Universidad de Málaga

Departamento de Matemática Aplicada

A graphical approach to monad compositions

P. Eklund, M.A. Galán, J. Medina, M. Ojeda-Aciego, A. Valverde

Technical Report MA/00/03

A Graphical Approach to Monad Compositions

P. Eklund [a] M.A. Galán [a,1] J. Medina [b] M. Ojeda-Aciego [b]
A. Valverde [b]

[a] *Dept. Computing Science. Umeå University.*
{peklund,magalan}@cs.umu.se

[b] *Dept. Matemática Aplicada. Universidad de Málaga.*
{jmedina,aciego,a_valverde}@ctima.uma.es

Abstract

Techniques for monad compositions can be used to provide a basis for categorical unification in the framework of generalised terms. In [4], we gave results for the many-valued sets of terms, and showed that this composition of set functors can be extended to a monad. In this work we introduce new sufficient conditions for two monads being composable, and show that the construction in [4] also satisfies these new condition. In addition, we give a theorem of structure of the multiplication of the composite monad, i.e. its structure can be determined under certain conditions.

1 Introduction

Monads have shown to be useful in different fields related to computer science. In functional programming monad compositions are applied to structuring of functional programs [11]. In particular, in functional programs like parsers or type checkers the monad needed is often a composed monad [13]. In logic programming, unification has been identified as the provision of co-equalisers in Kleisli categories of term monads [12].

The foundational understanding of monads has been well-known for decades, but proof techniques, especially related to monad compositions have not been developed. As monad compositions are basically built upon operations of corresponding natural transformations, proof techniques require an adequate han-

[1] Supported by the Swedish Research Council for Engineering Sciences.

dling of the basic combinatorial properties of functors and natural transformations (Godement rules). In [3,6] it was discovered that these combinatorial properties can be represented more visually, in that the basic observation relates to distributivity of the star product of natural transformation with respect to composition of natural transformations.

This improves readability of expressions involving compositions of natural transformations and supports proofs involving more complex properties. This visual technique is not widely known and has been used mainly in purely algebraic contexts [2].

The aim of this paper is to further develop these ideas, and to demonstrate the use of this technique by providing some concrete examples on generalised terms where various set functors are composed with the conventional term functor [4,5]. In particular, we will illuminate the use of this technique by providing results on preservation properties for iterated compositions and subconstructs.

2 Notations and pictorial representations

Let $\mathtt{C}$ be a category and consider (covariant) endofunctors $F, G, H, \ldots : \mathtt{C} \to \mathtt{C}$, together with natural transformations $\tau, \sigma, \ldots : F \to G$ between such endofunctors. For $\tau : F \to G$ and $\sigma : G \to H$, let $\sigma \circ \tau : F \to H$ be the usual composition of natural transformations, and for $\tau' : F' \to G'$, let $\tau' \star \tau : F' \circ F \to G' \circ G$ be the star product given by

$$\tau' \star \tau = \tau' G \circ F' \tau = G' \tau \circ \tau' F. \tag{1}$$

The star product, like composition, is associative.

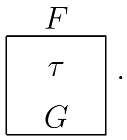
For the identity transformation $id_F : F \to F$, also written as 1_F or 1, note that

$$1_F \star 1_G = 1_{F \circ G}. \tag{2}$$

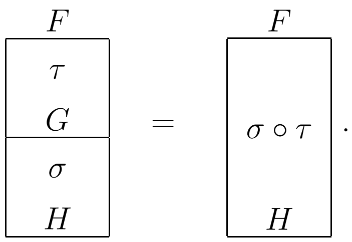
For a natural transformation $\tau : F \to G$, and a functor H, $(H\tau)_X = H\tau_X$ and $(\tau H)_X = \tau_{HX}$, or equivalently, $H\tau = 1_H \star \tau$ and $\tau H = \tau \star 1_H$. The following distributivity laws hold:

$$1 \star (\sigma \circ \tau) = (1 \star \sigma) \circ (1 \star \tau), \tag{3}$$
$$(\sigma \circ \tau) \star 1 = (\sigma \star 1) \circ (\tau \star 1). \tag{4}$$

