y \to X} \lambda y^{\forall Z.Z}.(x)\{X\}\ y\{X\}$ is such that erase$(t_2) = \tau$ and $\vdash t_2 : (\forall Y.Y \to X) \to (\forall Z.Z) \to X$. For any $D \in$ `Types`, we have $t_2\langle D/X\rangle = \lambda x^{\forall Y.Y \to X} \lambda y^{\forall Z.Z}.(x)\{D\}\ y\{D\}$ so:

$$\vdash t_2\langle D/X\rangle : (\forall Y.Y \to X) \to (\forall Z.Z) \to X$$

This means that τ is dinatural in X in this typing context.

Further directions

Among the future developments of this work lies a possible connection between the process of simple expansion described in this article, and the programming concept of *hooking* (see for example [6]).

Another perspective is the analysis of the ideas presented in [9], where dinaturality is avoided by strongly restricting the category we are working on. This analysis could be done either syntactically, in the spirit of the present work, or semantically using game models.

Acknowledgements. I would like to thank Philip Scott, who shared with me his deep knowledge on the question of dinaturality, and Olivier Laurent for his many precious comments on this article.

References

[1] M. Abadi, L. Cardelli, and P.-L. Curien. Formal parametric polymorphism. *Theoretical Computer Science*, 121:9–58, 1993.

[2] E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70(1):35–64, 1990.

[3] J. de Lataillade. Strachey parametricity and game semantics. Submitted for publication. Available at `http://www.pps.jussieu.fr/~delatail/param.pdf`.

[4] J. de Lataillade. *Quantification du second ordre dans les jeux - Application aux isomorphismes de types*. Thèse de Doctorat, Université Paris 7, November 2007.

[5] P. J. Freyd, J.-Y. Girard, A. Scedrov, and P. J. Scott. Semantic parametricity in polymorphic lambda calculus. In *Proceedings of the third annual symposium on Logic In Computer Science*, pages 274–279. IEEE, IEEE Computer Society Press, 1988.

[6] G. Froehlich, H. J. Hoover, L. Liu, and P. Sorenson. Hooking into object-oriented application frameworks. In *Proceedings of the 19th International Conference on Software Engineering*, pages 491–501, New York, NY, USA, 1997. ACM.

[7] N. Ghani. Eta expansions in system F. Technical report, LIENS-DMI, Ecole Normale Superieure, 1996.

[8] J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*. Thèse de doctorat, Université Paris VII, 1972.

[9] J.-Y. Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45:159–192, 1986.

[10] J.-Y. Girard, A. Scedrov, and P. J. Scott. Normal forms and cut-free proofs as natural transformations. In Y. Moschovakis, editor, *Logic from Computer Science*, volume 21, pages 217–242. MSRI Publications, Springer, 1992.

[11] M. Hasegawa. Recursion from cyclic sharing: Traced monoidal categories and models of cyclic lambda calculi. In *Proceedings of the Third International Conference on Typed Lambda Calculi and Applications*, pages 196–213, London, UK, 1997. Springer-Verlag.

[12] D. Hughes. *Hypergame semantics: full completeness for system F*. D.Phil. thesis, Oxford University, 2000.

[13] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119:447–468, 1996.

[14] J. Laird. Exceptions, continuations and macro-expressiveness. In *Proceedings of the 11th European Symposium on Programming Languages and Systems*, pages 133–146, London, UK, 2002. Springer-Verlag.

[15] S. M. Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, Second edition, 1998.

[16] O. Laurent. Classical isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5):969–1004, Oct. 2005.

[17] F. W. Lawvere. Equality in hyperdoctrines and the comprehension schema as an adjoint functor. In *Proceedings on Applications of Categorical Logic*, 1970.

[18] J. C. Mitchell. Polymorphic type inference and containment. *Inf. Comput.*, 76(2-3):211–249, 1988.

[19] A. Murawski and L. Ong. Evolving games and essential nets for affine polymorphism. In S. Abramsky, editor, *Typed Lambda Calculi and Applications '01*, volume 2044 of *LNCS*. Springer, 2001.

[20] C.-H. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science*, pages 81–90, Washington, DC, USA, 2006. IEEE Computer Society.

[21] G. Plotkin and M. Abadi. A logic for parametric polymorphism. In M. Bezem and J. F. Groote, editors, *International Conference on Typed Lambda Calculi and Applications*, pages 361–375, Utrecht, The Netherlands, 1993. Springer-Verlag.

[22] J. C. Reynolds. Towards a theory of type structure. In *Programming Symposium, Proceedings Colloque sur la Programmation*, pages 408–423, London, UK, 1974. Springer-Verlag.

[23] J. C. Reynolds. Types, abstraction and parametric polymorphism. In *International Federation for Information Processing Congress*, pages 513–523, 1983.

[24] R. A. G. Seely. Categorical semantics for higher-order polymorphic lambda-calculus. *Journal of Symbolic Logic*, 52(4):969–989, 1987.

