Categorical Innovations for Rough Sets

P. Eklund[1], M.A. Galán[2,⋆], and J. Karlsson[3]

[1] Department of Computing Science, Umeå University, Sweden
peklund@cs.umu.se
[2] Department of Applied Mathematics, University of Málaga, Spain
magalan@ctima.uma.es
[3] Department of Computer Architecture, University of Málaga, Spain
johan@ac.uma.es

Summary. Categories arise in mathematics and appear frequently in computer science where algebraic and logical notions have powerful representations using categorical constructions. In this chapter we lean towards the functorial view involving natural transformations and monads. Functors extendable to monads, further incorporating order structure related to the underlying functor, turn out to be very useful when presenting rough sets beyond relational structures in the usual sense. Relations can be generalized with rough set operators largely maintaining power and properties. In this chapter we set forward our required categorical tools and we show how rough sets and indeed a theory of rough monads can be developed. These rough monads reveal some canonic structures, and are further shown to be useful in real applications as well. Information within pharmacological treatment can be structured by rough set approaches. In particular, situations involving management of drug interactions and medical diagnosis can be described and formalized using rough monads.

1 Introduction

Monads are useful e.g. for generalized substitutions as we have extended the classical concept of a term to a many-valued set of terms [21]. This builds essentially upon composing various set functors, as extendable to monads, with the term functor and its corresponding monad. The most trivial set functor is the powerset functor for which a substitution morphism in the corresponding Kleisli category is precisely a relation. Thus relations are seen as connected to a powerset functor that can be extended to a monad. Further, whenever general powerset monads can be extended to partially ordered monads, this structure is sufficient for the provision of rough set operations in a category theory setting. This categorical presentation of rough sets will establish connections to other categorical structures with the objective to enrich the theory. Key in these constructions is the first observation that relations are morphisms in the Kleisli category of the monad extended from the powerset functor.

Fuzzy sets, closely related to rough sets, are founded on the notion of many-valued membership, and is considered as a gradual property for fuzzy sets. Fuzzy

⋆ Partially supported by Spanish projects P06-FQM-02049 and TIN2006-15455-C03-01.

A. Abraham, R. Falcón, and R. Bello (Eds.): Rough Set Theory, SCI 174, pp. 45–69.
springerlink.com

set theory offers a more expressive mathematical language and has many applications in a very wide variety of fields. Fuzzy sets, originally introduced by Zadeh [39] in 1965, increase the expressiveness of classical mathematics to deal with information systems that are incomplete, uncertain and imprecise. In 1967, Goguen, [22], extended the idea of fuzzy sets to L-fuzzy sets, considering more general order structures L beyond the unit interval. The extended notion of L-fuzzy sets also represent the extension of the crisp powerset monad. Thus the powerset monads is the categorical way to represent fuzzy sets. Beyond fuzzy sets, and introducing partial order into monads, using so called *partially ordered monads*, we can then also represent rough sets.

The outline of the chapter is as follows. Section 2 gives a historical background to the categorical apparatus used in the chapter. In sect. 3 we provide the categorical preliminaries and notations making the chapter easier to read and comprehend. Section 4 includes important examples of monads, one of the underlying categorical tools used in this chapter. Section 5 then describes partially ordered monads upon which rough monads are built in Sect. 7. In sect. 6, rough sets are conceptually embedded into the categorical machinery. Section 8 outlines applications related to management of drug interactions and cognitive disorder diagnosis, respectively. Section 9 concludes the chapter.

2 Historical Remarks and Related Work

Monads were initiated by Godement around 1958. Huber shows in 1961 that adjoint pairs give rise to monads. Kleisli [28] and also Eilenberg and Moore [4] proves the converse in 1965. Kleisli categories were explicitely constructed in those contributions. Partially ordered monads are monads [32], where the underlying endofunctor is equipped with an order structure, which makes them useful for various generalized topologies and convergence structures [18, 20]. They are indeed derived from studies on convergence, initiated by [30]. Partially ordered monads were initially studied in [18, 19]. Topology and convergence were forerunners in the development of partially ordered monads, but these monads contain structure also for modelling rough sets [33] in a generalized setting with set functors. Partially ordered monads contribute to providing a generalised notion of powerset Kleene algebras [5]. This generalisation builds upon a more general powerset functor setting far beyond just strings [27] and relational algebra [37]. These structures are typical representatives of Kleene algebras, which are widely used e.g. in formal languages [36] and analysis of algorithms [1]. Rough sets and their purely algebraic properties are studied e.g. within shadowed sets [3]. There is further an interesting interaction between monads and algebras, which is well-known. The tutorial example is the isomorphism between the Kleisli category of the powerset monad and the category of 'sets and relations'. The Eilenberg-Moore category of the powerset monad is isomorphic to the category of complete lattices and join-preserving mappings. The Kleisli category of the term monad coincides with its Eilenberg-Moore category and is isomorphic to the category of Ω-algebras. A rather intrepid example, although still folklore, is the isomorphism between the Eilenberg-Moore category of the ultrafilter monad and the category of compact Hausdorff spaces. Here is where "algebra and topology meet".

3 Categorical Preliminaries

A major advantage of category theory is its 'power of abstraction' in the sense that many mathematical structures can be characterized in terms of relatively few categorical ones. This fact enables to pursue a more general study towards generalizations of the structures. Category theory has been successfully applied in different areas such as topology, algebra, geometry or functional analysis. In recent years, category theory has also contributed to the development of computer science: the abstraction of this theory has brought the recognition of some of the constructions as categories. This growing interest towards categorical aspects can be found in, for instance, term rewriting systems, game semantics and concurrency. In a gross manner one can say a *category* is given by a class of *object* and a class of *morphisms* between the objects under certain mathematical conditions. Examples of categories come not only from mathematics (the category of groups and group homomorphisms, the category of topological spaces and continuous functions, etc.) but also from computer science. Deductive systems is a category where the objects are formulas and morphisms are proofs. Partially ordered sets form a category where objects are partially ordered sets and morphisms are monotone mappings. A particular partially ordered set also forms a category where objects are its elements and there is exactly one morphism from an element x to an element y if and only if $x \leq y$. We can go beyond categories and wonder if there is a category of categories. The answer is yes (provided the underlying selected set theory is properly respected). In this category of categories the objects are categories and the morphisms are certain structure-preserving mappings between categories, called functors. Examples of functors are for instance the list functor, the powerset functor and the term functor. The concept of naturality is important in many of the applications of category theory. Natural transformations are certain structure-preserving mappings from one functor to another. It might seem abstract to consider morphisms between morphisms of categories, but natural transformations appear in a natural way very frequently both in mathematics as well as in computer science. Natural transformations are cornerstones in the concept of monads.

3.1 Categories

A *category* $\mathtt{C}$ consists of *objects*, and for each pair (A, B) of objects we have *morphisms* f from A to B, denoted by $f : A \longrightarrow B$ or $A \xrightarrow{f} B$. Further there is an (A-)identity morphism $A \xrightarrow{id_A} A$ and a composition $\circ$ among morphisms that composes $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ to $A \xrightarrow{g \circ f} C$ in order to always guarantee $h \circ (g \circ f) = (h \circ g) \circ f$, and also $id_B \circ f = f \circ id_A = f$ for any $A \xrightarrow{f} B$. The set of $\mathtt{C}$-morphisms from A to B is written as $Hom_{\mathtt{C}}(A, B)$ or $Hom(A, B)$.

Example 1. The category of sets, `Set`, consists of sets as objects and functions as morphisms together with the ordinary composition and identity.

Example 2. The category of partially ordered sets, `Poset`, consists of partially ordered sets as objects and order-preserving functions as morphisms. The category of boolean algebras, `Boo`, consists of boolean algebras as objects and boolean homomorphisms as morphisms. The category of groups, `Grp`, consists of groups as objects and group homomorphisms as morphisms.

Example 3. A poset (partially ordered set) forms also a category where objects are its elements and there is exactly one morphism from an element x to an element y if and only if $x \leq y$. Composition is forced by transitivity. Identity is forced by reflexivity.

Example 4. The category of Ω-algebras, $\mathtt{Alg}(\Omega)$, consists of Ω-algebras as objects and Ω-homomorphisms as morphisms between them. Recall that a Ω-algebra is a pair $(X, (\omega_i)_{i\in I})$ where X is a set and $\omega_i : X^{n_i} \longrightarrow X$ are the n_i-ary operations on X. An Ω-homomorphism $f : (X, (\omega_i)_{i\in I}) \longrightarrow (\hat{X}, (\hat{\omega}_i)_{i\in I})$ consists of a function $f : X \longrightarrow \hat{X}$ such that the diagram

$$\begin{array}{ccc} X^{n_i} & \xrightarrow{f^{n_i}} & \hat{X}^{n_i} \\ {\scriptstyle \omega_i}\downarrow & & \downarrow{\scriptstyle \hat{\omega}_i} \\ X & \xrightarrow[f]{} & \hat{X} \end{array}$$

commutes, i.e. $f(\omega_i(x_1, \ldots, x_{n_i})) = \hat{\omega}_i(f(x_1), \ldots, f(x_{n_i}))$.

