Relative Definability of Boolean Functions via Hypergraphs

Antonio Bucciarelli

Dipartimento di Scienze dell'Informazione, Universitá di Roma "La Sapienza", via Salaria 113, 00195 Roma, Italy
and
LIENS-DMI, Ecole Normale Supérieure, 45 rue d'Ulm 75005 Paris, France

Pasquale Malacaria

Department of Computing, Imperial College of Science Technology and Medicine, 180 Queen's Gate London SW7 2AZ, UK

Abstract

The aim of this work is to show how hypergraphs can be used as a *sistematic* tool in the classification of continous boolean functions according to their *degree of parallelism*. Intuitively f is "less parallel" than g if it can be defined by a sequential program using g as its only free variable. It turns out that the poset induced by this preorder is (as for the degrees of recursion) a sup-semilattice.

Although hypergraphs had already been used in [6] as a tool for studying degrees of parallelism, no general results relating the former to the latter have been proved in that work. We show that the sup-semilattice of degrees has a categorical counterpart: we define a category of hypergraphs such that every object "represents" a monotone boolean function; finite coproducts in this category correspond to lubs of degrees. Unlike degrees of recursion, where every set has a recursive upper bound, monotone boolean functions may have no sequential upper bound. However the ones which do have a sequential upper bound can be nicely characterised in terms of hypergraphs. These subsequential functions play a major role in the proof of our main result, namely that f is less parallel than g if there exists a morphism between their associated hypergraps.

1 Introduction

In this paper we will consider first-order continous functions of type $\mathcal{B}^n \to \mathcal{B}$ where $\mathcal{B}$ is the flat domain of boolean values $\{\perp, \texttt{tt}, \texttt{ff}\}$. Tuples of boolean

Preprint submitted to Elsevier Science 21 August 1996

values are ordered componentwise. Note that continous functions of this type are just monotone functions.

Given two continuous functions f and g, we say that f is *less parallel than* g ($f \leq_{\mathtt{par}} g$) if there exists a closed PCF-term M such that $[\![M]\!]g = f$ (where $[\![M]\!]$ denotes the interpretation of M in the standard Scott model [15]).

A *degree of parallelism* is a class of the equivalence relation associated with the preorder $\leq_{\mathtt{par}}$. Two functions in the same class will be called *equiparallel.* The degree of a given continous function f will be denoted by $[f]$.

We will use sometimes the expression f is g*−definable* for $f \leq_{\mathtt{par}} g$.

One more intuitive yet equivalent definition for degrees is obtained by replacing "PCF" by any imperative typed sequential language and "term" by "program".

In order to study $\leq_{\mathtt{par}}$ we introduce a category of hypergraphs. Continuous function will be projected on the objects of this category, and hypergraph morphisms will be witnesses of $\leq_{\mathtt{par}}$ relations.

An informal way of gradually describing the passage from function to hypergraph is the following:

Any function is a set of pairs (argument,value): its graph.

Monotone functions on finite posets can be represented by a set of pairs (minimal argument, value): their trace (for a formal definition of trace see the next section).

In the hypergraph representations the arity of the function and the actual content of minimal arguments are forgotten. The vertexes of the hypergraph stand for minimal arguments, and the edges encode a partial information on the actual content of such minimal arguments. The values of the encoded function are recorded by coloring the vertexes.

Consider for instance the n-ary logical connective that outputs `tt` if all its arguments are `tt` and is undefined otherwise. Then the hypergraph associated to any such function is the same for all n, namely the hypergraph with a unique vertex and no arcs. Indeed any hypergraph represents infinitely many functions whereas traces are in a one-to-one correspondence with (monotone) functions.

A natural question is hence how faithful the hypergraph representation is. This question is indeed twofold, namely:

– Which properties of functions are characterised in terms of hypergraphs?

– Is it the case that two functions having the same hypergraph are equiparallel?

Concerning the first questions the results in this paper are summarised in the following table (rows stand for type of the function, column for hypergraph properties characterising that type of function) [1]:

	Hypergraphs			
Functions	*functional*	*no hyperarcs*	*no binary hyperarcs*	*monochromatic*
continous	Yes	No	No	No
stable	Yes	No	Yes	No
sequential	Yes	Yes	Yes	No
subsequential	Yes	No	No	Yes

Concerning the second questions let us consider an example which gives some evidence of the fact that the question itself is non-trivial:

Example 1: Let us consider, for $n \in \omega$, $n \geq 1$ the monotone functions $f_n, g_n : \mathcal{B}^n \to \mathcal{B}$ defined by the following traces:

$$\mathtt{tr}(f_n) = \{(v,\ \mathtt{tt}\ ), (\sigma^1(v),\ \mathtt{tt}\ ), \ldots, (\sigma^{n-1}(v),\ \mathtt{tt}\ )\}$$

$$\mathtt{tr}(g_n) = \{(w,\ \mathtt{tt}\ ), (\sigma^1(w),\ \mathtt{tt}\ ), \ldots, (\sigma^{n-1}(w),\ \mathtt{tt}\ )\}$$

where $v = (\ \mathtt{tt}\ , \underbrace{\bot, \ldots, \bot}_{n-1})$, $w = (\bot, \underbrace{\ \mathtt{tt}\ , \ldots,\ \mathtt{tt}\ }_{n-1})$, and $\sigma((b_1, \ldots, b_{n-1}, b_n)) = (b_n, b_1, \ldots, b_{n-1})$.

i.e. f_n is the function that outputs $\mathtt{tt}$ if it has at least one $\mathtt{tt}$ in its n arguments whereas g_n outputs $\mathtt{tt}$ if it has at least $n-1$ $\mathtt{tt}$ among its arguments.

For a given n the maps f_n and g_n are represented by the same hypergraph, namely the complete hypergraph of order n (that is the hypergraph in which all but singletons subsets of vertices are hyperarcs). Hence there is a trivial morphism (namely the identity) between the hypergraphs of f_n and g_n. How-

[1] Stable and sequential functions are introduced in section 2. Functional hypergraphs in section 3; monochromatic hypergraphs and subsequential function in section 4.

ever the PCF term M_n defining f_n in terms of g_n has at least $n-1$ "recursive" calls of g_n.

For example for $n = 3$ we have

$$f_3 = \lambda xyz \; g_3(x \; g_3(\; \mathtt{tt} \; y \; z) \;\; \mathtt{tt}\;)$$

and for $n = 4$

$$f_4 = \lambda xyzw \; g_4(x \; g_4(y \; g_4(\; \mathtt{tt} \;\; \mathtt{tt} \; z \; w) \;\; \mathtt{tt} \;\; \mathtt{tt}\;) \;\; \mathtt{tt} \;\; \mathtt{tt}\;)$$

■

The moral is that if we could prove that hypergraphs isomorphisms *reflect* equivalence of degrees (i.e. that functions whose hypergraphs are isomorphic are equiparallel) then we would have a simple and effective tool for the study of degrees. We will indeed prove such a result as a corollary of our main result: hypergraphs morphisms reflect $\leq_{\mathtt{par}}$ relations.

1.1 Related works

The study of degrees of parallelism was pioneered by Sazonov and Tracktembrot [14,18] who singled out some finite subposet of degrees. Some results on degrees are corollary of well known facts: for instance Plotkin full abstraction result for PCF+`por` implies that this poset has a top. The bottom of degrees is the set of PCF-definable functions which is fully characterized by the notion of sequentiality (in any of its formulations). Moreover Sieber's *sequentiality relations* [16] provide a characterization of first order degrees of parallelism and this characterization is effective: given f and g one can decide if $f \leq_{\mathtt{par}} g$, and recently A. Stoughton [17] has implemented an algorithm which solves this decision problem.

Hypergraphs for the study of degrees were first introduced in [6] where an infinite subposet of degrees was pointed out. However no precise connection between hypergraphs and monotone functions was established there. The definition of functional hypergraphs bears striking resemblance to Ehrhard's definition of *parallel* hypercoherence [8] and indeed we owe him the condition [H2'] in section 3.

