Matching Modulo Proximity Theories

Temur Kutsia and Cleo Pau
RISC, Johannes Kepler University Linz, Austria

1 Introduction

Matching is a fundamental technique in rule-based computational formalisms. Given a pattern expression and a data expression, its task is to instantiate the pattern in such a way that it "fits" the data, where "fitting" typically means syntactic equality, equivalence modulo the given theory, or some other predefined relation. In this paper, we study matching modulo theories that generalize equivalence to tolerance: reflexive, symmetric, but not necessarily transitive relations. Our relations are quantitative: they are fuzzy proximities. We introduce conditional proximity equations (CPEs) of a special type, define the notion of proximity theory induced by CPEs, and develop an algorithm for solving matching problems in proximity theories.

Our work is a generalization and extension of some earlier works. The notion of CPEs is quite powerful and flexible and helps to model different concepts of proximities between first-order terms. Consequently, in our framework we can express class-based proximity matching in basic [4] and fully fuzzy signatures [6]. It can be also used with block-based algorithms that can solve proximity matching problems [1, 2]. Moreover, proximity theories as we define them can be seen as an extension of shallow collapse-free equational theories from crisp (two-valued) to a fuzzy setting.

2 Preliminaries

Proximity relations. A binary *fuzzy relation* on a set S is a mapping from $S \times S$ to the real interval $[0, 1]$. If $\mathcal{R}$ is a fuzzy relation on S and λ is a number $0 < \lambda \leq 1$ (called *cut value*), then the λ-*cut* of $\mathcal{R}$ on S, denoted $\mathcal{R}_\lambda$, is an ordinary (crisp) relation on S defined as $\mathcal{R}_\lambda := \{(s_1, s_2) \mid \mathcal{R}(s_1, s_1) \geq \lambda\}$.

A fuzzy relation $\mathcal{R}$ on a set S is called a *proximity relation* on S iff it is reflexive ($\mathcal{R}(s, s) = 1$ for all $s \in S$) and symmetric ($\mathcal{R}(s_1, s_2) = \mathcal{R}(s_2, s_1)$ for all $s_1, s_2 \in S$).

Two objects s_1 and s_2 are called δ-*proximal* in the proximity relation $\mathcal{R}$ to each other if $\mathcal{R}(s_1, s_2) = \delta$. We write it as $s_1 \simeq_{\mathcal{R},\delta} s_2$ or just $s_1 \simeq_\delta s_2$ if the relation $\mathcal{R}$ is clear from the context. The bigger the δ is, the more proximal the objects are to each other.

A *triangular norm* (T-norm) $\otimes$ on $[0, 1]$ is a binary operation on this interval, which is associative, commutative, nondecreasing in both arguments, and having 1 as its unit element. T-norms have been studied in detail in [3]. In this paper we assume that the T-norm is minimum (Gödel T-norm).

Language. The set of first-order *terms* $\mathcal{T}(\mathcal{F}, \mathcal{V})$ over disjoint sets of variables $\mathcal{V}$ and fixed arity function symbols $\mathcal{F}$ is defined as usual. We use s, r, t to denote them. $\mathcal{V}(t)$ stands for the set of variables of t. *Substitutions* over $\mathcal{T}(\mathcal{F}, \mathcal{V})$ are mappings from variables to terms, where all but finitely many variables are mapped to themselves. The symbols $\sigma, \vartheta, \varphi$ are used for substitutions. The identity substitution is denoted by Id. We use the usual set notation for substitutions. *Substitution application* to terms is written in the postfix notation such as $t\sigma$ and is defined recursively as $x\sigma = \sigma(x)$ and $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$.

