Strachey Parametricity and Game Semantics

Joachim de Lataillade

Abstract

We propose and analyse in this paper a criterion for Strachey parametricity, which is satisfied in particular by system F with its usual equality rules. The criterion is based on a simple restriction of the dinaturality diagram.

Game semantics is used as an illustration of this criterion, providing both a parametric model and a non-parametric one. But it is also as a tool to explain how the criterion copes with second-order quantification, and to characterise dinaturality as a local property.

We also see some limitations of the criterion, due to the requirement of totality we impose on strategies.

1 Introduction

1.1 Polymorphism and Strachey parametricity

In a given programming language, polymorphism is the possibility for the same algorithm to have many different types. This can take many flavours, but here we will be interested by the case where these types can be quantified: a term t of type $\forall X.A$ is such that, for any type B, t has the type $A[B/X]$, or generates a term of this type. We commonly see this type B as an input: the environment provides B, and it results in the term of type $A[B/X]$ generated from t.

Polymorphism appears mainly in two distinct forms[1]:

- **ad-hoc polymorphism**, where a term of type $\forall X.A$ will behave differently depending on the type B given as input
- **parametric polymorphism**, where the behaviour for all possible inputs B is given by the behaviour for an input X, the "parameter".

This distinction was first made by Christopher Strachey [Str00], who coined the terminology. Note that Strachey did not give a very precise statement of what parametric polymorphism is. One should have in mind that for a parametric term t of type $\forall X.A$, the input given for X is a *black box*, which can be used in the term but not looked inside. This notion will be referred to as **(intuitive) Strachey parametricity**.

We will see below some examples of parametric and non-parametric terms. But this notion is still intuitive, and our our goal in this article is to formulate and discuss a precise definition of

[1]Subtyping could be added to this list, but in the present article it will be considered as a case of ad hoc polymorphism.

Strachey parametricity. What we want then is a criterion, that would allow to discriminate between syntaxes that are intuitively Strachey parametric and those which are not.

In fact the question of parametricity is not only relevant for syntax: it can also apply to mathematical models. Indeed, if we have a model of a calculus with polymorphic types, one may ask if, intuitively, the behaviour of a term of type $\forall X.A$ will depend on the input given for X. Again this is still an intuitive notion that we want to characterise by a precise property. Fortunately, as the extension of a syntax can be seen in general as a model of that syntax, we will use the same criterion to characterise Strachey parametricity for both syntaxes and models which extends, or interpret, Church-style system F.

1.2 Strachey parametricity and system F

System F, invented independently by Reynolds [Rey74] and Girard [Gir72], is a prototype of a programming languages with polymorphic types. It essentially exists in two flavours:

- **Church-style** system F, where terms contain implicit type indications
- **Curry-style** system F, in which there are no explicit types appearing in the terms, so that the grammar of terms is just λ-calculus.

In this article we will focus mainly on Church-style system F, but the ideas we develop can successfully be applied to Curry-style system F.

Church style system F is the natural extension of the Curry-Howard correspondence to second-order logic. Its grammar of terms has the same constructors as the λ-calculus, plus two new operations: $t \mapsto \Lambda X.t$ and $t \mapsto t\{B\}$ where B is any type. They correspond respectively to the introduction and elimination of the universal quantifier in logic. The rules of system F are described on Fig. 1, where $X \notin FTV(t)$ (or $x \notin FV(t)$, or $X \notin FTV(\Gamma)$) means that X (or x) does not appear freely in t (or in Γ).

Church-style system F is then the perfect prototype of a parametric language: indeed, the axiom for introducing the quantifier exactly expresses that the input type has to be a parameter (a fresh variable in this case). And a term cannot look "inside" the parameter to adapt his behaviour. Consequently, it is essential that the criterion of parametricity that we will define is satisfied by the terms of system F. As we shall see, it is a requirement that usual formalisations of parametricity do not meet, which explains the motivation for this paper.

To understand parametricity better, one can also ask what a non-parametric term would be. The basic example of ad-hoc polymorphism being overloading, our example will come from this idea: let us suppose that we have a language (possibly extending system F), with an ML-like syntax, where the following term can be written:

```
let PLUS (x,y:X) = match X with
          | int -> x+y
          | float -> x+.y
          | list -> x@y
          | _ -> x
```

Grammars:

$$A ::= \quad X \mid \bot \mid A \to A \mid A \times A \mid \forall X.A$$
$$t ::= \quad x \mid \lambda x^A.t \mid (tt) \mid \langle t, t\rangle \mid \pi_1(t) \mid \pi_2(t) \mid \Lambda X.t \mid t\{A\}$$

Typing rules:

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

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

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

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u\rangle : A \times B} (\times I)$$

$$\frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_1(t) : A} (\times E1) \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_2(t) : B} (\times E2)$$

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

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

Equalities:

$$\begin{array}{rclll}
(\lambda x^A.t)u & = & t[u/x] & & (\beta) \\
\lambda x^A.tx & = & t & \text{if } x \notin FV(t) & (\eta) \\
\pi_1(\langle u, v\rangle) & = & u & & (\pi_1) \\
\pi_2(\langle u, v\rangle) & = & v & & (\pi_2) \\
\langle \pi_1(u), \pi_2(u)\rangle & = & u & & (\times) \\
(\Lambda X.t)\{B\} & = & t[B/X] & & (\beta 2) \\
\Lambda X.t\{X\} & = & t & \text{if } X \notin FTV(t) & (\eta 2)
\end{array}$$

Figure 1: Church-style system F

where `+` is the addition for integers, `+.` the addition for floats and `@` the concatenation of lists. This term if of type $X \times X \to X$ for any X, so it generates a term of type $\forall X.X \times X \to X$, which will of course not be parametric: the behaviour will depend on the value given for X. Hence, we want our criterion to be able to discriminate this kind of bad behaviour.

1.3 Relational parametricity

Probably the most well-known formalisation of parametricity is the notion of **relational parametricity** introduced by Reynolds [Rey83]. It originally came from a semantic study of second-order quantification, in the model **PER** of partial equivalence relations. Then followed a syntactic formalisation of Reynolds' ideas, first by Abadi, Cardelli and Curien [ACC93] and then by Plotkin and Abadi [PA93].

In the latter case, the notion of parametricity is defined by a logic of terms, types and relations between types (with system F being part of this logic), and by adding an axiom called **parametricity schema**:

$$\forall u : (\forall X.A[X]).\forall Y \forall Z.\forall \mathcal{R} \subseteq Y \times Z.u\{Y\}A[\mathcal{R}]u\{Z\}$$

This means the following: if two types Y and Z are related by $\mathcal{R}$ and u is of type $\forall X.A[X]$, then $u\{Y\}$ and $u\{Z\}$ are related by $A[\mathcal{R}]$. The latter relation may be an equality: this means that the equality of relational parametricity will be stronger than equality for system F.

There is a huge literature with many results on relational parametricity. In fact the term "parametricity" often refers directly to this particular formalisation of parametricity. That is why we chose to use the term **Strachey parametricity** for our criterion, to distinguish it from the more usual relational parametricity.

One great strength of relational parametricity is that it provides many nice structural features: for example, in a relational parametric model we have the following (as proved in [BFSS90]):

- $\forall X.X \to X$ is a terminal object
- $\forall X.(A \to B \to X) \to X$ is a product of A and B
- $\forall X.X$ is an initial object
- $\forall X.(A \to X) \to (B \to X) \to X$ is a coproduct of A and B.

These properties are only weakly true (weak terminal object, weak product, etc.) if we do not require relational parametricity.

Another positive point of the relational setting is that it can be used as a tool to prove properties on system F: the so-called "theorems for free"of Wadler [Wad89], as well as results on the observational behaviour of system F.

However, relational parametricity is not true for the Church-style system F ! This is easy to see: for example, the type $\forall X.X$ is not initial in the syntactic category, as there are at least two distinct morphisms from $\forall X.X$ to `Bool`:

$$\lambda x^{\forall X.X}.\mathtt{true} \qquad \lambda x^{\forall X.X}.\mathtt{false}$$

This is then a bad candidate for our criterion for Strachey parametricity: as we said before, system F being a prototype of a parametric language, a proposal characterisation of parametricity should be satisfied by this syntax.

But we stress the fact that it concerns only system F modulo the usual equalities given on Fig. 1. As far as observational behaviour is concerned, relational parametricity is definitely a meaningful and powerful notion.

1.4 Dinaturality

An alternative fruitful look at parametricity is given by the notion of **dinaturality**.

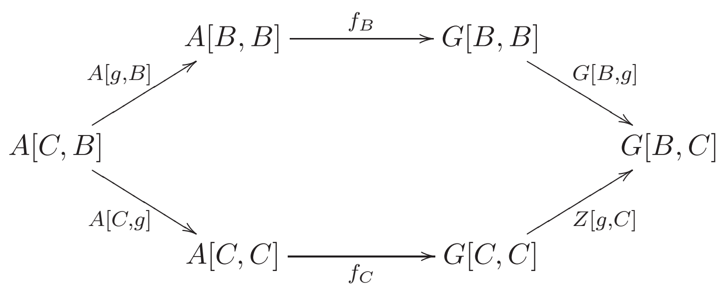
The idea is the following: given a type A with one free variable X, we would like to interpret A as a functor F_A, which sends each type B to $A[B/X]$ and each term t to an appropriate term. But it immediately appears that then F_A should be both covariant and contravariant: think about the type $A = X \to X$, if we see it as a functor it should be covariant on the right hand variable ($t \mapsto (X \to t)$ is covariant) and contravariant on the left hand variable ($t \mapsto (t \to X)$ is contravariant). So, one better solution is to ask the functor to take the form

$$F_A : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$$

where $\mathcal{C}$ is the category we want to work with. We note $F_A[x, y]$ the application of this functor to x and y (which can be objects or morphisms in $\mathcal{C}$).

What are then the morphisms between two of these functors ? The most obvious answer is to extend the notion of natural transformation, giving rise to **dinatural transformations**:

Definition 1 (dinatural transformation) *Let $\mathcal{C}$ be a category and $F, G : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ two functors, a* ***dinatural transformation*** *from F to G, denoted $f : F \ddot{\to} G$, is a family $\{f_A : FAA \to GAA\}$ of morphisms in $\mathcal{C}$ such that:*



for every morphism $g : B \to C$.

Note that we do not obtain a category with functors $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ as objects and dinatural transformations as morphisms, because dinatural transformations do not compose in general.

Dinaturality has been proposed as a setting for parametricity in [BFSS90, FGSS88]. In particular it has been connected with the **PER** model, with the resulting structural consequences listed in section 1.3. Furthermore, many "theorems for free" can be recovered as dinaturality properties, as shown in [GSS92].

But the dinaturality property is wrong in the case of Church-style (or even Curry-style) system F: in section 2.5 we will see what it precisely means, and why it is wrong. So it is not a good candidate either for our notion of Strachey parametricity.

However, it was proven in [GSS92] that the dinaturality diagram is commutative if we consider a language with only simple types, that is, without second-order quantification. Our proposal in the present paper is simply to turn this into our criterion, even in the case where second order appears in the calculus:

Strachey parametricity
=
dinaturality on simple types

There is something strange about this criterion: it put some constrains on simple types only, whereas our intuitive notion of Strachey parametricity is about the behaviour of the second order quantification. An important part of our work in the paper will be devoted to explain why this condition is nevertheless strong enough to ensure a parametric behaviour in a syntax, or in a model.

This condition is in fact rather strong, and we will see in section 5. So it might be seen in general as a sufficient but not necessary condition for parametricity. However, it has the important advantage, when compared to other known characterisations, to be true in the most standard case of Strachey parametricity, namely system F.

1.5 Content of the paper

As dinaturality was also proposed as a setting for parametricity, our tentative criterion is not a very original choice. But the point of the paper is to show how such a criterion can discriminate models that are intuitively parametric or not parametric, in particular with game models. Moreover, game models will provide a deep understanding of the property of dinaturality, by interpreting it as a local condition.

Thus, the contributions of the paper are the following:

(i) claim that dinaturality on simple types is a criterion for Strachey parametricity, and formalise this criterion in any hyperdoctrine (section **2**)

(ii) use game models to explain how dinaturality on simple types is related to intuitive parametricity: cf. sections **3.3** and **3.6**

(iii) provide two examples of game models (in sections **3** and **4**), and prove that one of them satisfies the criterion whereas the other does not

(iv) interpret dinaturality as a local property in games, which allows to see in which case it will be true and in which case it will not: see section **3.5**

(v) study the problem of totality and finiteness, and their equivalent in syntax: without these properties, the dinaturality property will fail in general; this will be discussed in section **5**.

2 Criterion for Strachey parametricity

In this section we propose a formal criterion to decide whether a model of system F is Strachey parametric or not. This criterion will be justified in the next sections through a study of the game models.

The definition of the criterion may look a bit formal at first sight. This formalisation should not be taken too seriously: it just expresses the idea that Strachey parametricity is dinaturality for simple types, in such a way that asking for this property will make sense in any hyperdoctrine.

2.1 Hyperdoctrine

We first recall the definition of an hyperdoctrine, in which we will define our criterion. This categorical notion, introduced by Lawvere [Law70] was used by Seely [See87] and Pitts [Pit88] to give a categorical interpretation of system F. As any model of system F can be seen as an hyperdoctrine, defining our criterion for hyperdoctrines will be the same as defining it for any model.

In what follows, **CCC** stands for the category of cartesian closed categories and strict functors. Also, given a cartesian category $\mathcal{C}$, and morphisms $f_i : A \to A_i$ for $1 \le i \le n$ in $\mathcal{C}$, we note $\langle f_1, \dots, f_n \rangle$ the morphism from A to $A_1 \times \cdots \times A_n$ generated by the cartesian structure.

Definition 2 (hyperdoctrine) *An* ***hyperdoctrine*** $\mathcal{H}$ *is specified by:*

(i) a cartesian category $\mathbb{H}$ *with a distinguished object* U *such that for every object* I *in* $\mathbb{H}$ *there exists* $n \in \mathbb{N}$ *such that* $I = U^n$*; we denote* $\pi_n^i : U^n \to U$ *the projection on the* i*th component, and* $\pi_{U^n} = \langle \pi_{n+1}^1, \dots, \pi_{n+1}^n \rangle : U^{n+1} \to U^n$

(ii) a functor $H : \mathbb{H}^{op} \to \textbf{CCC}$ *such that if we compose* H *with the forgetful functor* $\mathit{fff} : \textbf{CCC} \to \textbf{Set}$ *we obtain the functor* $\mathbb{H}(-, U)$

(iii) for each I *object in* $\mathbb{H}$*, a functor* $\Pi_I : H(I \times U) \to H(I)$ *such that :*

- Π_I *is right adjoint to the functor* $H(\pi_I) : H(I) \to H(I \times U)$
- Π_I *is natural in* I*: for any* $\mathbf{A} : I \to J$*,* $H(\mathbf{A}) \circ \Pi_J = \Pi_I \circ H(\mathbf{A} \times id_U)$
- *for any* $\mathbf{A} : I \to J$*, for any object* B *of* $H(J \times U)$*, the morphism from* $(H(\mathbf{A}) \circ \Pi_J)(B)$ *to* $(\Pi_I \circ H(\mathbf{A} \times id_U))(B)$ *generated by the adjunction is the identity.*

One could also consider a more general setting by forgetting the condition (iii). The definitions below would still hold in this case, and it would correspond to a syntax where we do not necessarily have a quantification, but still many atomic types.

