Some category theoretical properties related to a model for a polymorphic lambda-calculus

Some category theoretical properties related to a model for a polymorphic lambda-calculus

by

H.M.M. Ten Eikelder
C. Hemerik

89/3

February, 1989

SOME CATEGORY THEORETICAL PROPERTIES RELATED TO A MODEL FOR A POLYMORPHIC λ-CALCULUS

H.M.M. Ten Eikelder
C. Hemerik

Abstract.

A model for a second order polymorphic lambda calculus is sketched. Some category theoretical questions appearing in the model construction are extensively treated.

COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science Eindhoven University of Technology. Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review. Copies of these notes are available from the author or the editor.

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands

editors: prof.dr.M.Rem
prof.dr.K.M.van Hee.

CONTENTS

1. INTRODUCTION AND MODEL DESCRIPTION

Consider a typed λ-calculus, where type expressions are generated by production rules of the form

$$\text{Texp} ::= \text{Tconst} \mid \text{Tvar} \mid \text{Texp} \to \text{Texp} \mid \forall(\Lambda\, \text{Tvar} \mid \text{Texp}) .$$

Here Tconst generates a set of type constants (for instance int, bool) and Tvar generates a set of type variables. The other two possible type expressions correspond to function types and polymorphic types. Other type constructors like $+$, $\times$, ν and $\sum$ (corresponding to sum types, product types, recursive types and abstract data types) can easily be added and do not affect the following essentially.

Expressions are generated by

$$\begin{aligned}\text{Exp} ::= {} & \text{Const} \mid \text{Var} \mid (\lambda\, \text{Var} : \text{Texp} | \text{Exp}) \mid (\text{Exp}\, \text{Exp}) \mid \\ & (\Lambda\, \text{Tvar} | \text{Exp}) \mid \text{Exp}\, \text{Texp} ,\end{aligned}$$

supplied with a type deduction system. Const yields a set of constants (for instance 0, succ, true), while Var generates a set of variables. The next two rules are the introduction and elimination rules for expressions with function types. The last two rules are the introduction and elimination rules for expressions with a polymorphic type.

A generalized version of this language (where type expressions are constructor expressions of kind T) can be found in Bruce, Meyer and Mitchell [BMM] or Hemerik and Ten Eikelder [HTE].

The type deduction system strongly resembles the one given in Ten Eikelder and Mak [TEM]. The type deduction rules for expressions are given by

$$A \;\triangleright\; c_{te} : te \qquad\qquad c_{te} \in \text{Const},\ \text{FTV}(te) = \emptyset$$

$$\frac{A_1 \;\triangleright\; tx}{A_1; x : tx; A_2 \;\triangleright\; x : tx} \qquad\qquad \begin{array}{l} x \notin \text{FV}(A_2) \\ \text{FTV}(A_2) \cap \text{FTV}(tx) = \emptyset \end{array}$$

$$\frac{\begin{array}{l} A \;\triangleright\; tx, te \\ A; x : tx \;\triangleright\; e : te \end{array}}{A \;\triangleright\; (\lambda x : tx \,|\, e) : tx \to te}$$

$$\frac{A \;\triangleright\; f : te \to te1 ,\, e : te}{A \;\triangleright\; fe : te1}$$

$$\frac{A; t \;\triangleright\; e : te}{A \;\triangleright\; (\Lambda t.e) : \forall(\Lambda t.te)}$$

$$\frac{A \;\triangleright\; e : \forall(\Lambda t.te) ,\, te1}{A \;\triangleright\; e\, te1 : te^{t}_{te1}}$$

Here A is a syntactic type assignment, i.e. a sequence consisting of type variables and type assignments $(x : tx)$. The functions FV and FTV yield the free variables and free type variables of their arguments (!).

In the remaining part of this section we shall briefly indicate how a model for the polymorphic λ-calculus, given above, can be constructed. The model described below can easily be seen to fit with the general model definition given by Bruce, Meyer and Mitchell [BMM]. In the model definition we shall meet some technical, category theoretical problems. In Sections 2, 3, 4 and 5 of this note these problems will be extensively treated.

Let Texp(V) be the set of type expressions with free type variables in the set V. An important aspect of a model construction is to associate a suitable domain to each type expression. As ultimately all free type variables will be bound to a closed type expression (using a type environment η : Tvar $\rightarrow$ Texp($\emptyset$)), we shall associate a domain (in fact a C.P.O.) DOM_{te} to each closed type expression te.

The domain associated to a function type $te1 \rightarrow te2$ should be equal or isomorphic to a sufficiently large subset (denoted by square brackets) of the functions of Dom_{te1} to Dom_{te2}, i.e.

$$\mathrm{Dom}_{te1 \rightarrow te2} \approx [\mathrm{Dom}_{te1} \rightarrow \mathrm{Dom}_{te2}] . \tag{1.1}$$

The domain $\mathrm{Dom}_{\forall(\Lambda t.te)}$ related to a polymorphic type should be equal or isomorphic to the product of the domains $\mathrm{Dom}_{te^t_{te1}}$, where $te1$ runs over all closed type expressions. In fact $te1$ corresponds to the possible "type arguments" of an expression with type $\forall(\Lambda t.te)$. Hence

$$\mathrm{Dom}_{\forall(\Lambda t.te)} \approx \prod_{te1 \in \mathrm{Texp}(\emptyset)} \mathrm{Dom}_{te^t_{te1}} . \tag{1.2}$$

It is easily seen that the domain in the left-hand side in general also appears in the right-hand side, take for instance $\forall(\Lambda t.t \rightarrow t)$ (the type of the polymorphic identity). This means that (1.1) and (1.2) can only be solved simultaneously, i.e. we compute a "vector of domains" $\mathrm{Dom} = < \mathrm{Dom}_{te} \,|\, te \in \mathrm{Texp}(\emptyset) >$ as a solution of a system of isomorphic domain equations.
More precisely Dom will be found as (the domain part of) the initial fixed point of an ω-continuous endofunction F on a product category $\prod_{te \in \mathrm{Texp}(\emptyset)} K$, where K is some suitable category. Since type expressions contain function types, the function space functor will be used in the definition of F (see (1.3)). On the ω-categories <u>SET</u> and <u>$\mathrm{CPO}_\perp$</u>, the function space functor is contravariant in its first argument, so K cannot be one of these categories. The function space functor FS on the category <u>$\mathrm{CPO}_{\mathrm{PR}}$</u>is an ω-continuous (covariant) bifunctor, which leads to this choice for K.

From now on we shall write Π<u>$\mathrm{CPO}_{\mathrm{PR}}$</u> instead of $\prod_{te \in \mathrm{Texp}(\emptyset)}$ (<u>$\mathrm{CPO}_{\mathrm{PR}}$</u>). With every object D of $\underline{\text{CPO}}_{\text{PR}}$ is associated a constant functor $C_D : \Pi\, \underline{\text{CPO}}_{\text{PR}} \rightarrow \underline{\text{CPO}}_{\text{PR}}$. Let GP, $\pi_{te} : \Pi\, \underline{\text{CPO}}_{\text{PR}} \rightarrow \underline{\text{CPO}}_{\text{PR}}$, be the generalized product functor (see Sections 4 and 5) and the projection functor on component te. The function $F : \Pi\, \underline{\text{CPO}}_{\text{PR}} \rightarrow \Pi\, \underline{\text{CPO}}_{\text{PR}}$ is defined by its components $F_{te} : \Pi\, \underline{\text{CPO}}_{\text{PR}} \rightarrow \underline{\text{CPO}}_{\text{PR}}$, i.e. $F = < F_{te} \mid te \in \text{Texp}(\emptyset) >$. We now give these components. Let ρ be a function which maps type constants to C.P.O.'s. Then

$$\begin{aligned} F_c &= C_{\rho(c)} \qquad (c \in \text{Tconst}) \\ F_{te1 \rightarrow te2} &= FS \circ < \pi_{te1}, \pi_{te2} > \\ F_{\forall(\Lambda t.te)} &= \text{GP} \circ < \pi_{te^t_{te1}} \mid te1 \in \text{Texp} > \end{aligned} \tag{1.3}$$

If the function F is ω-continuous and $\Pi\, \underline{\text{CPO}}_{\text{PR}}$ is an ω-category, there exists an initial fixed point (Dom,Φ) of F (see for instance [SP]). The components of the isomorphism Φ : Dom $\rightarrow F$(Dom) are isomorphisms $\Phi_{te} : \text{Dom}_{te} \rightarrow F_{te}(\text{Dom})$. In particular, this means

$$\Phi_{te1 \rightarrow te2} : \text{Dom}_{te1 \rightarrow te2} \rightarrow F_{te1 \rightarrow te2}(\text{Dom}) = [\text{Dom}_{te1} \rightarrow \text{Dom}_{te2}]$$

and

$$\Phi_{\forall(\Lambda t.te)} : \text{Dom}_{\forall(\Lambda t.te)} \rightarrow F_{\forall(\Lambda t.te)}(\text{Dom}) = \prod_{te1 \in \text{Texp}(\emptyset)} \text{Dom}_{te^t_{te1}} ,$$

in accordance with (1.1) and (1.2). The construction above can easily be extended to allow recursive types and abstract data types. This leads to components of F given by

$$\begin{aligned} F_{\nu(\Lambda t|te)} &= \pi_{te^t_{\nu(\Lambda t|te)}} , \\ F_{\sum(\Lambda t|te)} &= \text{GS} \circ < \pi_{te^t_{te1}} \mid te1 \in \text{Texp}(\emptyset) > , \end{aligned}$$

where GS : $\Pi\, \underline{\text{CPO}}_{\text{PR}} \rightarrow \underline{\text{CPO}}_{\text{PR}}$ is the generalized sum functor.

