JOHANNES KEPLER
UNIVERSITY LINZ

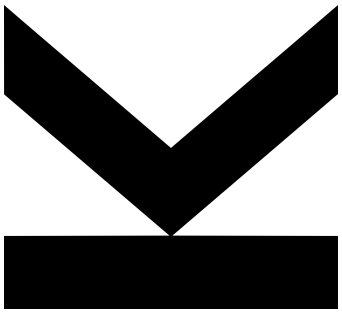
Author
Maximilian Donnermair
12005908

Submission
RISC

Thesis Supervisor
Dipl.-Math. Dr. Temur Kutsia

August 2024

Proximity-based matching with arbitrary T-norms



Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelor's Program

Technische Mathematik

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

Abstract

Matching algorithms modulo proximity relations have been developed with various approaches, where the fuzzy setting usually requires a T-norm to model the notion of fuzzy conjunction. In literature however, only one specific T-norm, the minimum (Gödel) T-norm is used.

We relax this and consider arbitrary T-norms by reformulating a class-based algorithm to this more general setting. We prove that the correctness properties still hold and formulate generalized conditions on solutions.

Contents

1 Introduction

Matching is the fundamental computational mechanism in applications where objects are transformed using rules such as, e.g., functional or rule-based programming, term rewriting, equational theorem proving, querying, etc. In such cases, the object to be transformed is matched by the pattern side of a rule. The problem to be solved during the matching process can be formulated as a problem of solving equations between two logical expressions (a pattern term and a data term), where the solution assigns values to the variables in the pattern term in such a way that the given expressions become equal.

Matching problems are a special case of unification problems where the right-hand side is ground [1], but for efficiency reasons, it is preferred to use dedicated matching algorithms instead of the more general unification algorithms.

A matching problem as formulated above looks at precise solutions, which is also known as *syntactic matching*. In various applications however, background information has to be taken into account, which can be formulated as a set of equalities. In this case, one talks about *equational matching*. Both syntactic and equational matching deal with crisp problems, so terms are either equal (in the equational case modulo the background theory) or not.

For many real-world problems, provided data comes with incompleteness, vagueness or imprecision properties, making it problematic to tackle them in a crisp manner. Therefore, approximate techniques have been developed. One such popular approach is based on *fuzzy relations*, which replace the crisp equality notion by its quantitative approximation, indicating the *degree* (a real number from 0 to 1) by which two objects are considered to be similar, or *close*, to each other. In this setting, fuzzy versions of equivalence (reflexive, symmetric, and transitive) relations are called similarity relations. Dropping the transitivity property, we work with the more general fuzzy tolerance, or *proximity relations*.

To compute the overall *approximation degree* of two objects, represented by terms, one decomposes those terms into their individual symbols, on which the proximity relation is defined. These individual approximation degrees are conjuncted via special binary functions, so-called *T-norms*. Several earlier works tackled fuzzy unification and matching problems with similarity relations [2] and later with proximity relations with block-based [3] and class-based [4] approaches with logic or rule-based programming. However, these works mainly focus on one specific T-norm, namely the Gödel- or minimum-T-norm. In this thesis, we relax this assumption and generalize the proximity-based matching algorithm by considering arbitrary T-norms.

This paper aims to generalize the rule-based matching algorithm developed in [4] for arbitrary T-norms and show if the correctness of the algorithm still holds. Notions and definitions are taken mainly from [1].

The rest of the thesis is organized as follows: In Chapter 2, we introduce the basic notions and define the terminology. In Chapter 3, we introduce the matching problem and recall the proximity-based matching algorithm for the minimum T-norm from [4]. It is a rule-based algorithm, where, in cases of success, the final configuration provides all the solutions. We reformulate this algorithm for arbitrary T-norms, where we will see that the final configuration of the inference system does not automatically contain all desired solutions as it does with the minimum T-norm. Instead, it provides a description of constraints that have to be solved further to get solutions. In Chapter 4, we prove the termination, soundness, and completeness properties of the matching algorithm in this more general setting. In Chapter 5, we will examine the constraint-solving part and how the term representation comes into play. This chapter also contains a full explanatory example. In Chapter 6, concluding remarks and future work are discussed.

2 Preliminaries

In this chapter, we introduce the terminology and notation used throughout the thesis. We mainly follow [1] for notions related to terms and substitutions and [3, 4, 5] for notions related to proximity relations.

