Fuzzy Unification and Generalization of First-Order Terms over Similar Signatures

Hassan Aït-Kaci[1] and Gabriella Pasi[2]

[1] HAK Language Technologies — hak@acm.org
[2] Universitá de Milano-Bicocca — pasi@disco.unimib.it

Abstract. Unification and generalization are operations on two terms computing respectively their greatest lower bound and least upper bound when the terms are quasi-ordered by subsumption up to variable renaming (*i.e.*, $t_1 \preceq t_2$ iff $t_1 = t_2\sigma$ for some variable substitution σ). When term signatures are such that distinct functor symbols may be related with a fuzzy equivalence (called a *similarity*), these operations can be formally extended to tolerate mismatches on functor names and/or arity or argument order. We reformulate and extend previous work with a declarative approach defining unification and generalization as sets of axioms and rules forming a complete constraint-normalization proof system. These include the Reynolds-Plotkin term-generalization procedures, Maria Sessa's "weak" unification with partially fuzzy signatures and its corresponding generalization, as well as novel extensions of such operations to fully fuzzy signatures (*i.e.*, similar functors with possibly different arities). One advantage of this approach is that it requires no modification of the conventional data structures for terms and substitutions. This and the fact that these declarative specifications are efficiently executable conditional Horn-clauses offers great practical potential for fuzzy information-handling applications.[3]

1 Subsumption Lattice

The first-order term ($\mathcal{FOT}$) was introduced as a data structure in software programming by the Prolog language.[4] Just like the S-expression for LISP, the $\mathcal{FOT}$ is Prolog's universal data structure. Using formal algebra notation, we write $\mathcal{T}_{\Sigma,\mathcal{V}}$ for the set of $\mathcal{FOT}$s on an operator signature $\Sigma \stackrel{\text{def}}{=} \bigcup_{n\geq 0} \Sigma_n$ where Σ_n is a set of operator symbols of n arguments $\Sigma_n \stackrel{\text{def}}{=} \{f \,|\, \textbf{arity}(f) = n,\, n \in \mathbb{N}\}$, and $\mathcal{V}$ is a set of variables.[5] We shall designate an element f in Σ as a *functor*, with $\textbf{arity}(f)$ denoting its number of arguments.[6] This set $\mathcal{T}_{\Sigma,\mathcal{V}}$ can then be defined inductively as:

$$\mathcal{T}_{\Sigma,\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{V} \cup \{f(t_1,\ldots,t_n) \,|\, f \in \Sigma_n,\ n \geq 0,\ t_i \in \mathcal{T}_{\Sigma,\mathcal{V}},\ 0 \leq i \leq n\}.$$

[3] This article appeared in the pre-proceedings of LOPSTR 2017 with the title "*Lattice Operations on Terms over Similar Signatures*." This version's title is technically more accurate. This version is a corrected version of the paper in the conference proceedings. All proofs and more examples can be found in a more detailed paper [3]. This work is part of a wider study [2].

[4] https://en.wikipedia.org/wiki/Prolog

[5] We shall use Prolog's convention of writing variables with capitalized symbols.

[6] When $\textbf{arity}(f) = n$, this is often denoted by writing f/n.

We write c instead of $c()$ for a constant $c \in \Sigma_0$. Also, when the set Σ of functor symbols and the set $\mathcal{V}$ of variables are implicit from the context, we simply write $\mathcal{T}$ instead of $\mathcal{T}_{\Sigma,\mathcal{V}}$. The set $\mathbf{var}(t)$ of variables occurring in a $\mathcal{FOT}$ $t \in \mathcal{T}$ is defined as:

$$\mathbf{var}(t) \stackrel{\text{def}}{=} \begin{cases} \{X\} & \text{if } t = X \in \mathcal{V} \\ \bigcup_{i=1}^{n} \mathbf{var}(t_n) & \text{if } t = f(t_1, \ldots, t_n). \end{cases}$$

The lattice-theoretic properties of $\mathcal{FOT}$s as data structures were first exposed and studied by Reynolds (in [19]) and Plotkin (in [17] and [18]). They noted that the set $\mathcal{T}$ is ordered by term subsumption (denoted as '$\preceq$'); *viz.*, $t \preceq t'$ (and we say: "t' *subsumes* t") iff there exists a variable substitution $\sigma : \mathbf{var}(t') \to \mathcal{T}$ such that $t'\sigma = t$. Two $\mathcal{FOT}$s t and t' are considered "*equal up to variable renaming*" (denoted as $t \simeq t'$) whenever both $t \preceq t'$ and $t' \preceq t$. Then, the set of first-order terms modulo variable renaming, when lifted with a bottom element $\perp$ standing for "*no term*" (*i.e.*, the set $\mathcal{T}_{/\simeq} \cup \{\perp\}$) has a lattice structure for subsumption. It has a top element $\top = \mathcal{V}$ (indeed, since any variable in $\mathcal{V}$ can be substituted for any term, $\mathcal{V}$ is therefore the class of any variable modulo renaming). Unification corresponds to its greatest lower bound (**glb**) operation. The dual operation, generalization of two terms, yields a term that is their least upper bound (**lub**) for subsumption. This can be summarized as the lattice diagram shown in Fig. 1. In this diagram, given a pair of terms $\langle \mathbf{t}_1, \mathbf{t}_2 \rangle$, the pair of substitutions $\langle \sigma_1, \sigma_2 \rangle$ are their respective most general generalizers, and the substitution σ is the pair's most general unifier (**mgu**). We formalize next these lattice operations on $\mathcal{FOT}$s as declarative constraint normalization rules.

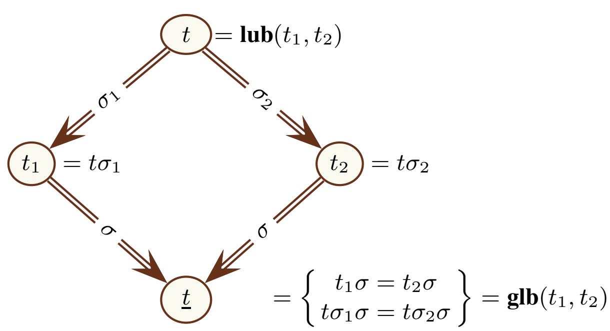

Fig. 1. Subsumption lattice operations

1.1 Unification rules

In Fig. 2, we give the set of equation normalization rules that we shall call Herbrand-Martelli-Montanari ([10] and [16]). Each rule is *provably correct* in that it is a solution-preserving transformation of a set of equations. We can use these rules to unify two $\mathcal{FOT}$s t_1 and t_2, starting with the singleton set of equations $E \stackrel{\text{def}}{=} \{ t_1 \doteq t_2 \}$.[7] Then, we transform this set of equations using any applicable rule in any order until none applies. This always terminates into a finite set of equations E'. If all the equations in

[7] In such equations, we use the notation $t_1 \doteq t_2$ not to confuse it with the equality symbol "$=$" (at the meta-level).

TERM DECOMPOSITION

$$\frac{E \cup \{f(s_1,\ldots,s_n) \doteq f(t_1,\ldots,t_n)\}}{E \cup \{s_1 \doteq t_1,\ldots,s_n \doteq t_n\}}\ [n \geq 0]$$

VARIABLE ERASURE

$$\frac{E \cup \{X \doteq X\}}{E}$$

VARIABLE ELIMINATION

$$\frac{E \cup \{X \doteq t\}}{E[X \leftarrow t] \cup \{X \doteq t\}}\ \left[\begin{matrix} X \notin \mathbf{var}(t) \\ X \text{ occurs in } E \end{matrix}\right]$$

EQUATION ORIENTATION

$$\frac{E \cup \{t \doteq X\}}{E \cup \{X \doteq t\}}\ [t \notin \mathcal{V}]$$

Fig. 2. Herbrand-Martelli-Montanari unification rules

E' are of the form $X \doteq t$ with X occurring nowhere else in E', then this is a most general unifying substitution (up to consistent variable renaming) $\sigma \stackrel{\text{def}}{=} \{\ t/X \mid X \doteq t \in E'\ \}$ solving the original equation (*i.e.*, $t_1\sigma = t_2\sigma$); otherwise, there is no solution.