Further examples are the category of groups and group homomorphisms, the category of vector spaces and linear transformations, and the category of topological spaces with continuous functions as morphisms. A useful category in computer science is the following.

3.2 Functors

To relate category theory with typed functional programming, we identify the objects of the category with types and the morphisms between the objects with functions. In a given category, the set of morphisms $Mor(\mathcal{C})$ is useful to establish connections between the different objects in the category. But also it is needed to define the notion of a transformation of a category into another one. This kind of transformation is called a *functor*. In the previous context, functors are not only transformations between types, but also between morphisms, so, at the end, they will be mappings between categories. Let us see an example. Given a type, for instance, *Int*, we can consider the linear finite list type of elements of this type, *integer lists*. Let us denote by *List(S)* to indicate the lists of elements with type S. Let us see how *List* actuates not only over types, but also over functions between types. Given a function $f : S \longrightarrow T$ we want to define a function $List(f)$,

$$List(f) : List(S) \longrightarrow List(T).$$

Note that here we are using the same name for two operations, one over objects and the other one over functions. This is the standard when using functors. To understand how to define *List* over functions, let us consider the function $sq : Int \longrightarrow Int$ defined as $sq(x) = x^2$. The type of $List(sq)$ is $List(sq) : List(Int) \longrightarrow List(Int)$. What should be the value of $List(sq)[-2, 1, 3]$? The obvious answer is the list $[(-2)^2, 1^2, 3^2] = [4, 1, 9]$. In the general case, the part that actuates over the morphisms of *List* is the *maplist* function, that distributes a function over the elements of a list. In this case we have defined how *List* actuates over objects and morphisms. Next step is to ask ourselves how does *List* respect the categorical structure, that is, what is the behavior over the composition of morphisms and over the identity morphism? It is expected that

$$List(g \circ f) = List(g) \circ List(f),$$
$$List(id_a) = id_{List(a)}.$$

It is not difficult to check this for *maplist*. Now we are ready for the functor definition. Let $\mathtt{C}$ and $\mathtt{D}$ be categories. A (covariant) *functor* φ from $\mathtt{C}$ to $\mathtt{D}$, denoted $\varphi : \mathtt{C} \longrightarrow \mathtt{D}$, is a mapping that assigns each $\mathtt{C}$-object A to a $\mathtt{D}$-object $\varphi(A)$ and each $\mathtt{C}$-morphism $A \xrightarrow{f} B$ to a $\mathtt{D}$-morphism $\varphi(A) \xrightarrow{\varphi(f)} \varphi(B)$, such that $\varphi(f \circ g) = \varphi(f) \circ \varphi(g)$ and $\varphi(id_A) = id_{\varphi(A)}$. We often write φA and φf instead of $\varphi(A)$ and $\varphi(f)$. For functors $\varphi : \mathtt{C} \longrightarrow \mathtt{D}$ and $\psi : \mathtt{D} \longrightarrow \mathtt{E}$, we can easily see that the composite functor $\psi \circ \varphi : \mathtt{C} \longrightarrow \mathtt{E}$ given by

$$(\psi \circ \varphi)(A \xrightarrow{f} A') = \psi(\varphi A) \xrightarrow{\psi(\varphi f)} \psi(\varphi A')$$

indeed is a functor.

Example 5. Any category $\mathtt{C}$ defines an identity functor $id_{\mathtt{C}} : \mathtt{C} \longrightarrow \mathtt{C}$ given by

$$id_{\mathtt{C}}(A \xrightarrow{f} B) = A \xrightarrow{f} B.$$

Example 6. The (covariant) powerset functor $P : \mathtt{Set} \longrightarrow \mathtt{Set}$ is defined by PA being the powerset of A, i.e. the set of subsets of A, and $Pf(X)$, for $X \subseteq A$, being the image of X under f, i.e. $Pf(X) = \{f(x) \mid x \in X\}$. The contravariant powerset functor $Q : \mathtt{Set}^{op} \longrightarrow \mathtt{Set}$ is defined by QA again being the powerset of A, and further

$$Q(A \xrightarrow{f} B) = QA \xrightarrow{Qf} QB$$

where $Qf(X)$, $X \subseteq A$, is the inverse image of X under the function $f : B \longrightarrow A$.

Example 7. The list functor $List : \mathtt{Set} \longrightarrow \mathtt{Set}$ is defined by $List(A)$ being the set of finite lists with elements in A, i.e. $List(A) = \bigcup_{n \in \mathbf{N}} A^n$, and further for $f : A \longrightarrow B$ we have

$$List f(L) = [f(a_1), \ldots, f(a_n)]$$

for finite lists $L = [a_1, \ldots, a_n]$ with $a_1, \ldots, a_n \in A$.

A functor $\varphi : \mathtt{C} \longrightarrow \mathtt{D}$ is a (functor) isomorphism if there is a functor $\psi : \mathtt{D} \longrightarrow \mathtt{C}$ such that $\psi \circ \varphi = id_{\mathtt{C}}$ and $\varphi \circ \psi = id_{\mathtt{D}}$.

Example 8. The category $\mathtt{Boo}$ is isomorphic to the category of boolean rings (ring with unit, and each element being idempotent with respect to multiplication, i.e. $a \cdot a = a$) and ring homomorphisms.

Example 9 (T-algebras). Let $T : \mathtt{X} \longrightarrow \mathtt{X}$ be a functor. A T-algebra is a pair (X, h), where X is an $\mathtt{X}$-object and $h : TX \longrightarrow X$ is an $\mathtt{X}$-morphism. A T-homomorphism $f : (X, h) \longrightarrow (X', h')$ between T-algebras is an $\mathtt{X}$-morphism $f : X \longrightarrow X'$ such that the diagram

$$\begin{array}{ccc} TX & \xrightarrow{\;h\;} & X \\ {\scriptstyle Tf}\big\downarrow & & \big\downarrow{\scriptstyle f} \\ TX' & \xrightarrow[\;h'\;]{} & X' \end{array}$$

commutes. We denote by $\mathtt{Alg}(T)$ the category consisting of T-algebras as $\mathtt{Alg}(T)$-objects and T-homomorphisms as $\mathtt{Alg}(T)$-morphisms. It can be shown that $\mathtt{Alg}(\Omega)$ (Example 4) is isomorphic to $\mathtt{Alg}(T)$ for some suitable functors T.

Example 10. If $\mathcal{G}$ and $\mathcal{H}$ are groups considered as categories with a single object, then a functor from $\mathcal{G}$ to $\mathcal{H}$ is exactly a group homomorphism.

Example 11. If $\mathcal{P}$ and $\mathcal{Q}$ are posets, a functor from $\mathcal{P}$ to $\mathcal{Q}$ is exactly a nondecreasing map.

Example 12. The list functor $List : \mathtt{Set} \longrightarrow \mathtt{Set}$ ($\mathtt{Set}$ denotes the category of sets) is defined by $List(A)$ being the set of finite lists with elements in A, i.e. $List(A) = \bigcup_{n \in \mathbf{N}} A^n$, and further for $f : A \longrightarrow B$ we have

$$List\, f(L) = [f(a_1), \ldots, f(a_n)]$$

for finite lists $L = [a_1, \ldots, a_n]$ with $a_1, \ldots, a_n \in A$.

3.3 Natural Transformations

In the same way as functors are defined as morphisms between categories, we could think of defining morphisms between functors. The concept of naturality is central in many of the applications of category theory. *Natural transformations* are certain structure-preserving mappings from one functor to another. Maybe, in a first approach to this, it seems abstract to consider morphisms between morphisms of categories. We will show here, how natural transformations appear in a natural way not only in mathematics, but also in programming. Continuing with lists, let us consider that the function that inverts lists, has type $rev : List(S) \longrightarrow List(S)$ where S is a type. Obviously, it is expected that rev inverts any kind of lists, i.e., it is expected that the definition rev is uniform with respect to the type of the elements on the list. One definition of *rev* can be given in the functional program Hope as follows:

- $rev(nil) <= nil$
- $rev(a :: l) <= rev(l) :: a$

Instead of considering *rev* as a list whose type is polymorphic, we can consider it as a collection of functions indexed by the element's type of the list, so we could write $rev_S : List(S) \longrightarrow List(S)$ or even, $rev : List \longrightarrow List$. In the last case, we apply the argument S, that is the type of the elements on the list, to *rev* and to the functor *List*. Note that when we apply *rev* to an argument, we get a function, in this way we can consider *rev* as a morphism from the functor *List* to the functor *List*. In this context we must make sure that, for types S and T, the functions rev_S and rev_T are well related to each other. The relation between these two mappings can be expressed through the commutativity of the following diagram

$$\begin{array}{ccccc} S & & List(S) & \xrightarrow{rev_S} & List(S) \\ \Big\downarrow f & & \Big\downarrow List(f) & & \Big\downarrow List(f) \\ T & & List(T) & \xrightarrow[rev_T]{} & List(T) \end{array}$$

In this case, the action of *List* over functions is the function *maplist*. It is easy to check that the diagram commutes, and for any $f : S \longrightarrow T$, it expresses a fundamental property of the function *rev*. Now, we can give the definition of natural transformation. Let $\varphi, \psi : \mathtt{C} \longrightarrow \mathtt{D}$ be functors.

