# Fuzzy Order-Sorted Feature Term Unification

Gian Carlo Milanese[1] and Gabriella Pasi[2]

[1] University of Milano-Bicocca, Milan, Italy
g.milanese1@campus.unimib.it
[2] University of Milano-Bicocca, Milan, Italy
gabriella.pasi@unimib.it

**Abstract**

This paper provides a generalized definition of the unification of Order-Sorted Feature (OSF) terms that considers a *fuzzy subsumption relation* between sort symbols rather than an ordinary (crisp) one. In this setting the unifier of two OSF terms is associated with a subsumption degree. We refer to the problem of unifying two OSF terms and computing the associated subsumption degree as *fuzzy OSF term unification*.

## 1 Introduction

Approximate reasoning grounded on fuzzy relations is a research area that has been investigated extensively. Early work includes Ying's logic for approximate reasoning [22] and the first papers on similarity-based logic programming [6, 12, 21]. One motivation behind the similarity-based approaches was to model a form of reasoning that may be referred to as *reasoning by analogy or similarity*. For example, this may be achieved by relaxing the *equality* constraint on two functor symbols, when unifying two first-order terms (FOTs), to the flexible constraint that they must be *similar*. This kind of relaxed unification is generally referred to as *weak unification*.

This research line has been extended in a number of ways, such as via proximity-based approaches [14, 16, 17]; moreover, weak unification has been implemented in fuzzy logic programming systems such as Bousi∼Prolog [15] and FASILL [13]. Aït-Kaci and Pasi [3] have presented a procedure for weak unification that not only tolerates different (but similar) functor symbols, but also allows the unification of FOTs with a different number and possibly a different order of arguments. This work has been generalized to proximity relations [20], and a possible incorporation in Bousi∼Prolog has been proposed [10].

The work by Aït-Kaci and Pasi was preliminary towards the definition of similarity-based reasoning with Order-Sorted Feature (OSF) logic, a knowledge representation language developed by Aït-Kaci [4], which has found applications in constraint logic programming and computational linguistics (see, e.g., [7] for feature-based logical formalisms and their use in linguistics). One advantage of OSF logic is that its unification algorithm takes into account a *subsumption* (IS-A) ordering between sorts, which enables a single unification step to potentially replace several resolution steps, possibly leading to more efficient computations [2, 9].

This paper provides a generalized definition of subsumption between OSF terms and of the unification of OSF terms that considers a *fuzzy subsumption relation between sorts symbols* rather than an ordinary (crisp) one. We refer to the problem of unifying two OSF terms with an underlying fuzzy subsumption relation and computing the associated subsumption degree as *fuzzy OSF term unification*. One benefit of the present approach is that the unification itself can be performed with the same rules of (crisp) OSF term unification. The introduction of fuzziness in the sort signature is meant to provide more modeling flexibility by allowing to represent imprecise knowledge. Ultimately, the goal of this research is to define a framework based on OSF logic that allows to deal with imperfect knowledge and information, and that enables approximate matching in applications such as information retrieval and question answering.

## 2 OSF Terms and Fuzzy Sort Subsumption

OSF logic [4] is based on *sort symbols* and *feature symbols*: sort symbols denote concepts or classes of objects, while feature symbols are interpreted as properties or attributes of sorts. These symbols together with *variables* (or *coreference tags*) are used to construct OSF terms.

**Definition 2.1** (OSF Terms, $\mathit{Root}(\psi)$ and $\mathit{Tags}(\psi)$)**.** Let $\mathcal{V}$ be a set of variables, $\mathcal{S}$ be a set of sort symbols and $\mathcal{F}$ be a set of feature symbols. Then (i) a variable $X \in \mathcal{V}$ is an OSF term; (ii) if $X \in \mathcal{V}$ and $s \in \mathcal{S}$, a sorted variable $X : s$ is an OSF term; (iii) if $\psi_1, \ldots, \psi_n$ are OSF terms, $f_1, \ldots, f_n \in \mathcal{F}$, $X \in \mathcal{V}$ and $s \in \mathcal{S}$, then $X : s(f_1 \to \psi_1, \ldots, f_n \to \psi_n)$ is an OSF term. The variable $X$ is called the root of the term $\psi$ and denoted $\mathit{Root}(\psi)$. The set of variables appearing in the term $\psi$ is denoted $\mathit{Tags}(\psi)$.