2 The upper semi-lattice of degrees

Given a monotone function $f : \mathcal{B}^n \to \mathcal{B}$, the *trace* of f is defined by

$$\mathtt{tr}(f) = \{(v, b) \mid v \in \mathcal{B}^n,\ b \in \mathcal{B},\ b \neq \perp,\ f(v) = b \text{ and } \forall v' < v\ f(v') = \perp\}.$$

Traces are in one-to-one correspondence with monotone functions. In particular, for any $v \in \mathcal{B}^n$

$$f(v) = \bigvee \{b \in \mathcal{B} \mid \exists\ (w, b) \in \mathtt{tr}(f)\ w \leq v\}$$

Remark that, since $\mathcal{B}$ is a flat cpo, the set $\{b \in \mathcal{B} \mid \exists\ (w, b) \in \mathtt{tr}(f)\ w \leq v\}$, is either empty or a singleton, for any $v \in \mathcal{B}^n$.

In order to introduce the first remark on degrees we recall the parallel or function `por` defined by

$$\mathtt{por}(x, y) = \begin{cases} \mathtt{tt} & \text{if } x = \mathtt{tt} \text{ or } y = \mathtt{tt} \\ \mathtt{ff} & \text{if } x = \mathtt{ff} \text{ and } y = \mathtt{ff} \\ \perp & \text{otherwise.} \end{cases}$$

Fact 1 *The poset of degrees of parallelism is a sup semilattice with a bottom element (the set of PCF-definable functions) and a top element (the equivalence class of parallel or).*

Proof: The set of PCF-definable functions is the $\perp$ of degrees by definition, whereas the fact that $[\mathtt{por}]$ is the $\top$ of degrees, is a corollary of Plotkin's definability result [13].

Given $f : \mathcal{B}^n \to \mathcal{B}$ and $g : \mathcal{B}^m \to \mathcal{B}$, we define $h : \mathcal{B}^k \to B$ such that $[h] = [f] \vee [g]$. Without loss of generality, let us suppose that there exists $l \geq 0$ such that $m = n - l$. Then we set $k = n + 1$, and

$$\mathtt{tr}(h) = \{((\ \mathtt{tt}\ , x_1, \dots, x_n), b) \mid ((x_1, \dots, x_n), b) \in \mathtt{tr}(f)\} \bigcup$$
$$\{((\ \mathtt{ff}\ \underbrace{\perp, \dots, \perp}_{l}, x_1, \dots, x_m), b) \mid ((x_1, \dots, x_m), b) \in \mathtt{tr}(g)\}.$$

In order to prove that $[h] = [f] \vee [g]$ we have first to show that $f \leq_{\mathtt{par}} h$ and $g \leq_{\mathtt{par}} h$. It is easy to check that

$$[\![\lambda d \lambda x_1 \dots x_n.\ d\ \ \mathtt{tt}\ \ x_1 \dots x_n]\!] h = f$$

and

$$[\![\lambda d \lambda x_1 \ldots x_m.\ d\ \texttt{ff}\ \underbrace{\bot \ldots \bot}_{l}\ x_1 \ldots x_m]\!] h = g.$$

Moreover, let $h' : \mathcal{B}^l \to \mathcal{B}$ be such $f, g \leq_{\texttt{par}} h'$, i.e. such that there exist M, N: $[\![M]\!]h' = f$ and $[\![N]\!]h' = g$. Then it is again easy to check that

$$[\![\lambda g \lambda x_1 \ldots x_k \texttt{if}\ x_1\ \texttt{then}\ M\ g\ x_2 \ldots x_k\ \texttt{else}\ N\ g\ x_{l+2} \ldots x_k]\!] h' = h$$

Hence $[h] = [f] \vee [g]$. ■

Given f, g as above the function h given in the proof of the proposition will be denoted by $f + g$.

The set of monotone functions which can be computed by sequential, purely functional programs is the $\bot$ of the hierarchy of degrees, and it has been the object of a considerable amount of research works. We end this section with a short overview of some of these works, pointing out some notions and results used in the rest of the paper.

The Full Abstraction problem for PCF led to the definition of classes of functions which are more constrained than the continous ones; in particular, as we will see, stable [3] and strongly stable [5] functions have a nice characterisation in term of hypergraphs.

A continuous function $f : \mathcal{B}^n \to \mathcal{B}$ is *stable* if for all $v_1, v_2 \in \mathcal{B}^n$, if v_1 and v_2 are bounded then $f(v_1 \wedge v_2) = f(v_1) \wedge f(v_2)$ (or equivalently if for all $v_1, v_2 \in \pi_1(tr(f))$, v_1 and v_2 are unbounded.)

A subset $A = \{v_1, \ldots, v_k\}$ of $\mathcal{B}^n$ is *linearly coherent* (or simply *coherent*) if

$$\forall j\ 1 \leq j \leq n\ ((\forall l\ 1 \leq l \leq k\ v_l^j \neq \bot) \ \Rightarrow\ (\forall l_1, l_2\ 1 \leq l_1 \leq l_2 \leq k\ v_{l_1}^j = v_{l_2}^j)).$$

The set $\{v_1^j, \ldots, v_k^j\}$ is the $j-$th *component* of A.

Example 2: Consider the sets $A, B \subseteq \mathcal{B}^3$ defined by

$$A = \{(\texttt{ tt }, \texttt{ tt }, \bot), (\texttt{ tt }, \texttt{ ff }, \bot), (\texttt{ ff }, \bot, \texttt{ tt }), (\texttt{ ff }, \bot, \texttt{ ff })\}$$

$$B = \{(\bot, \texttt{ tt }, \texttt{ ff }), (\texttt{ ff }, \bot, \texttt{ tt }), (\texttt{ tt }, \texttt{ ff }, \bot)\}$$

A is not coherent, since its first component does not contain $\bot$ nor it is a singleton. B is coherent since all its components do contain $\bot$. A is the set of minimal points of the if-then-else function, which is PCF-definable; B is the set of minimal points of the so called Berry function, which is stable but not PCF-definable.

$\blacksquare$

The set of coherent subsets of $\mathcal{B}^n$ is denoted $\mathcal{C}(\mathcal{B}^n)$.

Coherent sets play an important role in our description of monotone functions via hypergraphs: the vertexes of the hypergraph associated to a function f stand for the minimal points of f (i.e. the elements of the first projection of the trace of f), and a set $\{v_1, \ldots, v_k\}$ of vertexes is an arc if and only if the set of the corresponding minimal points of f is coherent. We will often use the following simple properties of traces and coherence:

Fact 2 – *If $A \in \mathcal{C}(\mathcal{B}^n)$ and B is an Egli-Milner lower bound of A (that is if $\forall x \in A \exists y \in B\ y \leq x$ and $\forall y \in B \exists x \in A\ y \leq x$) then $B \in \mathcal{C}(\mathcal{B}^n)$.*
– *If $f : \mathcal{B}^n \to \mathcal{B}$ is a monotone function, $A \subseteq \mathcal{B}^n$, and $f(A) \subseteq \mathcal{B} - \{\perp\}$, then there exists an Egli-Milner lower bound B of A such that $B \subseteq \pi_1(\mathtt{tr}(f))$ and $f(B) = f(A)$.*

A proof of the first item can be found in [5]; the second one is an immediate consequence of the definition of trace.

Definition 1 *A continuous function $f : B^n \to B^m$ is* linearly strongly stable *(or simply* strongly stable*) if for any $A \in \mathcal{C}(\mathcal{B}^n)$*

– $f(A) \in \mathcal{C}(\mathcal{B}^m)$.
– $f(\bigwedge A) = \bigwedge f(A)$.

Example 3: Let us see how strong stability rules out the Berry's function $g : \mathcal{B}^3 \to \mathcal{B}$ defined by

$$\mathtt{tr}(g) = \{((\perp, \mathtt{tt}, \mathtt{ff}), \mathtt{tt}), ((\mathtt{ff}, \perp, \mathtt{tt}), \mathtt{tt}), ((\mathtt{tt}, \mathtt{ff}, \perp), \mathtt{tt})\}$$

As we have seen in example 2 the set B of minimal points of g is coherent, but $\bigwedge g(B) = \mathtt{tt} \neq g(\bigwedge(B)) = \perp$. Hence g is not strongly stable.