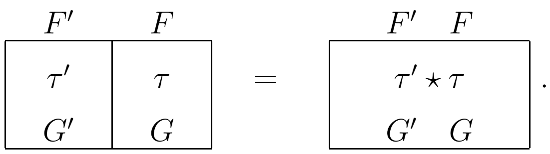
A natural transformation $\tau : F \to G$ as a basic building block is depicted as



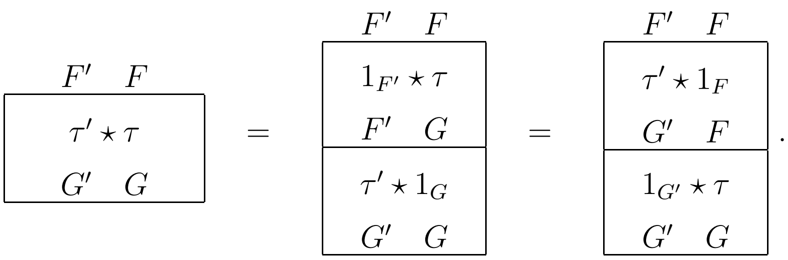
Blocks $\tau : F \to G$ and $\sigma : G \to H$ are built, or composed, vertically as



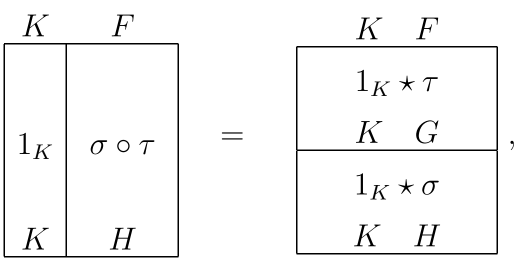
For $\tau' : F' \to G'$, horizontal block building is done as



Note that equation (1) can is pictorially represented by

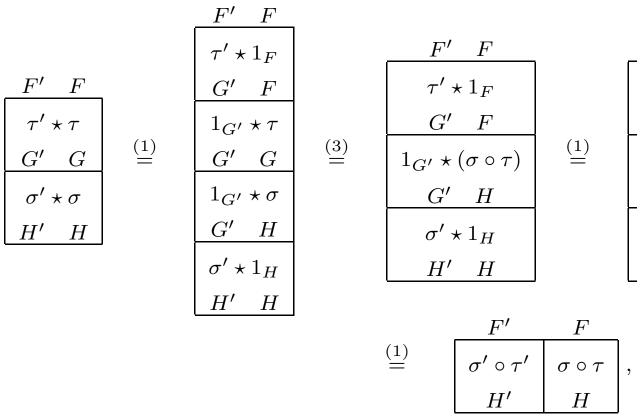


Equation (3) can be written as



i.e., in this case building blocks can be applied in any order. The same holds for equation (4).

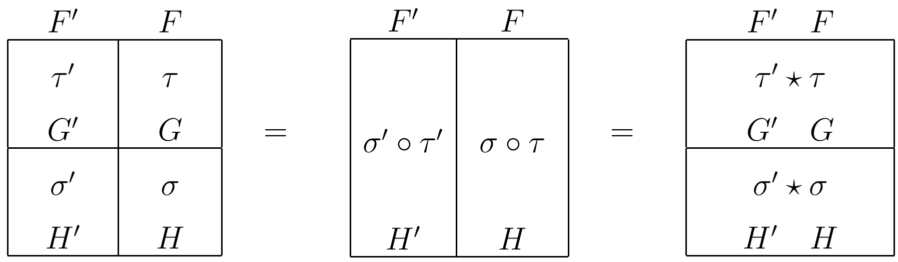
For natural transformations $F \xrightarrow{\tau} G \xrightarrow{\sigma} H$ and $F' \xrightarrow{\tau'} G' \xrightarrow{\sigma'} H'$ we then have