Definition 1. *A* natural transformation τ *from* φ *to* ψ*, written* $\tau : \varphi \longrightarrow \psi$ *or* $\varphi \xrightarrow{\tau} \psi$*, assigns to each* $\mathtt{C}$*-object* A *a* $\mathtt{D}$*-morphism* $\tau_A : \varphi A \longrightarrow \psi A$ *such that the diagram*

$$\begin{array}{ccccc} A & & \varphi A & \xrightarrow{\tau_A} & \psi A \\ \Big\downarrow f & & \Big\downarrow \varphi f & & \Big\downarrow \psi f \\ A' & & \varphi A' & \xrightarrow[\tau_{A'}]{} & \psi A' \end{array}$$

commutes.

Let φ be a functor. The identity natural transformation $\varphi \xrightarrow{id_\varphi} \varphi$ is defined by $(id_\varphi)_A = id_{\varphi A}$. For functors φ and natural transformations τ we often write $\varphi\tau$ and $\tau\varphi$, respectively, to mean $(\varphi\tau)_A = \varphi\tau_A$ and $(\tau\varphi)_A = \tau_{\varphi A}$. It is easy to see that $\eta : id_{\mathtt{Set}} \longrightarrow P$ given by $\eta_X(x) = \{x\}$, and $\mu : P \circ P \longrightarrow P$ given by $\mu_X(\mathcal{B}) = \bigcup \mathcal{B} (= \bigcup_{B \in \mathcal{B}} B)$ are natural transformations. Natural transformations can be composed *vertically* as well as *horizontally*. Let $\varphi, \psi, \vartheta : \mathtt{C} \longrightarrow \mathtt{D}$ be functors and let further $\varphi \xrightarrow{\tau} \psi$ and $\psi \xrightarrow{\sigma} \vartheta$ be natural transformations. The (vertical) *composition* $\varphi \xrightarrow{\sigma \circ \tau} \vartheta$, defined by $(\sigma \circ \tau)_A = \sigma_A \circ \tau_A$, is a natural transformation. In order to define the corresponding horizontal composition, let $\varphi', \psi' : \mathtt{C} \longrightarrow \mathtt{D}$ be functors and let $\varphi' \xrightarrow{\tau'} \psi'$ be a natural transformation. The *star product* (horizontal composition) $\varphi' \circ \varphi \xrightarrow{\tau' \star \tau} \psi' \circ \psi$ is defined by

$$\tau' \star \tau = \tau' \psi \circ \varphi' \tau = \psi' \tau \circ \tau' \varphi. \tag{1}$$

For the identity transformation $id_\varphi \colon \varphi \longrightarrow \varphi$, also written as 1_φ or 1, we have

$$1_\varphi \star 1_\psi = 1_{\varphi \circ \psi}. \tag{2}$$

For a natural transformation $\tau \colon \varphi \longrightarrow \psi$, and a functor ϑ, $\vartheta\tau = 1_\vartheta \star \tau$ and $\tau\vartheta = \tau \star 1_\vartheta$. For natural transformations $\varphi \xrightarrow{\tau} \psi \xrightarrow{\sigma} \vartheta$ and $\varphi' \xrightarrow{\tau'} \psi' \xrightarrow{\sigma'} \vartheta'$ we have the Interchange Law $(\sigma' \circ \tau') \star (\sigma \circ \tau) = (\sigma' \star \sigma) \circ (\tau' \star \tau)$.

3.4 Monads and Kleisli Categories

In the following we include some formal definitions of concepts required.

Definition 2. *Let* $\mathtt{C}$ *be a category. A monad (or triple, or algebraic theory) over* $\mathtt{C}$ *is written as* $\mathbf{\Phi} = (\varphi, \eta, \mu)$*, where* $\varphi : \mathtt{C} \to \mathtt{C}$ *is a (covariant) functor, and* $\eta : id \to \varphi$ *and* $\mu : \varphi \circ \varphi \to \varphi$ *are natural transformations for which* $\mu \circ \varphi\mu = \mu \circ \mu\varphi$ *and* $\mu \circ \varphi\eta = \mu \circ \eta\varphi = id_\varphi$ *hold.*

Definition 3. *A Kleisli category* $\mathtt{C}_{\mathbf{\Phi}}$ *for a monad* $\mathbf{\Phi}$ *over a category* $\mathtt{C}$ *is given with objects in* $\mathtt{C}_{\mathbf{\Phi}}$ *being the same as in* $\mathtt{C}$*, and morphisms being defined as* $hom_{\mathtt{C}_{\mathbf{\Phi}}}(X, Y) = hom_{\mathtt{C}}(X, \varphi Y)$*. Morphisms* $f \colon X \rightharpoonup Y$ *in* $\mathtt{C}_{\mathbf{\Phi}}$ *are thus morphisms* $f \colon X \to \varphi Y$ *in* $\mathtt{C}$*, with* $\eta^\varphi_X \colon X \to \varphi X$ *being the identity morphism. Composition of morphisms in* $\mathtt{C}_{\mathbf{\Phi}}$ *is defined as*

$$(X \overset{f}{\rightharpoonup} Y) \diamond (Y \overset{g}{\rightharpoonup} Z) = X \overset{\mu^\varphi_Z \circ \varphi g \circ f}{\longrightarrow} \varphi Z. \tag{3}$$

Composition in the case of the term monad comes down to substitution, and this brings us immediately to substitution theories in general for monads. Monads can be composed and especially the composition of the powerset monad with the term monad provides groundwork for a substitution theory as a basis for many-valued logic [21]. In the following we will elaborate on powerset monads. The concept of *subfunctors* and *submonads* can be used to provide a technique for constructing new monads from given ones.

Definition 4. *Let* φ *be a set functor. A set functor* φ' *is a subfunctor of* φ*, written* $\varphi' \leq \varphi$*, if there exists a natural transformation* $e \colon \varphi' \longrightarrow \varphi$*, called the inclusion transformation, such that* $e_X \colon \varphi' X \longrightarrow \varphi X$ *are inclusion mappings, i.e.,* $\varphi' X \subseteq \varphi X$*. The conditions on the subfunctor imply that* $\varphi f \mid_{\varphi' X} = \varphi' f$ *for all mappings* $f : X \longrightarrow Y$*. Further,* $\leq$ *is a partial ordering.*

Proposition 1 ([13]). *Let* $\mathbf{\Phi} = (\varphi, \eta, \mu)$ *be a monad over* $\mathtt{Set}$*, and consider a subfunctor* φ' *of* φ*, with the corresponding inclusion transformation* $e : \varphi'$

$\longrightarrow \varphi$, *together with natural transformations* $\eta' : id \longrightarrow \varphi'$ *and* $\mu' : \varphi'\varphi' \longrightarrow \varphi'$ *satisfying the conditions*

$$e \circ \eta' = \eta, \tag{4}$$

$$e \circ \mu' = \mu \circ \varphi e \circ e\varphi'. \tag{5}$$

Then $\mathbf{\Phi}' = (\varphi', \eta', \mu')$ *is a monad, called the submonad of* $\mathbf{\Phi}$*, written* $\mathbf{\Phi}' \preceq \mathbf{\Phi}$*.*

4 Examples of Monads

Monads have been used in many different areas such as topology or functional programming. The applications and use of monads in computer science is well-known and provides an abstract tool to handle properties of structures. Examples developed in this section have an important role in many applications. Powerset monads and their many-valued extensions are in close connection to fuzzification and are good candidates to represent situations with incomplete or imprecise information. With respect to topological application, the fuzzy filter monad is a key construction when studying convergence structures from a more general point of view. Unless otherwise stated, we assume L to be a completely distributive lattice. For $L = \{0, 1\}$ we write $L = 2$.

Remark 1. Extending functors to monads is not trivial, and unexpected situations may arise. Let the id^2 functor be extended to a monad with

$$\eta_X(x) = (x, x) \text{ and } \mu_X((x_1, x_2), (x_3, x_4)) = (x_1, x_4).$$

Further, the proper powerset functor P_0, where $P_0 X = PX \setminus \{\emptyset\}$, as well as $id^2 \circ P_0$ can, respectively, be extended to monads, even uniquely. However, as shown in [15], $P_0 \circ id^2$ cannot be extended to a monad.