$\blacksquare$

Even if the model of strongly stable functions is not fully abstract for PCF, i.e. there exist strongly stable functionals which are not PCF-definable, see [5], strong stability does capture the notion of sequentiality, or PCF-definability, at first order. In the following proposition "sequential" stands for "Kahn-Plotkin sequential" [10], "Milner sequential" [11] or "Vuillemin sequential" [19], since all these notions coincide for first order functions.

Proposition 1 *Let $f : \mathcal{B}^n \to \mathcal{B}$ be a monotone function. The following are equivalent:*

- *f is strongly stable.*
- *f is PCF-definable.*
- *f is sequential.*

A proof can be found in [6] and in [2]. The original proof of "sequential $\Leftrightarrow$ PCF-definable" is in [4].

Actually there exist several alternative characterization of the notion of PCF-definability for first order functions, for instance Sieber's logically sequential functions [16] and Colson-Ehrhard's hereditarily sequential ones [7]. Of course any fully abstract model of PCF [1], [9],[12] provide *a fortiori* a characterization of PCF-definability for monotone, first order functions.

3 Hypergraphs and monotone functions

Definition 2 *A* colored hypergraph *$H = (V_H, A_H, C_H)$ is given by a set V_H of* vertices*, a set $A_H \subseteq \{A \subseteq V_H | \#A \geq 2\}$ of* (hyper)arcs *and a* coloring function *$C_H : V_H \to \{\texttt{black}, \texttt{white}\}$.*

As a first approximation a map between two hypergraps is a set theoretical map from vertices to vertices which preserves hyperarcs; concerning colours, several notions are possible: One extreme is to ask for the preservation of colours; on the other hand a more liberal requirement is to say that the images of "adjacent" vertices of different colours have different colours (think of "adjacent" as "being in the same hyperarc").

Formally we consider two notion of morphisms on hypergraphs:

A *weak* morphism from a hypergraph H to a hypergraph H' is a function $m : V_H \to V_{H'}$ such that:

- For all $A \subseteq V_H$, if $A \in A_H$ then $m(A) \in A_{H'}$.
- for all $x, x' \in V_H$, if $x, x' \in X \in A_H$ and $C_H(x) \neq C_H(x')$ then $C_{H'}(m(x)) \neq C_{H'}(m(x'))$.

A *strong* morphism is more restrictive on colours: we require that for all $x \in V_H$, $C_H(x) = C_{H'}(m(x))$.

A *sub-hypergraph* H' of a hypergraph H has as set of vertices $V_{H'}$ a subset of V_H and as hyperarcs those of H whose vertices belong to H'. Colours are given by restriction.

Note that set theoretical inclusions are both weak and strong morphisms with this notion of sub-hypergraph.

We will restrict our attention on a particular class of hypergraphs which turns out to be in a very precise relationship with monotone functions.

A *functional hypergraph* is an hypergraph H such that:

H1 : If $\{x, y\} \in A_H$ then $C_H(x) = C_H(y)$.
H2 : If $X \subseteq V_H$ is not a hyperarc there exists a partition X_1, X_2 of X such that for all $Y \subseteq X$ if $Y \bigcap X_1 \neq \emptyset, Y \bigcap X_2 \neq \emptyset$ then Y is not a hyperarc.

Condition $[H2]$ can be equivalently and more synthetically expressed as follows:

H2' : If X_1, X_2 are hyperarcs and $X_1 \cap X_2 \neq \emptyset$ then $X_1 \cup X_2$ is an hyperarc.

It is trivial to check that a sub-hypergraph of a functional hypergraph is functional.

We are now ready to define our categories of interest: $\mathcal{SH}, \mathcal{WH}$

object $\mathcal{SH}$ = **object** $\mathcal{WH}$ = Functional Hypergraphs.

arrows $\mathcal{SH}$ = Strong Morphisms.

arrows $\mathcal{WH}$ = Weak Morphisms.

(it is trivial indeed to check that in both cases we have a category).

Definition 3 *Let* $f : \mathcal{B}^n \to \mathcal{B}$ *be the* n*-ary function defined by* $\mathtt{tr}(f) = \{(v_1, b_1), \ldots, (v_k, b_k)\}$. *The hypergraph* H_f *is defined by*

- $V_{H_f} = \{1, 2, \ldots, k\}$.
- $A_{H_f} = \{\{i_1, i_2, \ldots, i_l\} \subseteq V_{H_f} \mid l \geq 2 \; and \{v_{i_1}, v_{i_2}, \ldots, v_{i_l}\} \in \mathcal{C}(\mathcal{B}^n)\}$.
- $C_{H_f}(i) =$ `if` b_i `then` *white* `else` *black*.

Example 4: Consider the Berry function $g : \mathcal{B}^3 \to \mathcal{B}$ defined in example 3 and the parallel-or function *por* : $\mathcal{B}^2 \to \mathcal{B}$ defined in section 2, whose traces are respectively

$$tr(g) = \{((\bot, \texttt{ tt }, \texttt{ ff }), \texttt{ tt }), ((\texttt{ ff }, \bot, \texttt{ tt }), \texttt{ tt }), ((\texttt{ tt }, \texttt{ ff }, \bot), \texttt{ tt })\}$$

$$tr(por) = \{((\bot, \texttt{ tt }), \texttt{ tt }), ((\texttt{ tt } \ \bot), \texttt{ tt }), ((\texttt{ ff }, \texttt{ ff }), \texttt{ ff })\}$$

We have:

$$H_g = (\{1,2,3\}, \{\{1,2,3\}\}, C_{H_g}(1) = C_{H_g}(2) = C_{H_g}(3) = white)$$

$$H_{por} = (\{1,2,3\}, \{\{1,2\}, \{1,2,3\}\},$$
$$C_{H_{por}}(1) = C_{H_{por}}(2) = white, C_{H_{por}}(3) = black)$$

The map $\alpha : H_g \to H_{por}$ defined by $\alpha(1) = \alpha(2) = 1, \ \alpha(3) = 2$ is a (strong) morphism.

■

Proviso 1: The vertexes of H_f are in one-to-one correspondence with $\pi_1(\mathtt{tr}(f))$. We could have turned this correspondence into an identity, by stipulating that $V_{H_f} = \pi_1(\mathtt{tr}(f))$. However, since we will prove that whenever H_f and H_g are (weakly or strongly) isomorphic, f and g are equiparallel, and since hypergraph isomorphisms are clearly independent from vertexes' names, we do prefere to keep this identity implicit. Nevertheless in several proofs of the following sections, given H_f we will need to explicitly refer to minimal points of f (i.e. to elements of $\pi_1(\mathtt{tr}(f))$). Formally, given a functional hypergraph H, there exists a family of function $\{h_f\}_{f \in \{g \ | \ H_g = H\}} : V_H \to \bigcup_{n \in \omega} \mathcal{B}^n$ such that $h_f(V_H) = \pi_1(\mathtt{tr}(f))$.