In the rules of Figure 2, Rule **VARIABLE ELIMINATION** has the side condition $X \notin \mathbf{var}(t)$ to prevent circular terms, whose presence indicates no $\mathcal{FOT}$ solutions. This condition could be omitted if wished, thus extending the set of $\mathcal{FOT}$s and solutions of equations to rational $\mathcal{FOT}$s—also called "*infinite trees*" (see, *e.g.*, [9], [13], [7]).

1.2 Generalization rules

In 1970, John Reynolds and Gordon Plotkin published each an article, in the same volume ([19] and [18]), giving two identical algorithms (up to notation) for the generalization of two $\mathcal{FOT}$s. Each describes a procedural method computing the most specific $\mathcal{FOT}$ subsuming two given $\mathcal{FOT}$s in finitely many steps by comparing them simultaneously, and generating a pair of generalizing substitutions from a fresh variable wherever they disagree being scanned from left to right, each time replacing the disagreeing terms by the new variable everywhere they both occur in each term.

Next, we present a set of declarative normalization rules for generalization which are equivalent to these procedural algorithms. As far as we know, this is the first such presentation of a declarative set of rules for generalization besides its more general form as order-sorted feature term generalization in [5]. The advantage of specifying this operation in this manner rather than procedurally as done originally by Reynolds and Plotkin is that each rule or axiom relates a pair of prior substitutions to a pair of posterior substitutions based only on local syntactic-pattern properties of the terms to generalize, and this without resorting to side-effects on global structures. In this way, the terms and substitutions involved are derived as solutions of logical syntactic constraints. In addition, correctness of the so-specified operation is made much easier to establish since we only need to prove each rule's correctness independently of that of the others. Finally, the rules also provide an effective means for the derivation of an operational semantics for the so-specified operation by constraint solving, without need for control specification as any applicable rule may be invoked in any order.[8]

[8] Such as the Herbrand-Martelli-Montanari unification rules w.r.t. to Robinson's procedural unification algorithm.

Definition 1 (Generalization Judgment). *A* generalization judgement *is an expression of the form:*

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}t_1\\ t_2\end{pmatrix} t \begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} \tag{1}$$

where $\sigma_i : \textbf{\textit{var}}(t_i) \rightarrow \mathcal{T}$ and $\theta_i : \textbf{\textit{var}}(t) \rightarrow \mathcal{T}$ ($i = 1, 2$) are substitutions, and $t \in \mathcal{T}$ and $t_i \in \mathcal{T}$ ($i = 1, 2$) are $\mathcal{FOT}$s.

Definition 2 (Generalization Judgment Validity). *A generalization judgement such as* (1) *is said to be* valid *whenever $t_i\sigma_i = t\theta_i$, for $i = 1, 2$.*

Contrary to other normalization rules in this document which are expressed as conditional rewrite rules whereby a prior form (the "numerator") is related to a posterior form (the "denominator"), these normalization rules are more naturally rendered as (conditional) Horn clauses of judgements. This is as convenient as rewrite rules since a Prolog-like operational semantics can then readily provide an effective interpretation. This operational semantics is efficient because it does not need backtracking as long as the complete set of conditions of a ruleset covers all but mutually exclusive syntactic patterns. Thus, a generalization rule is of the form:

$$\frac{\begin{matrix}[\phi]\\ J_1 \quad \ldots \quad J_n\end{matrix}}{J} \tag{2}$$

where ϕ is a side meta-condition, and $J, J_1, \ldots, J_n$ are judgements, and it reads, "*whenever the side condition ϕ holds, if all the n antecedent judgements J_n are valid, then the consequent judgement J is also valid*." Such a generalization rule without a specified antecedent (a "numerator") is called a "*generalization axiom*." Such an axiom is said to be valid iff its consequent (the "denominator") is valid whenever its optional side condition holds. It is equivalent to a rule where the only antecedent is the trivial generalization judgement **TRUE**.

Definition 3 (Generalization Rule Correctness). *A conditional Horn rule such as Rule (2) is* correct *iff J_k is a valid judgment for all $k = 1, \ldots, n$ implies that J is a valid judgment, whenever the side condition ϕ holds.*

Given t_1 and t_2 two $\mathcal{FOT}$s having no variable in common, in order to find the most specific term t and most general substitutions σ_i, $i = 1, 2$, such that $t\sigma_i = t_i$, $i = 1, 2$, one needs to establish the generalization judgement:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix} \vdash \begin{pmatrix}t_1\\ t_2\end{pmatrix} t \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}. \tag{3}$$

In other words, this expresses the upper half of Fig. 1 whereby $t = \textbf{lub}(t_1, t_2)$, with most general substitutions σ_1 and σ_2. We give a complete set of normalization axioms and rule for generalization for all syntactic patterns in Fig. 3. Rule "**EQUAL FUNCTORS**" uses an "*unapply*" operation ('$\uparrow$') on a pair of terms (t_1, t_2) given a pair of substitutions

EQUAL VARIABLES

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}X\\ X\end{pmatrix} X \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}$$

VARIABLE-TERM

$[t_1 \in \mathcal{V}$ or $t_2 \in \mathcal{V};\quad t_1 \neq t_2;\quad X$ is new$]$

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}t_1\\ t_2\end{pmatrix} X \begin{pmatrix}\sigma_1\{t_1/X\}\\ \sigma_2\{t_2/X\}\end{pmatrix}$$

UNEQUAL FUNCTORS

$[m \geq 0, n \geq 0;\quad m \neq n$ or $f \neq g;\quad X$ is new$]$

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}f(s_1,\ldots,s_m)\\ g(t_1,\ldots,t_n)\end{pmatrix} X \begin{pmatrix}\sigma_1\{f(s_1,\ldots,s_m)/X\}\\ \sigma_2\{g(t_1,\ldots,t_n)/X\}\end{pmatrix}$$

EQUAL FUNCTORS

$[n \geq 0]$

$$\frac{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}s_1\\ t_1\end{pmatrix}\uparrow\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} u_1 \begin{pmatrix}\sigma_1^1\\ \sigma_2^1\end{pmatrix} \quad \ldots \quad \begin{pmatrix}\sigma_1^{n-1}\\ \sigma_2^{n-1}\end{pmatrix} \vdash \begin{pmatrix}s_n\\ t_n\end{pmatrix}\uparrow\begin{pmatrix}\sigma_1^{n-1}\\ \sigma_2^{n-1}\end{pmatrix} u_n \begin{pmatrix}\sigma_1^n\\ \sigma_2^n\end{pmatrix}}{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}f(s_1,\ldots,s_n)\\ f(t_1,\ldots,t_n)\end{pmatrix} f(u_1,\ldots,u_n) \begin{pmatrix}\sigma_1^n\\ \sigma_2^n\end{pmatrix}}$$

Fig. 3. Generalization axioms and rule

(σ_1, σ_2). It may be conceived as (and in fact is) the result of simultaneously "*unapplying*" σ_i from t_i into a common variable X only if such X is bound to t_i by σ_i, for $i = 1, 2$. If there is no such a variable, it is the identity. Formally, this is defined as:

$$\begin{pmatrix}t_1\\ t_2\end{pmatrix}\uparrow\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \stackrel{\text{def}}{=} \begin{cases}\begin{pmatrix}X\\ X\end{pmatrix} \text{ if } t_i = X\sigma_i, \text{ for } i = 1, 2;\\[2ex] \begin{pmatrix}t_1\\ t_2\end{pmatrix} \text{ otherwise.}\end{cases} \tag{4}$$

Note also that Rule "**EQUAL FUNCTORS**" is defined for $n \geq 0$. For $n = 0$ (for any constant c), it becomes the following axiom:

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} \vdash \begin{pmatrix}c\\ c\end{pmatrix} c \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}. \tag{5}$$

Theorem 1. *The axioms and the rule of Fig. 3 are correct.*

In particular, with empty prior substitutions, we obtain the following corollary.