[25] C. Strachey. Fundamental concepts in programming languages (reprinting of a course given in 1967). *Higher-Order and Symbolic Computation*, 13(1):11–49, 2000.

[26] P. Wadler. Theorems for free! In *Proceedings 4th Int. Conf. on Funct. Prog. Languages and Computer Arch., FPCA'89, London, UK, 11–13 Sept 1989*, pages 347–359. ACM Press, New York, 1989.

[27] J. B. Wells. *Type inference for system F with and without the eta rule*. PhD thesis, Boston University, Boston, MA, USA, 1996. Major Professor-Assaf J. Kfoury.

A Proof of Theorem 1

Theorem 1 *Let X be a type variable, and t_0 a term such that $x : A_1 \vdash t_0 : A_2$. Consider $t = NF(t_0)$, and suppose that $X \notin \mathcal{Z}(t)$. Then for any term $f = \lambda x^B.\ u$ such that $\vdash f : B \to C$, we have:*

$$A_1[(u,B)/X]; t_0[B/X]; A_2[(B,u)/X] =_F t[f/X] \quad (4)$$
$$A_1[(C,u)/X]; t_0[C/X]; A_2[(u,C)/X] =_F t[f/X] \quad (5)$$

which means that t_0 is dinatural in X.

PROOF: We first prove Equation (4). For dealing with the proof, we consider two mutually recursive functions on normal forms, $\mathcal{F}_\Gamma(t)$ and $\mathcal{G}_\Gamma(t)$, which are defined on Figure 2.

To prove (4), it suffices to show the following:

$$A_1[(u,B)/X]\ ;\ t[B/X] =_F \mathcal{F}_\Gamma(t) \quad (6)$$
$$\mathcal{F}_\Gamma(t)\ ;\ A_2[(B,u)/X] =_F \mathcal{E}_\Gamma(t) \quad (7)$$

with $\Gamma = \{x : \widehat{A_1}\}$ (one can work directly with t instead of t_0, as $t =_F t_0$).

To prove the equality (6), we consider a term variable y not appearing in t, we set $t^B = t[B/X][y/x]$ and we note that $A_1[(u,B)/X]\ ;\ t[B/X] = t^B[(A_1 f B)x/y]$. Then we proceed by induction[3] on A_1 to show that $t^B[(A_1 f B)x/y] =_F \mathcal{F}_\Gamma(t)$:

• If A_1 contains no arrow then $A_1 f B = id_{A_1[B/X]}$ and the result is immediate.

• If $A_1 = \forall_1.(\forall_2.A'_1 \to \ldots \to A'_p \to X) \to Z$ where each $\forall_i$ is a sequence of quantifiers $\forall X^i_1 \ldots \forall X^i_{N_i}$ and Z is a type variable, then

$$A_1 f B = \forall_1.(\forall_2.A'_1 f B \to \ldots \to A'_p f B \to f) \to Z[B/X]$$

The variable x being of type $A_1[C,B]$, we can write, by η-expanding it:

$$x =_F \mathbf{\Lambda}_1 \lambda z^{\overline{D}}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.(z)\mathbf{X}^2 x_1 \ldots x_p)$$

with $D = \forall_2.A'_1 \to \ldots \to A'_p \to X$, $\mathbf{\Lambda}_i = \Lambda X^i_1 \ldots \Lambda X^i_{N_i}$ and $\mathbf{X}^i = \{X^i_1\} \ldots \{X^i_{N_i}\}$. Thus:

$$(A_1 f B)x =_F \mathbf{\Lambda}_1 \lambda z^{D[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.\\ (f)(z)\mathbf{X}^2(A'_1 f B)x_1 \ldots (A'_p f B)x_p)$$

We can then write, for $\tilde{t} = t^B[(A_1 f B)x/y]$:

$$\tilde{t} =_F t^B[\mathbf{\Lambda}_1 \lambda z^{D[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.\\ (f)(z)\mathbf{X}^2(A'_1 f B)x_1 \ldots (A'_p f B)x_p)/y]$$
$$\tilde{t} =_F t^B[\mathbf{\Lambda}_1 \lambda z^{D[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.\\ (f)(z)y_1 \ldots y_p)/y][(A'_i f B)x_i/y_i]_i$$

where $y_1, \ldots, y_p$ are chosen fresh in t^B.

This gives us an inductive construction that is exactly equivalent to the definition of $\mathcal{F}_\Gamma(t)$ with $\Gamma = \{x : \widehat{A_1}\}$. Indeed, when constructing $\mathcal{F}_\Gamma(t)$, any time we see x in t applied to an argument T, we replace this argument by $\mathcal{G}_\Gamma(T)$: this corresponds to the application of f in the above formula; and then, all the variables that are bound in the head of T are added to the context Γ, so we will treat them like x: this corresponds to the substitution $[(A'_i f B)x_i/y_i]_i$ in the above formula.

