Under consideration for publication in Math. Struct. in Comp. Science

Curry-style Type Isomorphisms and Game Semantics

Joachim de Lataillade

Preuves Programmes Systèmes

University Paris 7 - Denis Diderot

Received 22 September 2007

Curry-style system F, i.e. system F with no explicit types in terms, can be seen as a core presentation of polymorphism from the point of view of programming languages.

This paper gives a characterisation of type isomorphisms for this language, by using a game model whose intuitions comes both from the syntax and from the game semantics universe. The model is composed of: an untyped part to interpret terms, a notion of arena to interpret types, and a typed part to express the fact that an untyped strategy σ plays on an arena A.

By analysing isomorphisms in the model, we prove that the equational system corresponding to type isomorphisms for Curry-style system F is the extension of the equational system for Church-style isomorphisms with a new, non-trivial equation: $\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if X appears only positively in A.

1. Introduction

Types isomorphisms. The problem of type isomorphisms is a purely syntactical question: two types A and B are isomorphic if there exist two terms $f : A \to B$ and $g : B \to A$ such that $f \circ g = id_B$ and $g \circ f = id_A$. This equivalence relation on data types allows to translate a program from one type to the other without any change on the calculatory meaning of the program. Thus, a search in a library up to type isomorphism will help the programmer to find all the functions that can potentially serve his purpose, and to reuse them in the new typing context (Rittri, 1991). This is particularly appealing with functional languages, because in this case the type can really be seen as a partial specification of the program: such a library search up to isomorphisms has been implemented in particular for Caml Light by Jérôme Vouillon. It can also be used in proof assistants to help finding proofs in libraries and reusing them (Barthe and Pons, 2001) (for more details on the use of type isomorphisms in computer science, see (Di Cosmo, 1995)). From a more general point of view, type isomorphisms are the natural answer to the question of equivalence between types in a programming language.

The question of characterising these type isomorphisms is then a very simple problem to formulate, however its resolution is often non-trivial, especially when dealing with polymorphism like in system F (Girard, 1972; Reynolds, 1974). Roberto Di Cosmo has solved syntactically this question for *Church-style* system F (i.e. system F where types appear explicitly in the terms) by giving an equational system on types equivalent to type isomorphisms (Di Cosmo, 1995). In a preceding work (de Lataillade, 2007), we have given a new proof of this result by using a game semantics model of Church-style system F. In this more geometrical approach, types were interpreted by an arborescent structure, *hyperforests*: the natural equality for this structure happened to be exactly the equality induced by type isomorphisms. The efficiency of game semantics in this context was an incitement to go further and to explore the possibility of resolving this question for other languages.

Curry-style system F. In the present work, we deal with type isomorphisms for *Curry-style* system F, i.e. system F where the terms grammar is simply the grammar of the untyped λ-calculus. Although this system appears to be less relevant than Church-style system F in proof-theory (a term does not correspond exactly to one proof), it is actually more accurate when we consider programming languages. Indeed, in Church-style system F, a term t of type $\forall X.A$ will not have the type $A[B/X]$: only $t\{B\}$ will be of this type; whereas in Curry-style, a term t of type $\forall X.A$ will have all the types $A[B/X]$, which is more the idea induced by the notion of polymorphism: the same function may be used with different types. The typing rules and equalities of this language are presented on figure 1, where $X \notin \Gamma$ (resp. $x \notin t$) means that the type variable X (resp. the variable x) does not appear freely in Γ (resp. in t).

Compared with this system, Church-style system F has a different grammar of terms:

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

different typing rules for the quantification:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \Lambda X.t : \forall X.A}\ (\forall I) \ \text{ if } X \notin \Gamma \qquad \frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t\{B\} : A[B/X]}\ (\forall E)$$

and two additional equalities:

$$\begin{array}{rclll} (\Lambda X.t)\{A\} & = & t[A/X] & & (\beta 2) \\ \Lambda X.t\{X\} & = & t & \text{if } X \notin t & (\eta 2) \end{array}$$

As can be seen on the typing rules, a λ-term t is of type A if there exists a term $\tilde{t}$ of Church-style system F such that t is obtained from $\tilde{t}$ by erasing all the type indications (for example, $\Lambda X.\lambda x^{\forall Y.Y}\lambda y^Y.x\{Y\}$ becomes $\lambda x \lambda y.x$). In this case, we say that t is the *erasure* of $\tilde{t}$.

The characterisation of type isomorphisms for Curry-style system F is not directly reducible to corresponding question for the Church-style system F: indeed, types of the form $\forall X.A$ and A with $X \notin A$ are not equivalent in the Church-style setting, but they are in the Curry-style one (where the isomorphism is realised by the identity). We prove in this paper that the distinction between Church-style and Curry-style type isomorphisms

Grammars:

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

Typing rules:

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

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

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

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

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

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

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

Equalities:

$$\begin{array}{rcll} (\lambda x.t)u & = & t[u/x] & & (\beta) \\ \lambda x.tx & = & t & \text{if } x \notin t & (\eta) \\ \pi_1(\langle u, v\rangle) & = & u & & (\pi_1) \\ \pi_2(\langle u, v\rangle) & = & v & & (\pi_2) \\ \langle \pi_1(u), \pi_2(u)\rangle & = & u & & (\times) \end{array}$$

Type isomorphism:

$$(t, u) \text{ s.t. } \begin{cases} \vdash t : A \to B \\ \vdash u : B \to A \\ \lambda x.t(ux) = \lambda x.u(tx) = \lambda x.x \end{cases}$$

Fig. 1. Curry-style system F

can be resumed in one new and non-trivial equation. To express it, one first has to recall the definition of positive and negative type variables in a formula†:

† All along this article we will identify the notions of *type* and *formula* (according to the Curry-Howard correspondence).

Definition 1. If A is a formula, its sets of **positive variables** Pos_A and **negative variables** Neg_A are defined by:

— $Pos_X = \{X\}$, $Neg_X = \emptyset$
— $Pos_\bot = Neg_\bot = \emptyset$
— $Pos_{A\times B} = Pos_A \cup Pos_B$, $Neg_{A\times B} = Neg_A \cup Neg_B$
— $Pos_{A\to B} = Neg_A \cup Pos_B$, $Neg_{A\to B} = Pos_A \cup Neg_B$
— $Pos_{\forall X.A} = Pos_A \setminus \{X\}$, $Neg_{\forall X.A} = Neg_A \setminus \{X\}$

We also define $FTV(A) = Pos_A \cup Neg_A$.

The new equation is then the following:

$$\forall X.A \simeq_\varepsilon A[\forall Y.Y/X] \qquad \text{if } X \notin Neg_A$$

It is true in Curry-style but false (in general) in Church-style system F. Note that, although the isomorphism is realised by the identity, the Church-style terms $t : \forall X.A \to A[\forall Y.Y/X]$ and $u : A[\forall Y.Y/X] \to \forall X.A$, from which we extract the identity by erasing explicit types, are not trivial (they will be explicitly described in the proof of theorem 2 at the end of the paper). This is a difference with Church-style system F, where type isomorphisms were exactly the expected ones, even if proving that point was not an elementary task.

Type isomorphisms for Curry-style system F are finally characterised by the following equational system:

$$\begin{aligned}
A \times B &\simeq_\varepsilon B \times A \\
A \times (B \times C) &\simeq_\varepsilon (A \times B) \times C \\
A \to (B \to C) &\simeq_\varepsilon (A \times B) \to C \\
A \to (B \times C) &\simeq_\varepsilon (A \to B) \times (A \to C) \\
\forall X.\forall Y.A &\simeq_\varepsilon \forall Y.\forall X.A \\
A \to \forall X.B &\simeq_\varepsilon \forall X.(A \to B) && \text{if } X \notin FTV(A) \\
\forall X.(A \times B) &\simeq_\varepsilon \forall X.A \times \forall X.B \\
\forall X.A &\simeq_\varepsilon A[\forall Y.Y/X] && \text{if } X \notin Neg_A
\end{aligned}$$

The purpose of this paper is to prove correctness and completeness of this characterisation by using a game model.

The model. Models of second order calculi do not come about easily due to impredicativity. Among the different possibilities, we choose models based on game semantics because of their high degree of adequation with the syntax: indeed, game semantics has been widely used to construct fully complete models for various calculi, such as PCF (Abramsky et al., 2000; Hyland and Ong, 2000), μPCF (Laird, 1997), Idealized Algol (Abramsky and McCusker, 1999), etc. This means that this semantics gives a very faithful description of the behaviour of the syntax modulo reduction rules in the system. And this is precisely what we need to deal semantically with type isomorphisms: a model which is so precise that it contains no more isomorphisms than the syntax.

The present paper introduces a game model for Curry-style system F. This model was largely inspired by two preceding game semantics works: the PhD thesis of Juliusz

Chroboczek (Chroboczek, 2003), which presents among others a game semantics for an untyped calculus that we reuse in this paper; and the game semantics model for generic polymorphism by Samson Abramsky and Radha Jagadeesan (Abramsky and Jagadeesan, 2003), from which we will extract many ideas in our context. Other game semantics models had an influence on our work: Dominic Hughes gave the first game models of Church-style system F (Hughes, 2000) and introduced the notion of *hyperforests* that we reuse here; Andrzej Murawski and Luke Ong presented a simple and efficient model for dealing with affine polymorphism (Murawski and Ong, 2001), and their presentation of moves inspired ours.

It shall be noticed that the design of our Curry-style game model is actually very connected to the concepts present in the syntax: the notion of erasure we introduce is of course reminiscent of the erasure of types in a Church-like term to obtain a Curry-like term. This is no surprise as we need a model describing very precisely the syntax (that is why, in particular, one cannot be satisfied by an interpretation of the quantification as an intersection or a greatest lower bound). The specificities of (HO-)game semantics, as for example the arborescent structure that interprets types, are however decisive for our demonstration.

Finally, the model we present in this paper is not a model for every equalities of Curry-style system F: the η-rule and the subjective pairing are not preserved by the interpretation. However, if they are oriented as reduction rules, they are interpreted by an inclusion in the model. One part of the job is to prove that this modelisation is strong enough to capture type isomorphisms.

2. Prolegomena on game semantics

Game semantics is the main technical tool used in this work. But as we will focus later on very specific details related with the introduction of polymorphism, we present in this section the basics on HO-style game semantics and its key ingredients[‡]:

— **arenas** which interpret types geometrically
— **strategies** which interpret terms dynamically
— the property of **innocence**, a constraint on strategies which connects them stronger with syntax
— the notion of **interaction** which determines the way that two strategies can combine to generate a new one.

The definitions given below correspond to the standard presentation of games as a modelisation of the language PCF or, more modestly, of the simply typed λ-calculus. As most of the notions will be redefined later in a polymorphic setting, we will refer to these games as HO-games, and use the terminology HO-arenas, HO-strategies, etc.

[‡] For a more complete description of these games, see (Hyland and Ong, 2000) or (Harmer, 1999).

2.1. *HO-Arenas*

Game semantics is based on a conflict between two players, Player **P** and Opponent **O**. Intuitively, the Player is the program in execution, whereas the Opponent is its environment (e.g. the value of its parameters). The two players interact by playing **moves** in an **arena**. More formally, if we note $\mathcal{X}$ the set of type variables:

Definition 2 (HO-arena). An **HO-arena** $A = (E_A, \lambda, \vdash, D)$ is a set E_A of **moves** together with a function of **polarity** $\lambda : E_A \to \{\mathbf{O}, \mathbf{P}\}$, a partial function of **decoration** $D : E_A \rightharpoonup \mathcal{X}$ and a relation of **justification** $\vdash \subseteq E_A + (E_A \times E_A)$, with the following properties:
— there is no cycle $x \vdash x_1 \vdash \cdots \vdash x_n \vdash x$
— if $\vdash x$ then $\lambda(x) = \mathbf{O}$
— if $x \vdash y$ then $\lambda(x) \neq \lambda(y)$.

The polarity indicates by which player a move can be played. The relation of justification says the following: if $\vdash m$ then m can be played by its player (necessarily **O**) without restriction; if $m \vdash n$ then n can be played only if m has been played earlier (necessarily by the other player). The decoration is not standard in HO-style games: it corresponds to the indication of a type variable; for example, in the arena corresponding to $X \to Y$ there will be two moves, one with the decoration X and the other with the decoration Y.

On a given HO-arena, we can define a set of **initial moves** $I_A \subseteq E_A$ by:

$$I_A = \{x \in E_A \mid \vdash x\}$$

and a partial order $<$ by:

$$x < y \Leftrightarrow \exists z_1, \ldots, z_n,\ x \vdash z_1 \vdash \cdots \vdash z_n \vdash y$$

We can recover the relation of justification if we know only I and $<$:
— $\vdash x$ iff $x \in I$
— $x \vdash y$ iff $x < y$ and $x \leq z \leq y \Rightarrow (z = x \vee z = y)$.

2.2. *Constructions on arenas*

In what follows, we note $E + F$ the disjoint union between two sets E and F. If $f : E \rightharpoonup G$ and $g : F \rightharpoonup G$ are two partial functions then $[f, g] : E + F \rightharpoonup G$ is defined by:

$$[f,g](x) = \begin{cases} f(x) & \text{if } x \in E \text{ and } f(x) \text{ is defined} \\ g(x) & \text{if } x \in F \text{ and } g(x) \text{ is defined} \end{cases}$$

The **aromic HO-arenas** are:
— $\top = (\emptyset, \emptyset, \emptyset, \emptyset)$
— $\bot = (\{\bullet\}, \bullet \mapsto \mathbf{O}, \{\bullet\}, \emptyset)$
— $X = (\{\bullet\}, \bullet \mapsto \mathbf{O}, \{\bullet\}, \bullet \mapsto X)$.

Given two HO-arenas $A = (E_A, \lambda_A, \vdash_A, D_A)$ and $B = (E_B, \lambda_B, \vdash_B, D_B)$,we define the **product HO-arena** $A \times B = (E_{A\times B}, \lambda_{A\times B}, \vdash_{A\times B}, D_{A\times B})$ and the **arrow HO-arena** $A \to B = (E_{A\to B}, \lambda_{A\to B}, \vdash_{A\to B}, D_{A\to B}))$ by:

— $E_{A\times B} = E_A + E_B$ $\quad E_{A\to B} = E_A + E_B$
— $\lambda_{A\times B} = [\lambda_A, \lambda_B]$ $\quad \lambda_{A\to B} = [\overline{\lambda}_A, \lambda_B]$
— $\vdash_{A\times B} = \vdash_A + \vdash_B$ $\quad \vdash_{A\to B} = (\vdash_A \backslash I_A) + \vdash_B + (I_B \times I_A)$
— $D_{A\times B} = [D_A, D_B]$ $\quad D_{A\to B} = [D_A, D_B]$

where $\overline{\lambda}(x) = \begin{cases} \mathbf{O} & \text{if } \lambda(x) = \mathbf{P} \\ \mathbf{P} & \text{if } \lambda(x) = \mathbf{O} \end{cases}$

2.3. *Plays and strategies*

Definition 3 (HO-justified sequence, HO-play). An **HO-justified sequence** on an HO-arena $A = (E_A, \lambda, \vdash, D)$ is given by a sequence $s = m_1 \dots m_n$ of elements of E_A and 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 $m_j \vdash m_i$ (we then say that m_i **justifies** m_j, or that there is a **pointer** from m_j to m_i).

An **HO-play** is a HO-justified sequence $s = x_1 \dots x_n$ such that: for every $1 \leq i \leq n$,

— if $\lambda(x_i) = \mathbf{P}$ then $\lambda(x_{i+1}) = \mathbf{O}$
— if $\lambda(x_i) = \mathbf{O}$ then $\lambda(x_{i+1}) = \mathbf{P}$ and $D(x_{i+1}) = D(x_i)$ (which also means that $D(x_{i+1})$ is undefined if $D(x_i)$ is undefined).

We note $\mathcal{P}_A^{HO}$ (resp. $\mathcal{E}_A^{HO}$) the set of HO-plays (resp. the set of even-length HO-plays) on A.

Let s and t be two HO-justified sequences on A, we write $t \leq s$ if t is a prefix of s.

Definition 4 (HO-strategy). An **HO-strategy** σ on an HO-arena $A = (E_A, \lambda, \vdash, D)$ is a non-empty set of even-length HO-plays on A, closed by even-length prefix and deterministic: if sm and sn are two HO-plays of σ then $sm = sn$.

If σ is an HO-strategy on A, we note $\sigma : A$.

An HO-strategy is simply the description of a behaviour of **P**: given a certain history and a move of **O**, it gives a unique move played by **P** as a response. HO-style game model interpret terms as strategies, thus confirming the above intuition of identifying **P** with the program or, more precisely, with the behaviour of **P**.

Buy we do not have at this point a strict identification between terms of the syntax and HO-strategies: there is still too much freedom in the behavior of **P**. The first thing to add to constraint it is a property of **innocence**:

Definition 5 (view, innocence). Let s be an HO-play, its **view** $\ulcorner s \urcorner$ is defined 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 $\lambda(m) = \mathbf{O}$ and m justifies n.

An HO-strategy $\sigma : A$ is called **innocent** if, for any HO-play sm of σ, the move which justifies m is in $\ulcorner s \urcorner$, and if we have: for any $smn \in \sigma$, $t \in \sigma$, if tm is a play on A and $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ then $tmn \in \sigma$.

According to that definition, what really matters in an innocent HO-strategy is its set of views. However, considering all the HO-plays of an HO-strategy will become necessary when dealing with composition.

HO-trategies that correspond to λ-terms are **winning HO-strategies**:

Definition 6 (winning HO-strategy). An HO-strategy σ on the HO-arena $A = (E_A, \lambda, \vdash, D)$ is called **winning** if the following conditions are satisfied:

— σ is innocent
— the set of views of σ is finite
— σ is **total** on A: for every HO-play $s \in \sigma$, if $sm \in \mathcal{P}_A^{HO}$ then there exists a move $n \in E_A$ such that $smn \in \sigma$.

It can then be proved (cf. (Hyland and Ong, 2000)) that, if an HO-arena A comes from a type T, any winning strategy σ on A is the interpretation of a simply typed λ-term t of type T. A similar result of completeness can be given for games interpreting Church-style system F (see (Hughes, 2000)), but in the case of the present paper completeness is not a necessary result.

2.4. *Interaction*

The central mechanism of game semantics is the notion of **interaction**: two HO-strategies $\sigma : A \to B$ and $\tau : B \to C$ can interact to give rise to a new HO-strategy $\sigma;\tau : A \to C$, the **composition** of σ and τ.