Corollary 1 **($\mathcal{FOT}$ Generalization).** *Whenever the judgement* $\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix} \vdash \begin{pmatrix}t_1\\ t_2\end{pmatrix} t \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}$ *is valid, then* $t\sigma_i = t_i$*, for* $i = 1, 2$.

2 Fuzzy Lattice Operations

2.1 Fuzzy unification

A fuzzy unification operation on $\mathcal{FOT}$s, dubbed "*weak unification*," was proposed by Maria Sessa in [20]. It normalizes equations between conventional $\mathcal{FOT}$s modulo a similarity relation $\sim$ over functor symbols. This similarity relation is then homomorphically extended to one over all $\mathcal{FOT}$s. It is: (1) the (crisp) identity relation on variables (*i.e.*, $X \sim_1 X$, for any X in $\mathcal{V}$); otherwise, (2) zero when either of the two terms is a variable (*i.e.*, $X \sim_0 t$ and $t \sim_0 X$, for any $X \neq t$ in $\mathcal{V}$); otherwise (3):

$$f(s_1, \ldots, s_n) \sim_{(\alpha \wedge \bigwedge_{i=1}^{n} \alpha_i)} g(t_1, \ldots, t_n) \ \text{ if } \ f \sim_\alpha g \text{ and } s_i \sim_{\alpha_i} t_i, \ \ i = 1, \ldots, n$$

where $\alpha \in [0, 1]$ and $\alpha_i \in [0, 1]$ $(i = 1, \ldots, n)$ denote the *unification degrees* to which each corresponding equation holds.[9]

In Fig. 4, we provide a set of declarative rewrite rules equivalent to Sessa's case-based "weak unification algorithm" [20]. To simplify the presentation of these rules while remaining faithful to Sessa's weak unification algorithm, it is assumed for now that functor symbols f/m and g/n of different arities $m \neq n$ are never similar. This is without any loss of generality since Sessa's weak unification fails on term structures of different arities.[10] Later, we will relax this and allow functors of different arities to be similar.

FUZZY TERM DECOMPOSITION

$$\frac{(E \cup \{f(s_1, \ldots, s_n) \doteq g(t_1, \ldots, t_n)\})_\alpha}{(E \cup \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\})_{\alpha \wedge \beta}} \begin{bmatrix} f \sim_\beta g \\ n \geq 0 \end{bmatrix}$$

VARIABLE ERASURE

$$\frac{(E \cup \{X \doteq X\})_\alpha}{E_\alpha}$$

VARIABLE ELIMINATION

$$\frac{(E \cup \{X \doteq t\})_\alpha}{(E[X \leftarrow t] \cup \{X \doteq t\})_\alpha} \begin{bmatrix} X \notin \textbf{var}(t) \\ X \text{ occurs in } E \end{bmatrix}$$

EQUATION ORIENTATION

$$\frac{(E \cup \{t \doteq X\})_\alpha}{(E \cup \{X \doteq t\})_\alpha} [t \notin \mathcal{V}]$$

Fig. 4. Normalization rules corresponding to Maria Sessa's "weak unification"

The rules of Fig. 4 transform E_α a finite conjunctive set E of equations among $\mathcal{FOT}$s along with an associated truth value, or "*unification degree*," $\alpha \in [0, 1]$, into $E'_{\alpha'}$ another set of equations E' with truth value $\alpha' \in [0, \alpha]$. Given to solve a fuzzy unification equation $s \doteq t$ between two $\mathcal{FOT}$s s and t, form the set $\{s \doteq t\}_1$ (*i.e.*, with unification degree 1), and apply any applicable rules in Fig. 4 until either the unification degree of the set of equations is 0 (in which case there is no solution to the original equation, not even a fuzzy one), or the final resulting set E_α is a solution with truth value α in the form of a variable substitution $\sigma \stackrel{\text{def}}{=} \{X/t \mid X \doteq t \in E\}$ such that $s\sigma \sim_\alpha t\sigma$.

From our perspective, a fuzzy unification operation ought to be able to fuzzify *full* $\mathcal{FOT}$ unification: whether (1) functor symbol mismatch, and/or (2) arity mismatch, and/or (3) in which order subterms correspond. Sessa's fuzzification of unification as weak unification misses on the last two items. This is unfortunate as this can turn out to be quite useful. In real life, there is indeed no such garantee that argument positions

[9] The $\wedge$ operation used by Sessa in this expression is $\min$; but other interpretations are possible ([8], [2]).

[10] See Case (2) of the weak unification algorithm given in [20], Page 413.

of different functors match similar information in data and knowledge bases, hence the need for alignment [15].

Still, it has several qualities:

- *It is simple*—specified as a straightforward extension of crisp unification: only one rule (Rule "**FUZZY TERM DECOMPOSITION**") may alter the fuzziness of an equation set by tolerating similar functors.
- *It is conservative*—neither $\mathcal{FOT}$s nor $\mathcal{FOT}$ substitutions *per se* need be fuzzified; so conventional crisp representations and operations can be used; if restricted to only 0 or 1 truth values, it is equivalent to crisp $\mathcal{FOT}$ unification.

We now give an extension of Sessa's weak unification which can tolerate such fuzzy similarity among functors of different arities. Given a similarity relation $\sim$ on a ranked signature $\Sigma \stackrel{\text{def}}{=} \Sigma_{n\geq 0}$, $\sim : \Sigma^2 \to [0,1]$ which, unlike M. Sessa's equal-arity condition, now allows mismatches of similar symbols with distinct arities or equal arities but different argument orders. Namely,

- it admits that $(\sim \cap\, \Sigma_m \times \Sigma_n) \neq \emptyset$ for some $m \geq 0$, $n \geq 0$, such that $m \neq n$;
- for each pair of functors $\langle f, g\rangle \in \Sigma^2$, such that $f \in \Sigma_m$ and $g \in \Sigma_n$, with $0 \leq m \leq n$, and $f \sim_\alpha g$, $(\alpha \in (0,1])$, there exists an injective (*i.e.*, one-to-one) mapping $p : \{1,\ldots,m\} \to \{1,\ldots,n\}$ associating each of the m argument positions of f to a unique position among the n arguments of g (which is denoted as $f \sim^p_\alpha g$).

Note that in the above, m and n are such that $0 \leq m \leq n$; so the one-to-one argument-position mapping goes from the lesser set to the larger set. There is no loss of generality with this assumption as this will be taken into account in the normalization rules.

Example 1. [*Similar functors with different arities*] Consider $\mathtt{person}/3$, a functor of arity 3, and $\mathtt{individual}/4$, a functor of arity 4 with:

- similarity truth value of .9; *i.e.*, $\mathtt{person}/3 \ \sim_{.9}\ \mathtt{individual}/4$; and,
- one-to-one position mapping $p : \{1,2,3\} \to \{1,2,3,4\}$:

  from $\mathtt{person}/3$ to $\mathtt{individual}/4$ with $p : \{1 \to 1, 2 \to 3, 3 \to 4\}$

  so that:

$$\mathtt{person}(\mathit{Name}, \mathit{SSN}, \mathit{Address}) \ \sim^p_{.9}\ \mathtt{individual}(\mathit{Name}, \mathit{DoB}, \mathit{SSN}, \mathit{Address})$$

writing $f \sim^p_\alpha g$ a similarity relation between a functor f and a functor g of truth value α and f-to-g argument-position mapping p; in our example, $\mathtt{person} \sim^{\{1\to1,2\to3,3\to4\}}_{.9} \mathtt{individual}$.