The semantics of expressions can now easily be given. Recall that we have introduced a mapping ρ : Tconst $\rightarrow$ obj($\underline{\text{CPO}}$), which maps every type constant to a c.p.o. Note that the construction of Dom_{te} (for closed type expressions te) depends only on ρ. We also have to associate a point of a suitable c.p.o. to each (expression-) constant. Hence, we extend ρ to a function on Tconst $\cup$ Const, such that

$$\rho|_{\text{Const}} : \text{Const} \rightarrow \bigcup_{te \in \text{Texp}(\emptyset)} \text{Dom}_{te}$$

with

$$\rho(c_{te}) \in \text{Dom}_{te} \qquad \text{for every } c_{te} \in \text{Const} .$$

Of course the semantics of an expression depends also on the values of the appearing type variables and (expression) variables. An environment is a mapping η with domain $\mathrm{Tvar} \cup \mathrm{Var}$ such that

$$\begin{aligned} \eta|_{\mathrm{Tvar}} &: \mathrm{Tvar} \to \mathrm{Texp}(\emptyset) \\ \eta|_{\mathrm{Var}} &: \mathrm{Var} \to \bigcup_{te \,\in\, \mathrm{Texp}(\emptyset)} \mathrm{Dom}_{te} \, . \end{aligned}$$

Let te be an arbitrary type expression. The closed type expression obtained by substituting $\eta(t)$ for every free type variable t of te will be denoted by $\{\eta\}te$. An environment η will be called consistent with respect to a syntactic type assignment A if for all variables x and type expressions tx such that $A \;\rhd\; x \;:\; tx$

$$\eta(x) \in \mathrm{Dom}_{\{\eta\}tx} \, .$$

Let η be an environment which is consistent with respect to a syntactic type assignment A. The semantics of expressions typable under A in the environment η is defined in the following way:

$$\begin{aligned} [\![A \;\rhd\; c_{te} \;:\; te]\!]\,\eta &= \rho(c_{te}) \, , \\ [\![A \;\rhd\; x \;:\; tx]\!]\,\eta &= \eta(x) \, , \\ [\![A \;\rhd\; (\lambda x \;:\; tx \,|\, e) \;:\; tx \to te]\!]\,\eta &= \\ \Phi^{-1}_{\{\eta\}(tx \to te)}\Big(\boldsymbol{\lambda}\, d \in \mathrm{Dom}_{\{\eta\}tx} \,|\, [\![A; x \;:\; tx \;\;\rhd\; e \;:\; te]\!]\,\eta[d/x]\Big) \, . \end{aligned}$$

Note that if η is consistent with respect to A, then for every $d \in \mathrm{Dom}_{\{\eta\}tx}$, $\eta\,[d/x]$ is consistent with respect to $A; x \;:\; tx$.

$$\begin{aligned} &[\![A \;\rhd\; e1e2 \;:\; te]\!]\,\eta = \\ &\quad \Big(\Phi_{\{\eta\}(te2 \to te)}([\![A \;\rhd\; e1 \;:\; te2 \to te]\!]\,\eta)\Big)\,[\![A \;\rhd\; e2 \;:\; te2]\!]\,\eta \\ &[\![A \;\rhd\; (\Lambda t.e) \;:\; \forall(\Lambda t.te)]\!]\,\eta = \\ &\quad \Phi^{-1}_{\{\eta\}\forall(\Lambda t.te)}\Big(< [\![A; t \;\;\rhd\; e \;:\; te]\!]\,\eta\,[te1/t] \,|\, te1 \in \mathrm{Texp}(\emptyset) >\Big) \, . \end{aligned}$$

Again if η is consistent with respect to A, then $\eta\,[te1/t]$ is consistent with respect to $A; t$.

$$\begin{aligned} &[\![A \;\rhd\; e\; te1 \;:\; te^{t}_{te1}]\!]\,\eta = \\ &\quad \Big(\Phi_{\{\eta\}\forall(\Lambda t.te)}[\![A \;\rhd\; e \;:\; \forall(\Lambda t.te)]\!]\,\eta\Big)_{te1} \, . \end{aligned}$$

The proof that the semantics of expressions is correctly defined and satisfies

$$[\![A \;\rhd\; e \;:\; te]\!]\,\eta \in \mathrm{Dom}_{\{\eta\}te}$$

will not be given here.

In the construction of the c.p.o.'s Dom_{te} various category theoretical questions arise. For instance:

i) Is $\Pi\,\underline{\mathrm{CPO}}_{\mathrm{PR}}$ an ω-category?

ii) Under which conditions is a functor $F : \Pi\,\underline{\mathrm{CPO}}_{\mathrm{PR}} \to \Pi\,\underline{\mathrm{CPO}}_{\mathrm{PR}}$ ω-continuous?

iii) What is the definition of the generalized product and generalized sum functors GP, GS : $\Pi\,\underline{\mathrm{CPO}}_{\mathrm{PR}} \to \underline{\mathrm{CPO}}_{\mathrm{PR}}$?
Are these functors ω-continuous?

The questions i) and ii) are studied extensively in Section 2.
In Section 3 an alternative proof is given of the property that an arbitrary product of an ω-complete category is ω-complete. This proof requires some more category theory.
In Section 4 we define generalized product and generalized sum functors $\Pi\,\underline{\mathrm{CPO}} \to \underline{\mathrm{CPO}}$ and show that they are locally continuous.
Finally, in Section 5 we define the generalized product and generalized sum functors GP, GS : $\Pi\,\underline{\mathrm{CPO}}_{\mathrm{PR}} \to \underline{\mathrm{CPO}}_{\mathrm{PR}}$ and show that they are ω-continuous.

2. INITIAL FIXED POINTS IN PRODUCT CATEGORIES

2.1. Introduction

In [SP,LS,BH] a solution method is described for equations of the form

$$X \cong F \cdot X$$

where X ranges over the objects of a category K and $F : K \to K$ is an endofunctor of that category. When K is an ω-complete category (i.e. a category that has an initial object and in which each ω-chain has a colimit) and F is an ω-cocontinuous functor (i.e. a functor that preserves colimits), the method yields a fixed point, i.e. a pair (A, Φ) such that

$$A \in \text{obj}(K)$$

$$\Phi : A \to F \cdot A \quad \text{is an isomorphism .}$$

Moreover, the fixed point (A, Φ) is initial in the category of fixed points of F. The construction method is a systematic generalization of the least fixed point construction for ω-continuous functions on ω-cpo's.

In this section we consider the solution to a system of equations

$$\begin{aligned} X_1 &\cong F_1 \cdot < X_1, \cdots, X_i, \cdots > \\ &\vdots \\ X_i &\cong F_i \cdot < X_1, \cdots, X_i, \cdots > \\ &\vdots \end{aligned}$$

where X_i ranges over the objects of a category K_i and $F_i : K_1 \times \cdots \times K_i \times \cdots \to K_i$. Such a system of equations can be handled as a single equation

$$< X_1, \cdots, X_i, \cdots > \cong < F_1, \cdots, F_i, \cdots > \cdot < X_1, \cdots, X_i, \cdots >$$

in the product category $K_1 \times \cdots \times K_i \times \cdots$. We are interested in conditions on the categories K_i and the functors F_i that imply solvability of the above equation. It will turn out that the conditions that each K_i is an ω-category and that each F_i is ω-continuous are sufficient. Although this is intuitively clear ("properties smoothly generalize to products if everything is defined pointwise") a full proof turns out to be rather involved and to contain some unexpected swaps of universal and existential quantifications, reason why it has been recorded in this note.

