Similarities between powersets of terms

P. Eklund\textsuperscript{a}, M.A. Galán\textsuperscript{b}, J. Medina\textsuperscript{c,1}, M. Ojeda-Aciego\textsuperscript{c,*,1}, A. Valverde\textsuperscript{c,1}

\textsuperscript{a}Department of Computer Science, Åbo Akademi University, Åbo FIN-20520, Finland
\textsuperscript{b}Department of Computing Science, Umeå University, Umeå SE-901 87, Sweden
\textsuperscript{c}Dept. Matemática Aplicada, Universidad de Málaga, Málaga E-29071, Spain

Abstract

Generalisation of the foundational basis for many-valued logic programming builds upon generalised terms in the 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.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Similarities; Fuzzy unification; Category theory and unification; Generalised terms

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 [17]. Recent years show an emerging development that allows also for the use of generalised terms; for instance, an application of fuzzy unification

\textsuperscript{*} This work has been developed as a cooperation organised within COST 274.
\textsuperscript{*} Corresponding author. Dept. Matemática Aplicada, Universidad de Málaga, Aptdo. de Correos 4114, Málaga E-29080, Spain.
\textsuperscript{E-mail address: aciego@ctima.uma.es} (M. Ojeda-Aciego).
\textsuperscript{1} Partially supported by MCyT Project BFM-1054-C02-02.
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].

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 and also reveal the 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 [10]. Further, new monads can be constructed e.g. by using techniques given by submonads [9].

In the classical situation, a substitution of variables can be considered as a mapping \( \sigma : X \to T_\Omega Y \), from a set of variables \( X \) to a set of terms over an operator domain \( \Omega \), using variables in \( Y \). Given two substitutions \( v_1 : X \to T_\Omega Y \) and \( v_2 : Y \to T_\Omega Z \), its composition \( v_2 \star v_1 : X \to T_\Omega Z \) cannot be the composition of the mappings. However, the ‘composite’ \( v_2 \star v_1 \) can be defined by the composition

\[
X \xrightarrow{v_1} T_\Omega Y \xrightarrow{T_\Omega v_2} T_\Omega T_\Omega Z \xrightarrow{\mu_Z} T_\Omega Z
\]

if we can justify mapping \( \mu \). The \( T_\Omega v_2 \) 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 substitution, should be \( v : X \to P T_\Omega Y \) where \( P \) denotes the powerset functor. The flattening \( \mu_Z : P T_\Omega P T_\Omega Z \to P T_\Omega Z \) is now far from obvious. However, as the composed functor \( P T_\Omega \) 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, then only named as “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 $\mathcal{C}$ is written as $\Phi = (\Phi, \eta, \mu)$, where $\Phi: \mathcal{C} \to \mathcal{C}$ is a covariant functor, and $\eta: \text{id} \to \Phi$ and $\mu: \Phi \circ \Phi \to \Phi$ are natural transformations for which $\mu \circ \Phi \mu = \mu \circ \mu \Phi$ and $\mu \circ \Phi \eta = \mu \circ \eta \Phi = \text{id}_\Phi$ hold.

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

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

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 [10], where a term \(\omega(m_1, \ldots, m_n)\) is more formally written as \((n, \omega,(m_i)_{i \in n})\).

### 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 \(\eta_X : X \to 2X\) given by \(\eta_X(x) = \{x\}\) and \(\mu_X : 22X \to 2X\) given by \(\mu_X(B) = \bigcup B\), we have the well-known ordinary powerset monad \(2 = (2, \eta, \mu)\).

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) \(\alpha : X \to L\). For a morphism \(X \xrightarrow{f} Y\) in \(\text{Set}\), we use

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

Further, \(\eta_X : X \to L_{id}X\) is defined by

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

and \(\mu_X : L_{id}L_{id}X \to L_{id}X\) by

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

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

### 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 [5,8,10,15].
In [10] we made use of a swapper $\sigma_X : TLX \rightarrow LTX$ defined in the base case by $\sigma_X|^{TLX} = id_{LX}$, 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 $l = (n, \omega, (l_i)_{i \leq n}) \in T^xLX$, $x > 0$, $l_i \in T^\beta LX$, $\beta_i < x$, and it was proved that the composition of the monads $L_{id}$ and $T_{\beta}$ 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

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

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 $R = \{(n_j, \omega_j, (r_{ij})_{i \leq n_j}) | j \in J\} \in 2T2TX$ given by

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

Later the ad hoc swapper construction was abstracted and, in [8] 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:

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

Then $\Phi \bullet \Psi = (\Phi \circ \Psi, \eta^\Phi \Psi \circ \eta^\Psi, \Phi \mu^\Psi \circ \mu^\Phi \Psi \circ \Phi \sigma \Psi)$ 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 [14,26] for more details. 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 powersons 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 \( E : X \times X \to L \) satisfying

\[
E(x,x) = 1 \quad \text{(reflexivity)},
\]
\[
E(x,y) = E(y,x) \quad \text{(symmetry)},
\]
\[
E(x,y) \land E(y,z) \leq E(x,z) \quad \text{(transitivity)}
\]
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 \( E_T : TX \times TX \to L \) be a relation on \( TX \) such that

\[
E_T(x,x) = 1
\]
for all \( x \in X \), and for terms \( t = (n, \omega, (m_i)_{i \leq n}) \) and \( t' = (n', \omega', (m'_i)_{i \leq n'}) \),

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

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

\[
E_T((n, \omega, (m_i)_{i \leq n}), (n, \omega', (m'_i)_{i \leq n})) = E_\Omega(\omega, \omega') \land \bigwedge_{i \leq n} E_T(m_i, m'_i)
\]
\[
= E_\Omega(\omega', \omega) \land \bigwedge_{i \leq n} E_T(m'_i, m_i)
\]
\[
= E_T((n, \omega', (m'_i)_{i \leq n}), (n, \omega, (m_i)_{i \leq n})).
\]

Transitivity of \( E_\Omega \) gives

\[
E_T((n, \omega, (m_i)_{i \leq n}), (n, \omega', (m'_i)_{i \leq n})) \land E_T((n, \omega', (m'_i)_{i \leq n}), (n, \omega'', (m''_i)_{i \leq n}))
\]
\[
= E_\Omega(\omega, \omega') \land \bigwedge_{i \leq n} E_T(m_i, m'_i) \land E_\Omega(\omega', \omega'') \land \bigwedge_{i \leq n} E_T(m'_i, m''_i)
\]
\[
= E_\Omega(\omega, \omega') \land E_\Omega(\omega', \omega'') \land \bigwedge_{i \leq n} (E_T(m_i, m'_i) \land E_T(m'_i, m''_i))
\]
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 $2^TX$. The choice of this similarity on $2^TX$ is less obvious than the one on $TX$. For the similarity on $2^TX$ 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

$$E_{2T} : 2^TX \times 2^TX \to L$$

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

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

(3)

is a similarity on $2^TX$.

**Proof.** The proof is included for the 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,

$$\bigvee_{m \in M} E_T(m_1, m) \land \bigwedge_{m \in M} \bigvee_{m_2 \in M_2} E_T(m, m_2) = \bigvee_{m \in M} E_T(m_1, m) \land \bigvee_{f \in M_2^M} \bigwedge_{m' \in M} E_T(m', f(m'))$$

$$= \bigwedge_{m \in M} E_T(m_1, m) \land \bigvee_{f \in M_2^M} \bigwedge_{m' \in M} E_T(m', f(m'))$$

$$= \bigvee_{m \in M} \bigwedge_{m' \in M} (E_T(m_1, m) \land E_T(m', f(m')))$$

$$\leq \bigvee_{m \in M} (E_T(m_1, m) \land E_T(m, f(m)))$$

$$= \bigvee_{m \in M} (E_T(m_1, m) \land E_T(m, f(m)))$$

$$= \bigvee_{m \in M} E_T(m_1, m)$$

$$= \bigvee_{m \in M} E_T(m_1, m_2)$$

$$= \bigvee_{m \in M} E_T(m_1, m_2)$$
and similarly,
\[
\bigvee_{m \in M} E_T(m, m_2) \land \bigwedge_{m \in M} \bigvee_{m_1 \in M_1} E_T(m_1, m) \leq \bigvee_{m_1 \in M_1} E_T(m_1, m_2).
\]

Thus, transitivity follows from
\[
E_{2T}(M_1, M') \land E_{2T}(M', M_2) = \bigwedge_{m_1 \in M_1} \left( \bigvee_{m \in M} E_T(m_1, m) \land \bigwedge_{m_2 \in M_2} E_T(m_1, m_2) \right) \land \bigwedge_{m_2 \in M_2} \left( \bigvee_{m \in M} E_T(m, m_2) \land \bigwedge_{m_1 \in M_1} E_T(m_2, m) \right) \leq \bigwedge_{m_1 \in M_1} \bigwedge_{m_2 \in M_2} E_T(m_1, m_2) = E_{2T}(M_1, M_2).
\]

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*
\[
E_{T2} : T2X \times T2X \to L
\]
*defined as*
\[
E_{T2}(l_1, l_2) = E_{2T}(\sigma_X(l_1), \sigma_X(l_2)),
\]
*where \(\sigma_X : T2X \to 2TX\ is the swapper, is a similarity on \(T2X\).*

**Proof.** Reflexivity and symmetry are obvious. Transitivity follows from
\[
E_{T2}(l_1, l) \land E_{T2}(l, l_2) = E_{2T}(\sigma_X(l_1), \sigma_X(l)) \land E_{2T}(\sigma_X(l), \sigma_X(l_2)) \leq E_{2T}(\sigma_X(l_1), \sigma_X(l_2)) = E_{T2}(l_1, l_2).
\]
Note that, although we have the similarity of $T^2$, it is still an open question whether the functor composition $T^2$ 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 $\theta : X \to TY$.

