

# Similarities between powersets of terms<sup>1</sup>

P. Eklund<sup>a</sup>, M.A. Galán<sup>b</sup>, J. Medina<sup>c</sup>, M. Ojeda-Aciego<sup>c</sup> and  
A. Valverde<sup>c</sup>

<sup>a</sup>*Åbo Akademi University, Dept. Computer Science, FIN-20520 Åbo, Finland*

<sup>b</sup>*Umeå University, Dept. Computing Science, SE-901 87 Umeå, Sweden*

<sup>c</sup>*Universidad de Málaga, Dept. Matemática Aplicada, E-29071 Málaga, Spain<sup>2</sup>*

---

## Abstract

Generalisation of the foundational basis for many-valued logic programming builds upon generalised terms in form of powersets of terms. A categorical approach involving set and term functors as monads allows for a study of monad compositions that provide variable substitutions, and compositions thereof. In this paper, substitutions and unifiers appear as constructs in Kleisli categories related to particular composed powerset term monads. Specifically, we show that a frequently used similarity-based approach to fuzzy unification is compatible with the categorical approach, and can be adequately extended in this setting; also some examples are included in order to illuminate the definitions.

---

## 1 Introduction

Many-valued and possibilistic extensions of logic programming involve lattice-theoretic considerations with respect to sets of truth values, and proper handling of many-valued sets of terms to allow for an appropriate theory for unification. Much work has been done focusing only on generalisations of truth values. For a survey on many-valued extensions of propositional calculi, see [13]. Restricting to finitely many truth values, using the framework suggested in [21], a many-valued predicate calculus using conventional terms was proposed in [16]. Recent years show an emerging development allowing also for the use of generalised terms; for instance, an application of fuzzy unification to possibilistic logic can be found in [6]. Achievements seen so far, however, have restricted to constants to be used in various term sets [2,11].

---

<sup>1</sup> This work has been developed as a cooperation organised within COST 274.

<sup>2</sup> Partially supported by MCyT Project BFM-1054-C02-02.

In this paper we propose to use powersets of terms in their full range, i.e. allowing also to use functions in the operator domain. In order to allow for this, we propose the study of *monads* and monad compositions that can categorically mimic composition of variable substitutions involving variable assignments using powersets of terms rather than just terms. A categorical approach provides a well-founded formalism but also reveals properties of powersets of terms required for such compositions of variable substitutions. Generalised terms based on monad compositions, and used in a unification framework, require inventiveness concerning the provision of monad compositions [8]. Further, new monads can be constructed e.g. by using techniques given by submonads [10].

In the classical situation, a substitution of variables can be considered as a mapping  $\sigma : X \rightarrow T_\Omega Y$ , from a set of variables  $X$  to a set of terms over an operator domain  $\Omega$ , using variables in  $Y$ . Given two substitutions  $\nu_1 : X \rightarrow T_\Omega Y$  and  $\nu_2 : Y \rightarrow T_\Omega Z$  its composition  $\nu_2 \star \nu_1 : X \rightarrow T_\Omega Z$  cannot be the composition of the mappings. However, the ‘composite’  $\nu_2 \star \nu_1$  can be defined by the composition

$$X \xrightarrow{\nu_1} T_\Omega Y \xrightarrow{T_\Omega \nu_2} T_\Omega T_\Omega Z \xrightarrow{\mu_Z} T_\Omega Z$$

if we can justify the mapping  $\mu$ . The  $T_\Omega \nu_2$  mapping comes from the functoriality of the term constructor. The ‘flattening’  $\mu_Z$  mapping could be obtained if the functor  $T_\Omega$  is provided with a ‘multiplication’. Later in the paper we will make these notions precise. Moving from conventional terms to a categorical framework involving powersets of terms calls for a level of formalism required to fully make use of the categorical machinery.

It is not an easy task to identify situations in which the underlying behaviour corresponds to a precise categorical concept. For instance, note that in the classical case, terms over terms are flattened to terms. This is possible due to a flattening operator which embeds terms over terms into terms. There exists a categorical construction involving operators comprising the properties above, namely the monad. The existence of such flattening operators is not obvious, and usually not credited to the term monad.

Continuing the discussion about composition, it is easy to see that the expression above corresponds to that of the *Kleisli category* of a monad  $T$ , and this, in fact, is one of the key reasons that makes the categorical unification a success. In [24], it is shown how unification may be considered as an instance of a colimit in a suitable category, and how general constructions of colimits provide recursive procedures for computing the unification of terms.

Approaching the generalisation of terms means also to consider a generalised concept of substitutions, when variables are not replaced by terms but by many-valued sets of terms. Consider a substitution of a variable by a (crisp) set of terms, for instance  $[x/\{t_1, t_3, t_6\}, y/\{t_2, t_3\}]$ . In this case, a variable sub-

stitution, should be  $\nu : X \rightarrow PT_{\Omega}Y$  where  $P$  denotes the powerset functor. The flattening  $\mu_Z : PT_{\Omega}PT_{\Omega}Z \rightarrow PT_{\Omega}Z$  is now far from obvious. However, as the composed functor  $PT_{\Omega}$  can indeed be extended to a monad, the technique to be used is to consider the corresponding Kleisli category of this monad, which enables to compose respective substitutions in a proper way.