The intuitive idea is the following: as the polarity on B is reversed in $B \to C$, the HO-strategy τ will play the role of the Opponent on this part, whereas σ will play the role of Player: this generates a dialogue between the two players on the common part B. Then, to obtain a strategy on $A \to C$, it suffices to keep record of the dialogue on A and C, and to forget the part of the dialogue played in B.

This dialogue interaction is the reason why we say that game semantics offers a *dynamical* approach to cut-elimination; in fact, this dynamical process is a feature from which we will take great advantage in the present work.

To express these ideas more formally, we begin with the **restriction** of an HO-justified sequence to a part of its HO-arena:

Definition 7 (restriction). Let s be an HO-justified sequence on an HO-arena $A = (E_A, \lambda_A, \vdash_A, D_A)$, and let $B = (E_B, \lambda_B, \vdash_B, D_B)$ an HO-arena such that $E_B \subseteq E_A$, $\leq_B \subseteq \leq_A$ and λ_A (resp. D_A) coincides with λ_B (resp. D_B) on E_B. Then the **restriction** of s to B, denoted $s\upharpoonright_B$, is the HO-justified subsequence of s obtained by keeping only the moves of E_B, with the associated pointers as long as we stay in E_B.

Suppose now that there is another HO-arena $C = (E_C, \lambda_C, \vdash_C, D_C)$ such that $E_C \subseteq E_A$, $\leq_C \subseteq \leq_C$ and λ_A (resp. D_A) coincides with λ_C (resp. D_C) on E_C. Then the **restriction** of s to B, C, denoted $s\upharpoonright_{B,C}$, is the HO-justified subsequence of s obtained by keeping only the moves of E_B or E_C, with the associated pointers as long as we stay in E_B or in E_C, and additionnal pointers from moves m_1 of E_B to moves m_2 of E_C if there exists a move $m_3 \in E_A \backslash (E_B \cup E_C)$ such that m_3 justifies m_1 and m_2 justifies m_3.

We can now define composition:

Definition 8 (HO-interaction sequence, composition). An **HO-interaction sequence** between A, B and C is an HO-justified sequence s on $(A \to B) \to C$ such that $s\upharpoonright_{A,B}$, $s\upharpoonright_{B,C}$ and $s\upharpoonright_{A,C}$ are HO-plays. The set of HO-interaction sequences between A, B and C is denoted $\mathbf{Int}^{HO}(A,B,C)$.

Let σ and τ be two strategies. We call **composition** of σ and τ the set

$$\sigma;\tau = \{u\upharpoonright_{A,C} \mid u \in \mathbf{Int}^{HO}(A,B,C),\ u\upharpoonright_{B,C} \in \tau \textit{ and } u\upharpoonright_{A,B} \in \sigma\}$$

It can then be proved (cf. (Hyland and Ong, 2000; Harmer, 1999)) that $\sigma;\tau$ is a strategy.

2.5. *Category of HO-games*

To conclude this introductory section on game semantics, we present how to buil a cartesian closed category of HO-games. The interpretation of the syntax is actually given most of the time through this category.

The **identity HO-strategy** on the HO-arena A is

$$id_A^{HO} = \{s \in \mathcal{E}^{HO}_{A_1 \to A_2} \mid \forall t \in \mathcal{E}^{HO}_{A_1 \to A_2}, t \leq s \Rightarrow t\upharpoonright_{A_1} = t\upharpoonright_{A_2}\}$$

where A_1 (resp. A_2) is the left (resp. right) copy of A in the disjoint union $A + A$.

The category of (innocent) HO-games is defined as follows:

— objects are the HO-arenas
— a morhism between A and B is an innocent HO-strategy on $A \to B$
— the identity on A is id_A^{HO}
— the composition of $\sigma : A \to B$ and $\tau : B \to C$ is $\sigma;\tau : A \to C$.

This category is equipped with the following morphisms:

— trivial strategy:

$$\diamond^{HO} = \{\epsilon\} : A \to \top$$

— projections:

$$\pi_A^{HO} = \{s \in \mathcal{E}^{HO}_{A \times B_1 \to B_2} \mid \forall t \in \mathcal{E}^{HO}_{A \times B_1 \to B_2}, t \leq s \Rightarrow t\upharpoonright_{B_1} = t\upharpoonright_{B_2}\} : A \times B \to B$$

$$\pi_B^{HO} = \{s \in \mathcal{E}^{HO}_{A_1 \times B \to A_2} \mid \forall t \in \mathcal{E}^{HO}_{A_1 \times B \to A_2}, t \leq s \Rightarrow t\upharpoonright_{A_1} = t\upharpoonright_{A_2}\} : A \times B \to A$$

— pairing: if $\sigma : A \to B$ and $\tau : A \to C$,

$$\langle \sigma, \tau \rangle = \{s \in \mathcal{E}^{HO}_{A \to (B \times C)} \mid s\upharpoonright_{A \to B} \in \sigma \text{ and } s\upharpoonright_{A \to C} \in \tau\} : A \to (B \times C)$$

— evaluation :

$$\begin{aligned} eval^{HO} \quad = \quad & \{s \in \mathcal{E}^{HO}_{(A_1 \to B_1) \times A_2 \to B_2} \mid \forall t \in \mathcal{E}^{HO}_{(A \to B) \times A \to B}, \\ & t \leq s \Rightarrow t\upharpoonright_{A_1} = t\upharpoonright_{A_2} \wedge t\upharpoonright_{B_1} = t\upharpoonright_{B_2}\} \quad : (A \to B) \times A \to B \end{aligned}$$

— abstraction : if $\sigma : A \times B \to C$, $\Lambda(\sigma)$ is the HO-strategy on $A \to (B \to C)$ whose HO-plays are the same as those of σ, but seen as HO-plays on $A \to (B \to C)$

These morphisms make the category cartesian closed, as proved in (Harmer, 1999). Note that if we forget the property of innocence, we loose the cartesian closed structure.

So, the somewhat sophisticated definitions of HO-style game semantics describe simple and efficient geometrical intuitions to modelise the syntax. This is the reason why games are used in the present work as a semantic tool: the strong intuitions they give can enlight even difficult syntactic questions.

3. General definitions

In this section we give general constructions that will apply on the different grammars we use in the model. These constructions are derived from usual HO-games notions given above.

3.1. *Moves*

We consider the set of type variables $X,\ Y,\ \ldots$ to be in bijection with $\mathbb{N}\backslash\{0\}$, and we will further write this set $\mathcal{X} = \{X_j \mid\ j > 0\}$.

All along this article, we define several grammars of the form:

$$\mu ::= \uparrow\mu \mid\ \downarrow\mu \mid\ \alpha_i\mu \mid\ j \qquad (i \in I,\ j \in \mathbb{N})$$

Let us write $\mathcal{M}$ for the set of words (often called **moves**) defined by this grammar.

Intuitively, the token $\uparrow$ (resp. $\downarrow$) corresponds to the right side (resp. the left side) of an arrow type, the α_i's are related to additional (covariant) connectors, the constants $j \in \mathbb{N}\backslash\{0\}$ correspond to free type variables X_j and the constant 0 corresponds either to bound type variables or to $\bot$.

On such a grammar, we define automatically a function λ of **polarity**, with values in $\{\mathbf{O}, \mathbf{P}\}$:

— $\lambda(j) = \mathbf{O}$
— $\lambda(\uparrow\mu) = \lambda(\alpha_i\mu) = \lambda(\mu)$
— $\lambda(\downarrow\mu) = \overline{\lambda}(\mu)$

where $\overline{\mathbf{O}} = \mathbf{P}$ and $\overline{\mathbf{P}} = \mathbf{O}$.

We also introduce an **enabling relation** $\vdash\ \subseteq \mathcal{M} \cup (\mathcal{M} \times \mathcal{M})$:

— $\vdash j$
— if $\vdash \mu$ then $\vdash \alpha_i\mu$, and $\vdash \uparrow\mu$
— if $\vdash \mu$ and $\vdash \mu'$ then $\uparrow\mu \vdash \downarrow\mu'$
— if $\mu \vdash \mu'$ then $\alpha_i\mu \vdash \alpha_i\mu'$, $\uparrow\mu \vdash \uparrow\mu'$ and $\downarrow\mu \vdash \downarrow\mu'$.

which induces a partial order $\leq$ for this grammar by reflexive and transitive closure. If $\vdash \mu$ we say that μ is an **initial move** (in which case $\lambda(\mu) = \mathbf{O}$).

3.2. *Substitution*

As we want to deal with polymorphism, we need some operations acting directly on the leafs j:

— a function $\sharp$ of **leaf extracting**:

- $\sharp(j) = j$
- $\sharp(\uparrow\mu) = \sharp(\downarrow\mu) = \sharp(\alpha_i\mu) = \sharp(\mu)$

— an operation of **substitution** $\mu[\mu']$:

- $j[\mu'] = \mu'$
- $\uparrow\mu[\mu'] = \uparrow(\mu[\mu'])$, $\downarrow\mu[\mu'] = \downarrow(\mu[\mu'])$ and $\alpha_i\mu[\mu'] = \alpha_i(\mu[\mu'])$

We say that μ_1 is a **prefix** of μ_2 if there exists $\mu' \in \mathcal{M}$ such that $\mu_2 = \mu_1[\mu']$. This is denoted $\mu_1 \sqsubseteq^p \mu_2$.

3.3. *Plays and strategies*

Definition 9 (justified sequence, play). A **justified sequence** on a given grammar is a sequence $s = \mu_1 \dots \mu_n$ of moves, together with a partial function $f : \{1, \dots, n\} \rightharpoonup \{1, \dots, n\}$ such that: if $f(i)$ is not defined then $\vdash \mu_i$, and if $f(i) = j$ then $j < i$ and $\mu_j \vdash \mu_i$: in this case we say that μ_j **justifies** μ_i, or that there is a **pointer** from μ_j to μ_i.

A **play** on a grammar is a justified sequence $s = \mu_1 \dots \mu_n$ on this grammar such that, for every $1 \leq i \leq n-1$:

— if $\lambda(\mu_i) = \mathbf{P}$ then $\lambda(\mu_{i+1}) = \mathbf{O}$
— if $\lambda(\mu_i) = \mathbf{O}$ then $\lambda(\mu_{i+1}) = \mathbf{P}$ *and* $\sharp(\mu_i) = \sharp(\mu_{i+1})$.

We note $\mathbb{E}$ the set of plays of even length. If s and t are two plays, we note $t \leq s$ if t is a prefix of s.

The definition of a play implies that if $s\mu\nu$ is an even-length play then $\sharp(\mu) = \sharp(\nu)$. This will be a very significant property in our model.

Definition 10 (strategy). A **strategy** σ on a given grammar is a non-empty set of even-length plays, which is closed under even-length prefix and deterministic: if $s\mu$ and $s\nu$ are two plays of σ then $s\mu = s\nu$.

Definition 11 (view, innocence). Let s be a play on a grammar, we define its **view** $\ulcorner s \urcorner$ by:

— $\ulcorner \varepsilon \urcorner = \varepsilon$
— $\ulcorner s\mu \urcorner = \ulcorner s \urcorner \mu$ if $\lambda(\mu) = \mathbf{P}$
— $\ulcorner s\mu \urcorner = \mu$ if $\vdash \mu$
— $\ulcorner s\mu t\nu \urcorner = \ulcorner s \urcorner \mu\nu$ if $\lambda(\nu) = \mathbf{O}$ and μ justifies ν

A strategy σ is called **innocent** if, for every play $s\nu$ of σ, the justifier of ν is in $\ulcorner s \urcorner$, and if we have: if $s\mu\nu \in \sigma$, $t \in \sigma$, $t\mu$ is a play and $\ulcorner s\mu \urcorner = \ulcorner t\mu \urcorner$ then $t\mu\nu \in \sigma$.

Definition 12 (bi-view). A **bi-view** on a given grammar is a justified sequence $s = \mu_1 \dots \mu_n$ (with $n \geq 1$) such that any move is justified by its predecessor. The set of bi-views is denoted $\mathcal{BV}$.

3.4. *Composition*

Composition is usually defined between strategies on arenas of the form $A \to B$ and $B \to C$. We are going to define it in a context where arenas do not explicitly exist, but are however represented by the tokens $\uparrow$ and $\downarrow$.

Definition 13 (shape). Let $\zeta \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$, a move μ is said to be of **shape** ζ if $\zeta 0 \sqsubseteq^p \mu$.

Let Σ be a finite set of elements $\zeta_j \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$. A justified sequence is said to be of shape Σ if each of its moves is of shape ζ_j for some j. A strategy is of shape Σ if each of its plays is of shape Σ.

In the case where $\Sigma = \{\uparrow, \downarrow\}$, we say that the justified sequence (or the strategy) is of **arrow shape**.

Consider a justified sequence $s = \mu_1 \ldots \mu_n$, we define the sequence $s \upharpoonright_\zeta$ as the restriction of s to the moves of shape ζ where the prefix ζ has been erased, and the pointers are given as follows: if $\mu_i = \zeta \mu'_i$ is justified by $\mu_j = \zeta \mu'_j$ in s, then the corresponding occurrence of μ'_i is justified by μ'_j.

Suppose $\zeta, \xi \in \{\uparrow, \downarrow, r, l\}^*$, and consider the restriction s' of s to the moves of shape ζ, and to the moves of shape ξ hereditarily justified by a move of shape ζ. The justified sequence $s \upharpoonright_{\zeta,\xi}$ is defined as the sequence s' where each prefix ζ (resp. ξ) has been replaced by $\uparrow$ (resp. $\downarrow$), and where pointers are defined as follows:

— if $\mu_i = \zeta \mu'_i$ (resp. $\mu_i = \xi \mu'_i$) is justified by $\mu_j = \zeta \mu'_j$ (resp. $\mu_j = \xi \mu'_j$) in s, then the corresponding occurrence of $\uparrow \mu'_i$ (resp. $\downarrow \mu'_i$) in $s \upharpoonright_{\zeta,\xi}$ is justified by $\uparrow \mu'_j$ (resp. $\downarrow \mu'_j$)
— if $\mu_i = \xi \mu'_i$ is hereditarily justified by $\mu_j = \zeta \mu'_j$ in s with $\vdash \mu'_i$ and $\vdash \mu'_j$ (μ'_j is then necessarily unique), then the corresponding occurrence of $\downarrow \mu'_i$ is justified by the corresponding occurrence of $\uparrow \mu'_j$.

Definition 14 (interacting sequence, composition). An **interacting sequence** $s = \mu_1 \ldots \mu_n$ is a justified sequence 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 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

$$\sigma; \tau = \{u \upharpoonright_{\uparrow,\downarrow\downarrow} \mid u \in \mathbf{Int},\ u \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau \textit{ and } u \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma\}$$

$\sigma; \tau$ is a strategy: this can be proven like in the standard HO-style game model. Moreover if σ and τ are innocent then $\sigma; \tau$ is innocent.

Definition 15 (totality on a shape). Let σ be a strategy and $\zeta \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$. We say that σ is **total** on the shape ζ if, for every play $s \in \sigma$ of shape ζ, for every move μ such that $s\mu$ is a play of shape ζ, there exists a move ν of shape ζ such that $s\mu\nu \in \sigma$.

3.5. *Presentation of the Curry-style model*

Our model is defined through three grammars:

— $\mathbb{X}$ is the grammar of **untyped moves** which generate the untyped model to interpret untyped lambda-terms

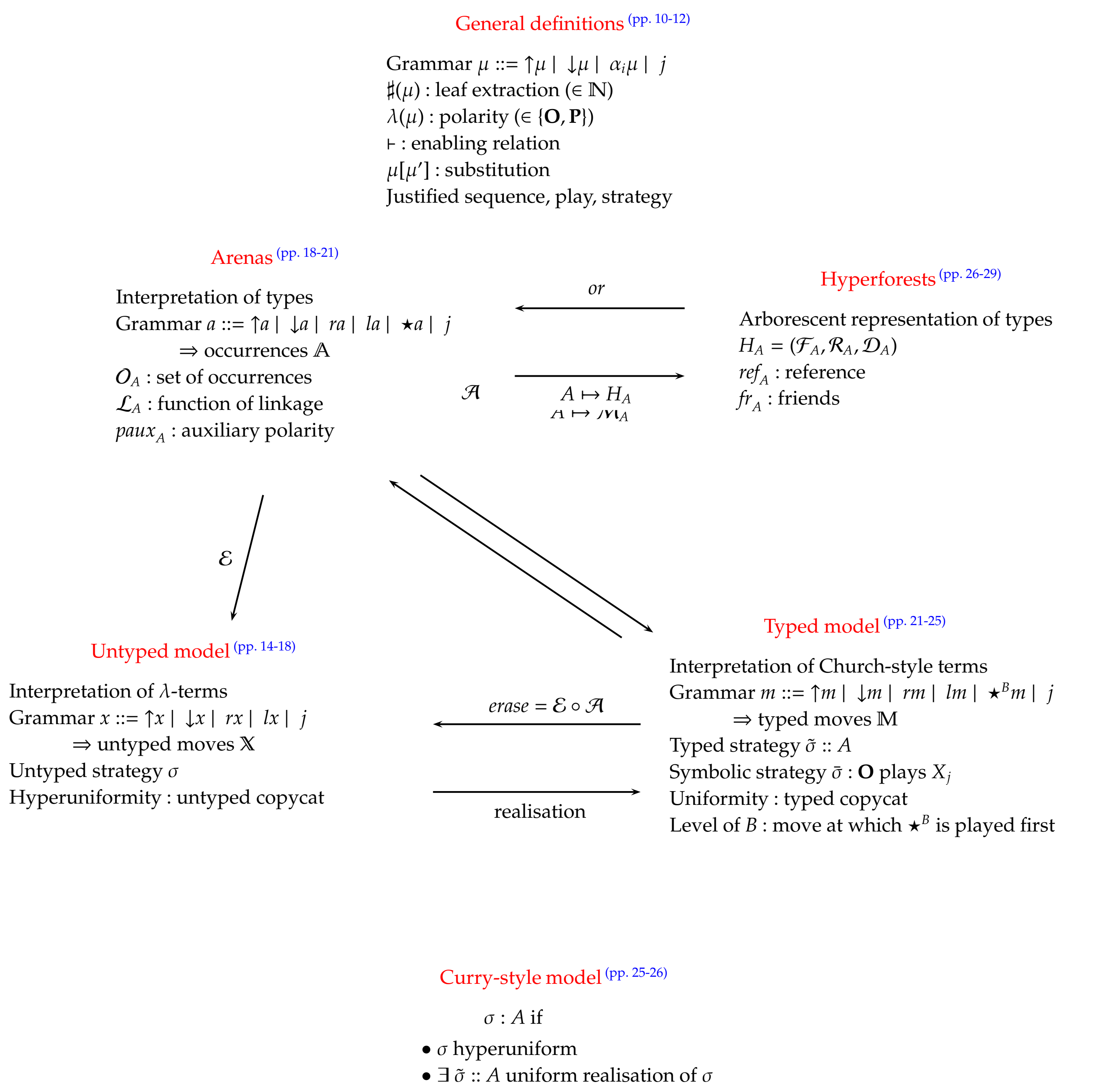


Fig. 2. Summary of the model

— $\mathbb{A}$ is the grammar of **occurrences** which are used for the interpretation of formulas
— $\mathbb{M}$ is the grammar of **typed moves** which generate an interpretation of the terms of Church-style system F.