2.2. Preliminaries

The notation $< E \,|\, i \in I >$ will be used instead of $(\boldsymbol{\lambda}\, i \in I \,|\, E)$ for functions with parameter i ranging over domain I.
Sequences will be considered as functions with domain $\mathbb{N}$. Function application will

be denoted by subscripting; so if x is a function with domain I, then $x = < x_i \mid i \in I >$. In some cases the domain will not be mentioned explicitly to reduce notational clutter. In such cases the variable named i is always assumed to range over a set I mentioned in the context; the variable j is always assumed to range over $\mathbb{N}$. So $< E \mid i >$ abbreviates $< E \mid i \in I >$, and $< E \mid j >$ abbreviates $< E \mid j \in \mathbb{N} >$.
Functor application will be denoted by $\cdot$.
The symbol Π will be defined for categories and functors. In all other cases it denotes generalized Cartesian product of sets. $\Pi \{V_i \mid i \in I\}$ will be written $\prod_{i \in I} V_i$, or just $\prod_i V_i$. Note that if $(\forall i \in I \ : \ v_i \in V_i)$, then $< v_i \mid i \in I > \in \prod_i V_i$.

In some proof steps use will be made of the following marked equivalences:

(Eq. 1) $(\forall i \in I \ : \ (\forall v_i \in V_i \ : \ P(v_i)))$

$$\Leftrightarrow$$

$$(\forall v \in \prod_i V_i \ : \ (\forall i \in I \ : \ P(v_i)))$$

(Eq. 2) $(\forall i \in I \ : \ (\exists! \, v_i \in V_i \ : \ P(v_i)))$

$$\Leftrightarrow$$

$$(\exists! \, v \in \prod_i V_i \ : \ (\forall i \in I \ : \ P(v_i)))$$

As for many of the properties we consider, the equivalence

$$P(v) \Leftrightarrow (\forall i \in I \ : \ P(v_i))$$

also holds, these equivalences might be considered the formal rendering of the aforementioned phrase that "properties defined pointwise generalize to products".

2.3. Product of categories

Given a collection $\{K_i \mid i \in I\}$ of categories, we can form a new category, which has as objects tuples $< A_i \mid i \in I >$ of objects $A_i \in \mathrm{obj}(K_i)$. For two such objects $< A_i \mid i \in I >$ and $< B_i \mid i \in I >$ the set of morphisms consists of tuples $< f_i \mid i \in I >$ where, for all i, f_i is a morphism from A_i to B_i. The category thus obtained is called the product of $\{K_i \mid i \in I\}$ and is denoted by $\Pi \{K_i \mid i \in I\}$. In this section we consider relations between properties of the categories K_i and of their product. The most important one is Lemma 3.5 which states that the product is an ω- category iff each K_i is an ω-category.

Definition 2.3.1 [product of categories].

Let I be a set;
$K = \{K_i \mid i \in I\}$ a collection of categories.

The product category ΠK is the category M with

- $\text{obj}(M) = \prod_{i \in I} \text{obj}(K_i)$.

- for all $A, B \in \text{obj}(M)$: $\text{hom}_M(A, B) = \prod_{i \in I} \text{hom}_{K_i}(A_i, B_i)$.

- for all $A \in \text{obj}(M)$: $I_A = < I_{A_i} \mid i \in I >$.

- for all $A, B, C \in \text{obj}(M)$; $f \in \text{hom}(A, B)$; $g \in \text{hom}(B, C)$: $g \circ f = < g_i \circ f_i \mid i \in I >$.

□

Definition 2.3.2 [projection functors].

Let I be a set;
$K = \{K_i \mid i \in I\}$ a collection of categories.

The projection functors π_i $(i \in I)$ are given by

- $\pi_i \ : \ \Pi K \rightarrow K_i$.

- $\pi_i \cdot u = u_i$, where u is either an object or a morphism.

□

Lemma 2.3.3.

Let I be a set;
$K = \{K_i \mid i \in I\}$ a collection of categories;
For all i:
let $U_i, A_i \in \text{obj}(K_i)$;
for all j:
let $D_{ij} \in \text{obj}(K_i)$;
$f_{ij} \in \text{hom}(D_{ij}, D_{i,j+1})$;
$\alpha_{ij} \in \text{hom}(D_{ij}, A_i)$.

1. $(\forall i \ : \ U_i$ is initial in $K_i) \Leftrightarrow (< U_i \mid i >$ is initial in $\prod_i K_i)$.

2. $(\forall i \ : < (D_{ij}, f_{ij}) \mid j >$ is ω-chain in $K_i)$

   $\Leftrightarrow$

   $< (< D_{ij} \mid i >, < f_{ij} \mid i >) \mid j >$ is ω-chain in $\prod_i K_i$.

3. $(\forall i \ : \ (A_i, < \alpha_{ij} \mid j >)$ is cocone for $< (D_{ij}, f_{ij}) \mid j >$

   $\Leftrightarrow$

   $(< A_i \mid i >, << \alpha_{ij} \mid i > \mid j >)$ is cocone for $< (< D_{ij} \mid i >, < f_{ij} \mid i >) \mid j >$.

4. As 3, with cocone replaced by colimit.

5. $(\forall i \ : \ K_i$ is ω-category$) \Leftrightarrow \prod_i K_i$ is ω-category.

6. The projection functors $\pi_i : \prod_i K_i \to K_i$ are ω-(co)continuous.

Proof.

1. $(\forall i : U_i$ is initial in $K_i)$
$= $ [def. initiality]
$(\forall i : (\forall V_i \in \text{obj}(K_i) : (\exists! f_i \in \text{hom}(U_i, V_i))))$
$= $ [Eq. 1]
$(\forall V \in \text{obj}(\prod_i K_i) : (\forall i : (\exists! f_i \in \text{hom}(U_i, V_i))))$
$= $ (Eq. 2, Def. 2.3.1]
$(\forall V \in \text{obj}(\prod_i K_i) : (\exists! f \in \text{hom}(< U_i \mid i >, V)))$
$= $ [def. initiality]
$< U_i \mid i >$ is initial in $\prod_i K_i$.

2. $(\forall i : < (D_{ij}, f_{ij}) \mid j >$ is ω-chain in $K_i)$
$= $ [def. ω-chain]
$(\forall i : (\forall j : D_{ij} \in \text{obj}(K_i)$ and $f_{ij} \in \text{hom}(D_{ij}, D_{i,j+1})))$
$=$
$(\forall j : (\forall i : D_{ij} \in \text{obj}(K_i)$ and $f_{ij} \in \text{hom}(D_{ij}, D_{i,j+1})))$
$= $ [Def. 2.3.1]
$(\forall j : < D_{ij} \mid i > \in \text{obj}(\prod_i K_i)$ and $< f_{ij} \mid i > \in \text{hom}(< D_{ij} \mid i >, < D_{i,j+1} \mid i >))$.
$= $ [def. ω-chain]
$< (< D_{ij} \mid i >, < f_{ij} \mid i >) \mid j >$ is ω-chain in $\prod_i K_i$.

3. $(\forall i : (A_i, < \alpha_{ij} \mid j >)$ is cocone for $< (D_{ij}, f_{ij}) \mid j >)$
$= $ [def. cocone]
$(\forall i : (\forall j : \alpha_{ij} = \alpha_{i,j+1} \circ f_i))$
$=$
$(\forall j : (\forall i : \alpha_{ij} = \alpha_{i,j+1} \circ f_i))$
$=$
$(\forall j : < \alpha_{ij} \mid i > = < \alpha_{i,j+1} \circ f_i \mid i >)$
$= $ [Def. 2.3.1]
$(\forall j : < \alpha_{ij} \mid i > = < \alpha_{i,j+1} \mid i > \circ < f_i \mid i >)$
$= $ [def. cocone]
$(< A_i \mid i >, << \alpha_{ij} \mid i > \mid j >)$ is cocone for $< (< D_{ij} \mid i >, < f_{ij} \mid i >) \mid j >$.

