Powersets of Terms and Composite Monads

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

[a] *Umeå University, Department of Computing Science, SE-901 87 Umeå, Sweden*

[b] *University of Málaga, Department of Applied Mathematics, E-29071 Málaga, Spain* [1]

Abstract

Composing various powerset functors with the term monad gives rise to the concept of generalised terms. This in turn provides a technique for handling many-valued sets of terms in a framework of variable substitutions, thus being the prerequisite for categorical unification in many-valued logic programming using an extended notion of terms. As constructions of monads involve complicated calculations with natural transformations, proofs are supported by a graphical approach that provides a useful tool for handling various conditions, such as those well-known for distributive laws.

Key words: Composite monads, generalised terms, many-valued logic

1 Introduction

A monad can be seen as the abstraction of the concept of adjoint functors and in a sense an abstraction of universal algebra. It is interesting to note that monads are useful not only in universal algebra, but it is also an important tool in topology when handling regularity, iteratedness and compactifications, and also in the study of toposes and related topics.

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 [18]. In particular, in functional programs like parsers or type checkers the monad needed is often a composed monad [21]. In logic

* Corresponding author.

1 Partially supported by Spanish DGI project TIC2003-09001-C02-01.

programming, unification has been identified as the provision of co-equalisers in Kleisli categories of term monads [19].

This paper is motivated by the use of categorical methods in many-valued logic programming, our long term goal being the generalisation of the categorical unification algorithm given by Rydeheard and Burstall in [19], in which variable substitutions are viewed as morphisms in the corresponding Kleisli categories of term monads.

Several heuristic approaches have been suggested to extend many-valued logic programming. However, the lack of a foundational base, is an obstacle for a wider acceptance of these models, and further, formal approaches typically build upon conventional terms. In particular, the generalisation of terms can be achieved by composing monads, so that generalised terms can be seen as variables assigned e.g. to (many-valued) sets of terms.

Composite monads and their algebras, together with required distributive laws were introduced and studied in [4]. However, it is not evident how to provide composite monads in general. As pointed out in [14], distributive laws between doctrines [13] are rare. Street [20], providing a formal theory of monads with respect to 2-categories [13], inspires to investigations on simplified distributive laws [16,17].

Even if the foundational understanding of monads has been well-known for decades, 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 handling of the basic combinatorial properties of functors and natural transformations (Godement rules). In [5,10] 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.

In [9] it was shown how set functors can be composed to providing monads, in particular, many-valued set monads [15] were composed with the conventional term monad.

In this paper we step towards the concept of generalised terms, for the purpose of unification aspects, by providing alternative results on constructing new monads from given ones. We also present a graphical approach that supports proving properties involving, in the end, monad compositions.

The rest of the paper is organised as follows:

In Section 2, the graphical notation for composing natural transformations is presented. We stress the usefulness of this notation, based on the Interchange

Law, which allows to perform calculations with natural transformations in an easy manner.

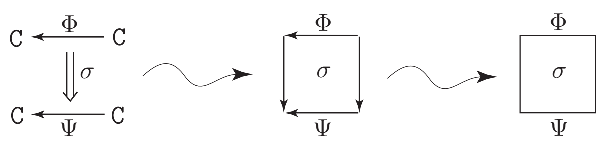
In Section 3, the concept of monad and Kleisli category of a monad are revisited. In Section 4, a first example of generalised terms is given as the composite of the powerset and the term monads. Later, in Section 5 the *ad hoc* result in the previous section is generalised categorically to apply to any pair of monads; the powerset and the term monads are shown to satisfy these new conditions, which are also proven equivalent to those of a distributive law [4]. A partially converse result is given later in Section 6, which is also related to distributive laws.

In Section 7, we provide more general results on constructing new monads from given ones. In particular, in the case of a composition of two monads, the composition of submonads of respective original monads again provide monads under a rather general condition. Finally, in Section 8, the conclusions are presented.

2 Graphical approach to natural transformations

In this section we introduce the graphical notation we will use in the rest of the paper. It was introduced in [5,10] as a means of rapid calculation with natural transformations.

Let Φ and Ψ be (covariant) endofunctors in a category $\mathtt{C}$. A natural transformation σ from Φ to Ψ is usually written as $\sigma\colon \Phi \Longrightarrow \Psi$; the graphical notation uses a different representation which, somehow, resembles the vertical writing of natural transformations as a basic building block



In addition, since functors are always oriented right to left and natural transformations are oriented top-down we can also omit the head arrows.

The *interchange law*, to be considered later, together with the box representation leads to a better management of operations with natural transformations which is not, at least directly, possible with the usual representation.

Consider endofunctors $\Phi, \Psi, \Upsilon, \ldots$ in $\mathtt{C}$, together with natural transformations $\tau, \sigma, \ldots$ between such endofunctors. For $\tau\colon \Phi \Longrightarrow \Psi$ and $\sigma\colon \Psi \Longrightarrow \Upsilon$, let $\sigma \circ$

$\tau\colon \Phi \Longrightarrow \Upsilon$ be the usual composition of natural transformations, represented by the vertical *stacking* of boxes

$$\begin{array}{c} \Phi \\ \begin{array}{|c|}\hline \tau \\ \Psi \\ \hline \sigma \\ \Upsilon \\ \hline \end{array} \end{array} \overset{def}{=} \begin{array}{c} \Phi \\ \begin{array}{|c|}\hline \\ \sigma \circ \tau \\ \\ \Upsilon \\ \hline \end{array} \end{array}$$

For $\tau'\colon \Phi' \Longrightarrow \Psi'$, the star product $\tau' \star \tau\colon \Phi' \circ \Phi \Longrightarrow \Psi' \circ \Psi$ is defined by

$$\tau' \star \tau = \tau'\Psi \circ \Phi'\tau = \Psi'\tau \circ \tau'\Phi \tag{1}$$

and its box representation will be the following

$$\begin{array}{cc} \Phi' & \Phi \\ \multicolumn{2}{c}{\begin{array}{|c|c|}\hline \tau' & \tau \\ \Psi' & \Psi \\ \hline \end{array}} \end{array} \overset{def}{=} \begin{array}{c} \Phi' \quad \Phi \\ \begin{array}{|c|}\hline \tau' \star \tau \\ \Psi' \quad \Psi \\ \hline \end{array} \end{array}$$

Note how the associativity of the product and composition is implicit in the representation.

In the sequel we show how this representation is helpful in order to prove several well-known results regarding composition and star product.

For the identity transformation $id_\Phi\colon \Phi \Longrightarrow \Phi$, also written as 1_Φ or 1, note that $1_\Phi \star 1_\Psi = 1_{\Phi\circ\Psi}$. For a natural transformation $\tau\colon \Phi \Longrightarrow \Psi$, and a functor Υ, it is possible to define its composition $(\Upsilon\tau)_X = \Upsilon\tau_X$ and $(\tau\Upsilon)_X = \tau_{\Upsilon X}$, or equivalently, $\Upsilon\tau = 1_\Upsilon \star \tau$ and $\tau\Upsilon = \tau \star 1_\Upsilon$, which allows us to pictorially represent equation (1) by

$$\begin{array}{c} \Phi' \quad \Phi \\ \begin{array}{|c|}\hline \tau' \star \tau \\ \Psi' \quad \Psi \\ \hline \end{array} \end{array} = \begin{array}{c} \Phi' \quad \Phi \\ \begin{array}{|c|}\hline 1_{\Phi'} \star \tau \\ \Phi' \quad \Psi \\ \hline \tau' \star 1_\Psi \\ \Psi' \quad \Psi \\ \hline \end{array} \end{array} = \begin{array}{c} \Phi' \quad \Phi \\ \begin{array}{|c|}\hline \tau' \star 1_\Phi \\ \Psi' \quad \Phi \\ \hline 1_{\Psi'} \star \tau \\ \Psi' \quad \Psi \\ \hline \end{array} \end{array}$$

It will be useful to recall the following distributivity laws together with its box representation:

$$1 \star (\sigma \circ \tau) = (1 \star \sigma) \circ (1 \star \tau), \tag{2}$$
$$(\sigma \circ \tau) \star 1 = (\sigma \star 1) \circ (\tau \star 1). \tag{3}$$

Equation (2) can be written as

$$\begin{array}{cc} \Gamma & \Phi \\ \hline \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{\sigma\circ\tau} \\ \multicolumn{1}{|c|}{\Gamma} & \multicolumn{1}{c|}{\Upsilon} \\ \hline \end{array} \quad = \quad \begin{array}{|cc|} \multicolumn{2}{c}{\Gamma \quad \Phi} \\ \hline \multicolumn{2}{|c|}{1\star\tau} \\ \Gamma & \Psi \\ \hline \multicolumn{2}{|c|}{1\star\sigma} \\ \Gamma & \Upsilon \\ \hline \end{array}$$

i.e., in this case building blocks can be applied in any order. The same holds for equation (3).

Proposition 1 (Interchange Law) *Let* $\Phi \overset{\tau}{\Longrightarrow} \Psi \overset{\sigma}{\Longrightarrow} \Upsilon$ *and* $\Phi' \overset{\tau'}{\Longrightarrow} \Psi' \overset{\sigma'}{\Longrightarrow} \Upsilon'$ *be natural transformations. Then we have*

$$(\sigma' \star \sigma) \circ (\tau' \star \tau) = (\sigma' \circ \tau') \star (\sigma \circ \tau) \tag{4}$$

PROOF.

$$\begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{\tau'\star\tau} \\ \Psi' & \Psi \\ \hline \multicolumn{2}{|c|}{\sigma'\star\sigma} \\ \Upsilon' & \Upsilon \\ \hline \end{array} \overset{(1)}{=} \begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{\tau'\star 1_\Phi} \\ \Psi' & \Phi \\ \hline \multicolumn{2}{|c|}{1_{\Psi'}\star\tau} \\ \Psi' & \Psi \\ \hline \multicolumn{2}{|c|}{1_{\Psi'}\star\sigma} \\ \Psi' & \Upsilon \\ \hline \multicolumn{2}{|c|}{\sigma'\star 1_\Upsilon} \\ \Upsilon' & \Upsilon \\ \hline \end{array} \overset{(2)}{=} \begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{\tau'\star 1_\Phi} \\ \Psi' & \Phi \\ \hline \multicolumn{2}{|c|}{1_{\Psi'}\star(\sigma\circ\tau)} \\ \Psi' & \Upsilon \\ \hline \multicolumn{2}{|c|}{\sigma'\star 1_\Upsilon} \\ \Upsilon' & \Upsilon \\ \hline \end{array} \overset{(1)}{=} \begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{\tau'\star 1_\Phi} \\ \Psi' & \Phi \\ \hline \multicolumn{2}{|c|}{\sigma'\star 1_\Phi} \\ \Upsilon' & \Phi \\ \hline \multicolumn{2}{|c|}{1_{\Upsilon'}\star(\sigma\circ\tau)} \\ \Upsilon' & \Upsilon \\ \hline \end{array}$$