Below we will be using proximity degree variables, proximity degree constraints, and their solutions. A *degree variable* $\mathfrak{d}$ is a variable that takes its values from the real interval $[0, 1]$. A *degree expression* $\updelta$ is defined by the grammar $\updelta ::= n \mid \mathfrak{d} \mid \updelta_1 \otimes \updelta_2$, where $n \in [0, 1]$. A *degree constraint* is an atomic formula $a \leq \updelta \leq b$, where a and b are numbers from $[0, 1]$. We will also write shortly $\updelta \geq a$ for $a \leq \updelta \leq 1$ and $\updelta = a$ for $a \leq \updelta \leq a$. A *degree mapping* is a mapping from degree variables to degree expressions. The application of a degree mapping $\upmu$ on a degree constraint $a \leq \updelta \leq b$, denoted $\upmu(a \leq \updelta \leq b)$, is defined as $a \leq \upmu(\updelta) \leq b$.

Background Knowledge. Background knowledge specifies which terms are to be considered proximal (and by which degree). It is represented by a set of conditional proximal equations (called *axioms* of the background proximal theory), which are defined in the following way:

Definition 1 (Conditional proximal equation). *A* conditional proximal equation *(CPE) is a formula of the form* $f(x_1,\dots,x_n) \simeq_{\lambda\otimes\mathfrak{d}_1\otimes\cdots\otimes\mathfrak{d}_k} g(y_1,\dots,y_m) \Leftarrow \mathsf{Def}\wedge\mathsf{Res}\wedge\mathsf{Deg}$ *where* $\lambda\in(0,1]$, $x_1,\dots,x_n,y_1,\dots,y_m$ *are pairwise distinct variables,* $\mathfrak{d}_1,\dots,\mathfrak{d}_k$ *are pairwise distinct degree variables, and* Def, Res, *and* Deg *are conjunctive formulas (called respectively* defining, restricting, and degree constraints*) defined as follows:*

- $\mathsf{Def} = u_1 \simeq_{\mathfrak{d}_1} v_1 \wedge\cdots\wedge u_k \simeq_{\mathfrak{d}_k} v_k$, $k\geq 0$, *where for each* $1\leq i\leq k$, $u_i\in\{x_1,\dots,x_n\}$, $v_i\in\{y_1,\dots,y_m\}$, *and they are not necessarily distinct;*
- $\mathsf{Res} = z_1 \simeq_{\mathfrak{g}_1} t_1 \wedge\cdots\wedge z_l \simeq_{\mathfrak{g}_l} t_l$, $l\geq 0$, *where* $\mathfrak{g}_1,\dots,\mathfrak{g}_k$ *are pairwise distinct degree variables that do not appear among* $\mathfrak{d}_1,\dots,\mathfrak{d}_k$, *and for each* $1\leq j\leq l$,
  - $z_j\in\{x_1,\dots,x_n,y_1,\dots,y_m\}$,
  - $t_j\in\mathcal{T}(\mathcal{F},\mathcal{V}(\mathsf{Def}))$ *(i.e., only variables in* t_j *are those appearing in* Def*),*
  - *if* t_j *is a variable, then* $z_j\in\mathcal{V}(\mathsf{Def})$, *but* $\{z_j,t_j\}\neq\{u_i,v_i\}$ *for all* $1\leq i\leq k$ *(i.e.,* Def *and* Res *do not share equations);*
- $\mathsf{Deg} = a_1\leq\mathfrak{d}_1\leq b_1\wedge\cdots\wedge a_k\leq\mathfrak{d}_k\leq b_k\wedge c_1\leq\mathfrak{g}_1\leq d_1\wedge\cdots\wedge c_l\leq\mathfrak{g}_l\leq d_l$ *such that for each* $1\leq j\leq l$,
  - *if* $c_j=0$, *then* $z_j\in\mathcal{V}(\mathsf{Def})$,
  - *if for* $z_j\simeq_{\mathfrak{g}_j} t_j\in\mathsf{Res}$, z_j *and* t_j *are variables such that one of them belongs to* $\{u_1,\dots,u_k\}$ *and the other one to* $\{v_1,\dots,v_k\}$, *then* $c_j=d_j=0$.