With this kind of specification, we can tolerate not only fuzzy mismatching of terms with distinct functors person and individual, but also up to a correspondance of argument positions from person to *individual* specified as p, all with a truth value of .9.

Starting with the Herbrand-Martelli-Montanari ruleset of Fig. 2, fuzziness is introduced by relaxing "**TERM DECOMPOSITION**" to make it also tolerate possible arity or argument-order mistmatch in two structures being unified. In other words, the given functor similarity relation $\sim$ is adjoined a position mapping from argument positions of a functor f to those of a functor g when $f \neq g$ and $f \sim_\alpha g$ with $\alpha \in (0,1]$. This is then taken into account in tolerating a fuzzy mismatch between two term structures $s = f(s_1,\ldots,s_m)$ and $t = g(t_1,\ldots,t_n)$. This may involve a mismatch between the terms' functor symbols (f and g), their arities (m and n), subterm orders, or a combination. We first reorient all such equations by flipping sides so that the left-hand side is the one wih lesser or equal arity. In this manner, assuming $f \sim^p_\beta g$ and $0 \leq \alpha, \beta \leq 1$, an equation of the form: $\left\{f(s_1,\ldots,s_m) \doteq g(t_1,\ldots,t_n)\right\}_\alpha$ for $0 \leq m \leq n$ acquires its truth value $\alpha \wedge \beta$ due to functor and arity mismatch when equated. A fully fuzzified term-decomposition rule should proceed with replacing such a fuzzy structure equation with the following conjunction of fuzzy equations between subterms at corresponding indices given by the one-to-one argument mapping $p : \{1,\ldots,m\} \to \{1,\ldots,n\}$: $\left\{s_1 \doteq t_{p(1)}, \quad \ldots, \quad s_m \doteq t_{p(m)}, \quad \ldots \quad \right\}_{\alpha\wedge\beta}$. Note that all the subterms in the right-hand side term that are arguments at indices which are not p-images are ignored as they have no counterparts in the left-hand side. These terms are simply dropped as part of the fuzzy approximative unification. This generic rule is shown in Fig. 5 along with another rule needed to make it fully effective: a rule reorienting a term equation into one with a lesser-arity term on the left.

GENERIC WEAK TERM DECOMPOSITION

$$\left[0 \leq m \leq n;\ f \sim^p_\beta g\right]$$

$$\frac{\left(E \cup \{f(s_1,\ldots,s_m) \doteq g(t_1,\ldots,t_n)\}\right)_\alpha}{\left(E \cup \{s_1 \doteq t_{p(1)},\ldots,s_m \doteq t_{p(m)}\}\right)_{\alpha\wedge\beta}}$$

FUZZY EQUATION REORIENTATION

$$[0 \leq n < m]$$

$$\frac{\left(E \cup \{f(s_1,\ldots,s_m) \doteq g(t_1,\ldots,t_n)\}\right)_\alpha}{\left(E \cup \{g(t_1,\ldots,t_n) \doteq f(s_1,\ldots,s_m)\}\right)_\alpha}$$

Fig. 5. Generic fuzzification of $\mathcal{FOT}$ unification's decomposition rule

Theorem 2. *The fuzzy unification rules of Fig. 4 where Rule "*FUZZY TERM DECOMPOSITION*" is replaced by the rules of Fig. 5 are correct.*

In other words, applying this modified ruleset to $E_1 \stackrel{\text{def}}{=} \{s \doteq t\}_1$, an equation set of truth value 1 (in any order as long as a rule applies and its truth value is not zero) always terminates. And when the final equation set is a substitution σ, it is a fuzzy solution with truth value α such that $s\sigma \sim_\alpha t\sigma$.

Example 2. [$\mathcal{FOT}$ *fuzzy unification with similar functors of different arities*] Let us take a functor signature such that: $\{a,b,c,d\} \subseteq \Sigma_0$, $\{f,g,\ell\} \subseteq \Sigma_2$, $\{h\} \subseteq \Sigma_3$; and let us further assume that the only non-zero similarities argument mappings among these functors are:

- $a \sim_{.7} b$,
- $c \sim_{.6} d$,
- $f \sim^{\{1\to2,2\to1\}}_{.9} g$ and $g \sim^{\{1\to2,2\to1\}}_{.9} f$,

– $\ell \sim_{.8}^{\{1\to 2, 2\to 3\}} h$.

Let us consider the fuzzy equation set $\{\mathbf{t}_1 \doteq \mathbf{t}_2\}_1$:

$$\{h(X, g(Y, b), f(Y, c)) \doteq \ell(f(a, Z), g(d, c))\}_1 \quad (6)$$

and let us apply the rules of Figure 4 with rule **Weak Term Decomposition** is replaced by the rules of Figure 5:

- apply Rule **Fuzzy Equation Reorientation** with $\alpha = 1$ since $\mathbf{arity}(\ell) < \mathbf{arity}(h)$:

  $$\{\ell(f(a, Z), g(d, c)) \doteq h(X, g(Y, b), f(Y, c))\}_1 \,;$$

- apply Rule **Generic Weak Term Decomposition** to:

  $$\ell(f(a, Z), g(d, c)) \doteq h(X, g(Y, b), f(Y, c))$$

  with $\alpha = 1$ and $\beta = .8$ since $\ell \sim_{.8}^{\{1\to 2, 2\to 3\}} h$, to obtain:

  $$\{f(a, Z) \doteq g(Y, b), g(d, c) \doteq f(Y, c)\}_{.8} \,;$$

- apply Rule **Generic Weak Term Decomposition** to $f(a, Z) \doteq g(Y, b)$ with $\alpha = .8$ and $\beta = .9$ since $f \sim_{.9}^{\{1\to 2, 2\to 1\}} g$, to obtain:

  $$\{a \doteq b, Z \doteq Y, g(d, c) \doteq f(Y, c)\}_{.8} \,;$$

- apply Rule **Generic Weak Term Decomposition** to $a \doteq b$ with $\alpha = .8$ and $\beta = .7$ since $a \sim_{.7} b$, to obtain:

  $$\{Z \doteq Y, g(d, c) \doteq f(Y, c)\}_{.7} \,;$$

- apply Rule **Generic Weak Term Decomposition** to $g(d, c) \doteq f(Y, c)$ with $\alpha = .7$ and $\beta = .9$ since $f \sim_{.9}^{\{1\to 2, 2\to 1\}} g$, to obtain:

  $$\{Z \doteq Y, d \doteq c, c \doteq Y\}_{.7} \,;$$

- apply Rule **Generic Weak Term Decomposition** to $d \doteq c$ with $\alpha = .7$ and $\beta = .6$ since $d \sim_{.6} c$, to obtain:

  $$\{Z \doteq Y, c \doteq Y\}_{.6} \,;$$

- apply Rule **Equation Orientation** to $c \doteq Y$ with $\alpha = .6$, to obtain:

  $$\{Z \doteq Y, Y \doteq c\}_{.6} \,.$$

- apply Rule **Variable Elimination** to $Y \doteq c$ with $\alpha = .6$, to obtain:

  $$\{Z \doteq c, Y \doteq c\}_{.6} \,.$$

This last equation set is in normal form with truth value .6 and defines the substitution $\sigma = \{\, c/Z, c/Y \,\}$ so that:

$$\mathbf{t}_1\sigma = h(X, g(Y, b), f(Y, c))\,\{\, c/Z, c/Y \,\} \;\sim_{.6}\; \mathbf{t}_2\sigma = \ell(f(a, Z), g(d, c))\,\{\, c/Z, c/Y \,\}\,, \quad (7)$$

that is:

$$\mathbf{t}_1\sigma = h(X, g(c, b), f(c, c)) \;\sim_{.6}\; \mathbf{t}_2\sigma = \ell(f(a, c), g(d, c)). \quad (8)$$

Example 3. **[*The same fuzzy unification with more expressive symbols*]** Let us give more expressive names to functors of Example 2 in the context of, say, a gift-shop Prolog database which describes various configurations for multi-item gift boxes or bags containing such items as flowers, sweets, *etc.*, which can be already joined as pairs or not joined as loose couples.