4. $(\forall i : (A_i, < \alpha_{ij} \mid j >)$ is colimit for $< (D_{ij}, f_{ij}) \mid j >)$
$= $ [def. colimit]
$(\forall i : (\forall (B_i, < \beta_{ij} \mid j >) \in$ cocones for $< (D_{ij}, f_{ij}) \mid j >$
$: (\exists! f_i \in \text{hom}(A_i, B_i) : (\forall j : \beta_{ij} = f_i \circ \alpha_{ij}))$
$)$
$)$
$= $ [Eq. 1]

$(\forall (< B_i \,|\, i >, << \beta_{ij} \,|\, j > \,|\, i >) \in$ cocones for $< (< D_{ij} \,|\, i >, < f_{ij} \,|\, i >) \,|\, j >$
$: \ (\forall i \ : \ (\exists! f_i \in \hom(A_i, B_i) \ : \ (\forall j \ : \ \beta_{ij} = f_i \circ \alpha_{ij})))$
)
= [Eq. 2]
$(\forall (< B_i \,|\, i >, << \beta_{ij} \,|\, j > \,|\, i >) \in$ cocones for $< (< D_{ij} \,|\, i >, < f_{ij} \,|\, i >) \,|\, j >$
$: (\exists! f \in \hom(< A_i \,|\, i >, < B_i \,|\, i >)$
$: \ (\forall i, j \ : \ \beta_{ij} = f_i \circ \alpha_{ij})$
)
)
= [Def. 2.3.1]
$(\forall (< B_i \,|\, i >, << \beta_{ij} \,|\, j > \,|\, i >) \in$ cocones for $< (< D_{ij} \,|\, i >, < f_{ij} \,|\, i >) \,|\, j >$
$: \ (\exists! f \in \hom(< A_i \,|\, i >, < B_i \,|\, i >)$
$: \ (\forall j \ : < \beta_{ij} \,|\, i > = < f_i \,|\, i > \circ < \alpha_{ij} \,|\, i >)$
)
)
= [def. colimit]
$(< A_i \,|\, i >, << \alpha_{ij} \,|\, i > \,|\, j >)$ is colimit.

5. From 1 and 4.

6. From 4.

□

2.4. Product and tupling of functors

Given two collections $\{K_i \,|\, i \in I\}$ and $\{L_i \,|\, i \in I\}$ of categories and a collection $\{F_i \ : \ K_i \to L_i \,|\, i \in I\}$ of functors, we can form a new functor $\prod_i F_i$, called the product of the F_i, which is defined "pointwise".
Similarly, given a category K, a collection $\{L_i \,|\, i \in I\}$ of categories and a collection $\{F_i \ : \ K \to L_i \,|\, i \in I\}$ of functors, we can form a functor tuple $< F_i \,|\, i \in I >$, which maps an object or a morphism u of K to the tuple $< F_i \cdot u \,|\, i \in I >$ in $\prod_i L_i$.

Continuity of $< F_i \,|\, i \in I >$ is important for solving systems of equations. In Lemma 2.4.7 it is related to continuity of the F_i. The proof is based on relations with the continuity of $\prod_i F_i$, stated in Lemma 2.4.6.

Definition 2.4.1 [product of functors].

Let I be a set;
$K = \{K_i \,|\, i \in I\}$ and $L = \{L_i \,|\, i \in I\}$ collections of categories;
$F = \{F_i \ : \ K_i \to L_i \,|\, i \in I\}$ a collection of functors.

The product functor $\Pi\, F$ is given by

- $\Pi\, F \ : \ \Pi\, K \to \Pi\, L$

- $(\Pi\, F) \cdot < u_i \,|\, i \in I > = < F_i \cdot u_i \,|\, i \in I >$,
  where $< u_i \,|\, i \in I >$ is either an object or a morphism.

□

Definition 2.4.2 [tupling of functors].

Let I be a set;
K a category;
$L = \{L_i \,|\, i \in I\}$ a collection of categories;
$F = \{F_i \,:\, K \to L_i \,|\, i \in I\}$ a collection of functors.

The functor tuple $< F_i \,|\, i \in I >$ is given by

- $< F_i \,|\, i \in I > :\ K \to \Pi\, L$

- $< F_i \,|\, i \in I >\ u = < F_i \cdot u \,|\, i \in I >$,
  where u is either an object or a morphism.

□

Definition 2.4.3 [diagonal functor].

Let I be a set;
Ka category.

The diagonal functor $\Delta_{K,I}$ is given by

- $\Delta_{K,I} \,:\, K \to \prod_i K$

- $\Delta_{K,I} \cdot u = < u \,|\, i \in I >$,
  where u is either an object or a morphism.

□

Lemma 2.4.4.

Let I be a set;
K a category;
$L = \{L_i \,|\, i \in I\}$ a collection of categories;
$F = \{F_i \,:\, K \to L_i \,|\, i \in I\}$ a collection of functors.
Then
$< F_i \,|\, i \in I > = \Pi\, F \circ \Delta_{K,I}$.

Proof.
Let u be an object or a morphism in K.
$< F_i \,|\, i \in I > \cdot u = < F_i \cdot u \,|\, i \in I > = \Pi\, F \cdot < u \,|\, i \in I > = (\Pi\, F \circ \Delta_{K,I}) \cdot u$.
□

Lemma 2.4.5.
$\Delta_{K,I}$ is ω-cocontinuous.

Proof.

Let I be a set;
K a category;
$< (D_j, f_j) \,|\, j >$ and ω-chain in K;

1. $(A, <\alpha_j \mid j>)$ a colimit for $<(D_j, f_j) \mid j>$.

Then

2. $(<A \mid i \in I>, <<\alpha_j \mid i \in I>, j>)$ is a colimit for $<(<D_j \mid i \in I>, <f_j \mid i \in I>) \mid j>$.
[1, Lemma 2.3.3(4)]

3. $(\Delta_{K,I} A, <\Delta_{K,I}\alpha_j \mid j>)$ is a colimit for $<(\Delta_{K,I} D_j, \Delta_{K,I} f_j) \mid j>$.
[2, Def. 2.4.3]

□

Lemma 2.4.6.

Let I be a set;
$K = \{K_i \mid i \in I\}$ and $L = \{L_i \mid i \in I\}$ collections of categories;
$F = \{F_i : K_i \rightarrow L_i \mid i \in I\}$ a collection of functors.
Then
$(\forall i : F_i$ is ω-cocontinuous$) \Leftrightarrow \Pi F$ is ω-cocontinuous.

Proof.

$(\forall i : F_i$ is ω-cocontinuous$)$
= [def. continuity]
$(\forall i : (\forall (B_i, <\beta_{ij} \mid j>) \in$ colimits of ω-chain $<(D_{ij}, f_{ij}) \mid j>$
$: (F_i \cdot B_i, <F_i \cdot \beta_{ij} \mid j>) \in$ colimits of $<(F_i \cdot D_{ij}, F_i \cdot f_{ij}) \mid j>$
$)$
$)$
= [Eq. 1]
$(\forall (<B_i \mid i>, <<\beta_{ij} \mid i> \mid j>) \in$ colimits of $<(<D_{ij} \mid i>, <f_{ij} \mid i>) \mid j>$
$: (\forall i : (F_i \cdot B_i, <F_i \cdot \beta_{ij} \mid j> \in$ colimits of $<(F_i \cdot D_{ij}, F_i \cdot f_{ij}) \mid j>$
$)$
$)$
= [Lemma 2.3.3(4)]
$(\forall (<B_i \mid i>, <<\beta_{ij} \mid i> \mid j>) \in$ colimits of $<(<D_{ij} \mid i>, <f_{ij} \mid i>) \mid j>$
$: (<F_i \cdot B_i \mid i>, <<F_i \cdot \beta_{ij} \mid i> \mid j>) \in$ colimits of
$<(<F_i \cdot D_{ij} \mid i>, <F_i \cdot f_{ij} \mid i>) \mid j>$
$)$
= [def. 2.4.1]
$(\forall (<B_i \mid i>, <<\beta_i j \mid i> \mid j>) \in$ colimits of $<(<D_{ij} \mid i>, <f_{ij} \mid i>) \mid j>$
$(\Pi F \cdot <B_i \mid i>, <\Pi F \cdot <\beta_{ij} \mid i> \mid j>) \in$ colimits of
$<(\Pi F \cdot <D_{ij} \mid i>, \pi F \cdot <f_{ij} \mid i>) \mid j>$
$)$
= [def. continuity]
πF is ω-cocontinuous.

□

Lemma 2.4.7.

Let I be a set;
K a category;
$L = \{L_i \mid i \in I\}$ a collection of categories;
$F = \{F_i : K \to L_i \mid i \in I\}$ a collection of functors.
Then
$(\forall i : F_i$ is ω-cocontinuous$) \Leftrightarrow < F_i \mid i \in I >$ is ω-cocontinuous.

Proof.
Immediately from Lemmas 2.4.4, 2.4.5, 2.4.6 and the fact that composition preserves ω-cocontinuity.
□

2.5. The initial fixed point theorem for product categories

In the introduction we set out to solve a system of equations

$$\begin{array}{rcl} X_1 & \cong & F_1(X_1, \cdots, X_i, \cdots) \\ & \vdots & \\ X_i & \cong & F_i(X_1, \cdots, X_i, \cdots) \\ & \vdots & \end{array}$$

Using the notions an notations of the previous sections we can reformulate the system as

$$< X_i \mid i \in I > \cong < F_i \mid i \in I > \cdot < X_i \mid i \in I >$$

for a suitably chosen index set I. So $< X_i \mid i \in I >$ should be a fixed point of the endofunctor $< F_i \mid i \in I >$ on the category $\prod_i K_i$. This equation can be solved by means of the initial fixed point construction described in [SP,LS,BH] provided $\prod_i K_i$ is an ω- category and $< F_i \mid i \in I >$ is an ω-cocontinuous functor. By means of Lemmas 2.3.3.(5) and 2.4.7 these requirements can be reduced to requirements for the categories K_i and the functors F_i. The result is stated formally in Theorem 2.5.1.