For the sake of simplicity we will omit h_f whenever possible, and in particular we will fell free of considering the vertexes of H_f as if they were labeled by $\pi_1(\mathtt{tr}(f))$. ■

We can observe that for any monotone function $f : \mathcal{B}^n \to \mathcal{B}$, the hypergraph H_f is functional: the requirement H1 is satisfied by H_f since if two minimal points v_1, v_2 of f are coherent, then they are bounded (note that this is true only for binary sets), hence $f(v_1) = f(v_2)$. H2 is verified as well, since if a set $A = \{v_1, \ldots, v_k\}, \ k \geq 2$ of minimal points of f is not coherent, then there exists $1 \leq j \leq n$ such that the j-th component $\{v_1^j, \ldots, v_k^j\}$ of A is $\{\texttt{ tt }, \texttt{ ff }\}$. Hence the partition of $\{1, \ldots, k\}$ given by $\{\ \{i \mid v_i^j = \texttt{ tt }\}, \{i \mid v_i^j = \texttt{ ff }\}\}$ satisfies H2. Actually the converse does hold, too:

Proposition 2 *Given an hypergraph H there exists a monotone function $f : \mathcal{B}^n \to \mathcal{B}$ such that $H_f = H$ if and only if H is a functional hypergraph.*

Proof: The function F_H associated to a functional hypergraph $H = (V_H, A_H, C_H)$ is defined as follows: $F_H : \mathcal{B}^n \to \mathcal{B}$ where $n = \#V_H + \#\overline{A_H}$ with

$$\overline{A_H} = \{B \subseteq V_H \mid \#B \geq 2 \text{ and } B \notin A_H\}.$$

The trace of F_H has $m = \#V_H$ elements. We fix enumerations $v_1, \ldots v_m$ for the set V_H and $B_i, \ldots B_l$ for the set $\overline{A_H}$. For all $B_i \in \overline{A_H}$ let (B_i^1, B_i^2) the partition of B given by the definition of functional hypergraph (condition H2).

The $i-$th element of $\mathtt{tr}(F_H)$ is then defined as follows:

$$((\underbrace{\perp, \ldots, \perp}_{i-1}, \ \mathtt{tt}\ , \underbrace{\perp, \ldots, \perp}_{m-i}, b_i^1, \ldots, b_i^l), c_i)$$

where

$$b_i^j = \begin{cases} \mathtt{tt} & \text{if } v_i \in B_j^1 \\ \mathtt{ff} & \text{if } v_i \in B_j^2 \\ \perp & \text{otherwise} \end{cases}$$

and

$$c_i = \begin{cases} \mathtt{tt} & \text{if } C_H(v_i) = \mathtt{white} \\ \mathtt{ff} & \text{if } C_H(v_i) = \mathtt{black}. \end{cases}$$

We leave to the reader to check that F_H is a monotone function whose hypergraph is H. ∎

It is easy to see that the function F_{H_f} bears in general no resemblance with f for example if $f = \mathtt{por} : \mathcal{B}^2 \to \mathcal{B}$ then $F_{H_f} : \mathcal{B}^5 \to \mathcal{B}$.

We end this section with a nice property of the categories $\mathcal{SH}, \mathcal{WH}$.

Proposition 3 *$\mathcal{SH}, \mathcal{WH}$ have coproducts.*

Proof: Let us define the binary coproducts: Given H, H' let H'' be the hypergraph given by the disjoint union of vertices of H, H', the disjoint union of hyperarcs of H, H' and the disjoint union of the colouring maps of H, H'. Then H'' is a functional hypergraph (condition $H1$ is trivial and condition $H2$ is trivially checked as well by using $H2'$).

The inclusion maps h (resp h') from H (resp H') to H'' provide the injections. Finally is easy to see that any pair of maps f, f' from H (resp H') to H''' factorise trough H''. ∎

Note that categorical coproduct and l.u.b. of degrees are related in the following sense:

Fact 3 *The coproduct $H_f \oplus H_g$ (in both category $\mathcal{SH}, \mathcal{WH}$) is the hypergraph of $f + g$.*

Proof: By definition the trace of $f+g$ has $l+r$ elements with l (resp r) being the number of element in the trace of f (resp g); this means that H_{f+g} has as vertices the disjoint union of vertices of H_f, H_g. By the definition of trace of $f + g$ is also clear that the colouring map of H_{f+g} is the disjoint union of the maps in H_f, H_g.

The only thing we are left to check is hence the hyperarcs. Again by definition of trace of $f+g$ and by definition of coherence is easy to check that a coherent subset of trace of f (resp of trace of g) is a coherent subset $\mathtt{tr}(f + g)$. For the opposite direction note that by the definition of coherence a coherent subset of $\mathtt{tr}(f + g)$ cannot contain elements from both $\mathtt{tr}(f)$ and $\mathtt{tr}(g)$ (again by definition of $\mathtt{tr}(f + g)$ because of the first argument). This implies that the hyperarcs of H_{f+g} are indeed the disjoint union of the hyperarcs of H_f and H_g. ∎

3.1 Relating hypergraphs and degrees

First we can observe how clearly hypergraphs classify PCF-definable and stable functions versus general monotone functions.

Fact 4 *Let $f : \mathcal{B}^n \to \mathcal{B}$ be a continuous function: f is stable if and only if H_f has no binary hyperarc. It is strongly stable if and only H_f has no hyperarc.*

Proof: Let us prove the statemement concerning strongly stable functions: given $f : \mathcal{B}^n \to \mathcal{B}$, if H_f has a hyperarc $A = \{v_1, \ldots, v_k\}$ (see proviso 1), then by definition $\{v_1, \ldots, v_k\} \in \mathcal{C}(\mathcal{B}^n)$. Now either all the vertexes of A have the same colour in H_f, and hence $f(\bigwedge A) < \bigwedge f(A)$, or they have not, hence $f(A) \notin \mathcal{C}(\mathcal{B})$. In both cases f is not strongly stable.

Conversely if H_f has no hyperarc, let $A \in \mathcal{C}(\mathcal{B}^n)$ be such that $\perp \notin f(A)$ (otherwise $f(A) \in \mathcal{C}(\mathcal{B})$ and $f(\bigwedge A) = \bigwedge f(A)$ holds trivially). By fact 2, there exists an Egli-Milner lower bound B of A such that $B \subseteq \pi_1(\mathtt{tr}(f))$ and $f(A) = f(B)$. Since B is coherent and H_f has no hyperarc, $\#B = 1$, hence $f(A) \in \mathcal{C}(\mathcal{B})$ and $f(\bigwedge A) = \bigwedge f(A)$.

The proof of the statement concerning stable functions is a particular case of the one above, with $k = 2$. ∎

Hypergraphs have already been used in [6] in order to show that the poset of degrees is highly non-trivial; in particular it contains both infinite (ascending and descending) chains and infinite antichains. Bucciarelli defined a class of hypergraphs as follows.

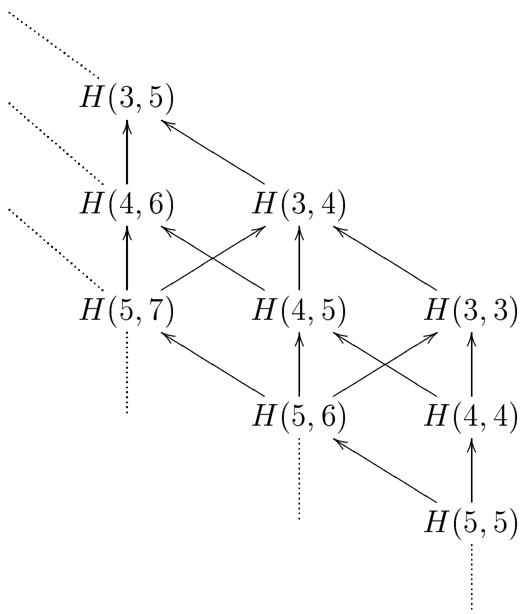
Definition 4 *Given two natural numbers $m \geq n \geq 3$, let $H(n,m)$ be the hypergraph defined by:*

$$H(n,m) = (\{1,2,\ldots,m\}, \{A \subseteq \{1,2,\ldots,m\} \mid \#A \geq n\},\ \textit{for all } i\ \ C(i) = \textit{white})$$

It is easy to check that the $H(n,m)$'s are functional hypergraphs. Let's call $\mathcal{SH}'$ the full subcategory of $\mathcal{SH}$ whose objects are the $H(n,m)$. The main result of [6] is then:

Proposition 4 $\mathcal{SH}'(H_f, H_g) \neq \emptyset$ *iff* $f \leq_{\texttt{par}} g$.