- $a \stackrel{\text{def}}{=} \texttt{violet}$,
- $b \stackrel{\text{def}}{=} \texttt{lilac}$,
- $c \stackrel{\text{def}}{=} \texttt{chocolate}$,
- $d \stackrel{\text{def}}{=} \texttt{candy}$,
- $f \stackrel{\text{def}}{=} \texttt{pair}$,
- $g \stackrel{\text{def}}{=} \texttt{couple}$,
- $\ell \stackrel{\text{def}}{=} \texttt{small-gift-bag}$,
- $h \stackrel{\text{def}}{=} \texttt{small-gift-box}$,

with the following similarity degrees and argument mappings,:

- $\texttt{violet} \sim_{.7} \texttt{lilac}$,
- $\texttt{chocolate} \sim_{.6} \texttt{candy}$,
- $\texttt{pair} \sim_{.9} \texttt{couple}$,
- $\texttt{pair} \sim_{.9}^{\{1\to2,2\to1\}} \texttt{couple}$ and $\texttt{couple} \sim_{.9}^{\{1\to2,2\to1\}} \texttt{pair}$,
- $\texttt{small-gift-bag} \sim_{.8}^{\{1\to2,2\to3\}} \texttt{small-gift-box}$.

With these functors Equation (6) now reads:

$$
\begin{array}{ll}
(\mathbf{t}_1) & \begin{array}{lll} \texttt{small-gift-box} & (& X \\ & , & \texttt{couple}(Y, \texttt{lilac}) \\ & , & \texttt{pair}(Y, \texttt{chocolate}) \\ &) & \end{array} \\
& \doteq \\
(\mathbf{t}_2) & \begin{array}{lll} \texttt{small-gift-bag} & (& \texttt{pair}(\texttt{violet}, Z) \\ & , & \texttt{couple}(\texttt{candy}, \texttt{chocolate}) \\ &) & \end{array}
\end{array}
$$

With the new functor symbols, the substitution $\sigma = \{\, \texttt{chocolate}/Z, \texttt{chocolate}/Y \,\}$ obtained after normalization yields the fuzzy solution:

$$
\begin{array}{ll}
(\mathbf{t}_1\sigma) & \begin{array}{lll} \texttt{small-gift-box} & (& X \\ & , & \texttt{couple}(\texttt{chocolate}, \texttt{lilac}) \\ & , & \texttt{pair}(\texttt{chocolate}, \texttt{chocolate}) \\ &) & \end{array} \\
& \sim_{.6} \\
(\mathbf{t}_2\sigma) & \begin{array}{lll} \texttt{small-gift-bag} & (& \texttt{pair}(\texttt{violet}, \texttt{chocolate}) \\ & , & \texttt{couple}(\texttt{candy}, \texttt{chocolate}) \\ &) & \end{array}
\end{array}
$$

with truth value .6 capturing the unification degree to which σ solves the original equation.

Rule GENERIC WEAK TERM DECOMPOSITION is a very general rule for normalizing fuzzy equations over $\mathcal{FOT}$ structures. It has the following convenient properties:

1. it accounts for fuzzy mismatches of similar functors of possibly different arity or order of arguments;

2. when restricted to tolerating only similar equal-arity functors with matching argument positions, it reduces to Sessa's weak unification's **WEAK TERM DECOMPOSITION** rule;
3. when truth values are further restricted to be in $\{0, 1\}$, it reduces to Herbrand-Martelli-Montanari's **TERM DECOMPOSITION** rule;
4. it requires no alteration of the standard notions of $\mathcal{FOT}$s and $\mathcal{FOT}$ substitutions: similarity among $\mathcal{FOT}$s is derived from that of signature symbols;
5. finally, and most importantly, it keeps fuzzy unification in the same complexity class as crisp unification: that of Union-Find ([14], [21]).[11]

As a result, it is more general than all other extant approaches we know which propose a fuzzy $\mathcal{FOT}$ unification operation. The same will be established for the fuzzification of the dual operation: first a limited "*functor-weak*" $\mathcal{FOT}$ generalization corresponding to the dual operation of Sessa's "weak" unification, then to a more expressive "*functor/arity-weak*" $\mathcal{FOT}$ generalization corresponding to our extension of Sessa's unification to functor/arity weak unification.

2.2 Fuzzy generalization

Let t_1 and t_2 be two $\mathcal{FOT}$s in $\mathcal{T}$ to generalize. We shall use the following notation for a fuzzy generalization judgement:

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha \vdash \begin{pmatrix}t_1\\ t_2\end{pmatrix} t \begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}_\beta \tag{9}$$

given:

- $\sigma_i : \mathbf{var}(t_i) \to \mathcal{T}$ $(i = 1, 2)$: two prior substitutions with prior truth value α,
- t_i $(i = 1, 2)$: two prior $\mathcal{FOT}$s,
- t: a posterior $\mathcal{FOT}$,
- $\theta_i : \mathbf{var}(t) \to \mathcal{T}$ $(i = 1, 2)$: two posterior substitutions with truth value β.

Definition 4 (Fuzzy Generalization Judgment Validity). *A fuzzy generalization judgement such as* (9) *is* valid *whenever* $0 \leq \beta \leq \alpha \leq 1$ *and* $t_i\sigma_i \sim_\beta t\theta_i$ *for* $i = 1, 2$.

Definition 5 (Fuzzy Generalization Rule Correctness). *A fuzzy generalization rule is correct iff, whenever the side condition holds, if all the fuzzy generalization judgments making up its antecedent are valid, then necessarily the generalization judgment in its consequent is valid.*

In Fig. 6, we give a fuzzy version of the generalization rules of Fig. 3. As was the case in Sessa's weak unification, we assume as well (for now) that we are only given a similarity relation $\sim \in \Sigma \times \Sigma \to [0, 1]$ on the signature $\Sigma = \cup_{n \geq 0}\Sigma_n$ such that for all $m \geq 0$ and $n \geq 0$, $m \neq n$ implies $\sim \cap \Sigma_m \times \Sigma_n = \emptyset$ (*i.e.*, if functors f and g have different arities, then $f \not\sim g$).

Rule **SIMILAR FUNCTORS** uses a "*fuzzy unapply*" operation ('$\uparrow_\alpha$') on a pair of terms (t_1, t_2) given a pair of substitutions (σ_1, σ_2) and a similarity degree α. It is the result of

[11] Quasi-linear; *i.e.*, linear with a log . . . log coefficient [1].