• If $A_1 = \forall_1.(\forall_2.A'_1 \to \ldots \to A'_p \to Y) \to Z$ with $Y \neq X$, then

$$A_1 f B = \forall_1.(\forall_2.A'_1 f B \to \ldots \to A'_p f B \to Y) \to Z[B/X]$$

Again we rewrite x as

$$x =_F \mathbf{\Lambda}_1 \lambda z^{\overline{D}}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.(z)\mathbf{X}^2 x_1 \ldots x_p)$$

so that:

$$(A_1 f B)x =_F \mathbf{\Lambda}_1 \lambda z^{D[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.\\ (z)\mathbf{X}^2(A'_1 f B)x_1 \ldots (A'_p f B)x_p)$$

and finally, for $\tilde{t} = t^B[(A_1 f B)x/y]$:

$$\tilde{t} =_F t^B[\mathbf{\Lambda}_1 \lambda z^{D[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_1^{\widehat{A'_1}} \ldots \lambda x_p^{\widehat{A'_p}}.\\ (z)\mathbf{X}^2 y_1 \ldots y_p)/y][(A'_i f B)x_i/y_i]_i$$

where $y_1, \ldots, y_p$ are chosen fresh in t^B.

Again, this is equivalent to the definition of $\mathcal{F}_\Gamma(t)$ with $\Gamma = \{x : \widehat{A_1}\}$: any time we see x in t applied to an argument T, we replace this argument by $\mathcal{G}_\Gamma(T)$ (but this time we do not apply f because $\mathcal{V}_\Gamma(T) \neq X$), and all the variables that are bound in the head of T are added to the context Γ, so we will treat them like x.

• If $A_1 = \forall_1.D_1 \to \ldots \to D_k \to Z$ where each D_j has the form $D_j = \forall_{j+1}.A^j_1 \to \ldots \to A^j_{p_j} \to Y_j$, then we can write

$$x =_F \mathbf{\Lambda}_1 \lambda z_1^{\overline{D_1}} \ldots \lambda z_k^{\overline{D_k}}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{\widehat{A^1_1}} \ldots \lambda x_{1,p_1}^{\widehat{A^1_p}}.\\ (z_1)\mathbf{X}^2 x_{1,1} \ldots x_{1,p_1}) \ldots (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A^k_1}} \ldots \lambda x_{k,p_k}^{\widehat{A^k_{p_k}}}.\\ (z_k)\mathbf{X}^{k+1} x_{k,1} \ldots x_{k,p_k})$$

[3] We take steps in this induction to make the ideas more obvious, but the general case is in fact given by the fourth case of the induction.

Consider t defined as in (1):

$$t = \alpha_1 \dots \alpha_P.\ (z)\ T_1 \dots T_n$$

where each α_i is either of the form $\lambda x_j^{A_j}$ ($1 \leq j \leq p$) or ΛX_k ($1 \leq k \leq P - p$).

Then we set:

$$\mathcal{F}_\Gamma(t) = \alpha_1^B \dots \alpha_P^B.\ \begin{cases} (z)\ \mathcal{F}_\Gamma(T_1) \dots \mathcal{F}_\Gamma(T_n) & \text{if } z \notin \Gamma \\ (z)\ \mathcal{G}_\Gamma(T_1) \dots \mathcal{G}_\Gamma(T_n) & \text{if } z \in \Gamma \end{cases}$$

$$\mathcal{G}_\Gamma(t) = \widehat{\alpha_1} \dots \widehat{\alpha_P}.\ \begin{cases} (z)\ \mathcal{F}_{\Gamma\cup\Delta}(T_1) \dots \mathcal{F}_{\Gamma\cup\Delta}(T_n) & \text{if } z \notin \Gamma \text{ and } \mathcal{V}_\Gamma(t) \neq X \\ (z)\ \mathcal{G}_{\Gamma\cup\Delta}(T_1) \dots \mathcal{G}_{\Gamma\cup\Delta}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) \neq X \\ (f)\ (z)\ \mathcal{F}_{\Gamma\cup\Delta}(T_1) \dots \mathcal{F}_{\Gamma\cup\Delta}(T_n) & \text{if } z \notin \Gamma \text{ and } \mathcal{V}_\Gamma(t) = X \\ (f)\ (z)\ \mathcal{G}_{\Gamma\cup\Delta}(T_1) \dots \mathcal{G}_{\Gamma\cup\Delta}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) = X \end{cases}$$

where

- $\widehat{\alpha_i} = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\widehat{\alpha_i} = \lambda x_j^{\widehat{A_j}}$ if $\alpha_i = \lambda x_j^{A_j}$
- $\alpha_i^B = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\alpha_i^B = \lambda x_j^{A_j[B/X]}$ if $\alpha_i = \lambda x_j^{A_j}$
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{F}_\Gamma(\{D\}) = \mathcal{G}_\Gamma(\{D\}) = \{D\}$ if $D \in \mathsf{Types}$