Regarding other recent approaches to fuzzy or many-valued unification, there are several references which include concepts as either fuzzy equality relation, or fuzzy equivalence relation, or similarity relation. For a formal treatment of similarities and equalities used in many-valued predicate logics, see [13]. In this paper, we generalise a similarity frequently used between terms, that is, in the image of the functor  $T$ , to a similarity between standard sets of terms and between terms on sets of variables, interpreting the construction in the image of the functors  $PT$  and  $TP$ , respectively; this approach gives us the possibility of defining unifiers between generalised terms.

The paper is organised as follows. In Section 2, we provide required definitions and notations of the categorical framework. Section 3 introduces the concept of similarity, and some examples are developed for selected functors: the term functor, the powerset term functor and the term powerset functor. Later, in Section 4 the definitions of variable substitutions and unifier are given for generalised terms. Section 5 includes some illuminating examples. Finally, in Section 6 some conclusions and pointers to related work are presented.

## 2 Powerset and term monads

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 they are also an important tool in topology when handling regularity, iteratedness and compactifications, and also in the study of toposes and related topics. See [1,3] for category theoretic notions.

As remarked in [3], the naming and identification of monads, in particular as associated with adjoints, can be seen as initiated around 1958. Godement was at that time one of the very first authors to use monads, even if then only named “standard constructions”. Huber in 1961 showed that adjoint pairs give rise to monads. Kleisli [18] and Eilenberg and Moore [7] proved the converse in 1965. The construct of a Kleisli category was thus made explicit in those contributions. Lawvere [19] introduced universal algebra into category theory. This can be seen as the birth of the term monad. These developments then contain all categorical elements for substitution theories. The exploitation of terms and unifications within logic programming is formally described in [22]

as early as in 1965. It is therefore somewhat surprising that the categorical connection to unification was not found until twenty years later by Rydeheard and Burstall in [24].

In the following subsections we will provide definitions and notations for monads and Kleisli categories. Further, we will present those set functors that can be extended to monads, and which are relevant from the many-valued point of view. We will also review some results concerning monad compositions.

### 2.1 Monads and Kleisli categories

A *monad* (or triple, or algebraic theory) over a category  $\mathbf{C}$  is written as  $\Phi = (\Phi, \eta, \mu)$ , where  $\Phi: \mathbf{C} \rightarrow \mathbf{C}$  is a covariant functor, and  $\eta: id \rightarrow \Phi$  and  $\mu: \Phi \circ \Phi \rightarrow \Phi$  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.

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

Composition of morphisms is defined as

$$(X \xrightarrow{f} Y) \circ (Y \xrightarrow{g} Z) = X \xrightarrow{\mu_Z^\Phi \circ \Phi g \circ f} \Phi Z.$$

The Kleisli category is equivalent to the full subcategory of free  $\Phi$ -algebras of the monad, and its definition makes it clear that arrows are substitutions.

The term functor  $T_\Omega$ , or  $T$  for short, with  $TX$  being the set of terms over the operator domain  $\Omega$  and the variable set  $X$ , is extended to a monad in the usual way. The categorical unification algorithm in [24] is based on the Kleisli category of the term monad.

In this paper we will follow the categorical notation adopted in [8], where a term  $\omega(m_1, \dots, m_n)$  is more formally written as  $(n, \omega, (m_i)_{i \leq n})$ .

### 2.2 Set functors extended to monads

In the previous section we have used the, perhaps more intuitive, notation  $PX$  to denote the powerset of  $X$ . From now on, in a generalised context we feel more comfortable with the notation  $2X$  to represent the classical *boolean* powerset, according to the general notation  $LX$  where  $L$ -fuzzy subsets are considered.

Consider ordinary powersets and let  $2X$  denote the set of subsets of  $X$ , so that  $2$  denotes the ordinary powerset functor. With  $\eta_X : X \rightarrow 2X$  given by  $\eta_X(x) = \{x\}$  and  $\mu_X : 22X \rightarrow 2X$  given by  $\mu_X(\mathbf{B}) = \bigcup \mathbf{B}$ , we have the well-known ordinary powerset monad  $\mathbf{2} = (2, \eta, \mu)$ .

The extension to many-valued sets is according to [12]. Let  $L$  be a completely distributive lattice. The covariant powerset functor  $L_{id}$  is obtained by  $L_{id}X = L^X$ , i.e. the set of mappings (or  $L$ -fuzzy sets)  $\alpha : X \rightarrow L$ . For a morphism  $X \xrightarrow{f} Y$  in  $\mathbf{Set}$ , we use

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

Further,  $\eta_X : X \rightarrow L_{id}X$  is defined by

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

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

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

In [20] it was proved that  $\mathbf{L}_{id} = (L_{id}, \eta, \mu)$  is a monad. Note that for  $L = \{0, 1\}$  we have  $\mathbf{L}_{id} = \mathbf{2}$ .