The interpretation of Curry-style system F in the model will be as follows:

— a type A will be interpreted as an *arena* (also denoted A), i.e. a specific structure based on the grammar $\mathbb{A}$
— a term t of type A will be interpreted as a strategy σ on the grammar $\mathbb{X}$, with the condition that this strategy has a **realisation** $\tilde{\sigma}$, defined on the grammar $\mathbb{M}$ and played on the arena A (this will be denoted $\tilde{\sigma} :: A$)
— two additional properties are required: **hyperuniformity** which applies on σ, and **uniformity** which applies on $\tilde{\sigma}$.

We have summarised on figure 2 the different parts that compose the model, with the specific operations for each part that will be defined later. This diagram is intended to serve as an index to locate in which context each of these operations is defined.

In what follows, we first define the untyped model to interpret untyped lambda-terms, then we define arenas and typed strategies on arenas, and finally we introduce the notion of erasure and set up our mmodelisation of Curry-style system F. Next we prove, using this model, our result on type isomorphisms.

4. The untyped model

In this section we give a semantics for the untyped λ-calculus with binary products, i.e. for the calculus of figure 1 restricted to the language of terms with their reduction rules. This model is largely inspired by the work of Juliusz Chroboczek in his PhD thesis (Chroboczek, 2003).

Note however that we do not obtain a model in the usual sense: the equalities of the syntax are not equalities in the interpretation. But we will show that, if we give an orientation to the rules of the syntax, these oriented rules will correspond to an inclusion in the interpretation: we can say then that we have a model of the *reduction*. We will however make use of the word *model* for simplicity, keeping this restriction in mind.

4.1. *A confluent calculus*

On the Curry-style terms, we consider the rewriting system $\twoheadrightarrow$ defined by the following rules:

$$\begin{array}{rcll} (\lambda x.t)u & \twoheadrightarrow_\beta & t[u/x] & \\ \lambda x.tx & \twoheadrightarrow_\eta & t & \text{si } x \notin t \\ \langle \pi_1(t), \pi_2(t) \rangle & \twoheadrightarrow_\times & t & \\ \pi_1(\langle t, u \rangle) & \twoheadrightarrow_{\pi_1} & t & \\ \pi_2(\langle t, u \rangle) & \twoheadrightarrow_{\pi_2} & u & \end{array}$$

We know from (Klop, 1978) that this system is not confluent for the terms of λ-calculus with products. But we will show that, if we restrict ourselves to well-typed terms of the Curry-style system F, then it is confluent.

We first check that, in the Curry-style system F, if $\Gamma \vdash \langle t, u \rangle : A$ then either $A = B \times C$ or $A = \forall X.B$. Similarly, if $\Gamma \vdash \lambda x.t : A$ then either $A = B \to C$ or $A = \forall X.B$. This allows to conclude that terms of the form $\pi_1(\lambda x.t)$, $\pi_2(\lambda x.t)$ or $(\langle t, u \rangle)v$ are not well-typed in the Curry-style system F.

The system $\twoheadrightarrow$ is locally confluent: given the preceding remark, the only true critical pairs are created by the following terms:

— $(\lambda x.tx)u$ where $x \notin t$
— $\lambda x.(\lambda y.t)x$ where $x \notin t$
— $\pi_1(\langle \pi_1(t), \pi_2(t) \rangle)$
— $\pi_2(\langle \pi_1(t), \pi_2(t) \rangle)$

and they can be closed trivially.

Lemma 1. The rewriting system $\twoheadrightarrow$ terminates on well-typed Curry-style terms.

Proof. We note $\twoheadrightarrow_{\beta\pi}$ the union of the rules $\twoheadrightarrow_\beta$, $\twoheadrightarrow_{\pi_1}$ and $\twoheadrightarrow_{\pi_2}$, and $\twoheadrightarrow_{\eta\times}$ the union of the rules $\twoheadrightarrow_\eta$ and $\twoheadrightarrow_\times$. We want to show that

$$\twoheadrightarrow^* = \twoheadrightarrow^*_{\beta\pi} \twoheadrightarrow^*_{\eta\times}$$

It suffices to check that any rule $\twoheadrightarrow_\eta$ or $\twoheadrightarrow_\times$ used before a rule $\twoheadrightarrow_\beta$, $\twoheadrightarrow_{\pi_1}$ ou $\twoheadrightarrow_{\pi_2}$ can be delayed after one or several rules of this kind, or even forgotten. Indeed, $\pi_1(\langle \pi_1(t), \pi_2(t) \rangle) \twoheadrightarrow_\times \pi_1(t)$ can be replaced by $\pi_1(\langle \pi_1(t), \pi_2(t) \rangle) \twoheadrightarrow_{\pi_1} \pi_1(t)$, $(\lambda x.(\lambda y.t)x)u \twoheadrightarrow_\eta (\lambda y.t)u \twoheadrightarrow_\beta t[u/x]$ with $x \notin t$ can be replaced by $(\lambda x.(\lambda y.t)x)u \twoheadrightarrow_\beta (\lambda y.t)u \twoheadrightarrow_\beta t[u/x]$, etc. The only cases that should be a problem are when we have terms of the form $\langle \pi_1(\lambda x.t), \pi_1(\lambda x.t) \rangle u$ or $\pi_1(\lambda x.\langle t, u \rangle x)$, but as seen above such terms are not well-typed.

The rewriting system $\twoheadrightarrow_{\eta\times}$ trivially terminates (the size of terms decreases). Concerning $\twoheadrightarrow_{\beta\pi}$, its termination is a well-known result (cf. (Girard, 1972)). □

Local confluence and termination allow to conclude :

Corollary 1. The rewriting system $\twoheadrightarrow$ applied to Curry-style system F is confluent.

This result is essential for our work. Indeed, as we previously said, the interpretation defined futher will be a model for the *reduction*, in the sense that any reduction in the syntax will be interpreted by an inclusion. But an equality in the syntax will not necessarily correspond to an equality in the model. When talking about isomorphisms, we focus on an equality between two terms $v = \lambda x.t(ux)$ and $w = \lambda x.x$. Fortunately, w is a normal form for the system $\twoheadrightarrow$. Then, confluence allows us to say that v can be reduced into w, and its interpretation is included in the interpretation of w.

4.2. *Untyped moves*

The grammar of **untyped moves** is the following:

$$x ::= \uparrow x \mid \downarrow x \mid rx \mid lx \mid j \qquad (j \in \mathbb{N})$$

The set of untyped moves is denoted $\mathbb{X}$.

The justified sequences, plays and strategies induced by this grammar will be called *untyped* justified sequences, plays and strategies.

4.3. *Basic strategies*

We define the following strategies:

— **identity**:

$$id = \{s \in \mathbb{E} \mid s \text{ of arrow shape and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow}\}$$

— **projections**:

$$\pi_r = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow r, \downarrow l\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow r}\}$$

$$\pi_l = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow r, \downarrow l\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow l}\}$$

— **evaluation**:

$$eval = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow l\uparrow, \downarrow l\downarrow, \downarrow r\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow l\uparrow} \wedge t\upharpoonright_{\downarrow r} = t\upharpoonright_{\downarrow l\downarrow}\}$$

We also define three basic operations on strategies:

— **pairing without context**: if σ and τ are two strategies,

$$\langle \sigma, \tau \rangle_a = \{s \in \mathbb{E} \mid s \text{ of shape } \{r, l\} \text{ and } s\upharpoonright_l \in \sigma \text{ and } s\upharpoonright_r \in \tau\}$$

— **pairing with context**: if σ and τ are two strategies of arrow shape,

$$\langle \sigma, \tau \rangle_b = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow r, \uparrow l, \downarrow\} \text{ and } s\upharpoonright_{\uparrow l, \downarrow} \in \sigma \text{ and } s\upharpoonright_{\uparrow r, \downarrow} \in \tau\}$$

— **abstraction**: if σ is a strategy of shape $\{\uparrow, \downarrow r, \downarrow l\}$, $\Lambda(\sigma)$ is the strategy of shape $\{\uparrow\uparrow, \uparrow\downarrow, \downarrow\}$ which is deduced from σ by replacing each move $\uparrow x$ by $\uparrow\uparrow x$, each move $\downarrow rx$ by $\uparrow\downarrow x$ and each move $\downarrow lx$ by $\downarrow x$.

4.4. *Hyperuniformity*

We have enough material to define our untyped model. However, our use of untyped strategies in the Curry-style model forces us to impose new requirements: for example, consider the formula $X_1 \to X_1$. It would be reasonable to think that the innocent strategy σ whose set of views is $\{\varepsilon, \uparrow 1 \cdot \downarrow 1\}$ has this type. However, because we deal with a Curry-style model, any strategy of type $X_1 \to X_1$ should also have the type $\forall X_1.X_1 \to X_1$, and thus $A \to A$ for any A, and should be able to do a copycat between the left and the right side of the arrow.

This is the meaning of the notion of **hyperuniformity** defined below.

Definition 16 (copycat extension of an untyped play). Let $s = x_1 \dots x_n$ be an untyped play, x_i an **O**-move of s and $v = y_1 \dots y_p \in \mathcal{BV}$. Suppose $s = s_1 x_i x_{i+1} s_2$. The **copycat extension** of s at position i with parameter v is the untyped play $s' = cc^s(i, v)$, defined by :

— $s' = s_1 x_i [y_1] x_{i+1} [y_1] s_2$ if $p = 1$

— $s' = s_1 x_i[y_1]x_{i+1}[y_1]x_{i+1}[y_2]x_i[y_2]\ldots x_{i+1}[y_p]x_i[y_p]$ if p even
— $s' = s_1 x_i[y_1]x_{i+1}[y_1]x_{i+1}[y_2]x_i[y_2]\ldots x_i[y_p]x_{i+1}[y_p]$ if $p > 1$ and p odd

Definition 17 (hyperuniform strategy). An untyped strategy σ is called **hyperuniform** if it is innocent and if, for any play $s \in \sigma$, any copycat extension of s is in σ.

Lemma 2. The identity strategy, the projections and the evaluation strategy are hyperuniform. If σ and τ are hyperuniform then $\langle\sigma,\tau\rangle$ and $\Lambda(\sigma)$ are hyperuniform.

The preceding lemma is straightforward. The interesting case is composition:

Lemma 3. If σ and τ are hyperuniform then $\sigma;\tau$ is hyperuniform.

Proof. Let us consider a play $s = x_1 \ldots x_p \in \sigma;\tau$, an **O**-move x_i of s and a bi-view $v = y_1 \ldots y_q$. We have to prove that $s' = cc^s(i,v)$ belongs to $\sigma;\tau$.

There exists a justified sequence u such that $u \upharpoonright_{\uparrow,\downarrow\downarrow} = s$, $u \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$ and $u \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$. If $u = t_1 x_i b_1 \ldots b_q x_{i+1} t_2$, we build a new justified sequence U depending on the value of p :

— if $p = 1$, $U = t_1 x_i[y_1]b_1[y_1]\ldots b_q[y_1]x_{i+1}[y_1]t_2$
— if p even,

$$\begin{aligned} U \;&=\; t_1 x_i[y_1]b_1[y_1]\ldots b_q[y_1]x_{i+1}[y_1]x_{i+1}[y_2]b_q[y_2]\ldots b_1[y_2]x_i[y_2] \\ &\quad\;\ldots\ldots x_{i+1}[y_p]b_q[y_p]\ldots b_1[y_p]x_i[y_p] \end{aligned}$$

— if p odd and $p > 1$,

$$\begin{aligned} U \;&=\; t_1 x_i[y_1]b_1[y_1]\ldots b_q[y_1]x_{i+1}[y_1]x_{i+1}[y_2]b_q[y_2]\ldots b_1[y_2]x_i[y_2] \\ &\quad\;\ldots\ldots x_i[y_p]b_1[y_p]\ldots b_q[y_p]x_{i+1}[y_p] \end{aligned}$$

We have $U\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$ and $U\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ by hyperuniformity of σ and τ. So, $U\upharpoonright_{\uparrow,\downarrow\downarrow} = s' \in \sigma;\tau$. □

4.5. *Interpretation of the untyped λ-calculus with binary products*

We now present the interpretation of the untyped calculus. Instead of directly interpreting terms, we interpret sequents of the form $\Gamma \vdash t$, where t is a term and Γ is simply a list of variables that includes the free variables occurring in t.

The interpretation is as follows:

$$\begin{aligned} [\![\Gamma, x \vdash x]\!] &= \pi_r \\ [\![\Gamma, y \vdash x]\!] &= \pi_l;[\![\Gamma \vdash x]\!] \\ [\![\Gamma \vdash \lambda x.t]\!] &= \Lambda([\![\Gamma, x \vdash t]\!]) \\ [\![\Gamma \vdash (tu)]\!] &= \langle[\![\Gamma \vdash t]\!],[\![\Gamma \vdash u]\!]\rangle; eval \\ [\![\Gamma \vdash \langle t,u\rangle]\!] &= \langle[\![\Gamma \vdash t]\!],[\![\Gamma \vdash u]\!] \\ [\![\Gamma \vdash \pi_1(t)]\!] &= [\![\Gamma \vdash t]\!];\pi_l \\ [\![\Gamma \vdash \pi_2(t)]\!] &= [\![\Gamma \vdash t]\!];\pi_r \end{aligned}$$

From lemmas 2 and 3 we derive:

Lemma 4. Let t be a term whose free variables are contained in the list Γ, then $[\![\Gamma \vdash t]\!]$ is a hyperuniform strategy.

What we have obtained up to now is not a model of system F. However we have a characterisation of the reduction in our context:

Proposition 1. Let t and u be two terms whose free variables are contained in the list Γ. If $t \twoheadrightarrow u$ then $[\![\Gamma \vdash t]\!] \subseteq [\![\Gamma \vdash u]\!]$.

Proof. The relation $\sigma \subseteq \tau$ is reflexive and transitive, and it is a congruence for every construction used in the interpretation: if $\sigma \subseteq \tau$ then
— $\Lambda(\sigma) \subseteq \Lambda(\tau)$
— $\langle \sigma, \rho \rangle_x \subseteq \langle \tau, \rho \rangle$ for any untyped strategy ρ
— $\sigma; \rho \subseteq \tau; \rho$ and $\rho; \sigma \subseteq \rho; \tau$ for any strategy ρ.

If $t \twoheadrightarrow_\beta u$ then $[\![\Gamma \vdash t]\!] = [\![\Gamma \vdash u]\!]$: indeed, in the cartesian closed category of HO-games the partial order $\mathbb{X}$ is a reflexive object, which means that the β-equality is an equality in the interpretation (see (Barendregt, 1984)).

If $x \notin t$ then $[\![\Gamma \vdash \lambda x.tx]\!]$ is the restriction of $[\![\Gamma \vdash t]\!]$ to the plays of shape $\{\uparrow\uparrow, \uparrow\downarrow, \downarrow\}$.

If $t = \langle \pi_1(u), \pi_2(u) \rangle$ then $[\![\Gamma \vdash t]\!]$ is the restriction of $[\![\Gamma \vdash u]\!]$ to the plays of shape $\{\uparrow r, \uparrow l, \downarrow\}$.

Finally, if $t \twoheadrightarrow_{\pi_1} u$ or $t \twoheadrightarrow_{\pi_2} u$ then $[\![\Gamma \vdash t]\!] = [\![\Gamma \vdash u]\!]$. □

Corollary 2. Let us consider the strategy

$$\alpha = \{ s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow r, \downarrow\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\upharpoonright_{\uparrow r} = t\upharpoonright_{\downarrow} \}$$

If $t = \lambda x.t_0$ and $u = \lambda x.u_0$ are two closed terms such that $\lambda x.t(ux) = \lambda x.x$ then $\alpha; [\![x \vdash u_0]\!]; \alpha; [\![x \vdash t_0]\!] \subseteq id$.

Proof. We note $\sigma = \alpha; [\![x \vdash u_0]\!]; \alpha; [\![x \vdash t_0]\!]$. It is easy to see (by induction on t_0) that $\alpha; [\![\vdash \lambda x.t(ux)]\!] = \sigma$. Moreover, and $\lambda x.x$ is a normal form for the reduction system $\twoheadrightarrow$, so $\lambda x.t(ux) \twoheadrightarrow \lambda x.x$ by confluence of $\twoheadrightarrow$. Thus, as $\alpha; [\![\vdash \lambda x.x]\!] = id$, we have $\sigma \subseteq id$. □

5. Arenas

5.1. *Interpretation of a formula*

In this section we introduce the notion of **arena**, the structure that will interpret Curry-style types. This structure is very similar to the one presented in (Abramsky and Jagadeesan, 2003).

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 $\mathbb{A}$.

We define a translation $\mathcal{E}$ from $\mathbb{A}$ to $\mathbb{X}$: $\mathcal{E}(a)$ is obtained by erasing all the tokens $\star$ in a. Inductively:

— $\mathcal{E}(i) = i$
— $\mathcal{E}(\star a) = \mathcal{E}(a)$
— $\mathcal{E}(\alpha a) = \alpha\mathcal{E}(a)$ if $\alpha \in \{\uparrow, \downarrow, r, l\}$.