For what follows, we will consider that any hyperdoctrine $\mathcal{H}$ comes with its base category $\mathbb{H}$ and the functor $H : \mathbb{H}^{op} \to \textbf{CCC}$. The morphisms in $\mathbb{H}$ will often be written in bold letters, as they will be considered as tuples (a morphism $\mathbf{A} : U^n \to U^p$ can be seen as a p-uple of morphisms $A_i : U^n \to U$).

2.2 Multivariance

An important notion for defining our criterion will be that of a **multivariance**. The idea is the same as what was exposed in section 1.4 : we want to have functors of the form

$$F_A : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$$

and we want to have them for many (not necessarily all) values of A.

We define the category $\overline{\textbf{Cat}}$ as follows:

- objects are categories
- a morphism from $\mathcal{C}$ to $\mathcal{D}$ is a functor $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{D}$, and we will denote the application of this functor by $F[x, y]$ where x (resp. y) is either an object or a morphism in $\mathcal{C}$ (resp. in $\mathcal{C}^{op}$)
- the composition of $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D}^{op} \times \mathcal{D} \to \mathcal{E}$ is given by

$$(G \circ F)[x, y] = G[F[y, x], F[x, y]]$$

- the identity $I_{\mathcal{C}} : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ is given by

$$I_{\mathcal{C}}[x, y] = y$$

Note that it is a cartesian category, with the same binary product as in **Cat**.

Definition 3 (multivariance) *A **multivariance** in an hyperdoctrine $\mathcal{H}$ is given by a strict cartesian functor $\mathcal{V}^I : \mathbb{H} \to \overline{\boldsymbol{Cat}}$ for each I object in $\mathbb{H}$, satisfying the following properties:*

- $\mathcal{V}^I(U^n) = H(I)^n$
- *for* $\mathbf{A} : J \to K$ *and* $\mathbf{B} : I \to J$ *in* $\mathbb{H}$ *we have*

$$\mathcal{V}^I(\mathbf{A})[\mathbf{B}, \mathbf{B}] = H(\mathbf{B})(\mathbf{A})$$

The second requirement of this definition does not look properly typed at first sight: if $\mathbf{B}$ is is a morphism in $\mathbb{H}$, we should not be able to apply $\mathcal{V}^I(\mathbf{A})$ to it. But one has to remember the equality

$$fff \circ H = \mathbb{H}(-, U)$$

If we consider $\mathbf{B} : I \to J$, we have $J = U^n$ for some n, so $\mathbf{B} = \langle B_1, \ldots, B_n \rangle$ with $B_i : I \to U$. Thus, each B_i can be seen as an object in $H(I)$, and $\mathbf{B}$ as an object in $H(I)^n$, so that $\mathcal{V}^I(\mathbf{A})[\mathbf{B}, \mathbf{B}]$ is well defined as an object in $H(I)^p$ if $K = U^p$.

It will not be always possible to define a multivariance in a given hyperdoctrine. So we allow the possibility of defining it only for a particular subcategory:

Definition 4 (restricted multivariance) *In an hyperdoctrine $\mathcal{H}$, if $\mathbb{K}$ is a cartesian subcategory of $\mathbb{H}$, then a **multivariance restricted to** $\mathbb{K}$ is given by a strict cartesian functor $\mathcal{V}^I : \mathbb{K} \to \overline{\boldsymbol{Cat}}$ for each I object in $\mathbb{H}$, with the properties:*

- $\mathcal{V}^I(U^n) = H(I)^n$
- *for* $\mathbf{A} : J \to K$ *in* $\mathbb{K}$ *and* $\mathbf{B} : I \to J$ *in* $\mathbb{H}$ *we have*

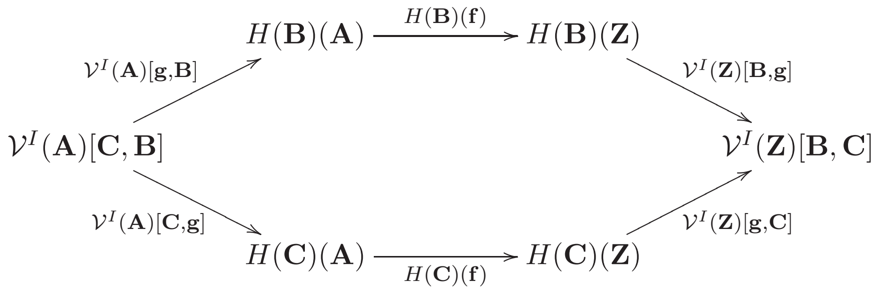
$$\mathcal{V}^I(\mathbf{A})[\mathbf{B}, \mathbf{B}] = H(\mathbf{B})(\mathbf{A})$$

2.3 Dinaturality

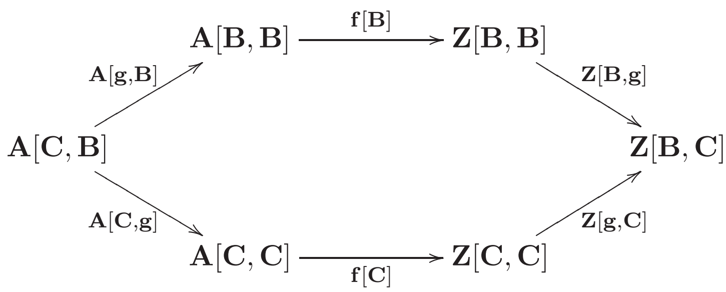
Given $\mathbf{A} = \langle A_1, \ldots, A_p \rangle : U^n \to U^p$ and $\mathbf{Z} = \langle Z_1, \ldots, Z_p \rangle : U^n \to U^p$ in $\mathbb{K}$, a morphism from $A_1 \times \cdots \times A_p$ to $Z_1 \times \cdots \times Z_p$ in $H(U^n)^p$ will take the form $\mathbf{f} = (f_1, \ldots, f_p)$ with $f_i : A_i \to Z_i$. In this case, we simply note $\mathbf{f} : \mathbf{A} \to \mathbf{Z}$. We also note $H(\mathbf{B})(\mathbf{A}) = H(\mathbf{B})(A_1) \times \cdots \times H(B)(A_p)$ and $H(\mathbf{B})(\mathbf{f}) = (H(\mathbf{B})(f_1), \ldots, H(\mathbf{B})(f_p))$ for $\mathbf{B} : U^k \to U^n$. Finally, we will note $\mathcal{V}^I(\mathbf{A})[\mathbf{B}, \mathbf{g}]$ (resp. $\mathcal{V}^I(\mathbf{A})[\mathbf{g}, \mathbf{B}]$) for $\mathcal{V}^I(\mathbf{A})[id_{\mathbf{B}}, \mathbf{g}]$ (resp. $\mathcal{V}^I(\mathbf{A})[\mathbf{g}, id_{\mathbf{B}}]$).

Definition 5 (dinaturality) *In an hyperdoctrine $\mathcal{H}$, let $\mathbb{K}$ be a cartesian subcategory of $\mathbb{H}$ and $(\mathcal{V}^I : \mathbb{K} \to \overline{\boldsymbol{Cat}})_I$ a multivariance restricted to $\mathbb{K}$.*

*We say that the multivariance $(\mathcal{V}^I)_I$ is **dinatural** if, for every $\mathbf{A}, \mathbf{Z} : J \to K$ in $\mathbb{K}$ and $\mathbf{f} : \mathbf{A} \to \mathbf{Z}$, and given $\mathbf{B}, \mathbf{C} : J \to I$ and $\mathbf{g} : \mathbf{B} \to \mathbf{C}$, the following diagram commutes:*



It is easier to understand this diagram with more standard notations: if we note $H(\mathbf{B})(x) = x[\mathbf{B}]$ and $\mathcal{V}^I(\mathbf{A})[x, y] = \mathbf{A}[x, y]$, and use the equality $\mathcal{V}^I(\mathbf{A})[\mathbf{B}, \mathbf{B}] = H(\mathbf{B})(\mathbf{A})$, we obtain:



which is the usual dinaturality diagram. In general we will only prove this diagram for $\mathbf{A} = \langle A_1 \rangle$ and $\mathbf{f} = f_1$. Thanks to the properties of the cartesian product it will imply that the diagram is true in the general case.

2.4 Simple objects and Strachey parametricity

There is not always a multivariance in a hyperdoctrine, so dinaturality cannot be stated in general. However, a restricted multivariance exists for objects that correspond to simple types in syntax, and that we just call **simple objects**.

Because of the equality $\mathit{fff} \circ H = \mathbb{H}(-, U)$, each projection $\pi_n^i : U^n \to U$ can be seen as an object in $H(U^n)$. This gives sense to the following definition:

Definition 6 (simple objects) *Given an hyperdoctrine $\mathcal{H}$, the **simple objects** in $H(I)$, with $I = U^n$, are the objects of $H(I)$ generated by π_n^i for $1 \leq i \leq n$, the product and the exponentiation.*

The category of simple objects of $\mathcal{H}$ is the subcategory of $\mathbb{H}$, full on objects, whose morphisms from U^n to U^p are $\langle A_1, \ldots, A_p \rangle$ where each A_i is a simple object in $H(U^n)$.

Proposition 1 *In any hyperdoctrine $\mathcal{H}$, there exists a multivariance restricted to the simple objects of $\mathbb{H}$, called the **standard multivariance** on simple objects.*

PROOF: Let $\mathbb{K}$ be the category of simple objects of $\mathcal{H}$. For any object I in $\mathbb{H}$, we define the multivariance $\mathcal{V}^I : \mathbb{K} \to \overline{\mathbf{Cat}}$ as follows:

- if $A : U^n \to U$ is a simple object, then we proceed by induction:

$$\mathcal{V}^I(\pi_n^i)[x, y] = y_i$$
$$\mathcal{V}^I(A \times B)[x, y] = \mathcal{V}^I(A)[x, y] \times \mathcal{V}^I(B)[x, y]$$
$$\mathcal{V}^I(A \to B)[x, y] = \mathcal{V}^I(A)[y, x] \to \mathcal{V}^I(B)[x, y]$$

  where x, y are either objects or morphisms in $H(I)^n$, and y_i is obtained by applying to y the projection on the ith component

- if $\mathbf{A} = \langle A_1, \ldots, A_p\rangle : U^n \to U^p$ with $A_1, \ldots, A_p$ simple objects, then

$$\mathcal{V}^I(\mathbf{A}) = \langle \mathcal{V}^I(A_1), \ldots, \mathcal{V}^I(A_p)\rangle$$

Note that there is not much choice in the way that $\mathcal{V}^I$ if defined: if, for example, we could not set

$$\mathcal{V}^I(A \to B)[x, y] = \mathcal{V}^I(A)[x, y] \to \mathcal{V}^I(B)[x, y]$$

because the functor $(A, B) \mapsto (A \to B)$ is contravariant in A, so it has to switch the positions of x and y. □

Thanks to this multivariance, we are finally able to define our criterion for Strachey parametricity:

Definition 7 (Strachey parametricity) *The hyperdoctrine $\mathcal{H}$ is called **Strachey parametric** if the standard multivariance on simple objects is dinatural.*

2.5 The case of system F

To understand our definition of Strachey parametricity, let us see what happens in the case of the syntactic hyperdoctrine of system F.

First we describe this hyperdoctrine: the base category $\mathbb{H}$ has as objects natural numbers, and a morphism from n to m is an m-uple of second order types $\langle A_1, \ldots, A_m\rangle$ whose free type variables are among $X_1, \ldots, X_n$. Composition in $\mathbb{H}$ is substitution: if $\mathbf{A} = \langle A_1, \ldots, A_m\rangle : n \to m$ and $\mathbf{B} = \langle B_1, \ldots, B_n\rangle : p \to n$ then $\mathbf{A} \circ \mathbf{B} = \langle A_1[\mathbf{B}/X], \ldots, A_m[\mathbf{B}/X]\rangle : p \to m$ where $A_i[\mathbf{B}/X]$ stands for $A_i[B_1/X_1, \ldots, B_n/X_n]$.

The functor $H : \mathbb{H}^{op} \to \mathbf{CCC}$ is defined as follows:

- for each $n \in \mathbb{N}$, $H(n)$ has as objects second order types whose free type variables are among $X_1, \ldots, X_n$, and a morphism from A to B is a term t whose free type variables are among $X_1, \ldots, X_n$ and such that $x : A \vdash t : B$, up to the equalities of the language (otherwise said, it is an equivalence class of such terms modulo the equalities)
- for each $\mathbf{A} = \langle A_1, \ldots, A_m\rangle : n \to m$, $H(\mathbf{A})$ is the functor which associates to each type B in $H(m)$ the type $B[A_1/X_1, \ldots, A_m/X_m]$ in $H(n)$.

Finally the functor $\Pi_n : H(n+1) \to H(n)$ is defined by:

- $\Pi_n(A) = \forall X_{n+1}.A$
- if σ is the equivalence class of the term $x : A \vdash t : B$ then $\Pi_n(\sigma)$ is the equivalence class of the term $y : \forall X_{n+1}.A \vdash \Lambda X_{n+1}.(t[y\{X_{n+1}\}/x]) : \forall X_{n+1}.B$

As well as system F is the paradigmatic example of a language satisfying Strachey parametricity, this syntactic hyperdoctrine has to be a paradigmatic example of a Strachey parametric hyperdoctrine. To understand why it is the case, let us define a multivariance in this hyperdoctrine as follows:

- for a given type $A : n \to 1$, we proceed by induction:

$$\begin{aligned}
\mathcal{V}^k(\bot)[x,y] &= i_\bot(y) \\
\mathcal{V}^k(X_i)[x,y] &= y_i \\
\mathcal{V}^k(A \times B)[x,y] &= \mathcal{V}^k(A)[x,y] \times \mathcal{V}^k(B)[x,y] \\
\mathcal{V}^k(A \to B)[x,y] &= \mathcal{V}^k(A)[y,x] \to \mathcal{V}^k(B)[x,y] \\
\mathcal{V}^k(\forall X_n.A)[x,y] &= \Pi_n(\mathcal{V}^k(A)[\langle x, X_n\rangle, \langle y, X_n\rangle])
\end{aligned}$$

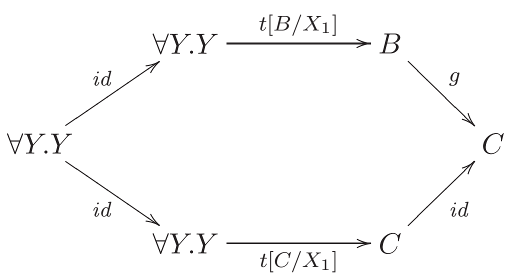
  where x, y are n-uples of either types or (equivalence classes of) terms[2] in $H(n)$, $i_\bot(y)$ is either $\bot$ if y is a tuple of types, or $id_\bot$ if y is a tuple of terms, y_i is the ith component of y, and $\langle x, X_n\rangle$ is the n-uple whose $n-1$ component are the same as in x and the last component is X_n.