$$\mathcal{F}'_\Gamma(t) = \alpha_1^C \dots \alpha_P^C.\ \begin{cases} (z)\ \mathcal{F}'_\Gamma(T_1) \dots \mathcal{F}'_\Gamma(T_n) & \text{if } z \notin \Gamma \\ (z)\ \mathcal{G}'_\Gamma(T_1) \dots \mathcal{G}'_\Gamma(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) \neq X \\ (f)\ (z)\ \mathcal{G}'_\Gamma(T_1) \dots \mathcal{G}'_\Gamma(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) = X \end{cases}$$

$$\mathcal{G}'_\Gamma(t) = \widehat{\alpha_1} \dots \widehat{\alpha_P}.\ \begin{cases} (z)\ \mathcal{F}'_{\Gamma\cup\Delta}(T_1) \dots \mathcal{F}'_{\Gamma\cup\Delta}(T_n) & \text{if } z \notin \Gamma \\ (z)\ \mathcal{G}'_{\Gamma\cup\Delta}(T_1) \dots \mathcal{G}'_{\Gamma\cup\Delta}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) \neq X \\ (f)\ (z)\ \mathcal{G}'_{\Gamma\cup\Delta}(T_1) \dots \mathcal{G}'_{\Gamma\cup\Delta}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t) = X \end{cases}$$

where

- $\widehat{\alpha_i} = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\widehat{\alpha_i} = \lambda x_j^{\widehat{A_j}}$ if $\alpha_i = \lambda x_j^{A_j}$
- $\alpha_i^C = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\alpha_i^C = \lambda x_j^{A_j[C/X]}$ if $\alpha_i = \lambda x_j^{A_j}$
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{F}'_\Gamma(\{D\}) = \mathcal{G}'_\Gamma(\{D\}) = \{D\}$ if $D \in \mathsf{Types}$

Figure 2. Constructions used in the proof of Theorem 1

so that

$$
\begin{aligned}
(A_1 f B)x =_F\ & \mathbf{\Lambda}_1 \lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(x)\mathbf{X}^1 \\
& (\mathbf{\Lambda}_2.\lambda x_{1,1}^{\widehat{A_1^1}} \ldots \lambda x_{1,p_1}^{\widehat{A_p^1}}.(v_1)(z_1)\mathbf{X}^2(A_1^1 f B)x_{1,1} \\
& \ldots (A_{p_1}^1 f B)x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A_1^k}} \ldots \lambda x_{k,p_k}^{\widehat{A_{p_k}^k}}. \\
& (v_k)(z_k)\mathbf{X}^{k+1}(A_1^k f B)x_{k,1} \ldots (A_{p_k}^k f B)x_{k,p_k})
\end{aligned}
$$

with $v_j = f$ if $Y_j = X$ and $v_j = id_{Y_j}$ otherwise. Finally, we obtain, for $\tilde{t} = t^B[(A_1 f B)x/y]$:

$$
\begin{aligned}
\tilde{t} =_F\ & t^B[\lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{\widehat{A_1^1}} \ldots \\
& \lambda x_{1,p_1}^{\widehat{A_p^1}}.(v_1)(z_1)\mathbf{X}^2 y_{1,1} \ldots y_{1,p_1}) \ldots (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A_1^k}} \\
& \ldots \lambda x_{k,p_k}^{\widehat{A_{p_k}^k}}.(v_k)(z_k)\mathbf{X}^{k+1} y_{k,1} \ldots y_{k,p_k})/y] \\
& \quad [(A_j^i f B)x_{i,j}/y_{i,j}]_{i,j}
\end{aligned}
$$

where all the $y_{i,j}$'s are chosen fresh.

This is again equivalent to the definition of $\mathcal{F}_\Gamma(t)$ with $\Gamma = \{x : \widehat{A_1}\}$: in particular when T is an argument of x, we apply f in T exactly in the case where $\mathcal{V}_\Gamma(T) = X$.

- Every other form for the type A_1 can be related to the previous case by applying the isomorphism $\forall Y.A \to B \simeq A \to \forall Y.B$ (with $Y \notin FTV(A)$). The result for $t^B[(A_1 f B)x/y]$ is then similar to the one given above (only the positions of the $\mathbf{\Lambda}$'s and the $\mathbf{X}$'s change), and can be shown to be equal to $\mathcal{F}_\Gamma(t)$.

For Equation (7) we note that $\mathcal{F}_\Gamma(t)\ ;\ A_2[(B,u)/X] = (A_2 B f)\mathcal{F}_\Gamma(t)$ and we use again an induction, this time on A_2:

- If $A_2 = X$ then $(A_2 B f)\mathcal{F}_\Gamma(t) = (f)\mathcal{F}_\Gamma(t)$. And the inductive construction of $\mathcal{F}_\Gamma(t)$ only calls the function $\mathcal{G}_\Gamma(t)$ in this case, so we have $(f)\mathcal{F}_\Gamma(t) = \mathcal{E}_\Gamma(t)$.