FUZZY EQUAL VARIABLES

$$\binom{\sigma_1}{\sigma_2}_\alpha \vdash \binom{X}{X} X \binom{\sigma_1}{\sigma_2}_\alpha$$

FUZZY VARIABLE-TERM

$[t_1 \in \mathcal{V}$ or $t_2 \in \mathcal{V};\ \ t_1 \neq t_2;\ \ X$ is new$]$

$$\binom{\sigma_1}{\sigma_2}_\alpha \vdash \binom{t_1}{t_2} X \binom{\sigma_1\{t_1/X\}}{\sigma_2\{t_2/X\}}_\alpha$$

DISSIMILAR FUNCTORS

$[f \not\sim g;\ m \geq 0, n \geq 0;\ X$ is new$]$

$$\binom{\sigma_1}{\sigma_2}_\alpha \vdash \binom{f(s_1,\ldots,s_m)}{g(t_1,\ldots,t_n)} X \binom{\sigma_1\{f(s_1,\ldots,s_m)/X\}}{\sigma_2\{g(t_1,\ldots,t_n)/X\}}_\alpha$$

SIMILAR FUNCTORS

$\left[f \sim_\beta g;\ \ n \geq 0;\ \ \alpha_0 \stackrel{\text{def}}{=} \alpha \wedge \beta\right]$

$$\frac{\binom{\sigma_1}{\sigma_2}_{\alpha_0} \vdash \binom{s_1}{t_1} \uparrow_{\alpha_0} \binom{\sigma_1}{\sigma_2} u_1 \binom{\sigma_1^1}{\sigma_2^1}_{\alpha_1} \ \cdots\ \binom{\sigma_1^{n-1}}{\sigma_2^{n-1}}_{\alpha_{n-1}} \vdash \binom{s_n}{t_n} \uparrow_{\alpha_{n-1}} \binom{\sigma_1^{n-1}}{\sigma_2^{n-1}} u_n \binom{\sigma_1^n}{\sigma_2^n}_{\alpha_n}}{\binom{\sigma_1}{\sigma_2}_\alpha \vdash \binom{f(s_1,\ldots,s_n)}{g(t_1,\ldots,t_n)} f(u_1,\ldots,u_n) \binom{\sigma_1^n}{\sigma_2^n}_{\alpha_n}}$$

Fig. 6. Functor-weak generalization axioms and rule

"*unapplying*" σ_i from t_i, for $i = 1, 2$, into a common variable X, if any such exists such that the terms $X\sigma_i$ are respectively similar to t_i with similarity degrees α_i. It returns a fuzzy pair of terms and a similarity degree in $(0, \alpha]$ defined as:

$$\binom{t_1}{t_2} \uparrow_\alpha \binom{\sigma_1}{\sigma_2} \stackrel{\text{def}}{=} \begin{cases} \binom{X}{X}_{\alpha\wedge\alpha_1\wedge\alpha_2} & \text{if } \exists\, X \in \mathcal{V},\ t_i \sim_{\alpha_i} X\sigma_i \\ & \text{for some } \alpha_i \in (0,1]\ i = 1, 2; \\ \binom{t_1}{t_2}_\alpha & \text{otherwise.} \end{cases} \tag{10}$$

Importantly, note that fuzzy unapplication defined by Equation (10) returns a pair of terms and a (possibly lesser) approximation degree, unlike crisp unapplication defined by Equation (4) which returns only a pair of terms. Because of this, when we write a fuzzy judgment, as we do in Rule **SIMILAR FUNCTORS**, such as:

$$\binom{\sigma}{\sigma'}_\alpha \vdash \binom{t}{t'} \uparrow_\alpha \binom{\sigma}{\sigma'} u \binom{\theta}{\theta'}_\beta \tag{11}$$

this is shorthand to indicate that the posterior similarity degree β is *at most* the one returned by the fuzzy unapplication $\binom{t}{t'} \uparrow_\alpha \binom{\sigma}{\sigma'}$. Formally, the notation of the fuzzy judgment (11) is equivalent to:

$$\binom{t}{t'} \uparrow_\alpha \binom{\sigma}{\sigma'} = \binom{s}{s'}_{\beta'} \quad \text{and} \quad \binom{\sigma}{\sigma'}_{\beta'} \vdash \binom{s}{s'} u \binom{\theta}{\theta'}_\beta \tag{12}$$

for some β' such that $\beta \leq \beta' \leq \alpha$. This is because a fuzzy unapplication invoked while proving the validity of a fuzzy judgment may require, by Expression (10), lowering the *prior* approximation degree of the judgment.

Note also that Rule "**Similar Functors**" is defined for $n \geq 0$. For $n = 0$, it becomes the following fuzzy judgment:

$$\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha} \vdash \begin{pmatrix}c\\ c\end{pmatrix} c \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha} \tag{13}$$

which can be verified to be an axiom since it is valid at any approximation degree α in $[0, 1]$, for any constant c in Σ_0, and any substitutions σ_1 and σ_2 in $\mathbf{Subst}_{\mathcal{T}}$, thanks to the reflexivity of the similarity $\sim_\alpha$ on $\mathcal{T}$.

Theorem 3. *The fuzzy generalization rules of Fig. 6 are correct.*

In Fig. 7, we give a fuzzy version of the generalization rules taking into account mismatches not only in functors, but also in arities; *i.e.*, number and/or order of arguments. Unlike Sessa's unification, we now assume that we are not only given a similarity relation $\sim \in \Sigma \times \Sigma \to [0, 1]$ on the signature $\Sigma = \cup_{n \geq 0} \Sigma_n$, but also that functors of different arities may be similar with some non-zero truth value as specified by an one-to-one argument-position mapping for each pair of so-similar functors associating to each argument position of the functor of least arity a distinct argument position of the functor of larger arity. The only rule among those of Figure 6 that differs is the last one (**Similar Functors**) which is now a pair of rules called **Functor/Arity Similarity Left** and **Functor/Arity Similarity Right** to account for similar functors's argument positions depending which side has less arguments. If the arities are the same, the two rules are equivalent.

Functor/Arity Similarity Left

$$\left[f \sim^p_\beta g;\;\; 0 \leq m \leq n;\;\; \alpha_0 \overset{\text{def}}{=} \alpha \wedge \beta\right]$$

$$\frac{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha_0} \vdash \begin{pmatrix}s_1\\ t_{p(1)}\end{pmatrix}\uparrow_{\alpha_0}\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} u_1 \begin{pmatrix}\sigma_1^1\\ \sigma_2^1\end{pmatrix}_{\alpha_1} \cdots \begin{pmatrix}\sigma_1^{m-1}\\ \sigma_2^{m-1}\end{pmatrix}_{\alpha_{m-1}} \vdash \begin{pmatrix}s_m\\ t_{p(m)}\end{pmatrix}\uparrow_{\alpha_{m-1}}\begin{pmatrix}\sigma_1^{m-1}\\ \sigma_2^{m-1}\end{pmatrix} u_m \begin{pmatrix}\sigma_1^m\\ \sigma_2^m\end{pmatrix}_{\alpha_m}}{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha} \vdash \begin{pmatrix}f(s_1,\ldots,s_m)\\ g(t_1,\ldots,t_n)\end{pmatrix} f(u_1,\ldots,u_m) \begin{pmatrix}\sigma_1^m\\ \sigma_2^m\end{pmatrix}_{\alpha_m}}$$

Functor/Arity Similarity Right

$$\left[g \sim^p_\beta f;\;\; 0 \leq n \leq m;\;\; \alpha_0 \overset{\text{def}}{=} \alpha \wedge \beta\right]$$

$$\frac{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha_0} \vdash \begin{pmatrix}s_{p(1)}\\ t_1\end{pmatrix}\uparrow_{\alpha_0}\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix} u_1 \begin{pmatrix}\sigma_1^1\\ \sigma_2^1\end{pmatrix}_{\alpha_1} \cdots \begin{pmatrix}\sigma_1^{n-1}\\ \sigma_2^{n-1}\end{pmatrix}_{\alpha_{n-1}} \vdash \begin{pmatrix}s_{p(n)}\\ t_n\end{pmatrix}\uparrow_{\alpha_{n-1}}\begin{pmatrix}\sigma_1^{n-1}\\ \sigma_2^{n-1}\end{pmatrix} u_n \begin{pmatrix}\sigma_1^n\\ \sigma_2^n\end{pmatrix}_{\alpha_n}}{\begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_{\alpha} \vdash \begin{pmatrix}f(s_1,\ldots,s_m)\\ g(t_1,\ldots,t_n)\end{pmatrix} g(u_1,\ldots,u_n) \begin{pmatrix}\sigma_1^n\\ \sigma_2^n\end{pmatrix}_{\alpha_n}}$$