- if $\mathbf{A} = \langle A_1, \ldots, A_p\rangle : n \to p$, then

$$\mathcal{V}^k(\mathbf{A}) = \langle \mathcal{V}^k(A_1), \ldots, \mathcal{V}^k(A_p)\rangle$$

This can be stated more simply by saying that $\mathcal{V}^k(A)[x,y]$ is the same as A except that every positive occurrence of X_i has been replaced by y_i, and every negative occurrence of X_i has been replaced by x_i; again, we note it $A[x,y]$. This multivariance, that we call **second order multivariance**, obviously extends the standard multivariance of the hyperdoctrine. So if we can prove that the second order multivariance is dinatural then we are done.

But the interesting point is that it is precisely not true: in system F, the dinaturality diagram is not commutative in general ! To see that, simply consider the types $A = \forall Y.Y$ and $Z = X_1$, and the term $t = x\{X_1\}$ such that $x : A \vdash t : Z$. Then the dinaturality diagram would write

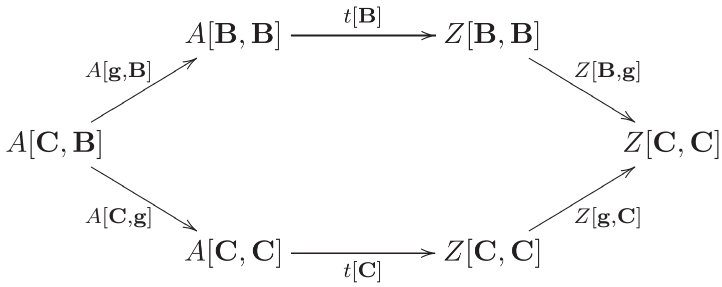


[2]It is safe to mix types and terms here because all the required constructions (like $A \to t$, $t \times A \ldots$) are easily defined in system F.

for any $g : B \to C$, which would imply that $g(x\{B\}) = x\{C\}$. And this is of course wrong in general for (Church-style, or even Curry-style) system F. Note that on this example it is easy to understand why dinaturality would become true in the case of Reynolds parametric interpretation: in that case, $g(x\{B\}) = x\{C\} : (\forall Y.Y) \to C$ because $\forall Y.Y$ is initial[3].

So, to prove that our syntactic hyperdoctrine is Strachey parametric, the solution is to show that the restriction of the multivariance to simple objects is dinatural. This is in fact the origin of our formal definition of Strachey parametricity: as dinaturality is a satisfying property but is not true for all second order types in system F, we restrict it to the cases where it is true. As we shall see with the game model, imposing this property just for simple objects will in fact generate big constraints on the behaviour of the second order quantifier.

In the case of our syntactic category, simple objects are exactly simple types, i.e. types generated by the types variables X_i for $i \in \mathbb{N}$ and the connectors $\times, \to$. So the property of dinaturality is expressed by the following diagram:



where A, Z are simple types with free variables among $X_1, \ldots X_n$, t is a term such that $x : A \vdash t : Z$, $\mathbf{B} = \langle B_1, \ldots B_n \rangle$ and $\mathbf{C} = \langle C_1, \ldots C_n \rangle$ are n-uples of types, $\mathbf{g} = (g_1, \ldots, g_n)$ is a n-uple of terms such that $x : B_i \vdash g_i : C_i$, and $t[\mathbf{B}]$ stands for $t[B_1, /X_1, \ldots, B_n/X_n]$ (and similarly for $t[\mathbf{C}]$).

This property has been proved in [GSS92] (although with a slightly different formulation). We shall see later an interpretation through games that leads to the same result.

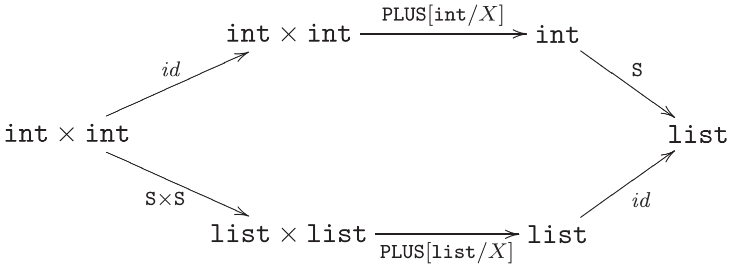
We can now conclude, as we originally wanted, that the syntactic category for system F is Strachey parametric. This is a formalised way to express the fact that system F satisfies the intuitive notion of Strachey parametricity. As far as we know, it is the first proposal for a parametricity criterion which holds for system F.

Finally, we shall see through an example how our criterion allows to discriminate non-parametric syntaxes. Imagine that, in a given language, we are given the term `PLUS` we introduced before:

```
let PLUS (x,y:X) = match X with
     | int -> x+y
     | float -> x+.y
     | list -> x@y
     | _ -> x
```

[3]This can also be understood in terms of observational behaviour of system F: if we consider the equality up to observational behaviour, then $g(x\{B\})$ and $x\{C\}$ have to be equal because they cannot be applied to any term.

If we choose the function `S : int → list` which associate to an integer `n` the singleton list `[n]`, then dinaturality would give us the following diagram:



But if we take the pair `(2,3)` as input, then the upper side will output the list `[5]` whereas the lower side will output the list `[2;3]`, which makes the diagram not commutative. This means that the language, no matter how it is defined exactly, will not be Strachey parametric. Thus, our criterion is able to distinguish between parametric (P) and non-parametric (NP) syntaxes[4]. How does this distinction work for semantics ? We shall now see it through two game models.

3 The uniform game model

The criterion for Strachey parametricity we have introduced still has to be justified. Indeed, it is a rather strange condition, as it constrains the behaviour of simply-typed terms whereas, intuitively, Strachey parametricity is about second order quantification. To understand why the criterion is accurate, we will see two examples of semantics, both using games.

The first setting in which we will develop our formalisation of Strachey parametricity is the game model for system F as it was developed in [dL08a] and [dL07]. Many ideas of the model come from previous works by Hughes [Hug00], Murawski-Ong [MO01] and Abramsky-Jagadesaan [AJ03].

As we shall see, in this model, the interpretation of simple types, which is dinatural, will constrain the morphisms of the model to satisfy a condition of **uniformity**, which is a straightforward translation of the intuitive notion of Strachey parametricity we gave in the introduction. This will give us a first link between our criterion and the intuitive notion of parametricity.

The presentation of the model requires many definitions and properties to check. We will give all the definitions but be rather elliptic concerning the proofs, as long as they are not central to our problem; complete proofs can be found in [dL07].

3.1 Arenas

In both the uniform and non-uniform game model, types will be interpreted as **arenas**. It would be possible to work directly on formulas, but we want to keep a setting closer to usual HO-games [HO00]. This will give us in particular a closer relationship between arenas and moves on these arenas.

We define the following grammar of **occurrences**:

$$a ::= \uparrow a \mid \downarrow a \mid \; ra \mid \; la \mid \; \star a \mid \; j \qquad (j \in \mathbb{N})$$

The set of all occurrences is denoted $\mathbb{A}$.

[4] Which proves that $P \neq NP$. Doesn't it ?

It is important to understand the relation between this grammar and the logic connectors: the symbols $\uparrow$ and $\downarrow$ correspond respectively to the right-side and the left-side of an arrow, the symbols r and l correspond respectively to the right-side and the left-side of a product, the symbol $\star$ corresponds to a quantifier and the index j stands for a type variable (free if $j \neq 0$, bounded otherwise) or a unit atomic type ($j = 0$ in this case).

An occurrence $a \in \mathbb{A}$ being a word on the above grammar, ended by an index $j \in \mathbb{N}$, we note $\sharp(a)$ the value of this index. We also define an operation of **juxtaposition** $a[a']$ between two occurrences as follows:

- $j[a'] = a'$
- $(\alpha a)[a'] = \alpha(a[a'])$ for $\alpha \in \{\uparrow, \downarrow, r, l, \star\}$.

We say that a is a **prefix** of a' if there exists $b \in \mathbb{A}$ such that $a' = a[b]$. This is denoted $a \sqsubseteq a'$. Finally, we set $\lambda(a) = \mathbf{O}$ (resp. $\lambda(a) = \mathbf{P}$) if a contains an even number (resp. an odd number) of symbols $\downarrow$.

Definition 8 (arena) *An* ***arena*** $A = (\mathcal{O}_A, \mathcal{L}_A)$ *is defined by a finite set* $\mathcal{O}_A \subseteq \mathbb{A}$ *and a function of* ***linkage*** $\mathcal{L}_A : \mathcal{O}_A \to \mathbb{A} \cup \{\dagger\}$ *satisfying the following conditions:*

- $\mathcal{O}_A$ *is* ***non-ambiguous****:* $\forall a, a' \in \mathcal{O}_A$*, if* $a \sqsubseteq a'$ *then* $a = a'$
- $\mathcal{L}_A$ *is* ***valid****: for every* $a \in \mathcal{O}_A$*, either* $\mathcal{L}_A(a) = \dagger$ *or* $\mathcal{L}_A(a) = a'[\star 0] \sqsubseteq a$ *for some* $a' \in \mathbb{A}$
- $\mathcal{L}_A$ *is* ***zero-valued****: for every* $a \in \mathcal{O}_A$*, if* $\sharp(a) \neq 0$ *then* $\mathcal{L}_A(a) = \dagger$*.*

The set of arenas is denoted $\mathcal{A}$*.*

The idea behind this definition is that $\mathcal{O}_A$ will represent the occurrences of type variables (or of unit atomic types) in a given type, and $\mathcal{L}_A$ will be the binding function for second-order quantifiers.

We define the following constructions on arenas:

- **(atoms)** $X_i = (\{i\}, i \mapsto \dagger)$ for every $i > 0$ $\quad \bot = (\{0\}, 0 \mapsto \dagger) \quad \top = (\emptyset, \emptyset)$
- **(product)** if $A, B \in \mathcal{A}$, we define $A \times B$ by:
  - $\mathcal{O}_{A\times B} = \{la \mid \ a \in \mathcal{O}_A\} \cup \{rb \mid \ b \in \mathcal{O}_B\}$
  - $\mathcal{L}_{A\times B}(la) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ l\mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{A\times B}(rb) = \begin{cases} \dagger & \text{if } \mathcal{L}_B(b) = \dagger \\ r\mathcal{L}_B(b) & \text{otherwise} \end{cases}$
- **(arrow)** if $A, B \in \mathcal{A}$, we define $A \to B$ by:
  - $\mathcal{O}_{A\to B} = \{\downarrow a \mid \ a \in \mathcal{O}_A\} \cup \{\uparrow b \mid \ b \in \mathcal{O}_B\}$
  - $\mathcal{L}_{A\to B}(\downarrow a) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ \downarrow \mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{A\to B}(\uparrow b) = \begin{cases} \dagger & \text{if } \mathcal{L}_B(b) = \dagger \\ \uparrow \mathcal{L}_B(b) & \text{otherwise} \end{cases}$
- **(quantification)** if $A \in \mathcal{A}$ and $i > 0$, we define $\forall X_i.A$ by:

- $\mathcal{O}_{\forall X_i.A} = \{\star a \mid \ a \in \mathcal{O}_A \wedge \sharp(a) \neq i\} \cup \{\star a[0] \mid \ a \in \mathcal{O}_A \wedge \sharp(a) = i\}$
- $\mathcal{L}_{\forall X_i.A}(\star a) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ \star\mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{\forall X_i.A}(\star a[0]) = \star 0$

This gives rise to an inductive interpretation of formulas. We also define an operation of substitution on arenas:

Definition 9 (substitution) *Let $A, B \in \mathcal{A}$ and $i > 0$. The **substitution** of B for X_i in A is the arena $A[B/X_i]$ defined by:*

$$\mathcal{O}_{A[B/X_i]} = \{a \in \mathcal{O}_A \mid \ \sharp(a) \neq i\} \cup \{a[b] \mid \ a \in \mathcal{O}_A \wedge \sharp(a) = i \wedge b \in \mathcal{O}_B\}$$

$$\mathcal{L}_{A[B/X_i]}(a) = \mathcal{L}_A(a) \ \textit{and} \ \mathcal{L}_{A[B/X_i]}(a[b]) = \begin{cases} \dagger & \textit{if } \mathcal{L}_B(b) = \dagger \\ a[\mathcal{L}_B(b)] & \textit{otherwise} \end{cases}$$

Another construction that will be of interest in our context is the double substitution, which will allow to define a multivariance in games:

Definition 10 (double substitution) *Let $A, B \in \mathcal{A}$ and $i > 0$. The **double substitution** of X_i by (B, C) in A is the arena $A[(B, C)/X_i]$ defined by:*

$$\begin{array}{rcl} \mathcal{O}_{A[(B,C)/X_i]} & = & \{a \in \mathcal{O}_A \mid \ \sharp(a) \neq i\} \\ & & \cup\{a[c] \mid \ a \in \mathcal{O}_A \wedge \sharp(a) = i \wedge \lambda(a) = \mathbf{O} \wedge c \in \mathcal{O}_C\} \\ & & \cup\{a[b] \mid \ a \in \mathcal{O}_A \wedge \sharp(a) = i \wedge \lambda(a) = \mathbf{P} \wedge b \in \mathcal{O}_B\} \end{array}$$

$$\mathcal{L}_{A[(B,C)/X_i]}(a) = \mathcal{L}_A(a)$$

$$\mathcal{L}_{A[(B,C)/X_i]}(a[c]) = \begin{cases} \dagger & \textit{if } \mathcal{L}_C(c) = \dagger \\ a[\mathcal{L}_C(c)] & \textit{otherwise} \end{cases}$$

$$\mathcal{L}_{A[(B,C)/X_i]}(a[b]) = \begin{cases} \dagger & \textit{if } \mathcal{L}_B(b) = \dagger \\ a[\mathcal{L}_B(b)] & \textit{otherwise} \end{cases}$$

3.2 Moves and strategies

3.2.1 Moves on an arena

We introduce the grammar of **moves**:

$$m ::= \uparrow m \mid \downarrow m \mid \ rm \mid \ lm \mid \ \star^B m \mid \ j \qquad (B \in \mathcal{A}, j \in \mathbb{N})$$

These moves form the set $\mathbb{M}$. For the moment they are defined in general, but we wish to be able to specify when a given move can be played on a given arena.