The **syntactic tree** of a formula A is a tree with nodes labelled by type connectors $(\rightarrow, \times, \forall)$ or integers, edges labelled by the tokens $\uparrow, \downarrow, r, l, \star$, and possibly some arrows linking a leaf to a node. It is defined as follows:

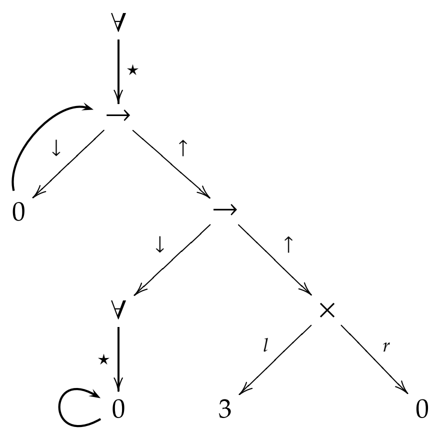
— $T_\perp$ is reduced to a leaf 0
— T_{X_i} is reduced to a leaf i
— $T_{A\rightarrow B}$ consists in a root $\rightarrow$ with the two trees T_A and T_B as sons; the edge between $\rightarrow$ and T_A (resp. T_B) is labelled $\downarrow$ (resp. $\uparrow$)
— $T_{A\times B}$ consists in a root $\times$ with the two trees T_A and T_B as sons; the edge between $\times$ and T_A (resp. T_B) is labelled l (resp. r)
— $T_{\forall X_i.A}$ consists in a root $\forall$ with the tree T as unique son, where T is deduced from T_A by linking each of its leafs labelled by i to its root, and relabelling these leafs by 0; the edge between $\forall$ and T is labelled $\star$.

A maximal branch in a syntactic tree is a path from the root to a leaf; it will be described by the sequence of labels of its edges, with the index of the leaf at the end of the sequence. Such a maximal branch is then an occurrence.

The set $\mathcal{O}_A$ of occurrences of a formula A is the set of maximal branches of T_A. We define a function of **linkage** $\mathcal{L}_A : \mathcal{O}_A \rightarrow \mathbb{A} \cup \{\dagger\}$ as follows: if the leaf reached by the maximal branch a is linked to a node c, then $\mathcal{L}_A(a)$ is the sequence of labels of the edges we cross to reach c starting from the root, with a 0 at the end; otherwise, $\mathcal{L}_A(a) = \dagger$.

The structure $(\mathcal{O}_A, \mathcal{L}_A)$ will be called an **arena**. It will also be denoted A, with no risk of confusion.

Example: The type $A = \forall X_1.(X_1 \rightarrow ((\forall X_2.X_2) \rightarrow (X_3 \times \perp)))$ has the following syntactic tree:



Its set of occurrences is then:

$$\mathcal{O}_A = \{\star\downarrow 0\ ,\ \star\uparrow\downarrow\star 0\ ,\ \star\uparrow\uparrow l3\ ,\ \star\uparrow\uparrow r0\}$$

And its function of linkage is given by:

$$\begin{cases} \mathcal{L}_A(\star\downarrow 0) & = & \star 0 \\ \mathcal{L}_A(\star\uparrow\downarrow\star 0) & = & \star\uparrow\downarrow\star 0 \\ \mathcal{L}_A(\star\uparrow\uparrow l3) & = & \dagger \\ \mathcal{L}_A(\star\uparrow\uparrow r0) & = & \dagger \end{cases}$$

Definition 18 (arena). An **arena** 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 inhabited: $\exists a \in \mathcal{O}_A,\ \vdash a$

— $\mathcal{O}_A$ is **non-ambiguous**: $\forall a, a' \in \mathcal{O}_A$, if $\mathcal{E}(a) \sqsubseteq^p \mathcal{E}(a')$ then $a = a'$

— for every $a \in \mathcal{O}_A$, either $\mathcal{L}_A(a) = \dagger$ or $\mathcal{L}_A(a) = a'[\star 0] \sqsubseteq^p a$ for some $a' \in \mathbb{A}$

— for every $a \in \mathcal{O}_A$, if $\sharp(a) \neq 0$ then $\mathcal{L}_A(a) = \dagger$

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

Note that $\mathcal{O}_A$ shall not be empty: this corresponds to the fact that the grammar of types does not contain $\top$.

Definition 19 (auxiliary polarity). Given an arena A, we define its **auxiliary polarity** as a partial function $paux_A : \mathcal{O}_A \rightharpoonup \{\mathbf{O}, \mathbf{P}\}$ by: $paux_A(c) = \lambda(\mathcal{L}_A(c))$ if $\mathcal{L}_A(c) \neq \dagger$, otherwise it is undefined.

We also define $FTV(A) = \{X_i \in \mathbb{N} \mid\ \exists a \in \mathcal{O}_A, \sharp(a) = i\}$.

5.2. *Alternative, inductive interpretation of a formula*

We define the following constructions on arenas:

(atoms) $\bot = (\{0\}, 0 \mapsto \dagger)$ $\qquad X_i = (\{i\}, i \mapsto \dagger)$ for $i > 0$.

(product) if $A, B \in \mathcal{G}$, 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 $A, B \in \mathcal{G}$, 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 $A \in \mathcal{G}$ 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 a formula, which coincides with the one defined from the syntactic tree.

Finally, we define an operation of substitution on arenas:

Definition 20 (substitution). Let $A, B \in \mathcal{G}$. The **substitution** of B for X_i in A is the arena $A[B/X_i]$ defined by:

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

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

One can check that this coincides with the operation of substitution on formulas.

6. The typed model

6.1. *Moves and strategies on an arena*

We are now going to describe how we can play in an arena. We will take advantage of the way we have defined arenas: whereas in many second order game models like (Hughes, 2000) or (de Lataillade, 2007) moves have a complex structure, here they will be easy to derive from $\mathcal{O}_A$ and $\mathcal{L}_A$.

As in (Abramsky and Jagadeesan, 2003), the intuition is that a move in A can either be built directly from an occurrence of $\mathcal{O}_A$, or it can be decomposed as $m_1[m_2]$, where m_1 is built from an occurrence of $\mathcal{O}_A$ and m_2 is a move in another arena B which substitutes a quantifier.

Note that the moves and strategies defined this way do not constitute the morphisms of our model, but they will be used as interpretations of Church-style terms.

We introduce the grammar of **typed moves**:

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

These moves form the set $\mathbb{M}$.

The operation of **anonymity** $\mathcal{A} : \mathbb{M} \to \mathbb{A}$ erases the arena indication in a typed move:

— $\mathcal{A}(i) = i$ for $i \geq 0$

— $\mathcal{A}(\star^A m) = \star\mathcal{A}(m)$

— $\mathcal{A}(\alpha m) = \alpha\mathcal{A}(m)$ for $\alpha \in \{r, l, \uparrow, \downarrow\}$.

For $m \in \mathbb{M}$ and $a \in \mathbb{A}$, we define a partial operation of **formula extraction** $\frac{m}{a}$ by:

— $\frac{\star^B m}{\star 0} = B$

— if $\frac{m}{a}$ is defined, $\frac{\star^B m}{\star a} = \frac{\alpha m}{\alpha a} = \frac{m}{a}$ where $\alpha \in \{\uparrow, \downarrow, r, l\}$

Definition 21 (moves of 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:

— if $\mathcal{A}(m) = a \in O_A$ and $\mathcal{L}_A(a) = \dagger$ then $m \in \mathcal{M}_A$
— if $m = m_1[m_2]$, where $\mathcal{A}(m_1) = a \in O_A$, $\mathcal{L}_A(a) \neq \dagger$ and $m_2 \in \mathcal{M}_B$ with $B = \frac{m_1}{\mathcal{L}_A(a)}$, then $m \in \mathcal{M}_A$.

This definition is well-founded, because in the second case we necessarily have at least one token $\star^B$ in m_1, so the size of m_2 is strictly smaller than the size of $m_1[m_2]$: that is why we say that the definition is inductive.

Example: Let us recall the type $A = \forall X_1.(X_1 \to ((\forall X_2.X_2) \to (X_3 \times \bot)))$ of the preceding example. One possible way to "play a move" in this arena[§] is to instantiate the variable X_1 with a type B (take $B = \bot \times X_3$ for example), then to go on the left side of the first arrow and to play a move of B.

This corresponds to a move like $m = \star^B \downarrow r3$. One can check with the definition that this move indeed belongs to $\mathcal{M}_A$: $m = m_1[m_2]$ with $m_1 = \star^B \downarrow 0$ and $m_2 = r3$. $\mathcal{A}(m_1) = \star\downarrow 0 \in O_A$, $\mathcal{L}_A(\star\downarrow 0) = \star 0$ and $\frac{\star^B\downarrow 0}{\star 0} = B$. Moreover, $\mathcal{A}(m_2) = r3 \in O_B$ and $\mathcal{L}_B(m_2) = \dagger$ so $m_2 \in \mathcal{M}_B$ (first case of the definition). So, $m \in \mathcal{M}_B$ (second case of the definition).

Intuitively, we have the following:

— m_1 is the part of the move played in A, and $c = \mathcal{A}(m_1)$ is the corresponding occurrence
— $\mathcal{L}_a(c)$ indicates where the interesting quantifier has been instantiated
— $\frac{m_1}{\mathcal{L}_A(c)} = B$ indicates by which arena it has been instantiated
— m_2 is the part of the move played in B.

Definition 22 (justified sequence, play on an arena). Let A be an arena and s be a play (resp. a justified sequence) on the grammar $\mathbb{M}$. If every move of s belongs to $\mathcal{M}_A$, then we say that s is a play (resp. a justified sequence) on the arena A. The set of plays on the arena A is denoted $\mathcal{P}_A$.

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

Example: Let us consider the play $s = \star^B\uparrow\uparrow l3 \cdot \star^B\downarrow r3$ with $B = \bot \times X_3$. This is of course a play *in* $A = \forall X_1.(X_1 \to (\forall X_2.X_2) \to (X_3 \times \bot))$.

What is interesting to notice is that, if for example $C = X_3 \times \bot$, then the sequence $s' = \star^C\uparrow\uparrow l3 \cdot \star^B\downarrow r3$ is not a play because it is not a justified sequence: indeed, one must have $B = C$ if we want $m_2 = \star^B\downarrow r3$ to be justified by $m_1 = \star^C\uparrow\uparrow l3$.

More generally, for any move m in a play s which contains the token $\star^B$, there is a sequence of moves $m_1, \ldots, m_n$ that also contains the token $\star^B$ at the same place, with $m_n = m$ and m_i justifies m_{i+1} for $1 \leq i < n$. If this sequence is chosen to be of maximal length, then m_1 is the minimal hereditarily justifier of m which contains the token $\star^B$: it is the first time that it appears (at the right place). We will say that B is played by $\lambda(m_1)$ at the **level** of m_1. Note that $\lambda(m_1) = paux_A(m)$.

One can formalise this definition:

[§] This notion is related to the idea of **evolving game** introduced in (Murawski and Ong, 2001) and reused in (de Lataillade, 2007).

Definition 23 (level). If a move m in a play $s \in \mathcal{P}_A$ contains the token $\star^B$, then it can be written $m = m_0 \star^B [m_1]$. We say that B is played (by $\lambda(m_0)$) at the **level** of m if m_1 does not contain the token $\downarrow$.

Typed strategies are defined as expected:

Definition 24 (strategy on an arena). Let σ be a strategy on the grammar $\mathbb{M}$, we say that σ is a strategy on A and we note $\sigma : A$ if any play of σ belongs to $\mathcal{P}_A$. We say that σ is a typed strategy in this case.

Strategies on arenas have to be understood as interpretations[¶] of Church-style system F terms; they will be used in the Curry-style model because we have to express in the model the fact that a well-typed Curry-style term is the erasure of a well-typed Church-style term.

6.2. *Uniformity*

In (de Lataillade, 2007), we saw that strategies defined as generally as possible were not able to capture exactly the type isomorphisms of the syntax, because they were generating too many isomorphisms in the model. That is why we introduced a notion of *uniformity*, which restrained the behaviour of strategies (in order to avoid confusion, we will call *weak uniformity* the notion of uniformity defined in (de Lataillade, 2007); by the way, weak uniformity plays no role in the present model).

The situation is similar here: we are not able to derive the characterisation of Curry-style type isomorphisms if the well-typed Church-style terms are interpreted by the (typed) strategies defined above. So we introduce a property of **uniformity** on these strategies.

The intuition of this notion is the following: consider an η-long, β-normal term t of the Church-style system F, and suppose $\vdash t : \forall X.A$. The term t has the form $t = \Lambda X.t'$ with $\vdash t' : A$: so it behaves like if it was instantiating the quantifier ($\forall X$) with a variable (X). More generally, the terms of the Church-style system F should be interpreted by strategies where, each time **O** has to play an arena, he gives a variable arena X_i.

But these strategies (that we will call **symbolic**) do not compose: in the Church-style syntax, this corresponds to the fact that the term $\vdash t : \forall X.A$ can be instantiated at any type B through the operation $t \mapsto t\{B\}$, and so the term t can be extended to any type $A[B/X]$. In the interpretation, this means that the symbolic strategy interpreting t must be extensible to a more complete strategy, where **O** can play any arena he wants. This extension consists in playing copycat plays between the different occurrences of the variables X (like in the syntax, the η-long β-normal form of $t\{B\}$ is generated from t through η-expansions), that is why it is called the **copycat extension**.

To sum up, a uniform strategy will be a symbolic strategy extended by copycat extension. This idea has to be related with the strategies of Dominic Hughes (Hughes,

¶ We chose not to explicit this interpretation because we do not need it; one could also prove that we have a model of Church-style system F, but it is not an important question here.

2000) and, above all, with Murawski's notion of *good strategies* (Murawski and Ong, 2001). The notion of weak uniformity discussed above is an analogous, but less restrictive, condition: uniformity implies weak uniformity. Finally, uniformity has of course a strong connection with hyperuniformity: the two notions express analogous ideas, but hyperuniformity applies on untyped strategies, whereas uniformity is formulated in a typed context, and then requires more cautiousness.

In the following definition, $\mathcal{BV}(A)$ stands for the set of bi-views in an arena A, and $m[B/j]$ (resp. $s[B/j]$) is obtained from the move m (resp. the play s) by replacing each token of the form $\star^A$ by $\star^{A[B/X_j]}$. Note that $s[B/j]$ is a play, but does not necessarily belong to any $\mathcal{M}_A$ for some A: actually, this play will only be used as an intermediate construction.

Definition 25 (copycat extension of a typed play). Let $s = m_1 \dots m_n$ be a typed play on the arena A, let $B \in \mathcal{G}$ and $j > 0$.

We first define the **flat extension** of s: given a sequence of initial moves $r = (r_i)_{i\in\mathbb{N}}$ in $\mathcal{M}_B$, $Fl^s_{j,B}(r)$ is the play $t[B/j]$ where t is obtained from s by replacing each sequence $m_i m_{i+1}$ such that $\sharp(m_i) = j$ and $\lambda(m_i) = \mathbf{O}$ by $m_i[r_i]m_{i+1}[r_i]$ and, if $\sharp(m_n) = j$ and $\lambda(m_n) = \mathbf{O}$, by replacing m_n by $m'_n[r_n]$.

Let m_i be an **O**-move of s such that $\sharp(m_i) = j$, suppose $Fl^s_{j,B}(r) = s_1 m'_i[r_i]m'_{i+1}[r_i]s_2$ with $m'_i = m_i[B/j]$ and $m'_{i+1} = m_{i+1}[B/j]$, and let $v = n_1 \dots n_p \in \mathcal{BV}(B)$. The B**-copycat extension** of s at position i along the index j (with parameters v, r) is the play $s' = CC^s_{j,B}(i, v, r)$ defined by:

— $s = s_1$ if $p = 0$ (i.e. $v = \epsilon$)
— $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]s_2$ if $p = 1$
— $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]m'_i[n_2] \dots m'_{i+1}[n_p]m'_i[n_p]$ if p even
— $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]m'_i[n_2] \dots m'_i[n_p]m'_{i+1}[n_p]$ if $p > 1$ and p odd

Finally, if $i = n$, $Fl^s_{j,B}(r) = s_1 m'_n[r_n]$ and $v = n_1 \dots n_p$ then

$$CC^s_{j,B}(n, v, r) = \begin{cases} s_1 & \text{if } p = 0 \\ s_1 m'_n[n_1] & textotherwise \end{cases}$$

Definition 26 (copycat variable, symbolic strategy). Let $s = s_1 m$ be a play on the arena A with $\lambda(m) = \mathbf{O}$, and X_i a variable arena played at the level of m. X_i is called a **copycat variable** of s if $X_i \notin FTV(\ulcorner s_1 \urcorner) \cap FTV(A)$.

A view s on the arena A is called **symbolic** if, for any move m of s such that $\lambda(m) = \mathbf{O}$, the arenas played at the level of m are two by two distinct copycat variables $X_{i_1}, \dots, X_{i_n}$.

A play (resp. a strategy) is called **symbolic** if all its views are symbolic.

Definition 27 (uniform strategy). Let σ be a strategy on the arena A. σ is said to be *uniform* if there exists a symbolic innocent strategy $\bar{\sigma}$ on A such that: σ is the smallest innocent strategy containing $\bar{\sigma}$ which is stable by copycat extension along a copycat variable.

Later we will say that s' is an **extension** of s if it is obtained from s by a sequence of copycat extensions along copycat variables. A uniform strategy is then stable by extension.

Arenas and uniform straegies in fact give us a model of the Church-style system F (see (de Lataillade, 2007)) Intuitively, the views of the symbolic strategy are the direct representation of a term of the Church-style term, whereas the associated uniform strategy is the extension of this strategy, designed to be able to compose with other strategies:

Proposition 2. If $\sigma : A \to B$ and $\tau : B \to C$ are two uniform strategies then $\sigma ; \tau : A \to C$ is uniform.

The proof of this proposition can be found in appendix A.

7. Typing the untyped strategies

We are now ready to define our interpretation of the Curry-style system F: the key ingredient will be to relate untyped strategies with typed strategies through a notion of **erasure**. First we relate untyped moves with typed moves through the function $erase : \mathbb{M} \to \mathbb{X}$ defined by:

$$erase = \mathcal{E} \circ \mathcal{A}$$

The function *erase* can be extended to plays and strategies in the obvious way:

$$erase(m_1 \dots m_n) = erase(m_1) \dots erase(m_n)$$
$$erase(\sigma) = \{erase(s) \mid s \in \sigma\}$$

At present we have all the ingredients to define our category of games:

— **objects** are arenas
— a **morphism** between A and B is an untyped strategy σ such that:
  - σ is hyperuniform
  - there exists a uniform typed strategy $\tilde{\sigma} : A \to B$ such that $erase(\tilde{\sigma}) \subseteq \sigma$.

  In this case we note $\sigma :: A \to B$. The strategy $\tilde{\sigma}$ will be called a **realisation** of σ.