### 2.3 Monad compositions

Monad compositions require the use of a swapper transformation  $\sigma$  (see below) together with conditions related to this swapper. In [4], a set of such conditions was given and the conditions were called *distributive laws*. Conditions on monad composability are discussed also e.g. in [5,8,9,15].

In [8] we made use of a swapper  $\sigma_X : TLX \rightarrow LTX$  defined in the base case by  $\sigma_X|_{T^0LX} = id_{LX}$ , and further inductively given by

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

where  $l = (n, \omega, (l_i)_{i \leq n}) \in T^\alpha LX$ ,  $\alpha > 0$ ,  $l_i \in T^{\beta_i} LX$ ,  $\beta_i < \alpha$ , and it was proved that the composition of the monads  $\mathbf{L}_{id}$  and  $\mathbf{T}_\Omega$  is as well a monad.

In case of  $L = 2$ , the swapper  $\sigma_X : T2X \rightarrow 2TX$  coincides with the expected result given by intuition, and can be written in set-theoretical terms as follows:

for the base case  $\sigma_X|_{2X} = id_{2X}$  and, otherwise by recursion as

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

The natural transformation  $\eta^{2T}: id \rightarrow 2T$  specializes to  $\eta_X^{2T}(x) = \{x\}$ , and the natural transformation  $\mu^{2T}: 2T2T \rightarrow 2T$  is for  $R = \{(n_j, \omega_j, (r_{ij})_{i \leq n_j}) \mid j \in J\} \in 2T2TX$  given by

$$\{(n_j, \omega_j, (m_{ij})_{i \leq n_j}) \mid j \in J, m_{ij} \in \sigma_{TX}(r_{ij})\}$$

Later the *ad hoc* swapper construction was abstracted and, in [9] a generalisation of Beck's conditions was given as follows

Let  $\Phi = (\Phi, \eta^\Phi, \mu^\Phi)$  and  $\Psi = (\Psi, \eta^\Psi, \mu^\Psi)$  be monads and let  $\sigma: \Psi \circ \Phi \rightarrow \Phi \circ \Psi$  be a natural transformation such that the following conditions hold:

$$\begin{aligned} \sigma \circ \eta^\Psi \Phi &= \Phi \eta^\Psi, \\ \sigma \circ \Psi \eta^\Phi &= \eta^\Phi \Psi, \\ \Phi \mu^\Psi \circ \sigma \Psi \circ \Psi \mu^\Phi \Psi \circ \Psi \Phi \sigma &= \mu^\Phi \Psi \circ \Phi \sigma \circ \Phi \mu^\Psi \Phi \circ \sigma \Psi \Phi. \end{aligned}$$

Then  $\Phi \bullet \Psi = (\Phi \circ \Psi, \eta^\Phi \Psi \circ \eta^\Psi, \Phi \mu^\Psi \circ \mu^\Phi \Psi \Psi \circ \Phi \sigma \Psi)$  is a monad.

The result above gave a set of sufficient conditions for the compositionality of two given monads, provided we have a swapper. Although the result above provides a very abstract framework for generalised terms, for the purposes of this paper, when discussing possible definitions for unifiers using similarities, we will use the **2T** monad only.

### 3 Similarities

There are several references in the literature that include concepts as either fuzzy equality relation, or fuzzy equivalence relation, or similarity relation. We will adopt the latter terminology which will be formally defined later. It should be remarked that it is useful to consider similarities from a topos-theoretic point of view. The topos oriented situation where Heyting algebras have idempotent conjunctions is not acceptable in many-valued considerations and thus we need amendments such as those involving monoidal non-idempotent conjunctions, leading to the weak topos framework, see [26,14] for more detail. The position of this paper is, however, not to include topos-theoretic considerations at this stage, as our primary purpose is to demonstrate how the concept of unifiers can be extended to consider also powersets of terms.

In this section, we recall the concept of similarity and, for selected functors

such as the term functor, the powerset term functor and the term powerset functor, some examples are developed. We will assume that  $L$  denotes a completely distributive lattice.

A *similarity* on  $X$  is a mapping  $E: X \times X \rightarrow L$  satisfying

$$\begin{aligned} E(x, x) &= 1 && \text{(reflexivity)} \\ E(x, y) &= E(y, x) && \text{(symmetry)} \\ E(x, y) \wedge E(y, z) &\leq E(x, z) && \text{(transitivity)} \end{aligned}$$

for all  $x, y, z \in X$ .