On moves we define again an operation of juxtaposition $m[m']$ as follows:

- $j[a'] = a'$
- $(\alpha a)[a'] = \alpha(a[a'])$ for $\alpha \in \{\uparrow, \downarrow, r, l\} \cup \{\star^B \mid \ B \in \mathcal{A}\}$.

We also have an operation of **anonymity** $\partial : \mathbb{M} \to \mathbb{A}$ erasing the arenas indications in a move:

- $\partial(i) = i$ for $i \geq 0$
- $\partial(\star^A m) = \star\partial(m)$
- $\partial(\alpha m) = \alpha\partial(m)$ if $\alpha \in \{\uparrow, \downarrow, r, l\}$.

Finally, given $m \in \mathbb{M}$ and $a \in \mathbb{A}$, we will need the partially-defined **formula extraction** $\frac{m}{a}$ given by:

- $\frac{\star^B m}{\star 0} = B$
- if $\frac{m}{a}$ is defined, $\frac{\star^B m}{\star a} = \frac{\alpha m}{\alpha a} = \frac{m}{a}$.

Given all this, we are able to express when a move is played in a given arena:

Definition 11 (moves on an arena) *Let A be an arena. Its set of moves $\mathcal{M}_A \subseteq \mathbb{M}$ is given by defining the relation $m \in \mathcal{M}_A$ by induction on m: $m \in \mathcal{M}_A$ if*

- *either $\partial(m) = a \in \mathcal{O}_A$ and $\mathcal{L}_A(a) = \dagger$*
- *or $m = m_1[m_2]$ where $\partial(m_1) = a \in \mathcal{O}_A$, $\mathcal{L}_A(a) \neq \dagger$ and $m_2 \in \mathcal{M}_B$ with $B = \frac{m_1}{\mathcal{L}_A(a)}$.*

According to this definition, a move can take two forms: either it directly comes from an occurrence in $\mathcal{O}_A$, but with some arenas indications, or it comes from a juxtaposition: one part comes from an occurrence in $\mathcal{O}_A$ with arenas indications, the other part is a move in B, where B was an arena given in the first part of the move (the exact place of this arena being given by $\mathcal{L}_A$). This means that either we play directly with the set of occurrences $\mathcal{O}_A$ as if there was no quantification, or we play one part in this set, and one part inside the arenas that have been played to instantiate quantifiers.

3.2.2 Enabling relation

Now that we have defined arenas together with their moves, we are close to be able to define a game model in the style of Hyland and Ong [HO00]. What we still need is an **enabling relation** between moves[5], which will allow to introduce the justification pointers.

The **depth** of a move m is the number of occurrences of the symbol $\downarrow$ in this move. Depths 0 moves are **initial** moves, and we note $\vdash m$ if m is initial. We define an enabling relation $m \vdash n$ between a move m of depth d and a move n of depth $d+1$ through the following recursion:

- if $\vdash m$ and $\vdash n$ then $\uparrow m \vdash \downarrow n$
- if $m \vdash n$ then $\alpha m \vdash \alpha n$ for $\alpha \in \{\uparrow, \downarrow, r, l\} \cup \{\star^B \mid B \in \mathcal{A}\}$.

The transitive closure of $\vdash$ is denoted $\vdash^+$.

[5]The present definition of enabling was proposed by Dominic Hughes.

3.2.3 Strategies

We now have the whole setting for an HO-style game semantics (see [HO00, McC96]): arenas with their sets of moves, a notion of initial moves and an enabling relation between moves. So we could directly say that we define strategies, innocence and composition in the same way as in HO-games, our arenas being just a particular case of HO-arenas. We will however quickly recall the main definitions, as we will use them later in our study of parametricity and dinaturality in games.

Definition 12 (justified sequence, play) *A **justified sequence** on an arena A is a sequence $s = m_1 \dots m_n$ of moves on A, together with a partial function $f : \{1, \dots, n\} \rightharpoonup \{1, \dots, n\}$ such that: if $f(i)$ is not defined then $\vdash m_i$, and if $f(i) = j$ then $j < i$ and $m_j \vdash m_i$: in this case we say that m_j **justifies** m_i, or that there is a **pointer** from m_j to m_i.*

*A **play** on A is a justified sequence $s = m_1 \dots m_n$ on A such that, for every $1 \leq i \leq n-1$:*

- *if $\lambda(m_i) = \mathbf{P}$ then $\lambda(m_{i+1}) = \mathbf{O}$*
- *if $\lambda(m_i) = \mathbf{O}$ then $\lambda(m_{i+1}) = \mathbf{P}$ and $\sharp(m_i) = \sharp(m_{i+1})$.*

Note the condition $\sharp(m_i) = \sharp(m_{i+1})$ if $\lambda(m_i) = \mathbf{O}$ in the definition of a play: this is an important requirement, that will make it possible later to define copycat expansion, as well as expansions along a strategy.

Given two justified sequences s and t we say that s is a **presequence** of t if $t = sm_1 \dots m_p$.

A useful notion related to our second order games is the **importation** of an arena.

Definition 13 (importation) *If a move m in a play s on A contains the symbol $\star^B$, then it can be written $m = m_0[\star^B m_1]$. We say that B is **imported by** $\lambda(m_0)$ **at** m if m_1 does not contain the symbol $\downarrow$.*

We note $\mathcal{A}(s)$ the set of arenas imported during a play s, and $FTV(s) = \{FTV(A) \mid A \in \mathcal{A}(s)\}$.

Definition 14 (strategy on an arena) *A **strategy** σ on an arena A, denoted $\sigma : A$, is a non-empty set of even-lengths plays on A which is closed by even-length prefix and deterministic: if sm and sn are two plays of σ then $sm = sn$.*

Definition 15 (view, bi-view) *A **view** on A is a play $s = m_1 \dots m_n$ on A such that, for every $2 \leq i \leq n-1$, if $\lambda(m_i) = \mathbf{P}$ (i.e. i is even) then m_{i+1} is justified by m_i.*

*A **bi-view** on A is a justified sequence $s = m_1 \dots m_n$ on A such that, for every $2 \leq i \leq n-1$, m_{i+1} is justified by m_i.*

Definition 16 (innocence) *To each play s on an arena A, one can associate a view $\ulcorner s \urcorner$ by:*

- $\ulcorner \epsilon \urcorner = \epsilon$
- $\ulcorner sm \urcorner = \ulcorner s \urcorner m$ *if* $\lambda(m) = \mathbf{P}$
- $\ulcorner sm \urcorner = m$ *if* $\vdash m$
- $\ulcorner smtn \urcorner = \ulcorner s \urcorner mn$ *if* $\lambda(n) = \mathbf{O}$ *and m justifies n.*

*A strategy $\sigma : A$ is called **innocent** if, for any play sn in σ, the move that justifies n is in $\ulcorner s \urcorner$, and if we have: for any $smn \in \sigma$, $t \in \sigma$, if tm is a play in A and $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ then $tmn \in \sigma$.*

An innocent strategy is then given only by its views[6]. Then, for a strategy σ, we define $\ulcorner \sigma \urcorner$ as its set of views. And, given a set S of even-length views, we can define the strategy $\exp(S)$ as the smallest innocent strategy containing S, provided that we have the property of **compatibility** on S: if $s_1 m, s_2 n \in S$ with s_1 presequence of s_2 then $s_1 m$ is a presequence of $s_2 n$.

3.2.4 Composition

Finally, we explain the concept of interaction, which is fundamental in game semantics as it explains how strategies compose. Like in [dL08a], we present this process, and especially the notion of **restriction**, in a very formal way that might look a bit heavy at first sight. Nevertheless, we argue that this formalism has many advantages when manipulating sequences: for example, when considering a play s on $A \to B$, the projections on A will be $s\restriction_{\downarrow}$ and the projection on B will be $s\restriction_{\uparrow}$. This is both formally exact and very convenient as it allows to manipulate the sequence while forgetting everything about A and B.

Definition 17 (shape, restriction) *Let Σ be a finite set of words over the alphabet $\{\uparrow, \downarrow, r, l\}$. A move m is said to be of **shape** Σ if there exists $\zeta \in \Sigma$ such that $\zeta 0 \sqsubseteq m$. A justified sequence is of shape Σ if each of its moves is of shape Σ, and a strategy is of shape Σ if each of its plays is of shape Σ.*

*Let s be a justified sequence on an arena A. If $\Sigma = \{\zeta, \xi\}$ with $\zeta 0 \vdash^+ \xi 0$, consider the subsequence s' of s which contains only the moves of shape Σ. Then the **restriction** of s to Σ, denoted $s\restriction_{\zeta,\xi}$, is the sequence s' where each prefix ζ has been replaced by $\uparrow$, and each prefix ξ has been replaced by $\downarrow$ (the choice of pointers is straightforward).*

Definition 18 (interacting sequence, composition) *An **interacting sequence** $s = m_1 \dots m_n$ is a justified sequence of shape $\{\uparrow, \downarrow\uparrow, \downarrow\downarrow\}$ such that $s\restriction_{\uparrow,\downarrow\uparrow}$, $s\restriction_{\downarrow\uparrow,\downarrow\downarrow}$ and $s\restriction_{\uparrow,\downarrow\downarrow}$ are plays. The set of interacting sequences is denoted **Int**.*

*Suppose we have two strategies σ and τ. We call **composition** of σ and τ the set of plays*

$$\sigma;\tau = \{u\restriction_{\uparrow,\downarrow\downarrow} \mid u \in \textbf{\textit{Int}},\ u\restriction_{\uparrow,\downarrow\uparrow} \in \tau \text{ and } u\restriction_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma\}$$

Proposition 2 *If $\sigma : A \to B$ and $\tau : B \to C$ then $\sigma;\tau : A \to C$.*

Proof: The proof is straightforward: indeed it suffices to consider $\mathcal{M}_A$ as the arena (instead of A) to see that we are, strictly speaking, in the usual HO-setting. So the result comes from free (see for example [McC96]), as well as the associativity of composition, and the preservation of innocence by composition. □

[6]One can then wonder why we define a strategy as a set of plays, and not directly as a set of views: this is due to the algorithm for composition, which is a bit simpler to define with this notion of strategy.

3.3 Uniformity

3.3.1 Introduction

What we have defined up to now allows us to construct the strategies that will interpret system F, and to compose them. However there are lots of different strategies that can be constructed in this setting, and most of them will not correspond to any term of the syntax.

Where does it come from ? The main point is that, in the syntax, the behaviour of **O** is not symmetric to the behaviour of **P**, especially because of the second-order quantification. This is expressed by the opposition

$$\Lambda X \qquad \text{vs.} \qquad \{B\}$$

The first one corresponds to the behaviour of **O**, the second one to the behaviour of **P**. On the contrary, in our strategies the behaviours of **O** and **P** are symmetrical (if we forget about innocence). So we have to break this symmetry, and this will be done through the notion of **uniformity**.

The hint to break the symmetry is simply to mimic syntax: the second-order quantification appears in **O**-moves through the symbols ΛX where X is a fresh name. So we will require that **O** can only play moves of the form $\star^{X_i}$, where X_i is "fresh" in a suitable sense. This gives us the notion of **symbolic strategy**.

However, even if they describe the syntax with a satisfactory precision, symbolic strategy do not compose properly[7] because of the asymmetry we have introduced between **O** and **P**. Indeed, the interaction process is based on symmetrical behaviours of both players.

The solution is then to extend the symbolic strategy, through a process called **copycat expansion**, to allow **O** to play moves with the symbol $\star^B$ for any B, but without giving more freedom to the strategy. A strategy generated by a symbolic strategy through copycat expansion will be called a **uniform strategy**. Uniform strategies compose properly, although their behaviour is still given by their symbolic part.

The obvious reason for introducing uniformity is to obtain a model that is closer to the syntax: in particular, one can obtain full completeness results if we have uniformity and a few other constraints on strategies. But surprisingly, the condition of uniformity would appear in our model even if we did not care about modelising precisely system F. We will see in section 3.3.4 that uniformity is in fact *necessary* to build a proper model of system F from the strategies we have defined.

The notion of uniformity in games appeared implicitly in [Hug00], and more explicitly in [MO01].

3.3.2 Copycat expansion

Here we define the operation of copycat expansion, which lies in the core of the uniformity property. Note that copycat expansion has another interest in our model: it will be used to define the substitution on strategies, i.e. the operation that transforms a strategy $\sigma : \forall X.A$ into $\sigma[B/X] : A[B/X]$.

Given a justified sequence t, an arena B and an integer $j > 0$, we define the sequence $t[B/X_j]$ as the arena where each symbol $\star^D$ has been replaced by $\star^{D[B/X_j]}$. This construction is used for defining copycat expansion:

[7] Actually they compose, but the result is not the same result as what we would obtain through the syntax.

Definition 19 (copycat expansion) *Given an innocent strategy $\sigma : A$, a type B and an integer $j > 0$, we define*

$$S_{B/X_j} = \{\ulcorner t \urcorner \mid \ \exists s_0 \in \sigma,\ s_0[B/X_j] \rightsquigarrow_{B/X_j} t\}$$

where $s \rightsquigarrow_{B/X_j} t$ means that the play t can be obtained from s by replacing any subsequence

$$m_1\ m_2 \qquad \textit{with } \lambda(m_1) = \mathbf{O} \textit{ and } \sharp(m_1) = j$$

by the sequence[8]

$$\begin{cases} m_1[M_1]\ m_2[M_1]\ m_2[M_2]\ m_1[M_2]\ \ldots\ m_1[M_p]\ m_2[M_p] & \textit{if } p \textit{ odd} \\ m_1[M_1]\ m_2[M_1]\ m_2[M_2]\ m_1[M_2]\ \ldots\ m_2[M_p]\ m_1[M_p] & \textit{if } p \textit{ even} \end{cases}$$

for some bi-view $u = M_1 \ldots M_p$ in B.

*S_{B/X_j} is called the B-**copycat expansion** of σ along X_j.*

To have a good understanding of copycat expansion, one should think of it as a translation of the operation of η-expansion into games. Indeed, in usual HO-games, the set of views of a strategy can be seen as the β-normal, η-long form of a term. Then, to go from the term $\Gamma \vdash t : A$ in β-normal, η-long form (with $X \notin FTV(\Gamma)$) to the term $\Gamma \vdash t[B/X] : A[B/X]$ in β-normal, η-long form, one only needs to do an η-expansion, and to replace every X appearing in the term by B. This is precisely what copycat expansion does with games.