Theorem 2.5.1 [initial fixed point theorem for product categories].

Let I be a set;
For all $i \in I$: Let K_i be an ω-category;
$F_i : \prod_i K_i \to K_i$ an ω-continuous functor;
U_i an initial object of K_i;
u_i the unique arrow from U_i to $F_i \cdot < U_i \mid i \in I >$.

Let $F = < F_i \mid i \in I >$;
$U = < U_i \mid i \in I >$;
$u = < u_i \mid i \in I >$;

Let $(< A_i \mid i >, < \alpha_j \mid j >)$ be a colimit of the ω-chain

$<(F^j \cdot U, F^j \cdot u) \,|\, j>$.

Let $< \Phi_i \,|\, i >$ be the mediating morphism from
$(< A_i \,|\, i >, < \alpha_j \,|\, j >)$ to $(F \cdot < A_i \,|\, i >, < F \cdot \alpha_j \,|\, j >)$.

Then $(< A_i \,|\, i >, < \Phi_i \,|\, i >^{-1})$ is an initial fixed point of $< F_i \,|\, i >$, i.e.

$$< \Phi_i, i > : < A_i \,|\, i > \rightarrow < F_i, i > \cdot < A_i, i > \text{ is an isomorphism}$$

and consequently,

$$\text{for all } i \in I \; : \; \Phi_i \; : \; A_i \rightarrow F_i \cdot < A_i \,|\, i > \text{ is an isomorphism .}$$

Proof.

1. $\prod_i K_i$ is an ω-category. [Lemma 2.3.3.(5)]
2. U is initial in $\prod_i K_i$. [Lemma 2.3.3.(1)]
3. u is the unique arrow from U to $F \cdot U$. [Lemma 2.3.3.(1)]
4. F is ω-cocontinuous. [Lemma 2.4.7]
5. $(< A_i \,|\, i >, < \Phi_i \,|\, i >^{-1})$ is i.f.p. of F. [1-4, i.f.p. theorem]

□

3. COMPLETENESS OF PRODUCT CATEGORIES

Let I be an arbitrary set. The discrete category J corresponding to I is defined by

i) $\mathrm{obj}(J) = I$.

ii) $\mathrm{Hom}_J(i,j) = \begin{cases} \emptyset & \text{if } i \neq j \\ \{id_i\} & \text{if } i = j \end{cases}$.

Let K be an arbitrary category.

Lemma 3.1.
The categories K^J and $\prod_{i \in I} K$ are isomorphic.

Proof.
An object of the functor category K^J is a functor $F : J \to K$. Since the only morphisms of J are identities, the functor F is completely defined by a mapping $F' : \mathrm{obj}(J) = I \to \mathrm{obj}(K)$. F' can also be considered as an element of $\prod_{i \in I} \mathrm{obj}(K) = \mathrm{obj}\Big(\prod_{i \in I} K\Big)$.

A morphism $\eta : F_1 \to F_2$ of the functor category K^J is a natural transformation between the functors F_1 and F_2. This means that η is a family of morphisms $(\eta_i)_{i \in \mathrm{obj}(J)}$, such that for each morphism $\xi : i \to j$ of the category J the following diagram commutes.

$$\begin{array}{ccccc}
i & \quad & F_1(i) & \xrightarrow[\eta_i]{} & F_2(i) \\
\downarrow \xi & & \downarrow F_1(\xi) & \emptyset & \downarrow F_2(\xi) \\
j & & F_1(j) & \xrightarrow[\eta_j]{} & F_2(j)
\end{array}$$

Since $\xi : i \to j$ is only possible if $i = j$ and $\xi = id_i$, the condition on the family $(\eta_i)_{i \in \mathrm{obj}(J)}$ reduces to: for all $i \in \mathrm{obj}(J)$ η_i is a morphism $F_1(i) \to F_2(i)$ in the category K. Hence, η can be seen as a morphism of $\prod_{i \in I} K$.

After these preliminaries it is clear that we can define a functor $H : K^J \to \prod_{i \in I} K$ by

$$H(F) = < F(i) \,|\, i \in I > \ , \quad \text{for } F \in \mathrm{obj}(K^J);$$
$$H(\eta) = < \eta_i \,|\, i \in I > \ , \quad \text{for } \eta \in \mathrm{mor}(K^J).$$

It is easily proved that H is a functor, and that its inverse $H^{-1} : \prod_{i \in I} K \to K^J$ exists.

□

A category is called ω-(co)complete if every ω-chain has a (co)limit.
As a consequence of this lemma we now have

Lemma 3.2.
If a category K is ω-(co)complete, then also $\prod_{i \in I} K$ is ω- (co)complete.

Proof.
If K is ω-(co)complete, then $K^{\mathcal{A}}$ is ω- (co)complete for every category $\mathcal{A}$, see for instance Herrlich & Strecker [HS, §25.7]. The result now follows from Lemma 1 and the remark that if a category is ω-(co)complete, the same holds for isomorphic categories.

□

4. LOCAL CONTINUITY OF GENERALIZED PRODUCT AND SUM FUNCTORS

4.1. Introduction

In this section we define a generalized sum functor and a generalized product functor. Moreover, we give a detailed proof of the local continuity of these functors. Recall that (see for instance Bos & Hemerik [BH] or Smyth & Plotkin [SP]), if K and L are O-categories, K is localized and $F : K \to L$ is a locally continuous functor, the corresponding functor $F_{\mathrm{PR}} : K_{\mathrm{PR}} \to L_{\mathrm{PR}}$ is ω- continuous.

4.2. Definition of the generalized sum and generalized product functors

Let A be some nonempty (index) set. In this note we shall frequently work with objects and morphisms of the category $\prod_{a\in A} \underline{\mathrm{CPO}}$. Objects of $\prod_{a\in A} \underline{\mathrm{CPO}}$ are tuples $< D_a \,|\, a \in A >$, where each D_a is an object of $\underline{\mathrm{CPO}}$. The set of morphisms between the objects $< D_a \,|\, a \in A >$ and $< E_a \,|\, a \in A >$ consists of tuples $< m_a \,|\, a \in A >$, where each m_a is a morphism in the category $\underline{\mathrm{CPO}}$ between D_a and E_a.

The generalized sum functor $\widehat{\mathrm{GS}} : \prod_{a\in A} \underline{\mathrm{CPO}} \to \underline{\mathrm{CPO}}$ is defined in the following way.

i) For an object $< D_a \,|\, a \in A >$ of $\prod_{a\in A} \underline{\mathrm{CPO}}$

$$\widehat{\mathrm{GS}}(< D_a \,|\, a \in A >) = \sum_{a\in A} D_a \in \mathrm{obj}(\underline{\mathrm{CPO}}) \,.$$

Here $\sum_{a\in A} D_a$ is the c.p.o. which consists of the disjoint sum of the c.p.o.'s D_a.

Elements of $\sum_{a\in A} D_a$ which are different from $\bot_{\Sigma D_a}$, are of the form $< b, d_b >$, with $b \in A$ and $d_b \in D_b$.

ii) For a morphism $< m_a \,|\, a \in A > : < D_a \,|\, a \in A > \to < E_a \,|\, a \in A >$ of $\prod_{a\in A} \underline{\mathrm{CPO}}$

$$\widehat{\mathrm{GS}}(< m_a \,|\, a \in A >) = \mathop{\mathrm{Sum}}_{a\in A} m_a \in \mathrm{mor}(\underline{\mathrm{CPO}}) \,.$$

Here $\mathop{\mathrm{Sum}}_{a\in A} m_a$ is the morphism between $\sum_{a\in A} D_a$ and $\sum_{a\in A} E_a$ defined by:

$$\Big(\mathop{\mathrm{Sum}}_{a\in A} m_a\Big) \bot_{\Sigma D_a} = \bot_{\Sigma E_a} \,,$$

$$\Big(\mathop{\mathrm{Sum}}_{a\in A} m_a\Big) < b, d_b > = < b, m_b(d_b) > \,.$$

It is easily verified that $\widehat{\mathrm{GS}}$ is indeed a functor. Next we define the generalized product functor $\widehat{\mathrm{GP}} : \prod_{a \in A} \underline{\mathrm{CPO}} \to \underline{\mathrm{CPO}}$.

i) For an object $< D_a \,|\, a \in A >$ of $\prod_{a \in A} \underline{\mathrm{CPO}}$

$\widehat{\mathrm{GP}}(< D_a \,|\, a \in A >) = \prod_{a \in A} D_a$.