Intuitively, if $s=f(x_1,\dots,x_n)$ and $t=g(y_1,\dots,y_m)$, the definition says:

- Def (together with Deg) defines which arguments of s and t should be close to each other. Their proximity degrees are used in the computation of the proximity degree between s and t;
- Res puts additional restrictions that do not affect the computation of the proximity degree between s and t. It may require, e.g. that some arguments within s or within t are close to each other or should not be close to each other; or that some argument of s (or of t) should be close or not close with some term (guaranteed to be ground in matching problems); or that some argument of s must not be close to some argument of t (also in this case, during matching, this check will be performed on ground terms);
- Deg makes sure that if an equation from Res is allowed to have the zero degree, then the sides of this equation contain only variables from Def (guaranteeing the ground term check in matching); and if an equation from Res is between arguments of s and t, then they must not be close to each other (proximal with degree zero). Valid degree constraints are usually dropped from Deg.

Remark 1. Using conditional proximities, we can encode proximities between symbols in fully fuzzy signatures [5,6]. For instance, $f \sim_{0.6}^{\{(1,1),(1,2),(3,2)\}} g$ for a ternary f and binary g can be written as the conditional proximity $f(x_1,x_2,x_3)\simeq_{\min\{0.6,\mathfrak{d}_1,\mathfrak{d}_2,\mathfrak{d}_3\}} g(y_1,y_2) \Leftarrow x_1\simeq_{\mathfrak{d}_1} y_1\wedge x_1\simeq_{\mathfrak{d}_2} y_2\wedge x_3\simeq_{\mathfrak{d}_3} y_2$. It uses only the Def part. The Res and Deg constraints are empty and, hence, valid conjunctions.[1]

Definition 2 (Proximity theory). *Given a set of conditional proximities* $\mathcal{A}$ *(over the set of terms* $\mathcal{T}(\mathcal{F},\mathcal{V})$*), the* set of proximities induced by $\mathcal{A}$, *denoted by* $\mathcal{P}(\mathcal{A})$, *is the least set satisfying the conditions:*

1. $t\simeq_1 t\in\mathcal{P}(\mathcal{A})$ *for all* $t\in\mathcal{T}(\mathcal{F},\mathcal{V})$.
2. *If* $t\simeq_\lambda s\in\mathcal{P}(\mathcal{A})$, *then* $s\simeq_\lambda t\in\mathcal{P}(\mathcal{A})$.
3. *If* $t_1\simeq_{\lambda_1} s_1,\dots,t_n\simeq_{\lambda_n} s_n\in\mathcal{P}(\mathcal{A})$, $\lambda_1\otimes\cdots\otimes\lambda_n=\lambda$, *and* $f\in\mathcal{F}$ *is an* n*-ary symbol, then* $f(t_1,\dots,t_n)\simeq_\lambda f(s_1,\dots,s_n)\in\mathcal{P}(\mathcal{A})$.
4. *For each axiom from* $\mathcal{A}$ *of the form* $f(x_1,\dots,x_n)\simeq_\delta g(y_1,\dots,y_m)\Leftarrow\mathsf{Def}\wedge\mathsf{Res}\wedge\mathsf{Deg}$, *if there exist a substitution* σ *and a degree variable mapping* μ *such that*
   (a) *for each* $u_i\simeq_{\mathfrak{d}_i} v_i\in\mathsf{Def}$, *we have* $\mu(\mathfrak{d}_i)>0$ *and there exists* $\gamma_i\leq\mu(\mathfrak{d}_i)$ *such that* $u_i\sigma\simeq_{\gamma_i} v_i\sigma\in\mathcal{P}(\mathcal{A})$,
   (b) *for each* $z_j\simeq_{\mathfrak{g}_j} t_j\in\mathsf{Res}$,
      - *if* $\mu(\mathfrak{g}_j)>0$, *then there exists* $\gamma_j\leq\mu(\mathfrak{g}_j)$ *such that* $z_j\sigma\simeq_{\gamma_j} t_j\sigma\in\mathcal{P}(\mathcal{A})$,
      - *if* $\mu(\mathfrak{g}_j)=0$, *then* $z_j\sigma\simeq_0 t_j\sigma\notin\mathcal{P}(\mathcal{A})$,
   (c) *for each* $D\in\mathsf{Deg}$, *the formula* $\mu(D)$ *is valid, and*

[1] Strictly speaking, Deg is $0\leq\mathfrak{d}_1\leq 1\wedge 0\leq\mathfrak{d}_2\leq 1\wedge 0\leq\mathfrak{d}_3\leq 1$, which is a valid formula.