**Example 2.2** (OSF Term)**.** An example of an OSF term is the following:

$$\psi = X_0 : \mathit{movie} \begin{pmatrix} \mathit{directed\_by} & \to & X : \mathit{director}\,(\mathit{name} \to X_1 : \mathit{string})\,, \\ \mathit{written\_by} & \to & X \end{pmatrix} \qquad (\psi)$$

where *movie*, *director* and *string* are sort symbols, *name*, *written_by* and *directed_by* are feature symbols, and $X, X_0$ (the root) and $X_1$ are variables denoting objects of the domain (e.g., $X$ must denote a director). Informally, this term represents movies that are written and directed by the same individual. OSF terms generalize FOTs: for instance, the FOT `movie("Psycho","Hitchcock")` can be translated into OSF syntax as $\mathit{movie}(1 \to$ *"Psycho"*, $2 \to$ *"Hitchcock"*), or extended using feature symbols into $\mathit{movie}(\mathit{name} \to$ *"Psycho"*, $\mathit{directed\_by} \to$ *"Hitchcock"*) for increased interpretability (variables are omitted for readability).

In the regular setting, the set $\mathcal{S}$ of sort symbols is ordered by an IS-A *subsumption relation* $\preceq$, formally a finite lattice. In this paper we generalize this relation to a *fuzzy subsumption relation*, that is, we consider a fuzzy lattice $(\mathcal{S}, \preceq)$ on the (finite) set $\mathcal{S}$ of sort symbols, meaning that $\preceq$ is a fuzzy binary relation (a function) $\preceq : \mathcal{S} \times \mathcal{S} \to [0,1]$ that satisfies:

$$\forall s \in \mathcal{S},\ \ \preceq(s, s) = 1 \qquad \text{(Fuzzy Reflexivity)}$$

$$\forall s, s' \in \mathcal{S},\ \text{if } s \preceq s'^{1} \text{ and } s' \preceq s, \text{then } s = s' \qquad \text{(Fuzzy Antisymmetry)}$$

$$\forall s_0, s_1, s_2 \in \mathcal{S},\ \ \preceq(s_0, s_2) \geq \min(\preceq(s_0, s_1), \preceq(s_1, s_2)) \qquad \text{(Max-Min Transitivity)}$$

and such that the *fuzzy greatest lower bound (GLB)* $s_0 \curlywedge s_1$ exists for each $s_0, s_1 \in \mathcal{S}$, where $s_0 \curlywedge s_1$ is defined as the unique $s \in \mathcal{S}$ such that $s \preceq s_0$, $s \preceq s_1$, and, for all $s' \in \mathcal{S}$, if $s' \preceq s_0$ and $s' \preceq s_1$, then $s' \preceq s$ [8, 18]. We also assume that the bottom ($\bot$) and top ($\top$) elements of the lattice are such that, for any $s \in \mathcal{S}$, $\preceq(\bot, s) = \preceq(s, \top) = 1$.

The GLB $s \curlywedge s'$ of two sorts $s, s' \in \mathcal{S}$ is associated with a *subsumption degree* defined as $\mathit{GLBDegree}(s, s') = \min(\preceq(s \curlywedge s', s), \preceq(s \curlywedge s', s'))$. We write $s \curlywedge_\alpha s'$ to express that the (fuzzy) GLB $s \curlywedge s'$ is associated with the subsumption degree $\mathit{GLBDegree}(s, s') = \alpha$.