Given an innocent strategy σ, we define the **substitution** of X_j by B in σ as the strategy

$$\sigma[B/X_j] = \exp(S_{B/X_j})$$

We also note, for $\mathbf{B} = \langle B_1, \ldots, B_n \rangle$ and $\mathbf{X} = \langle X_1, \ldots, X_n \rangle$,

$$\sigma[\mathbf{B}/\mathbf{X}] = \sigma[B_1/X_1] \ldots [B_n/X_n]$$

This is a strategy on the arena

$$A[\mathbf{B}/\mathbf{X}] = A[B_1/X_1] \ldots [B_n/X_n]$$

3.3.3 Uniform and winning strategies

Definition 20 (copycat variable, symbolic strategy) *Consider a view $s = s_1 m$ on an arena A and an atomic arena X_i imported by $\mathbf{O}$ at move m. We say that X_i is a **copycat variable** if we have $X_i \notin FTV(s_1) \cup FTV(A)$.*

*A view s in T is called **symbolic** if, for any $\mathbf{O}$-move m of s, the types imported at move m are two by two distinct copycat variables.*

A strategy is symbolic if it is innocent and all its views are symbolic.

Definition 21 (uniform strategy) *A strategy σ is called **uniform** if there is a symbolic strategy $\bar{\sigma}$ such that: σ is the smallest innocent strategy containing $\bar{\sigma}$ which is stable by any copycat expansion along any copycat variable.*

[8] Note that we need to define the pointers for our new sequences of moves. In fact they are straightforward: $m_2[M_1]$ is justified by $m_1[M_1]$, $m_2[M_2]$ is justified by $m_2[M_1]$ if M_1 justifies M_2 in u, etc.

Remarks :

1. The definition of a copycat variable obviously corresponds to the requirement of freshness of a type variable, translated into games.
2. Given a symbolic strategy $\bar{\sigma}$, the "smallest innocent strategy containing $\bar{\sigma}$ which is stable by any copycat expansion along any copycat variable" always exists: it is the intersection of all strategies verifying these conditions.
3. According to the definition, it is a priori possible to have distinct symbolic strategies generating the same uniform strategy σ. In fact these distinct strategies only differ by the choice of the copycat variables: indeed, as they are all contained in σ, it is possible to reconstruct them from each other by copycat expansions. We then allow ourselves, by a slight abuse, to talk about *the* symbolic strategy associated to a uniform strategy.

It is important for the sake of this article to understand that uniformity is very much connected with the intuitive notion of parametricity we gave in the introduction. Indeed, we have introduced parametricity as the idea that a function $f : \forall X.A$ cannot depend on the value we give for X, that X has to be a *black box* with respect to f. And this is exactly what uniformity is about: a quantifier in an Opponent position will result in symbols $\star^B$ where B is imported by **O**, but the behaviour of the strategy shall not depend on the values of B that **O** chooses, it must be entirely given by the case where $B = X_i$ for some fresh X_i. Here, this X_i plays the role of the black box.

Note that we make no mention here of quantifiers in a Player position; similarly, the intuitive requirement of parametricity says nothing about functions f of type $(\forall X.A) \to B$.

Given the definition of a uniform strategy, one is able to express what a winning strategy will be:

Definition 22 (winning strategy) *We say that a strategy $\sigma : A$ is* ***winning*** *if it satisfies the following conditions:*

- *it is uniform, with an associated symbolic strategy $\bar{\sigma}$*
- *it is total on A: for every $s \in \sigma$, if sm is a play in A, then $smn \in \sigma$ for some move n*
- *$\ulcorner \bar{\sigma} \urcorner$ is a finite set of views.*

These winning strategies will be the ones we use in the model. Hence we need the following:

Proposition 3 *If $\sigma : A \to B$ and $\tau : B \to C$ are winning strategies, the the strategy $\sigma ; \tau : A \to C$ is winning.*

The preservation of uniformity was attested by a direct proof in [dL08a]. The proof of preservation of totality and finiteness is always a long and winding road in game semantics, but in our case it can be done by using the full completeness theorem proved in [dL07].

3.3.4 The necessity of uniformity

The uniformity condition was presented originally as a way to make our model more precise. But in fact it happens that we do not have a choice: in order to obtain a model of system F, we have to require uniformity (or at least a similar condition).

This all comes from the second-order η-equality imposed by syntax:

$$\Lambda X.t\{X\} = t \qquad \text{if } X \notin FTV(t)$$

If we note σ_f the strategy interpreting a term f, the equality means the following: given $\sigma_t : \forall X.A$, we have to be able to recover this strategy from the strategy $\sigma_{t\{X\}}$, through a deterministic operation (that will correspond to $u \mapsto \Lambda X.u$ in the syntax). So, more generally, given any $\sigma : \forall X.A$, the strategy $\sigma\{X\}$ must contain as much information as σ. Otherwise said, the behaviour of σ for any **O**-move starting with $\star^B$ must be driven by its behaviour for **O**-moves starting with $\star^X$.

This is not a formal argument, but it shows that, to obtain a model of our calculus, we need to require a specific condition, and that uniformity is a good candidate for such a condition. But does it mean that every game model of system F has to impose such a condition ? In section 4 we will see that the answer is no: it is possible, by changing the way strategies are defined, to produce a model without any condition resembling uniformity. But the resulting model will not be Strachey parametric, whereas our uniform model is, as we will see later on.

3.4 The model

Now we are going to describe how to build a hyperdoctrine from our games. We will be sketchy here, the reader interested by more details may refer to [Hug00] or [dL07].

The base category $\mathbb{H}$ will have as objects natural numbers and as morphism $n \to m$ the m-tuples $\langle A_1, \dots, A_m \rangle$, where each A_i is a type with $FTV(A_i) \subseteq \{X_1, \dots X_n\}$. The composition in this category is substitution: if $\mathbf{A} = \langle A_1, \dots, A_m \rangle : n \to m$ and $\mathbf{B} = \langle B_1, \dots, B_n \rangle : k \to n$ then

$$\mathbf{A} \circ \mathbf{B} = \langle A_1[\mathbf{B}/\mathbf{X}], \dots, A_m[\mathbf{B}/\mathbf{X}] \rangle : k \to m$$

where $\mathbf{X} = \langle X_1, \dots, X_n \rangle$.

The functor $H : \mathbb{H}^{op} \to \mathbf{CCC}$ is defined as follows:

- For any n, $H(n)$ is the categories whose objects are types A with $FTV(A) \subseteq \{X_1, \dots X_n\}$ and a morphism from A to B is a winning strategy such that

  $$\forall sm \in \sigma,\ FTV(sm) \subseteq \{X_1, \dots, X_n\} \cup FTV(s)$$

  Again we will not give an explicit proof that this forms a cartesian closed category, see [dL07]. We just give the definition of the identity strategy, as we will need it later:

  $$id_{A \to A} = \exp(S_{id})$$

  where S_{id} is the set of views of the form

  $$\begin{cases} s = \uparrow m_1 \ \downarrow m_1 \ \downarrow m_2 \uparrow m_2 \dots \uparrow m_p \ \downarrow m_p & \text{if } p \text{ odd} \\ s = \uparrow m_1 \ \downarrow m_1 \ \downarrow m_2 \uparrow m_2 \dots \downarrow m_p \ \uparrow m_p & \text{if } p \text{ even} \end{cases}$$

  with $u = m_1 \dots m_p$ bi-view in A

  Unsurprisingly, this definition contains similarities with the copycat expansion process: that is because in the syntax η-expansion is connected with the identity.

- For any morphism $\mathbf{B} : n \to m$ in $\mathbb{H}$, the functor $H(\mathbf{B}) : H(m) \to H(n)$ is defined as follows (with $\mathbf{X} = \langle X_1, \ldots, X_n \rangle$) :
  - for any A object in $H(m)$,
  $$H(\mathbf{B})(A) = A[\mathbf{B}/\mathbf{X}]$$
  - for any $\sigma : A \to B$,
  $$H(\mathbf{B})(\sigma) = \sigma[\mathbf{B}/\mathbf{X}]$$

In the category $\mathbb{H}$, the projection is $\mathbf{X} = \langle X_1, \ldots, X_n \rangle : n+1 \to n$, it gives us a functor $H(\mathbf{X}) : H(n) \to H(n+1)$. To find a right adjoint to it, we introduce the **quantification over a strategy**: if A, B are objects of $H(n+1)$ and $\sigma : A \to B$, then the quantification of X_{n+1} over σ is the (winning) strategy $\forall_{n+1}\sigma = \exp(S_1)$ with

$$S_1 = \{\ulcorner t \urcorner \mid \ \exists B \in \mathcal{A},\ \exists s \in \sigma[B/X_{n+1}],\ s \rightsquigarrow^1_B t\}$$

where $s \rightsquigarrow^1_B t$ means that the play t is obtained from s by replacing every move $\uparrow m$ by $\uparrow \star^B m$ and every move $\downarrow m$ by $\downarrow \star^B m$. We then have $\forall_{n+1}\sigma : (\forall X_{n+1}.A) \to (\forall X_{n+1}.B)$.

One can now define the functor Π_n by $\Pi_n(A) = \forall X_{n+1}.A$ and $\Pi_n(\sigma) = \forall_{n+1}\sigma$. To see that it is a right adjoint to $H(\mathbf{X})$, we construct the bijection

$$\kappa : \mathrm{Hom}_{H(n+1)}(H(\mathbf{X})(\Gamma), A) \to \mathrm{Hom}_{H(n)}(\Gamma, \Pi_n(A))$$

as $\kappa(\sigma) = \exp(S_2)$ with

$$S_2 = \{\ulcorner t \urcorner \mid \ \exists B \in \mathcal{A},\ \exists s \in \sigma[B/X_{n+1}],\ s \rightsquigarrow^2_B t\}$$

where $s \rightsquigarrow^2_B t$ means that the play t is obtained from s by replacing every move $\uparrow m$ by $\uparrow \star^B m$. The inverse of κ is

$$\kappa^{-1} : \mathrm{Hom}_{H(n)}(\Gamma, \Pi_n(A)) \to \mathrm{Hom}_{H(n+1)}(H(\mathbf{X})(\Gamma), A)$$

defined by $\kappa^{-1}(\tau) = \exp(S_3)$ with

$$S_3 = \{\ulcorner t \urcorner \mid \ \exists s \in \tau,\ s \rightsquigarrow^3 t\}$$

where $s \rightsquigarrow^3 t$ means that every move $\uparrow \star^B m$ in s is such that $B = X_{n+1}$, and t is obtained from s by replacing each move of this kind by $\uparrow m$.

Note that to have $\kappa^{-1} \circ \kappa = \kappa \circ \kappa^{-1} = id$ we need the property of uniformity for our strategies.

The reader can check the last required properties to verify that we obtain a hyperdoctrine of games, denoted $\mathcal{G}$.

3.5 Dinaturality as a local condition

We will now see how multivariance and dinaturality can be defined for our games. Considering strategies up to dinaturality is a rather common operation, even in games: see for example [AM99]. The difference here is that dinaturality is not taken as a part of the definition of the strategies, but rather proved (on simple objects) for the strategies we have defined. But more importantly, we give an interpretation of the two morphisms of the dinaturality diagram that allows to understand dinaturality as a local property on the subsequences of the plays of the strategy.

3.5.1 Multivariance in games

We can define a (not restricted) multivariance on our games. We have given in section 3.1 the definition of a double substitution which generates, starting from three arenas A, B, C and an integer $j > 0$, the arena $A[(B,C)/X_j]$. This construction defines our multivariance on objects, now let us define it for morphisms.

Consider an arena A, two strategies $\sigma : B_1 \to C_1$ and $\tau : B_2 \to C_2$, and an integer $j > 0$. We define the set

$$S = \{\ulcorner t \urcorner \mid \ \exists s \in id_{A \to A},\ s \rightsquigarrow_{(\sigma,\tau)/X_j} t\}$$

where $s \rightsquigarrow_{(\sigma,\tau)/X_j} t$ means that the play t can be obtained from s by replacing any subsequence

$$m_1\ m_2 \qquad \text{with } \lambda(m_1) = \mathbf{O},\ \partial(m_1) \in \mathcal{O}_A \text{ and } \sharp(m_1) = j$$

by the sequence

$$\mu_1[N_1]\ \mu_2[N_2]\ \ldots\ \mu_p[N_p]$$

with

$$N_i = \begin{cases} c & \text{if } M_i = \uparrow c \\ b & \text{if } M_i = \downarrow b \end{cases} \qquad \mu_i = \begin{cases} m_1 & \text{if } M_i = \uparrow c \\ m_2 & \text{if } M_i = \downarrow b \end{cases}$$

where $u = M_1 \ldots M_p$ is a play in τ if $m_1 = \uparrow m$, and a play in σ if $m_1 = \downarrow m$.

We define $A[(\sigma,\tau)/X_j] = \exp(S)$, this is a winning strategy on $A[(C_1, B_2)/X_j] \to A[(B_1, C_2)/X_j]$. Given $\mathbf{X} = \langle X_1, \ldots, X_n \rangle$, $\boldsymbol{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ and $\boldsymbol{\tau} = \langle \tau_1, \ldots, \tau_n \rangle$, we can define a strategy $A[(\boldsymbol{\sigma}, \boldsymbol{\tau})/\mathbf{X}]$ in a similar manner, by substituting every subsequence $m_1\ m_2$ such that $\lambda(m_1) = \mathbf{O}$, $\partial(m_1) \in \mathcal{O}_A$ and $\sharp(m_1) = j$ with $j \in [1, n]$ by the appropriate sequence.

We can now define our multivariance: for any integer n, $\mathcal{V}^n : \mathbb{H} \to \overline{\mathbf{Cat}}$ is constructed as follows:

- $\mathcal{V}^n(m) = H(n)^m$
- for $\mathbf{A} = \langle A_1, \ldots, A_m \rangle : p \to m$, the functor

  $$\mathcal{V}^n(\mathbf{A}) : (H(n)^p)^{op} \times H(n)^p \to H(n)^m$$

  is given by

  $$\mathcal{V}^n(\mathbf{A})[x, y] = (A_1[(x,y)/\mathbf{X}], \ldots, A_m[(x,y)/\mathbf{X}])$$

  with $\mathbf{X} = \langle X_1, \ldots, X_p \rangle$.

This multivariance extends the standard multivariance on simple types: indeed, we have, given two arenas A and B,

$$\begin{aligned} X_i[(x,y)/\mathbf{X}] &= y_i \\ (A \times B)[(x,y)/\mathbf{X}] &= A[(x,y)/\mathbf{X}] \times B[(x,y)/\mathbf{X}] \\ (A \to B)[(x,y)/\mathbf{X}] &= A[(y,x)/\mathbf{X}] \to B[(x,y)/\mathbf{X}] \end{aligned}$$

where $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are tuples of arenas or of strategies.

3.5.2 Study of the dinaturality diagram

Given the multivariance defined above, do we have the property of dinaturality ? As we shall see, the answer to this question is no. However, one will be able to express very formally what the two morphisms implied in the dinaturality diagram are, and why they differ in general. And this will lead us to the conclusion that, in this model, dinaturality is true for simple types.