(d) $\mu(\mathfrak{d}_i)$ *for each* $1 \le i \le k$ *and* $\mu(\mathfrak{g}_j)$ *for each* $1 \le j \le l$ *are the smallest numbers satisfying the properties (4a)–(4c),*[2]

then $f(x_1, \dots, x_n)\sigma \simeq_{\mu(\delta)} g(y_1, \dots, y_m)\sigma \in \mathcal{P}(\mathcal{A})$.

The proximity theory induced by $\mathcal{A}$, *denoted* $[\![\mathcal{P}(\mathcal{A})]\!]$, *is the set*

$$[\![\mathcal{P}(\mathcal{A})]\!] := \{t \simeq_\lambda s \mid t \simeq_\lambda s \in \mathcal{P}(\mathcal{A}) \text{ and } \lambda \ge \lambda_i \text{ for all } t \simeq_{\lambda_i} s \in \mathcal{P}(\mathcal{A})\}\,.$$

Remark 2. The definition implies that if $t \simeq_\lambda s \in [\![\mathcal{P}(\mathcal{A})]\!]$, then $\lambda > 0$.

Example 1. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the sets of axioms:

$$\mathcal{A}_1 := \{a \simeq_{0.5} b, \quad f(x_1, x_2) \simeq_{0.6 \otimes \mathfrak{d}_1 \otimes \mathfrak{d}_2} f(y_1, y_2) \Leftarrow x_1 \simeq_{\mathfrak{d}_1} y_2 \wedge x_2 \simeq_{\mathfrak{d}_2} y_1 \wedge 0.4 \le \mathfrak{d}_1 \wedge 0.4 \le \mathfrak{d}_2\}$$

$$\mathcal{A}_2 := \{a \simeq_{0.5} b, \quad a \simeq_{0.6} c, \quad b \simeq_{0.7} d, \quad c \simeq_{0.8} d,$$
$$f(x) \simeq_{0.7} g(y) \Leftarrow x \simeq_{\mathfrak{g}_1} a \wedge y \simeq_{\mathfrak{g}_2} d \wedge 0.5 \le \mathfrak{g}_1 \le 0.6 \wedge \mathfrak{g}_2 \ge 0.8\}.$$

Then

- $f(a, b) \simeq_{0.5} f(a, b) \in \mathcal{P}(\mathcal{A}_1)$ but $f(a, b) \simeq_{0.5} f(a, b) \notin [\![\mathcal{P}(\mathcal{A}_1)]\!]$, because $f(a, b) \simeq_1 f(a, b) \in \mathcal{P}(\mathcal{A}_1)$.
- On the other hand, $f(a, b) \simeq_{0.6} f(b, a) \in \mathcal{P}(\mathcal{A}_1)$ and also $f(a, b) \simeq_{0.6} f(b, a) \in [\![\mathcal{P}(\mathcal{A}_1)]\!]$.
- $\{a \simeq_{0.5} b,\ a \simeq_{0.6} c,\ b \simeq_{0.7} d,\ c \simeq_{0.8} d\} \cup \{f(s) \simeq_{0.7} g(t) \mid s \in \{a, b, c\}, t \in \{c, d\}\} \subset [\![\mathcal{P}(\mathcal{A}_2)]\!]$.