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

Proof. If we note $\tilde{\sigma}$ and $\tilde{\tau}$ two realizations of σ and τ respectively, we obtain a realization of $\sigma ; \tau$ on $A \to C$ by taking the composite $\tilde{\sigma} ; \tilde{\tau}$ in the grammar $\mathbb{M}$. Indeed, if $s \in \tilde{\sigma} ; \tilde{\tau}$ then there exists a justified sequence u such that $u \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \tilde{\sigma}$, $u \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tilde{\tau}$ and $u \upharpoonright_{\uparrow,\downarrow\downarrow} = s$. Then $U = erase(u)$ is such that $U \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$, $U \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ and $U \upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play, so $erase(s) = U \upharpoonright_{\uparrow,\downarrow\downarrow} \in \sigma ; \tau$.
Moreover, $\tilde{\sigma}$ and $\tilde{\tau}$ are uniform, so $\tilde{\sigma} ; \tilde{\tau}$ is uniform by prop. 2; σ and τ are hyperuniform so $\sigma ; \tau$ is hyperuniform by lemma 3. ☐

Lemma 6. If $\sigma : \Gamma \to A$ and $X_j \notin \Gamma$ then $\sigma : \Gamma \to \forall X_j . A$

Proof. Let us consider $\tilde{\sigma} :: \Gamma \to A$, a realization of σ on $\Gamma \to A$: if $\tilde{\sigma}$ is the copycat extension of a symbolic strategy $\bar{\sigma}$, then we define the strategy $\bar{\sigma}'$ as the strategy $\bar{\sigma}$ where each move written $\uparrow m$ in a play has been replaced by $\uparrow \star^{X_j} m$. This strategy is symbolic on $\Gamma \to \forall X_j.A$, and its copycat extension $\tilde{\sigma}'$ is a realization of σ because of hyperuniformity (indeed, the only difference between $\tilde{\sigma}$ and $\tilde{\sigma}'$ is a copycat extension along X_j). □

Lemma 7. If $\sigma : \Gamma \to \forall X_j.A$ and B is a arena then $\sigma : \Gamma \to A[B/X_j]$.

Proof. If $\tilde{\sigma}$ is a realization of σ on $\Gamma \to \forall X_j.A$, a realization $\tilde{\sigma}'$ on $\Gamma \to A[B/X_j]$ is obtained by taking only plays where each initial move takes the form $\uparrow \star^B m$, and by replacing each move $\uparrow \star^B m$ by $\uparrow m$.

Let us now prove the uniformity of $\tilde{\sigma}'$: if $\tilde{\sigma}$ is the copycat extension of a symbolic strategy $\bar{\sigma}$, we consider a view s of $\bar{\sigma}$. Let X_j be the first copycat variable appearing in s, and let us define $E(s)$ as the smallest set of plays containing s and stable by B-copycat extensions along j. The strategy $\bar{\sigma}'$ will be the smallest innocent strategy containing all the sets $E(s)$, for s describing all the views of $\bar{\sigma}$. Then one can check that $\tilde{\sigma}'$ is the copycat extension of $\bar{\sigma}'$. □

Lemma 8. The following holds:
— $id : A \to A$
— $\pi_r : \Gamma \times A \to A$
— If $\sigma : \Gamma \to A$ and $\tau : \Gamma \to B$ then $\langle \sigma, \tau \rangle : \Gamma \to (A \times B)$.
— $eval : (A \to B) \times A \to B$
— If $\sigma : \Gamma \times A \to B$ then $\Lambda(\sigma) : \Gamma \to (A \to B)$.

These cases are trivial: for example, a realization of id on $A \to A$ is

$$\rho = \{s \in \mathcal{P}_{A \to A} \mid s \text{ of arrow shape and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow}\}$$

and it is uniform, with symbolic strategy $\bar{\rho}$ defined by:

$$\bar{\rho} = \{s \in \mathcal{P}_{A \to A} \mid s \text{ of arrow shape}, s \text{ symbolic and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow}\}$$

If Γ is a typing context of the form $\Gamma = x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$, we define the sequence of variables $\overline{\Gamma} = x_1, x_2, \ldots, x_n$ and the type $|\Gamma| = \bot \times A_1 \times A_2 \times \cdots \times A_n$ ($|\Gamma| = \bot$ if Γ is empty), and we have:

Proposition 3. If $\Gamma \vdash t : A$ then $[\![\overline{\Gamma} \vdash t]\!] : |\Gamma| \to A$.

As we did not have a model of the untyped calculus, we do not obtain a model of the Curry-style system F either. But, fortunately, the relations we get from the untyped interpretation, together with our notion of realisation, will be sufficiently efficient tools to be able to characterise type isomorphisms in this language.

8. Hyperforests

In this section we introduce the notion of **hyperforest**, an arborescent structure built from arenas. In (de Lataillade, 2007), following (Hughes, 2000), we interpreted second-order types directly as hyperforests (that we called polymorphic arenas). But the substitution was difficult to define in this context, and moves had a complicated formulation; that is why in this paper we introduce hyperforests only as an indirect interpretation of types.

Hyperforests will be the fundamental structure for our work on isomorphisms.

8.1. *Forests and hyperforests*

In what follows, the set of subsets of a set E will be denoted $\mathbb{P}(E)$. We also make use of the notion of **multiset**: we recall that a multiset of elements of E is a function $f : E \to \mathbb{N}$, and we note $e \in f$ iff $f(e) > 1$. The set of multisets of E is denoted $\mathbb{P}_{mult}(E)$.

Definition 28 (forest). A **forest** is an ordered set $(E, \leq)$ such that, for every y in E, $\{x \mid x \leq y\}$ is finite and totally ordered by $\leq$. The forest is **finite** if E is finite.

Definition 29 (hyperforest). An **hyperforest** $H = (\mathcal{F}, \mathcal{R}, \mathcal{D})$ is a finite forest $\mathcal{F}$ together with a multiset of **hyperedges** $\mathcal{R} \in \mathbb{P}_{mult}(\mathcal{F} \times \mathbb{P}(\mathcal{F}))$ and a partial function of **decoration** $\mathcal{D} : \mathcal{F} \rightharpoonup \mathcal{X}$, where:

— for every $(t, S) \in \mathcal{R}$, if $s \in S$ then $t \leq s$ and $\mathcal{D}(s)$ is undefined
— for every $b, b' \in \mathcal{R}$ with $b = (t, S)$ and $b' = (t', S')$, $S \cap S' \neq \emptyset \Rightarrow b = b'$.

We note $\mathcal{T}^H = \{t \in \mathcal{F} \mid \exists S \subseteq \mathcal{F},\ (t, S) \in \mathcal{R}\}$ and $\mathcal{S}^H = \{s \in \mathcal{F} \mid \exists (t, S) \in \mathcal{R},\ s \in S\}$.

Definition 30 (reference, friends). Let $H = (\mathcal{F}, \mathcal{R}, \mathcal{D})$ be an hyperforest. For any $s \in \mathcal{F}$, if $s \in \mathcal{S}^H$ then there exists a unique $(t, S) \in \mathcal{R}$ with $s \in S$: the **reference** of s is defined as $ref^H(s) = t$ and the set of **friends** of s is $fr^H(s) = S \backslash \{s\}$. If $s \notin \mathcal{S}^H$, ref^H and fr^H are not defined in s.

We are now going to exhibit the hyperforest structure associated with an arena A.

8.2. *From partially ordered sets to forests*

Let $(E, \leq)$ be a partially ordered set. The relation $\vdash\, \subseteq E \cup (E \times E)$ is given by:

$$\begin{cases} \vdash e & \text{iff } e' \leq e \Rightarrow (e' = e) \\ e \vdash e' & \text{iff } e \leq e' \wedge \forall f,\ e \leq f \leq e' \Rightarrow (e = f \vee e' = f) \end{cases}$$

One defines the set F of **paths** in $(E, \leq)$, i.e. the set of sequences $e_1 e_2 \dots e_n$ of elements of E such that $\vdash e_1$ and $e_i \vdash e_{i+1}$ for $1 \leq i \leq n-1$. If we consider the prefix ordering $\leq'$ on F, then $(F, \leq')$ is a forest.

We also define the operation $or : F \to E$ by $or(f) = e_n$ if $f = e_1 \dots e_n$ ($or(f)$ is called the **origin** of f).

8.3. *From arenas to hyperforests*

If A is an arena, O_A is a finite partially ordered set, to which one can associate a forest $\mathcal{F}_A$ through the preceding construction. Extending $\vdash$ to $\mathcal{F}_A$ generates the enabling relation of the forest: this justifies *a posteriori* the definition of an enabling relation for arbitrary moves given in section 3.

Furthermore, one deduces from $\mathcal{L}_A$ the multiset $\mathcal{R}_A \in \mathbb{P}_{mult}(\mathcal{F}_A \times \mathbb{P}(\mathcal{F}_A))$ as follows : we set

$$\mathcal{L} = \{a[\star 0] \in \mathbb{A} \mid \exists a' \in \mathcal{O}_A, a[\star 0] \sqsubseteq^p a'\}$$

The value of $\mathcal{R}_A$ on (r, S) is equal to the number of occurrences $y \in \mathcal{L}$ such that:

— $y \sqsubseteq^p or(t)$
— for every $t' \leq t$, if $y \sqsubseteq^p or(t')$ then $t' = t$
— $S = \{s \in \mathcal{F}_A \mid t \leq s \wedge \mathcal{L}_A(or(s)) = y\}$

One also defines the partial function $\mathcal{D}_A : \mathcal{F}_A \rightharpoonup \mathcal{X}$ by: $\mathcal{D}_A(x) = X_i$ iff $\sharp(or(x)) = i$ $(i > 0)$.

Then we have:

Lemma 9. If A is an arena, then $H_A = (\mathcal{F}_A, \mathcal{R}_A, \mathcal{D}_A)$ is an hyperforest.

Example: Consider the type $A = \forall X_1.((X_1 \times X_2) \rightarrow (X_1 \times \bot))$. We have:

$$\mathcal{O}_A = \{\star\!\downarrow\! l0, \star\!\downarrow\! r2, \star\!\uparrow\! l0, \star\!\uparrow\! r0\}$$

and:

$$\begin{cases} \mathcal{L}_A(\star\!\downarrow\! l0) & = & \star 0 \\ \mathcal{L}_A(\star\!\downarrow\! r2) & = & \dagger \\ \mathcal{L}_A(\star\!\uparrow\! l0) & = & \star 0 \\ \mathcal{L}_A(\star\!\uparrow\! r0) & = & \dagger \end{cases}$$

The paths are: $a = \star\!\uparrow\! l0$, $b = \star\!\uparrow\! l0 \cdot \star\!\downarrow\! l0$, $c = \star\!\uparrow\! l0 \cdot \star\!\downarrow\! r2$, $d = \star\!\uparrow\! r0$, $e = \star\!\uparrow\! r0 \cdot \star\!\downarrow\! l0$ and $f = \star\!\uparrow\! r0 \cdot \star\!\downarrow\! r2$. Besides, $\mathcal{L} = \{\star 0\}$.

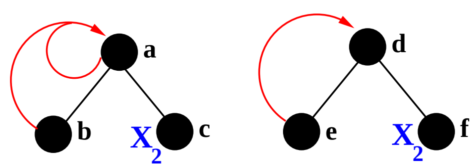
Hence the hyperforest H_A is given by:

$$\mathcal{F}_A = \{a, b, c, d, e, f\}$$

$$\mathcal{R}_A(r) = 1 \text{ if } r \in \{(a, \{a, b\}), (d, \{e\})\} \qquad \mathcal{R}_A(r) = 0 \text{ otherwise}$$

$$\mathcal{D}_A(c) = \mathcal{D}_A(f) = X_2$$

This can be resume in the following representation of H_A:



One can extend the definition of polarity to the nodes of the hyperforest: if A is an arena with associated hyperforest $H_A = (\mathcal{F}_A, \mathcal{R}_A, \mathcal{D}_A)$, then for $a \in \mathcal{F}_A$ we define $\lambda(a) = \lambda(or(a))$. This coincides with an alternative definition of polarity, which is common in HO-style games: $\lambda(a) = \mathbf{O}$ (resp. $\lambda(a) = \mathbf{P}$) if the set $\{a' \in \mathcal{F}_A \mid a' \leq a\}$ has an odd cardinality (resp. an even cardinality). Note also that $paux_A(or(a)) = \lambda(ref_A(a))$.

Finally, if A is an arena, we note:

$$fr_A = fr^{H_A} \qquad ref_A = ref^{H_A} \qquad \mathcal{S}_A = \mathcal{S}^{H_A} \qquad \mathcal{T}_A = \mathcal{T}^{H_A}$$

Note that the nodes of the forest $\mathcal{F}_A$ contain more *information* than the occurrences of O_A. Indeed, given a node $c \in \mathcal{F}_A$, one is able to give the ordered list of its ancestors, whereas for an occurrence we may have many ancestors that are not compatible one with the order for the ordering. This idea will be used in the proof of theorem 1 to reason about plays with nodes instead of occurrences.

9. Type isomorphisms

9.1. *Isomorphisms in the model*

If $R \in \mathbb{P}_{mult}E$ and $f : E \to F$, we note $f(R)$ the multiset defined by:

$$f(R)(g) = \sum_{e \in R \wedge f(e) = g} R(e)$$

Definition 31 (Church-isomorphism). Let $H_1 = (\mathcal{F}_1, \mathcal{R}_1, \mathcal{D}_1)$ and $H_2 = (\mathcal{F}_2, \mathcal{R}_2, \mathcal{D}_2)$ be two hyperforests. We say that H_1 and H_2 are **Church-isomorphic** ($H_1 \simeq_{Ch} H_2$) if there exists a bijection $f : \mathcal{F}_1 \to \mathcal{F}_2$ which preserves the hyperforest structure, i.e. such that:

— $a \leq a'$ iff $f(a) \leq f(a')$
— $\mathcal{R}_2 = f(\mathcal{R}_1)$
— $\mathcal{D}_2 \circ f = \mathcal{D}_1$

Definition 32 (Curry-isomorphism). Let $H_1 = (\mathcal{F}_1, \mathcal{R}_1, \mathcal{D}_1)$ and $H_2 = (\mathcal{F}_2, \mathcal{R}_2, \mathcal{D}_2)$ be two hyperforests. We say that H_1 and H_2 are **Curry-isomorphic** ($H_1 \simeq_{Cu} H_2$) if there exists a bijection $f : \mathcal{F}_1 \to \mathcal{F}_2$ such that:

— $a \leq a'$ iff $f(a) \leq f(a')$
— $\mathcal{S}^{H_2} = f(\mathcal{S}^{H_1})$
— for every $(t, S) \in \mathcal{R}_1$ (resp. $(t, S) \in \mathcal{R}_2$), if there exists $s \in S$ such that $\lambda(s) \neq \lambda(t)$, then $(f(t), f(S)) \in \mathcal{R}_2$ (resp. $(f^{-1}(t), f^{-1}(S)) \in \mathcal{R}_1$)
— $\mathcal{D}_2 \circ f = \mathcal{D}_1$.

Definition 33 (game isomorphism). A **game isomorphism** between two arenas A and B is a couple of untyped strategies $\sigma : A \to B$ and $\tau : B \to A$ such that $\sigma ; \tau \subseteq id$ and $\tau ; \sigma \subseteq id$.

We say that this isomorphism is **total** if there exists a total realisation of σ and of τ. We note $A \simeq_g B$ if there is a total game isomorphism between A and B.

We are now able to formulate the key theorem of our paper. This theorem provides a geometrical characterisation of isomorphisms in the model, which is the core of the proof of equational characterisation for the syntax.

Theorem 1. Let $A, B \in \mathcal{G}$. If there exists a total game isomorphism (σ, τ) between A and B ($A \simeq_g B$) then their hyperforests are Curry-isomorphic ($H_A \simeq_{Cu} H_B$).

The proof of this theorem can be found in appendix B.

9.2. *Characterisation of Curry-style type isomorphisms*

Proving theorem 1 was the main step towards the characterisation of Curry-style isomorphisms: we are now able to establish our final result.

Let us recall the equational system $\simeq_\varepsilon$ which we claim to characterise Curry-style type isomorphisms:

$$
\begin{aligned}
A \times B &\simeq_\varepsilon B \times A \\
A \times (B \times C) &\simeq_\varepsilon (A \times B) \times C \\
A \to (B \to C) &\simeq_\varepsilon (A \times B) \to C \\
A \to (B \times C) &\simeq_\varepsilon (A \to B) \times (A \to C) \\
\forall X.\forall Y.A &\simeq_\varepsilon \forall Y.\forall X.A \\
A \to \forall X.B &\simeq_\varepsilon \forall X.(A \to B) && \text{if } X \notin FTV(A) \\
\forall X.(A \times B) &\simeq_\varepsilon \forall X.A \times \forall X.B \\
\forall X.A &\simeq_\varepsilon A[\forall Y.Y/X] && \text{if } X \notin Neg_A
\end{aligned}
$$

Lemma 10. Let A and B be two types such that the hyperforests H_A and H_B are Curry-isomorphic. Then A and B are equal up to the equational system $\simeq_\varepsilon$.

Proof. Let A' and B' be the normal forms of A and B for the following rewriting system:

$$\forall X.C \Rightarrow C[\forall Y.Y/X] \quad \text{if } X \notin Neg_C \text{ and } C \neq X$$

If $D_1 = \forall X.C$ and $D_2 = C[\forall Y.Y/X]$ with $X \notin Neg_C$, then $H_{D_1} \simeq_{Cu} H_{D_2}$: indeed, the bijection $f : \mathcal{F}_{D_1} \to \mathcal{F}_{D_2}$ which preserves the ordering and such that $\mathcal{S}_{D_2} = f(\mathcal{S}_{D_1})$ and $\mathcal{D}_{D_2} \circ f = \mathcal{D}_1$ is easy to define (in fact $\mathcal{O}_{D_1}$ and $\mathcal{O}_{D_2}$ are already in bijection). The fact that $X \notin Neg_C$ precisely implies that, for any $(t, S) \in \mathcal{R}_{D_1}$ corresponding to the quantification $\forall X$ (i.e. such that $\mathcal{L}_{\forall X.A}(or(s)) = \star 0$ for every $s \in S$), there is no $s \in S$ such that $\lambda(s) \neq \lambda(t)$. Reciprocally, if for any $(t, S) \in \mathcal{R}_{D_2}$ corresponding to a quantification $\forall Y.Y$, $S = \{t\}$ so there is no $s \in S$ such that $\lambda(s) \neq \lambda(t)$. Any other hyperedge is preserved by f.