Here $\prod_{a \in A} D_a$ is the c.p.o. which is the Cartesian product of the c.p.o.'s D_a.

Elements of $\prod_{a \in A} D_a$ are tuples $< d_a \,|\, a \in A >$ where $d_a \in D_a$ for all $a \in A$.

ii) For a morphism $< m_a \,|\, a \in A > : < D_a \,|\, a \in A > \to < E_a \,|\, a \in A >$ of $\prod_{a \in A} \underline{\mathrm{CPO}}$

$\widehat{\mathrm{GP}}(< m_a \,|\, a \in A >) = \mathop{\mathrm{Prod}}_{a \in A} m_a \in \mathrm{mor}(\underline{\mathrm{CPO}})$.

Here $\mathop{\mathrm{Prod}}_{a \in A} m_a$ is the morphism between $\prod_{a \in A} D_a$ and $\prod_{a \in A} E_a$ defined by:

$\left(\mathop{\mathrm{Prod}}_{a \in A} m_a\right)(< d_a \,|\, a \in A >) = < m_a(d_a) \,|\, a \in A >$.

It is also easily seen that $\widehat{\mathrm{GP}}$ is a functor.

4.3. Technical results

Recall (see for instance Bos & Hemerik [BH] or Smyth & Plotkin [SP]) that an O-category is a category such that

i) every hom set is a poset in which every ω-chain has a lub,

ii) composition of morphisms is ω-continuous.

For the category $\underline{\mathrm{CPO}}$ the hom sets have a natural c.p.o. structure that satisfies i) and ii). For the category $\prod_{a \in A} \underline{\mathrm{CPO}}$ the hom sets consist of tuples of continuous mappings, which by the componentwise ordering, also have a c.p.o. structure satisfying i) and ii). More precisely, if $m_i = < m_{i,a} \,|\, a \in A >$ $(i \in \mathbb{N})$ is an ω-chain in $\mathrm{Hom}_{\Pi \underline{\mathrm{CPO}}}(< D_a \,|\, a \in A >, < E_a \,|\, a \in A >)$, then for all $a \in A$ $m_{i,a}$ is an ω-chain in $\mathrm{Hom}_{\underline{\mathrm{CPO}}}(D_a, E_a)$. The lub m of the ω-chain m_i is then given by

$$m = < m_a \,|\, a \in A > = < \bigsqcup_{i=0}^{\infty} m_{i,a} \,|\, a \in A > . \tag{4.3.1}$$

Next we derive some properties of the mappings Prod and Sum, as defined in Section 4.2.

Let $D = < D_a \,|\, a \in A >$ and $E = < E_a \,|\, a \in A >$ be two objects of $\prod_{a \in A} \underline{\text{CPO}}$. Let $m_i = < m_{i,a} \,|\, a \in A >$ be an ω-chain in $\text{Hom}(D, E)$ with lub $m = < m_a \,|\, a \in A >$. Suppose $b \in A$, $d_b \in D_b$ and so $d = < b, d_b > \in \sum_a D_a$.

Then

$$
\begin{aligned}
& \bigsqcup_{i=0}^{\infty} \Big(\underset{a}{\text{Sum}}\ m_{i,a})(d)\Big) \\
&= \bigsqcup_{i=0}^{\infty} \Big(< b, m_{i,b}(d_b) >\Big) && \text{[def. Sum]} \\
&= < b, \bigsqcup_{i=0}^{\infty} m_{i,b}(d_b) > && \text{[lubs in } \textstyle\sum E_a \text{ are computed in a component } (E_b)] \\
&= < b, \Big(\bigsqcup_{i=0}^{\infty} m_{i,b}\Big)(d_b) > && \text{[lubs of functions are computed pointwise]} \\
&= < b, m_b(d_b) > && [(4.3.1)] \\
&= \Big(\underset{a}{\text{Sum}}\ m_a\Big)(d) && \text{[def. Sum]}
\end{aligned}
$$

Also

$$
\begin{aligned}
& \bigsqcup_{i=0}^{\infty} \Big((\underset{a}{\text{Sum}}\ m_{i,a})(\perp_{\Sigma D_a})\Big) \\
&= \bigsqcup_{i=0}^{\infty} \perp_{\Sigma E_a} && \text{[def. Sum]} \\
&= \perp_{\Sigma E_a} \\
&= \Big(\underset{a}{\text{Sum}}\ m_a\Big) \perp_{D_a} && \text{[def. Sum]}
\end{aligned}
$$

From these two computations and the fact that lubs of functions are computed pointwise (i.e.: $\Big(\bigsqcup_{i=0}^{\infty} \underset{a}{\text{Sum}}\ m_{i,a}\Big)(d) = \bigsqcup_{i=0}^{\infty} \Big((\underset{a}{\text{Sum}}\ m_{i,a})(d)\Big)$) we conclude that

$$
\bigsqcup_{i=0}^{\infty} \underset{a}{\text{Sum}}\ m_{i,a} = \underset{a}{\text{Sum}}\ m_a \,. \tag{4.3.2}
$$

A similar result holds for "Prod". Let $d = < d_a \,|\, a \in A > \in \prod_{a \in A} D_a$. Then

$$\bigsqcup_{i=0}^{\infty} \Big(\big(\operatorname*{Prod}_{a} \ m_{i,a}\big)(d)\Big)$$

$$= \bigsqcup_{i=0}^{\infty} \ (< m_{i,a}(d_a) \mid a \in A >) \qquad \text{[def. Prod]}$$

$$= < \bigsqcup_{i=0}^{\infty} \ (m_{i,a}(d_a)) \mid a \in A > \qquad \text{[lubs in product c.p.o.'s are computed componentwise]}$$

$$= < \Big(\bigsqcup_{i=0}^{\infty} \ m_{i,a}\Big)(d_a) \mid a \in A > \qquad \text{[lubs of functions are computed pointwise]}$$

$$= < m_a(d_a) \mid a \in A > \qquad \text{[(4.3.1)]}$$

$$= \Big(\operatorname*{Prod}_{a} \ m_a\Big)(d) \qquad \text{[def. Prod]}$$

Together with the fact that lubs of functions are computed pointwise

(i.e. $\Big(\bigsqcup_{i=0}^{\infty} \ \operatorname*{Prod}_{a} \ m_{i,a}\Big)(d) = \bigsqcup_{i=0}^{\infty} \ \Big(\big(\operatorname*{Prod}_{a} \ m_{i,a}\big)(d)\Big)$), this implies

$$\bigsqcup_{i=0}^{\infty} \ \operatorname*{Prod}_{a} \ m_{i,a} = \operatorname*{Prod}_{a} \ m_a \ . \tag{4.3.3}$$

4.4. Local continuity

A functor $F : \prod_{a \in A} \underline{\mathrm{CPO}} \to \underline{\mathrm{CPO}}$ is locally continuous if for all objects D, E of $\prod_{a \in A} \underline{\mathrm{CPO}}$, F, viewed as a map : $\mathrm{Hom}_{\Pi \underline{\mathrm{CPO}}}(D,E) \to \mathrm{Hom}_{\underline{\mathrm{CPO}}}(F(D),F(E))$ is ω- continuous. Note that this definition is only useful since $\prod_{a \in A} \underline{\mathrm{CPO}}$ and $\underline{\mathrm{CPO}}$ are O-categories. Using the results of Section 3 the following two theorems can easily be proved.

Theorem 4.4.1.
The generalized sum functor $\widehat{\mathrm{GS}}$ is locally continuous.

Proof.
Let D and E be objects of $\prod_{a \in A} \underline{\mathrm{CPO}}$ and let $m_i =< m_{i,a} \mid a \in A >$ be an ω-chain in $\mathrm{Hom}_{\Pi \underline{\mathrm{CPO}}}(D,E)$ with lub $m = \bigsqcup_{i} \ m_i$.

Then

$$\bigsqcup_{i=0}^{\infty} \widehat{\mathrm{GS}}(m_i)$$

$$= \bigsqcup_{i=0}^{\infty} \left(\operatorname*{Sum}_{a} \; m_{i,a} \right) \qquad [\text{def. } \widehat{\mathrm{GS}}]$$

$$= \operatorname*{Sum}_{a} \; m_a \qquad [(4.3.2)]$$

$$= \widehat{\mathrm{GS}}(m) \qquad [\text{def. } \widehat{\mathrm{GS}}]$$

□

Theorem 4.4.2.
The generalized product functor $\widehat{\mathrm{GP}}$ is locally continuous.