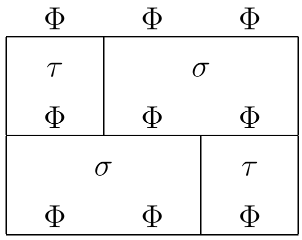
$$\overset{(3)}{=} \begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{(\sigma'\circ\tau')\star 1_\Phi} \\ \Upsilon' & \Phi \\ \hline \multicolumn{2}{|c|}{1_{\Upsilon'}\star(\sigma\circ\tau)} \\ \Upsilon' & \Upsilon \\ \hline \end{array} \overset{(1)}{=} \begin{array}{|c|c|} \multicolumn{1}{c}{\Phi'} & \multicolumn{1}{c}{\Phi} \\ \hline \sigma'\circ\tau' & \sigma\circ\tau \\ \Upsilon' & \Upsilon \\ \hline \end{array} \qquad \blacksquare$$

The graphical representation of the Interchange Law leads to the following rule

$$\begin{array}{|c|c|} \multicolumn{1}{c}{\Phi'} & \multicolumn{1}{c}{\Phi} \\ \hline \tau' & \tau \\ \Psi' & \Psi \\ \hline \sigma' & \sigma \\ \Upsilon' & \Upsilon \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|} \multicolumn{1}{c}{\Phi'} & \multicolumn{1}{c}{\Phi} \\ \hline \sigma'\circ\tau' & \sigma\circ\tau \\ \Upsilon' & \Upsilon \\ \hline \end{array} \quad = \quad \begin{array}{|cc|} \multicolumn{2}{c}{\Phi' \quad \Phi} \\ \hline \multicolumn{2}{|c|}{\tau'\star\tau} \\ \Psi' & \Psi \\ \hline \multicolumn{2}{|c|}{\sigma'\star\sigma} \\ \Upsilon' & \Upsilon \\ \hline \end{array}$$

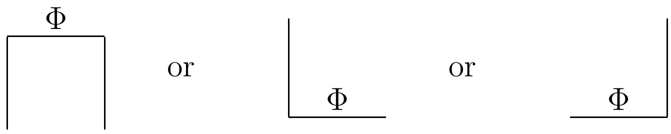
which allows to rearrange a stack of blocks as desired, showing how blocks with particular positions generally can be attached vertically and horizontally in any order without changing the resulting transformation.

Note in the transformation



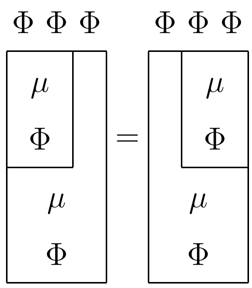
that the composition $(\sigma \star \tau) \circ (\tau \star \sigma)$ indeed exists, but neither $\tau \circ \sigma$ nor $\sigma \circ \tau$ do. This indicates how the applicability of the Interchange Law is more easily seen in the pictorial representation of the transformation.

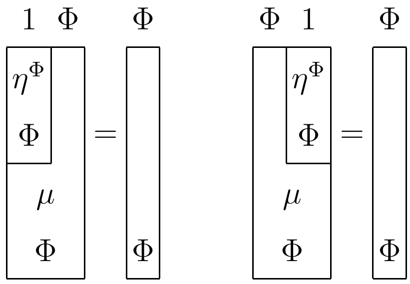
In order to further improve readability of transformation expressions, identity transformations $1_\Phi : \Phi \Longrightarrow \Phi$ as blocks within transformation expressions are depicted as



This choice for the representation of identity transformations will allow the use of asymmetric stacking of boxes.

3 Monads and Kleisli categories

Let $\mathtt{C}$ be a category. A monad (or triple, or algebraic theory) over $\mathtt{C}$ is written as $\mathbf{\Phi} = (\Phi, \eta, \mu)$, where $\Phi\colon \mathtt{C} \to \mathtt{C}$ is a covariant functor, and $\eta\colon id \Longrightarrow \Phi$ and $\mu\colon \Phi \circ \Phi \Longrightarrow \Phi$ are natural transformations for which $\mu \circ \Phi\mu = \mu \circ \mu\Phi$ and $\mu \circ \Phi\eta = \mu \circ \eta\Phi = id_\Phi$ hold. Pictorially, Associativity: 

Units: 

It is useful to write η^Φ and μ^Φ if we need to distinguish between natural transformations in different monads. For notations and results within general category theory and universal algebra, we refer to [1,3,15].

A Kleisli category $\mathtt{C}_{\mathbf{\Phi}}$ for a monad $\mathbf{\Phi}$ over a category $\mathtt{C}$ is defined as follows: Objects in $\mathtt{C}_{\mathbf{\Phi}}$ are the same as in $\mathtt{C}$, and the morphisms are defined as $hom_{\mathtt{C}_{\mathbf{\Phi}}}(X, Y) = hom_{\mathtt{C}}(X, \Phi Y)$, that is morphisms $f\colon X \rightharpoondown Y$ in $\mathtt{C}_{\mathbf{\Phi}}$ are simply morphisms $f\colon X \to \Phi Y$ in $\mathtt{C}$, with $\eta^\Phi_X\colon X \to \Phi X$ being the identity morphism.

Composition of morphisms is defined as

$$(X \overset{f}{\rightharpoondown} Y) \circ (Y \overset{g}{\rightharpoondown} Z) = X \xrightarrow{\mu^\Phi_Z \circ \Phi g \circ f} \Phi Z.$$

The Kleisli category is equivalent to the full subcategory of free Φ-algebras of the monad, and its definition makes it clear that the arrows are substitutions. Indeed, the categorical unification algorithm in [19] is based on the Kleisli category of the term monad.

A monad (Φ, η, μ) written as $(\Phi, \eta, \circ)$, where $\circ$ is the composition of morphisms in the corresponding Kleisli category, is said to be a monad in clone form. In fact, there is a one-to-one correspondence between monads, respectively, in monoid and clone forms [15].

4 Composing powerset and term monads: an example

4.1 The powerset monad

Let L be a completely distributive lattice. For $L = \{0, 1\}$ we will write $L = 2$. The covariant powerset functor L_{id} is obtained by $L_{id}X = L^X$, i.e. L-fuzzy sets $A\colon X \to L$, and following [12], for a morphism $f\colon X \to Y$ in `Set`, by defining

$$L_{id}f(A)(y) = \bigvee_{f(x)=y} A(x)$$

In [15] it is proved that $\boldsymbol{L}_{id} = (L_{id}, \eta, \mu)$ is a monad with the following identity and multiplication: $\eta_X\colon X \to L_{id}X$ is defined by

$$\eta_X(x)(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and $\mu_X\colon L_{id}L_{id}X \to L_{id}X$ is defined by

$$\mu_X(\mathcal{A})(x) = \bigvee_{A \in L_{id}X} A(x) \wedge \mathcal{A}(A). \tag{6}$$

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

The problem of extending a functor to a monad is not a trivial one, and some strange situations may well arise as shown below. Note that the id^2 functor can be extended to a monad with $\eta_X(x) = (x, x)$ and $\mu_X((x_1, x_2), (x_3, x_4)) = (x_1, x_4)$. Similarly, id^n can be extended to a monad. In addition, the proper powerset functor P_0, where $P_0X = PX \setminus \{\varnothing\}$, as well as $id^2 \circ P_0$ can, respectively, be extended to a monad in a unique way. However, $P_0 \circ id^2$ cannot be made to a monad [6].

Remark 1 *Let $\mathbf{\Phi} = (\Phi, \eta^\Phi, \mu^\Phi)$ and $\mathbf{\Psi} = (\Psi, \eta^\Psi, \mu^\Psi)$ be monads over* `Set`*. The composition $\Phi \circ \Psi$ cannot always be extended to a monad as seen in the case of $P_0 \circ id^2$.*

4.2 The term monad

Regarding the set of terms it is useful to adopt a more functorial presentation of it, as opposed to using the conventional inductive definition of terms, where we bind ourselves to certain styles of proofs. Even if a purely functorial presentation might seem complicated, there are advantages when we define corresponding monads, and, further, a functorial presentation simplifies efforts to prove results concerning compositions of monads. This completely categorical approach was given in [11].

Let $\Omega = \coprod_{n=0}^{\infty} \Omega_n$ be an operator domain, where each Ω_n is intended to contain operators of arity n. Then, the definition of the term functor $T_\Omega \colon \mathtt{Set} \to \mathtt{Set}$ can be intuitively given as $T_\Omega(X) = \bigcup_{k=0}^{\infty} T_\Omega^k(X)$, where $T_\Omega^0(X) = X$ and $T_\Omega^{k+1}(X)$ is intended to represent

$$\{\omega(x_1, \ldots, x_n) \mid \omega \in \Omega_n, x_i \in T_\Omega^k(X)\}$$

The formal categorical definition of the term monad requires some previous notation and definitions:

Definition 1 *(1) For a set A, the* constant set functor $A_{\mathtt{Set}}$ *is the covariant set functor for which:*

$$A_{\mathtt{Set}}(X) = A \qquad A_{\mathtt{Set}}(f) = id_A$$

for all set X and morphism f in the category $\mathtt{Set}$.

We will usually drop out the subscript $\mathtt{Set}$ *whenever no confusion arise.*

(2) The sum (coproduct) $\coprod_{i \in I} \Phi_i$ of covariant set functors Φ_i *is defined for each set X and for each morphism $f \colon X \to Y$ in* $\mathtt{Set}$ *as follows:*

$$\Big(\coprod_{i \in I} \Phi_i\Big) X = \bigcup_{i \in I} (\{i\} \times \Phi_i X) \qquad \Big(\coprod_{i \in I} \Phi_i\Big) f(i, m) = (i, \Phi_i f(m))$$

where $(i, m) \in (\coprod_{i \in I} \Phi_i) X$.