Moreover, being Curry-isomorphic is a congruence (i.e. it is preserved by context), so $H_A \simeq_{Cu} H_{A'}$, $H_B \simeq_{Cu} H_{B'}$, and hence $H_{A'} \simeq_{Cu} H_{B'}$. $H_{A'}$ and $H_{B'}$ are such that for every $(t, S) \in \mathcal{R}_{A'}$ (or $(t, S) \in \mathcal{R}_{B'}$), either $S = \{t\}$ or S contains a node s with $\lambda(t) \neq \lambda(s)$. Because of the definitions of $\simeq_{Cu}$ and $\simeq_{Ch}$, this implies $H_{A'} \simeq_{Ch} H_{B'}$.

It has already been proved in (de Lataillade, 2007)[‖] that in this case $A' \simeq'_\varepsilon B'$, where $\simeq'_\varepsilon$ is the same equational system as $\simeq_\varepsilon$, except that it does not make use of the last equation. Hence, we have $A \simeq_\varepsilon B$. □

[‖] In (de Lataillade, 2007) the interpretation of types was directly hyperforests.

Theorem 2. Two types A and B are isomorphic in Curry-style system F if and only if $A \simeq_\varepsilon B$.

Proof. For the implication, suppose that $\lambda x.t$ and $\lambda x.u$ is the couple of terms that realise the type isomorphism. We know from corollary 2 that it implies: $\alpha[\![x \vdash u]\!]; \alpha[\![x \vdash t]\!] \subseteq id$ and $\alpha; [\![x \vdash u]\!]; \alpha[\![x \vdash t]\!] \subseteq id$. And we also have $\alpha; [\![x \vdash t]\!] : A \to B$ and $\alpha; [\![x \vdash u]\!] : B \to A$.

Furthermore, one can prove that there exists a total realisation of $\sigma_1 = [\![x \vdash t]\!]$ and $\sigma_2 = [\![x \vdash u]\!]$: indeed, consider t', the $\beta\pi$-normal form of t, we know that $[\![x \vdash t']\!] = \sigma_1$. We choose the realisation of $[\![x \vdash t]\!]$ obtained by applying the constructions given in the proofs of section 7. The strategy σ_1 is the interpretation of a normal form, where applications are necessarily of the form yT where y is a variable and T is a term. Then the associated realisation is total on $A \to B$: indeed, the projections are total, and the constructions used the interpret $(\to I)$, $(\times I)$, $(\times E1)$, $(\times E_2)$, $(\forall I)$, $(\forall E)$ and the application of a variables y to a term T preserve totality.

So, we have a total game isomorphism between A and B. From theorem 1 and lemma 10 we deduce $A \simeq_\varepsilon B$.

For the reciprocal, we already know from (Di Cosmo, 1995) the existence in the Church-style system F of the isomorphisms corresponding to each equation of $\simeq_\varepsilon$, except the last one ($\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if $X \notin Neg_A$). This implies their existence in the Curry-style system F.

Hence, we need, given a type A such that $X \notin Neg_A$, to find two Curry-style terms $t : \forall X.A \to A[\forall Y.Y/X]$ and $u : A[\forall Y.Y/X] \to \forall X.A$ which compose in both ways to give the identity. We suppose Y does not appear at all in A, even as a bound variable.

We take $t = \lambda x.x$: indeed, the identity can be shown to be of type $\forall X.A \to A[\forall Y.Y/X]$ through the following type derivation:

$$\dfrac{\dfrac{x : \forall X.A \vdash x : \forall X.A}{x : \forall X.A \vdash x : A[\forall Y.Y/X]}}{\vdash \lambda x.x : \forall X.A \to A[\forall Y.Y/X]}$$

To build a term of type $A[\forall Y.Y/X] \to \forall X.A$ in the Curry-style system F, we will make use of the associated Church-style term.

Consider the term P of the Church-style system F which corresponds to the $\eta\times$-long form of the identity on A. This term has the form $P = \lambda x^A.P'$. We then build the term Q obtained from P' by replacing each variable y appearing in the scope of a binder λy^X by $y\{X\}$, and each binder λz^B such that $X \in Pos_B$ by $\lambda z^{B[\forall Y.Y/X]}$. We finally set $N = \lambda x^{A[\forall Y.Y/X]}\Lambda X.Q$. For example, if $A = (X \to \bot) \to \bot$, then $N = \lambda x^{((\forall Y.Y) \to \bot) \to \bot}.\Lambda X.\lambda y^{X \to \bot}.x(\lambda z^{\forall Y.Y}.y(z\{X\}))$.

The fact that $X \notin Neg_A$ ensures that N is of type $A[\forall Y.Y/X] \to \forall X.A$ in the Church-style system F. We note u the untyped λ-term obtained by erasing the type indications in N (otherwise said, u is the erasure of N). Going from P to N, we have only modified the type indications appearing in the terms, so u is simply an $\eta\times$-expansion of the identity $\lambda x.x$

Finally, t and u are both equal to the identity through the equalities of the Curry-style system F, so they compose in both ways to give the identity. $\square$

10. Conclusion

We have proved that type isomorphisms in Curry-style system F can be characterised by adding to the equational system of Church-style system F isomorphisms a new, non-trivial equation: $\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if $X \notin Neg_A$. Otherwise said, this equation characterises all the new type equivalences one can generate by erasing type indications in Church-style terms.

We used a game semantics model in order to take advantage of its dynamical and geometrical properties. The main features of the model were however often inspired by a precise analysis of the syntax: indeed, an interpretation of the quantifier as an intersection (or a lower bound like in (Chroboczek, 2003)) was not precise enough to be able to characterise type isomorphisms.

One can notice that our type system does not contain the type $\top$; correspondingly, our model has no empty arena. This is because the rule generally associated to $\top$ takes the form: $t = \star$ if $\Gamma \vdash t : \top$. This rule is of course difficult to insert in a Curry-style setting, where terms are not typed a priori, and we have no clue whether such a rule can be adapted to this context. Anyway, the introduction of an empty arena in the model would break the proof and, more interestingly, give raise to new isomorphisms like $\forall X.(X \to \bot) \simeq_g \bot$. The characterisation of isomorphisms in this model, and the possible connection with an actual syntax, have to be explored.

But the main trail of future exploration concerns parametric polymorphism. The notion of relational parametricity, introduced by Reynolds (Reynolds, 1983), comes historically from the idea that a second-order function shall not depend on the type at which it is instantiated. This has led first to a semantic definition of parametricity, then to a syntactic formalisation of this notion, first by Abadi, Cardelli and Curien (Abadi et al., 1993) and then by Plotkin and Abadi (Plotkin and Abadi, 1993). Dunphy (Dunphy, 2002) recently gave a categorical characterisation of parametric polymorphism.

The great advantage of parametric models is that second-order enjoys nice and natural properties in these models. For example:

— $\forall X.X \to X$ 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.

All these properties are of course wrong in the model described in the present paper.

Trying to build a parametric game model is a highly appealing challenge: one would be glad to extend the concrete notions and flexible features of games into a context where parametricity is understood. Studying isomorphisms in this context would be a natural question, considering the particularly powerful ones corresponding to the above properties.

Longo, Milsted and Soloviev introduced two properties related to the study of parametricity: genericity and the axiom (C) (Longo et al., 1993). The study of these properties in our game semantics context could be done probably more easily, as a first step, than the study of parametricity itself. For instance, the game model given by Abramsky and

Jagadeesan in (Abramsky and Jagadeesan, 2003) was designed in such a way that most types are generic.

Finally, relational parametricity seems to be related to Curry-style system F, if we believe in a conjecture of Abadi-Cardelli-Curien which says the following: suppose you have two terms of type A whose type erasures are the same. Then they are parametrically equal (the converse is false). This means that the parametric equality is (strictly) stronger than the Curry-style equality: the study on both Curry-style system F and parametricity in the context of games may help to explore this question.

References

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

Abramsky, S. and Jagadeesan, R. (2003). A game semantics for generic polymorphism. In Gordon, A. D., editor, *Foundations of Software Science and Computational Structures*, volume 2620 of *LNCS*, pages 1–22. Springer.

Abramsky, S., Jagadeesan, R., and Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation*, 163(2):409–470.

Abramsky, S. and McCusker, G. (1999). Full abstraction for idealized algol with passive expressions. *Theoretical Computer Science*, 227:3–42.

Barendregt, H. (1984). *The lambda calculus, its syntax and semantics*. Number 103 in Studies in Logic and the Foundations of Mathematics. North-Holland, second edition.

Barthe, G. and Pons, O. (2001). Type isomorphisms and proof reuse in dependent type theory. In Honsell, F. and Miculan, M., editors, *Foundations of Software Science and Computation Structures*, volume 2030 of *LNCS*.

Chroboczek, J. (2003). *Game Semantics and Subtyping*. Ph.D. thesis, University of Edinburgh.

de Lataillade, J. (2007). Second-order type isomorphisms through game semantics. To appear in *Annals of Pure and Applied Logic*, Special Issue on Game Semantics. Available at `http://www.pps.jussieu.fr/~delatail/isotypes.pdf` .

Di Cosmo, R. (1995). *Isomorphisms of Types*. Progress in Theoretical Computer Science. Birkhäuser.

Dunphy, B. P. (2002). *Parametricity as a notion of uniformity in reflexive graphs*. Ph.D. thesis, University of Illinois.

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

Harmer, R. (1999). *Games and Full Abstraction for Nondeterministic Languages*. Ph.D. thesis, Imperial College and University of London.

Hughes, D. (2000). *Hypergame semantics: full completeness for system F*. D.Phil. thesis, Oxford University.

Hyland, M. and Ong, L. (2000). On full abstraction for PCF. *Information and Computation*, 163(2):285–408.

Klop, J. W. (1978). A counterexample to the church-rosser property for lambda calculus with subjective pairing. Prépublication, University of Utrecht.

Laird, J. (1997). Full abstraction for functional languages with control. In *Proceedings of the twelfth annual symposium on Logic In Computer Science*, pages 58–67, Warsaw. IEEE, IEEE Computer Society Press.

Longo, G., Milsted, K., and Soloviev, S. (1993). 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.

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

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

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

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

Rittri, M. (1991). Using types as search keys in function libraries. *Journal of Functional Programming*, 1(1):71–89.

Appendix A. Uniform strategies compose

Proposition 2. If $\sigma :: A \to B$ and $\tau :: B \to C$ are two uniform strategies then $\sigma;\tau :: A \to C$ is uniform.

Proof. Consider the following strategy

$$\bar{\rho} = \{u\upharpoonright_{\uparrow,\downarrow\downarrow} \mid u \in \mathbf{Int} \wedge u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma \wedge u\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau \wedge u\upharpoonright_{\uparrow,\downarrow\downarrow} \text{ symbolic play}\}$$

It is an innocent strategy on $A \to C$ (because it is a restriction of $\sigma;\tau$, which is innocent like in HO models), and it is of course symbolic. We call ρ its copycat extension, and we want to prove that $\rho = \sigma;\tau$.

We first prove that $\rho \subseteq \sigma;\tau$, which means that$\sigma;\tau$ contains all copycat extensions of $s \in \bar{\rho}$ along a copycat variable X_j.

Let us consider the play $s' = Fl^s_{j,D}(r)$ for $s = m_1 \dots m_n \in \bar{\rho}$, $D \in \mathcal{G}$ and r sequence of initial moves in $\mathcal{M}_D$. There exists a justified sequence u and two plays $s_1 \in \sigma$ and $s_2 \in \tau$ such that $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s_1$, $u\upharpoonright_{\uparrow,\downarrow\uparrow} = s_2$ and $u\upharpoonright_{\uparrow,\downarrow\downarrow} = s$. s_1 (resp. s_2) is not necessarily symbolic, but its views are copycat extensions of symbolic plays, and furthermore **O** plays symbolically for moves of shape $\downarrow\downarrow$ (resp. $\uparrow$).

Let U_0 be be the justified sequence obtained from u by replacing each sequence of moves $m_i b_1 \dots b_q m_{i+1}$ with $\sharp(m_i) = j$ by $m_i[r_i]b_1[r_i]\dots b_q[r_i]m_{i+1}[r_i]$, and let us set $U = U_0[D/j]$. Then $U\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s'_1 \in \sigma$ because all the views of s'_1 are copycat extensions of views of s_1 along copycat variables. Similarly $U\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$, so $U\upharpoonright_{\uparrow,\downarrow\downarrow} = s' \in \sigma;\tau$.

Let us now consider a move m_i of s such that $\sharp(m_i) = j$, and a bi-view $v = n_1 \dots n_p$ in the arena D, and let us set $S = CC^s_{j,D}(i,v,r)$. If $U = U_1 m'_i[r_i]b_1[r_i]\dots b_q[r_i]m'_{i+1}[r_i]U_2$ with $m'_i = m_i[D/j]$ et $m'_{i+1} = m_{i+1}[D/j]$, we build another justified sequence U', defined as follows:

— if $p = 1$, $U' = U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]U_2$

— if p even,

$$\begin{array}{rcl} U' & = & U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]b_q[n_2]\dots b_1[n_2]m'_i[n_2] \\ & & \dots\dots m'_{i+1}[n_p]b_q[n_p]\dots b_1[n_p]m'_i[n_p] \end{array}$$

— if p odd and $p > 1$,

$$\begin{array}{rcl} U' & = & U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]b_q[n_2]\dots b_1[n_2]m'_i[n_2] \\ & & \dots\dots m'_i[n_p]b_1[n_p]\dots b_q[n_p]m'_{i+1}[n_p] \end{array}$$

$U'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ is a copycat extension of s_1 (s'_1 was the flat extension) so $U'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$, and similarly $U'\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$. $U'\upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play so $U'\upharpoonright_{\uparrow,\downarrow\downarrow} = S \in \sigma;\tau$.

The last thing to prove is that $\sigma;\tau \subseteq \rho$. We suppose that σ and τ are the copycat extensions of the symbolic strategies $\bar{\sigma}$ and $\bar{\tau}$ respectively. Consider a view $s \in \sigma;\tau$, there exists a justified sequence U for which $U\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s_1 \in \sigma$, $U\upharpoonright_{\uparrow,\downarrow\uparrow} = s_2 \in \tau$ and $U\upharpoonright_{\uparrow,\downarrow\downarrow} = s$. We allow ourselves to identify a move of U with its projection along $\{\uparrow,\downarrow\downarrow\}$, $\{\uparrow,\downarrow\uparrow\}$ or $\{\downarrow\uparrow,\downarrow\downarrow\}$.

Our goal is to prove that s is obtained from $\bar{\rho}$ by a certain number of copycat extensions.

We will build, move after move, a justified sequence $\overline{u}$ such that s is an extension of $\overline{u}\upharpoonright_{\uparrow,\downarrow\downarrow} \in \rho$.

Let u be a justified sequence, prefix of U, we will define by induction a justified sequence $\overline{u}$ such that:

— $\overline{u}\upharpoonright_{\uparrow,\downarrow\downarrow}$ symbolic play
— $\ulcorner u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner$ extension of $\ulcorner\overline{u}\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner$
— $\ulcorner u\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ extension of $\ulcorner\overline{u}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$
— $u\upharpoonright_{\uparrow,\downarrow\downarrow}$ extension of $\overline{u}\upharpoonright_{\uparrow,\downarrow\downarrow}$.

If $u = \epsilon$, we set $\overline{u} = \epsilon$.

If $u = u_0 m$, as the interaction mechanism is the same as in HO-games, we can reuse the results on state automata (cf (Harmer, 1999)). So we have three cases:

— Case #1: $u_0\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$, $u_0\upharpoonright_{\uparrow,\downarrow\uparrow}$ and $u_0\upharpoonright_{\uparrow,\downarrow\downarrow}$ are of even length. Then m is of shape $\uparrow$ or $\downarrow\downarrow$. In the case where m is of shape $\uparrow$, we have $\ulcorner u_0\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner mn \in \tau$ for some n, and as τ is uniform there exists a symbolic play $s_2\bar{m}\bar{n}$ for which $\ulcorner u_0\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner mn$ is an extension. As we also have by induction a play $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ for which $\ulcorner u_0\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ is an extension, this means that $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ is an extension of s_2. Applying to $s_2\bar{m}$ all the copycat extensions that necessary to transform s_2 into $\ulcorner\overline{u_0}\urcorner\upharpoonright_{\uparrow,\downarrow\uparrow}$, we obtain the play $\ulcorner\overline{u_0}\urcorner\upharpoonright_{\uparrow,\downarrow\uparrow} m'$, we set $\overline{u} = \overline{u_0}m'$ and we can check that $\overline{u}$ satisfies all the required properties. The case where m is of shape $\downarrow\downarrow$ is similar.
— Case #2: $u_0\upharpoonright_{\uparrow,\downarrow\uparrow}$ is of even length, $u_0\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ and $u_0\upharpoonright_{\uparrow,\downarrow\downarrow}$ are of odd length. Then m is of shape $\downarrow\uparrow$ or $\downarrow\downarrow$. In both cases, we have $u_0 = u_1 n$ with $u_1\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} nm \in \sigma$ for some n, and we set $\ulcorner u_1\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} n\urcorner = s_0 n$. As σ is uniform there exists a symbolic play $s_1\bar{m}$ for which $s_0 nm$ is an extension. As we also have by induction a play $\ulcorner\overline{u_0}\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner$ for which $s_0 n$ is an extension, this means that $\ulcorner\overline{u_0}\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner$ is an extension of s_1. Applying to $s_1\bar{m}$ all the copycat extensions necessary to transform s_1 into $\ulcorner\overline{u_0}\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner$, we obtain the play $\ulcorner\overline{u_0}\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}\urcorner m'$, we set $\overline{u} = \overline{u_0}m'$ and we can check that $\overline{u}$ satisfies all the required properties.
— Case #3: $u_0\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ is of even length, $u_0\upharpoonright_{\uparrow,\downarrow\uparrow}$ and $u_0\upharpoonright_{\uparrow,\downarrow\downarrow}$ are of odd length. Then m is of shape $\uparrow$ or $\downarrow\uparrow$. In both cases, we have $u_0 = u_1 n$ with $u_1\upharpoonright_{\uparrow,\downarrow\uparrow} nm \in \tau$ for some n, and we set $\ulcorner u_1\upharpoonright_{\uparrow,\downarrow\uparrow} n\urcorner = s_0 n$. As τ is uniform there exists a symbolic play $s_1\bar{m}$ for which $s_0 nm$ is an extension. As we also have by induction a play $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ for which $s_0 n$ is an extension, this means that $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$ is an extension of s_1. Applying to $s_1\bar{m}$ all the copycat extensions necessary to transform s_1 into $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner$, we obtain the play $\ulcorner\overline{u_0}\upharpoonright_{\uparrow,\downarrow\uparrow}\urcorner m'$, we set $\overline{u} = \overline{u_0}m'$ and we can check that $\overline{u}$ satisfies all the required properties.