We consider as a first step the case with only one type containing one free variable. The general case with several types and several variables will be an obvious expansion of this.

Definition 23 (upper and lower expansions) *We consider two uniform strategies $\sigma : A$ and $\tau : B \to C$, and an integer $j > 0$.*

We define the sets

$$S_u = \{\ulcorner t \urcorner \mid \exists s_0 \in \sigma, s_0[B/X_j] \rightsquigarrow_u t\}$$
$$S_l = \{\ulcorner t \urcorner \mid \exists s_0 \in \sigma, s_0[C/X_j] \rightsquigarrow_l t\}$$

where $s \rightsquigarrow_u t$ (resp. $s \rightsquigarrow_l t$) means that the play t can be obtained from s by replacing any subsequence

$$m_1\ m_2 \qquad \text{with } \lambda(m_1) = \mathbf{O} \text{ and } \sharp(m_1) = j$$

by:

- *the sequence*
$$\mu_1[N_1]\ \mu_2[N_2]\ \ldots\ \mu_p[M_p]$$
with
$$N_i = \begin{cases} c & \text{if } M_i = \uparrow c \\ b & \text{if } M_i = \downarrow b \end{cases} \qquad \mu_i = \begin{cases} m_1 & \text{if } M_i = \uparrow c \\ m_2 & \text{if } M_i = \downarrow b \end{cases}$$
$$\text{where } u = M_1 \ldots M_p \text{ is a play in } \tau$$
in the case where $\partial(m_1) \in \mathcal{O}_A$ (resp. $\partial(m_2) \in \mathcal{O}_A$)

- *the sequence*
$$\begin{cases} m_1[M_1]\ m_2[M_1]\ m_2[M_2]\ m_1[M_2]\ \ldots\ m_1[M_p]\ m_2[M_p] & \text{if } p \text{ odd} \\ m_1[M_1]\ m_2[M_1]\ m_2[M_2]\ m_1[M_2]\ \ldots\ m_2[M_p]\ m_1[M_p] & \text{if } p \text{ even} \end{cases}$$
$$\text{where } u = M_1 \ldots M_p \text{ is a bi-view in } B$$
otherwise.

*Then the strategy $\sigma[\tau/X_j]_u = exp(S_u)$ (resp. $\sigma[\tau/X_j]_l = exp(S_l)$) is called the **upper expansion** (resp. the **lower expansion**) of σ by τ along X_j.*

Here comes the main technical result of this paper: it states that the the two strategies defined above, $\sigma[\tau/X_j]_u$ and $\sigma[\tau/X_j]_l$ are, respectively, the upper and lower morphisms appearing in the dinaturality diagram. That is why we say that we have a local characterisation of dinaturality: from the morphism for which we want to test dinaturality, the two morphisms of the diagram are obtained by using only local operations: namely modifying some specific subsequences (and also the instantiations $\star^B$). Then, to see if the dinaturality property is true, we will just need to compare the result of these local transformations, to see if they coincide.

The result states as follows:

Proposition 4 *Given $\sigma : A \to Z$, $\tau : B \to C$ and $j > 0$, if σ and τ are winning strategies then we have the following equalities:*

$$\sigma[\tau/X_j]_u = A[(\tau, id_B)/X_j] \ ; \ \sigma[B/X_j] \ ; \ Z[(id_B, \tau)/X_j]$$
$$\sigma[\tau/X_j]_l = A[(id_C, \tau)/X_j] \ ; \ \sigma[C/X_j] \ ; \ Z[(\tau, id_C)/X_j]$$

PROOF: We will prove only the first equality, the proof for the second one is similar.

σ and τ being winning strategies, we also have that $A[(\tau, id_B)/X_j]$, $\sigma[B/X_j]$, $Z[(id_B, \tau)/X_j]$ and $\sigma[\tau/X_j]_u$ are winning, so both sides of the equality are winning strategies on $A[(C, B)/X_j] \to Z[(B, C)/X_j]$. We note

$$\rho = A[(\tau, id_B)/X_j] \ ; \ \sigma[B/X_j] \ ; \ Z[(id_B, \tau)/X_j]$$

Any play in ρ has the form $u\restriction_{\uparrow,\downarrow\downarrow\downarrow}$ where u is a justified sequence of shape $\{\uparrow, \downarrow\uparrow, \downarrow\downarrow\uparrow, \downarrow\downarrow\downarrow\}$ such that $u\restriction_{\uparrow,\downarrow\uparrow} \in Z[(id_B, \tau)/X_j]$, $u\restriction_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u\restriction_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau, id_B)/X_j]$.

If we can prove that for any $smn \in \sigma[\tau/X_j]_u$ with $s \in \rho$ we have $smn \in \rho$, then we are done: indeed, it would imply by induction that any play s in $\sigma[\tau/X_j]_u$ is also in ρ, and consequently $\sigma[\tau/X_j]_u = \rho$ because the strategies are total.

The idea of the proof will be to consider that play s together with the sequence u such that $u\restriction_{\uparrow,\downarrow\downarrow\downarrow} = s$, $u\restriction_{\uparrow,\downarrow\uparrow} \in Z[(id_B, \tau)/X_j]$, $u\restriction_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u\restriction_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau, id_B)/X_j]$, and to identify a sequence u', of the form $u' = um'n'$ or $u' = um'm''n''n'$, such that $u'\restriction_{\uparrow,\downarrow\downarrow\downarrow} = smn$, $u'\restriction_{\uparrow,\downarrow\uparrow} \in Z[(id_B, \tau)/X_j]$, $u'\restriction_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u'\restriction_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau, id_B)/X_j]$. In each case we will only give the sequence u'; the verification that it satisfies the conditions is not necessarily trivial, but it is always ensured by the definitions of $Z[(id_B, \tau)/X_j]$, $\sigma[B/X_j]$ and $A[(\tau, id_B)/X_j]$.

So, let us consider $smn \in \sigma[\tau/X_j]_u$ with $s \in \rho$. Without loss of generality we can ask smn to be a view. There exists a play $s_0 \in \sigma$ such that $s_0[B/X_j] \rightsquigarrow_u smn$, and hence a subsequence $m_1 m_2$ (with $\lambda(m_1) = \mathbf{O}$) in $s_0[B/X_j]$ such that $m = \mu_1[N_1]$ and $m_2 = \mu_2[N_2]$ with $\mu_1, \mu_2 \in \{m_1, m_2\}$. We have three cases:

1. either m_1 is such that $\sharp(m_1) \neq j$
2. or m_1 is such that $\sharp(m_1) = j$ and $\partial(m_1) \in \mathcal{O}_{A \to Z}$
3. or m_1 is such that $\sharp(m_1) = j$ and $\partial(m_1) \notin \mathcal{O}_{A \to Z}$

Let us consider the case 1: in that case we have that $m = m_1$ and $n = m_2$. As $s \in \rho$, there exists u such that $u\restriction_{\uparrow,\downarrow\downarrow\downarrow} = s$, $u\restriction_{\uparrow,\downarrow\uparrow} \in Z[(id_B, \tau)/X_j]$, $u\restriction_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u\restriction_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau, id_B)/X_j]$. If we set

$$m'_1 = \begin{cases} \uparrow M_1 & \text{if } m_1 = \uparrow M_1 \\ \downarrow\downarrow\downarrow M_1 & \text{if } m_1 = \downarrow M_1 \end{cases} \qquad m''_1 = \begin{cases} \downarrow\uparrow M_1 & \text{if } m_1 = \uparrow M_1 \\ \downarrow\downarrow\uparrow M_1 & \text{if } m_1 = \downarrow M_1 \end{cases}$$

and

$$m_2' = \begin{cases} \uparrow M_2 & \text{if } m_2 = \uparrow M_2 \\ \downarrow\downarrow\downarrow M_2 & \text{if } m_2 = \downarrow M_2 \end{cases} \qquad m_2'' = \begin{cases} \downarrow\uparrow M_2 & \text{if } m_1 = \uparrow M_2 \\ \downarrow\downarrow\uparrow M_2 & \text{if } m_1 = \downarrow M_2 \end{cases}$$

then the sequence $u' = um_1'm_1''m_2''m_2'$ is such that $u'\upharpoonright_{\uparrow,\downarrow\downarrow\downarrow} = smn$, $u'\upharpoonright_{\uparrow,\downarrow\uparrow} \in Z[(id_B,\tau)/X_j]$, $u'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u'\upharpoonright_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau,id_B)/X_j]$. Hence, $smn \in \rho$.

Suppose now that we are in case 2. Again, as $s \in \rho$ there exists u such that $u\upharpoonright_{\uparrow,\downarrow\downarrow\downarrow} = s$, $u\upharpoonright_{\uparrow,\downarrow\uparrow} \in Z[(id_B,\tau)/X_j]$, $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u\upharpoonright_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau,id_B)/X_j]$. We recall that $m = \mu_1[N_1]$ and $m_2 = \mu_2[N_2]$ with $\mu_1,\mu_2 \in \{m_1,m_2\}$.

If we have $\mu_1 = \mu_2$ then the sequence $u' = u\mu_1'[N_1]\mu_1'[N_2]$ with

$$\mu_1' = \begin{cases} \uparrow M_1 & \text{if } \mu_1 = \uparrow M_1 \\ \downarrow\downarrow\downarrow M_1 & \text{if } \mu_1 = \downarrow M_1 \end{cases}$$

is such that $u'\upharpoonright_{\uparrow,\downarrow\downarrow\downarrow} = smn$, $u'\upharpoonright_{\uparrow,\downarrow\uparrow} \in Z[(id_B,\tau)/X_j]$, $u'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u'\upharpoonright_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau,id_B)/X_j]$. Hence, $smn \in \rho$.

If we have $\mu_1 \neq \mu_2$ then we set

$$\mu_1' = \begin{cases} \uparrow M_1 & \text{if } \mu_1 = \uparrow M_1 \\ \downarrow\downarrow\downarrow M_1 & \text{if } \mu_1 = \downarrow M_1 \end{cases} \qquad \mu_1'' = \begin{cases} \downarrow\uparrow M_1 & \text{if } \mu_1 = \uparrow M_1 \\ \downarrow\downarrow\uparrow M_1 & \text{if } \mu_1 = \downarrow M_1 \end{cases}$$

and

$$\mu_2' = \begin{cases} \uparrow M_2 & \text{if } \mu_2 = \uparrow M_2 \\ \downarrow\downarrow\downarrow M_2 & \text{if } \mu_2 = \downarrow M_2 \end{cases} \qquad \mu_2'' = \begin{cases} \downarrow\uparrow M_2 & \text{if } \mu_1 = \uparrow M_2 \\ \downarrow\downarrow\uparrow M_2 & \text{if } \mu_1 = \downarrow M_2 \end{cases}$$

The sequence $u' = u\mu_1'\mu_1''\mu_2''\mu_2'$ is such that $u'\upharpoonright_{\uparrow,\downarrow\downarrow\downarrow} = smn$, $u'\upharpoonright_{\uparrow,\downarrow\uparrow} \in Z[(id_B,\tau)/X_j]$, $u'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u'\upharpoonright_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau,id_B)/X_j]$. Hence, $smn \in \rho$.

Finally, let us consider the case 3. If we write $m_1 = \mu_1[N_1]$ and $m_2 = \mu_2[N_2]$ with $\mu_1,\mu_2 \in \{m_1,m_2\}$, we have in that case $N_1 = M_2$ and $\mu_1 \neq \mu_2$. Then if we set

$$\mu_1' = \begin{cases} \uparrow M_1 & \text{if } \mu_1 = \uparrow M_1 \\ \downarrow\downarrow\downarrow M_1 & \text{if } \mu_1 = \downarrow M_1 \end{cases} \qquad \mu_1'' = \begin{cases} \downarrow\uparrow M_1 & \text{if } \mu_1 = \uparrow M_1 \\ \downarrow\downarrow\uparrow M_1 & \text{if } \mu_1 = \downarrow M_1 \end{cases}$$

and

$$\mu_2' = \begin{cases} \uparrow M_2 & \text{if } \mu_2 = \uparrow M_2 \\ \downarrow\downarrow\downarrow M_2 & \text{if } \mu_2 = \downarrow M_2 \end{cases} \qquad \mu_2'' = \begin{cases} \downarrow\uparrow M_2 & \text{if } \mu_1 = \uparrow M_2 \\ \downarrow\downarrow\uparrow M_2 & \text{if } \mu_1 = \downarrow M_2 \end{cases}$$

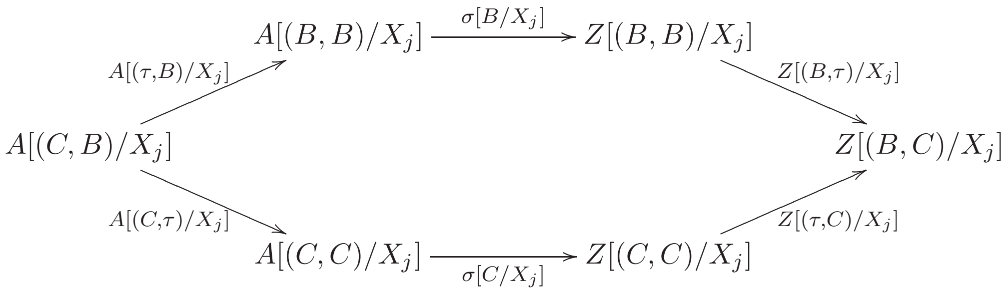
the sequence $u' = u\mu_1'\mu_1''\mu_2''\mu_2'$ is such that $u'\upharpoonright_{\uparrow,\downarrow\downarrow\downarrow} = smn$, $u'\upharpoonright_{\uparrow,\downarrow\uparrow} \in Z[(id_B,\tau)/X_j]$, $u'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow\uparrow} \in \sigma[B/X_j]$ and $u'\upharpoonright_{\downarrow\downarrow\uparrow,\downarrow\downarrow\downarrow} \in A[(\tau,id_B)/X_j]$. Hence, $smn \in \rho$. □

Now what happens if we suppose that A and Z are simple objects ? If A is a simple object, this means that it can be generated from the arenas X_i and the operations $\times$ and $\rightarrow$. In particular, for each $c \in \mathcal{O}_A$, c does not contain the symbol $\star$ and $\mathcal{L}_A(c) = \dagger$. As a consequence, if $\sigma : A \rightarrow Z$ with A and Z simple objects, then for any $\tau : B \rightarrow C$ we have:

$$\sigma[\tau/X_j]_u = \sigma[\tau/X_j]_l$$

Indeed, the definitions of the two expansions will coincide: the operations $s_0 \mapsto s_0[B/X_j]$ and $s_0 \mapsto s_0[C/X_j]$ have no incidence on the plays of σ, and whenever $m_1 m_2$ is a subsequence of a play of σ, then we have $\partial(m_1) = m_1 \in \mathcal{O}_{A\to Z}$ as well as $\partial(m_2) = m_2 \in \mathcal{O}_{A\to Z}$, so that $\rightsquigarrow_u$ and $\rightsquigarrow_l$ coincide.