The term functor can now be defined by transfinite induction as stated in the following definition:

Definition 2 *Let κ be a cardinal number and $(\Omega_n)_{n \le \kappa}$ be a family of sets, and $\Omega = \coprod_{n \le \kappa} \Omega_n$. The* term functor *is given by transfinite induction considering*

$$T_\Omega^0 = id$$

and defining

$$T_\Omega^\alpha = \Big(\coprod_{n \le \kappa} (\Omega_n \times id^n)\Big) \circ \bigcup_{\beta < \alpha} T_\Omega^\beta$$

for each positive ordinal α. Finally, let

$$T_\Omega = \bigcup_{\alpha < \bar{\kappa}} T_\Omega^\alpha$$

where $\bar{\kappa}$ is the least cardinal greater than κ and $\aleph_0$.

Note that the elements in $T^{\alpha}_{\Omega}X$, with $\alpha \neq 0$, have the form $(n, \omega, (m_i)_{i\leq n})$ with $m_i \in T^{\beta_i}_{\Omega}X$ for $\beta_i < \alpha$, and were represented as $\omega(m_1, \ldots, m_n)$ in the intuitive approach to the definition of the term monad.

The interest of Ω being the sum of the Ω_n lies in the fact that it builds Ω-algebras. Note that in the definition of the term functor, the sum structure of Ω is not used.

Definition 3 *(1) Let κ be a cardinal number and $(\Omega_n)_{n\leq\kappa}$ be a family of sets. The sum $\Omega = \coprod_{n\leq\kappa}\Omega_n$ is called an* operator domain.

(2) An Ω-algebra *is a pair $(X, (s_{n\omega})_{(n,\omega)\in\Omega})$ where $s_{n\omega}\colon X^n \to X$ are n-ary operations.*

(3) The category of Ω-algebras *consists of Ω-algebras as objects, and morphisms between $(X, (s_{n\omega})_{(n,\omega)\in\Omega})$ and $(Y, (t_{n\omega})_{(n,\omega)\in\Omega})$ as mappings $f\colon X \to Y$ satisfying*

$$f(s_{n\omega}(m_i)_{i\leq n}) = t_{n\omega}((f(m_i))_{i\leq n})$$

The sets $T_\Omega X$ can be interpreted as an Ω-algebra, $(T_\Omega X, (\sigma_{n\omega})_{(n,\omega)\in\Omega})$, just defining $\sigma_{n\omega}((m_i)_{i\leq n}) = (n, \omega, (m_i)_{i\leq n})$ for $\omega \in \Omega_n$ and $m_i \in T_\Omega X$. Actually, this algebra is a freely generated algebra in the category of Ω-algebras, that is, for an Ω-algebra $B = (Y, (t_{n\omega})_{(n,\omega)\in\Omega})$, a morphism $f\colon X \to Y$ in `Set` can be extended to an Ω-homomorphism $f^*\colon (T_\Omega X, (\sigma_{n\omega})_{(n,\omega)\in\Omega}) \to (Y, (t_{n\omega})_{(n,\omega)\in\Omega})$, called the Ω-extension of f associated to B, given by $f^*_{|T^0_\Omega X} = f$ and

$$f^*(n, \omega, (m_i)_{i\leq n}) = t_{n\omega}((f^*(m_i))_{i\leq n})$$

for all $n \in \mathbb{N}$ and $(n, \omega, (m_i)_{i\leq n}) \in T^{\alpha}_{\Omega}X$.

A morphism $f\colon X{\to}Y$ in `Set` can also be extended to the corresponding Ω-homomorphism

$$(T_\Omega X, (\sigma_{n\omega})_{(n,\omega)\in\Omega}) \xrightarrow{T_\Omega f} (T_\Omega Y, (\tau_{n\omega})_{(n,\omega)\in\Omega})$$

where $T_\Omega f$ is defined to be the Ω-extension of $X \xrightarrow{f} Y \hookrightarrow T_\Omega Y$ associated to $(T_\Omega Y, (\tau_{n\omega})_{(n,\omega)\in\Omega})$.

In [15] it was shown that T_Ω can be extended to a monad $\boldsymbol{T}_\Omega = (T_\Omega, \eta^{T_\Omega}, \mu^{T_\Omega})$ with the following definition for the unit $\eta^{T_\Omega}_X(x) = x$ and, regarding the multiplication, $\mu^{T_\Omega}_X = id^*_{T_\Omega X}$ is the Ω-extension of $id_{T_\Omega X}$ with respect to $(T_\Omega X, (\sigma_{n\omega})_{(n,\omega)\in\Omega})$.

5 On the composition of monads

Regarding the composition of monads we are interested in the following general problem: Given monads $\mathbf{\Phi}$ and $\mathbf{\Psi}$, we consider the composition of the corresponding functors, $\Phi \circ \Psi$: Is it possible to extend this composition to a monad? Is this extension compatible with the monad structure in the initial monads? Does there exist a natural definition of the unit and multiplication in the composed functor which uses the units and multiplication in $\mathbf{\Phi}$ and $\mathbf{\Psi}$? Is this extension, in some sense, unique?

A first approach to the study of the structure of monad of the composition of powerset monads and term monads was presented in [9], where it was shown how set functors can be composed to providing monads, and some motivation to investigate techniques for constructing new monads from given ones was presented. Specifically, it was shown that the composition $L_{id} \circ T_\Omega$ could be extended to a monad.

As we have already pointed out, it is not clear to what extent 'monads composed as functors are extendable to monads' is a rule rather than exception. In the case of complicated functors it is rather tempting intuitively to believe that monad compositions are not common simply by observing how the use of distributivity laws in proofs makes diagrams grow to sizes extremely difficult to overlook. On the other hand Beck says that proofs involving the distributivity law 'are just long naturality calculations.' It is the latter view that inspires us to provide a calculus for computing with natural transformations.

By using the graphical calculus it is easy to obtain a set of equations which are sufficient conditions for defining a monad structure on the composition of two monads.

Proposition 2 *Given monads* $\mathbf{\Phi} = (\Phi, \eta^\Phi, \mu^\Phi)$ *and* $\mathbf{\Psi} = (\Psi, \eta^\Psi, \mu^\Psi)$*, let* $\sigma\colon \Psi \circ \Phi \to \Phi \circ \Psi$ *be a natural transformation, then the composition can be provided with a multiplication* $\mu(\sigma)\colon \Phi\Psi\Phi\Psi \to \Phi\Psi$ *defined by* $\mu(\sigma) = (\mu^\Phi * \mu^\Psi) \circ \Phi\sigma\Psi$ *which makes* $\mathbf{\Phi} \bullet \mathbf{\Psi} = (\Phi \circ \Psi, \eta^\Phi * \eta^\Psi, \mu(\sigma))$ *to be a monad if the following properties hold:*

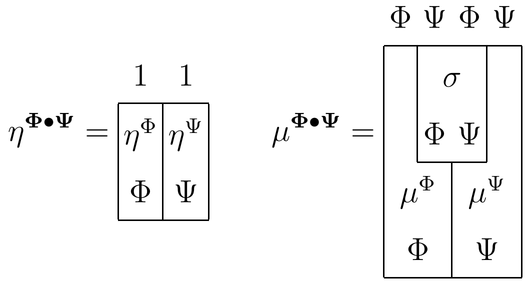
(1σ) $\mu^\Phi\Psi \circ \Phi\sigma \circ \Phi\mu^\Psi\Phi \circ \sigma\Psi\Phi = \Phi\mu^\Psi \circ \sigma\Psi \circ \Psi\mu^\Phi\Psi \circ \Psi\Phi\sigma$
(2σ) $\sigma \circ \eta^\Psi\Phi = \Phi\eta^\Psi$
(3σ) $\sigma \circ \Psi\eta^\Phi = \eta^\Phi\Psi$

Note that

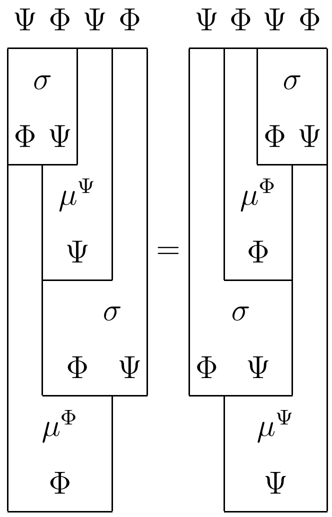
- (1σ) states the independence of the swapper from the order of application.
- By (2σ) the swapper extracts Φ from η^Ψ.
- By (3σ) the swapper introduces Ψ in η^Φ.

PROOF.

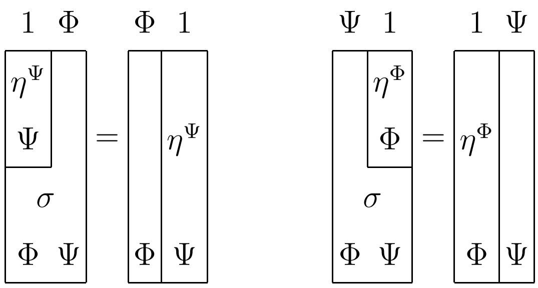
We introduce firstly the pictorial representations for the conditions. The natural transformations $\mu(\sigma)$ and $\eta^{\Phi} * \eta^{\Psi}$ are



The condition (1σ) is:



And the conditions (2σ) and (3σ) are:



By definition, $\mu(\sigma)$ is clearly a natural transformation. We have only to prove that it is associative and that it also satisfies the right and left identities.

The associativity of the multiplication is straightforward, for we have that $\mu(\sigma) \circ \mu(\sigma)\Phi\Psi$ can be pictorially represented as the first diagram in Figure 1: the first equality follows from the associativity of μ^{Φ}, and the second one by condition (1σ). An additional application of the associativity of μ^{Ψ} gives the

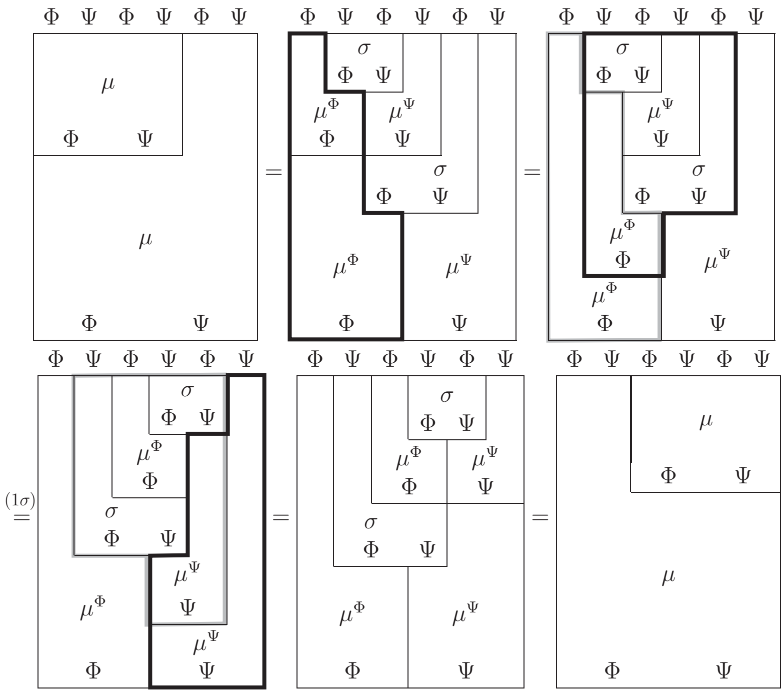


Fig. 1. Proof of the associativity of the multiplication.