Given two terms  $(n, \omega, (m_i)_{i \leq n})$  and  $(n', \omega', (m'_i)_{i \leq n'})$ , we need now to obtain a similarity between them. Intuitively, we must join the similarity between  $\omega$  and  $\omega'$  with the combined similarities between respective components  $m_i$  and  $m'_i$ . In order to be more precise, let  $\Omega$  be a set of operations, and let further  $E_\Omega$  be a similarity on  $\Omega$ , which will be used to define a similarity on  $TX$ .

**Proposition 3.1** *Let  $E_T: TX \times TX \rightarrow L$  be a relation on  $TX$  such that*

$$E_T(x, x) = 1 \tag{1}$$

for all  $x \in X$ , and for terms  $t = (n, \omega, (m_i)_{i \leq n})$ , and  $t' = (n', \omega', (m'_i)_{i \leq n'})$ ,

$$E_T(t, t') = \begin{cases} E_\Omega(\omega, \omega') \wedge \bigwedge_{i \leq n} E_T(m_i, m'_i) & \text{if } n = n', \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Then  $E_T$  is a similarity on  $TX$ .

*Proof:* The proof considers the iterative construction in order to build upon the fact that a join of equality relations is again an equality relation.

The reflexivity of  $E_T$  follows easily by structural induction and the reflexivity of  $E_\Omega$ .

Using symmetry of  $E_\Omega$  we obtain

$$\begin{aligned} E_T\left(\left(n, \omega, (m_i)_{i \leq n}\right), \left(n, \omega', (m'_i)_{i \leq n}\right)\right) &= E_\Omega(\omega, \omega') \wedge \bigwedge_{i \leq n} E_T(m_i, m'_i) \\ &= E_\Omega(\omega', \omega) \wedge \bigwedge_{i \leq n} E_T(m'_i, m_i) \\ &= E_T\left(\left(n, \omega', (m'_i)_{i \leq n}\right), \left(n, \omega, (m_i)_{i \leq n}\right)\right). \end{aligned}$$

Transitivity of  $E_\Omega$  gives

$$\begin{aligned}
& E_T\left(\left(n, \omega, (m_i)_{i \leq n}\right), \left(n, \omega', (m'_i)_{i \leq n}\right)\right) \wedge \\
& \qquad \qquad \qquad E_T\left(\left(n, \omega', (m'_i)_{i \leq n}\right), \left(n, \omega'', (m''_i)_{i \leq n}\right)\right) \\
& = E_\Omega(\omega, \omega') \wedge \bigwedge_{i \leq n} E_T(m_i, m'_i) \wedge E_\Omega(\omega', \omega'') \wedge \bigwedge_{i \leq n} E_T(m'_i, m''_i) \\
& = E_\Omega(\omega, \omega') \wedge E_\Omega(\omega', \omega'') \wedge \bigwedge_{i \leq n} \left(E_T(m_i, m'_i) \wedge E_T(m'_i, m''_i)\right) \\
& \leq E_\Omega(\omega, \omega'') \wedge \bigwedge_{i \leq n} E_T(m_i, m''_i) \\
& = E_T\left(\left(n, \omega, (m_i)_{i \leq n}\right), \left(n, \omega'', (m''_i)_{i \leq n}\right)\right).
\end{aligned}$$

■

Note that the definition of similarity on  $TX$  adopted in this paper differs from the one adopted in [25] in that the latter requires additionally that  $E_T(x_1, x_2) = 0$  if  $x_1 \neq x_2$ .

For unification purposes we will need a similarity between powersets of terms, and for this purpose we will now use  $E_T$  in order to define a similarity on  $2TX$ . The choice of this similarity on  $2TX$  is less obvious than the one on  $TX$ . For the similarity on  $2TX$  we adopt the viewpoint of combining all similarities of pairwise terms from respective sets of terms.

**Proposition 3.2** *Let  $E_T$  be a relation satisfying conditions (1) and (2). The relation*

$$E_{2T}: 2TX \times 2TX \rightarrow L$$

*defined for all  $M_1, M_2 \in 2TX$  as*

$$E_{2T}(M_1, M_2) = \bigwedge_{m_1 \in M_1} \bigvee_{m_2 \in M_2} E_T(m_1, m_2) \wedge \bigwedge_{m_2 \in M_2} \bigvee_{m_1 \in M_1} E_T(m_1, m_2) \quad (3)$$

*is a similarity on  $2TX$ .*

*Proof:* The proof is included for sake of completeness. A similar proof has been presented in [23] in the case of implication measures.

Reflexivity and symmetry are obvious. For transitivity, we will prove two partial inequalities, in which the notation  $M^N$  means the set of all the functions from  $N$  to  $M$ .