3 The Algorithm

The problem of matching t to a ground term s with the cut value λ with respect to the proximity theory induced by $\mathcal{A}$ is to find a σ such that $(t\sigma \simeq_\delta s) \in [\![\mathcal{P}(\mathcal{A})]\!]$ for some $\delta \ge \lambda$. To solve such a problem, we create the tuple $\{t \precsim^?_{\mathfrak{d}} s\}; \emptyset; \{\mathfrak{d} \ge \lambda\}; Id; \mathfrak{d}$ and apply the rules below. They work on tuples $M; R; D; \sigma; \alpha$, where M is a set of matching equations to be solved, R is the set of restricting constraints to be satisfied, D is a set of degree constraints to be satisfied, σ is the matching substitution computed so far, and α is the degree expression that gives the proximity degree of the solution. Obviously, t and $\mathcal{A}$ are variable disjoint.

The matching rules are the following:

Dec-1: **Decomposition 1**

$\{f(t_1, \dots, t_n) \precsim^?_{\mathfrak{d}} f(s_1, \dots, s_n)\} \uplus M; R; D; \sigma; \alpha \Longrightarrow M \cup \{t_1 \precsim^?_{\mathfrak{d}_1} s_1, \dots, t_n \precsim^?_{\mathfrak{d}_n} s_n\}; R; \mu(D); \sigma; \mu(\alpha)$,
where $\mu = \{\mathfrak{d} \mapsto \mathfrak{d}_1 \otimes \dots \otimes \mathfrak{d}_n\}$.

Dec-2: **Decomposition 2**

$\{f(t_1, \dots, t_n) \precsim^?_{\mathfrak{d}} g(s_1, \dots, s_m)\} \uplus M; R; D; \sigma; \alpha \Longrightarrow$
$M \cup \{t_i \precsim^?_{\mathfrak{d}_i} s_j \mid x_i \simeq_{\mathfrak{d}_i} y_j \in \mathsf{Def}\}; R \cup \{z\vartheta \simeq_{\mathfrak{g}} r\vartheta \mid z \simeq_{\mathfrak{g}} r \in \mathsf{Res}\}; \mu(D) \cup \mathsf{Deg}; \sigma; \mu(\alpha)$,
if $f(x_1, \dots, x_n) \cong_\delta g(y_1, \dots, y_m) \Leftarrow \mathsf{Def} \wedge \mathsf{Res} \wedge \mathsf{Deg} \in \mathcal{A}$,[3] $\vartheta = \{x_i \mapsto t_i \mid 1 \le i \le n\} \cup \{y_j \mapsto s_j \mid 1 \le j \le m\}$, and $\mu = \{\mathfrak{d} \mapsto \delta\}$.

Var-E-M: **Variable elimination**

$\{x \precsim^?_{\mathfrak{d}} s\} \uplus M; R; D; \sigma; \alpha \Longrightarrow M\sigma; R\sigma; \mu(D); \sigma\vartheta; \mu(\alpha)$,
where $t \simeq_\lambda s \in [\![\mathcal{P}(\mathcal{A})]\!]$ for some λ,[4] $\mu = \{\mathfrak{d} \mapsto \lambda\}$, and $\vartheta = \{x \mapsto t\}$.

Solve-R: **Solving restriction constraints, big step**

$\emptyset; R; D; \sigma; \alpha \Longrightarrow \emptyset; \emptyset; D'; \sigma'; \alpha, \qquad$ if $R; \emptyset; D; \sigma; 1 \Longrightarrow^* \emptyset; \emptyset; D'; \sigma'; \_$.

[2] If $\mu(\mathfrak{d}_i)$, $1 \le i \le k$, and $\mu(\mathfrak{g}_j)$, $1 \le j \le l$, satisfy (4a)–(4c), then we can select their smallest values because of the *nonstrict* inequalities that constrain them in Deg.

[3] $s \cong_\delta t$ is a meta-notation for $s \simeq_\delta t$ or $t \simeq_\delta s$.

[4] In variable elimination, one needs to consider only finitely many choices for t because ground terms have only finitely many λ-proximal terms (since $\mathcal{A}$ is finite).

Valid-D: **Valid degree constraint**

$\emptyset; R; \{dc\} \uplus D; \sigma; \alpha \Longrightarrow \emptyset; R; D; \sigma; \alpha,$ if dc is a valid degree constraint.