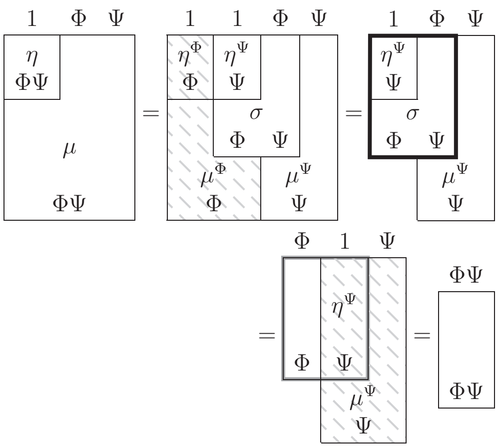


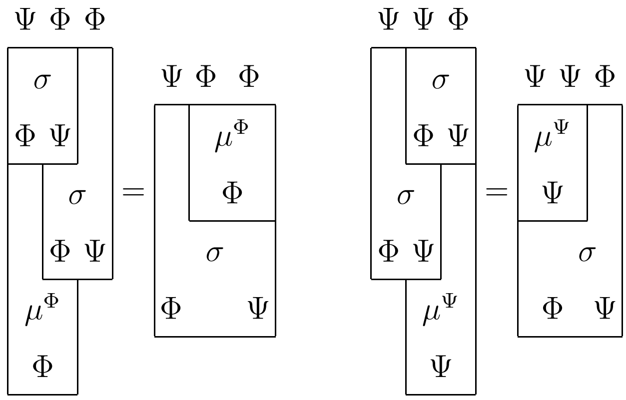
Fig. 2. Proof of left and right identities.

diagram of $\mu(\sigma) \circ \Phi\Psi\mu(\sigma)$.

For the left and right identity, just consider the diagrams in Figure 2, in which the first and third equalities in any row hold from the fact that both Φ and Ψ are monads, and the second equality follows from condition (2σ) and (3σ), respectively. ■

The conditions given by the previous result can be proven equivalent to Beck's conditions on a distributive law as an easy application of the graphical calculus of natural transformations. Firstly, let us recall the definition of a distributive law [4]:

Definition 4 (Beck) *Let $\mathbf{\Phi} = (\Phi, \eta^{\Phi}, \mu^{\Phi})$ and $\mathbf{\Psi} = (\Psi, \eta^{\Psi}, \mu^{\Psi})$ be two monads, a* distributive law *of $\mathbf{\Phi}$ over $\mathbf{\Psi}$ is a natural transformation $\sigma \colon \Psi \circ \Phi \to \Phi \circ \Psi$ such that conditions (2σ) and (3σ) are satisfied, together with the two additional conditions below*



Specifically, we have the following theorem establishing the correspondence between Proposition 2 and distributive laws.

Theorem 1 *Given monads $\mathbf{\Phi} = (\Phi, \eta^{\Phi}, \mu^{\Phi})$ and $\mathbf{\Psi} = (\Psi, \eta^{\Psi}, \mu^{\Psi})$, let $\sigma \colon \Psi \circ \Phi \to \Phi \circ \Psi$ be a natural transformation, then σ is a distributive law if and only if conditions (1σ), (2σ) and (3σ) are satisfied.*

PROOF.

It is easy to show that Beck's conditions imply (1σ), (2σ) and (3σ) (actually (2σ) and (3σ) are shared).

The proof of Beck's condition on μ^{Φ} is shown in Figure 3, the one respect to μ^{Ψ} is similar. In the proof above equalities tagged with (1) follow from the addition or supression of identities (represented as dashed boxes); equalities (2) and (3) follow, respectively, from application of (1σ) and (2σ) to the bold framed box. $\blacksquare$

It is interesting to note that Proposition 2 applies the particular case of the definition of many-valued powerset monad[2] $L = \Phi$ and the term monad $T = \Psi$, and the swapper σ defined as a natural transformation with mappings $\sigma_X \colon TLX \to LTX$ as follows:

[2] In the following we will write L instead of L_{id} and T instead of T_{Ω}.

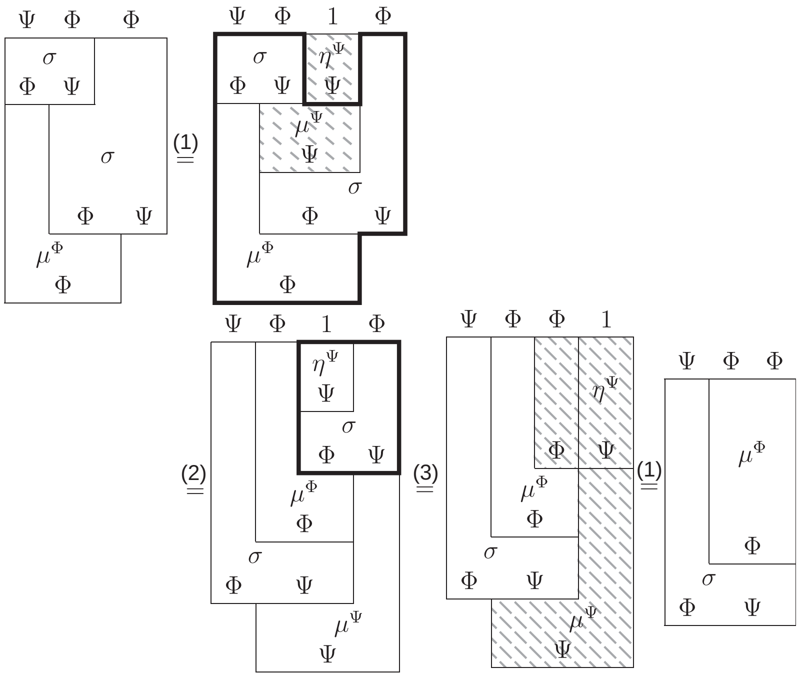


Fig. 3.

For the base case $\sigma_{X|_{T^0LX}} = id_{LX}$. Further, for $l = (n, \omega, (l_i)_{i \leq n}) \in T^k LX$, $k > 0$, $l_i \in T^{n_i} LX$, $n_i < k$, let $\sigma_X(l)((n', \omega', (m_i)_{i \leq n'}))$ be defined as

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

It is not difficult to show that the swapper just defined is actually a distributive law.

Proposition 3 *Monads L and T, and the swapper σ defined above satisfy properties (1σ), (2σ) and (3σ) in Proposition 2. Thus, $L \circ T$ can be extended to a monad.*

For the proof of this proposition the following technical lemma is needed:

Lemma 1 *Consider the composition $LTLX \xrightarrow{L\sigma_X} LLTX \xrightarrow{\mu^L_{TX}} LTX$, and $R \in LTLX$ and $m \in TX$, then*

$$(\mu^L_{TX}(L\sigma_X(R)))(m) = \bigvee_{r \in TLX} R(r) \wedge (\sigma_X(r))(m)$$

PROOF.

$$
\begin{aligned}
(\mu^L_{TX}(L\sigma_X(R)))(m) &\overset{\text{Def. of } \mu^L}{=} \bigvee_{A\in LTX} (L\sigma_X(R))(A) \wedge A(m) \\
&\overset{\text{Def. of } L}{=} \bigvee_{A\in LTX} \Big(\bigvee_{\sigma_X(r)=A} R(r)\Big) \wedge A(m) \\
&\overset{\text{Distr.}}{=} \bigvee_{A\in LTX} \bigvee_{\sigma_X(r)=A} (R(r) \wedge A(m)) \\
&= \bigvee_{r\in TLX} (R(r) \wedge \sigma_x(r)(m))
\end{aligned}
$$

In the last equality we use the fact that the elements in $A \in LTX$ for which an element r exists such that $\sigma_X(r) = A$ are exactly those in the image of σ_X.

■

Proof of Proposition 3

Condition (1σ): Let us prove that the swapper σ satisfies

$$L\mu^T \circ \sigma_T \circ T\mu^L_T \circ TL\sigma = \mu^L_T \circ L\sigma \circ L\mu^T_L \circ \sigma_{TL}$$

Given $d \in TLTLX$ and $m \in TX$, we will inductively show that

$$\sigma_{TX}((T(\mu^L_{TX} \circ L\sigma))(d))(m) = (\mu^L_{TX} \circ L\sigma_X)(\sigma_{TLX}(d))(m)$$

- If $d \in LTLX$ and $m \in X$ the equality is trivial.
- By induction, assume that $d = (n, \omega, (d_i)_{i\leq n})$ and $m = (n, \omega, (m_i)_{i\leq n})$, (otherwise, the right hand side equals 0):

$$
\begin{aligned}
&\sigma_{TX}((T(\mu^L_{TX} \circ L\sigma))(d))(m) = \\
&\overset{\text{Def. of } \sigma}{=} \bigwedge_{i\leq n} \sigma_{TX}((T(\mu^L_{TX} \circ L\sigma))(d_i))(m_i) \\
&\overset{\text{Induc.}}{=} \bigwedge_{i\leq n} (\mu^L_{TX} \circ L\sigma_X)(\sigma_{TLX}(d_i))(m_i) \\
&\overset{\text{Lemma}}{=} \bigwedge_{i\leq n} \bigvee_{r\in TTLX} \sigma_{TLX}(d_i)(r) \wedge (\sigma_X)(r)(m_i) \\
&\overset{\text{Distr.}}{=} \bigvee_{(n,\omega,(r_i)_{i\leq n})\in TTLX} \bigwedge_{i\leq n} \sigma_{TLX}(d_i)(r_i) \wedge (\sigma_X)(r_i)(m_i) \\
&= \bigvee_{r\in TTLX} \sigma_{TLX}(d)(r) \wedge (\sigma_X)(r)(m) \\
&= \bigvee_{r\in TLX} \sigma_{TLX}(d)(r) \wedge (\sigma_X)(r)(m) \\
&\overset{\text{Lemma}}{=} (\mu^L_{TX} \circ L\sigma_X)(\sigma_{TLX}(d))(m)
\end{aligned}
$$

Condition (2σ): Let us prove that the swapper σ satisfies

$$\sigma \circ \eta^T_L = L\eta^T$$

By the definitions $\eta^T_X = T^0X = X$, $\eta^T_{LX} = T^0LX = LX$ and σ_X on LX is LX; therefore the equality holds.