The relations between the degrees of $H(n,m)$'s are summarised by the following picture:



4 Subsequential functions

A monotone function $f : \mathcal{B}^n \to \mathcal{B}$ is *subsequential* if it is extensionally upper bounded by a strongly stable function. As shown in proposition 5 subsequential functions correspond to hypergraphs with monochromatic hyperarcs and to functions preserving linear coherence. Such a class of functions admits hence a natural characterisation in order theoretic, graph theoretic and algebraic terms. Moreover, thanks to their properties subsequential functions will be an important combinatorial tool in our work.

Lemma 1 *Let* $\{B_x\}_{x\in X}$ *(*X *a set of indices) be such that* $\forall x \in X$, $B_x \in \mathcal{C}(\mathcal{B}^n)$ *and* $A = \{\bigwedge B_x | x \in X\} \in \mathcal{C}(\mathcal{B}^n)$. *Then* $\bigcup_{x\in X} B_x \in \mathcal{C}(\mathcal{B}^n)$.

Proof: Suppose that $Y = \bigcup_{x\in X} B_x \notin \mathcal{C}(\mathcal{B}^n)$; then there exists a component $1 \leq j \leq n$ and a non-trivial partition (Y_1, Y_2) of Y such that for all $y_1 \in Y_1, (y_1)^j = \texttt{tt}$ and for all $y_2 \in Y_2, (y_2)^j = \texttt{ff}$.

It is easy to see that $\forall x \in X, B_x \subseteq Y_1$ or $B_x \subseteq Y_2$; hence given any $a \in A$ we get

- $a^j = \texttt{tt}$ if $a = \bigwedge B_x$ and $B_x \subseteq Y_1$.
- $a^j = \texttt{ff}$ if $a = \bigwedge B_x$ and $B_x \subseteq Y_2$.

We hence deduce a non-trivial partition (A_1, A_2) of A such that $a \in A_1$ iff $a^j = \texttt{tt}$ and $a \in A_2$ iff $a^j = \texttt{ff}$. This is a contradiction since $A \in \mathcal{C}(\mathcal{B}^n)$. ■

Proposition 5 *Let* $f : \mathcal{B}^n \to \mathcal{B}$ *be a monotone function. The following are equivalent:*

1. *For all* $A \subseteq \texttt{tr}(f)$, *if* $\pi_1(A) \in \mathcal{C}(\mathcal{B}^n)$ *then* $\pi_2(A) \in \mathcal{C}(\mathcal{B})$ [2]
2. *For all* $A \in \mathcal{C}(\mathcal{B}^n)$, $f(A) \in \mathcal{C}(\mathcal{B})$. *(i.e.* f *preserves the linear coherence of* $\mathcal{B}^n$*.)*
3. f *is subsequential.*
4. *If* $X \in A_{H_f}$ *then for all* $x, y \in X$ $C_{H_f}(x) = C_{H_f}(y)$ *(i.e.* X *is* monochromatic*).*

Proof:

$1 \Rightarrow 2$: Let $A \in \mathcal{C}(\mathcal{B}^n)$ be such that $\perp \notin f(A)$ (otherwise $f(A) \in \mathcal{C}(\mathcal{B})$). By fact 2 there exists $B \subseteq \texttt{tr}(f)$ such that $\pi_1(B)$ is an Egli-Milner lower bound of A, and $\pi_2(B) = f(A)$. Since $\pi_1(B)$ is coherent (fact 2) we are done.

$2 \Rightarrow 3$: We have to define a strongly stable upper bound of f. Let $\overline{f} : \mathcal{B}^n \to \mathcal{B}$ be the function defined as follows:

$$\overline{f}(x) = \bigvee_{A\in\mathcal{C}(\mathcal{B}^n),\ x\geq\bigwedge A} (\bigwedge_{y\in A} f(y)).$$

[2] Remark that since by definition of trace $\perp \notin \pi_2(A)$, $\pi_2(A) \in \mathcal{C}(\mathcal{B})$ if and only if $\pi_2(A)$ is a singleton

First of all we have to show that $\overline{f}$ is a function, i.e. that, given $x \in \mathcal{B}^n$, if $A, B \in \mathcal{C}(\mathcal{B}^n)$ are such that $x \geq \bigwedge A, \bigwedge B$, then $\bigwedge f(A)$ and $\bigwedge f(B)$ are bounded (this is sufficient since $\mathcal{B}$ is clearly a *coherent* bounded complete cpo, i.e. any set of pairwise bounded boolean values is bounded, and hence has a l.u.b.). If A and B are as above, let us suppose, without loss of generality, that $\bigwedge f(A) = \texttt{tt}$ and $\bigwedge f(B) = \texttt{ff}$. Since $C = \{\bigwedge A, \bigwedge B\}$ is Egli-Milner smaller than $\{x\}$, which is coherent, C is coherent (see fact 2), hence by lemma 1 $A \bigcup B \in \mathcal{C}(\mathcal{B}^n)$. Since $f(\bigwedge A) = \texttt{tt}$ and $f(\bigwedge B) = \texttt{ff}$ we conclude that $f(A \bigcup B) = \{ \texttt{tt} , \texttt{ff} \} \notin \mathcal{C}(\mathcal{B})$, hence f does not preserve $\mathcal{C}(\mathcal{B}^n)$. Contradiction.

Moreover $\overline{f}$ is clearly monotone, and it is an upper bound of f since for any $x \in \mathcal{B}^n$, $\{x\} \in \mathcal{C}(\mathcal{B}^n)$.

In order to prove that $\overline{f}$ is strongly stable, given $A \in \mathcal{C}(\mathcal{B}^n)$, let us prove that (1) $\overline{f}(A) \in \mathcal{C}(\mathcal{B})$ and (2) $\overline{f}(\bigwedge A) = \bigwedge \overline{f}(A)$.

(1) If $\perp \in \overline{f}(A)$ then $\overline{f}(A) \in \mathcal{C}(\mathcal{B})$. Let us suppose that $\perp \notin \overline{f}(A)$. In this case, by definition of $\overline{f}$, for any $x \in A$ there exists $B_x \in \mathcal{C}(\mathcal{B}^n)$ such that $\bigwedge B_x \leq x$ and $\bigwedge f(B_x) > \perp$. Since $\{\bigwedge B_x \mid x \in A\}$ is Egli-Milner smaller than A, we conclude as above by fact 2 and lemma 1, that $\bigcup_{x \in A} B_x \in \mathcal{C}(\mathcal{B}^n)$. Hence $f(\bigcup_{x \in A} B_x) \in \mathcal{C}(\mathcal{B})$. Now since for all $x \in A$ $\overline{f}(x) = \bigwedge f(B_x) > \perp$, we have $\overline{f}(A) = f(\{\bigwedge B_x \mid x \in A\}) = f(\bigcup_{x \in A} B_x) \in \mathcal{C}(\mathcal{B})$ and we are done.

(2) Since $\overline{f}$ is monotone, $\overline{f}(\bigwedge A) \leq \bigwedge \overline{f}(A)$. Let $\bigwedge \overline{f}(A) = b > \perp$, and for any $x \in A$ let B_x be as above, that is $B_x \in \mathcal{C}(\mathcal{B}^n)$, $\bigwedge B_x \leq x$ and $\bigwedge f(B_x) = b > \perp$. Again we have that $D = \bigcup_{x \in A} B_x \in \mathcal{C}(\mathcal{B}^n)$. Moreover $\bigwedge(D) \leq \bigwedge A$, since for any x in A, $\bigwedge B_x \leq x$, hence by definition of $\overline{f}$, $\overline{f}(\bigwedge A) \geq \bigwedge f(D) = b$, and we are done.

$3 \Rightarrow 4$: If $X \in A_{H_f}$ and $x, y \in X$ are such that $C_{H_f}(x) \neq C_{H_f}(y)$ then we can find a subset A of $\texttt{tr}(f)$ such that $\pi_1(A) \in \mathcal{C}(\mathcal{B}^n)$ and $\pi_2(A) \notin \mathcal{C}(\mathcal{B})$; it is clear then that any extensional upper bound of f will not preserve the coherence on $\pi_1(A)$ and henceforth will not be sequential.

$4 \Rightarrow 1$: Immediate by definition of H_f.