Proof.
In the same setting as in the proof of the previous theorem we have

$$\bigsqcup_{i=0}^{\infty} \widehat{\mathrm{GP}}(m_i)$$

$$= \bigsqcup_{i=0}^{\infty} \left(\operatorname*{Prod}_{a} \; m_{i,a} \right) \qquad [\text{def. } \widehat{\mathrm{GP}}]$$

$$= \operatorname*{Prod}_{a} \; m_a \qquad [(4.3.3)]$$

$$= \widehat{\mathrm{GP}}(m) \qquad [\text{def. } \widehat{\mathrm{GP}}]$$

□

5. ω-CONTINUITY OF GENERALIZED PRODUCT AND SUM FUNCTORS

5.1. Introduction

In Section 4 we described generalized sum and product functors $\widehat{GS}, \widehat{GP} : \prod_{a \in A} \underline{CPO} \rightarrow \underline{CPO}$ and proved their local continuity. Since local continuity implies local monotonicity, there exist functors

$$\widehat{GS}_{PR}, \widehat{GP}_{PR} : \Big(\prod_{a\in A} \underline{CPO} \Big)_{PR} \rightarrow \underline{CPO}_{PR} .$$

In this section we show how these functors can be used to construct ω-continuous functors

$$GS, GP : \prod_{a\in A} (\underline{CPO}_{PR}) \rightarrow \underline{CPO}_{PR} .$$

First we show that the categories $\Big(\prod_{a\in A} \underline{CPO} \Big)_{PR}$ and $\prod_{a\in A} (\underline{CPO}_{PR})$ are isomorphic.

Since for an arbitrary O-category K $\text{obj}(K_{PR}) = \text{obj}(K)$, the objects of both categories are identical.

Next suppose $D =< D_a \,|\, a \in A >$ and $E =< E_a \,|\, a \in A >$ are objects of $\Big(\prod_{a\in A} \underline{CPO} \Big)_{PR}$ and suppose $m \in \text{Hom}_{(\prod_{a\in A} \underline{CPO})_{PR}}(D, E)$. This means that m is a projection pair (m^L, m^R), where

$$m^L \in \text{Hom}_{\prod_{a\in A} \underline{CPO}}(D, E) ,$$

$$m^R \in \text{Hom}_{\prod_{a\in A} \underline{CPO}}(E, D)$$

and

$$m^L \circ m^R \sqsubseteq id_E ,$$
$$m^R \circ m^L = id_D .$$

Because $\prod_{a\in A} \underline{CPO}$ is a product category, m^L and m^R consist of tuples:

$$m^L = < m^L_a \,|\, a \in A > ,$$
$$m^R = < m^R_a \,|\, a \in A > ,$$

where

$$m_a^L \in \mathrm{Hom}_{\underline{\mathrm{CPO}}}(D_a, E_a) \ ,$$
$$m_a^R \in \mathrm{Hom}_{\underline{\mathrm{CPO}}}(E_a, D_a)$$

and

$$m_a^L \circ m_a^R \sqsubseteq id_{E_a} \ ,$$
$$m_a^R \circ m_a^L = id_{D_a} \ .$$

Hence,

$$(m_a^L, m_a^R) \in \mathrm{Hom}_{\underline{\mathrm{CPO}}_{\mathrm{PR}}}(D_a, E_a)$$

and

$$< (m_a^L, m_a^R) \,|\, a \in A > \in \mathrm{Hom}_{\prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}})}(D, E) \ .$$

The above implies that we can define a functor $S : \left(\prod_{a \in A} \underline{\mathrm{CPO}} \right)_{\mathrm{PR}} \to \prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}})$ in the following way

i) for an object D of $\left(\prod_{a \in A} \underline{\mathrm{CPO}} \right)_{\mathrm{PR}}$

$S(D) = D.$

ii) for a morphism $m = (< m_a^L \,|\, a \in A >, < m_a^R \,|\, a \in A >)$ of $\left(\prod_{a \in A} \underline{\mathrm{CPO}} \right)_{\mathrm{PR}}$

$S(m) = \ < (m_a^L, m_a^R) \,|\, a \in A >.$

It is easily seen that S is a functor. Moreover, S is an isomorphism, its inverse is the functor $T : \prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}}) \to \left(\prod_{a \in A} \underline{\mathrm{CPO}} \right)_{\mathrm{PR}}$, defined by

i) for an object D of $\prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}})$

$T(D) = D.$

ii) for a morphism $m = < (m_a^L, m_a^R) \,|\, a \in A >$
$T(m) = (< m_a^L \,|\, a \in A >, < m_a^R \,|\, a \in A >).$

The generalized sum and product functors on the PR categories are defined by

$$\mathrm{GS} = \widehat{\mathrm{GS}}_{\mathrm{PR}} \circ T \ : \ \prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}}) \to \underline{\mathrm{CPO}}_{\mathrm{PR}} \ ; \tag{5.1.1}$$

$$\mathrm{GP} = \widehat{\mathrm{GP}}_{\mathrm{PR}} \circ T \ : \ \prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}}) \to \underline{\mathrm{CPO}}_{\mathrm{PR}} \ . \tag{5.1.2}$$

For an object $D =< D_a \,|\, a \in A >$ of $\prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}})$ this means

$$\mathrm{GS}(D) = \widehat{\mathrm{GS}}_{\mathrm{PR}}(T(D)) = \widehat{\mathrm{GS}}_{\mathrm{PR}}(D) = \widehat{\mathrm{GS}}(D) = \sum_a D_a \ ,$$

$$\mathrm{GP}(D) = \widehat{\mathrm{GP}}_{\mathrm{PR}}(T(D)) = \widehat{\mathrm{GP}}_{\mathrm{PR}}(D) = \widehat{\mathrm{GP}}(D) = \prod_a D_a \ .$$

For a morphism $m =< (m_a^L, m_a^R) \,|\, a \in A >$ of $\prod_{a \in A} (\underline{\mathrm{CPO}}_{\mathrm{PR}})$ this means

$$\begin{aligned} \mathrm{GS}(m) &= \widehat{\mathrm{GS}}_{\mathrm{PR}}(T(m)) = \widehat{\mathrm{GS}}_{\mathrm{PR}}\Big((< m_a^L \,|\, a \in A >, < m_a^R \,|\, a \in A >)\Big) \\ &= \Big(\widehat{\mathrm{GS}}(< m_a^L \,|\, a \in A >), \widehat{\mathrm{GS}}(< m_a^R \,|\, a \in A >)\Big) \\ &= \Big(\operatorname*{Sum}_{a \in A} \ m_a^L, \ \operatorname*{Sum}_{a \in A} \ m_a^R \Big) \ , \end{aligned}$$

$$\begin{aligned} \mathrm{GP}(m) &= \widehat{\mathrm{GP}}_{\mathrm{PR}}(T(m)) = \widehat{\mathrm{GP}}_{\mathrm{PR}}\Big((< m_a^L \,|\, a \in A >, < m_a^R \,|\, a \in A >)\Big) \\ &= \Big(\widehat{\mathrm{GP}}(< m_a^L \,|\, a \in A >), \widehat{\mathrm{GP}}(< m_a^R \,|\, a \in A >)\Big) \\ &= \Big(\operatorname*{Prod}_{a \in A} \ m_a^L, \ \operatorname*{Prod}_{a \in A} \ m_a^R \Big) \ . \end{aligned}$$

For the definitions of Prod and Sum, see Section 4.

5.2. ω-Continuity

The functor T is an isomorphism, so it is ω-continuous. Hence, to prove the ω-continuity of GS and GP, it is sufficient to show that $\widehat{\mathrm{GS}}_{\mathrm{PR}}$ and $\widehat{\mathrm{GP}}_{\mathrm{PR}}$ are ω- continuous.
Recall that $\widehat{\mathrm{GS}}_{\mathrm{PR}}$ and $\widehat{\mathrm{GP}}_{\mathrm{PR}}$ are "PR versions" of the locally continuous functors (see Section 4)

$$\widehat{\mathrm{GS}}, \widehat{\mathrm{GP}} \ : \ \prod_{a \in A} \ \underline{\mathrm{CPO}} \to \underline{\mathrm{CPO}} \ .$$

From the continuity theorem (see for instance Bos & Hemerik [BH, Th. 3.15]) it follows that it remains to be shown that $\prod_{a \in A} \underline{\mathrm{CPO}}$ is localized. In [BH, Prop. 3.18] it is shown that an O-category is localized if every idempotent is split.