Condition (3σ): Let us prove that the swapper σ satisfies

$$\sigma \circ T\eta^L = \eta^L_T$$

For all X we have $\sigma_X \circ T\eta^L_X : TX \to TLX \to LTX$, therefore we will inductively show that if $m \in TX$, then $\sigma_X(T\eta^L_X(m)) = \eta^L_{TX}(m)$.

- If $m \in X$, then $\sigma_X(T\eta^L_X(m)) = \sigma_X(\eta^L_X(m)) = \eta^L_X(m) = \eta^L_{TX}(m)$.
- If $m = (n, \omega, (m_i)_{i \leq n})$, then $\sigma_X(T\eta^L_X(m)) : TX \to L$. Consider $m' \in TX$, the only case in which $\sigma_X(T\eta^L_X(m))(m')$ can be nonzero is when $m' = (n, \omega, (m'_i)_{i \leq n})$:

$$\begin{aligned}
\sigma_X(T\eta^L_X(m))(m') &\overset{\text{Def of } Tf}{=} \sigma_X(n, \omega, (T\eta^L_X(m_i))_{i \leq n})(m') \\
&\overset{\text{Def of } \sigma}{=} \bigwedge_{i \leq n} \sigma_X(T\eta^L_X(m_i))(m'_i) \\
&\overset{\text{Induc.}}{=} \bigwedge_{i \leq n} \eta^L_{TX}(m_i)(m'_i) \\
&= \eta^L_{TX}(m)(m') \qquad \blacksquare
\end{aligned}$$

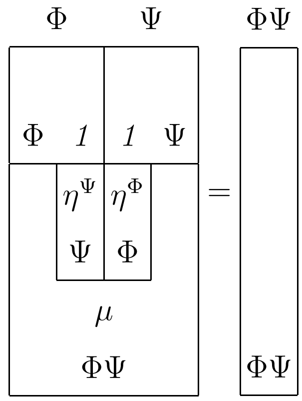
6 Reverse engineering of monads

In this section we will, again using our graphical tool, re-establish the well-known converse result to Proposition 2. See e.g. [2] for original proofs.

Theorem 2 (Beck) *Let* $\mathbf{\Phi} = (\Phi, \eta^\Phi, \mu^\Phi)$ *and* $\mathbf{\Psi} = (\Phi, \eta^\Psi, \mu^\Psi)$ *be two monads. If* $\mathbf{\Phi} \bullet \mathbf{\Psi} = (\Phi \circ \Psi, \eta^\Phi * \eta^\Psi, \mu)$ *is a monad such that:*

(R1) $\Phi\eta^\Psi : \Phi \Rightarrow \Phi\Psi$ *is a morphism of monads.*
(R2) $\eta^\Phi\Psi : \Psi \Rightarrow \Phi\Psi$ *is a morphism of monads.*

(R3) The middle unitary law, given below, is satisfied.

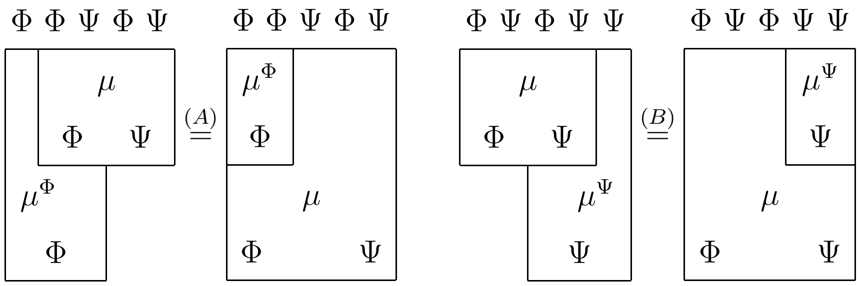


Then the natural transformation $\sigma(\mu)\colon \Psi\Phi \Rightarrow \Phi\Psi$ *defined as*

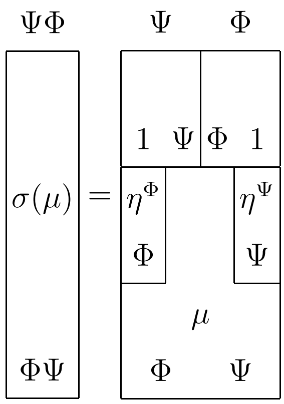
$$\sigma(\mu) = \mu \circ \Phi\Psi\Phi\eta^\Psi \circ \eta^\Phi\Psi\Phi$$

is a distributive law.

Theorem 3 *Let* $\mathbf{\Phi} = (\Phi, \eta^\Phi, \mu^\Phi)$ *and* $\mathbf{\Psi} = (\Phi, \eta^\Psi, \mu^\Psi)$ *be two monads. If* $\mathbf{\Phi} \bullet \mathbf{\Psi} = (\Phi \circ \Psi, \eta^\Phi * \eta^\Psi, \mu)$ *is a monad such that equalities* (A) *and* (B) *below are satisfied:*



then the natural transformation $\sigma(\mu)\colon \Psi\Phi \Rightarrow \Phi\Psi$ *defined below is a distributive law:*



PROOF. Conditions (2σ) and (3σ) are trivially satisfied, as shown in Figure 4, just note that the unit transformation of the composed monad is the composition of the units of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ and the dashed box is the identity.

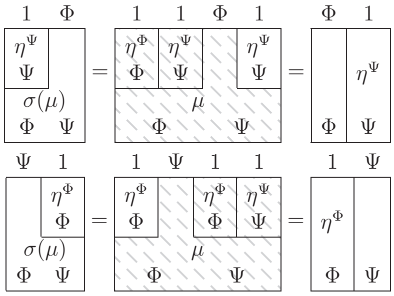


Fig. 4.

For (1σ) consider the picture in Figures 5,6, in which the bold lines represent the boxes on which properties (A) and (B) will be applied, and the dashed boxes are equivalent to the identity.

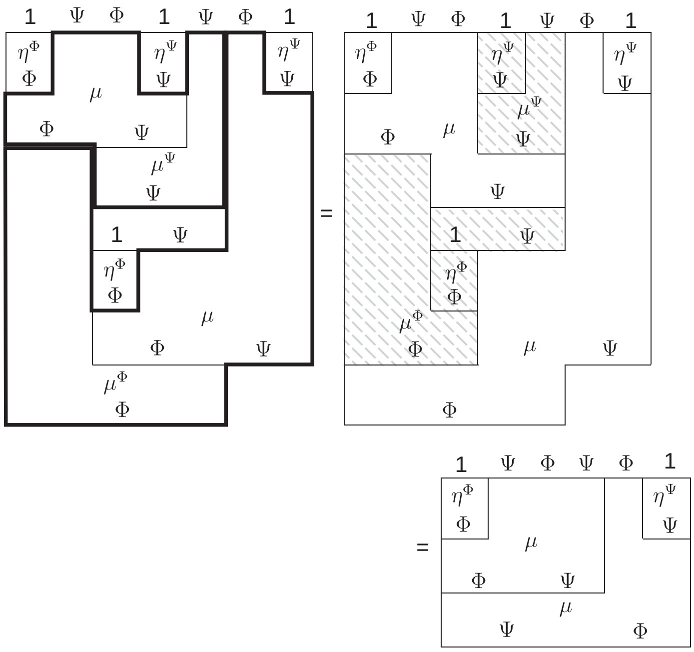
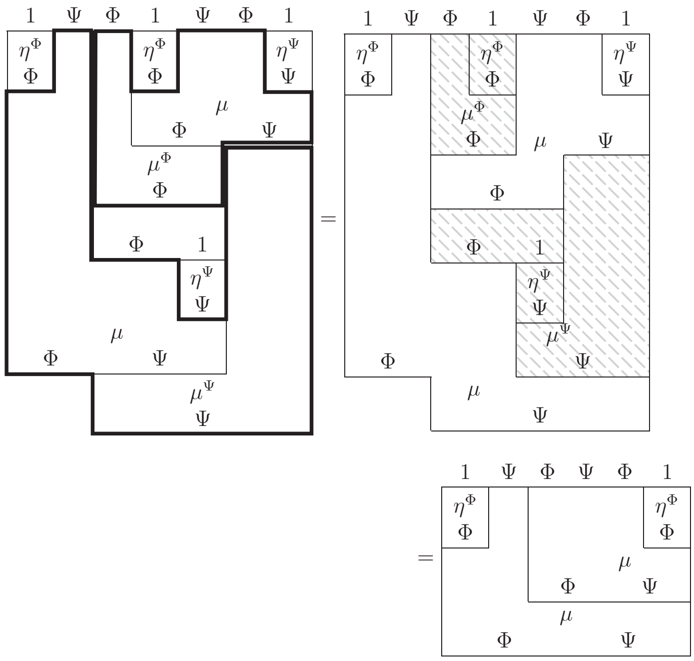


Fig. 5. Conditions (A) and (B) imply (1σ), part I.



■

Fig. 6. Conditions (A) and (B) imply (1σ), part II.

The theorem below shows that the two conditions (A) and (B) are equivalent to the three conditions given in Beck's Theorem 2. Once again, the graphical representation permits us to give a straightforward proof which avoids the naturality calculations.

Theorem 4 *Let* $\mathbf{\Phi} = (\Phi, \eta^\Phi, \mu^\Phi)$ *and* $\mathbf{\Psi} = (\Phi, \eta^\Psi, \mu^\Psi)$ *be two monads. If* $\mathbf{\Phi} \bullet \mathbf{\Psi} = (\Phi \circ \Psi, \eta^\Phi * \eta^\Psi, \mu)$ *is a monad, then conditions (R1), (R2) and (R3) hold if and only if conditions (A) and (B) hold.*

PROOF. Firstly, in Figure 7 we show that (A) and (B) imply the middle unitary law (R3), where equalities tagged with (1) follow from the addition or supression of identities (represented as dashed boxes); equality (2) follows from application of (A) to the bold framed box; and equality (3) follows similarly from application of (B).

The proof of '(A) and (B) imply (R1)' is given in Figure 8, where for equalities (1) and (3) the dashed parts correspond to identities; equality (2) is the application of (A) on the bold box; the proof for (R2) is similar.