From Prop. 4, this implies that the dinaturality diagram is true for one type $A \to Z$ containing one free variable X_j. Namely:



This can be seen as an alternate proof of the dinaturality of simply-typed λ-calculus, known since [GSS92]. Indeed, once restricted to simple objects, our games form a fully and faithful complete model of this calculus. So if dinaturality is true in their case, it is true for the calculus.

But games tell us more than that: they express the exact conditions which can make the dinaturality diagram go wrong. These conditions rely on the distinction between $\sigma[\tau/X_j]_u$ and $\sigma[\tau/X_j]_l$:

- the substitution of X_j by B or C
- the distinction between the cases where $\partial(m_1) \in \mathcal{O}_A$ and $\partial(m_2) \in \mathcal{O}_A$ (for $m_1 m_2$ consecutive moves in a play of σ).

Each of these points is an obstacle to dinaturality. The first one is obvious, just by looking at a term like $x : \forall Y.Y \vdash x\{X_j\} : X_j$ we know it will not be dinatural. But the second point is more interesting: it can be useful if for example one is interested in the Curry-style system F, because in this case the first distinction would disappear. Consequently, it might be possible to use a game model[9] of Curry-style system F to discriminate between dinatural and not dinatural terms of the syntax.

The generalisation of the above diagram to our setting for dinaturality is now straightforward: given $\sigma : A \to Z$, $\mathbf{X} = \langle X_1, \dots, X_n \rangle$ and $\boldsymbol{\tau} = \langle \tau_1, \dots, \tau_n \rangle$, the strategies $\sigma[\boldsymbol{\tau}/\mathbf{X}]_u$ and $\sigma[\boldsymbol{\tau}/\mathbf{X}]_l$ can be defined similarly as above (any subsequence $m_1 m_2$ such that $\sharp(m_1) = j$ with $1 \le j \le n$ has to be replaced by the appropriate sequence), and we have

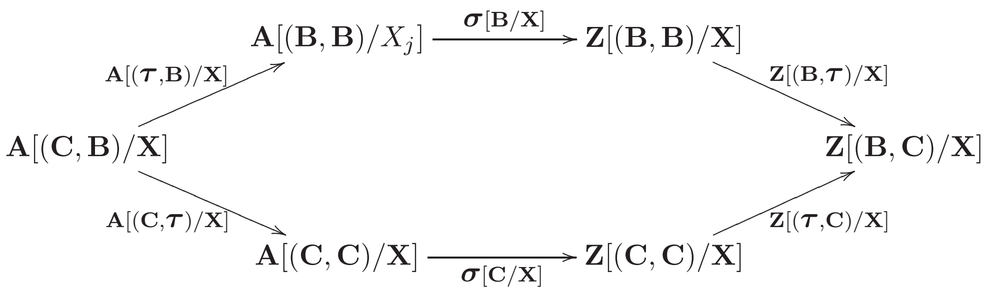
$$\sigma[\boldsymbol{\tau}/\mathbf{X}]_u = \sigma[\boldsymbol{\tau}/\mathbf{X}]_l$$

if A, Z are simple objects. And given $\boldsymbol{\sigma} = \langle \sigma_1, \dots, \sigma_p \rangle$ where $\sigma_i : A_i \to Z_i$ with A_i, Z_i simple objects, we also have

$$\boldsymbol{\sigma}[\boldsymbol{\tau}/\mathbf{X}]_u = \boldsymbol{\sigma}[\boldsymbol{\tau}/\mathbf{X}]_l$$

[9]Such a game model was given in [dL08a], but with a slight difficulty concerning η-equality.

which means that the dinaturality diagram commutes:



So we can finally state:

Theorem 1 *The hyperdoctrine $\mathcal{G}$ is Strachey parametric.*

3.6 Conclusion

This model is an illustration of the way that the criterion for Strachey parametricity works.

Indeed, we did the following: we wanted to construct a model for which we have dinaturality on simple types. But then, in order to satisfy the second-order η-equality, we needed a condition of uniformity. And this condition precisely says something similar to the intuitive parametricity condition.

Of course there is no proof that this is the property of dinaturality which implies uniformity. To be more convinced of this, it would be interesting to have a model which relaxes the condition of dinaturality, and to see if the necessity of uniformity is still there. This is exactly what we will do in the next section.

There is however one puzzling fact in the model, which is the requirement for the strategies to be not only uniform, but winning. As we shall see in section 5, this property is necessary to prove the dinaturality property on simple types. The discussion will show that it is clearly a misfeature due to the dinaturality condition being too strong.

4 A non-parametric example : the non-uniform game model

We shall now introduce a second game model of system F, to illustrate the fact that, without dinaturality on simple types, we can obtain a model with no condition of uniformity, that is, which does not intuitively satisfy parametricity. The model was first presented in [dL08b].

The main idea of this model is that, instead of playing on arenas with free variables, we want to bind all the variables before playing. So, instead of playing on A with $FTV(A) = \{X_1, \ldots, X_n\}$, we want to play on $\forall X_1 \ldots \forall X_n.A$. Indeed, most of the technicalities in the previous model come from the distinction between bound and free variables, so by suppressing this distinction we will make all definitions easier.

In practice, it will mean that a play (containing only one initial move) will be a sequence of moves preceded by a function, denoted θ, instantiating each type variable. Giving this function θ is exactly equivalent to allowing **O** to instantiate each free variable.

Of course, as we do not have a uniformity condition, the resulting model will not be fully complete at all.

4.1 Close moves and non-uniform plays

Arenas are defined as before, by a set of occurrences and a function of linkage. A **close arena** is an arena $A = (\mathcal{O}_A, \mathcal{L}_A)$ such that $\sharp(a) = 0$ for every $a \in \mathcal{O}_A$. The set of close arenas is denoted $\mathcal{A}_c$.

Now the **close moves** are defined on the grammar

$$m ::= \uparrow m \mid \downarrow m \mid \ rm \mid \ lm \mid \ \star^B m \mid \ j \qquad (B \in \mathcal{A}_c, j \in \mathbb{N})$$

Close moves are special kinds of moves, so in particular a close move m on a close arena A is such that $m \in \mathcal{M}_A$.

But using close moves instead of moves clearly creates a lack: when we define the operation $\sigma \mapsto \sigma\{B\}$, we want it to exist for any arena B, not any close arenas ! Be patient, we will see the solution for that in a few moments.

Now comes the main difference with the previous model: the construction of a play. Intuitively, we want a play on A to be a sequence of close moves on a close arena A', that arena being obtained by replacing every type variable in A by a close arena. This means that it should consist of a function θ instantiating each type variable in A, followed by a sequence of close moves.

However, for technical reasons, we cannot define plays as sequences preceded by a θ function[10], this function has to exist for any initial move appearing in the play. That is why we will deal with the notion of **thread**: a thread will consist of a sequence with only one initial move, and a function θ instantiating the free variables. As we have to specify which variables to instantiate, we will deal with n-threads, this n meaning that type variables are among $X_1, \ldots, X_n$. This parameter will sometimes be omitted, when no confusion may arise.

Definition 24 (n-thread) *An* ***n-thread*** *s on an arena A, with $FTV(A) \subseteq \{X_1, \ldots, X_n\}$, is given by:*

- *a function $\theta : \{1, \ldots, n\} \to \mathcal{A}_c$*
- *a sequence $m_1 \ldots m_p$ of close moves on the close arena*

  $$A' = A[\theta(\mathbf{X})/\mathbf{X}]$$

  with $\mathbf{X} = \langle X_1, \ldots, X_n \rangle$, $\theta(\mathbf{X}) = \langle \theta(X_1), \ldots, \theta(X_n) \rangle$ and m_1 initial
- *a function $f : \{2, \ldots, p\} \to \{1, \ldots, p\}$ such that: if $f(i) = j$ then $j < i$ and $m_j \vdash m_i$.*

In this case we generally note $s = \theta m_1 \ldots m_p$, the pointer f being left implicit.

A thread $s = \theta m_1 \ldots m_p$ is ***alternating*** *if we have, for every $1 \leq i \leq p - 1$:*

- *if $\lambda(m_i) = \mathbf{P}$ then $\lambda(m_{i+1}) = \mathbf{O}$*
- *if $\lambda(m_i) = \mathbf{O}$ then $\lambda(m_{i+1}) = \mathbf{P}$.*

Unfortunately, to define composition properly we still need the notions of justified sequences and plays, where there may be several initial moves, each one being preceded by a theta function. But this is only a technicality, and to understand the model it is better to think that we work with alternating threads.

[10] If we did so, we would not have a well-defined cartesian product.

Definition 25 **(n-justified sequence, n-play)** *An* **n-*justified sequence*** *is given by a sequence $s = M_1 \dots M_p$ and a partial function $f : \{1, \dots, p\} \rightharpoonup \{1, \dots, p\}$, such that:*

- *each M_i is either a non-initial move m_i, or a couple (θ, m_i) with $\theta : \{1, \dots, n\} \to \mathcal{A}_c$ and m_i an initial move*
- *if $f(i) = j$ then $j < i$ and $m_j \vdash m_i$; we say that M_u hereditarily justifies M_v if there is a sequence of moves $M_{i_1}, \dots, M_{i_k}$ such that $i_1 = u$, $i_k = v$ and $f(i_r) = i_{r-1}$ for $2 \leq r \leq k$*
- *if M_i is hereditarily justified by $M_j = (\theta, m_j)$ then m_i is a move on $A' = A[\theta(\mathbf{X})/\mathbf{X}]$, with $\mathbf{X} = \langle X_1, \dots, X_n \rangle$ and $\theta(\mathbf{X}) = \langle \theta(X_1), \dots, \theta(X_n) \rangle$.*

This justified sequence is an **n-*play*** *if, for very $1 \leq i \leq p-1$:*

- *if $\lambda(m_i) = \mathbf{P}$ then $\lambda(m_{i+1}) = \mathbf{O}$*
- *if $\lambda(m_i) = \mathbf{O}$ then $\lambda(m_{i+1}) = \mathbf{P}$.*

Of course a thread $\theta m_1 \dots m_p$ can be seen as a justified sequence $M_1 m_2 \dots m_p$ with $M_1 = (\theta, m_1)$.

4.2 Strategies and composition

Strategies and innocence are defined in a very consensual way:

Definition 26 **(n-strategy)** *An* **n-*strategy*** *σ on an arena A, denoted $\sigma : A$, is a non-empty set of even-lengths n-plays on A which is closed by even-length prefix and deterministic: if sm and sn are two plays of σ then $sm = sn$.*

Definition 27 **(n-view)** *An* **n-*view*** *on A is an n-thread $s = \theta m_1 \dots m_p$ on A such that, for every $2 \leq i \leq p-1$, if $\lambda(m_i) = \mathbf{P}$ (i.e. i is even) then m_{i+1} is justified by m_i.*

Definition 28 **(innocence)** *To each n-play s on an arena A, one can associate an n-view $\ulcorner s \urcorner$ by:*

- $\ulcorner \epsilon \urcorner = \epsilon$
- $\ulcorner sM \urcorner = \ulcorner s \urcorner M$ *if* $\lambda(M) = \mathbf{P}$
- $\ulcorner sM \urcorner = M$ *if* $\vdash M$
- $\ulcorner sMtN \urcorner = \ulcorner s \urcorner MN$ *if* $\lambda(N) = \mathbf{O}$ *and M justifies N.*

An n-strategy $\sigma : A$ is called ***innocent*** *if, for any play sN in σ, the move that justifies N is in $\ulcorner s \urcorner$, and if we have: for any $sMN \in \sigma$, $t \in \sigma$, if tM is a play in A and $\ulcorner sM \urcorner = \ulcorner tM \urcorner$ then $tMN \in \sigma$.*

Again, an innocent strategy σ can be defined from a compatible set of views S, it is denoted $\exp(S)$. As a thread is a view, an innocent strategy is completely determined by its set of threads (and even this description will be redundant in general). We will use this later to describe operations on strategies only by their action on threads.

The composition is also straightforward, but we must specify how the restriction copies the θ function:

Definition 29 (restriction) *Let $s = M_1 \dots M_p$ be an n-justified sequence on an arena A, with, for each i, either $M_i = m_i$ and m_i non initial, or $M_i = (\theta_i, m_i)$ and m_i initial. If $\Sigma = \{\zeta, \xi\}$ with $\zeta 0 \vdash^+ \xi 0$, consider the subsequence t_1 of $m_1 \dots m_p$ which contains only the moves of shape Σ, and replace each prefix ζ by $\uparrow$, and each prefix ξ by $\downarrow$, to obtain a sequence $t_2 = m'_1 \dots m'_q$.*

*Then the **restriction** of s to Σ is the sequence $s\upharpoonright_{\zeta,\xi} = M'_1 \dots M'_q$, where $M'_i = m'_i$ if m'_i is non initial, and $M'_i = (\theta, m'_i)$ if m'_i is initial and comes from the element M_j in s, hereditarily justified by $M_k = (\theta, m_k)$.*

Definition 30 (n-interacting sequence, composition) *An **n-interacting sequence** is an n-justified sequence $s = M_1 \dots M_p$ of shape $\{\uparrow, \downarrow\uparrow, \downarrow\downarrow\}$ such that $s\upharpoonright_{\uparrow,\downarrow\uparrow}$, $s\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ and $s\upharpoonright_{\uparrow,\downarrow\downarrow}$ are defined and are plays. The set of n-interacting sequences is denoted $\boldsymbol{Int}^n$.*

*Suppose we have two n-strategies σ and τ. We call **composition** of σ and τ the set of plays*

$$\sigma;\tau = \{u\upharpoonright_{\uparrow,\downarrow\downarrow} \mid u \in \boldsymbol{Int}^n,\ u\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau \text{ and } u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma\}$$

Again, we have:

Proposition 5 *If $\sigma : A \to B$ and $\tau : B \to C$ then $\sigma;\tau : A \to C$.*

This is because the model is very close to an HO-setting, see [dL08b] for more details.

4.3 Construction of the hyperdoctrine

Now comes the time to explain how the model of system F is built in this case. As we shall see, the constructions will be much simpler than in the uniform model. In particular, there is no additional constraint on strategies to ask for.

In what follows, innocent n-strategies are defined through their set of threads.

The base category $\mathbb{H}$ is the same as before: objects are natural numbers, and a morphism from n to m is an m-uple of arenas $\langle A_1, \dots, A_m\rangle$ with $FTV(A_i) \subseteq \{X_1, \dots, X_n\}$.