∎

We can observe that Berry's function g is subsequential, whereas *por* is not (see example 4).

Given a set $A = \{v_1, \ldots, v_k\} \subseteq \mathcal{B}^n$, there exist in general a number of functions whose minimal points are exactly the elements of A. For instance, if the v_i are pairwise unbounded, there exists 2^k such functions. The following lemma states that, among these functions, the subsequential ones are those whose degree of parallelism is minimal.

Lemma 2 *Let $f, g : \mathcal{B}^n \to \mathcal{B}$ be such that g is subsequential and $\pi_1(\mathtt{tr}(f)) = \pi_1(\mathtt{tr}(g))$. Then $g \leq_{\mathtt{par}} f$.*

Proof: Let M be a PCF term which defines the sequential upper bound $\overline{g}$ of g, defined as in proposition 5.

Let us define $g_0 : \mathcal{B}^n \to \mathcal{B}$ by

$$g_0 = [\![\lambda f \lambda x_1 \ldots x_n.\mathtt{if}\ f x_1 \ldots x_n\ \mathtt{then}\ M x_1 \ldots x_n\ \mathtt{else}\ M x_1 \ldots x_n]\!] f$$

If we prove that $g_0 = g$ we are done. Let $\overline{a} = (a_1, \ldots, \_n) \in \mathcal{B}^n$, and suppose $g(\overline{a}) = b \neq \perp$; then $f(\overline{a}) \neq \perp$ and $\overline{g}(\overline{a}) = b$. Hence $g_0(\overline{a}) = b$. Viceversa if $g_0(\overline{a}) = b \neq \perp$ then $f(\overline{a}) \neq \perp$ and hence $g(\overline{a}) \neq \perp$ as well. Since $g(\overline{a}) \leq \overline{g}(\overline{a}) = b$, we get $g(\overline{a}) = b = g_0(\overline{a})$ and we are done. ■

Our main result of section 5 is that, if there exists a morphism $\alpha : H_f \to H_g$, then $f \leq_{\mathtt{par}} g$. The following lemma introduce a key notion towards that result, namely the one of *slice function.* The idea is the following: in order to reduce $f : \mathcal{B}^m \to \mathcal{B}$ to $g : \mathcal{B}^n \to g$ we start by trasforming the minimal points of f into the ones of g. This amounts to define a function from B^m to B^n, that we describe as a set of functions $f_1, \ldots, f_n : \mathcal{B}^m \to \mathcal{B}$. If these functions are g-definable, then we can already g-define a function which converges if and only if f converges, namely

$$h = \lambda\ x_1 \ldots x_n\ g((f_1\overline{x}) \ldots (f_n\overline{x}))$$

and we are left with the problem of forcing h to agree with f whenever it converges.

For the time being we show that, if the f_i's are defined via a hypergraph morphism $\alpha : H_f \to H_g$, then they are subsequential, hence "relatively simple".

Lemma 3 *Let $f : \mathcal{B}^m \to \mathcal{B}$, $g : B^n \to \mathcal{B}$ be monotone functions and $\alpha : H_f \to H_g$ be a weak morphism. For $1 \geq i \geq n$ let f_i be the function defined by* [3]

$$\mathtt{tr}(f_i) = \{(v, w^i) | v \in \pi_1(\mathtt{tr}(f)), \alpha(v) = w, w^i \neq \perp\}$$

Then for all $A \subseteq \mathtt{tr}(f_i)$, if $\pi_1(A) \in \mathcal{C}(\mathcal{B}^m)$ then $\pi_2(A) \in \mathcal{C}(\mathcal{B})$.

*(we will call f_i the $ith-$*slice of g following f and α*)*

[3] see proviso 1.

Proof: Let A be a coherent subset of $\mathtt{tr}(f_i)$. By definition of f_i we know that for any $v \in \pi_1(A)$, $\alpha(v)^i \neq \perp$. Moreover $\alpha(\pi_1(A)) \in \mathcal{C}(\mathcal{B}^n)$, since α preserves hyperarcs. Hence we conclude that for all $v, v' \in \pi_1(A)$, $\alpha(v)^i = \alpha(v')^i$, i.e. that $\pi_2(A) = \{\alpha(v)^i \mid v \in \pi_1(A)\} \in \mathcal{C}(\mathcal{B})$.

∎

By proposition 5 and lemma 3 we get:

Corollary 1 *Let $f : \mathcal{B}^m \to \mathcal{B}$, $g : B^n \to \mathcal{B}$ be monotone functions and $\alpha : H_f \to H_g$ be a weak morphism. All the slices of g following f and α are subsequential.*

Example 5: Berry's function g, defined in example 3, is *por*-definable, as any other monotone function. Let us define a morphism $\alpha : H_g \to H_{por}$, and see how the construction of the two slices of *por* following g and α provides directly a way of constructing the PCF-term defining g with respect to *por*. Let $v_1 = (\perp, \mathtt{tt}, \mathtt{ff})$, $v_2 = (\mathtt{ff}, \perp, \mathtt{tt})$ and $v_3 = (\mathtt{tt}, \mathtt{ff}, \perp)$ be the minimal points of g and $w_1 = (\perp, \mathtt{tt})$, $w_2 = (\mathtt{tt}, \perp)$ and $w_3 = (\mathtt{ff}, \mathtt{ff})$ those of *por*. It is easy to check that the function $\alpha : V_{H_g} \to V_{H_{por}}$ defined by $\alpha(v_1) = \alpha(v_2) = w_1$ and $\alpha(v_3) = w_2$ is a (strong) morphism from H_g to H_{por}.

The morphism α defines the map from $\pi_1(\mathtt{tr}(f))$ into $\pi_1(\mathtt{tr}(g))$ shown in the following picture:

$$\begin{tikzcd}
(\perp, \mathtt{tt}, \mathtt{ff}) \arrow[r] & (\perp, \mathtt{tt}) \\
(\mathtt{ff}, \perp, \mathtt{tt}) \arrow[ur] & (\mathtt{tt}, \perp) \\
(\mathtt{tt}, \mathtt{ff}, \perp) \arrow[ur] & (\mathtt{ff}, \mathtt{ff})
\end{tikzcd}$$

The corresponding slice functions $f_1, f_2 : \mathcal{B}^3 \to \mathcal{B}$ are then defined by:

$$\mathtt{tr}(f_1) = \{((\mathtt{tt}, \mathtt{ff}, \perp), \mathtt{tt})\}$$
$$\mathtt{tr}(f_2) = \{((\perp, \mathtt{tt}, \mathtt{ff}), \mathtt{tt}), ((\mathtt{ff}, \perp, \mathtt{tt}), \mathtt{tt})\}$$

Both f_1 and f_2 are sequential, hence PCF-definable. For example the following terms M_1, M_2 define f_1, f_2 respectively:

$$M_1 = \lambda x\ y\ z\ \mathtt{if}\ x\ \mathtt{then}\ (\mathtt{if}\ y\ \mathtt{then}\ \perp\ \mathtt{else}\ \mathtt{tt})\ \mathtt{else}\ \perp$$

$M_2 = \lambda x\, y\, z\, \texttt{if}\ z\ \texttt{then}\ (\texttt{if}\ x\ \texttt{then}\ \perp\ \texttt{else}\ \texttt{tt}\,)\ \texttt{else}\ (\texttt{if}\ y\ \texttt{then}\ \texttt{tt}\ \texttt{else}\ \perp)$

The pair (M_1, M_2) realizes a sequential trasformation of the minimal points of g onto (some of) the minimal points of f. This allows to construct a term M defining g with respect to *por* as follows:

$$M = \lambda f\ \lambda x\ y\ z\ f(M_1\ x\ y\ z)\ (M_2\ x\ y\ z)$$

It is easy to check that $[\![M]\!]por = g$.

■

The theorem of the following section generalize the situation above: we show that, given a (weak) morphism $\alpha : H_f \to H_g$, the slices of g following f are g-definable (even if in general they are not sequential), and this is enough to construct a PCF-term which g-defines f.