Fig. 7. Functor/arity-weak generalization axioms and rule

Theorem 4. *The fuzzy generalization rules of Fig. 6 where Rule "**Similar Functors**" is replaced with the rules in Fig. 7 are correct.*

Example 4. **[$\mathcal{FOT}$ *fuzzy generalization*]** Consider the signature Σ containing $\Sigma_0 = \{a, b, c, d\}$, $\Sigma_2 = \{f, g, l\}$, and $\Sigma_3 = \{h\}$, and the closure $\sim$ of the similar pairs $a \sim_{.7}$, $c \sim_{.6} d$, $f \sim_{.8} g$, and $l \sim_{.9} h$. Let us take all argument-position mappings as the default (identity on least-arity set). Let us apply the fuzzy generalization axioms of Figure 6 and the rule of Figure 7 to $\mathbf{t}_1 \stackrel{\text{def}}{=} h(g(b, Y), f(Y, c), V)$, and $\mathbf{t}_2 \stackrel{\text{def}}{=} l(f(a, Z), g(c, d))$; that is, let us find term t, substitutions $\sigma_i \in \textbf{SUBST}_{\mathcal{T}}$ $(i = 1, 2)$, and similarity degree α in $[0, 1]$, such that $t\sigma_1 \sim_\alpha h(g(b, Y), f(Y, c), V)$ and $t\sigma_2 \sim_\alpha l(f(a, Z), g(c, d))$. This is expressed as the following fuzzy judgment:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_1 \vdash \begin{pmatrix}h(g(b, Y), f(Y, c), V)\\ l(f(a, Z), g(c, d))\end{pmatrix} t \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha .$$

By Rule **FUNCTOR/ARITY SIMILARITY RIGHT**, we can infer that $t = l(u_1, u_2)$:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_1 \vdash \begin{pmatrix}h(g(b, Y), f(Y, c), V)\\ l(f(a, Z), g(c, d))\end{pmatrix} l(u_1, u_2) \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha$$

which, when replaced by the rule's antecedents, since $h \sim_{.9} l$ and $1 \wedge .9 = .9$, becomes the sequence:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.9} \vdash \begin{pmatrix}g(b, Y)\\ f(a, Z)\end{pmatrix} \uparrow_{.9} \begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix} u_1 \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'}, \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'} \vdash \begin{pmatrix}f(Y, c)\\ g(c, d)\end{pmatrix} \uparrow_{\alpha'} \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha .$$

By evaluating the fuzzy unapplication in its first judgment, this sequence becomes:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.9} \vdash \begin{pmatrix}g(b, Y)\\ f(a, Z)\end{pmatrix} u_1 \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'}, \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'} \vdash \begin{pmatrix}f(Y, c)\\ g(c, d)\end{pmatrix} \uparrow_{\alpha'} \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha .$$

By Rule **FUNCTOR/ARITY SIMILARITY LEFT**,[12] it comes that $u_1 = g(u_3, u_4)$ and, since $g \sim_{.8} f$ and $.9 \wedge .8 = .8$, the sequence becomes:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.8} \vdash \begin{pmatrix}b\\ a\end{pmatrix} \uparrow_{.8} \begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix} u_3 \begin{pmatrix}\sigma_1''\\ \sigma_2''\end{pmatrix}_{\alpha''}, \begin{pmatrix}\sigma_1''\\ \sigma_2''\end{pmatrix}_{\alpha''} \vdash \begin{pmatrix}Y\\ Z\end{pmatrix} \uparrow_{\alpha''} \begin{pmatrix}\sigma_1''\\ \sigma_2''\end{pmatrix} u_4 \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'},$$

$$\begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'} \vdash \begin{pmatrix}f(Y, c)\\ g(c, d)\end{pmatrix} \uparrow_{\alpha'} \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha .$$

By evaluating the fuzzy unapplication in the first judgment, and using Rule **FUNCTOR/ARITY SIMILARITY LEFT** in the 0-arity case as Axiom (13), since $b \sim_{.7} a$ and $.8 \wedge .7 = .7$, we have $u_3 = b$, and the sequence becomes:

$$\begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.7} \vdash \begin{pmatrix}b\\ a\end{pmatrix} b \begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.7}, \begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\ Z\end{pmatrix} \uparrow_{.7} \begin{pmatrix}\emptyset\\ \emptyset\end{pmatrix} u_4 \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'},$$

$$\begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix}_{\alpha'} \vdash \begin{pmatrix}f(Y, c)\\ g(c, d)\end{pmatrix} \uparrow_{\alpha'} \begin{pmatrix}\sigma_1'\\ \sigma_2'\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\ \sigma_2\end{pmatrix}_\alpha .$$

[12] Since f and g have equal arities, we could also use Rule **FUNCTOR/ARITY SIMILARITY RIGHT**. This would end in an equivalent final result, modulo functor similarities at the final approximation degree. In the remainder of this example, we shall omit making this remark, and choose the left rule over the right for equal-arity functors.

The validity of the first fuzzy judgment is thereby established. We proceed with the remaining sequence of fuzzy judgments evaluating the fuzzy unapplication in the first of its judgments, which sets $\alpha' = .7$:

$$\begin{pmatrix}\emptyset\\\emptyset\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\Z\end{pmatrix} u_4 \begin{pmatrix}\sigma_1'\\\sigma_2'\end{pmatrix}_{.7},\ \begin{pmatrix}\sigma_1'\\\sigma_2'\end{pmatrix}_{.7} \vdash \begin{pmatrix}f(Y,c)\\g(c,d)\end{pmatrix} \Uparrow_{.7} \begin{pmatrix}\sigma_1'\\\sigma_2'\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha}.$$

By Axiom **Fuzzy Variable-Term**, we infer from this that $u_4 = X_1$, a new variable, and the judgments become:

$$\begin{pmatrix}\emptyset\\\emptyset\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\Z\end{pmatrix} X_1 \begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7},$$

$$\begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}f(Y,c)\\g(c,d)\end{pmatrix} \Uparrow_{.7} \begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha}.$$

The validity of the first fuzzy judgment of the above sequence is thereby established. We proceed with the remainder evaluating the fuzzy unapplication in the first of its judgments, which returns the same pair of terms with the similarity degree kept at .7:

$$\begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}f(Y,c)\\g(c,d)\end{pmatrix} u_2 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha}.$$

and by Rule **Functor/Arity Similarity Left** with $u_2 = f(u_5, u_6)$, this becomes:

$$\begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\c\end{pmatrix} \Uparrow_{.7} \begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix} u_5 \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix}_{\beta},\ \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix}_{\beta} \vdash \begin{pmatrix}c\\d\end{pmatrix} \Uparrow_{\beta} \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix} u_6 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha}.$$