- If $A_2 = Y$ with $Y \neq X$ then $(A_2 B f)\mathcal{F}_\Gamma(t) = \mathcal{F}_\Gamma(t)$. And the inductive construction of $\mathcal{F}_\Gamma(t)$ only calls the function $\mathcal{G}_\Gamma(t)$ in this case, so we have $\mathcal{F}_\Gamma(t) = \mathcal{E}_\Gamma(t)$.

- If $A_2 = \forall_1.D_1 \to \ldots \to D_k \to Z$ where each D_j has the form $D_j = \forall_{j+1}.A_1^j \to \ldots \to A_{p_j}^j \to Y_j$, then for any term h of type $A_2[B/X]$ we can write

$$
\begin{aligned}
h =_F & \mathbf{\Lambda}_1 \lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(h)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]} \\
& \ldots \lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2 x_{1,1} \ldots x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}. \\
& \lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}.(z_k)\mathbf{X}^{k+1} x_{k,1} \ldots x_{k,p_k})
\end{aligned}
$$

so that

$$
\begin{aligned}
(A_2 B f)h =_F\ & \mathbf{\Lambda}_1 \lambda z_1^{\widehat{D_1}} \ldots \lambda z_k^{\widehat{D_k}}.(v)(h)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]} \\
& \ldots \lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2(A_1^1 B f)x_{1,1} \ldots (A_{p_1}^1 B f) \\
& x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}. \\
& (z_k)\mathbf{X}^{k+1}(A_1^k B f)x_{k,1} \ldots (A_{p_k}^k B f)x_{k,p_k})
\end{aligned}
$$

with $v = f$ if $Z = X$ and $v = id_Z$ otherwise. Now let us take $h = \mathcal{F}_\Gamma(t)$; we can rewrite it as

$$
\mathcal{F}_\Gamma(t) =_F \Lambda Y_1 \ldots \Lambda Y_{N_1} \lambda y_1^{D_1[B/X]} \ldots \lambda y_k^{D_k[B/X]}.\, F
$$

for some $F = (z)T_1 \ldots T_n$. If we note $F' = F[X_1^1/Y_1] \ldots [X_{N_1}^1/Y_{N_1}]$, we get, for $\breve{t} = (A_2 B f)\mathcal{F}_\Gamma(t)$:

$$
\begin{aligned}
\breve{t} =_F & \mathbf{\Lambda}_1 \lambda z_1^{\widehat{D_1}} \ldots \lambda z_k^{\widehat{D_k}}.(v)\ F'[\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]} \ldots \\
& \lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2 y_{1,1} \ldots y_{1,p_1}/y_1] \ldots [\mathbf{\Lambda}_{k+1}. \\
& \lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}.(z_k)\mathbf{X}^{k+1} y_{k,1} \ldots \\
& y_{k,p_k}/y_k]\ [(A_j^i B f)x_{i,j}/y_{i,j}]_{i,j}
\end{aligned}
$$

where the $y_{i,j}$'s are again chosen fresh.

The result of this construction is exactly the term $\mathcal{E}_\Gamma(t)$: indeed, the variables $y_1, \ldots, y_k$ and $y_{i,j}$ in F' are the places, in the definition of $\mathcal{F}_\Gamma(t)$, where we have used $\mathcal{F}_\Gamma$ and not $\mathcal{G}_\Gamma$. Hence, with the application of v at those places, we retrieve $\mathcal{E}_\Gamma(t)$.

- Every other form for the type A_2 is isomorphic to the above form, and the result remains true in these cases (we just have to move the positions of the $\mathbf{\Lambda}$'s and the $\mathbf{X}$'s in the formulas).

Thus we have proved the equality (4). The proof of (5) is very similar to what we have done above, with two new functions $\mathcal{F}'_\Gamma(t)$ and $\mathcal{G}'_\Gamma(t)$ defined on Figure 2.

The equalities to prove are in this case:

$$
A_1[(C,u)/X]\ ;\ t[C/X] =_F \mathcal{F}'_\Gamma(t) \tag{8}
$$

$$
\mathcal{F}'_\Gamma(t)\ ;\ A_2[(u,C)/X] =_F \mathcal{E}_\Gamma(t) \tag{9}
$$

with $\Gamma = \{x : \widehat{A_1}\}$. □

B Proof of Theorem 2

Theorem 2 *Let X be a type variable, and t_0 a term such that $x : A_1 \vdash t_0 : A_2$. If we note $t = NF(t_0)$, then for any term $f = \lambda x^B.\, u$ such that $\vdash f : B \to C$, we have:*

$$
A_1[(u,B)/X]; t_0[B/X]; A_2[(B,u)/X] =_F t[f/X]_u \tag{10}
$$

$$
A_1[(C,u)/X]; t_0[C/X]; A_2[(u,C)/X] =_F t[f/X]_l \tag{11}
$$

Consider $t^\star$ defined as in (2):

$$t^\star = \alpha_1 \dots \alpha_P.\,(z)\,T_1 \dots T_n$$

where each α_i is either of the form $\lambda x_j^{A_j}$ ($1 \le j \le p$) or ΛX_k ($1 \le k \le P - p$).