4.1 The Term Monad

Notations in this part follow [17], which were adopted also in [15, 11]. Let $\Omega = \bigcup_{n=0}^{\infty} \Omega_n$ be an operator domain, where Ω_n contains n-ary operators. The term functor $T_\Omega \colon \mathtt{SET} \to \mathtt{SET}$ is given as $T_\Omega(X) = \bigcup_{k=0}^{\infty} T_\Omega^k(X)$, where

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

In our context, due to constructions related to generalised terms [14, 13, 11], it is more convenient to write terms as $(n, \omega, (x_i)_{i \leq n})$ instead of the more common $\omega(x_1, \ldots, x_n)$. It is clear that $(T_\Omega X, (\sigma_\omega)_{\omega \in \Omega})$ is an Ω-algebra, if $\sigma_\omega((m_i)_{i \leq n}) = (n, \omega, (m_i)_{i \leq n})$ for $\omega \in \Omega_n$ and $m_i \in T_\Omega X$. Morphisms $X \xrightarrow{f} Y$ in $\mathtt{Set}$ are extended in the usual way to the corresponding Ω-homomorphisms $(T_\Omega X, (\sigma_\omega)_{\omega \in \Omega}) \xrightarrow{T_\Omega f} (T_\Omega Y, (\tau_\omega)_{\omega \in \Omega})$, where $T_\Omega f$ is given as the Ω-extension of $X \xrightarrow{f} Y \hookrightarrow T_\Omega Y$ associated to $(T_\Omega Y, (\tau_{n\omega})_{(n,\omega) \in \Omega})$. To obtain the term monad, define $\eta_X^{T_\Omega}(x) = x$, and let $\mu_X^{T_\Omega} = id_{T_\Omega X}^\star$ be the Ω-extension of $id_{T_\Omega X}$ with respect to $(T_\Omega X, (\sigma_{n\omega})_{(n,\omega) \in \Omega})$.

Proposition 2. *[32]* $\mathbf{T}_\Omega = (T_\Omega, \eta^{T_\Omega}, \mu^{T_\Omega})$ *is a monad.*

4.2 The Powerset Monad

The covariant powerset functor L_{id} is obtained by $L_{id}X = L^X$, i.e. the set of mappings (or L-fuzzy sets) $A : X \to L$, and following [22], for a morphism $f\colon X \to Y$ in `Set`, the category of sets and functions, by defining

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

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

$$\eta_X(x)(x') = \begin{cases} 1 & if \text{x=x'} \\ 0 & otherwise \end{cases} \tag{6}$$

and $\mu : L_{id} \circ L_{id} \to L_{id}$ by

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

Proposition 3. *[32]* $\mathbf{L}_{id} = (L_{id}, \eta, \mu)$ *is a monad.*

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 $\mu_X(\mathcal{B}) = \bigcup \mathcal{B}$.

4.3 Powerset Monads with Fuzzy Level Sets

In [12], a number of set functors extending the powerset functor together with their *extension principles* are introduced. By *extension principles* we mean the two possible generalizations of a mapping $f\colon X \longrightarrow Y$ where X, Y are sets, when working in the fuzzy case according to an optimistic or pessimistic interpretation of the fuzziness degree.

1. Maximal extension principle: $Ff_M\colon FX \longrightarrow FY$,

$$Ff_M(A)(y) = \begin{cases} \sup\{A(x) \mid f(x) = y \text{ and } A(x) > 0\} & \text{if the set is nonempty} \\ 0 & \text{otherwise} \end{cases}$$

2. Minimal extension principle: $Ff_m\colon FX \longrightarrow FY$,

$$Ff_m(A)(y) = \begin{cases} \inf\{A(x) \mid f(x) = y \text{ and } A(x) > 0\} & \text{if the set is nonempty} \\ 0 & \text{otherwise} \end{cases}$$

Both extensions Ff_M and Ff_m coincide with the direct image extension in the case of crisp subsets, that is, given $A \in PX$, then $Pf_M(A) = Pf_m(A) = f(A) \in PY$. These maximal and minimal extension principles can be further generalized to the L-fuzzy powersets, just changing the calculations of suprema and infima by the lattice join and meet operators. We will use the set $I = \{x \in X \mid f(x) = y \text{ and } A(x) > 0\}$:

1. Maximal L-fuzzy extension principle: $Lf_M \colon LX \longrightarrow LY$ is

$$Lf_M(A)(y) = \begin{cases} \bigvee_I A(x) & \text{if } I \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

2. Minimal L-fuzzy extension principle: $Lf_m \colon LX \longrightarrow LY$,

$$Lf_m(A)(y) = \begin{cases} \bigwedge_I A(x) & \text{if } I \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

We can now extend the definition of powersets to powersets with fuzzy level sets. Functors for α-upper L-fuzzy sets and α-lower L-fuzzy sets, denoted L_α and L^α, respectively, are given as follows:

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

For mappings $f \colon X \longrightarrow Y$, we define $L_\alpha f : L_\alpha X \longrightarrow L_\alpha Y$ as the restriction of the mapping given by the minimal L-fuzzy extension principle to the L-fuzzy set $L_\alpha X$. Similarly, $L^\alpha f : L^\alpha X \longrightarrow L^\alpha Y$ is given as the restriction of the mapping given by the maximal L-fuzzy extension principle. L-fuzzy set categories are defined for each of these extended power set functors and the rationality of the extension principle is proved in the categorical sense, i.e. the associated L-fuzzy set categories are shown to be equivalent to the category of sets and mappings. We can easily generalize the fact that (L_{id}, η, μ) is a monad and obtain:

Proposition 4. *[12] $(L^\alpha, \eta^\alpha, \mu^\alpha)$ is a monad.*

For L_α we define:

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

$$\mu_{\alpha X}(\mathcal{A})(x) = \begin{cases} \bigwedge\limits_{A \in I} A(x) \wedge \mathcal{A}(A) & \text{if } I = \{A \in L_\alpha X \mid A(x) \wedge \mathcal{A}(A) > 0\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Proposition 5. *[12] $(L_\alpha, \eta_\alpha, \mu_\alpha)$ is a monad.*

Remark 2. For mappings $f \colon X \longrightarrow Y$, we could obtain $L^\alpha f$ as $L_{id}f_{|L^\alpha X}$. Thus, L^α become subfunctor of L_{id} and $\mathbf{L}^\alpha = (L^\alpha, \eta^{L^\alpha}, \mu^{L^\alpha})$ is a submonads of $\mathbf{L_{id}}$.

Remark 3. For $L = 2$, $\mathbf{L}_\alpha = \mathbf{L}^\alpha = \mathbf{2_{id}}$.

4.4 The Covariant Double Contravariant Powerset Monad

The contravariant powerset functor L^{id} is the contravariant hom-functor related to L, i.e. $L^{id} = hom(-, L) : \mathtt{Set} \longrightarrow \mathtt{Set}$, which to each set X and mapping $f : X \longrightarrow Y$ assigns the set L^X of all mappings of X into L, and the mappings $hom(f, L)(g) = g \circ f \ \ (g \in L^Y)$, respectively. Note that 2^{id} is the usual contravariant powerset functor, where $2^{id}X = PX$, and morphisms $X \xrightarrow{f} Y$ in $\mathtt{Set}$ are mapped to $2^{id}f$ representing the mapping $M \mapsto f^{-1}[M]$ $(M \in PY)$ from PY to PX. For double powerset functors it is convenient to write $L_{L_{id}} = L_{id} \circ L_{id}$ and $L^{L^{id}} = L^{id} \circ L^{id}$. Note that $L^{L^{id}}$ is a covariant functor. It may be interesting also to note that the *filter*[1] functor is a subfunctor of $2^{2^{id}}$, but not a subfunctor of $2_{2_{id}}$. In the case of $L^{L^{id}}$, for $X \xrightarrow{f} Y$ in $\mathtt{Set}$ and $\mathcal{M} \in L^{L^X}$, we have $L^{L^{id}}f(\mathcal{M}) = \mathcal{M} \circ L^{id}f$, and hence, $L^{L^{id}}f(\mathcal{M})(g) = \mathcal{M}(g \circ f)$.

Proposition 6. *[15] The covariant set functor $LL = L^{id} \circ L^{id}$ can be extended to a monad, considering the following definitions of the natural transformations η^{LL} and μ^{LL}:*

$$\eta^{LL}_X(x)(A) = A(x), \qquad \mu^{LL}_X(\mathcal{U}) = \mathcal{U} \circ \eta^{LL}_{LX}.$$

It is well-known that the proper[2] filter functor F_0 becomes a monad where $\eta^{F_0} : id \longrightarrow F_0$ is the unique natural transformation and $\mu^{F_0} : F_0 \circ F_0 \longrightarrow F_0$ is given by

$$\mu^{F_0}_X(\mathcal{U}) = \bigcup_{R \in \mathcal{U}} \bigcap_{\mathcal{M} \in R} \mathcal{M}$$

i.e. the contraction mapping suggested in [30].

Remark 4. In relation with the functor $2^{2^{id}}$, it can easily be seen that $\mu^{2^{2^{id}}}_X(\mathcal{U}) = \mu^{F_0}_X(\mathcal{U})$.