Theorem 5.2.1.
Let K be an O-category in which every idempotent is split. Then every idempotent in the O-category $\prod_{a\in A} K$ is split.

Proof.
Let $f \in \mathrm{Hom}_{\prod K}(D,D)$ be such that $f \circ f = f$. Then for all components $f_a \in \mathrm{Hom}_K(D_a, D_a)$ we have $f_a \circ f_a = f_a$. Since every idempotent in K is split, this means that there exists an object E_a of K and morphisms $g_a : D_a \to E_a$ and $h_a : E_a \to D_a$ such that $f_a = h_a \circ g_a$ and $g_a \circ h_a = id_{E_a}$. Let $D = < D_a \,|\, a \in A > \in \mathrm{obj}\Big(\prod_{a\in A} K\Big)$, and let $h = < h_a \,|\, a \in A > : E \to D$ and $g = < g_a \,|\, a \in A > : D \to E$. Then clearly $f = h \circ g$ and $g \circ h = id_E$, so f is split in $\prod_{a\in A} K$.

□

In the proof of Theorem 3.2.2 in [BH], it is shown that every idempotent in CPO is split. Hence, every idempotent in $\prod_{a\in A}$ CPO is split, which implies (see [BH, Prop. 3.18]) that $\prod_{a\in A}$ CPO is a localized O-category. The discussion at the beginning of this section now yields:

Theorem 5.2.2.
The functions GS,GP : $\prod_{a\in A}$ $(\mathrm{CPO_{PR}}) \to \mathrm{CPO_{PR}}$, as given in (5.1.1) and (5.1.2) are ω-continuous.
□

6. REFERENCES

BH Bos, R. and Hemerik, C., "An introduction to the category theoretical solution of recursive domain equations", Computing Science Note 88/15, Eindhoven University of Technology.

BMM Bruce, K.B., Meyer, A.R. and Mitchell, J.C., "The semantics of second order lambda calculus", to appear in Information and Computation.

HTE Hemerik, C. and Ten Eikelder, H.M.M., "A model for a second order lambda calculus with recursion", to appear.

HS Herrlich, H. and Strecker, G.E., "Category Theory", Allyn and Bacon, Inc., Boston (1973).

LS Lehmann, D.J. and Smyth, M.B., "Algebraic specification of data types: a synthetic approach", Math. Syst. Theory 14, 97–139 (1981).

SP Smyth, M.B. and Plotkin, G.D., "The category theoretic solution of recursive domain equations", SIAM J. Comput. 11, 761–783 (1982).

TEM Ten Eikelder, H.M.M. and Mak, R.H., "Language Theory of a λ-calculus with Recursive Types", Computing Science Note 88/14, Eindhoven University of Technology.

In this series appeared :

No.	Author(s)	Title
85/01	R.H. Mak	The formal specification and derivation of CMOS-circuits
85/02	W.M.C.J. van Overveld	On arithmetic operations with M-out-of-N-codes
85/03	W.J.M. Lemmens	Use of a computer for evaluation of flow films
85/04	T. Verhoeff H.M.J.L. Schols	Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate
86/01	R. Koymans	Specifying message passing and real-time systems
86/02	G.A. Bussing K.M. van Hee M. Voorhoeve	ELISA, A language for formal specifications of information systems
86/03	Rob Hoogerwoord	Some reflections on the implementation of trace structures
86/04	G.J. Houben J. Paredaens K.M. van Hee	The partition of an information system in several parallel systems
86/05	Jan L.G. Dietz Kees M. van Hee	A framework for the conceptual modeling of discrete dynamic systems
86/06	Tom Verhoeff	Nondeterminism and divergence created by concealment in CSP
86/07	R. Gerth L. Shira	On proving communication closedness of distributed layers
86/08	R. Koymans R.K. Shyamasundar W.P. de Roever R. Gerth S. Arun Kumar	Compositional semantics for real-time distributed computing (Inf.&Control 1987)
86/09	C. Huizing R. Gerth W.P. de Roever	Full abstraction of a real-time denotational semantics for an OCCAM-like language
86/10	J. Hooman	A compositional proof theory for real-time distributed message passing
86/11	W.P. de Roever	Questions to Robin Milner - A responder's commentary (IFIP86)
86/12	A. Boucher R. Gerth	A timed failures model for extended communicating processes

86/13	R. Gerth W.P. de Roever	Proving monitors revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)
86/14	R. Koymans	Specifying passing systems requires extending temporal logic
87/01	R. Gerth	On the existence of sound and complete axiomatizations of the monitor concept
87/02	Simon J. Klaver Chris F.M. Verberne	Federatieve Databases
87/03	G.J. Houben J.Paredaens	A formal approach to distri-buted information systems
87/04	T.Verhoeff	Delay-insensitive codes - An overview
87/05	R.Kuiper	Enforcing non-determinism via linear time temporal logic specification.
87/06	R.Koymans	Temporele logica specificatie van message passing en real-time systemen (in Dutch).
87/07	R.Koymans	Specifying message passing and real-time systems with real-time temporal logic.
87/08	H.M.J.L. Schols	The maximum number of states after projection.
87/09	J. Kalisvaart L.R.A. Kessener W.J.M. Lemmens M.L.P. van Lierop F.J. Peters H.M.M. van de Wetering	Language extensions to study structures for raster graphics.
87/10	T.Verhoeff	Three families of maximally nondeter-ministic automata.
87/11	P.Lemmens	Eldorado ins and outs. Specifications of a data base management toolkit according to the functional model.
87/12	K.M. van Hee and A.Lapinski	OR and AI approaches to decision support systems.
87/13	J.C.S.P. van der Woude	Playing with patterns, searching for strings.
87/14	J. Hooman	A compositional proof system for an occam-like real-time language

87/15	C. Huizing R. Gerth W.P. de Roever	A compositional semantics for statecharts
87/16	H.M.M. ten Eikelder J.C.F. Wilmont	Normal forms for a class of formulas
87/17	K.M. van Hee G.-J.Houben J.L.G. Dietz	Modelling of discrete dynamic systems framework and examples
87/18	C.W.A.M. van Overveld	An integer algorithm for rendering curved surfaces
87/19	A.J.Seebregts	Optimalisering van file allocatie in gedistribueerde database systemen
87/20	G.J. Houben J. Paredaens	The R^2 -Algebra: An extension of an algebra for nested relations
87/21	R. Gerth M. Codish Y. Lichtenstein E. Shapiro	Fully abstract denotational semantics for concurrent PROLOG
88/01	T. Verhoeff	A Parallel Program That Generates the Möbius Sequence
88/02	K.M. van Hee G.J. Houben L.J. Somers M. Voorhoeve	Executable Specification for Information Systems
88/03	T. Verhoeff	Settling a Question about Pythagorean Triples
88/04	G.J. Houben J.Paredaens D.Tahon	The Nested Relational Algebra: A Tool to handle Structured Information
88/05	K.M. van Hee G.J. Houben L.J. Somers M. Voorhoeve	Executable Specifications for Information Systems
88/06	H.M.J.L. Schols	Notes on Delay-Insensitive Communication
88/07	C. Huizing R. Gerth W.P. de Roever	Modelling Statecharts behaviour in a fully abstract way
88/08	K.M. van Hee G.J. Houben L.J. Somers M. Voorhoeve	A Formal model for System Specification
88/09	A.T.M. Aerts K.M. van Hee	A Tutorial for Data Modelling

88/10	J.C. Ebergen	A Formal Approach to Designing Delay Insensitive Circuits
88/11	G.J. Houben J.Paredaens	A graphical interface formalism: specifying nested relational databases
88/12	A.E. Eiben	Abstract theory of planning
88/13	A. Bijlsma	A unified approach to sequences, bags, and trees
88/14	H.M.M. ten Eikelder R.H. Mak	Language theory of a lambda-calculus with recursive types
88/15	R. Bos C. Hemerik	An introduction to the category theoretic solution of recursive domain equations
88/16	C.Hemerik J.P.Katoen	Bottom-up tree acceptors
88/17	K.M. van Hee G.J. Houben L.J. Somers M. Voorhoeve	Executable specifications for discrete event systems
88/18	K.M. van Hee P.M.P. Rambags	Discrete event systems: concepts and basic results.
88/19	D.K. Hammer K.M. van Hee	Fasering en documentatie in software engineering.
88/20	K.M. van Hee L. Somers M.Voorhoeve	EXSPECT, the functional part.
89/1	E.Zs.Lepoeter-Molnar	Reconstruction of a 3-D surface from its normal vectors.
89/2	R.H. Mak P.Struik	A systolic design for dynamic programming.
89/3	H.M.M. Ten Eikelder C. Hemerik	Some category theoretical properties related to a model for a polymorphic lambda-calculus.