To define the functor $H : \mathbb{H}^{op} \to \mathbf{Cat}$ we need the operation $\sigma \mapsto \sigma[\mathbf{B}/\mathbf{X}]$ with σ an innocent n-strategy, $\mathbf{X} = \langle X_1, \dots, X_n\rangle$ and $\mathbf{B} = \langle B_1, \dots, B_n\rangle$. The arenas B_i are not necessarily close, their free variables are among $Y_1, \dots, Y_p$. If we note $\mathbf{Y} = \langle Y_1, \dots, Y_p\rangle$, the definition is then :

$$\begin{gathered}\theta m_1 \dots m_n \in \sigma[\mathbf{B}/\mathbf{X}] \\ \text{iff} \\ \theta' m_1 \dots m_n \in \sigma \\ \text{with } \theta'(X_i) = B_i[\theta(\mathbf{Y})/\mathbf{Y}]\end{gathered}$$

Now, H is defined as follows:

- $H(n)$ is the category whose objects are arenas A such that $FTV(A) \subseteq \{X_1, \dots, X_n\}$, and a morphism from A to B is an innocent n-strategy on $A \to B$. It forms a cartesian close category, cf. [dL08b].
- given $\mathbf{B} = \langle B_1, \dots, B_m\rangle : n \to m$, the functor $H(\mathbf{B}) : H(m) \to H(n)$ acts on objects and morphisms as

$$\begin{aligned}H(\mathbf{B})(A) &= A[\mathbf{B}/\mathbf{X}] \\ H(\mathbf{B})(\sigma) &= \sigma[\mathbf{B}/\mathbf{X}]\end{aligned}$$

  with $\mathbf{X} = \langle X_1, \dots, X_m\rangle$.

The quantifier functor $\Pi_n : H(n+1) \to H(n)$ is very easy to define:

$$\Pi_n(A) = \forall X_{n+1}.A$$

and, given $\sigma : A \to B$,

$$\theta m_1 \dots m_p \in \Pi_n(\sigma)$$
$$\text{iff}$$
$$\theta' m'_1 \dots m'_p \in \sigma$$
$$\text{with } \theta'(X_i) = \theta(X_i) \text{ for } 1 \leq i \leq n$$
$$\text{and } m_i = \begin{cases} \uparrow \star^{\theta'(X_{n+1})} M & \text{if } m'_i = \uparrow M \\ \downarrow \star^{\theta'(X_{n+1})} M & \text{if } m'_i = \downarrow M \end{cases}$$

The bijection $\kappa : \mathrm{Hom}_{H(n+1)}(H(\mathbf{X})(\Gamma), A) \to \mathrm{Hom}_{H(n)}(\Gamma, \Pi_n(A))$ is defined in a similar way:

$$\theta m_1 \dots m_p \in \kappa(\sigma)$$
$$\text{iff}$$
$$\theta' m_1 \dots m_p \in \sigma$$
$$\text{with } \theta'(X_i) = \theta(X_i) \text{ for } 1 \leq i \leq n$$
$$\text{and } m'_i = \begin{cases} \uparrow \star^{\theta(X_{n+1})} M & \text{if } m_i = \uparrow M \\ \downarrow M & \text{if } m_i = \downarrow M \end{cases}$$

Finally, all the remaining conditions required to have an hyperdoctrine are easily satisfied, so this construction defines a (non-uniform) model of system F. We note $\mathcal{G}^*$ this new hyperdoctrine.

4.4 Counterexamples to dinaturality

To prove that $\mathcal{G}^*$ is not Strachey parametric we just need a counterexample to the property of dinaturality on simple types, so it will be pretty easy.

Let us consider the types $A = \bot$ and $Z = X_1$, and the 1-strategy $\sigma : A \to Z$ defined by:

$$\sigma = \exp(S) \text{ with } S = \{\theta_1 m_1 n_1 \ , \ \theta_2 m_2 n_2\}$$

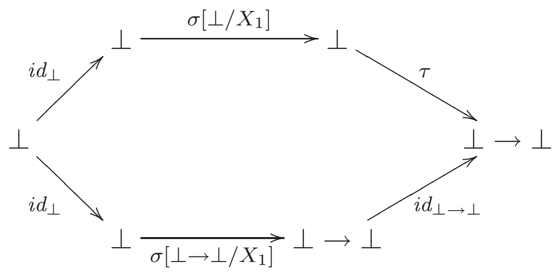
where

$$\theta_1(X_1) = \bot \qquad m_1 = \uparrow 0 \qquad n_1 = \downarrow 0$$

and

$$\theta_2(X_1) = \bot \to \bot \qquad m_2 = \uparrow\uparrow 0 \qquad n_2 = \uparrow\downarrow 0$$

The dinaturality diagram for some $\tau : \bot \to (\bot \to \bot)$ writes as follows:



The lower morphism of the diagram is then $\sigma_1 = \sigma[\bot \to \bot/X_1] = \exp(S_1)$ with $S_1 = \{m_2 n_2\}$ and the upper morphism is $\sigma_2 = \sigma[\bot/X_1]\ ;\ \tau$. Now if we choose τ to be the 0-strategy defined by

$$\tau = \exp(T) \text{ with } T = \{m_2 n_1\}$$

we obtain $\sigma_2 = \exp(S_2)$ with $S_2 = \{m_2 n_1\}$, so the dinaturality diagram is not commutative.

The counterexample above works well because the strategy behaves differently depending on the θ function. But in fact one can generate a counterexample even when the behaviour of the strategy is independent of that function.

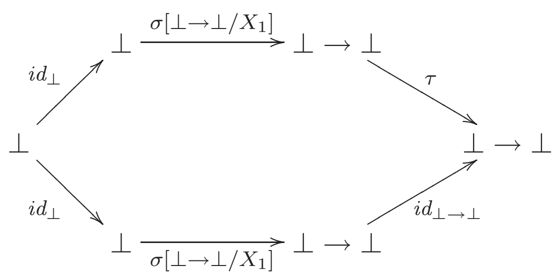
Indeed, consider the strategy $\sigma : \bot \to X_1$ defined by:

$$\sigma = \exp(S) \text{ with } S = \{\theta m n\}$$

where

$$\theta(X_1) = \bot \to \bot \qquad m =\uparrow\uparrow 0 \qquad n =\downarrow 0$$

The dinaturality diagram for some $\tau : (\bot \to \bot) \to (\bot \to \bot)$ writes as follows:



So if we choose

$$\tau = \exp(T) \text{ with } T = \{mn'\}$$

where $n' =\uparrow\downarrow 0$, we have $\sigma_1 = \exp(S_1)$ with $S_1 = \{mn\}$ as lower morphism and $\sigma_2 = \exp(S_2)$ with $S_2 = \{mn'\}$ as upper morphism of the diagram. So, again, we have a contradiction to dinaturality.

These counterexamples show that dinaturality is wrong for simple types in this model, thus:

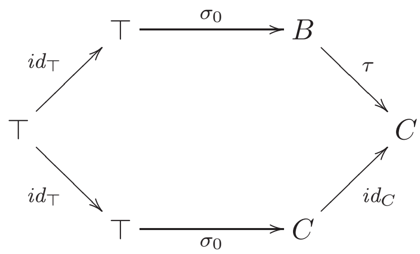
Theorem 2 *The hyperdoctrine $\mathcal{G}^*$ is not Strachey parametric.*

This is coherent with the fact that we do not have any uniformity property in the model. So, once again, the criterion seems to be in agreement with the intuitive idea behind Strachey parametricity. Can we say then that it perfectly matches the intuition ? We shall see in the next section that this is not the case: the property is too strong for some models or syntaxes that are intuitively Strachey parametric.

5 The question of totality

In section 3 we have required our strategies to be winning, that is, total and finite (i.e. the set of views of the symbolic strategy is finite). This property was used to prove our central result, Prop. 4. However, it is not a necessary property to build a game model of system F. One can then wonder what happens if we suppress this property from the model; so, from now on, the strategies will still be considered as uniform, but not winning anymore.

It is easily seen that if we do that we lose the dinaturality on simple types. Indeed, consider the strategy $\sigma_0 = \{\epsilon\}$, which can now be seen as a strategy on any arena A. In particular, $\sigma_0 : \top \to X_1$. The dinaturality property for simple types would imply:



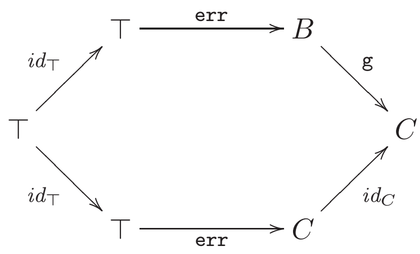
for any $\tau : B \to C$. Now take the strategy τ such that $c_1 c_2 \in \tau$ with c_1 and c_2 of shape $\{\uparrow\}$. Then the upper morphism of the diagram contains at least the play $c_1 c_2$, whereas the lower morphism is just σ_0.

Note however that the two strategies do not contradict themselves, and this will be a general fact in this model: if both strategies give an answer to a move by **O** (provided a given history of moves), then they give the same answer. Otherwise said, the upper strategy σ_1 and the lower strategy σ_2 are compatible, in the sense that $\ulcorner\sigma_1\urcorner \cup \ulcorner\sigma_2\urcorner$ is a compatible set. It would then be tempting to consider a uniform model with strategies defined "up to compatibility". But compatibility between strategies is not a transitive relation...

This failure of dinaturality is a rather puzzling fact because it implies that the model with uniform, but not necessarily winning, strategies does not form a Strachey parametric hyperdoctrine. Whereas the strategies of the model still behave in a proper, intuitively parametric, way. This means that our criterion is not adequate for this model: it is intuitively parametric, without satisfying the criterion. More generally, our criterion can be seen as a sufficient, but not necessary condition for intuitive Strachey parametricity.

Interestingly, and as it is the case with many game semantics features, this model has an equivalent in syntax: it corresponds to a syntax where errors, or loops, can occur. Let us consider for illustration a language containing system F and a term `err` such that $\Gamma \vdash \texttt{err} : A$ for any type A and any context Γ, and such that it provides an exception any time it is called. Adding this new term does not change the intuitively parametric behaviour of the language.

Then the dinaturality diagram is similar to the one we had with games:



Now choose $C = X \to X$ and $\texttt{g} = \lambda y^X.y$, so that $x : B \vdash \texttt{g} : C$. The upper morphism will be `g` whereas the lower morphism will be `err`. So we have a syntax that do not respect dinaturality for simple types, whereas it is intuitively parametric. Note that similar examples could be built by using terms corresponding to non-finite strategies (finiteness is in fact connected with totality, as a non-finite strategy may interact with another one to produce a non-total strategy).

As a conclusion, we want to stress the fact that the criterion we analysed in this paper is just a tentative criterion for intuitive Strachey parametricity, and might be refined in the future by a more flexible criterion. The requirement of dinaturality, even restricted to simple types, is indeed rather strong, and it might fail for models with a "uniformity" condition, even for reasons not related to totality and finiteness.

This is however a criterion which is true for system F, unlike most known parametricity characterisations[11], and it has an interesting relationship with second order quantification, although it concerns only simple types. It will be fruitful to study the validity of this criterion for a larger variety of models, and it might lead to define a more general criterion for Strachey parametricity.

References

[ACC93] Martín Abadi, Luca Cardelli, and Pierre-Louis Curien. Formal parametric polymorphism. In *Conf. Record 20th Ann. ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL'93, Charleston, SC, USA, 10–13 Jan 1993*, pages 157–170. ACM Press, New York, 1993.

[AJ03] Samson Abramsky and Radha Jagadeesan. A game semantics for generic polymorphism. In Andrew D. Gordon, editor, *Foundations of Software Science and Computational Structures*, volume 2620 of *LNCS*, pages 1–22. Springer, 2003.

[AM99] Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *Proceedings of the fourteenth annual symposium on Logic In Computer Science*, pages 431–442, Trento, July 1999. IEEE, IEEE Computer Society Press.

[BFSS90] E. S. Bainbridge, Peter J. Freyd, Andre Scedrov, and Philip J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70(1):35–64, 1990.

[dL07] Joachim de Lataillade. *Quantification du second ordre dans les jeux - Application aux isomorphismes de types*. Thèse de Doctorat, Université Paris 7, novembre 2007.

[dL08a] Joachim de Lataillade. Curry-style type isomorphisms and game semantics. To appear in *Mathematical Structures of Computer Science*, Special Issue on Type Isomorphisms, 2008.

[dL08b] Joachim de Lataillade. Second-order type isomorphisms through game semantics. *Annals of Pure and Applied Logic*, Special Issue on Game Semantics, 2008.

[FGSS88] Peter J. Freyd, Jean-Yves Girard, Andre Scedrov, and Philip 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.

[11] We already explained the case of relational parametricity and dinaturality. Another classical criterion which does not stand for Church-style system F, although it is closer to be true, is the genericity property from [LMS93].

[Gir72] Jean-Yves Girard. *Interprtation fonctionnelle et limination des coupures de l'arithmtique d'ordre suprieur.* Thèse de doctorat, Universit Paris VII, 1972.

[GSS92] Jean-Yves Girard, André Scedrov, and Phil 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.

[HO00] Martin Hyland and Luke Ong. On full abstraction for PCF. *Information and Computation*, 163(2):285–408, December 2000.

[Hug00] Dominic Hughes. *Hypergame semantics: full completeness for system F.* D.Phil. thesis, Oxford University, 2000.

[Law70] F. William Lawvere. Equality in hyperdoctrines and the comprehension schema as an adjoint functor. In *Proceedings on Applications of Categorical Logic*, 1970.

[LMS93] Giuseppe Longo, Kathleen Milsted, and Sergei Soloviev. The genericity theorem and the notion of parametricity in the polymorphic λ-calculus. In *Proc. of 8th IEEE Ann. Symp. on Logic in Computer Science, LICS'93, Montreal, Canada, 19–23 Jun. 1993*, pages 1–6. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[McC96] Guy McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types.* Ph.D. thesis, Department of Computing, Imperial College, University of London, 1996.

[MO01] Andrzej Murawski and Luke Ong. Evolving games and essential nets for affine polymorphism. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications '01*, volume 2044 of *LNCS*. Springer, 2001.

[PA93] Gordon Plotkin and Martín 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.

[Pit88] Andrew Pitts. Polymorphism is set-theoretic constructively. In D. Pitt, editor, *CTCS*, volume 283 of *LNCS*, 1988.

[Rey74] John C. Reynolds. Towards a theory of type structure. In *Programming Symposium, Proceedings Colloque sur la Programmation*, pages 408–423, London, UK, 1974. Springer-Verlag.

[Rey83] John C. Reynolds. Types, abstraction and parametric polymorphism. In *International Federation for Information Processing Congress*, pages 513–523, 1983.

[See87] Robert A. G. Seely. Categorical semantics for higher-order polymorphic lambda-calculus. *Journal of Symbolic Logic*, 52(4):969–989, 1987.

[Str00] Christopher Strachey. Fundamental concepts in programming languages (reprinting of a course given in 1967). *Higher-Order and Symbolic Computation*, 13(1):11–49, 2000.

[Wad89] Philip 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.