Finally, in Figure 9 it is proved that (R1), (R2) and (R3) imply (A). The proof of (B) is similar. ■

In practical cases it is convenient to have a set of weak conditions which are simpler to check than the general necessary and suficient ones. For the converse

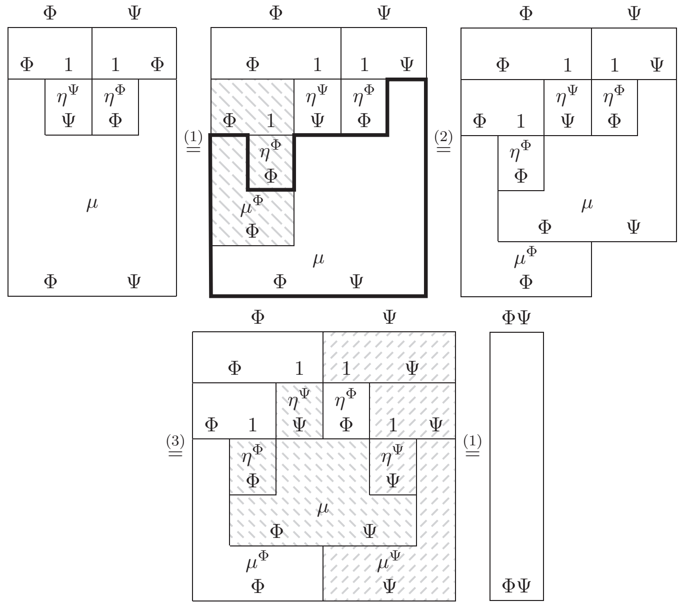


Fig. 7. (A) and (B) imply the middle unitary law.

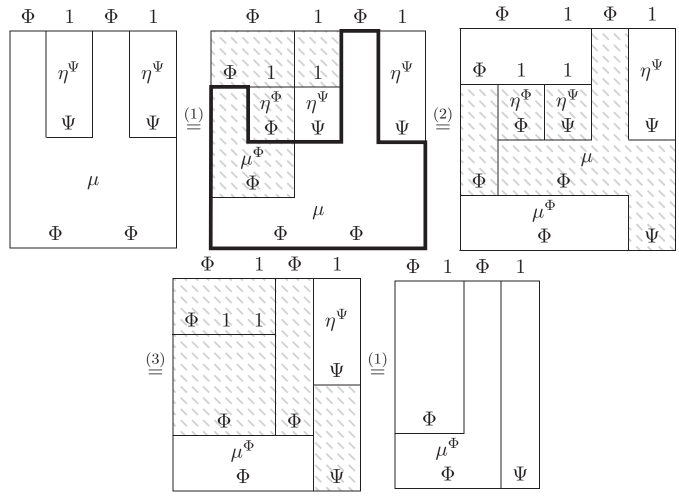


Fig. 8. (A) and (B) imply (R1).

theorem we are dealing with in this section, the following proposition gives as two easy simpler conditions.

Proposition 4 *Let* $\mathbf{\Phi} = (\Phi, \eta^{\Phi}, \mu^{\Phi})$ *and* $\mathbf{\Psi} = (\Phi, \eta^{\Psi}, \mu^{\Psi})$ *be two monads. If* $\mathbf{\Phi} \bullet \mathbf{\Psi} = (\Phi \circ \Psi, \eta^{\Phi} * \eta^{\Psi}, \mu)$ *is a monad, then conditions (A) and (B) are*

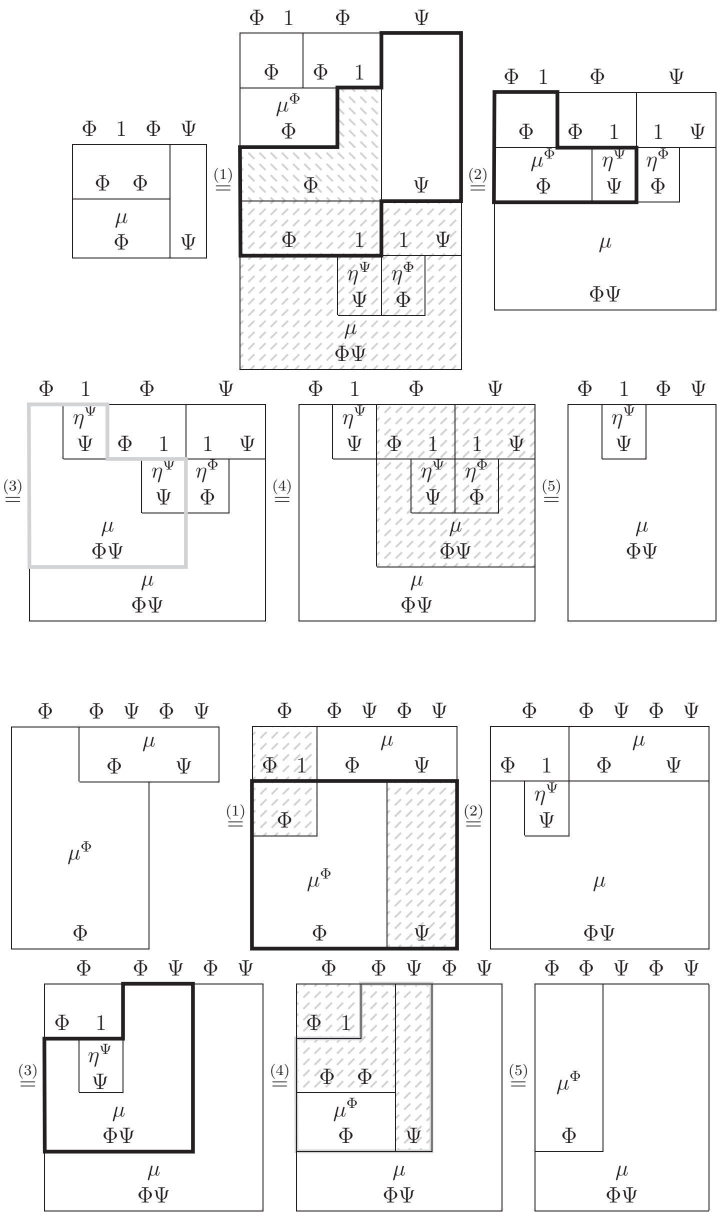
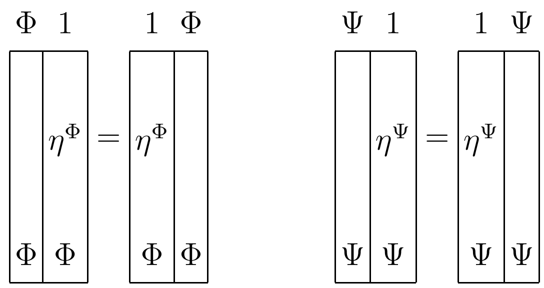


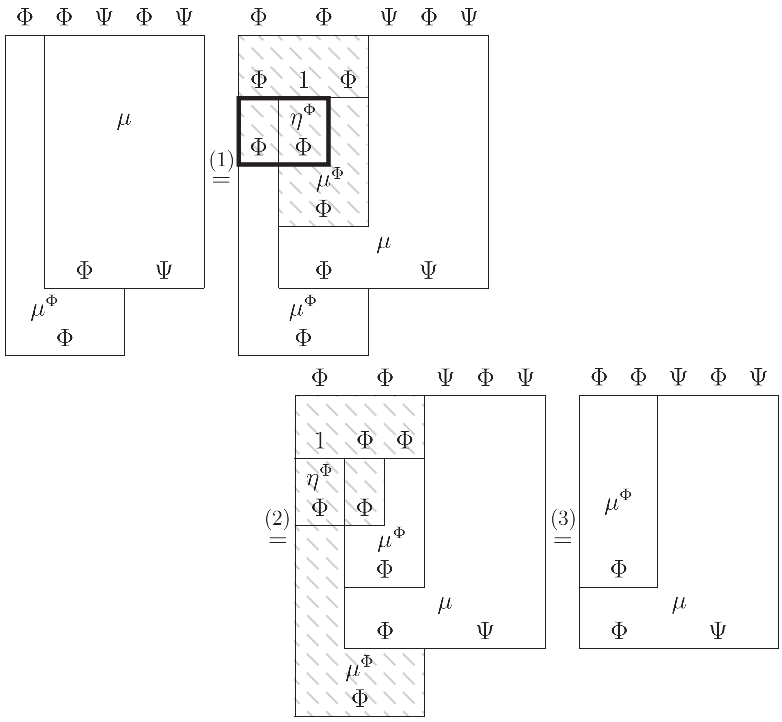
Fig. 9. (R1), (R2) and (R3) imply (A).

implied by the two conditions below



PROOF.

Let us prove (A), the case of (B) is similar



■

Example 1 *It is not difficult to check that the covariant powerset monad L and the term monad T satisfy Theorem 3. Actually, conditions in Proposition 4 are trivially satisfied since $L\eta^L = \eta^L L$ and $T\eta^T = \eta^T T$.*

7 Composing submonads

In this section we provide more general results on constructing new monads from given ones. In particular, in the case of a composition of two monads, the composition of submonads of respective original monads again provide monads under a rather general condition.

Definition 5 *Let Φ and Φ' be set functors. Functor Φ' is a* subfunctor *of Φ, written $\Phi' \leq \Phi$, if there is a natural transformation $e\colon \Phi' \to \Phi$, called the inclusion transformation, such that $e_X\colon \Phi' X \to \Phi X$ are inclusion maps, i.e., $\Phi' X \subseteq \Phi X$.*

The conditions on the subfunctor imply that $\Phi f_{|\Phi' X} = \Phi' f$ for all mappings $f\colon X \to Y$. Further, $\leq$ is a partial ordering.

Definition 6 *Let* $\mathbf{\Phi} = (\Phi, \eta, \mu)$ *be a monad over* `Set`*, and consider a subfunctor* Φ' *of* Φ*, with the corresponding inclusion transformation* $e\colon \Phi' \to \Phi$*, together with natural transformations* $\eta'\colon id \to \Phi'$ *and* $\mu'\colon \Phi'\Phi' \to \Phi'$ *satisfying the conditions*

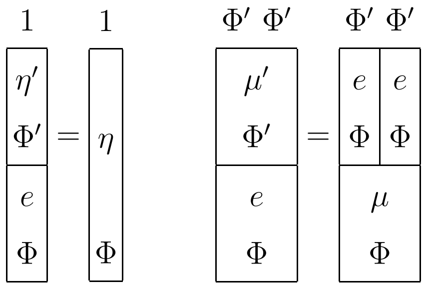
$$e \circ \eta' = \eta, \tag{7}$$
$$e \circ \mu' = \mu \circ (e \star e). \tag{8}$$

Then $\mathbf{\Phi'} = (\Phi', \eta', \mu')$ *is called the* submonad of $\mathbf{\Phi}$ defined by the subfunctor Φ'*, written* $\mathbf{\Phi'} \preceq \mathbf{\Phi}$.