i.e., we have (re)proved the Interchange Law

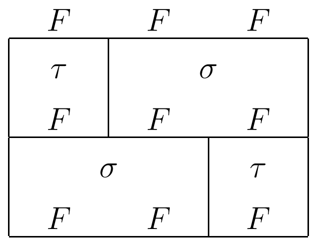
$$(\sigma' \circ \sigma) * (\tau' \circ \tau) = (\sigma' * \tau') \circ (\sigma * \tau) \tag{5}$$

which can be summarized as



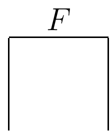
showing how blocks with particular positions generally can be attached vertically and horizontally in any order without changing the resulting transformation.

Note in the transformation



that the composition $(\sigma \star \tau) \circ (\tau \star \sigma)$ indeed exists, but neither $\tau \circ \sigma$ nor $\sigma \circ \tau$ do. This indicates how the applicability of the Interchange Law is more easily seen in the pictorial representation of the transformation.

In order to further improve readability of transformation expressions, identity transformations $1_F : F \to F$ as blocks within transformation expressions are depicted as



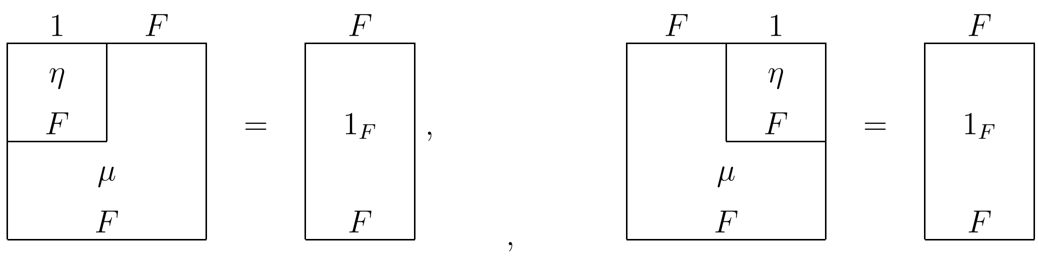
.

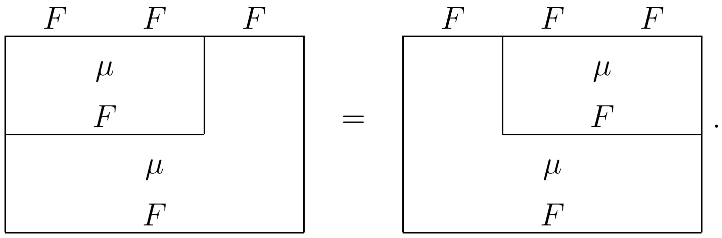
3 Monad compositions

A *monad* (or *triple*, or *algebraic theory*) over $\mathtt{C}$ is written as $\mathbf{F} = (F, \eta, \mu)$, where $F : \mathtt{C} \to \mathtt{C}$ is a (covariant) functor, and $\eta : id_{\mathtt{C}} \to F$ and $\mu : F \circ F \to F$ are natural transformations such that

$$\mu \circ (\eta \star 1_F) = 1_F, \tag{6}$$
$$\mu \circ (1_F \star \eta) = 1_F, \tag{7}$$
$$\mu \circ (1_F \star \mu) = \mu \circ (\mu \star 1_F). \tag{8}$$

We say that η is respectively a left and right unit, and that the multiplication μ is associative. These monad conditions, with the identity functor $id_{\mathtt{C}} : \mathtt{C} \to \mathtt{C}$ written as 1, can be depicted as





The following result appears in [5]. Similar results appear also in [3,4,8]

Proposition 1 *Let* $\mathbf{F} = (F, \eta^F, \mu^F)$ *and* $\mathbf{G} = (G, \eta^G, \mu^G)$ *be monads. Let* $\sigma : G \circ F \rightarrow F \circ G$ *be a natural transformation such that the following properties hold:*