5 Partially Ordered Monads

Godement in 1958 used monads named as *standard constructions* and Huber in 1961 showed that adjoint pairs give rise to monads. In 1965, Kleisli [28], Eilenberg and Moore [4] proved the converse. Lawvere [31] introduced universal algebra and thereby the term monad. These developments provide all categorical tools for generalized substitutions. In 2000, Gähler develops partially ordered monads [18], where topology and convergence provided underlying theories. Partially ordered monads contain sufficient structure also for modelling rough sets [33] in a generalized setting with set functors. This generalization builds upon a more general powerset functor setting far beyond just strings [27] and relational algebra [37]. Let $\mathtt{acSLAT}$ be the category of almost complete

[1] A *filter* on a set X is a nonempty set $\mathcal{F}$ of subsets of X such that: (i) $\emptyset \notin \mathcal{F}$, (ii) $A, B \in \mathcal{F} \Rightarrow A \cap B \in \mathcal{F}$, (iii) $A \in \mathcal{F} \quad A \subseteq B \Rightarrow B \in \mathcal{F}$.

[2] $F_0 X = FX \setminus \{\emptyset\}$

semilattices, i.e. partially ordered sets $(X, \leq)$ such that the suprema $\sup \mathcal{M}$ of all non-empty subsets $\mathcal{M}$ of X exists. Morphisms $f : (X, \leq) \to (Y, \leq)$ satisfy $f(\sup \mathcal{M}) = \sup f[\mathcal{M}]$ for non-empty subsets $\mathcal{M}$ of X. A *basic triple* ([18]) is a triple $\Phi = (\varphi, \leq, \eta)$, where $(\varphi, \leq) : \mathtt{SET} \to \mathtt{acSLAT}$, $X \mapsto (\varphi X, \leq)$ is a covariant functor, with $\varphi : \mathtt{SET} \to \mathtt{SET}$ as the underlying set functor, and $\eta : \mathrm{id} \to \varphi$ is a natural transformation. If $(\varphi, \leq, \eta^{\varphi})$ and $(\psi, \leq, \eta^{\psi})$ are basic triples, then also $(\varphi \circ \psi, \leq, \eta^{\varphi}\psi \circ \eta^{\psi})$ is a basic triple.

Definition 5. *A partially ordered monad is a quadruple $\Phi = (\varphi, \leq, \eta, \mu)$, such that*

(i) $(\varphi, \leq, \eta)$ is a basic triple.
(ii) $\mu : \varphi\varphi \to \varphi$ is a natural transformation such that (φ, η, μ) is a monad.
(iii) For all mappings $f, g : Y \to \varphi X$, $f \leq g$ implies $\mu_X \circ \varphi f \leq \mu_X \circ \varphi g$, where $\leq$ is defined argumentwise with respect to the partial ordering of φX.
(iv) For each set X, $\mu_X : (\varphi\varphi X), \leq) \to (\varphi X, \leq)$ preserves non-empty suprema.

The usual covariant powerset monad $\mathbf{P} = (P, \eta, \mu)$, can be extended to a partially ordered monad, $(P, \subseteq, \eta, \mu)$, considering as the partial ordering the inclusion, $\subseteq$. Clearly by the properties of the monad, $(P, \subseteq, \eta)$ is a basic triple, μ is a natural transformation and $\mu_X : (PPX), \subseteq) \to (PX, \subseteq)$ preserves non-empty suprema. Given $f, g : Y \longrightarrow PX$ with $f \subseteq g$ e.g. $f(y) \subseteq g(y)$ for all $y \in Y$ implies $\mu_X \circ Pf \subseteq \mu_X \circ Pg$:

$$(\mu_X \circ Pf)(B) = \bigcup_{y \in B \subseteq Y} f(y) \subseteq \bigcup_{y \in B \subseteq Y} g(y) = (\mu_X \circ Pg)(B)$$

The powerset monad, (L_{id}, η, μ) can also be extended to a partially ordered monad, considering the partial order defined as $A \leq A'$, with $A, A' \in L_{id}X$ if $A(x) \leq A'(x)$ for all $x \in X$. Let us see that $\mu_X \circ L_{id}f \leq \mu_X \circ L_{id}g$: provided that $f \leq g$ where $f, g : Y \longrightarrow L_{id}X$.

$$\begin{aligned}
\mu_X^{L_{id}}(L_{id}f(B))(x) &= \bigvee_{A \in L_{id}X} A(x) \wedge L_{id}f(B)(A) \\
&= \bigvee_{A \in L_{id}X} A(x) \wedge \bigvee_{f(y)=A} B(y) \\
&= \bigvee_{A \in L_{id}X} \bigvee_{f(y)=A} A(x) \wedge B(y) \\
&= \bigvee_{y \in Y} f(y)(x) \wedge B(y) \\
&\leq \bigvee_{y \in Y} g(y)(x) \wedge B(y) \\
&= \mu_X^{L_{id}}(L_{id}g(B))(x).
\end{aligned}$$

Finally, also the monad $(L^\alpha, \eta^\alpha, \mu^\alpha)$ can be extended to a partially ordered monad. This result is a generalization of L_{id} being extendable to a partially ordered monad. To provide L_α with the partially ordered monad structure we need to check that if $f, g : Y \longrightarrow L_\alpha X$ are such that $f \leq g$ then $\mu_X \circ L_\alpha f \leq \mu_X \circ L_\alpha g$. In the same way as the case of L_{id}, the partial order is defined as $A \leq A'$, with $A, A' \in L_\alpha X$ meaning $A(x) \leq A'(x)$ for all $x \in X$.

$$
\begin{aligned}
\mu_X^{L_\alpha}(L_\alpha f(B))(x) &= \bigwedge_{A \in L_\alpha X, A(x)>0, L_\alpha f(B)(A)>0} A(x) \wedge L_\alpha f(B)(A) \\
&= \bigwedge_{A \in L_\alpha X, A(x)>0, L_\alpha f(B)(A)>0} A(x) \wedge \bigwedge_{y \in Y, f(y)=A, B(y)>0} B(y) \\
&= \bigwedge_{A \in L_\alpha X, A(x)>0, f(y)=A, B(y)>0} A(x) \wedge B(y) \\
&= \bigwedge_{B(y)>0} f(y)(x) \wedge B(y) \\
&\leq \bigwedge_{B(y)>0} g(y)(x) \wedge B(y) \\
&= \mu_X^{L_\alpha}(L_\alpha g(B))(x).
\end{aligned}
$$

Note that $f \leq g$ implies $f(y)(x) \wedge B(y) \leq g(y)(x) \wedge B(y)$ for all $x \in X$ and therefore $\mu_X^{L_\alpha}(L_\alpha f(B))(x) \leq \mu_X^{L_\alpha}(L_\alpha g(B))(x)$.

6 Relations, Kleisli Categories and Rough Sets

Rough sets and fuzzy sets are both methods to represent uncertainty. By using partially ordered monads we can find connections between these two concepts. Partially ordered monads are appropriate categorical formalizations and generalizations of rough sets. In this section we introduce relations from a categorical point of view and justify how its composition can be seen within Kleisli categories. Partially ordered monadic reformulation of rough sets based on the powerset partially ordered monad and the fuzzy powerset monad are presented and some properties are studied.

6.1 Crisp Situation

Let us consider a binary relation $R \subseteq X \times Y$. We will use the notation xRy to represent that the element $(x, y) \in R$. Considering P, the crisp powerset functor, we can represent the relation as a mapping $\rho : X \longrightarrow PY$, where

$$\rho(x) = \{y \in Y \text{ such that } xRy\}$$

As regarded as mappings, considering the composition of two relations, $\rho : X \longrightarrow PY$ and $\rho' : Y \longrightarrow PZ$ we clearly see that the conventional composition of mappings can not be done since the domain of ρ' and codomain of ρ are different.

To find the appropriate definition of this composition we have to consider the Kleisli composition as defined previously by (3), i.e. we need to use that P is a monad and has a "flattering" operator, μ:

$$(X \xrightarrow{\rho} Y) \diamond (Y \xrightarrow{\rho'} Z) = X \xrightarrow{\mu_Z^P \circ P\rho' \circ \rho} PZ.$$

The reason for this to work is the following proposition:

Proposition 7. *The Kleisli category associated to the crisp powerset monad is equivalent to the category of sets and relations, SetRel.*

Indeed, $\rho : X \longrightarrow PY$ corresponds to a relation $R \subseteq X \times Y$ by the observation $(x, y) \in R$ if and only if $y \in \rho(x)$.