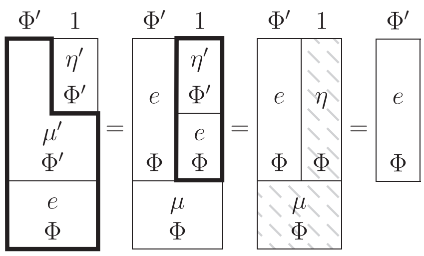
This definition actually defines a monad, as shown below.

Proposition 5 *A submonad* $\mathbf{\Phi'}$ *of a monad* $\mathbf{\Phi}$ *is a monad.*

PROOF. The pictorial representation of equations (7) and (8) respectively is given below



Let $\mathbf{\Phi'} = (\Phi', \eta', \mu')$ be a submonad of the monad $\mathbf{\Phi} = (\Phi, \eta, \mu)$, $\mathbf{\Phi'} \preceq \mathbf{\Phi}$. Let us verify that $\mathbf{\Phi'}$ indeed is a monad.



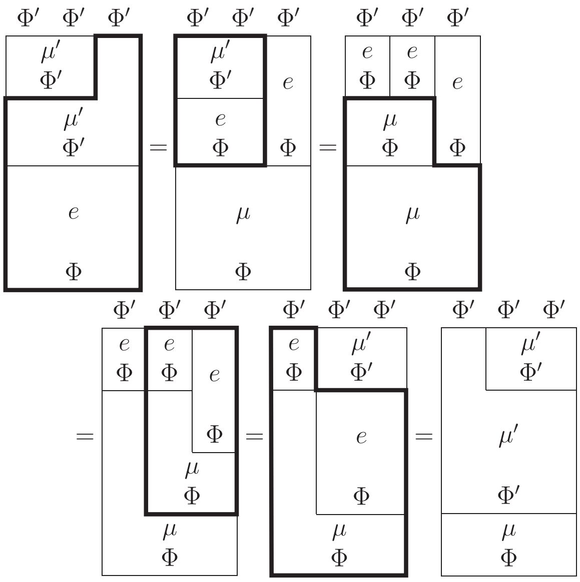
i.e. $\mu'_X \circ \Phi'\eta'_X = \mathrm{id}_{\Phi' X}$.

The proof of the other unit is similar, just to stress the usefulness of the pictorial approach we will prove it equationally:

$$
\begin{aligned}
e \circ \mu' \circ \eta'\Phi' &= \mu \circ \Phi e \circ e\Phi' \circ \eta'\Phi' \\
&= \mu \circ \Phi e \circ (e \circ \eta')\Phi' \\
&\overset{(7)}{=} \mu \circ \Phi e \circ \eta\Phi' \\
&= \mu \circ \eta\Phi \circ e \\
&= e,
\end{aligned}
$$

which shows, $\mu'_X \circ \eta'_{\Phi' X} = \mathrm{id}_{\Phi' X}$.

Finally, the associativity of the multiplication is also shown graphically:



and therefore, $\mu'_X \circ \Phi'\mu'_X = \mu'_X \circ \mu'_{\Phi' X}$. ∎

Proposition 6 $\preceq$ *is a partial ordering.*

PROOF. Reflexivity is obvious.

For antisymmetry, if $\mathbf{\Phi'} \preceq \mathbf{\Phi}$ and $\mathbf{\Phi} \preceq \mathbf{\Phi'}$, then $\Phi = \Phi'$, and corresponding inclusion transformations are identities, and in fact equal. Therefore, it follows that $\eta = \eta'$ and $\mu = \mu'$. Thus, $\mathbf{\Phi' = \Phi}$.

For transitivity, let $\mathbf{\Phi''} \preceq \mathbf{\Phi'}$ and $\mathbf{\Phi'} \preceq \mathbf{\Phi}$. Consider $e^* = e^{\Phi} \circ e^{\Phi'} \colon \Phi'' \to \Phi$, where $e^{\Phi} \colon \Phi' \to \Phi$, and $e^{\Phi'} \colon \Phi'' \to \Phi'$ are the transformations given by the subfunctor conditions. Then, $\Phi'' \le \Phi$, with e^* as the inclusion transformation.

Obviously, $e^* \circ \eta'' = \eta$, therefore

$$
\begin{array}{|c|}
\multicolumn{1}{c}{\Phi'' \ \Phi''}\\ \hline
\mu'' \\ \Phi'' \\ \hline
e^{\Phi'} \\ \Phi' \\ \hline
e^{\Phi} \\ \Phi \\ \hline
\end{array}
=
\begin{array}{|c|c|}
\multicolumn{2}{c}{\Phi'' \ \Phi''}\\ \hline
e^{\Phi'} & e^{\Phi'} \\ \Phi' & \Phi' \\ \hline
\multicolumn{2}{|c|}{\mu'} \\ \multicolumn{2}{|c|}{\Phi'} \\ \hline
\multicolumn{2}{|c|}{e^{\Phi}} \\ \multicolumn{2}{|c|}{\Phi} \\ \hline
\end{array}
=
\begin{array}{|c|c|}
\multicolumn{2}{c}{\Phi'' \ \Phi''}\\ \hline
e^{\Phi'} & e^{\Phi'} \\ \Phi' & \Phi' \\ \hline
e^{\Phi} & e^{\Phi} \\ \Phi & \Phi \\ \hline
\multicolumn{2}{|c|}{\mu} \\ \multicolumn{2}{|c|}{\Phi} \\ \hline
\end{array}
$$

that is, $e^* \circ \mu'' = \mu \circ \Phi e^* \circ e^* \Phi''$. Therefore, $\boldsymbol{\Phi''} \preceq \boldsymbol{\Phi}$. ∎

Given $\boldsymbol{\Phi'} \preceq \boldsymbol{\Phi}$ and $\boldsymbol{\Psi'} \preceq \boldsymbol{\Psi}$, we will now provide conditions under which compositions of the functors $\Phi' \circ \Psi$, $\Phi \circ \Psi'$ and $\Phi' \circ \Psi'$ can be extended to submonads of $\boldsymbol{\Phi} \circ \boldsymbol{\Psi}$.

Theorem 5 *Let $\boldsymbol{\Phi} \bullet \boldsymbol{\Psi} = (\Phi \circ \Psi, \eta^\Phi \star \eta^\Psi, (\mu^\Phi \star \mu^\Psi) \circ \Phi\sigma\Psi)$ be the monad given by Proposition 2 and let $\boldsymbol{\Phi'}$ and $\boldsymbol{\Psi'}$ be submonads of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively. If there exists a natural transformation $\sigma^* \colon \Psi'\Phi' \to \Phi'\Psi'$ such that $(e^\Phi \star e^\Psi) \circ \sigma^* = \sigma \circ (e^\Psi \star e^\Phi)$, that is*

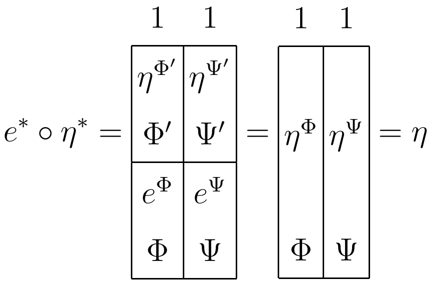
$$
\begin{array}{|c|c|}
\multicolumn{2}{c}{\Psi' \ \Phi'}\\ \hline
\multicolumn{2}{|c|}{\sigma^*} \\ \Phi' & \Psi' \\ \hline
e^{\Phi} & e^{\Psi} \\ \Phi & \Psi \\ \hline
\end{array}
=
\begin{array}{|c|c|}
\multicolumn{2}{c}{\Psi' \ \Phi'}\\ \hline
e^{\Psi} & e^{\Phi} \\ \Psi & \Phi \\ \hline
\multicolumn{2}{|c|}{\sigma} \\ \Phi' & \Psi' \\ \hline
\end{array}
$$

where e^Φ and e^Ψ are the mappings given by the submonad condition of Φ' and Ψ', respectively, then $\boldsymbol{\Phi'} \bullet \boldsymbol{\Psi'} = (\Phi' \circ \Psi', \eta^{\Phi'} \star \eta^{\Psi'}, (\mu^{\Phi'} \star \mu^{\Psi'}) \circ \Phi'\sigma^*\Psi')$ is a submonad of $\boldsymbol{\Phi} \bullet \boldsymbol{\Psi}$.

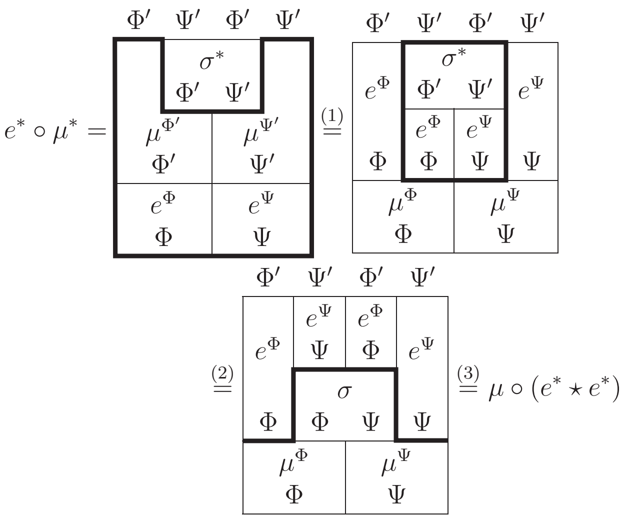
PROOF. Consider $\boldsymbol{\Phi} \bullet \boldsymbol{\Psi} = (\Phi \circ \Psi, \eta, \mu) = (\Phi \circ \Psi, \eta^\Phi \star \eta^\Psi, (\mu^\Phi \star \mu^\Psi) \circ \Phi\sigma\Psi)$ and $\boldsymbol{\Phi'} \bullet \boldsymbol{\Psi'} = (\Phi' \circ \Psi', \eta^*, \mu^*) = (\Phi' \circ \Psi', \eta^{\Phi'} \star \eta^{\Psi'}, (\mu^{\Phi'} \star \mu^{\Psi'}) \circ \Phi'\sigma^*\Psi')$.