2.1 Terms and substitutions

We work on an alphabet consisting of a finite set $\mathcal{V}$ of *variables* whose elements are denoted by x, y, z, and a finite set $\mathcal{F} = \bigcup_{n \geq 0} \mathcal{F}^n$ of *function symbols* of fixed arity n, denoted with f, g, h for positive arity and a, b, c for *constants* ($\in \mathcal{F}^0$).

Terms over $\mathcal{V}$ and $\mathcal{F}$ are defined as

$$t := x \mid f(t_1, \ldots, t_n)$$

with $x \in \mathcal{V}$ and $f \in \mathcal{F}^n$. They are denoted by t, s, r. The notation $\mathcal{T}(\mathcal{F}, \mathcal{V})$ stands for the set of terms over $\mathcal{F}$ and $\mathcal{V}$. $\mathcal{V}(t)$ denotes the set of all variables in a term t. A term t is called *ground* if it contains no variables, i.e., if $\mathcal{V}(t) = \emptyset$.

Positions in terms are defined with respect to their tree representation as strings of integers, where the empty string is denoted by ϵ. The *subterm* of t at position p is denoted by $t|_p$, and $t[p]$ is the *symbol* at position p in t.

Example 1. To compare:

- $f(g(a), b)|_\epsilon = f(g(a), b)$, but $f(g(a), b)[\epsilon] = f$.
- $f(g(a), b)|_1 = g(a)$, but $f(g(a), b)[1] = g$.

- $f(g(a), b)|_{1.1} = a$, and $f(g(a), b)[1.1] = a$.
- $f(g(a), b)|_{2} = b$, and $f(g(a), b)[2] = b$.

A *substitution* is a mapping from Variables to terms, i.e. $\sigma : \mathcal{V} \to \mathcal{T}$, where all but finitely many variables are mapped to themselves. Substitutions are denoted with Greek letters σ or θ. The *domain* of a substitution σ, denoted by $dom(\sigma)$, is the set containing all variables changed by σ, i.e., $dom(\sigma) = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$.

Application of a substitution σ to a term t, denoted by $t\sigma$, is defined as follows:

$$t\sigma = \begin{cases} \sigma(t), & \text{if } t = x, \\ f(t_1\sigma, \ldots, t_n\sigma), & \text{if } t = f(t_1, \ldots, t_n). \end{cases}$$

Example 2.

$t = f(x, y)$,
$\sigma = \{x \mapsto a, y \mapsto b\}$,
$t\sigma = f(x, y)\{x \mapsto a, y \mapsto b\} = f(a, b)$.

2.2 Fuzzy sets, proximity relations, and proximity classes

Definition 1. A *T-norm*

$$\otimes : [0, 1]^2 \to [0, 1]$$

is a binary function that satisfies the following properties for $a, b, c, d \in [0, 1]$:

- commutativity: $a \otimes b = b \otimes a$;
- monotonicity: $a \otimes b \leq c \otimes d$ if $a \leq c \wedge b \leq d$;
- associativity: $a \otimes (b \otimes c) = (a \otimes b) \otimes c$;
- neutral element 1: $a \otimes 1 = a$.

Example 3. Some of the most important T-norms include:

- Minimum T-norm: $a \otimes_M b = \min(a, b)$;
- Product T-norm: $a \otimes_P b = a * b$;
- Łukasiewicz T-norm: $a \otimes_L b = \max(0, a + b - 1)$;
- Drastic T-norm: $a \otimes_D b = \begin{cases} b, & \text{if } a = 1, \\ a, & \text{if } b = 1, \\ 0, & \text{else.} \end{cases}$

Definition 2 (Fuzzy set)**.** A *fuzzy set* is a pair (S, μ), where $S \neq \emptyset$ is a set and $\mu : S \to [0, 1]$ is a membership function.

We say that $s \in S$ *appears in* the fuzzy set (S, μ) (or (S, μ) *contains* s) if $\mu(s) > 0$. A fuzzy set is called *finite* if finitely many elements appear in it. We usually represent finite fuzzy sets explicitly as a finite set of pairs $(s, \mu(s))$, where $\mu(s) > 0$.