**Example 2.3** (Fuzzy GLBs)**.** Consider the weighted directed acyclic graph (DAG) of Figure 1. Its (max-min) transitive closure [11] is a fuzzy lattice (the bottom and top elements are omitted in the figure). For example, the (fuzzy) GLB of $u$ and $t$ is $u \curlywedge t = q$ and $\mathit{GLBDegree}(u, t) = 0.6$.

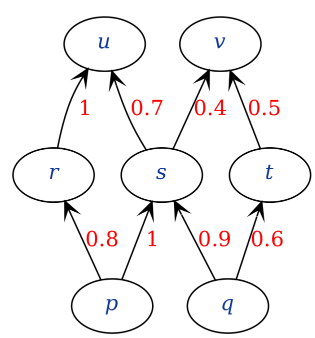


Figure 1: Fuzzy Lattice

[1] We write $s \preceq s'$ to abbreviate $\preceq(s, s') > 0$.

**Proposition 2.4** (GLBs in the Support of a Fuzzy Lattice). *Let* $|\preceq|$ *denote the* support *of the fuzzy binary relation* $\preceq$, *which is defined as* $|\preceq| = \{(s, s') \in \mathcal{S} \times \mathcal{S} \mid \preceq(s, s') > 0\}$. *It holds that (i) if* $(\mathcal{S}, \preceq)$ *is a fuzzy lattice, then* $(\mathcal{S}, |\preceq|)$ *is a lattice, and (ii) fuzzy GLBs in* $(\mathcal{S}, \preceq)$ *correspond to GLBs in* $(\mathcal{S}, |\preceq|)$, *i.e.,* $s \curlywedge s' = s \wedge s'$ *for all* $s, s' \in \mathcal{S}$, *where* $\curlywedge$ *and* $\wedge$ *are the (fuzzy) GLB operations on* $(\mathcal{S}, \preceq)$ *and* $(\mathcal{S}, |\preceq|)$, *respectively.*

Thanks to this fact it is possible to compute fuzzy GLBs exactly as we would in an ordinary lattice, which can be performed in constant time after a linear time preprocessing of the DAG representing the sort subsumption relation [1]. On the other hand, for any two sorts $s, s' \in \mathcal{S}$, the values $\preceq(s, s')$ and $GLBDegree(s, s')$ can be computed in linear time (in the size of the weighted DAG representing the fuzzy sort subsumption) with a shortest-paths-like algorithm that can be optimized thanks to the same preprocessing step [19].

# 3 Fuzzy OSF Term Subsumption and Unification

The definition of OSF terms given above does not rule out the presence of redundant or even contradictory information (e.g., consider the OSF term $s(f \to s_0, f \to s_0, f \to s_1)$, which is contradictory if $s_0 \curlywedge s_1 = \bot$). OSF terms that are well-behaved to this regard are called *normal OSF terms* and are defined as follows [4]. Unless explicitly stated otherwise, *it is always assumed that an OSF term is in normal form*[2].

**Definition 3.1** (Normal OSF term). An OSF term $\psi = X : s(f_1 \to \psi_1, \ldots, f_n \to \psi_n)$ is *normal* (or *in normal form*) if: (i) $s \neq \bot$, (ii) the features $f_1, \ldots, f_n$ are pairwise distinct, (iii) each $\psi_i$ is in normal form, and (iv) for all $Y \in Tags(\psi)$, there is at most one occurrence of $Y$ in $\psi$ such that $Y$ is the root variable of a non-trivial OSF term (i.e., different from $Y : \top$).

For an OSF term $\psi$ in normal form and $X \in Tags(\psi)$, we let $Sort_\psi(X)$ be the most specific sort $s$ such that $X : s$ appears in $\psi$. The expression $X.f \doteq_\psi Y$ means that there is a feature $f$ pointing from a subterm of $\psi$ with root $X$ to a subterm of $\psi$ with root $Y$.

The fuzzy subsumption relation between sorts can be extended to (normal) OSF terms according to the following definition, which generalizes the one for OSF term subsumption [4][3].