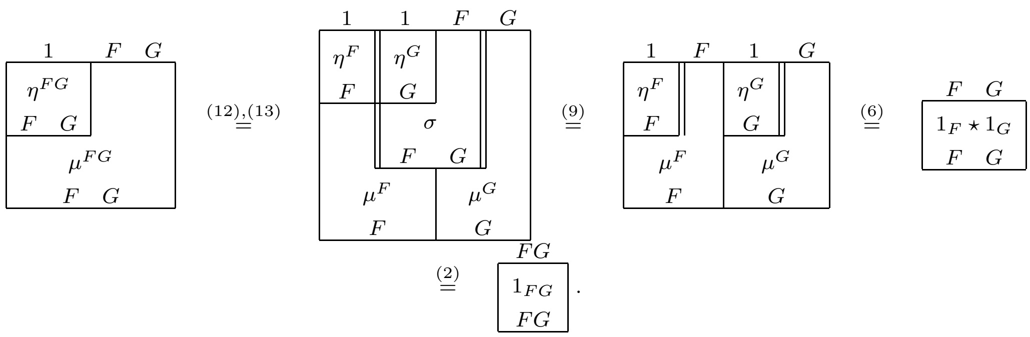
$$\sigma \circ (\eta^G \star 1_F) = 1_F \star \eta^G,$$
$$\sigma \circ (1_G \star \eta^F) = \eta^F \star 1_G,$$
$$(1_F \star \mu^G) \circ (\sigma \star 1_G) \circ (1_G \star \mu^F \star 1_G) \circ (1_{GF} \star \sigma) = (\mu^F \star 1_G) \circ (1_F \star \sigma) \circ (1_F \star \mu^G \star 1_F) \circ (\sigma \star 1_G$$

Then $\mathbf{F} \bullet \mathbf{G} = (F \circ G, \eta^{FG}, \mu^{FG})$ *is a monad, where*

$$\eta^{FG} = \eta^F \star \eta^G, \tag{12}$$
$$\mu^{FG} = (\mu^F \star \mu^G) \circ (1_F \star \sigma \star 1_G). \tag{13}$$

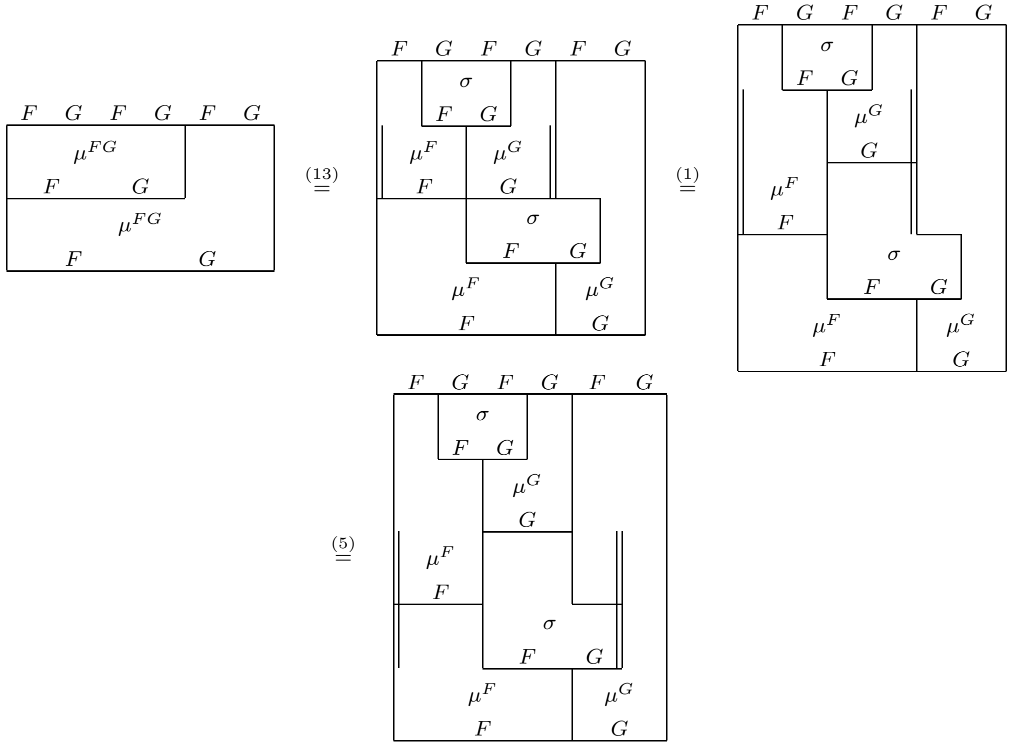
PROOF. The following proof demonstrates the use of our pictorial representations.