□

Appendix B. Proof of $A \simeq_g B \Rightarrow H_A \simeq_{Cu} H_B$

For the sake of simplicity, we will from now on identify the occurrences of O_A (resp. of O_B) with the corresponding occurrences of $O_{A\to B}$ or $O_{B\to A}$.

Before proving the implication $A \simeq_g B \Rightarrow H_A \simeq_{Cu} H_B$ we need a first lemma:

Lemma 11. Let $s_1 \in \mathcal{P}_{B\to A}$ and $s_2 \in \mathcal{P}_{A\to B}$ be two even-length plays such that:

— s_1 and s_2 are **zig-zag**, which means:
  - each Player move following an Opponent move of shape $\uparrow$ (resp. $\downarrow$) is of shape $\downarrow$ (resp. $\uparrow$)
  - each Player move following an initial Opponent move is justified by it
  - $s\upharpoonright_{\downarrow}$ and $s\upharpoonright_{\uparrow}$ have the same pointers

— $erase(s_1\upharpoonright_{\uparrow}) = erase(s_2\upharpoonright_{\downarrow})$

— the arenas played by **O** in s_1 (resp. dans s_2) at the level of a move of shape $\uparrow$ (resp. $\downarrow$) are copycat variables.

Then there exists a justified sequence u such that: $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ flat extension of s_1, $u\upharpoonright_{\uparrow,\downarrow\uparrow}$ flat extension of s_2 and $u\upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play.

Proof. We build the sequence u by induction on n where $2n$ is the length of the play s_1.

If $n = 0$, $u = \epsilon$.

If $n = p + 1$ with p even, we have $s_2 = S_2 m_2 n_2$ and $s_1 = S_1 n_1 m_1$. We already have by induction a sequence u_0 such that $S_1' = u_0\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ flat extension of S_1 and $S_2' = u_0\upharpoonright_{\uparrow,\downarrow\uparrow}$ flat extension of S_2. We can write $n_2 = n_0[M]$ and $n_1 = n_0'[M']$ with $c = \mathcal{A}(n_0) = \mathcal{A}(n_0') \in O_B$.

If $paux_{A\to B}(c) = \mathbf{O}$, we have $M = i$ for some i. By applying to s_1 the extensions that transform S_1 into S_1', we obtain a flat extension $S_1' N_1^0 M_1^0$. By applying the extensions that transform n_0' into n_0 we obtain a new flat extension $S_1' N_1 M_1$. Then we apply to s_2 the flat extensions that transform S_2 into S_2' and those which transform i into M', and we obtain a flat extension $S_2' M_2 N_1$. We then set $u = u_0 M_2 N_1 M_1$.

If $paux_{A\to B}(c) = \mathbf{P}$, we have $M' = i$ for some i. By applying to s_2 the extensions which transform S_2 into S_2', we obtain a flat extension $S_2' M_2 N_2$. Then we apply to s_1 the flat extensions which transform S_1 into S_1', those who transform n_0' into n_0 and those which transform i into M, and we obtain a flat extension $S_1' N_2 M_1$. We then set $u = u_0 M_2 N_2 M_1$.

Finally, if $paux_{A\to B}(c)$ is not defined, we have $M' = M = i$. We apply to s_2 the flat extensions which transform S_2 into S_2', we obtain $S_2' M_2 N_2$; we apply to s_1 the extensions that transform S_1 into S_1' and those which transform n_0' into n_0, we obtain $S_1' N_2 M_1$. We then set $u = u_0 M_2 N_2 M_1$.

If $n = p + 1$ with p odd, we have $s_1 = S_1 m_1 n_1$ and $s_2 = S_2 n_2 m_2$. We already have by induction a sequence u_0 such that $S_1' = u_0\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ flat extension of S_1 and $S_2' = u_0\upharpoonright_{\uparrow,\downarrow\uparrow}$ flat extension of S_2. We can write $n_1 = n_0[M]$ and $n_2 = n_0'[M']$ with $c = \mathcal{A}(n_0) = \mathcal{A}(n_0') \in O_B$.

If $paux_{A\to B}(c) = \mathbf{O}$, we have $M' = i$ for some i. By applying to s_1 the extensions which transform S_1 into S_1', we obtain a flat extension $S_1' M_1 N_1$. We apply to s_2 the flat extensions which transform S_2 into S_2', those which transform n_0' into n_0 and those which transform i into M, and we obtain a flat extension $S_2' N_1 M_2$. We then set $u = u_0 M_1 N_1 M_2$.

If $paux_{A\to B}(c) = \mathbf{P}$, we have $M = i$ for some i. By applying to s_1 the extensions which transform S_1 into S'_1, we obtain a flat extension $S'_1M^0_1N^0_1$. By applying then the extensions which transform i into M' we obtain a new flat extension $S'_1M_1N_1$. We apply to s_2 the extensions which transform S_2 into S'_2 and those which transform n'_0 into n_0, and we obtain a flat extension $S'_2N_1M_2$. We then set $u = u_0M_1N_1M_2$.

Finally, if $paux_{A\to B}(c)$ is not defined then $M' = M = i$. We apply to s_2 the flat extensions which transform S_2 into S'_2 and those which transform n'_0 into n_0, we obtain $S'_2N_2M_2$; we apply to s_1 the extensions which transform S_1 into S'_1, we obtain $S'_1M_1N_2$. We then set $u = u_0M_1N_2M_2$.

To show that $u\restriction_{\uparrow,\downarrow\downarrow}$ is a play, it suffices to remark that, in each case of the above iteration, we have that $\sharp(M_2) = \sharp(M_1)$, and that, in $A \to A$, the polarity of M_2 is the opposite of the polarity of M_1. □

Theorem 1. Let $A, B \in \mathcal{G}$. If there exists a total game isomorphism (σ, τ) between A and B ($A \simeq_g B$) then their hyperforests are Curry-isomorphic ($H_A \simeq_{Cu} H_B$).

Proof. The proof of this theorem will take different steps:

— we first build two typed plays s^p_1 and s^p_2 whose respective erasures are the plays

$$a_1[i_1]f(a_1)[i_1]f(a_2)[i_2]a_2[i_2]\cdots \in \sigma$$
$$f(a_1)[i_1]a_1[i_1]a_2[i_2]f(a_2)[i_2]\cdots \in \tau$$

with $i_1, i_2, \cdots \in \mathbb{N}$

— these two plays give a bijection $f : \mathcal{F}_A \to \mathcal{F}_B$ such that

$$a_1[y_1]f(a_1)[y_1]f(a_2)[y_2]a_2[y_2]\cdots \in \sigma$$
$$f(a_1)[z_1]a_1[z_1]a_2[z_2]f(a_2)[z_2]\cdots \in \tau$$

for any choice of the moves y_i and z_i, by hyperuniformity

— we then build, move by move, two plays s_p et u_p, belonging to the total realisations of σ and τ respectively, such that their erasures take the above form for an adequate choice of y_i and z_i

— the choice of arenas and the conditions on the erasure of s_p and u_p will allow us to conclude that f satisfies the conditions of a Curry-isomorphism.

Construction of the bijection

Let a be a node of $\mathcal{F}_A$ and $a_1, \ldots, a_p$ be the sequence of nodes of $\mathcal{F}_A$ such that $\vdash a_1$, $a_i \vdash a_{i+1}$ and $a_p = a$. From now on, we will simply denote a_i the occurrence $or(a_i)$; similarly, for a node b of $\mathcal{F}_B$, we will denote b the occurrence $or(b)$. Finally, we note $\tilde{\sigma}$ and $\tilde{\tau}$ two total realisations of σ and τ respectively.

We are going to build, by induction on p, a function $f : \{a_1, \dots, a_p\} \to \mathcal{F}_B$ and two zig-zag plays $s_1^p \in \tilde{\sigma}$ and $s_2^p \in \tilde{\tau}$ such that[††]:

— $\vdash f(a_1)$ and $f(a_i) \vdash f(a_{i+1})$ for $1 \leq i < n$
— $s_1^p \upharpoonright_\uparrow = s_2^p \upharpoonright_\downarrow$
— every arena played by **O** in s_1^p (resp. s_2^p) at the level of a move of shape $\downarrow$ (resp. $\uparrow$) is a copycat variable
— $\mathcal{A}(s_2^p) = a_1[b_1] f(a_1)[c_1] f(a_2)[c_2] a_2[b_2] \dots$
— $\mathcal{A}(s_1^p) = f(a_1)[c_1] a_1[d_1] a_2[d_2] f(a_2)[c_2] \dots$
— $\vdash b_i$ and $\mathcal{E}(b_i) = \mathcal{E}(c_i) = \mathcal{E}(d_i) \in \mathbb{N}$ for $1 \leq i \leq n$.

If $p = 0$ then we set $s_1^0 = s_2^0 = \epsilon$.

If $p = p' + 1$ with p' even, $\mathcal{A}(s_2^{p'}) = a_1[b_1] f(a_1)[c_1] \dots f(a_{p'})[c_{p'}] a_{p'}[b_{p'}]$ and $\mathcal{A}(s_1^{p'}) = f(a_1)[c_1] a_1[d_1] \dots a_{p'}[d_{p'}] f(a_{p'})[c_{p'}]$.
We have different cases :

— First consider the case where $paux_{B \to A}(a_p) = \mathbf{O}$. Then $s_2^{p'} m \in \mathcal{P}_{B \to A}$ with $m = m_0[i]$ for some i, $\mathcal{A}(m_0) = a_p$ and every arena played at the level of m is a copycat variable. By totality of $\tilde{\tau}$, $s_2^{p'} mn \in \tilde{\tau}$ for some n, we note $n = m_1[m_2]$, $\mathcal{A}(n) = b \in \mathcal{O}_B$ ($\mathcal{A}(m_1) \in \mathcal{O}_A$ would contradict the fact that $\sigma;\tau \subseteq id$) and $m_2 = M[i]$.
By totality of $\tilde{\sigma}$ there exists m' such that $s_1^{p'} nm' \in \tilde{\sigma}$. We set $m' = m'_1[m'_2]$ with $\mathcal{A}(m'_1) \in \mathcal{O}_{A \to B}$.
Is it possible to have $\mathcal{A}(m'_1) \in \mathcal{O}_B$? In that case, we take the symbolic plays S_1 and S_2 from which $s_1^{p'} nm'$ and $s_2^{p'}$ respectively come from. By using lemma 11, we can build the justified sequence u such that $u \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ is a flat extension of S_2, $u \upharpoonright_{\uparrow,\downarrow\uparrow}$ is a flat extension of S_1 and $u \upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play. Then $u \upharpoonright_{\uparrow,\downarrow\downarrow} \in \tilde{\tau};\tilde{\sigma}$, and $u \upharpoonright_{\uparrow,\downarrow\downarrow} NM'$ where N and M' are ontained from n and m' by flat extensions so $u \upharpoonright_{\uparrow,\downarrow\downarrow} NM' \in id$. But this play contains two successive moves of shape $\uparrow$, which is impossible.
So, $\mathcal{A}(m'_1) \in \mathcal{O}_A$. As $\sigma;\tau \subseteq id$, one has $erase(m') = erase(m)$, so $erase(m'_1) = \mathcal{E}(a_p)$, thus $\mathcal{A}(m'_1) = a_p$ by non-ambiguity of $\mathcal{O}_A$, and besides $erase(m'_2) = i$.
If $p = 1$ then $\vdash b$. If $p > 1$, m' is justified in $s_1^{p'} nm'$ by the move n' such that $\mathcal{A}(n') = a_{p'}[b_{p'}]$. So, the justifier of n cannot be played before n', by innocence of $\tilde{\sigma}$, which means that n is justified by the move n'' of $s_1^{p'}$ such that $\mathcal{A}(n'') = f(a_{p'})[c_{p'}]$. So, $f(a_{p'}) \vdash b$.
We set $s_1^p = s_1^{p'} nm'$ and $s_2^p = s_2^{p'} mn$, which means: $f(a_p) = f(a_1) \dots f(a_{p'}) b$ (so that $or(f(a_p)) = b$), $b_p = i$, $c_p = M[i]$ and $d_p = m'_2$. The proof that $erase(M[i]) = i$ depends on the auxiliary polarity of b.
If $paux_{B \to A}(b) = \mathbf{O}$ then the symbolic play from which s_2^p comes from is $S\tilde{a}[j]\tilde{b}[j] \in \tilde{\tau}$ with $\mathcal{A}(\tilde{a}) = a_p$ and $\mathcal{A}(\tilde{b}) = b$. s_2^p being a copycat extension of this play, we have $M[i] = i$, so $erase(M[i]) = i$.

[††] The last condition, which is also the most important, is inspired by the condition of *genericity* of (Abramsky and Jagadeesan, 2003). But here we will prove this property for the strategies realising an isomorphism, whereas in (Abramsky and Jagadeesan, 2003) it is required for every strategy.

If $paux_{B\to A}(b) = \mathbf{P}$ then $paux_{A\to B}(b) = \mathbf{O}$. The symbolic play from which s_1^p comes from is $S\tilde{b}[j]\tilde{a}[M'[j]] \in \tilde{\tau}$ with $\mathcal{A}(\tilde{b}) = b$ and $\mathcal{A}(\tilde{a}) = a_p$. So by retrieving the copycat extension we obtain $s_1^{p'} nm' = s_1^{p'} m_1[M[i]]m_1'[M'[M[i]]]$, so finally $erase(M'[M[i]]) = i$, which implies $erase(M[i]) = i$.

The case where $paux_{B\to A}(b)$ is undefined is impossible because in this case $\sharp(b) = 0 \neq i$.

— Let us now consider the case where $paux_{B\to A}(a_p) = \mathbf{P}$. Then $s_2^{p'} m \in \mathcal{P}_{B\to A}$ with $m = m_0[M]$ for some M with $\vdash M$, $\mathcal{A}(m_0) = a_p$ and and every arena played at the level of m is a copycat variable. By totality of $\tilde{\tau}$, $s_2^{p'} mn \in \tilde{\tau}$ for some n, we note $n = m_1[m_2]$ and $\mathcal{A}(n) = b \in \mathcal{O}_B$ ($\mathcal{A}(m_1) \in \mathcal{O}_A$ would contradict the fact that $\sigma;\tau \subseteq id$).

By totality of $\tilde{\sigma}$ there exists m' such that $s_1^{p'} nm' \in \tilde{\sigma}$. We set $m' = m_1'[m_2']$ with $\mathcal{A}(m_1') \in \mathcal{O}_{A\to B}$.

Once again, is it possible to have $\mathcal{A}(m_1') \in \mathcal{O}_B$? We do the same reasoning as above: we suppose it is the case, and we take the symbolic plays S_1 and S_2 from which $s_1^{p'} nm'$ and $s_2^{p'}$ respectively come from. By using lemma 11, we can build the justified sequence u such that $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ is a flat extension of S_2, $u\upharpoonright_{\uparrow,\downarrow\uparrow}$ is a flat extension of S_1, and $u\upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play. Then $u\upharpoonright_{\uparrow,\downarrow\downarrow} \in \tilde{\tau};\tilde{\sigma}$, and $u\upharpoonright_{\uparrow,\downarrow\downarrow} NM' \in \tilde{\tau};\tilde{\sigma}$ where N and M' are obtained from n and m' by flat extensions, so $u\upharpoonright_{\uparrow,\downarrow\downarrow} NM' \in id$. But this play contains two successive moves of shape $\uparrow$, which is impossible.

So, $\mathcal{A}(m_1') \in \mathcal{O}_A$. As $\sigma;\tau \subseteq id$, one has $erase(m') = erase(m)$, so $erase(m_1') = \mathcal{E}(a_p)$, thus $\mathcal{A}(m_1') = a_p$ by non-ambiguity of $\mathcal{O}_A$, and besides $erase(m_2') = M$. But $paux_{A\to B}(a_p) = \mathbf{O}$, so $m_2' = i$ for some $i \in \mathbb{N}$.

m' is justified in $s_1^{p'} nm'$ by the move n' such that $\mathcal{A}(n') = a_{p'}[b_{p'}]$. So, the justifier of n cannot be played before n', by innocence of $\tilde{\sigma}$, which means that n is justified by the move n'' of $s_1^{p'}$ such that $\mathcal{A}(n'') = f(a_{p'})[c_{p'}]$. So, $f(a_{p'}) \vdash b$.

We set $s_1^p = s_1^{p'} nm'$ and $s_2^p = s_2^{p'} mn$, which means: $f(a_p) = f(a_1)\dots f(a_{p'})b$ (so that $or(f(a_p)) = b$), $b_p = M$, $c_p = m_2$ and $d_p = i$. The proof that $erase(m_2) = i$ depends on the auxiliary polarity of b.

If $paux_{B\to A}(b) = \mathbf{O}$ then the symbolic play from which s_2^p comes from is $S\tilde{a}[M'[j]]\tilde{b}[j] \in \tilde{\tau}$ with $\mathcal{A}(\tilde{a}) = a_p$ and $\mathcal{A}(\tilde{b}) = b$. So by retrieving the appropriate copycat extension one obtains $M = M'[m_2]$. As $erase(M) = i$, it implies $erase(m_2) = i$.

If $paux_{B\to A}(b) = \mathbf{P}$ then $paux_{A\to B}(b) = \mathbf{O}$ and $paux_{A\to B}(a_p) = \mathbf{O}$. The symbolic play from which s_1^p comes from is $S\tilde{b}[j]\tilde{a}[j] \in \tilde{\tau}$ with $\mathcal{A}(\tilde{b}) = b$ and $\mathcal{A}(\tilde{a}) = a_p$. By retrieving the appropriate copycat extension one obtains $m_2 = i$, so $erase(m_2) = i$.

In the case where $paux_{B\to A}(b)$ is undefined, then the symbolic play from which s_1^p comes from is $S\tilde{b}\tilde{a}[i]$ with $\mathcal{A}(\tilde{b}) = b$ and $\mathcal{A}(\tilde{a}) = a_p$. Which is impossible since in this case $\sharp(b) = 0 \neq i$.