Firstly,

$$\begin{aligned}
\bigvee_{m \in M} E_T(m_1, m) \wedge \bigwedge_{m \in M} \bigvee_{m_2 \in M_2} E_T(m, m_2) &= \bigvee_{m \in M} E_T(m_1, m) \wedge \bigvee_{f \in M_2^M} \bigwedge_{m' \in M} E_T(m', f(m')) \\
&= \bigvee_{\substack{m \in M \\ f \in M_2^M}} \bigwedge_{m' \in M} \left( E_T(m_1, m) \wedge E_T(m', f(m')) \right) \\
&\leq \bigvee_{\substack{m \in M \\ f \in M_2^M}} \left( E_T(m_1, m) \wedge E_T(m, f(m)) \right) \\
&\leq \bigvee_{\substack{m \in M \\ f \in M_2^M}} \left( E_T(m_1, f(m)) \right) \\
&= \bigvee_{m_2 \in M_2} E_T(m_1, m_2)
\end{aligned}$$

and similarly

$$\bigvee_{m \in M} E_T(m, m_2) \wedge \bigwedge_{m \in M} \bigvee_{m_1 \in M_1} E_T(m_1, m) \leq \bigvee_{m_1 \in M_1} E_T(m_1, m_2).$$

Thus, transitivity follows from

$$\begin{aligned}
E_{2T}(M_1, M) \wedge E_{2T}(M, M_2) &= \\
&= \bigwedge_{m_1 \in M_1} \bigvee_{m \in M} E_T(m_1, m) \wedge \bigwedge_{m \in M} \bigvee_{m_1 \in M_1} E_T(m_1, m) \\
&\quad \wedge \bigwedge_{m \in M} \bigvee_{m_2 \in M_2} E_T(m, m_2) \wedge \bigwedge_{m_2 \in M_2} \bigvee_{m \in M} E_T(m, m_2) \\
&= \bigwedge_{m_1 \in M_1} \left( \bigvee_{m \in M} E_T(m_1, m) \wedge \bigwedge_{m \in M} \bigvee_{m_2 \in M_2} E_T(m, m_2) \right) \\
&\quad \wedge \bigwedge_{m_2 \in M_2} \left( \bigvee_{m \in M} E_T(m, m_2) \wedge \bigwedge_{m \in M} \bigvee_{m_1 \in M_1} E_T(m_1, m) \right) \\
&\leq \bigwedge_{m_1 \in M_1} \bigvee_{m_2 \in M_2} E_T(m_1, m_2) \wedge \bigwedge_{m_2 \in M_2} \bigvee_{m_1 \in M_1} E_T(m_1, m_2) \\
&= E_{2T}(M_1, M_2)
\end{aligned}$$

■

Regarding similarities in  $2T$ , recall the approach used in [11] which introduced the pseudosimilarity (it is not reflexive) below

$$E'(M_1, M_2) = \bigwedge_{m_1, m_2 \in M_1 \cup M_2} E'_T(m_1, m_2).$$

however, the proposition above shows that there are reasonable similarities in  $2T$ .

A similarity on  $T2X$  can now easily be given using the similarity on  $2TX$ .

**Proposition 3.3** *The relation*

$$E_{T2}: T2X \times T2X \rightarrow L$$

defined as

$$E_{T2}(l_1, l_2) = E_{2T}(\sigma_X(l_1), \sigma_X(l_2)) \quad (4)$$

where  $\sigma_X: T2X \rightarrow 2TX$  is the swapper, is a similarity on  $T2X$ .

*Proof:* Reflexivity and symmetry are obvious. Transitivity follows from

$$\begin{aligned} E_{T2}(l_1, l) \wedge E_{T2}(l, l_2) &= E_{2T}(\sigma_X(l_1), \sigma_X(l)) \wedge E_{2T}(\sigma_X(l), \sigma_X(l_2)) \\ &\leq E_{2T}(\sigma_X(l_1), \sigma_X(l_2)) \\ &= E_{T2}(l_1, l_2). \end{aligned}$$

■

Note that, although we have the similarity of  $T2$ , it is still an open question whether the functor composition  $T2$  can be extended to a monad. This is why in the following section we restrict our attention to standard powersets of terms.

## 4 Unifiers

In this section we introduce the concepts of variable substitutions and unifiers, in particular within the context of powersets of terms. In the classical situation, variable substitutions are mappings assigning variables to terms, i.e. mappings  $\theta: X \rightarrow TY$ .

For powersets of terms in a generalised setting, given a monad  $\Phi$  such that the composition  $\Phi \bullet \mathbf{T}_\Omega$  still provides a monad, variable substitutions should then be viewed as mappings

$$\theta: X \rightarrow \Phi TY.$$

Given a term  $M \in \Phi TX$  in form of a generalised powerset of terms, the result

$M_1$	$M_2$	$E_{2T}(M_1, M_2)$
$\{t_1\}$	$\{t_1, t_2, t_3\}$	$E_T(t_1, t_2) \wedge E_T(t_1, t_3)$
$\{u_1(c_1)\}$	$\{u_1(u_1(c_1))\}$	0
$\{u_1(c_1)\}$	$\{u_2(x)\}$	$E_\Omega(u_1, u_2) \wedge E_T(c_1, x)$
$\{u_1(c_1), u_1(c_2)\}$	$\{u_2(c_1), u_2(c_2)\}$	$E_\Omega(u_1, u_2)$
$\{u_1(c_1), u_2(c_2)\}$	$\{u_1(c_2), u_2(c_1)\}$	$E_\Omega(u_1, u_2) \vee E_\Omega(c_1, c_2)$

Table 1

Equality of generalised terms.