Proposition 8. *Kleisli composition associated to P is given by:*

$$\mu_Z^P \circ P\rho'(\rho(x)) = \bigcup_{y \in \rho(x)} \rho'(y)$$

Clearly Kleisli composition, in this case, corresponds to the usual composition of relations $R \subseteq X \times Y$, $R' \subseteq Y \times Z$, $(x, z) \in R' \circ R$ if and only if $\exists y, y \in \rho(x), z \in \rho'(y)$. Based on indistinguishable relations, *rough sets* are introduced by defining the upper and lower approximation of sets. These approximations represent uncertain or imprecise knowledge. Let us consider a relation R on X, i.e. $R \subseteq X \times X$. We represent the relation as a mapping $\rho_X : X \longrightarrow PX$, where $\rho_X(x) = \{y \in X | xRy\}$. The corresponding inverse relation R^{-1} is represented as $\rho_X^{-1}(x) = \{y \in X | xR^{-1}y\}$. To be more formal, given a subset A of X, the lower approximation of A correspond to the objects that surely (with respect to an indistinguishable relation) are in A. The lower approximation of A is obtained by

$$A^{\downarrow} = \{x \in X | \rho_X(x) \subseteq A\}$$

and the upper approximation by

$$A^{\uparrow} = \{x \in X | \rho_X(x) \cap A \neq \emptyset\}.$$

Let us see now the partially ordered monadic reformulation of rough sets based on the powerset partially ordered monad. In what follows we will assume that the underlying almost complete semilattice has finite infima, i.e. is a join complete lattice. Considering P as the functor in its corresponding partially ordered monad we then immediately have

Proposition 9. *[6] The upper and lower approximations of a subset A of X are given by*

$$A^{\uparrow} = \bigvee_{\rho_X(x) \wedge A > 0} \eta_X(x) = \mu_X \circ P\rho_X^{-1}(A)$$

and

$$A^{\downarrow} = \bigvee_{\rho_X(x) \leq A} \eta_X(x),$$

respectively.

The corresponding R-weakened and R-substantiated sets of a subset A of X are given by

$$A^{\Downarrow} = \{x \in X | \rho_X^{-1}(x) \subseteq A\}$$

and

$$A^{\Uparrow} = \{x \in X | \rho_X^{-1}(x) \cap A \neq \emptyset\}.$$

Proposition 10. *[6] The R-weakened and R-substantiated sets of a subset A of X are given by*

$$A^{\Uparrow} = \mu_X \circ P\rho_X(A)$$

and

$$A^{\Downarrow} = \bigvee_{\rho_X^{-1}(x) \leq A} \eta_X(x),$$

respectively.

Proposition 11. *If $A \subseteq B$ then $A^{\uparrow} \subseteq B^{\uparrow}$, $A^{\downarrow} \subseteq B^{\downarrow}$, $A^{\Uparrow} \subseteq B^{\Uparrow}$, $A^{\Downarrow} \subseteq B^{\Downarrow}$.*

The upper and lower approximations, as well as the R-weakened and R-substantiated sets, can be viewed as $\uparrow_X, \downarrow_X, \Uparrow_X, \Downarrow_X \colon PX \longrightarrow PX$ with $\uparrow_X (A) = A^{\uparrow}$, $\downarrow_X (A) = A^{\downarrow}$, $\Uparrow_X (A) = A^{\Uparrow}$ and $\Downarrow_X (A) = A^{\Downarrow}$. Considering the crisp powerset monad we define equivalence relations (reflexive, symmetric and transitive) by

Definition 6. $\rho_X : X \longrightarrow PX$ *is* reflexive *if* $\eta_X \subseteq \rho_X$, symmetric *if* $\rho_X = \rho_X^{-1}$ *and* transitive *if $y \in \rho(x)$ implies $\rho(y) \subseteq \rho(x)$.*

In what follows, equivalence relations are now connected to upper and lower approximations.

Proposition 12. *The following properties hold:*

(i) If ρ_X is reflexive $A^{\downarrow} \subseteq A$ and $A \subseteq A^{\uparrow}$.
(ii) If ρ_X is symmetric $A^{\downarrow\uparrow} \subseteq A$ and $A \subseteq A^{\uparrow\downarrow}$.
(iii) If ρ_X is transitive $A^{\uparrow\uparrow} \subseteq A^{\uparrow}$ and $A^{\downarrow} \subseteq A^{\downarrow\downarrow}$.

Corollary 1. *If ρ_X is an equivalence relation, $A^{\downarrow\uparrow} = A^{\downarrow}$ and $A^{\uparrow\downarrow} = A^{\uparrow}$.*

Inverse relations in the ordinary case means to mirror pairs around the diagonal. The following propositions relate inverses to the multiplication of the corresponding monads.

Proposition 13. *[6] In the case of P,*

$$\bigvee_{\rho_X(x) \wedge A > 0} \eta_X(x) = \mu_X \circ P\rho_X^{-1}(A)$$

if and only if

$$\rho_X^{-1}(x) = \bigcup_{\eta_X(x) \leq \rho_X(y)} \eta_X(y).$$

6.2 Many-Valued Situation

We will show now how to extend this view of relations to fuzzy relations. In particular it will be interesting the situation where Kleisli composition is defined for composing fuzzy relations. This can be connected to situations where we want to combine different information systems and study rough approximations. Relations can now be extended to fuzzy relations. Let X and Y be nonempty sets. A fuzzy relation R is a fuzzy subset of the cartesian product $X \times Y$. If $X = Y$ we say that R is a binary fuzzy relation on X. $R(x, y)$ is interpreted as the degree of membership of the pair (x, y) in R. If we consider now the generalized powerset monad, $L_{id}X$ is the set of all L-fuzzy sets. An L-fuzzy set A is nothing but a mapping $A : X \longrightarrow L$. As a first step, and in the same way as before we can extend the concept of relation to a fuzzy relation, i.e. a mapping $\rho : X \longrightarrow L_{id}Y$, $\rho(x)$ is nothing but an element in $L_{id}Y$, a mapping $\rho(x) : Y \longrightarrow L$. An element $y \in Y$ will be assigned a membership degree, $\rho(x)(y)$ representing, as a value in L, the degree on which the elements x and y are fuzzy related. Note that this situation extend the classical relations (crisp powerset situation) in the sense that membership values are 1 if the elements are related and 0 otherwise. With respect to the Kleisli category associated to the powerset monad $\mathbf{L_{id}}$, the objects are sets and homomorphisms are given as mappings $X \longrightarrow L_{id}Y$ in `Set`.

Proposition 14. *[8] The Kleisli category associated to* $\mathbf{L_{id}}$ *is equivalent to the category of set and fuzzy relations,* `SetFuzzRel`.

Proposition 15. *[8] Kleisli composition associated to* $\mathbf{L_{id}}$ *is given by:*

$$\mu_Z^{L_{id}}(L_{id}\rho'(\rho(x)))(z) = \bigvee_{y \in Y} \rho'(y)(z) \wedge \rho(x)(y)$$

The previous proposition tells which membership grade we should assign to the composition of two fuzzy relations, i.e. the suprema of the membership grades on the fuzzy relations. This Kleisli composition of fuzzy relations can be connected to situations where we want to combine different information systems and study rough approximations. Similarly to the crisp situation we can now introduce rough set operators for the fuzzy powerset monad. Let $\rho_X : X \longrightarrow L_{id}X$ be a fuzzy relation on X and let $a \in L_{id}X$. The upper and lower approximations are then

$$\uparrow_X (a) = \mu_X \circ L_{id}\rho_X^{-1}(a) \qquad \downarrow_X (a) = \bigvee_{\rho_X(x) \leq a} \eta_X(x)$$

Corresponding generalizations of ρ-weakenedness and ρ-substantiatedness, are given by

$$\Uparrow_X (a) = \mu_X \circ L_{id}\rho_X(a) \qquad \Downarrow_X (a) = \bigvee_{\rho_X^{-1}(x) \leq a} \eta_X(x)$$

Concerning inverse relations, in the case of L_{id} we would accordingly define $\rho_X^{-1}(x)(x') = \rho_X(x')(x)$.

Proposition 16. *[6] In the case of L_{id},*

$$\mu_X \circ L_{id}\rho_X^{-1}(A)(x) = \bigvee_{x' \in X} (\rho_X(x) \wedge A)(x').$$

Consider now the powerset monads with fuzzy level sets, $\mathbf{L}^\alpha$ and $\mathbf{L}_\alpha$. For L^α is similar to L_{id} situation. Let us see how is the situation for L_α

Proposition 17. *[8] In the case of L_α,*

$$\mu_X \circ L_\alpha\rho_X^{-1}(A)(x) = \bigwedge_{x' \in X} (\rho_X(x) \wedge A)(x').$$

Note that in the case of $L = 2$, for the functor 2_α we obtain the classical definition of the upper approximation of a set A. Generalizing from the ordinary power set monad to a wide range of partially ordered monads requires attention to relational inverses and complement. The role of the diagonal clearly changes, and the representation of inverses is an open question. Inverses and complements must be based on negation operators as given by implication operators within basic many-valued logic [23].