5 Hypergraph morphisms and degrees

Theorem 1 *Let $f : \mathcal{B}^l \to \mathcal{B}$, $g : B^m \to \mathcal{B}$ be monotone functions. If $\mathcal{WH}(H_f, H_g) \neq \emptyset$ then $f \leq_{\texttt{par}} g$.*

Proof:

Let $\alpha : H_f \to H_g$ be a weak morphism. We prove the theorem by induction on $k = \#\texttt{tr}(f)$.

If $k = 1$ f is sequential (strongly stable), hence PCF-definable, and $f \leq_{\texttt{par}} g$ holds trivially.

Suppose now $k = n + 1$; we reason by cases on the structure of H_f:

– $V_{H_f} \notin A_{H_f}$: this means that there exists a sequentiality index for f, that is a component of $\pi_1(\texttt{tr}(f))$ which is not a singleton and which does not contain $\perp$; let i be such a component. Let now define

$$M = \lambda g\ \lambda \overline{x}.\ \texttt{if}\ x_i\ \texttt{then}\ M_{\texttt{tt}}\ g\overline{x}\ \texttt{else}\ M_{\texttt{ff}}\ g\overline{x}$$

where M_ρ, $\rho = \texttt{tt}, \texttt{ff}$, is the term g-defining the subfunction f_ρ of f such that $\pi_i(\pi_1(\texttt{tr}(f_\rho))) = \{\rho\}$. The terms M_ρ do exist by inductive hypotesis: $\#\mathtt{tr}(f_\rho) < \#\mathtt{tr}(f)$, and $\mathcal{WH}(H_{f_\rho}, H_g) \neq \emptyset$ since the restriction of α to H_{f_ρ} is a morphism.

It is easy to check that M $g-$defines f.

– $V_{H_f} \in A_{H_f}$:

Let f_i , $1 \leq i \leq m$, be the ith-slice of g following f and α, and now define $\hat{f}_i$ as

$$\hat{f}_i = \begin{cases} f_i & \text{if } \#\mathtt{tr}(f_i) < \#\mathtt{tr}(f) \\ \lambda\overline{x}.v \text{ for } v = \pi_2(\mathtt{tr}(f_i)) & \text{otherwise}^{4} \end{cases}$$

Let us prove that the $\hat{f}_i$'s are g-definable. The only case to be checked is $\hat{f}_i = f_i$ in the previous definition, since $\lambda\overline{x}.v$ is PCF-definable.

By corollary 1 we know that any f_i is subsequential, hence by lemma 2 $f_i \leq_{\mathtt{par}} f^i$, where $\mathtt{tr}(f^i) = \{v \in \mathtt{tr}(f) \mid \pi_1(v) \in \pi_1(tr(f_i))\}$. Now $\#\mathtt{tr}(f^i) < \#\mathtt{tr}(f)$, and, as above, $\mathcal{WH}(H_{f^i}, H_g) \neq \emptyset$. Hence by inductive hypothesis $f^i \leq_{\mathtt{par}} g$, and finally $f_i \leq_{\mathtt{par}} g$ by transitivity of $\leq_{\mathtt{par}}$. Let M_i be a term g-defining $\hat{f}_i$.

Before constructing a term M g-defining f let us prove that we can already g-define a "convergence test" for f, i.e. that for all $\overline{x} = (x_1, \ldots, x_l) \in \mathcal{B}^l$

$$f(\overline{x}) \neq \perp \;\Leftrightarrow\; g([\![M_1]\!]g\overline{x}, \ldots, [\![M_m]\!]g\overline{x}) \neq \perp$$

The direction $\Rightarrow$ is trivial, since the $\hat{f}_i$'s are upper bounds of the f_i's, hence if there exists $v \in \pi_1(\mathtt{tr}(f))$ such that $v \leq \overline{x}$, then $([\![M_1]\!]g\overline{x}, \ldots, [\![M_m]\!]g\overline{x}) \geq \alpha(v)$.

For the opposite direction, let us suppose that for all $v \in \pi_1(\mathtt{tr}(f))$, $\overline{x} \not\geq v$, and hence $f(\overline{x}) = \perp$. By definition of the $\hat{f}_i$'s we know that for all $w \in \alpha(V_{H_f})$, $([\![M_1]\!]g\overline{x}, \ldots, [\![M_m]\!]g\overline{x}) \leq w$, since, under the hypothesis $f(\overline{x}) = \perp$, we have that for all $1 \leq j \leq m$, $[\![M_j]\!]g\overline{x} \neq \perp$ if and only if $\hat{f}_j = \lambda\overline{x}\, b$ ($b \in \{$ `tt` , `ff` $\}$) if and only if for all $w \in \alpha(V_{H_f})$ $w^j = b$.

Since H_f has at least one hyperarc, we know that $\#\alpha(V_{H_f}) \geq 2$, and by minimality of the elements of $\pi_1(\mathtt{tr}(g))$ we conclude that for all $w \in \pi_1(\mathtt{tr}(g))$ $([\![M_1]\!]g\overline{x}, \ldots, [\![M_m]\!]g\overline{x}) < w$, and hence $g([\![M_1]\!]g\overline{x}, \ldots, [\![M_m]\!]g\overline{x}) = \perp$.

We can now conclude the proof, again by case-reasoning on the structure of H_f:

· V_{H_f} is a monochromatic hyperarc (w.l.o.g. assume that all vertices are white). Then it is easy to check that the term

$$M = \lambda g\ \lambda\overline{x}.\ \mathtt{if}\ g(M_1 g\overline{x} \ldots M_m g\overline{x})\ \mathtt{then\ tt\ else\ tt}$$

$g-$defines f.

· V_{H_f} is not monochromatic: we first note that in this case

$$\forall x, y\ \{x, y\} \subseteq X \in A_{H_f}\ C(x) = C(y) \Leftrightarrow C(\alpha(x)) = C(\alpha(y))$$

i.e. α acts as the identity or the "negation" on colours (this is a trivial consequence of the definition of weak morphism, since V_{H_f} is polychromatic hyperarc). We define then

$$M = \lambda g\ \lambda \overline{x}.\ \epsilon g(M_1 g\overline{x} \dots M_m g\overline{x})$$

where ϵ is the boolean identity or the boolean negation according to how α acts on colours. Then again it is easily checked that M $g-$defines f.

■

In the following example, we "run" the proof of the theorem in order to construct a PCF-term which defines f_3 relatively to g_3, these functions being defined in the example 1.

Example 6:

Since $H_{f_3} = H_{g_3} =$

$$(\{1,2,3\}, \{\{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}, C(1) = C(2) = C(3) = white)$$

we can choose $id : H_{f_3} \to H_{g_3}$ as morphism. The corresponding trasformation of $\pi_1(\mathtt{tr}(f_3))$ onto $\pi_1(\mathtt{tr}(g_3))$ is then:

$$(\mathtt{tt}, \perp, \perp) \longrightarrow (\perp, \mathtt{tt}, \mathtt{tt})$$

$$(\perp, \mathtt{tt}, \perp) \longrightarrow (\mathtt{tt}, \perp, \mathtt{tt})$$

$$(\perp, \perp, \mathtt{tt}) \longrightarrow (\mathtt{tt}, \mathtt{tt}, \perp)$$

The slice functions are hence defined by:

$$\mathtt{tr}(f_1) = \mathtt{tr}(\hat{f}_1) = \{((\perp, \mathtt{tt}, \perp), \mathtt{tt}), ((\perp, \perp, \mathtt{tt}), \mathtt{tt})\}$$

$$\mathtt{tr}(f_2) = \mathtt{tr}(\hat{f}_2) = \{((\mathtt{tt}, \perp, \perp), \mathtt{tt}), ((\perp, \perp, \mathtt{tt}), \mathtt{tt})\}$$

$$\mathtt{tr}(f_3) = \mathtt{tr}(\hat{f}_3) = \{((\mathtt{tt}, \perp, \perp), \mathtt{tt}), ((\perp, \mathtt{tt}, \perp), \mathtt{tt})\}$$