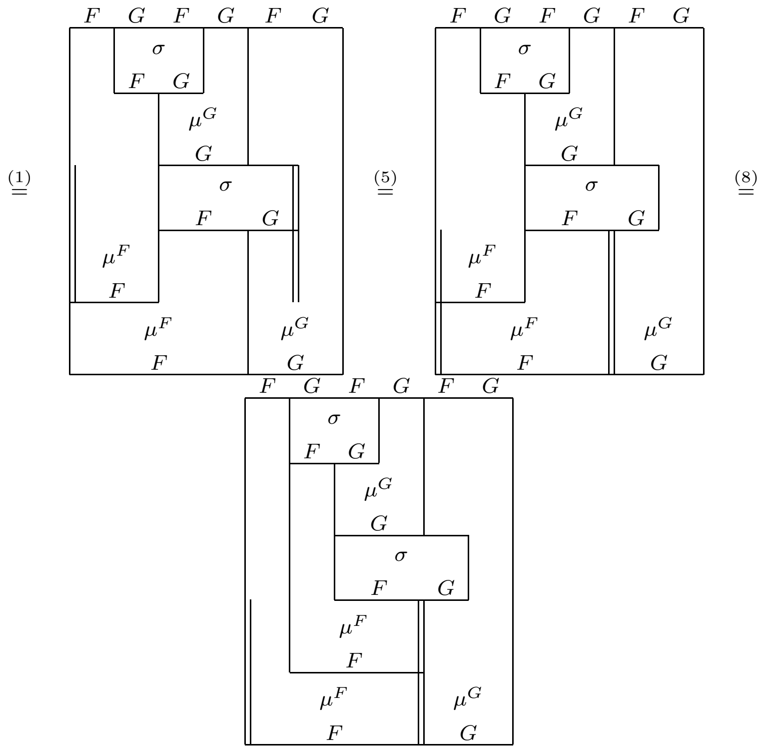
Firstly, we show that η^{FG} is a left unit.

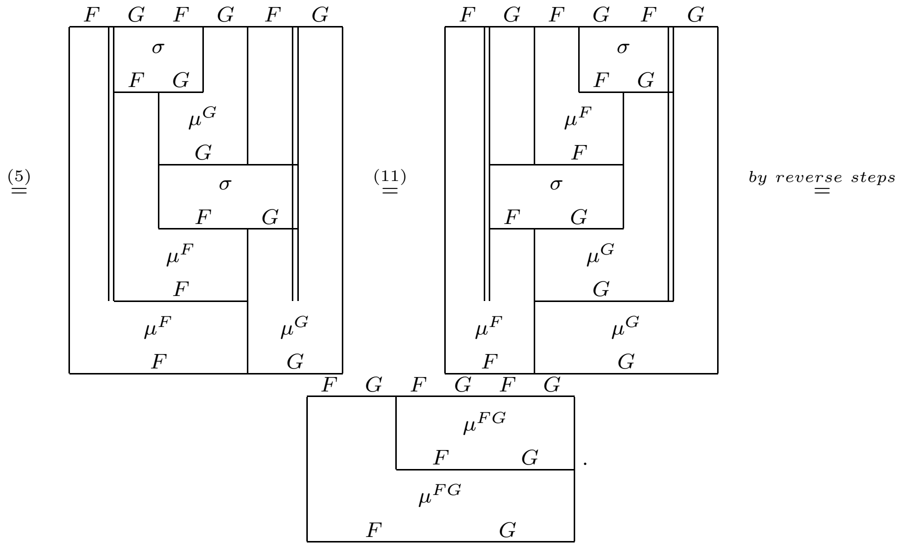


Note how the 'highlighting' of subexpressions is due to the Interchange Law. The right unit property is shown similarly.