Definition 3 ($\otimes$-intersection of fuzzy sets)**.** Given two fuzzy sets $F_1 = (S, \mu_1)$ and $F_2 = (S, \mu_2)$ and a T-norm $\otimes$, the $\otimes$-intersection of F_1 and F_2, denoted by $F_1 \cap_\otimes F_2$, is the fuzzy set (S, μ), where the membership function μ is defined as

$$\mu(s) := \mu_1(s) \otimes \mu_2(s).$$

Definition 4 (Graded λ-cut)**.** Given a fuzzy set $F = (S, \mu)$ and $\lambda \in (0, 1]$ (called a *cut value*), the *graded λ-cut* of F, denoted $[F]_\lambda$, is the fuzzy set (S, μ'), where μ' is defined as

$$\mu(s) = \begin{cases} \mu(s), & \text{if } \mu(s) \geq \lambda, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $[F]_\lambda = \{(s, \alpha) \mid (s, \alpha) \in F,\ \alpha \geq \lambda\}$.

Definition 5 (Proximity relation)**.** A binary fuzzy relation on a set S is a mapping from $S \times S$ to the real interval $[0, 1]$.

We say that a binary fuzzy relation $\mathcal{R}$ on a set S is a *proximity relation*, if it satisfies the following properties:

Reflexivity: $\mathcal{R}(s, s) = 1$ for all $s \in S$,

Symmetry: $\mathcal{R}(s_1, s_2) = \mathcal{R}(s_2, s_1)$ for all $s_1, s_2 \in S$.

Intuitively, a proximity relation defines how close all its elements are to each other using the degree of proximity between them. The close the degree is to 1, the close the elements are to each other.

Definition 6 (Proximity class)**.** Given set S, a proximity relation $\mathcal{R}$ on S, and $\lambda \in (0, 1]$ (called the *cut value*), the $(\mathcal{R}, \lambda)$-*proximity class* of $s' \in S$ is the fuzzy set $\mathbf{pc}_{\mathcal{R},\lambda}(s') \coloneqq (S, \mu)$, where the membership function μ is defined as

$$\mu(s) \coloneqq \begin{cases} \mathcal{R}(s, s'), & \text{if } \mathcal{R}(s, s') \geq \lambda, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, $\mathbf{pc}_{\mathcal{R},\lambda}(s')$ contains only those $s \in S$, which are α-close to s' for some $\alpha \geq \lambda$.

2.3 Proximity relations over terms

We require that any proximity relation $\mathcal{R}$ defined on the alphabet $\mathcal{V} \cup \mathcal{F}$ should satisfy the following properties:

- $\mathcal{R}(f, g) = 0$ if $f \in \mathcal{F}^n$, $g \in \mathcal{F}^m$ and $n \neq m$ (i.e., f and g have different arities),
- $\mathcal{R}(x, y) = 0$ if $x, y \in \mathcal{V}$ and $x \neq y$,
- $\mathcal{R}(x, f) = 0$ if $x \in \mathcal{V}$ and $f \in \mathcal{F}$,
- for each $f \in \mathcal{F}$, the set $\{g \mid \mathcal{R}(f, g) > 0\}$ is finite.

These properties imply that

- the proximity class $\mathbf{pc}_{\mathcal{R},\lambda}(x)$ is the singleton $\{(x, 1)\}$ for any $x \in \mathcal{V}$ and a cut value λ, and
- the proximity class $\mathbf{pc}_{\mathcal{R},\lambda}(f)$ is finite for any $f \in \mathcal{F}^n$, $n \geq 0$ and a cut value λ, and contains only symbols of arity n.

Below we assume that all our proximity relations satisfy these properties.

Definition 7 (Proximity relation over terms)**.** Given a proximity relation $\mathcal{R}$ over $\mathcal{F} \cup \mathcal{V}$ and a T-norm $\otimes$, we extend $\mathcal{R}$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$ with respect to $\otimes$, defining $\mathcal{R}_{\otimes}(t, s)$ for two terms t and s as follows:

- If $t = f(t_1, \ldots, t_n)$ and $s = g(s_1, \ldots, s_n)$, $n \geq 0$, then

$$\mathcal{R}_{\otimes}(t, s) := \mathcal{R}(f, g) \otimes \mathcal{R}_{\otimes}(t_1, s_1) \otimes \cdots \otimes \mathcal{R}_{\otimes}(t_n, s_n).$$

- If $t, s \in \mathcal{V}$, then $\mathcal{R}_{\otimes}(t, s) = \mathcal{R}(t, s)$.
- Otherwise, $\mathcal{R}_{\otimes}(t, s) = 0$.

(One can easily show that $\mathcal{R}_{\otimes}$ defined for terms in such a way is reflexive and symmetric, i.e., it is indeed a proximity relation.)