$M\theta$  of applying a variable substitution  $\theta$  on  $M$  is given by

$$M\theta = (\mu_Y^{\Phi T} \circ \Phi T\theta)(M)$$

i.e.  $M\theta$  is kind of a flattening of a set of terms over sets of terms, where  $\mu_Y^{\Phi T}$  provides the flattening operation. Note that in the particular case of  $\Phi$  being the identity monad,  $M\theta$  is nothing but the expected classical result when applying the variable substitution  $\theta$  to the term  $M \in TX$ .

We will focus on the particular case of  $\Phi = \mathbf{2}$ . To illuminate our constructions, let  $\Omega = \{c_1, c_2, u_1, u_2\}$ , where  $c_i$  are nullary operators (constants), and  $u_i$  unary operators,  $i = 1, 2$ . Further, let  $x$  be a variable. Table 1 shows similarity values for some typical pairs of generalised terms. For  $M_2 = \{u_2(x)\}$ , note the effect of the variable substitutions  $\theta(x) = c_1$  and  $\theta(x) = u_1(c_1)$ .

Variable substitutions can obviously be defined more generally over monads  $(F, \eta^F, \mu^F)$ . Indeed for an object  $A \in FX$ , and a variable substitution  $\theta: X \rightarrow FY$ , we will have

$$A\theta = (\mu_Y^F \circ F\theta)(A).$$

When composing substitutions, the utility of the flattening operator again becomes explicit.

**Definition 4.1** *The composition of two substitutions  $\theta': X \rightarrow \Phi TY$  and  $\theta'': Y \rightarrow \Phi TZ$  is given by*

$$\theta'\theta'' = \mu_Z^{\Phi T} \circ \Phi T\theta'' \circ \theta'$$

*i.e. the composition in the Kleisli category  $\mathbf{Set}_{\Phi \mathbf{T}}$  for the powerset monad  $\Phi \bullet \mathbf{T}$  over the category of sets.*

For  $\Phi$  being the identity monad, variable substitutions correspond to the classical case, where use of the idempotency of “terms over terms” is usually not made very explicit.

To continue with the case of  $\Phi = \mathbf{2}$ , given  $M_1, M_2 \in 2TX$ , let  $[M_1; M_2]$

represent an *equation* over  $2TX$ . In order to propose a definition of generalised unifiers for equations we will assume the existence of similarities, as those in the previous section.

**Definition 4.2** *A unifier of the equation  $[M_1; M_2]$  over  $2TX$  is a substitution,  $\theta: X \rightarrow 2TY$ , such that  $E_{2T}(M_1\theta, M_2\theta)$  equals*

$$\sup\{E_{2T}(M_1\vartheta, M_2\vartheta) \mid \vartheta \text{ is a substitution}\}.$$

It might be possible that the supremum above could not be attained by any substitution. The particular features of the lattice or the underlying application might require a weaker version of the definition, as follows:

Let  $\theta$  be a substitution, and  $[M_1; M_2]$  an equation over  $2TX$ . We say that  $\theta$  is a unifier if  $E_{2T}(M_1, M_2) \leq E_{2T}(M_1\theta, M_2\theta)$ , that is, if the substitution increases the similarity degree. Note that a unifier  $\theta$  is a so called *extensional mapping* according to [17].

## 5 Examples

Examples in this section are included in order to provide illuminations of our notions. More elaborate examples can be found in application papers, such as those dealing with fuzzy control.

### 5.1 Similarity between numerical functions

A possible situation in which the use of similarities is required is given below as a similarity between numerical real-valued functions. The similarity is defined in terms of a ‘uniform distance.’ Suppose that when solving a functional equation, we apply different iterative methods and, therefore, different approximations are obtained. The different approximations obtained to the actual solution  $f$  are classified in classes of functions  $[f]_{\epsilon_i}$  according to an increasing numerable sequence of bounds  $\epsilon_1, \epsilon_2, \dots, \epsilon_n < 1$  and the following measure: A function  $g$  is in the class  $[f]_{\epsilon_i}$  if and only if  $|f(x) - g(x)| < \epsilon_i$  and  $g \notin [f]_{\epsilon_j}$  for all  $j < i$ .

Let us designate a canonical representative  $g_{\epsilon_i}$  for each class  $[f]_{\epsilon_i}$ , and consider the set  $\Omega = \{f, g_{\epsilon_i}\}_{1 \leq i \leq n}$ . The relation  $E_\Omega: \Omega \times \Omega \rightarrow [0, 1]$  given by

$$\begin{aligned} E_\Omega(f, f) &= 1 = E_\Omega(g_{\epsilon_i}, g_{\epsilon_i}), \\ E_\Omega(g_{\epsilon_i}, g_{\epsilon_j}) &= 1 - \max\{\epsilon_i, \epsilon_j\} \quad \text{if } i \neq j \end{aligned}$$

and

$$E_{\Omega}(f, g_{\epsilon_i}) = 1 - \epsilon_i = E_{\Omega}(g_{\epsilon_i}, f)$$

indeed defines a similarity relation. The similarities  $E_{2T}$  and  $E_T$  are defined in terms of  $E_{\Omega}$  as stated previously.