Then we set:

$$\mathcal{M}_{\Gamma,A}(t^*) = \alpha_1^B \dots \alpha_P^B.\ \begin{cases} (z)\,\mathcal{M}_{\Gamma,\Sigma_1}(T_1)\dots\mathcal{M}_{\Gamma,\Sigma_n}(T_n) & \text{if } z \notin \Gamma \\ (z)\,\mathcal{N}_{\Gamma,\Sigma_1}(T_1)\dots\mathcal{N}_{\Gamma,\Sigma_n}(T_n) & \text{if } z \in \Gamma \end{cases}$$

$$\mathcal{N}_{\Gamma,A}(t^*) = \vec{\alpha_1} \dots \vec{\alpha_P}.\ \begin{cases} (z)\,\mathcal{N}_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{N}_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \notin \Gamma \text{ and } \mathcal{V}(A) \neq X \\ (z)\,\mathcal{N}_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{N}_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}(A) \neq X \\ (f)\ (z)\,\mathcal{M}_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{M}_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \notin \Gamma \text{ and } \mathcal{V}(A) = X \\ (f)\ (z)\,\mathcal{N}_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{N}_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}(A) = X \end{cases}$$

where

- $\vec{\alpha_i} = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\vec{\alpha_i} = \lambda x_j^{D_j}$ with $D_j = A_j[(C,B)/X][B/X^*]$ if $\alpha_i = \lambda x_j^{A_j}$
- $\alpha_i^B = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\alpha_i^B = \lambda x_j^{E_j}$ with $E_j = A_j[B/X][B/X^\star]$ if $\alpha_i = \lambda x_j^{A_j}$
- for $1 \le i \le n$, $\Sigma_i = \Sigma_\Gamma^i(t^*)$.
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{M}_{\Gamma,A}(\{D\}) = \mathcal{N}_{\Gamma,A}(\{D\}) = \{D[B/X^\star]\}$ if $D \in \texttt{Types}$

$$\mathcal{M}'_{\Gamma,A}(t^*) = \alpha_1^C \dots \alpha_P^C.\ \begin{cases} (z)\,\mathcal{M}'_{\Gamma,\Sigma_1}(T_1)\dots\mathcal{M}'_{\Gamma,\Sigma_n}(T_n) & \text{if } z \notin \Gamma \\ (z)\,\mathcal{N}'_{\Gamma,\Sigma_1}(T_1)\dots\mathcal{N}'_{\Gamma,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t^*) \neq X \\ (f)\ (z)\,\mathcal{N}'_{\Gamma,\Sigma_1}(T_1)\dots\mathcal{N}'_{\Gamma,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t^*) = X \end{cases}$$

$$\mathcal{N}'_{\Gamma,A}(t^*) = \vec{\alpha_1} \dots \vec{\alpha_P}.\ \begin{cases} (z)\,\mathcal{M}'_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{M}'_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \notin \Gamma \\ (z)\,\mathcal{N}'_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{N}'_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t^*) \neq X \\ (f)\ (z)\,\mathcal{N}'_{\Gamma\cup\Delta,\Sigma_1}(T_1)\dots\mathcal{N}'_{\Gamma\cup\Delta,\Sigma_n}(T_n) & \text{if } z \in \Gamma \text{ and } \mathcal{V}_\Gamma(t^*) = X \end{cases}$$

where

- $\vec{\alpha_i} = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\vec{\alpha_i} = \lambda x_j^{D_j}$ with $D_j = A_j[(C,B)/X][C/X^*]$ if $\alpha_i = \lambda x_j^{A_j}$
- $\alpha_i^C = \Lambda X_k$ if $\alpha_i = \Lambda X_k$, and $\alpha_i^C = \lambda x_j^{E_j}$ with $E_j = A_j[C/X][C/X^\star]$ if $\alpha_i = \lambda x_j^{A_j}$
- for $1 \le i \le n$, $\Sigma_i = \Sigma_\Gamma^i(t^*)$.
- $\Delta = \{x_1 : A_1, \dots, x_p : A_p\}$
- $\mathcal{M}'_{\Gamma,A}(\{D\}) = \mathcal{N}'_{\Gamma,A}(\{D\}) = \{D[C/X^\star]\}$ if $D \in \texttt{Types}$

Figure 3. Constructions used in the proof of Theorem 2

PROOF: Let us first consider Equation (10), and set $t^\star = t\langle X^\star/X\rangle$. As in the proof of Theorem 1, we consider two mutually recursive functions, $\mathcal{M}_{\Gamma,A}(t^*)$ and $\mathcal{N}_{\Gamma,A}(t^*)$, the difference being that we work with t^* (which is is not necessarily well-typed), and that there is a new parameter $A \in \mathtt{Types} \cup \{\dagger\}$. These functions are defined on Figure 3.