**Definition 3.2** (Fuzzy OSF Term Subsumption). An OSF term $\psi_1$ *is subsumed by an OSF term* $\psi_2$ *with degree* $\alpha \in [0, 1]$ (denoted $\psi_1 \preceq_\alpha \psi_2$) if there is a mapping $h : Tags(\psi_2) \to Tags(\psi_1)$ such that (i) $h(Root(\psi_2)) = Root(\psi_1)$, (ii) $\alpha = \min\{\preceq(Sort_{\psi_1}(h(X)), Sort_{\psi_2}(X)) \mid X \in Tags(\psi_2)\}$, and (iii) if $X.f \doteq_{\psi_2} Y$, then $h(X).f \doteq_{\psi_1} h(Y)$. The fuzzy relation $\preceq$ on sorts is extended to OSF terms by letting $\preceq(\psi_1, \psi_2) = \alpha$ if $\psi_1$ is subsumed by $\psi_2$ with degree $\alpha$.

**Example 3.3** (Fuzzy OSF Term Subsumption). Consider the fuzzy lattice of Figure 1 and the terms $\psi_1 = X_0 : q(f \to X_1 : s(g \to X_0))$ and $\psi_2 = Y_0 : v(f \to Y_1 : u(g \to Y_2 : t))$. Then $\psi_1 \preceq_{0.5} \psi_2$ via the mapping $h : Tags(\psi_2) \to Tags(\psi_1)$ defined by $h(Y_0) = h(Y_2) = X_0$ and $h(Y_1) = X_1$, since $h$ preserves the structure of $\psi_2$ and each tag $Y \in Tags(\psi_2)$ is mapped to a tag $h(Y) \in Tags(\psi_1)$ such that $Sort_{\psi_1}(h(Y)) \preceq Sort_{\psi_2}(Y)$; in particular $\preceq(q, v) = 0.5$.

**Proposition 3.4** (Fuzzy OSF Term Subsumption). *The fuzzy subsumption relation between OSF terms is a fuzzy partial order (modulo variable renaming), i.e., it satisfies* Fuzzy Reflexivity, Fuzzy Antisymmetry *(modulo variable renaming) and* Max-Min Transitivity.

[2]The normal form of an OSF term can be computed by applying the OSF constraint normalization rules to its corresponding OSF clause. These notions will be defined shortly.

[3]To be more precise, subsumption between OSF terms is defined semantically in [4], although an equivalent characterization based on homomorphisms between OSF graphs – an alternative syntactic representation of normal OSF terms – is also given. Also see the definition of subsumption between feature structures in [7].

The unification of two OSF terms – an operation that aims to combine two OSF terms in a consistent way – is better presented by introducing an alternative syntactic representation, namely *OSF clauses* [4].

**Definition 3.5** (OSF Clause)**.** An OSF clause $\phi$ is a conjunctive set of expressions (*constraints*) of the form $X : s$, $X \doteq X'$, and $X.f \doteq X'$. The set of variables occurring in $\phi$ is denoted $Tags(\phi)$, while $\phi[X/Y]$ is the OSF clause obtained by replacing all occurrences of $Y$ with $X$.

Informally, the constraint $X : s$ means that the value assigned to $X$ is of sort $s$; $X \doteq X'$ means that the same value is assigned to the variables $X$ and $X'$; while $X.f \doteq X'$ means that applying the feature $f$ to the value assigned to $X$ returns the value assigned to $X'$.

Any OSF term $\psi$ can be translated into an equivalent OSF clause $\phi(\psi)$. For example, the following is the OSF clause corresponding to the term $\psi$ from Example 2.2:

$$\phi(\psi) = \left\{ \begin{array}{lll} X_0 : movie, & X_0.directed\_by \doteq X, & X_0.written\_by \doteq X, \\ X : director, & X.name \doteq X_1, & X_1 : string \end{array} \right\}.$$

The next fact follows from Propositions 2.4 and 3.4 and the analogous result for (crisp) OSF term unification [4].