Evaluating the fuzzy unapplication gives $\beta = .7$:

$$\begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\c\end{pmatrix} u_5 \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix}_{.7},\ \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix}_{.7} \vdash \begin{pmatrix}c\\d\end{pmatrix} \Uparrow_{.7} \begin{pmatrix}\theta_1\\\theta_2\end{pmatrix} u_6 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha}.$$

and by Axiom **Fuzzy Variable-Term**, we infer from this that $u_5 = X_2$, a new variable, which yields:

$$\begin{pmatrix}\{\ Y/X_1\ \}\\\{\ Z/X_1\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}Y\\c\end{pmatrix} X_2 \begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix}_{.7},$$

$$\begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix}_{.7} \vdash \begin{pmatrix}c\\d\end{pmatrix} \Uparrow_{.7} \begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix} u_6 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha},$$

and establishes the penultimate judgment. The last remaining judgment, after evaluating its fuzzy unapplication, since $c \sim_{.6} d$ and $.7 \wedge .6 = .6$, is:

$$\begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix}_{.6} \vdash \begin{pmatrix}c\\d\end{pmatrix} u_6 \begin{pmatrix}\sigma_1\\\sigma_2\end{pmatrix}_{\alpha},$$

for which Axiom **Fuzzy Variable-Term** allows us to infer that $u_6 = c$ and $\alpha = .6$:

$$\begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix}_{.6} \vdash \begin{pmatrix}c\\d\end{pmatrix} c \begin{pmatrix}\{\ Y/X_1, Y/X_2\ \}\\\{\ Z/X_1, c/X_2\ \}\end{pmatrix}_{.6}.$$

This validates the last judgment and completes the fuzzy generalization whereby $t = l(g(b, X_1), f(X_2, c))$ is the least fuzzy generalizer of $\mathbf{t}_1 = h(g(b,Y),\ f(Y,c), V)$ and $\mathbf{t}_2 = l(f(a,Z), g(c,d))$ at approximation degree .6, with:

- $\sigma_1 = \{\ Y/X_1, Y/X_2\ \}$ so that $t\sigma_1 = l(g(b,Y), f(Y,c)) \sim_{.6} \mathbf{t}_1$; and,
- $\sigma_2 = \{\ Z/X_1, c/X_2\ \}$ so that $t\sigma_2 = l(g(b,Z), f(c,c)) \sim_{.6} \mathbf{t}_2$.

3 Conclusion

We have summarized the principal results regarding the derivation of fuzzy lattice operations for the data structure known as first-order term. This is achieved by means of syntax-driven constraint normalization rules for both unification and generalization. These operations are then extended to enable arbitrary mismatch between similar terms whether functor-based, arity-based (number and order), or combinations. The resulting lattice operations are in the same class of complexity as their crisp versions, of which they are conservative extensions—namely that of Union/Find. All these details, along with proofs and examples, are to be found in [3].

As for future work, there are several avenues to explore. The most immediate concerns implementation of such operations in the form of public libraries to complement extant tools for first-order terms and substitutions [12]. This is eased by the fact that the fuzzy lattice operations do no require altering these conventional first-order structures. There are several other disciplines where this technology has potential for fuzzifying applications wherever $\mathcal{FOT}$s are used for their lattice-theoretic properties such as linguistics and learning. Finally, most promising is using this work's approach to more generic and more expressive knowledge structures for applications such as Fuzzy Information Retrieval [6]. We are currently developing the same formal construction for fuzzy lattice operations over order-sorted feature ($\mathcal{OSF}$) graphs [4]. Encouraging initial results are being reported in [2].

References

1. Aho, A., Hopcroft, J., Ullmann, J.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, MA (1974)
2. Aït-Kaci, H., Pasi, G.: Fuzzy lattice-theoretic operations over data and knowledge structures. Technical Report (in preparation), HAK Language Technologies (2017), [online]
3. Aït-Kaci, H., Pasi, G.: Fuzzy lattice operations on first-order terms over signatures with similar constructors—a constraint-based approach. Journal Submission Preprint (submitted in July 2017, revised in September 2108), [online]
4. Aït-Kaci, H., Podelski, A., Goldstein, S.C.: Order-sorted feature theory unification. Journal of Logic Programming 30(2), 99–124 (1997), [online]
5. Aït-Kaci, H., Sasaki, Y.: An axiomatic approach to feature term generalization. In: de Raedt, L., Flach, P. (eds.) Proceedings of the 12^{th} European Conference on Machine Learning (ECML'01). pp. 1–12. LNCS 2167, Springer-Verlag, Berlin Heidelberg (September 2001), [online]
6. Baziz, M., Boughanem, M., Pasi, G., Prade, H.: A fuzzy set approach to concept-based information retrieval. In: Montseny, E., Sobrevilla, P. (eds.) Proceedings of the Joint 4^{th} Conference of the European Society for Fuzzy Logic and Technology and the *11èmes Rencontres Francophones sur la Logique Floue et ses Applications*, Barcelona (Spain). pp. 1287–1292. EUSFLAT/LFA (September 7–9, 2005), [online]
7. Colmerauer, A.: Prolog and infinite trees. In: Clark, K.L., Åke Tärnlund, S. (eds.) Logic Programming, pp. 231–251. Academic Press (1982)
8. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications, Mathematics in Science and Engineering, Edited by William F. Ames, Georgia Institute of Technology, vol. 144. Academic Press (180), [online]

9. van Emden, M.H., Lloyd, J.W.: A logical reconstruction of Prolog II. Journal of Logic Programming 2, 143–149 (1984), [online]
10. Herbrand, J.: Recherches sur la théorie de la démonstration. Ph.D. thesis, Faculté des sciences de l'université de Paris, Paris (France) (1930), [online]; English translation in [11]
11. Herbrand, J.: Logical Writings. Harvard University Press, Cambridge, MA (1971), edited by Warren D. Goldfarb
12. Iranzo, P.J., Manzano, C.R.: A similarity-based WAM for Bousi∼Prolog. In: Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN '09)—Part I: Bio-Inspired Systems: Computational and Ambient Intelligence. pp. 245–252. Springer, Salamanca, Spain (June 10–12, 2009), [online]
13. Jaffar, J.: Efficient unification over infinite terms. New Generation Computing 2(3), 207 – 219 (September 1984), [online]
14. Knight, K.: Unification: a multidisciplinary survey. ACM Computing Surveys 21(1), 93–124 (March 1989), [online]
15. Lacoste-Julien, S., Palla, K., Davies, A., Kasneci, G., Graepel, T., Ghahramani, Z.: SiGMa: Simple greedy matching for aligning large knowledge bases. In: Dhillon, I.S., Koren, Y., Ghani, R., Senator, T.E., Bradley, P., Parekh, R., He, J., Grossman, R.L., Uthurusamy, R. (eds.) Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2013—Chicago, IL, USA). pp. 572–580. Association for Computing Machinery, ACM, New York, NY (USA) (August 11–14, 2013), [online]; see also [online]
16. Martelli, A., Montanari, U.: An efficient unification algorithm. ACM Transactions on Programming Languages and Systems 4(2), 258–282 (April 1982), [online]
17. Plotkin, G.D.: Lattice theoretic properties of subsumption. Technical Memo MIP-R-77, Department of Machine Intelligence and Perception, University of Edinburgh, Edinburgh, Scotland (UK) (June 1970)
18. Plotkin, G.D.: A note on inductive generalization. In: Metzer, B., Michie, D. (eds.) Machine Intelligence 5, chap. 8, pp. 154–163. Edinburgh University Press, Edinburgh, Scotland (UK) (1970), [online]
19. Reynolds, J.C.: Transformational systems and the algebraic nature of atomic formulas. In: Metzer, B., Michie, D. (eds.) Machine Intelligence 5, chap. 7, pp. 135–151. Edinburgh University Press, Edinburgh, Scotland (UK) (1970), [online]
20. Sessa, M.I.: Approximate reasoning by similarity-based SLD resolution. Theoretical Computer Science 275, 389–426 (2002), [online]
21. Wayne, K.: Union-find. Tutorial lecture slides based on book "Algorithm Design" by Jon Kleinberg and Éva Tardos (Addison-Wesley, 2015), [online]