Secondly, we show that μ^{FG} is associative.


F G F G F G
μ^{FG}
F G
μ^{FG}
F G
(13) =
F G F G F G
σ
F G
μ^F μ^G
F G
σ
F G
μ^F μ^G
F G
(1) =
F G F G F G
σ
F G
μ^G
G
μ^F
F
σ
F G
μ^F μ^G
F G
(5) =
F G F G F G
σ
F G
μ^G
G
μ^F
F
σ
F G
μ^F μ^G
F G


(1) =
F G F G F G
σ
F G
μ^G
G
σ
F G
μ^F
F
μ^F μ^G
F G
(5) =
F G F G F G
σ
F G
μ^G
G
σ
F G
μ^F
F
μ^F μ^G
F G
(8) =
F G F G F G
σ
F G
μ^G
G
σ
F G
μ^F
F
μ^F μ^G
F G



□

4 Composing powerset monads with the term monad

Let L be a completely distributive lattice. For $L = \{0, 1\}$, write $L = 2$. The covariant powerset functor L_{id} is obtained by $L_{id}X = L^X$, i.e. the set of mappings $A : X \to L$, and following [7], for a morphism $f\colon X \to Y$ in `Set`, by defining

$$L_{id}f(A)(y) = \bigvee_{x\in X} A(x) \wedge 2^{id}f(\{y\})(x) = \bigvee_{f(x)=y} A(x).$$

Further, define $\eta_X : X \to L_{id}X$ by

$$\eta_X(x)(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$$

and $\mu_X : L_{id}L_{id}X \to L_{id}X$ by

$$\mu_X(\mathcal{A})(x) = \bigvee_{A\in L_{id}X} A(x) \wedge \mathcal{A}(A).$$

Then, $\mathbf{L}_{id} = (L_{id}, \eta, \mu)$ is a monad ([10]). Note that $\mathbf{2}_{id}$ is the usual covariant powerset monad $\mathbf{P} = (P, \eta, \mu)$, where PX is the set of subsets of X, $\eta_X(x) = \{x\}$ and $\mu_X(\mathcal{B}) = \bigcup \mathcal{B}$.

These powerset monads are suitably composed with the term monad. For an operator domain Ω, let $T_\Omega X$ be the usual set of terms over Ω and variables in X, i.e., we set $T_\Omega X = \bigcup_{k=0}^{\infty} T_\Omega^k(X)$, where

$$T^0_\Omega(X) = X,$$
$$T^{k+1}_\Omega(X) = \{(n, \omega, (m_i)_{i\le n}) \mid \omega \in \Omega_n, n \in N, m_i \in T^k_\Omega(X)\}.$$

The T_Ω set functor is extended to a monad $\mathbf{T}_\Omega = (T_\Omega, \eta^{T_\Omega}, \mu^{T_\Omega})$ in the usual way ([10]).