The two equalities we have to prove are:

$$A_1[(u,B)/X]\ ;\ t[B/X] =_F \mathcal{M}_{\Gamma,A}(t^*) \qquad (12)$$

$$\mathcal{M}_{\Gamma,A}(t^*)\ ;\ A_2[(B,u)/X] =_F \mathcal{J}_{\Gamma,A}(t^*) \qquad (13)$$

with $\Gamma = \{x : A_1\}$ and $A = A_2$ (again we work with t instead of t_0 because $t =_F t_0$).

We start by proving (12): considering a term variable y not appearing in t and setting $t^B = t[B/X][y/x]$, we note that $A_1[(u,B)/X]\ ;\ t = t^B[(A_1 f B)x/y]$ and we proceed by induction on A_1:

- If $A_1 = \forall_1.D_1 \to \ldots \to D_k \to Z$ where each D_j has the form $D_j = \forall_{j+1}.A_1^j \to \ldots \to A_{p_j}^j \to Y_j$ (where each $\forall_i$ is a sequence of quantifiers), then we can write

$$x =_F \mathbf{\Lambda}_1 \lambda z_1^{\overline{D_1}} \ldots \lambda z_k^{\overline{D_k}}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{\widehat{A_1^1}} \ldots \lambda x_{1,p_1}^{\widehat{A_p^1}}.$$
$$(z_1)\mathbf{X}^2 x_{1,1}\ldots x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A_1^k}} \ldots$$
$$\lambda x_{k,p_k}^{\widehat{A_{p_k}^k}}.(z_k)\mathbf{X}^{k+1} x_{k,1}\ldots x_{k,p_k})$$

where, as before, $\mathbf{\Lambda}_i = \Lambda X_1^i \ldots \Lambda X_{N_i}^i$ and $\mathbf{X}^i = \{X_1^i\}\ldots\{X_{N_i}^i\}$. Thus

$$(A_1 f B)x =_F \mathbf{\Lambda}_1 \lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.$$
$$\lambda x_{1,1}^{\widehat{A_1^1}} \ldots \lambda x_{1,p_1}^{\widehat{A_p^1}}.(v_1)(z_1)\mathbf{X}^2(A_1^1 f B)x_{1,1}\ldots$$
$$(A_{p_1}^1 f B)x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A_1^k}} \ldots \lambda x_{k,p_k}^{\widehat{A_{p_k}^k}}.$$
$$(v_k)(z_k)\mathbf{X}^{k+1}(A_1^k f B)x_{k,1}\ldots(A_{p_k}^k f B)x_{k,p_k})$$

with $v_j = f$ if $Y_j = X$ and $v_j = \mathit{id}_{Y_j}$ otherwise. Finally, we obtain, for $\tilde{t} = t^B[(A_1 f B)x/y]$:

$$\tilde{t} =_F t^B[\lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(x)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{\widehat{A_1^1}} \ldots$$
$$\lambda x_{1,p_1}^{\widehat{A_p^1}}.(v_1)(z_1)\mathbf{X}^2 y_{1,1}\ldots y_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{\widehat{A_1^k}}$$
$$\ldots \lambda x_{k,p_k}^{\widehat{A_{p_k}^k}}.(v_k)(z_k)\mathbf{X}^{k+1} y_{k,1}\ldots y_{k,p_k})/y]$$
$$[(A_j^i f B)x_{i,j}/y_{i,j}]_{i,j}$$

where the $y_{i,j}$'s are chose fresh.

This gives us an inductive construction that is exactly equivalent to the definition of $\mathcal{M}_{\Gamma,A}(t)$ with $\Gamma = \{x : \widehat{A_1}\}$. Indeed, when constructing $\mathcal{M}_{\Gamma,A}(t^\star)$, any time we see x or x in t applied to a list of arguments $T_1 \ldots T_n$, we replace each argument T_i by $\mathcal{N}_{\Gamma,A}(T_i)$: that is, we apply (f) to this argument if $\mathcal{V}(D_i) = X$, otherwise we do nothing; this corresponds exactly to the application of v in the above formula. Then, all the variables that are bound in the head of T are added to the context Γ, so we will treat them like x: this corresponds to the substitution $[(A_i' f B)x_i/y_i]_i$ in the above formula.

- Every other form for the type A_2 is isomorphic to the above form, and the result remains true in these cases (we just have to move the positions of the $\mathbf{\Lambda}$'s and the $\mathbf{X}$'s in the formulas).