Now, the similarity between two composite functions, for instance  $g_{\epsilon_i}(f(x))$  and  $f(g_{\epsilon_i}(x))$ , can be considered as *terms* over the variable  $x$  and, therefore, calculated using  $E_T$  by the expression

$$E_T(g_{\epsilon_i}(f(x)), f(g_{\epsilon_i}(x))) = 1 - \epsilon_i.$$

Furthermore, we compare the similarity of two sets of functions as sets of terms:

$$M_1 = \{f(f(x)), f(g_{\epsilon}(x))\}$$

$$M_2 = \{g_{\epsilon}(f(x)), f(g_{\epsilon}(x))\}$$

The similarity between  $M_1$  and  $M_2$  is then given by

$$E_{2T}(M_1, M_2) = 1 - \epsilon.$$

Let us now consider the sets  $N_1$  and  $N_2$ :

$$N_1 = \{f(a), g_{\epsilon}(x_1)\}$$

$$N_2 = \{f(x_1), g_{\epsilon}(a)\}$$

Then,  $E_{2T}(N_1, N_2) = 1 - \epsilon$ .

Let  $\theta_1, \theta_2: X \rightarrow 2TY$  be the substitutions given by  $\theta_1(x_1) = \{a, y\}$ , and  $\theta_2(x_1) = \{a\}$ . Then  $E_{2T}(N_1\theta_1, N_2\theta_1) = 1 - \epsilon$  and  $E_{2T}(N_1\theta_2, N_2\theta_2) = 1$ . Note that in this case  $\theta_2$  is a most general unifier. In fact the composition of  $\theta_1$  and  $\theta_2$  is again  $\theta_1$ .

## 5.2 Qualification for course participance

In order to qualify for participance in a course,  $C$ , the students should (preferentially) have taken the courses  $c_1, c_2, c_3$ . Alternatively, students may obtain part of the contents of these courses by studying the courses  $n_1, n_2$ .

The lectures for the courses  $c_1$  and  $n_1$  are held at the same time. The same happens for the courses  $c_2$  and  $n_2$ . This means that the students should choose between studying either  $c_1$  or  $n_1$  and studying either  $c_2$  or  $n_2$ .

The relation between the contents in the different courses are given by the similarity relation in Table 2. In order to admit students to participate in  $C$ ,

$E$	$c1$	$c2$	$c3$	$n1$	$n2$
$c1$	1	0.2	0.2	0.6	0.2
$c2$	0.2	1	0.2	0.2	0.7
$c3$	0.2	0.2	1	0.2	0.2
$n1$	0.6	0.2	0.2	1	0.2
$n2$	0.2	0.7	0.2	0.2	1

Table 2

Similarities between courses.

it is needed to check which students have an education more ‘similar’ to the one required.

In the following a student will be represented by the set of courses (s)he has studied. Let  $R = \{c_1, c_2, c_3\}$  be the set of requirements. The student  $S_1 = \{c_1, n_2\}$  has studied the courses  $c_1, n_2$ . The level of agreement with the  $C$  course for this student is then  $E_{2T}(S_1, R) = 0.2$ .

For  $S_2 = \{c_3, n_1, n_2\}$ , the level of agreement is  $E_{2T}(S_2, R) = 0.6$ , and for  $S_3 = \{c_2, c_3, n_1\}$ , the level is also  $E_{2T}(S_3, R) = 0.6$ .

If we know that a student has taken at least the courses  $c_1, n_2$ , i.e.  $S_x = \{c_1, n_2, x\}$ , then a most general unifier in this case is  $\theta(x) = \{c_3\}$  with corresponding level of agreement  $E_{2T}(S_x\theta, R) = 0.7$ . Note that  $\theta'(x) = \{c_2, c_3\}$  is also a unifier. Then  $E_{2T}(S_x\theta', R) = 0.7$  and not 1 as we might have expected. The reason for this is that we are comparing similarity of sets, even if in a real case this unifier is not worth to be considered since the student cannot study  $c_2$  and  $n_2$  simultaneously.

## 6 Conclusions and future work

We have shown how generalised terms, as given by powersets of terms, can be handled in equational settings involving substitutions and unifiers. The utility of categorical techniques as provided by monads is obvious and indeed encouraging for further investigations on more elaborate compositions and categorical techniques for unification as initiated in [24].