**Proposition 3.6** (Fuzzy OSF Term Unification)**.** *Two OSF terms $\psi_1$ and $\psi_2$ can be unified by non-deterministically applying any applicable constraint normalization rule (Figure 2) to the clause $\phi(\psi_1) \cup \phi(\psi_2) \cup \{Root(\psi_1) \doteq Root(\psi_2)\}$ until none applies. The resulting clause can be translated back into an OSF term $\psi$, called the unifier of $\psi_1$ and $\psi_2$ (the term $X : \bot$ if the unification fails). The term $\psi$ is the (fuzzy) GLB of $\psi_1$ and $\psi_2$ (up to variable renaming) with respect to the fuzzy subsumption relation of Definition 3.2 and is denoted $\psi_1 \curlywedge \psi_2$.*

**Sort Intersection**
$$\frac{\phi \cup \{X : s, X : s'\}}{\phi \cup \{X : s \curlywedge s'\}}$$

**Feature Functionality**
$$\frac{\phi \cup \{X.f \doteq Y, X.f \doteq Y'\}}{\phi \cup \{X.f \doteq Y, Y \doteq Y'\}}$$

**Inconsistent Sort**
$$\frac{\phi \cup \{X : \bot\}}{fail}$$

**Tag Elimination**
$$\frac{\phi \cup \{X \doteq Y\}}{\phi[X/Y] \cup \{X \doteq Y\}} \quad [Y \in Tags(\phi)]$$

Figure 2: OSF Constraint Normalization Rules

The (fuzzy) subsumption relation plays an essential role when unifying two OSF terms, as highlighted by the rules **Sort Intersection** and **Inconsistent Sort**. In particular, the unification fails if at any point the constraint $X : \bot$ is reached, possibly by applying the rule **Sort Intersection** to two constraints $X : s$ and $X : s'$ such that $s \curlywedge s' = \bot$.

We conclude by specifying how to compute the subsumption degree associated with the GLB of two OSF terms. Let $\phi$ be the OSF clause resulting from the application of the constraint normalization rules to the clause $\phi(\psi_1) \cup \phi(\psi_2) \cup \{Root(\psi_1) \doteq Root(\psi_2)\}$, and $\psi$ be the OSF term corresponding to $\phi$, so that $\psi = \psi_1 \curlywedge \psi_2$. As shown in Example 3.7, the mappings $h_1 : Tags(\psi_1) \to Tags(\psi)$ and $h_2 : Tags(\psi_2) \to Tags(\psi)$ witnessing the relations $\psi \preceq \psi_1$[4] and $\psi \preceq \psi_2$ can be computed from the clause $\phi$. The subsumption degree $\preceq(\psi, \psi_1)$ is given by $\min\{\preceq(Sort_\psi(h_1(X)), Sort_{\psi_1}(X)) \mid X \in Tags(\psi_1)\}$ and similarly for $\preceq(\psi, \psi_2)$. Finally, the

[4] We write $\psi_1 \preceq \psi_2$ to abbreviate $\preceq(\psi_1, \psi_2) > 0$.

subsumption degree associated with the GLB $\psi = \psi_1 \curlywedge \psi_2$ is equal to $\mathit{GLBDegree}(\psi_1, \psi_2) = \min(\preceq(\psi, \psi_1), \preceq(\psi, \psi_2))$.

**Example 3.7** (Fuzzy OSF Term Unification)**.** Consider the OSF terms

$$\psi_1 = Y_0 : u\,(f \to Y_1 : v\,(g \to Y_0, h \to Y_2 : r)) \text{ and } \psi_2 = X_0 : v\,(f \to X_1 : u\,(g \to X_2 : t))$$

and the corresponding clauses $\phi(\psi_1) = \{Y_0 : u, Y_0.f \doteq Y_1, Y_1 : v, Y_1.g \doteq Y_0, Y_1.h \doteq Y_2, Y_2 : r\}$ and $\phi(\psi_2) = \{X_0 : v, X_0.f \doteq X_1, X_1 : u, X_1.g \doteq X_2, X_2 : t\}$. An application of the rules of Figure 2 to $\phi(\psi_1) \cup \phi(\psi_2) \cup \{X_0 \doteq Y_0\}$ (with Figure 1 as the fuzzy subsumption) yields