The f_i's being non-sequential, we have to re-run our proof in order to define them relatively to g_3. Let us consider f_1. The following picture represents a morphism $\alpha' : H_{f_1} \to H_{g_3}$:

$$
\begin{array}{ccc}
 & & (\perp,\ \mathtt{tt}\ ,\ \mathtt{tt}\ ) \\
(\perp,\ \mathtt{tt}\ ,\perp) & \longrightarrow & (\ \mathtt{tt}\ ,\perp,\ \mathtt{tt}\ ) \\
(\perp,\perp,\ \mathtt{tt}\ ) & \longrightarrow & (\ \mathtt{tt}\ ,\ \mathtt{tt}\ ,\perp)
\end{array}
$$

The corresponding slice functions are

$$f'_1 = f_1 \neq \hat{f}'_1 = \lambda\overline{x}\ \ \mathtt{tt}$$

$$\mathtt{tr}(f'_2) = \mathtt{tr}(\hat{f}'_2) = \{((\perp,\perp,\ \mathtt{tt}\ ),\ \mathtt{tt}\ )\}$$

$$\mathtt{tr}(f'_3) = \mathtt{tr}(\hat{f}'_3) = \{((\perp,\ \mathtt{tt}\ ,\perp),\ \mathtt{tt}\ )\}$$

Now the f'_is are trivially g_3-definable (their traces are singletons). The corresponding terms are $M'_1 = \lambda h\lambda\overline{x}\ \ \mathtt{tt}$, $M'_2 = \lambda h\lambda\overline{x}$ $\mathtt{if}$ x_3 $\mathtt{then}\ \ \mathtt{tt}\ \ \mathtt{else}\ \ \perp$ $M'_3 = \lambda h\lambda\overline{x}$ $\mathtt{if}$ x_2 $\mathtt{then}\ \ \mathtt{tt}\ \ \mathtt{else}\ \perp$.

The term M_1 g_3-defining f_1 is thence:

$$M_1 = \lambda h\lambda\overline{x}\ \mathtt{if}\ h\ (M'_1\ h\ \overline{x})\ (M'_2\ h\ \overline{x})\ (M'_3\ h\ \overline{x})\ \mathtt{then}\ true\ \mathtt{else}\ true$$

By eliminating redundant conditional statements (and with some abuse of notation) we obtain the following definition of f_1:

$$f_1 = \lambda\overline{x}\ g_3(\ \mathtt{tt}\ , x_3,\ x_2)$$

similar constructions allow to obtains the terms g_3-defining f_2 and f_3, and finally we get (again with some simplifications)

$$f_3 = \lambda\ x_1\ x_2\ x_3\ g_3(g_3(\ \mathtt{tt}\ ,\ x_3,\ x_2),\ g_3(x_3,\ \ \mathtt{tt}\ ,\ x_1),\ g_3(x_2,\ x_1,\ \ \mathtt{tt}\ ))$$

We can observe that this construction leads to a term wich is more complex than the one showed in example 1.

■

We can of course remark that:

Corollary 2 *If H_f and H_g are strongly isomorphic, then $[f] = [g]$.*

This corollary answers to a question asked in the introduction: functions having the same hypergraph are equiparallel.

Another remark concerns subsequential functions: if H_f has monochromatic hyperarcs then any function $\alpha : V_{H_f} \to V_{H_g}$ which preserves hyperarcs is a weak morphism. Hence:

Corollary 3 *Let $\mathcal{F}$ be the forgetful functor from colored hypergraph to hypergraph, and let $\alpha : \mathcal{F}(\mathcal{H}_{\{}) \to \mathcal{F}(\mathcal{H}_{\}})$ be a hypergraph morphism. If f is subsequential then $f \leq_{\texttt{par}} g$.*

One natural question is of course whether hypergraph morphisms preserve $\leq_{\texttt{par}}$ relations. The answer is no; for exemple consider

Example 7: Let $3 - \texttt{por} : \mathcal{B}^3 \to \mathcal{B}$ be defined by

$$tr(3 - \texttt{por}) = \{((\texttt{ tt }, \bot, \bot), \texttt{ tt }), ((\bot, \texttt{ tt }, \bot), \texttt{ tt }), ((\bot, \bot, \texttt{ tt }), \texttt{ tt })\}$$

The associated hypergraph is:

$$H_{3-\texttt{por}} = (\{1,2,3\}, \{\{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}, C(1) = C(2) = C(3) = \texttt{white})$$

It is easy to see that there exists no morphism $m : H_{3-\texttt{por}} \to H_{\texttt{por}}$. Nevertheless $3 - \texttt{por} \leq_{\texttt{par}} \texttt{por}$, since for instance

$$3 - \texttt{por} = [\![M]\!]\texttt{por}$$

where

$$M = \lambda f\ \lambda x_1 x_2 x_3\ \texttt{if}\ f(f(x_1, x_2), x_3)\ \texttt{then tt else}\ \bot$$

■

Although the notion of hypergraph morphism is too weak in order to get a completeness result we do believe that hypergraph representation does retain enough information on functions in order to achieve such completeness. The price to pay seems to be the use of more involved notions than hypergraphs morphisms.

References

[1] S. Abramsky, R. Jagadeesan, P. Malacaria. *Full abstraction for PCF (Extended Abstract).* Proc. of TACS 94, Lecure Notes in Computer Science 789, Springer, 1994.

[2] R. Amadio, P.-L Curien. *Selected Domains.* To appear.

[3] G. Berry. *Stable models of typed lambda-calculi.* Proc. 5th Int. Coll. on Automata, Languages and Programming, Lecure Notes in Computer Science 62, Springer, 1978. LNCS 62, 1978.

[4] G. Berry. *Modèles complètement adéquats et stables des lambda-calculs typés.* Thèse de Doctorat d'Etat, Université Paris 7, 1979.

[5] A. Bucciarelli, T. Ehrhard. *Sequentiality in an extensional framework.* Information and Computation, Volume 110, Number 2, 1994.

[6] A. Bucciarelli. *Degrees of Parallelism in the Continous Type Hierarchy.* To appear in Theoretical Computer Science.

[7] L. Colson, T. Ehrhard.

[8] T. Ehrhard. *Parallel and Serial Hypercoherences.* Manuscript 1995.

[9] J.M.E. Hyland, L. Ong. *On full abstraction for PCF: I, II and III* (preliminary version, september 1995).

[10] G. Kahn and G. Plotkin. *Domaines Concrets.* Rapport IRIA-LABORIA 336, 1978, republished in the special issue of Theoretical Computer Science dedicated to Professor C. Böhm's 70th birthday, 1993.

[11] R. Milner. *Models of LCF.* Computer Science Departement Memo. AIM-186/CS 332, Stanford University, 1973.

[12] P. O'Hearn, J. Riecke. *Kripke Logical Relations and PCF.* To appear in Information and Computation.

[13] G. Plotkin. *LCF considered as a programming language.* Theoretical Computer Science 5, 223-256, 1977.

[14] V. Y. Sazonov. *Degrees of Parallelism in Computations.* Proc. Conference on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 45, 1976.

[15] D. Scott. *A type theoretic alternative to OWHY, CUCH, ISWIM.* 1969, published in the special issue of Theoretical Computer Science dedicated to Professor C. Böhm's 70th birthday, 1993.

[16] K. Sieber. *Reasoning about Sequential Functions via Logical Relations.* Proc. LMS Symposium on Applications of Categories in Computer Science, M. Fourman, P. Johnstone, A. Pitts eds, LMS Lecture Note Series 177, Cambridge University Press, 1992.

[17] A. Stoughton. *Mechanizing Logical Relations.* Proc. Ninth International Conference on Mathematical Foundations of Programming Semantics, Lecture Notes in Computer Science 802, 1994.

[18] M.B. Trakhtenbrot. *On Representation of Sequential and Parallel Functions.* In Proc. Fourth Symp. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 32, Springer 1975.

[19] J. Vuillemin. *Proof Techniques for Recursive Programs.* Ph.D. Thesis, Stanford University, 1973.