Cla: **Clash**

$\{f(t_1, \dots, t_n) \precsim^? g(s_1, \dots, s_m)\} \uplus M; R; D; \sigma; \alpha \Longrightarrow \perp,$

if $f \neq g$ and no equation in $\mathcal{A}$ has the head of the form $f(x_1, \dots, x_n) \simeq_\delta g(y_1, \dots, y_m)$.

Inc-R: **Inconsistent restriction, big step**

$\emptyset; R; D; \sigma; \alpha \Longrightarrow \perp,$ if $R; \emptyset; D; \sigma; 1 \Longrightarrow^* \perp$.

Inc-D: **Inconsistent degree constraint**

$M; R; D; \sigma; \alpha \Longrightarrow \perp,$ if D is inconsistent.

In the big steps of Solve-R and Inc-R, R can be treated as a matching problem because at each step of its transformation there will be an equation with a ground side. The matching algorithm $\mathfrak{M}$ uses these rules to transform tuples as long as possible, returning either $\perp$ (indicating failure), or $\emptyset; \emptyset; \emptyset; \vartheta; \alpha$ (indicating success). The theorem below shows that $\mathfrak{M}$ is terminating, sound, and complete.

Theorem 1. *Let t and s be two terms where s us ground, $\mathcal{A}$ be a set of proximity theory axioms, and λ be a cut value. Let $\mathcal{C}$ be the starting configuration $\{t \preceq^?_{\mathfrak{d}} s\}; \emptyset; \{\mathfrak{d} \geq \lambda\}; Id; \mathfrak{d}$. Then*

- *Starting from $\mathcal{C}$, the matching algorithm $\mathfrak{M}$ terminates.*
- *If it terminates with $\emptyset; \emptyset; \emptyset; \vartheta; \alpha$, then ϑ is a proximal matcher of t to s wrt $\mathcal{A}$ with the degree α.*
- *If all derivations via $\mathfrak{M}$ terminate with $\perp$, then there is no substitution that would match t to s with respect to $\mathcal{A}$ with the proximity degree at least λ.*

4 Concluding Remarks

Proximity theories are counterparts of equational theories in the fuzzy setting, where equalities are replaced by their quantitative approximations. They are generated by conditional proximal equations of a special form, which can be also seen as a generalization of shallow collapse-free equations to proximities. The algorithm proposed in this paper solves matching problems under such a background theory. A natural direction of future work would be fuzzy (proximity- or similarity-based) constraint solving modulo theories, aiming at their applications in approximate reasoning.

A version of this paper with an example illustrating the work of the matching algorithm can be found at `https://www.risc.jku.at/people/tkutsia/papers/mmp.pdf`.

Acknowledgment. Supported by the Austrian Science Fund (FWF), project P 35530.

References

1. Julián-Iranzo, P., Rubio-Manzano, C.: Proximity-based unification theory. Fuzzy Sets and Systems **262**, 21–43 (2015)
2. Julián Iranzo, P., Sáenz-Pérez, F.: Proximity-based unification: An efficient implementation method. IEEE Trans. Fuzzy Syst. **29**(5), 1238–1251 (2021)
3. Klement, E., Mesiar, R., Pap, E.: Triangular Norms, Trends in Logic, vol. 8. Springer (2000)
4. Kutsia, T., Pau, C.: Matching and generalization modulo proximity and tolerance relations. In: Özgün, A., Zinova, Y. (eds.) Language, Logic, and Computation - 13th International Tbilisi Symposium, TbiLLC 2019, Batumi, Georgia, September 16-20, 2019, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13206, pp. 323–342. Springer (2019)
5. Kutsia, T., Pau, C.: A framework for approximate generalization in quantitative theories. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13385, pp. 578–596. Springer (2022)
6. Pau, C., Kutsia, T.: Proximity-based unification and matching for fully fuzzy signatures. In: 30th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2021, Luxembourg, July 11-14, 2021. pp. 1–6. IEEE (2021)