For the equality (13) we note that $\mathcal{M}_{\Gamma,A}(t)\ ;\ A_2[(B,u)/X] = (A_2 B f)\mathcal{F}_\Gamma(t)$ and we proceed by induction on A_2:

- If $A_2 = \forall_1.D_1 \to \ldots \to D_k \to Z$ where each D_j has the form $D_j = \forall_{j+1}.A_1^j \to \ldots \to A_{p_j}^j \to Y_j$, then for any term h of type $A_2[B/X]$ we can write

$$h =_F \mathbf{\Lambda}_1 \lambda z_1^{D_1[B/X]} \ldots \lambda z_k^{D_k[B/X]}.(h)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]}$$
$$\ldots \lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2 x_{1,1}\ldots x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.$$
$$\lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}.(z_k)\mathbf{X}^{k+1} x_{k,1}\ldots x_{k,p_k})$$

so that

$$(A_2 B f)h =_F \mathbf{\Lambda}_1 \lambda z_1^{\widehat{D_1}} \ldots \lambda z_k^{\widehat{D_k}}.(v)(h)\mathbf{X}^1(\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]}$$
$$\ldots \lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2(A_1^1 B f)x_{1,1}\ldots(A_{p_1}^1 B f)$$
$$x_{1,p_1})\ \ldots\ (\mathbf{\Lambda}_{k+1}.\lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}.$$
$$(z_k)\mathbf{X}^{k+1}(A_1^k B f)x_{k,1}\ldots(A_{p_k}^k B f)x_{k,p_k})$$

with $v = f$ if $Z = X$ and $v = \mathit{id}_Z$ otherwise. Now let us take $h = \mathcal{M}_{\Gamma,A}(t)$; we can rewrite it as

$$\mathcal{M}_{\Gamma,A}(t) =_F \Lambda Y_1 \ldots \Lambda Y_{N_1} \lambda y_1^{D_1[B/X]} \ldots \lambda y_k^{D_k[B/X]}.\ M$$

for some $M = (z)T_1 \ldots T_n$. If we note $M' = M[X_1^1/Y_1]\ldots[X_{N_1}^1/Y_{N_1}]$, we obtain, for $\breve{t} = (A_2 B f)\mathcal{M}_{\Gamma,A}(t)$:

$$\breve{t} =_F \mathbf{\Lambda}_1 \lambda z_1^{\widehat{D_1}} \ldots \lambda z_k^{\widehat{D_k}}.(v)\ M'[\mathbf{\Lambda}_2.\lambda x_{1,1}^{A_1^1[B/X]} \ldots$$
$$\lambda x_{1,p_1}^{A_p^1[B/X]}.(z_1)\mathbf{X}^2 y_{1,1}\ldots y_{1,p_1}/y_1]\ \ldots\ [\mathbf{\Lambda}_{k+1}.$$
$$\lambda x_{k,1}^{A_1^k[B/X]} \ldots \lambda x_{k,p_k}^{A_{p_k}^k[B/X]}.(z_k)\mathbf{X}^{k+1} y_{k,1}\ldots$$
$$y_{k,p_k}/y_k][(A_j^i B f)x_{i,j}/y_{i,j}]_{i,j}$$

The result of this construction is exactly the term $\mathcal{J}_{\Gamma,A}(t)$: indeed, the variables $y_1,\ldots,y_k$ and $y_{i,j}$ in M' are the places, in the definition of $\mathcal{M}_{\Gamma,A}(t)$, where we have used

$\mathcal{M}_{\Gamma,A}$ and not $\mathcal{N}_{\Gamma,A}$. Hence, with the application of v at those places, we retrieve $\mathcal{J}_{\Gamma,A}(t)$.

- Every other form for the type A_2 is isomorphic to the above form, and the result remains true in these cases (we just have to move the positions of the $\mathbf{\Lambda}$'s and the $\mathbf{X}$'s in the formulas).

It might be surprising to see the similarity between these two inductions and the proof of Theorem 1, whereas the present case is supposed to be more complicated. The reason for this is that, with the partial substitution $t\langle X^\star/X\rangle$, we transformed every instantiation by X into an instantiation by $X^\star$, so that we got rid of the instantiations by X. Thus, after this substitution, things are very similar to the case where $X \notin \mathcal{Z}(t)$.

The proof of Equation (11) is very similar to what we have just done, with two new functions $\mathcal{M}'_{\Gamma,A}(t^\star)$ and $\mathcal{N}'_{\Gamma,A}(t^\star)$ defined on Figure 3.

The two equalities we have to prove are in this case:

$$A_1[(C,u)/X]\ ;\ t[C/X] =_F \mathcal{M}'_{\Gamma,A}(t^*) \tag{14}$$

$$\mathcal{M}'_{\Gamma,A}(t^*)\ ;\ A_2[(u,C)/X] =_F \mathcal{K}_{\Gamma,A}(t^*) \tag{15}$$

with $\Gamma = \{x : A_1\}$ and $A = A_2$. $\square$