In further work it is important to merge our efforts with developments, such as in [2], that have focused more on semantic aspects of many-valued logic programming. Semantic approaches seem to be fruitful in particular within possibilistic logic frameworks. These developments, however, still have a rather specialised use of terms as they typically restrict to using powersets of constants instead of generalised terms in their full range. However, restricting to

powersets of constants seems more to be a struggle with unification than with proof procedural issues, and there are no indications that the specialised use of terms is enforced by the semantic developments, such as those seen in possibilistic logic. The procedural issues being important it is equally worthwhile to underline the importance of further studies on monad compositions.

**Acknowledgements** We are grateful to the anonymous referees who helped us significantly to improve the paper and also drew our attention to further categorical issues to be considered in future work related to these problems.

## References

- [1] J. Adámek, H. Herrlich, G. Strecker, *Abstract and Concrete Categories*, John Wiley & Sons, 1990.
- [2] T. Alsinet, L. Godo, *A complete calculus for possibilistic logic programming with propositional variables*, Proc. Uncertainty in AI'00 conference, 2000, pages 1–10.
- [3] M. Barr, C. Wells, *Toposes, Triples and Theories*, Springer-Verlag, 1985.
- [4] J. Beck, *Distributive laws*, Seminars on Triples and Categorical Homology Theory, 1966/67, LNM 80, pp. 119–140. Springer-Verlag, 1969.
- [5] J. R. Caruncho Castro, *Triples theory* (in Spanish), Department of Algebra of the Univ. de Santiago de Compostela, Reports Series, Algebra 5, 1971.
- [6] D. Dubois, H. Prade, S. Sandri, Possibilistic logic augmented with fuzzy unification. In *Proc. of IPMU'96*, pp. 1009–1014, Granada, Spain, 1996.
- [7] S. Eilenberg, J. C. Moore, *Adjoint functors and triples*, Illinois J. Math. **9** (1965), 381–398.
- [8] P. Eklund, M.A. Galán, M. Ojeda-Aciego, A. Valverde, *Set functors and generalised terms*, Proc. IPMU 2000, 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, vol. III, 2000, 1595–1599.
- [9] P. Eklund, M.A. Galán, J. Medina, M. Ojeda-Aciego, A. Valverde, *Generalised terms and composition of monads*, Proc. X Spanish Conference on Fuzzy Logic and Technologies (ESTYLF 2000), 155–160.
- [10] P. Eklund, M.A. Galán, J. Medina, M. Ojeda-Aciego, A. Valverde, *Composing submonads*, Proc. 31st IEEE Int. Symposium on Multiple-Valued Logic (ISMVL 2001), May 22-24, 2001, Warsaw, Poland, 367–372.
- [11] F. Formato, G. Gerla, M. I. Sessa, *Similarity-based unification*, Fundamenta Informaticae **41**(4) (2000), 393–414.

- [12] J. A. Goguen, *L-fuzzy sets*, J. Math. Anal. Appl. **18** (1967), 145–174.
- [13] P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer Academic Publishers, 1998.
- [14] U. Höhle, *Presheaves over GL-monoids*, in U. Höhle and Klement, E.P., eds., Non-classical logics and their applications to fuzzy subsets, pp. 127–157 Kluwer Academic Publishers, 1995.
- [15] M. P. Jones, L. Duponcheel, *Composing Monads*, Research Report YALEU/DCS/RR-1004, Yale University, New Haven, Connecticut, USA, December 1993.
- [16] F. Klawonn, R. Kruse, *A Lukasiewicz logic based Prolog*, Mathware and Soft Computing **1** (1994), 5–29.
- [17] F. Klawonn, J. Gebhardt, R. Kruse, *Fuzzy Control on the Basis of Equality Relations - with an Example from Idle Speed Control*, IEEE Trans. Fuzzy Systems **3** (1995), 336–350.
- [18] H. Kleisli, *Every standard construction is induced by a pair of adjoint functors*, Proc. Amer. Math. Soc. **16** (1965), 544–546.
- [19] F. W. Lawvere, *Functorial semantics of algebraic theories*, Dissertation, Columbia University, 1963.
- [20] E. G. Manes, *Algebraic Theories*, Springer-Verlag, 1976.
- [21] J. Pavelka, *On fuzzy logic I, II, III*, Zeitschr. Math. Logik Grundle. Math. **25** (1979), 45–52, 119–134, 447–464.
- [22] J. A. Robinson, *A machine-oriented logic based on the resolution principle*, J. ACM **12** (1965), 23–41.
- [23] E. H. Ruspini, *On the semantics of fuzzy logic*, Intl. J. Approx. Reasoning **5** (1991), 45–88.
- [24] D. E. Rydeheard, R. M. Burstall, *A categorical unification algorithm*, Proc. Summer Workshop on Category Theory and Computer Programming, 1985, LNCS 240, Springer-Verlag, 1986, 493–505.
- [25] M. I. Sessa, *Translations and similarity-based logic programming*, Soft Computing **5** (2001), 160–170.
- [26] L. N. Stout, *A survey of fuzzy set and topos theory*, Fuzzy Sets and Systems **42** (1991), 3–14.