7 Rough Monads

In the previous section we have shown how rough sets can be given using partially ordered monads. From a more abstract point of view, we present in this section a generalized view of rough set constructions based on general partially ordered monads. We name these generalizations *rough monads*. Considering the partially ordered powerset monad, we showed in [6] how rough sets operations can be provided in order to complement the many-valued situation. This is accomplished by defining rough monads. Let $\Phi = (\varphi, \leq, \eta, \mu)$ be a partially ordered monad. We say that $\rho_X : X \longrightarrow \varphi X$ is a *Φ-relation* on X, and by $\rho_X^{-1} : X \longrightarrow \varphi X$ we denote its *inverse*. The inverse must be specified for the given set functor φ. For any $f \colon X \longrightarrow \varphi X$, the following condition is required:

$$\varphi f(\bigvee_i a_i) = \bigvee_i \varphi f(a_i)$$

This condition is valid both for P as well as for L_{id}.

Remark 5. Let ρ_X and ρ_Y be relations on X and Y, respectively. Then the mapping $f : X \longrightarrow Y$ is a congruence, i.e. $x' \in \rho_X(x)$ implies $f(x') \in \rho_Y(f(x))$, if and only if $Pf \circ \rho_X \leq \rho_Y \circ f$. Thus, congruence is related to kind of weak naturality.

Let $\rho_X : X \longrightarrow \varphi X$ be a Φ-relation and let $a \in \varphi X$. The upper and lower approximations are then

$$\uparrow_X (a) = \mu_X \circ \varphi\rho_X^{-1}(a) \qquad \downarrow_X (a) = \bigvee_{\rho_X(x) \leq a} \eta_X(x)$$

with the monadic generalizations of ρ-weakenedness and ρ-substantiatedness, for $a \in \varphi X$, being

$$\Uparrow_X (a) = \mu_X \circ \varphi\rho_X(a) \qquad\qquad \Downarrow_X (a) = \bigvee_{\rho_X^{-1}(x) \leq a} \eta_X(x)$$

Proposition 18. *[6] If* $a \leq b$*, then* $\Uparrow_X a \leq \Uparrow_X b$*,* $\downarrow_X a \leq \downarrow_X b$*,* $\uparrow_X a \leq \uparrow_X b$*,* $\Downarrow_X a \leq \Downarrow_X b$*.*

In the case of $\varphi = P$, i.e. the conventional powerset partially ordered monad, these operators coincide with those for classical rough sets. In this case inverse relations exist accordingly. In the case of fuzzy sets we use the many-valued powerset partially ordered monad based on the many-valued extension of P to L_{id}. Basic properties of relations can now be represented with 'rough monads terminology:

Definition 7. $\rho_X : X \longrightarrow \varphi X$ *is* reflexive *if* $\eta_X \leq \rho_X$*, and* symmetric *if* $\rho = \rho^{-1}$*.*

Note that in the case of relations for P and L_{id}, if the relations are reflexive, so are their inverses.

Proposition 19. *[6]*

(i) If ρ *is reflexive,* $a \leq \Uparrow_X (a)$*.*
(ii) ρ *is reflexive iff* $\downarrow_X (a) \leq a$*.*
(iii) ρ_X^{-1} *is reflexive iff* $a \leq \uparrow_X (a)$*.*
(iv) If ρ *is symmetric, then* $\uparrow_X (\downarrow_X (a)) \leq a$*.*

In the particular case $a = \eta_X(x)$ we have $a \leq \downarrow_X \circ \uparrow_X (a)$. The idea of submonad is similar to the idea of subsets.In this sense, the calculations related to submonads are a way to reduce data in a given information system. Let $\Phi' = (\varphi', \leq, \eta', \mu')$ be a partially ordered submonad of $\Phi = (\varphi, \leq, \eta, \mu)$. Given $a' \in \varphi' X$ we have the following proposition:

Proposition 20. *[8] For* $a' \in \varphi' X$*,*

$$\uparrow_X (a') = \mu_X \circ \varphi\rho_X^{-1}(a') \qquad\qquad \downarrow_X (a') = \bigvee_{\rho_X(x) \leq a'} \eta_X(x)$$

This proposition shows us that rough approximations are well defined wrt submonads, i.e. their definition in the submonad correspond to the one for the monad.

8 Applications

Our theoretical developments are an inspiration for application development. As a first step we have focused on ICT solutions within health care. Information representation based on medical ontologies are usually rather narrow and oriented towards crisp specifications of data information. At the same time, health care ICT solutions call for representation of vagueness and uncertainties both for use within medical records and information databases, as well as for decision support and guideline implementations. We will discuss various fields of health care and possible use of generalized rough sets, and we will in particular develop concrete examples in the area of decision support and, more specifically, decisions related to diagnosis and treatment.

8.1 Drug Interactions

Pharmacological treatment is an excellent area for our experimental purposes where e.g. drug interactions [10] can be favourably described using generalized rough sets. Pharmacological databases provide rich and complete information for therapeutic requirements. In particular, the ATC code with its unique identification of drug compound is the basis e.g. of modelling of generic substitutes and drug interactions. Two drugs are generic substitutes if they have the same ATC code, the same dosages and the same administration route. This is straightforward and precise but the notion of drug-drug interaction is more complicated. In addition, drug-condition interaction adds further complexity as medical conditions themselves are not easy to formalize. Rough sets described by partially ordered monads are able to capture interactions with respect to different granularities in the information hierarchy. The data structure for pharmacologic information is hierarchical in its subdivision according to anatomic, therapeutic and chemical information of the drug compound. National catalogues of drugs aim at being complete with respect to chemical declarations, indications/contraindications, warnings, interactions, side-effects, pharmacodynamics/pharmacokinetics, and pure pharmaceutical information. The Anatomic Therapeutic Chemical (ATC) classification system is a WHO (World Health Organization) standard. The ATC structure can be understood from Table 1 on the classification of verapamil (code C08DA01) for hypertension with stable angina pectoris. Drugs in ATC are, with a very few exceptions, classified according to their main indication of use. The ATC coded is for therapeutic use, while the article code is a unique identifier which is used in the patient's record. For drugs showing therapeutically significant interactions we need to distinguish between types of interactions and to what extent we have evidence for that particular type of interaction. The types of interaction are *recommended combination*, *neutral combination* (no harmful interactions), *risky combination* (should be monitored) and *dangerous combination* (should be avoided). The degrees of evidence are *strong evidence* (internationally), *reasonable belief* (several studies exist), *some indications* (only some studies exist, and results are not conclusive) and *no evidence*. With these qualifications it is clear that a linear quantification cannot be given. Further, the drugs are affected in different ways, according to *no change in effect*, *increases effect*, *reduces effect* and *other* (e.g. a new type of side effect). Interaction type, evidence level, and effect need to be considered in the guideline for respective treatments. In our subsequent discussion we focus on guideline

Table 1. Classification of *verapamil*

C	cardiac and vessel disease medication	1st level main anatomical group
C08	calcium channel blockers	2nd level, therapeutic subgroup
C08D	selective cardiac calcium channel blockers	3rd level, pharmacological subgroup
C08DA	phenylalcylamins	4th level, chemical subgroup
C08DA01	verapamil	5th level

based pharmacologic treatment of hypertension [38]. See also [34] for an implementation of these guidelines for primary care. Typical drugs for hypertension treatment are beta-blockers (C07, C07A) like an atenolol (C07AB03) and diuretics (C03) like thiazides (C03A, C03AA). Atenolol is a selective beta-1-blocker (C07AB). A frequently used thiazide is hydrochlorothiazide (C03AA03). Note that beta-blockers are both therapeutic as well as pharmacological subgroups. Similarly, thiazides are both pharmacological as chemical subgroups. As a basic example concerning interactions consider treatment of hypertension in presence of diabetes. Beta-blockers may mask and prolong beta-blockers insulin-induced hypoglycemia. If the patient shows the medical condition of diabetes without any other medical condition present, then the ACE inhibitor (C09A, C09AA) enalapril (C09AA02) is the first choice for treatment [38].

Drug interactions as relations can be interpreted as mappings $\rho_X^L : X \longrightarrow LX$, based on the many-valued powerset monad (L, η, μ). Let M be a set of medical conditions and let $\rho^L[M]$ be the subrelation of ρ which considers interactions with pharmacological treatments based on these medical conditions in M. We then observe that the clinical usefulness of these interpretations comes down to defining $\rho^L[M]$ so as to correspond to real clinical situations. Operating with these sets then becomes the first step to identify connections to guidelines for pharmacological treatment.

In [26], a software framework for pharmacological information representation is suggested. This framework enables clients to recover information from databases with pharmacological information. In the current implementation, the framework uses ATC codes in the drug metadata. Specifically, the framework provides information about interactions as a set of ATC codes for a particular ATC code (drug). This software framework will be used to recover pharmacological information and related drug interactions, and further using this information in a knowledge-discovery application using the rough set and monad theoretical framework as described in this chapter. The experiment will extract drug information relates also to hypertension treatment [34] from the drug database. Further, to demonstrate that this representation is usable in a realistic situation, the forming of the sets described earlier will indeed take into account a set of medical conditions. These conditions will be described codes from the ICD and corresponding diagnosis encoding system. The hypothesis is that rough monads provide drug interactions with an adequate representation for pharmacological hypertension treatment with respect to an individual and typical patient case.