$$\phi = \{X_0 : q, X_0.f \doteq X_1, X_1 : s, X_1.g \doteq X_0, X_1.h = Y_2, Y_2 : r, X_0 \doteq Y_0, X_0 \doteq X_2, X_1 \doteq Y_1\}$$

or an equivalent clause. The set $\mathit{Tags}(\phi)$ can be partitioned into the equivalence classes $[X_0]_{\doteq} = \{X_0, X_2, Y_0\}$, $[X_1]_{\doteq} = \{X_1, Y_1\}$ and $[Y_2]_{\doteq} = \{Y_2\}$. A new tag can be introduced for each class, say $Z_0$ for $[X_0]_{\doteq}$, $Z_1$ for $[X_1]_{\doteq}$ and $Z_2$ for $[Y_2]_{\doteq}$. The unifier of $\psi_1$ and $\psi_2$ can be constructed from $\phi$ using these new variables, resulting in

$$\psi = Z_0 : q(f \to Z_1 : s(g \to Z_0, h \to Z_2 : r)).$$

The functions $h_i : \mathit{Tags}(\psi_i) \to \mathit{Tags}(\psi)$ $(i = 1, 2)$ can be defined by mapping each variable to the tag associated with its equivalence class. Thus $h_2(X_0) = h_2(X_2) = h_1(Y_0) = Z_0$, $h_2(X_1) = h_1(Y_1) = Z_1$ and $h_1(Y_2) = Z_2$. The subsumption degrees are $\preceq(\psi, \psi_1) = \min\{\preceq(q, u), \preceq(s, v), \preceq(r, r)\} = 0.4$ and $\preceq(\psi, \psi_2) = \min\{\preceq(q, v), \preceq(s, u), \preceq(q, t)\} = 0.5$. Overall, the subsumption degree associated with the GLB $\psi = \psi_1 \curlywedge \psi_2$ is $\mathit{GLBDegree}(\psi_1, \psi_2) = 0.4$. The unification is depicted in Figure 3, where $\psi_1$ corresponds to the blue graph, $\psi_2$ to the red graph, $\psi$ to the yellow graph, and the dashed arrows represent the mappings.

# 4 Future Work

The generalized definition of the unification of OSF terms presented in this paper – which is currently being implemented – is relevant not only from a knowledge representation standpoint – as it provides more flexibility by allowing to model imprecise knowledge – but also to our current research towards the definition of similarity-based unification of OSF terms. Indeed, given both a subsumption and a similarity relation on $\mathcal{S}$, we are considering a method to define a fuzzy ordering on $\mathcal{S}$ in order to reduce the similarity-based setting to the one of this paper. This will be the subject of a future publication.

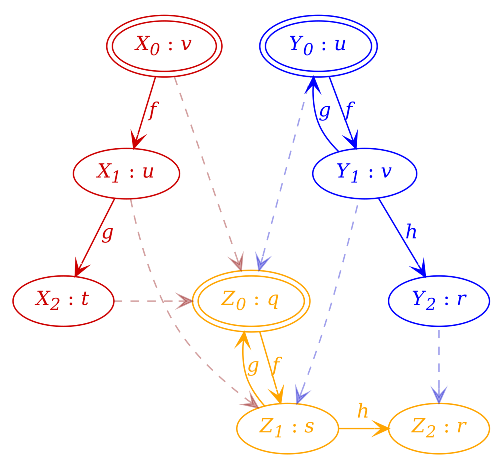


Figure 3: Fuzzy Unification of Example 3.7

Other research directions include the development of a fuzzy logic programming language (e.g., in the style of Login [2]) that supports a fuzzy subsumption relation and/or a similarity relation between sort symbols in order to provide approximate solutions to a program, or a fuzzy extension of the CEDAR Semantic Web reasoner [5].