Clearly $\Phi' \circ \Psi' \leq \Phi \circ \Psi$ since the inclusion transformation $e^* \colon \Phi' \circ \Psi' \to \Phi \circ \Psi$ is given by the star composition of the corresponding inclusion transformations for Φ' and Ψ' respectively, i.e., $e^* = e^\Phi \star e^\Psi$.

In the following we will verify the submonad conditions. Using the submonad conditions for Φ' and Ψ' we get,



and by the condition of σ^* and the naturality property, we get



■

Remark 2 *Under the conditions of Theorem 5, in the particular case that* $\Psi' = \Psi$*, if there exists a natural transformation* $\sigma'\colon \Psi\Phi' \to \Phi'\Psi$*, such that* $e^{\Phi}\Psi \circ \sigma' = \sigma \circ \Psi e^{\Phi}$*, where* e^{Φ} *is given by the submonad condition of* Φ'*, then* $\mathbf{\Phi'} \bullet \mathbf{\Psi} = (\Phi' \circ \Psi, \eta^{\Phi'} \star \eta^{\Psi}, (\mu^{\Phi'} \star \mu^{\Psi}) \circ \Phi'\sigma'\Psi)$ *is a submonad of* $\mathbf{\Phi} \bullet \mathbf{\Psi}$*. A similar observation can be made in the case of* $\Phi' = \Phi$*.*

Example 2 *Let* K *and* L *be completely distributive lattices. Assume* K *to be a sublattice of* L*, with* $\iota\colon K \to L$ *being the inclusion homomorphism. Further, assume* $\iota(0) = 0$ *and* $\iota(1) = 1$*, and additionally, that* $\iota(\vee_i x_i) = \vee_i \iota(x_i)$ *also in the non-finite case.*

Define $(\iota_{id})_X\colon K_{id}X \to L_{id}X$ *by* $(\iota_{id})_X(A) = \iota \circ A$*,* $A\colon X \to K$*. It is easily checked that* $\iota_{id}\colon K_{id} \to L_{id}$ *becomes a natural transformation, and that* $\boldsymbol{K}_{id}$ *is a submonad of* $\boldsymbol{L}_{id}$*.*

Further, by Remark 2, it is straightforward to show that $\boldsymbol{K}_{id} \bullet \boldsymbol{T}_{\Omega}$ *is a submonad* *of* $\boldsymbol{L}_{id} \bullet \boldsymbol{T}_\Omega$.

Example 3 *In [7] we defined functors for α-upper L-fuzzy sets and α-lower L-fuzzy sets, denoted $\mathcal{L}_\alpha$ and $\mathcal{L}^\alpha$, respectively, given as follows:*

$$\begin{aligned}\mathcal{L}_\alpha X &= \{A \in L_{id}X \mid A(x) \geq \alpha \text{ or } A(x) = 0, \text{ for all } x \in X\}\\ \mathcal{L}^\alpha X &= \{A \in L_{id}X \mid A(x) \leq \alpha \text{ or } A(x) = 1, \text{ for all } x \in X\}.\end{aligned}$$

For mappings $f\colon X \to Y$, we could obtain $\mathcal{L}_\alpha f$ and $\mathcal{L}^\alpha f$, respectively, as $L_{id}f_{|\mathcal{L}_\alpha X}$ and $L_{id}f_{|\mathcal{L}^\alpha X}$. Thus, $\mathcal{L}_\alpha$ and $\mathcal{L}^\alpha$ become subfunctors of L_{id}.

Further, $\mathcal{L}_\alpha$ and $\mathcal{L}^\alpha$ were extended to monads with $\eta^{\mathcal{L}_\alpha}$ and $\eta^{\mathcal{L}^\alpha}$ defined using (5), and additionally $\mu^{\mathcal{L}_\alpha}$ and $\mu^{\mathcal{L}^\alpha}$ defined using (6). It is now not difficult to show that $\boldsymbol{\mathcal{L}}_\alpha = (\mathcal{L}_\alpha, \eta^{\mathcal{L}_\alpha}, \mu^{\mathcal{L}_\alpha})$ and $\boldsymbol{\mathcal{L}}^\alpha = (\mathcal{L}^\alpha, \eta^{\mathcal{L}^\alpha}, \mu^{\mathcal{L}^\alpha})$ are submonads of $\boldsymbol{L}_{id}$.

By Remark 2, it follows that $\boldsymbol{\mathcal{L}}_\alpha \bullet \boldsymbol{T}_\Omega$ and $\boldsymbol{\mathcal{L}}^\alpha \bullet \boldsymbol{T}_\Omega$ are submonads of $\boldsymbol{L}_{id} \bullet \boldsymbol{T}_\Omega$.

Example 4 *Let Ω' and Ω be operator domains with $\Omega' \subseteq \Omega$, and let $\epsilon\colon \Omega' \to \Omega$ be the inclusion mapping.*

Define $\nu_X\colon T_{\Omega'}X \to T_\Omega X$ by $\nu_X(x) = x$, $x \in X$, and

$$\nu_X((n, \omega', (t'_i)_{i \leq n})) = (n, \epsilon(\omega'), (\nu_X(t'_i))_{i \leq n})$$

for $t'_i \in T_{\Omega'}X$. It is easily seen that $\nu\colon T_{\Omega'} \to T_\Omega$ is a natural transformation (actually, it is an inclusion) and that $\boldsymbol{T}_{\Omega'}$ is a submonad of $\boldsymbol{T}_\Omega$.

Again, by Remark 2, it is easy to verify that $\boldsymbol{L}_{id} \bullet \boldsymbol{T}_{\Omega'}$ is a submonad of $\boldsymbol{L}_{id} \bullet \boldsymbol{T}_\Omega$.

8 Conclusions

It is important to stress the non-triviality of providing monad compositions. An apparently useful composition of functors in corresponding monads can turn out not to be extendable to a monad. In fact, as there are no general methods on how to provide monad compositions, it is not clear if extendability of such compositions of functors to a monad composition is more a rule than an exception. Further, proofs of compliance with composability conditions, e.g. in connection with distributive laws, tend to be rather complicated as the complexity of the functors increase. A graphical proof technique to proving compliance with composability conditions has shown to provide a more mechanised approach to handling complicated constructions involving calculations with natural transformations.

For applications, our focus is on composing powerset monads with the term monad, in order to develop a concept for generalised terms. A first step towards similarities between powersets of terms was done in [8] involving fuzzy relations for crisp powersets of terms. Using proof techniques presented in this paper, a range of interesting powerset functors can be further investigated. It is expected that this approach provides an appropriate formal framework for useful developments of generalised terms as a basis for many-valued logic programming involving an extended notion of terms. Within the scope of many-valued logic programming, this opens up for further work on using categorical approaches to unification. The theoretical treatment will in addition benefit from a presentation and investigation within a 2-category framework.

Acknowledgements We are grateful to J.M. Barja, W. Gähler and A. Pultr for discussions and helpful suggestions.

References

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

[2] Barr, M. and C. Wells: 1985, *Toposes, Triples and Theories*. Springer.

[3] Barr, M. and C. Wells: 1990, *Category Theory for Computing Science*. Prentice Hall.

[4] Beck, J.: 1969, 'Distributive Laws'. In: *Seminar on Triples and Categorical Homology Theory*, No. 80 in Lecture Notes in Mathematics. Springer-Verlag, pp. 119–130.

[5] Caruncho Castro, J.: 1971, 'Triples theory'. Technical Report 5, Department of Algebra of the Univ. de Santiago de Compostela. In Spanish.

[6] Eklund, P. and W. Gähler: 1992, 'Fuzzy filter functors and convergence'. In: S. E. Rodabaugh et al. (ed.): *Applications of category theory to fuzzy subsets*, Theory and Decision Library B. Kluwer, pp. 109–136.

[7] Eklund, P., M. A. Galán, J. Medina, M. Ojeda-Aciego, and A. Valverde: 2002, 'Set functors, L-fuzzy set categories and generalized terms'. *Computers and Mathematics with Applications* **43**, 693–705.

[8] Eklund, P., M. A. Galán, J. Medina, M. Ojeda-Aciego, and A. Valverde: 2004, 'Similarities between powersets of terms'. *Fuzzy Sets and Systems* **144**, 213–225.

[9] Eklund, P., M. A. Galán, M. Ojeda-Aciego, and A. Valverde: 2000, 'Set functors and generalised terms'. In: *Proc. of IPMU 2000*, Vol. vol. III of *Information Processing and Management of Uncertainty in Knowledge-based Systems*. pp. 1595–1599.

[10] Freire Nistal, J.: 1971, 'Universal properties in higher order triples'. Technical Report 11, Department of Algebra of the Univ. de Santiago de Compostela. In Spanish.

[11] Gähler, W.: 1988, 'Monads and convergence'. In: *Proc. Conference Generalized Functions, Convergences Structures, and Their Applications*. pp. 29–46.

[12] Goguen, J.: 1967, 'L-fuzzy sets'. *J. Math. Anal. Appl.* **18**, 145–174.

[13] Gray, J. W.: 1974, *Formal Category Theory: Adjointness for 2-Categories*, No. 391 in Lecture Notes in Mathematics. Springer-Verlag.

[14] Kelly, G.: 1974, 'Coherence theorems for lax algebras and for distributive laws'. In: *Category Theory: Proceedings Sydney Category Theory Seminar 1972/1973*, No. 420 in Lecture Notes in Mathematics. Springer-Verlag, pp. 281–375.

[15] Manes, E.: 1976, *Algebraic Theories*. Springer-Verlag.

[16] Marmolejo, F.: 1999, 'Distributive laws for pseudomonads'. *Theory and Applications of Categories* **5**(5), 91–147.

[17] Marmolejo, F., R. Rosebrugh, and R. Wood: 2002, 'A basic distributive law'. *Journal of Pure and Applied Algebra* **168**, 209–226.

[18] Moggi, E.: 1991, 'Notions of computation and monads'. *Information and Computation* **93**(1), 55–92.

[19] Rydeheard, D. and R. Burstall: 1986, 'A categorical unification algorithm'. In: *Proc. Category Theory and Computer Programming*. pp. 493–505.

[20] Street, R.: 1972, 'The formal theory of monads'. *Journal of Pure and Applied Algebra* **2**, 149–168.

[21] Wadler, P.: 1992, 'Comprehending monads'. *Mathematical Structures in Computer Science* **2**, 461–493.