8.2 Dementia Differential Diagnosis

The differential diagnosis process in the case of dementia involves e.g. to distinguish between dementias of Alzheimer's and vascular type. In the case of Alzheimer's, pharmacological treatment following an early detection can be useful for maintaining acetylcholin in the synapsis between nerve cells. Receptors then remain stimulated thus maintaining activitity and nerve signals. In the scenario of early detection it is important to observe the situations where cognitive problems are encountered and by whom these observations are made. Clearly, the very first observations of cognitive decline are made by relatives (if not self-detected by the patient) or social workers in home care who would forward information about the problems encountered, thus seeking advice firstly from nurses and primary care doctors within their local health care centres. Representatives in social care and nursing will not perform any diagnosis. However, providing some observation and even 'qualified guesses' can speed up the process leading eventually to an accurate diagnosis with possibilities for further pharmacological treatments. It is then important to identify respective information types and rule representations for these professional groups providing everything from 'qualified guesses' to accurate diagnosis. Note that not even autopsy can provide higher diagnostic accuracy than around 80%, so early detection is really hard and challenging. Many-valuedness provides tools for logic transformations between professional groups. Regardless of where decision and/or observations are made, we always need to guarantee consistency when information and knowledge is mapped between ontological domains as understood and used by these professional groups. For further reading on the general logics approach to transformations, see [16]. The intuition of using rough sets and monads is here very natural and even rather obvious. In differential diagnosis we are viewing the set of attributes (symptoms and signs) in a relational setting. Indeed attributes are related not just on powerset level, but also in a 'sets of sets of attributes' fashion. Heteroanamnesis, for instance, is a set of attributes which are grouped according to their interrelations. Thus we are dealing with heteroanamnesis as a set of sets of attributes. Upper and lower approximations are useful as they provide operators transforming a set (or generalized sets), as a relation, to another boundary in some canonic way. Full interpretations are yet to be given, and the pragmatic is still somewhat open, but these developments build upon software developments and real clinical use of these software tools.

9 Conclusion

Rough sets are naturally categorical once we observe how rough set operators can be generalized using other set functors, extendable to partially ordered monads, than just the powerset partially ordered monad representing relations on ordinary sets. The categorical instrumentation reveals many aspects and possibilities for further developments of rough monads, both from theoretical as well as from application points of view. Theoretical developments involve extensions using partially ordered monads and invokes e.g. logical viewpoints that would not appear unless the categorical generalizations are used. Application developments make use of entirely new ways to arrange sets and sets of sets in a wide range of ways. Information pieces and blocks organized as many-valued sets of terms or even filters and ideals opens up an avenue of possibilities for further exploration of intuition combined with formalism.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading (1975)
2. Barr, M., Wells, C.: Toposes, Triples and Theories. Springer, Heidelberg (1985)
3. Cattaneo, G., Ciucci, D.: Shadowed sets and related algebraic structures. Fundamenta Iinformaticae 55, 255–284 (2003)
4. Eilenberg, S., Moore, J.C.: Adjoint functors and triples. Illinois J. Math. 9, 381–398 (1965)
5. Eklund, P., Gähler, W.: Partially ordered monads and powerset Kleene algebras. In: Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems Conference (IPMU 2004) (2004)
6. Eklund, P., Galán, M.A.: Monads can be rough. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS, vol. 4259, pp. 77–84. Springer, Heidelberg (2006)
7. Eklund, P., Galán, M.A.: On logic with fuzzy and rough powerset monads. In: Kryszkiewicz, M., Peters, J.F., Rybinski, H., Skowron, A. (eds.) RSEISP 2007. LNCS, vol. 4585, pp. 391–399. Springer, Heidelberg (2007)
8. Eklund, P., Galán, M.A.: Partially ordered monads and rough sets. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets VIII. LNCS, vol. 5084, pp. 53–74. Springer, Heidelberg (2008)
9. Eklund, P., Galán, M.A., Gähler, W., Medina, J., Ojeda Aciego, M., Valverde, A.: A note on partially ordered generalized terms. In: Proc. of Fourth Conference of the European Society for Fuzzy Logic and Technology and Rencontres Francophones sur la Logique Floue et ses applications (Joint EUSFLAT-LFA 2005), pp. 793–796 (2005)
10. Eklund, P., Galán, M.A., Karlsson, J.: Rough Monadic Interpretations of Pharmacologic Information. In: Proceedings of the 15th International Workshops on Conceptual Structures, ICCS 2007, pp. 108–113. Sheffield, UK (2007)

11. Eklund, P., Galán, M.A., Medina, J., Ojeda Aciego, M., Valverde, A.: A categorical approach to unification of generalised terms. Electronic Notes in Theoretical Computer Science 66(5) (2002), `http://www.elsevier.nl/locate/entcs/volume66.html`
12. Eklund, P., Galán, M.A., Medina, J., Ojeda Aciego, M., Valverde, A.: Set functors, L-fuzzy set categories and generalized terms. Computers and Mathematics with Applications 43, 693–705 (2002)
13. Eklund, P., Galán, M.A., Medina, J., Ojeda-Aciego, M., Valverde, A.: Composing submonads. In: Proc. 31st IEEE Int. Symposium on Multiple-Valued Logic (ISMVL 2001), Warsaw, Poland, May 22-24, pp. 367–372 (2001)
14. Eklund, P., Galán, M.A., Ojeda-Aciego, M., Valverde, A.: Set functors and generalised terms. In: Proc. 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000), pp. 1595–1599 (2000)
15. Eklund, P., Gähler, W.: Fuzzy Filter Functors and Convergence. In: Rodabaugh, S.E., Klement, E.P., Höhle, U. (eds.) Applications of category theory to fuzzy subsets. Theory and Decision Library B, pp. 109–136. Kluwer, Dordrecht (1992)
16. Eklund, P., Helgesson, R., Lindgren, H.: Towards refinement of clinical evidence using general logics. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS, vol. 5097, pp. 1029–1040. Springer, Heidelberg (2008)
17. Gähler, W.: Monads and convergence. In: Proc. Conference Generalized Functions, Convergences Structures, and Their Applications, Dubrovnik (Yugoslavia) 1987, pp. 29–46. Plenum Press (1988)
18. Gähler, W.: General Topology – The monadic case, examples, applications. Acta Math. Hungar. 88, 279–290 (2000)
19. Gähler, W.: Extension structures and completions in topology and algebra, Seminarberichte aus dem Fachbereich Mathematik, Band 70, FernUniversität in Hagen (2001)
20. Gähler, W., Eklund, P.: Extension structures and compactifications, In: Categorical Methods in Algebra and Topology (CatMAT 2000), pp. 181–205 (2000)
21. Galán, M.A.: Categorical Unification, Dissertation, Umeå University (2004)
22. Goguen, J.A.: L-fuzzy sets. J. Math. Anal. Appl. 18, 145–174 (1967)
23. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
24. Järvinen, J.: On the structure of rough approximations. Fundamenta Informaticae 53, 135–153 (2002)
25. Järvinen, J.: Lattice theory for rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 400–498. Springer, Heidelberg (2007)
26. Karlsson, J.: Interface for accessing pharmacological information. In: Proc. 21st IEEE International Symposium on Computer-Based Medical Systems, Jyväskylä (Finland), June 17-19. IEEE CS Press, Los Alamitos (to appear, 2008)
27. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Shannon, C.E., McCarthy, J. (eds.) Automata Studies, pp. 3–41. Princeton University Press, Princeton (1956)
28. Kleisli, H.: Every standard construction is induced by a pair of adjoint functors. Proc. Amer. Math. Soc. 16, 544–546 (1965)
29. Kortelainen, J.: A Topological Approach to Fuzzy Sets, Ph.D. Dissertation, Lappeenranta University of Technology, Acta Universitatis Lappeenrantaensis 90 (1999)

30. Kowalsky, H.-J.: Limesräume und Komplettierung. Math. Nachr. 12, 301–340 (1954)
31. Lawvere, F.W.: Functorial semantics of algebraic theories, Dissertation, Columbia University (1963)
32. Manes, E.G.: Algebraic Theories. Springer, Heidelberg (1976)
33. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sciences 5, 341–356 (1982)
34. Persson, M., Bohlin, J., Eklund, P.: Development and maintenance of guideline-based decision support for pharmacological treatment of hypertension. Comp. Meth. Progr. Biomed. 61, 209–219 (2000)
35. Rydeheard, D., Burstall, R.: A categorical unification algorithm. In: Poigné, A., Pitt, D.H., Rydeheard, D.E., Abramsky, S. (eds.) Category Theory and Computer Programming. LNCS, vol. 240, pp. 493–505. Springer, Heidelberg (1986)
36. Salomaa, A.: Two complete axiom systems for the algebra of regular events. J. ACM 13, 158–169 (1966)
37. Tarski, A.: On the calculus of relations. J. Symbolic Logic 6, 65–106 (1941)
38. The sixth report of the joint national committee on prevention detection, evaluation, and treatment of high blood pressure, Technical Report 98-4080, National Institutes of Health (1997)
39. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)