# References


[1] H. Aït-Kaci, R. Boyer, P. Lincoln, and R. Nasr. Efficient implementation of lattice operations. *ACM Trans. Program. Lang. Syst.*, 11(1):115–146, 1989.

[2] H. Aït-Kaci and R. Nasr. Login: a logic programming language with built-in inheritance. *The Journal of Logic Programming*, 3(3):185–215, 1986.

[3] H. Aït-Kaci and G. Pasi. Fuzzy lattice operations on first-order terms over signatures with similar constructors: A constraint-based approach. *Fuzzy Sets and Systems*, 391:1–46, 2020.

[4] H. Aït-Kaci and A. Podelski. Towards a meaning of life. *The Journal of Logic Programming*, 16(3):195–234, 1993.

[5] S. Amir and H. Aït-Kaci. An efficient and large-scale reasoning method for the semantic web. *J. Intell. Inf. Syst.*, 48(3):653–674, 2017.

[6] F. Arcelli-Fontana and F. Formato. A similarity-based resolution rule. *International Journal of Intelligent Systems*, 17:853–872, 2002.

[7] R. L. Carpenter. *The Logic of Typed Feature Structures: With Applications to Unification Grammars, Logic Programs and Constraint Resolution.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[8] I. Chon. Fuzzy partial order relations and fuzzy lattices. *Korean Journal of Mathematics*, 17(4):361–374, 2009.

[9] A. G. Cohn. Taxonomic reasoning with many-sorted logics. *Artificial intelligence review*, 3(2-3):89–128, 1989.

[10] M. E. Cornejo, J. Medina Moreno, and C. Rubio-Manzano. Towards a full fuzzy unification in the Bousi∼Prolog system. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018.

[11] D. Dubois and H. Prade. *Fuzzy Sets and Systems. Theory and Applications.*, volume 144 of *Mathematics in Science and Engineering*. Elsevier, 1980.

[12] G. Gerla and M. I. Sessa. *Similarity in Logic Programming*, pages 19–31. Springer US, Boston, MA, 1999.

[13] P. Julián-Iranzo, G. Moreno, and J. A. Riaza. The fuzzy logic programming language fasill: Design and implementation. *International Journal of Approximate Reasoning*, 125:139–168, 2020.

[14] P. Julián-Iranzo and C. Rubio-Manzano. Proximity-based unification theory. *Fuzzy Sets and Systems*, 262:21 – 43, 2015.

[15] P. Julián-Iranzo, C. Rubio-Manzano, and J. Gallardo-Casero. Bousi∼Prolog: a prolog extension language for flexible query answering. *Electronic Notes in Theoretical Computer Science*, 248:131 – 147, 2009. Proceedings of the Eighth Spanish Conference on Programming and Computer Languages (PROLE 2008).

[16] P. Julián-Iranzo and F. Sáenz-Pérez. Proximity-based unification: an efficient implementation method. *IEEE Transactions on Fuzzy Systems*, 29(5):1238–1251, 2020.

[17] T. Kutsia and C. Pau. Solving proximity constraints. In M. Gabbrielli, editor, *Logic-Based Program Synthesis and Transformation*, pages 107–122. Springer International Publishing, 2020.

[18] I. Mezzomo and B. C. Bedregal. On fuzzy $\alpha$-lattices. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016.

[19] G. C. Milanese and G. Pasi. Conjunctive reasoning on fuzzy taxonomies with order-sorted feature logic. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2021.

[20] C. Pau and T. Kutsia. Proximity-based unification and matching for fully fuzzy signatures. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2021.

[21] M. I. Sessa. Approximate reasoning by similarity-based SLD resolution. *Theoretical Computer Science*, 275(1):389–426, 2002.

[22] M. Ying. A logic for approximate reasoning. *The Journal of Symbolic Logic*, 59(3):830–837, 1994.