For powersets of terms in a generalised setting, given a monad $\Phi$ such that the composition $\Phi \circ T$ still provides a monad, variable substitutions should then be viewed as mappings $\theta : X \to \Phi TY$.

Given a term $M \in \Phi TX$ in form of a generalised powerset of terms, the result $M\theta$ of applying a variable substitution $\theta$ on $M$ is given by

$$M\theta = (\mu^\Phi_Y \circ \Phi \theta)(M)$$

i.e. $M\theta$ is kind of a flattening of a set of terms over sets of terms, where $\mu^\Phi_Y$ 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 $\Phi = 2$. 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 $u_1(c_1)$.

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 $\theta : X \to FY$, we will have

$$A\theta = (\mu^F_Y \circ F\theta)(A).$$

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

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

$$\theta' \theta'' = \mu^\Phi_Z \circ \Phi T \theta'' \circ \theta'$$

i.e. the composition in the Kleisli category $\mathbf{Set}_{\Phi T}$ for the powerset monad $\Phi \circ T$ 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 $\Phi = 2$, 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.
Table 1
Equality of generalised terms

<table>
<thead>
<tr>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$E_{2T}(M_1, M_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${t_1}$</td>
<td>${t_1, t_2, t_3}$</td>
<td>$E_{2T}(t_1, t_2) \land E_{2T}(t_1, t_3)$</td>
</tr>
<tr>
<td>${u_1(c_1)}$</td>
<td>${u_1(u_1(c_1))}$</td>
<td>$0$</td>
</tr>
<tr>
<td>${u_1(c_1)}$</td>
<td>${u_2(x)}$</td>
<td>$E_{2D}(u_1, u_2) \land E_{2T}(c_1, x)$</td>
</tr>
<tr>
<td>${u_1(c_1), u_2(c_2)}$</td>
<td>${u_2(c_1), u_2(c_2)}$</td>
<td>$E_{2D}(u_1, u_2)$</td>
</tr>
<tr>
<td>${u_1(c_1), u_2(c_2)}$</td>
<td>${u_1(c_2), u_2(c_1)}$</td>
<td>$E_{2D}(u_1, u_2) \lor E_{2D}(c_1, c_2)$</td>
</tr>
</tbody>
</table>

**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

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

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 [16].

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 $[f]_{e_i}$ according to an increasing numerable sequence of bounds $e_1, e_2, \ldots, e_n < 1$ and the following measure: A function $g$ is in the class $[f]_{e_i}$ if and only if $|f(x) - g(x)| < e_i$ and $g \notin [f]_{e_j}$ for all $j < i$.

Let us designate a canonical representative $g_{e_i}$ for each class $[g]_{e_i}$, and consider the set $\Omega = \{f, g_{e_i}\}_{1 \leq i \leq n}$. The relation $E_{2\Omega} : \Omega \times \Omega \rightarrow [0, 1]$ given by

$$E_{2\Omega}(f, f) = 1 = E_{2\Omega}(g_{e_i}, g_{e_i}),$$

$$E_{2\Omega}(g_{e_i}, g_{e_j}) = 1 - \max\{e_i, e_j\} \quad \text{if } i \neq j.$$
and

\[ E_\Omega(f, g_i) = 1 - \varepsilon_i = E_\Omega(g_i, f) \]

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 \( g_i(f(x)) \) and \( f(g_i(x)) \), can be considered as terms over the variable \( x \) and, therefore, calculated using \( E_T \) by the expression

\[ E_T(g_i(f(x)), f(g_i(x))) = 1 - \varepsilon_i. \]

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

\[ M_1 = \{ f(f(x)), f(g_i(x)) \}, \]
\[ M_2 = \{ g_i(f(x)), f(g_i(x)) \}. \]

The similarity between \( M_1 \) and \( M_2 \) is then given by

\[ E_{2T}(M_1, M_2) = 1 - \varepsilon. \]

Let us now consider the sets \( N_1 \) and \( N_2 \):

\[ N_1 = \{ f(a), g_i(x_1) \}, \]
\[ N_2 = \{ f(x_1), g_i(a) \}. \]

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

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 \( E_{2T}(N_1 \theta_1, N_2 \theta_1) = 1 - \varepsilon \) 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 participation

In order to qualify for participation 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 \( C \), 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 \( E_{2T}(S_3, R) = 0.6 \).
Table 2
Similarities between courses

<table>
<thead>
<tr>
<th>E</th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>n1</th>
<th>n2</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
<td>0.2</td>
</tr>
<tr>
<td>c2</td>
<td>0.2</td>
<td>1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>c3</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>n1</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>n2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>

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 \( E_{27}(S_x, \theta, R) = 0.7 \). Note that \( \theta'(x) = \{ c_2, c_3 \} \) is also a unifier. Then \( E_{27}(S_x, \theta', R) = 0.7 \) 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