— Finally, if $paux_{B\to A}(a_p)$ is undefined one has $s_2^{p'} m \in \mathcal{P}_{B\to A}$ with $\mathcal{A}(m) = a_p$ and every arena played at the level of m is a copycat variable. By totality of $\tilde{\tau}$, $s_2^{p'} mn \in \tilde{\tau}$ for some n, we note $n = m_1[m_2]$, $\mathcal{A}(n) = b \in \mathcal{O}_B$ ($\mathcal{A}(n) \in \mathcal{O}_A$ would contradict the fact that $\sigma;\tau \subseteq id$).

By totality of $\tilde{\sigma}$ one has $s_1^{p'} nm' \in \tilde{\sigma}$ for some $m' = m_1'[m_2']$ with $\mathcal{A}(m_1') \in \mathcal{O}_{A\to B}$. Once again we prove that $\mathcal{A}(m_1') \in \mathcal{O}_A$, $\mathcal{A}(m_1') = a_p$ and $erase(m') = erase(m)$ so $erase(m_2') = 0$.

We set $s_1^p = s_1^{p'} nm'$ and $s_2^p = s_2^{p'} mn$, which means: $f(a_p) = f(a_1)\dots f(a_{p'})b$ (so that $or(f(a_p)) = b$), $b_p = 0$, $c_p = m_2$ and $d_p = m'_2$. The proof that $erase(m_2) = 0$ depends on the auxiliary polarity of b.
If $paux_{B\to A}(b) = \mathbf{O}$ then the symbolic play from which s_2^p comes from is $S\tilde{a}\tilde{b}[j] \in \tilde{\tau}$ with $\mathcal{A}(\tilde{b}) = b$ and $\mathcal{A}(\tilde{a}) = a_p$. But this is impossible since $\sharp(a) = 0$.
If $paux_{B\to A}(b) = \mathbf{P}$ then the symbolic play from which s_1^p comes from is $S\tilde{b}[j]\tilde{a} \in \tilde{\tau}$ with $\mathcal{A}(\tilde{b}) = b$ and $\mathcal{A}(\tilde{a}) = a_p$. But this is impossible since $\sharp(a) = 0$.
So, $paux_{B\to A}(b)$ is not defined, and $m_2 = 0$. Hence $erase(m_2) = 0$.

If $p = p' + 1$ with p' odd, the reasoning is similar except that we define s_1^p before s_2^p.

To see that f is a bijection, let us consider for example the case $p = p' + 1$ with p' even.
As $S_1^{p'} m_1[i]m_0[i] \in \tilde{\sigma}$, we have $S_1\mathcal{E}(f(a_p))[i]\mathcal{E}(a_p)[i] \in \sigma$ with $S_1 = erase(S_1^{p'})$, and it is true for any $i \in \mathbb{N}$ by hyperuniformity.
If $f(a_p) = f(a'_p)$, one has $erase(s_1^p) = S_1\mathcal{E}(f(a_p))[i]\mathcal{E}(a_p)[i]$ for some i, so $S_1\mathcal{E}(f(a'_p))[i]\mathcal{E}(a_p)[i] \in \sigma$. We also have $S_1\mathcal{E}(f(a'_p))[j]\mathcal{E}(a'_p)[j] \in \sigma$ for some j, so $S_1\mathcal{E}(f(a'_p))[i]\mathcal{E}(a'_p)[i] \in \sigma$ by hyperuniformity, thus $a_p = a'_p$ by determinisme of σ.
Consider now b such that $f(a_{p'}) \vdash b$. We take the symbolic plays S_1 and S_2 from which $s^{p'_1}$ and $s_2^{p'}$ respectively come from. By using lemma 11, we can build the justified sequence u such that $S_2 = u\restriction_{\downarrow\uparrow,\downarrow\downarrow}$ is a flat extension of $s_2^{p'}$, $S_1 = u\restriction_{\uparrow,\downarrow\uparrow}$ is a flat extension of $s_1^{p'}$ and $u\restriction_{\uparrow,\downarrow\downarrow}$ is a play. By totality of $\tilde{\sigma}$ we have $S_1m_1[M]m_2[M'] \in \tilde{\sigma}$ with $\mathcal{A}(m_1) = b$ and $\mathcal{A}(m_2) \in \mathcal{O}_A$. By totality of $\tilde{\tau}$ we have $S = S_2m_2[M']m' \in \tilde{\tau}$. As $\sigma;\tau \subseteq id$ we have $erase(m') = erase(m_1[M])$, and by innocence one shows that $a_{p'} \vdash a$. $erase(S)$ is a copycat extension of $erase(s_1^p)$ so $erase(m') = \mathcal{E}(f(a))$. Hence $b = f(a)$.

We now consider the symbolic plays S_1^p and S_2^p from which s_1^p and s_2^p respectively come from. We still have

$$\mathcal{A}(S_2^p) = a_1[b'_1]f(a_1)[c'_1]f(a_2)[c'_2]a_2[b'_2]\dots$$
$$\mathcal{A}(S_1^p) = f(a_1)[c'_1]a_1[d'_1]a_2[d'_2]f(a_2)[c'_2]\dots$$

with $\mathcal{E}(b'_i) = \mathcal{E}(c'_i) = \mathcal{E}(d'_i) \in \mathbb{N}$ for $1 \le i \le n$: indeed, s_2^p (resp. s_1^p) is the copycat extension of S_2^p (resp. S_1^p), so if $\mathcal{E}(c'_i) \neq \mathcal{E}(b'_i)$ (resp. $\mathcal{E}(d'_i) \neq \mathcal{E}(c'_i)$) then $\mathcal{E}(c_i) \neq \mathcal{E}(b_i)$ (resp. $\mathcal{E}(d_i) \neq \mathcal{E}(c_i)$).

Construction of the appropriate plays

From now on, we identifiate the occurrences a_i and $f(a_i)$ with the untyped moves $\mathcal{E}(a_i)$ and $\mathcal{E}(f(a_i))$ respectively: it does not lead to any confusion thanks to the non-ambiguity of $\mathcal{O}_A$ and $\mathcal{O}_B$.
To prove that f satisfies the requirements of a Curry-isomorphism, we will extend the play S_1^p into a play $s_p \in \tilde{\sigma}$ with an appropriate choice of the arenas played by $\mathbf{O}$. The integer i such that $S_1^p = S_0a_p[i]$ or $S_1^p = S_0f(a_p)[i]$ will be replaced by an untyped move y_p, also chosen in an appropriate way.
In the plays s_p, we will use the arenas $(C_j)_{j\in\mathbb{N}}$ defined by: $C_1 = \bot\times\bot$ and $C_{j+1} = C_j\times C_j$.

Note that each initial move of C_j takes the form $b_1(b_2(\ldots(b_j(0))\ldots))$, where each b_i can be either r or l. We call r_j the initial move of C_j where each b_i is equal to r. These arenas will be used in order to have *fresh* moves, i.e. moves that cannot come from an arena defined before C_j is played. In what follows, the integer n_p is made to ensure that no arena defined before step p can belong to C_q for $q \geq p$.

Finally, we choose a function $\rho_\_ : \mathcal{G} \to \mathbb{A}$ which associates to each arena $D \in \mathcal{G}$ a move $\rho_D \in \mathcal{M}_D$ satisfying the following conditions:

— $\vdash \rho_D$
— if there exists $c \in O_D$ such that $\vdash c$ and $\mathcal{L}_D(c) \neq \dagger$, then $\rho_D = m_1[m_2]$ with $\mathcal{A}(m_1) = d$, $\mathcal{L}_D(d) \neq \dagger$, $\frac{m_1}{\mathcal{L}_D(c)} = \bot \times \bot$ and $m'_3 = l0$.

We now build the triple (s_p, y_p, n_p) inductively:

— If $p = 1$, we define the typed move $M_1 = m_1[m_2]$ such that: $\mathcal{A}(m_1) = f(a_1)$, the d_1 arenas played at the level of m_1 are $C_1, \ldots, C_{d_1}$ and we choose $m_2 = \sharp(f(a_1))$ if $paux_{A\to B}(f(a_1))$ is undefined and $m_2 = r_j$ if $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_1))} = C_j$. Let $s_1 = M_1M'_1$ be the copycat extension of S_1^p corresponding to these choices, we have $erase(M_1M'_1) = f(a_1)[y_1]a_1[y_1]$ where $y_1 = erase(m_2)$. If N is the biggest number of tokens r in any initial occurrence of an arena D defined at the level of M'_1, we choose $n_1 = max(d_1, N) + 1$.
— If $p = p' + 1$ with p' odd, we define the typed move $M_p = m_1[m_2]$ such that: $erase(m_1) = f(a_p)$, the d_p arenas defined at the level of m_1 are $C_{n_{p'}}, \ldots, C_{n_{p'}+d_p}$ and m_2 is chosen as follows:
  – if $paux_{A\to B}(f(a_p))$ is undefined, $m_2 = \sharp(f(a_p))$
  – if $paux_{A\to B}(f(a_p)) = \mathbf{O}$, $m_2 = r_j$ if $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_j$
  – if $paux_{A\to B}(f(a_p)) = \mathbf{P}$, let $D = \frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))}$. We choose $m_2 = \rho_D$, and we note $r_D = erase(m_2)$ ‡‡

  Let $s_p = s_{p'}M_pM'_p$ be the copycat extension of S_1^p corresponding to these choices, we have $erase(s_{p'}M_pM'_p) = erase(s_{p'})f(a_p)[y_p]a_1[y_p]$ if $y_p = erase(m_2)$. If N is the biggest number of tokens r in any initial occurrence of an arena D defined at the level of M'_p, we choose $n_p = max(n_{p'} + d_p, N) + 1$.
— If $p = p' + 1$ with p' odd, we do the same choices as in the preceding case, except that $f(a_p)$ must be replaced by a_p, and conversely.

Suppose $paux_{A\to B}(a_p)$ is defined, then $ref_{A\to B}(a_p) = b$ is also defined. It is important for the next section of the proof to understand the link between b and the play s_p. First, note that $b = a_i$ for some $1 \leq i \leq p$; then, because of the definition of the set $\mathcal{R}_{A\to B}$ of hyperedges, we know that a_i is the minimal occurrence c of $O_{A\to B}$ such that $\mathcal{L}_{A\to B}(a_p)$ is a prefix of c. Hence, if M_i (resp. M_p) is the move in s_p such that $erase(M_i) = a_i[y_i]$ (resp. $erase(M_p) = a_p[y_p]$) and if $D = \frac{M_p}{\mathcal{L}_{A\to B}(a_p)}$, then the arena D is played by $paux_{A\to B}(a_p)$ at the level of M_i. So, in the construction of s_p, D has been played at step i.

‡‡ In the case where there exists $c \in O_D$ such that $\vdash c$ and $\mathcal{L}_D(c) \neq \dagger$, m_2 is precisely built in such a way that we cannot have $r_D = r_j$ for any j.

We also need to build a play $u_p \in \tilde{\tau}$, which extends S_2^p in an appropriate way. The procedure is similar.

Curry-isomorphism

We are now going to prove that the bijection f satisfies each requirement of a Curry-isomorphism.

We first prove that $\mathcal{D}_B \circ f = \mathcal{D}_A$: suppose $\mathcal{D}_A(a_p) = X_i$, then $s_p = s_{p-1}MM'$ with $erase(M) = a_p[i]$ and $erase(M') = f(a_p)[i]$; likewise, $u_p = u_{p-1}NN'$ with $erase(N) = f(a_p)[i]$ and $erase(M') = a_p[i]$. If $paux_{A\to B}(f(a_p)) = \mathbf{O}$ then one should have $i = r_j$ for some j by construction of s_p, which is impossible. If $paux_{A\to B}(f(a_p)) = \mathbf{P}$ then $paux_{B\to A}(f(a_p)) = \mathbf{O}$ and one should have $i = r_j$ for some j by construction of u_p, which is impossible. Then $paux_{A\to B}(f(a_p))$ is not defined, and $\sharp(f(a_p)) = i$ which means $\mathcal{D}_B(f(a_p)) = X_i$. Similarly, $\mathcal{D}_B(f(a_p)) = X_i$ implies $\mathcal{D}_A(a_p) = X_i$ as well.

We then prove that $f(\mathcal{S}_A) = \mathcal{S}_B$: if $a_p \in S$ with $(t, S) \in \mathcal{R}_A$ for some t, suppose $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$. If $paux_{A\to B}(a_p) = \mathbf{O}$ then $s_p = s_{p-1}MM'$ with $erase(M) = a_p[y_p]$ and $erase(M') = f(a_p)[y_p]$, and one should have $y_p = r_j$ for some j by construction of s_p. But this is impossible since $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$ implies $\mathcal{A}(M') \in \mathcal{O}_{A\to B}$, so $y_p \in \mathbb{N}$. If $paux_{A\to B}(a_p) = \mathbf{P}$ then $paux_{B\to A}(a_p) = \mathbf{O}$, $u_p = u_{p-1}NN'$ with $erase(N) = f(a_p)[y_p]$ and $erase(N') = a_p[y_p]$ and one should have $y_p = r_j$ for some j by construction of u_p. But this is impossible since $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$ implies $\mathcal{A}(N) \in \mathcal{O}_{A\to B}$, so $y_p \in \mathbb{N}$.

Finally, we need to prove the following: for every $(t, S) \in \mathcal{R}_A$, if there exists $c \in S$ such that $\lambda(c) \neq \lambda(t)$, then $(f(t), f(S)) \in \mathcal{R}_B$ (the reciprocal would be done similarly). Let us take $a_1, \ldots, a_p$ the sequence of nodes such that: $\vdash a_1$, $a_i \vdash a_{i+1}$ and $a_p = c$. We necessarily have $t = a_i$ for some $i \leq p$.

First we prove that $ref_B(f(a_p)) = f(a_i)$: suppose that it is false, then $ref_B(f(a_p)) = f(a_j)$ with $j \neq i$. First take $j < i$: if $paux_{A\to B}(a_p) = \mathbf{O}$, then $f(a_p)$ is an $\mathbf{O}$-move on $A \to B$, so $s_p = SM_pM'_p$ where: $M_p = m_1[m_2]$, $\mathcal{A}(m_1) = f(a_p)$ and $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))} = D$ for some D chosen at step j; and $M'_p = m'_1[m'_2]$, $\mathcal{A}(m'_1) = a_p$ and $\frac{m'_1}{\mathcal{L}_{A\to B}(a_p)} = C_{k'}$ for some $k' \geq n_{i-1}$. So we should have $y_p = r_k$ to be the move we choose in D, which is impossible by construction of n_{i-1}. If $paux_{A\to B}(a_p) = \mathbf{P}$, we simply note that $paux_{B\to A}(a_p) = \mathbf{O}$ and do the same reasoning with u_p in $B \to A$. In the case where $i < j$, the reasoning is similar: if $paux_{A\to B}(a_p) = \mathbf{P}$, then $s_p = SM_pM'_p$ where: $M_p = m_1[m_2]$, $\mathcal{A}(m_1) = a_p$ and $\frac{m_1}{\mathcal{L}_{A\to B}(a_p)} = D$ for some D chosen at step i; and $M'_p = m'_1[m'_2]$, $\mathcal{A}(m'_1) = f(a_p)$ and $\frac{m'_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_{k'}$ for some $k' \geq n_{j-1}$. This leads to a contradiction. If $paux_{A\to B}(f(a_p)) = \mathbf{P}$, we work on $B \to A$.

Let us now have $b \in fr_A(a_p)$, and suppose that $f(b) \notin fr_B f(a_p)$. By what has been proved before we know that $paux_{A\to B}(f(b))$ is defined, but also that $ref_B(f(b))$ has the same polarity as $ref_A(b)$: indeed, if $paux_A(b) \neq \lambda(b)$ then $ref_B(f(b)) = f(ref_A(b))$, so $\lambda(ref_B(f(b))) = \lambda(f(ref_A(b))) = \lambda(ref_A(b))$; similarly, if $paux_B(f(b)) \neq \lambda(f(b))$ then $ref_A(b) = f^{-1}(ref_B(f(b)))$, so $\lambda(ref_A(b)) = \lambda(f^{-1}(ref_B(f(b)))) = \lambda(ref_B(f(b)))$. Finally, if $paux_A(b) = \lambda(b)$

and $paux_B(f(b)) = \lambda(f(b))$ then $paux_A(b) = paux_B(f(b))$ because b and $f(b)$ have the same polarity. Then, in all cases, $paux(b) = paux(f(b))$.

We consider that $paux_{A\to B}(f(b)) = \mathbf{O}$ (if not, one works with u_p on $B \to A$), so $paux_{A\to B}(a_p) = \mathbf{P}$ and $s_p = s_{p-1}m_1[m_2]m'_1[m'_2]$ with $\frac{m'_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_k$ for some k. Let $D = \frac{m_1}{\mathcal{L}_{A\to B}(a_p)}$, we necessarily have that $y_p = r_k = erase(r_D)$. But a problem arises with b and $f(b)$: as a first case, suppose that b has the polarity $\mathbf{P}$ in A. Then there is a play $s'_q = s'_{q-1}M_1[M_2]M'_1[M'_2]$ in $\tilde{\sigma}$ constructed the same way as s_p, such that $erase(s'_q) = erase(s'_{q-1})b[y'_q]f(b)[y'_q]$, and where $\frac{M_1}{\mathcal{L}_{A\to B}(b)} = D$ and $\frac{M'_1}{\mathcal{L}_{A\to B}(f(b))} = C_{k'}$ with $k' \neq k$. Then we should have $y'_q = erase(r_D)$ occurrence of $C_{k'}$, so $r_k = r_{k'}$ which is impossible.

The second case is where b has the polarity $\mathbf{O}$ in A. Then there is a play $s'_q = M_1[M_2]M'_1[M'_2]$ in $\tilde{\sigma}$ constructed the same way as s_p, such that $erase(s'_q) = s' f(b)[y'_q]b[y'_q]$, and where $\frac{M_1}{\mathcal{L}_{A\to B}(f(b))} = C_{k'}$ with $k' \neq k$ and $\frac{M'_1}{\mathcal{L}_{A\to B}(b)} = D$. Then we should have $y'_q = r_{k'} = erase(d')$ with d' move in D. But in this case $\mathcal{A}(d), \mathcal{A}(d') \in \mathcal{O}_D$ (if not we have a token l in d or d'), so $\mathcal{E}(d) = r_k$ and $\mathcal{E}(d') = r_{k'}$, hence $k = k'$ because D is unambiguous. This is impossible.

$f(b) \in fr_B(f(a_p))$ similarly implies $b \in fr_A(a_p)$, so $f(S) = \{b \mid s \in fr_B(f(a_p))\}$. This allows us to conclude that $(f(t), f(S)) \in \mathcal{R}_B$. □