In order to compose $\mathbf{L}_{id}$ and $\mathbf{T}_\Omega$, we need a swapper $\sigma : T_\Omega \circ L_{id} \to L_{id} \circ T_\Omega$. In [4], this was given by $\sigma_{X|T^0LX} = (1_L)_X$ and for $l = (n, \omega, (l_i)_{i\le n}) \in T^\alpha LX$, $\alpha > 0$, $l_i \in T^{\beta_i}LX$, $\beta_i < \alpha$, by

$$\sigma_X(l)((n', \omega', (m_i)_{i\le n})) = \begin{cases} \bigwedge_{i\le n} \sigma_X(l_i)(m_i) & \text{if } n = n' \text{ and } \omega = \omega', \\ \\ 0 & \text{otherwise.} \end{cases}$$

For $L = 2$, note that

$$\sigma_X(l) = \{(n, \omega, (m_i)_{i\le n}) \mid m_i \in \sigma_X(l_i)\}.$$

In [4] it was shown that σ is a natural transformation satisfying conditions (9), (10) and (11). Thus, $L_{id} \circ T_\Omega$ can be extended to a monad with a unit and a multiplication given by (12) and (13). These results can be generalised also to include double (contravariant) powerset functors and filter functors using multiplications originated from [9].

5 Composing substitutions

The Kleisli category $\mathtt{C}_\mathbf{F}$ for a monad $\mathbf{F}$ over a category $\mathtt{C}$ is given by $Ob(\mathtt{C}_\mathbf{F}) = Ob(\mathtt{C})$ and $hom_{\mathtt{C}_\mathbf{F}}(X, Y) = hom_\mathtt{C}(X, FY)$, where morphisms $f: X \rightharpoondown Y$ in $\mathtt{C}_\mathbf{F}$ are morphisms $f: X \to FY$ in $\mathtt{C}$, with $\eta^F_X: X \to FX$ being the identity morphism. Composition of morphisms in $\mathtt{C}_\mathbf{F}$ is according to

$$(X \overset{f}{\rightharpoondown} Y)\underline{\circ}(Y \overset{g}{\rightharpoondown} Z) = X \overset{\mu^F_Z \circ Fg \circ f}{\longrightarrow} FZ.$$

In the case of the term monad $\mathbf{T}_\Omega$, morphisms in the corresponding Kleisli category are variable substitutions, and most general unifiers are precisely the co-equalisers in this category ([12]).

References

[1] Adámek, J., Herrlich, H. and Strecker, G., *Abstract and Concrete Categories*, John Wiley & Sons, 1990.

[2] Barja Pérez, J.M. *Personal communication.* April, 2000.

[3] Caruncho Castro, J.R. *Triples theory.* (in Spanish) Department of Algebra of the Univ. de Santiago de Compostela Reports Series. Alxebra 5, 1971.

[4] Eklund, P., Galán, M.A., Ojeda-Aciego, M. and Valverde, A. *Set functors and generalised terms*, Proc. 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000), pp. 1595-1599.

[5] Eklund, P., Galán, M.A., Medina, J., Ojeda-Aciego, M. and Valverde, A. *Generalised terms and composition of monads*, X Spanish Conference on Fuzzy Logic and Technologies (ESTYLF 2000). Accepted. Available at `http://www.ctima.uma.es/aciego/TR/estylf.pdf`

[6] Freire Nistal, J.L. *Universal properties in higher order triples.* (in Spanish) Department of Algebra of the Univ. de Santiago de Compostela Reports Series. Alxebra 11, 1971.

[7] Goguen, J.A. *L-fuzzy sets*, J. Math. Anal. Appl. **18** (1967), 145-174.

[8] Jones, M.P. and Duponcheel, L. *Composing Monads*, Research Report YALEU/DCS/RR-1004, Yale University, New Haven, Connecticut, USA, December 1993.

[9] Kowalsky, H.-J. *Limesräume und Komplettierung*, Math. Nachr. **12** (1954), 301-340.

[10] Manes, E.G. *Algebraic Theories*, Springer-Verlag, 1976.

[11] Moggi, E. Computational lambda-calculus and monads. In *IEEE Symposium on Logic in Computer Science*, 1989.

[12] Rydeheard, D.E. and Burstall, R.M. *A categorical unification algorithm*, Proc. Summer Workshop on Category Theory and Computer Programming, 1985, LNCS 240, pp. 493–505. Springer-Verlag, 1986.

[13] Wadler, J. Comprehending monads. *Mathematical structures in computer science* 2:461–493, 1992.