Hence, to define a proximity relation over terms, it is enough to have a proximity relation over $\mathcal{F}$ and a T-norm. Below we write $\mathcal{R}$ over $\mathcal{F}$ in the form of equations $f \approx_{\delta} g$ with $\delta = \mathcal{R}(f, g) \in (0, 1]$. Also, we write $t \simeq_{\mathcal{R}_{\otimes}, \lambda} s$ to denote $\mathcal{R}_{\otimes}(t, s) \geq \lambda$.

Example 4. Let

- $\mathcal{F}^2 = \{f, g, h\}$ binary function symbols,
- $\mathcal{F}^0 = \{a, b, c, d\}$ constants,
- $\mathcal{V} = \{x, y\}$ variables.

We define approximation degrees between the function symbols:

- $\mathcal{R}(f, g) = 0.95$
- $\mathcal{R}(g, h) = 0.85$
- $\mathcal{R}(h, f) = 0.75$
- $\mathcal{R}(a, b) = 0.9$
- $\mathcal{R}(b, c) = 0.6$

- $\mathcal{R}(c, d) = 0.7$
- $\mathcal{R}(d, a) = 0.8$

Implicitly, we also have $\mathcal{R}(a, c) = \mathcal{R}(b, d) = 0$.

Then, our relation can be written as

$$\mathcal{R} = \{f \approx_{0.95} g, g \approx_{0.85} h, h \approx_{0.75} f, a \approx_{0.9} b, b \approx_{0.6} c, c \approx_{0.7} d, d \approx_{0.8} a\}.$$

With $\lambda = 0.8$, the $(\mathcal{R}, \lambda)$-proximity classes of f and h are

$$\mathbf{pc}_{\mathcal{R},\lambda}(f) = \{(f, 1), (g, 0.95)\} \qquad \mathbf{pc}_{\mathcal{R},\lambda}(h) = \{(g, 0.85), (h, 1)\}$$

Note that $(h, \mathcal{R}(f, h)) \notin \mathbf{pc}_{\mathcal{R},\lambda}(f)$ and $(h, \mathcal{R}(f, h)) \notin \mathbf{pc}_{\mathcal{R},\lambda}(h)$, since $\mathcal{R}(f, h) = 0.75 < \lambda$, and that $(f, 1) \in \mathbf{pc}_{\mathcal{R},\lambda}(f)$ and $(h, 1) \in \mathbf{pc}_{\mathcal{R},\lambda}(h)$, since due to the reflexivity property, the proximity class of an object o always contains the pair $(o, 1)$.

We have

$$\mathbf{pc}_{\mathcal{R},\lambda}(f) \cap_{\otimes} \mathbf{pc}_{\mathcal{R},\lambda}(h) = \{(g, 0.95 \otimes 0.85)\}$$

and the actual degree of g depends on the concrete T-norm.

- Let now $\otimes = \min$. Then the $(\mathcal{R}_{\otimes}, \lambda)$-proximity classes of the terms $f(a, b)$ and $h(c, d)$ are the following fuzzy sets:

$$\begin{aligned}
&\mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(f(a, b)) = \\
&\quad \{(f(a, b), 1), (f(b, b), 0.9), (f(d, b), 0.8), \\
&\quad\ (f(a, a), 0.9),\ (f(b, a), 0.9),\ (f(d, a), 0.8), \\
&\quad\ (g(a, b), 0.95),\ (g(b, b), 0.9),\ (g(d, b), 0.8), \\
&\quad\ (g(a, a), 0.9),\ (g(b, a), 0.9),\ (g(d, a), 0.8)\}.
\end{aligned}$$

$$\begin{aligned}
&\mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(h(d, b)) = \\
&\quad \{(h(d, b), 1),\ (h(a, b), 0.8),\ (h(d, a), 0.9),\ (h(a, a), 0.8), \\
&\quad\ (g(d, b), 0.85),\ (g(a, b), 0.8),\ (g(d, a), 0.85),\ (g(a, a), 0.8)\}.
\end{aligned}$$

The $\otimes$-intersection of $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a,b))$ and $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(h(d,b))$ gives

$$\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a,b)) \cap_\otimes \mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(h(d,b)) = \{g(a,b),0.8),\ (g(d,b),0.8),\ g(d,a),0.8),\ (g(a,a),0.8)\}.$$

- Now take $\otimes = *$ (product). Then the $(\mathcal{R}_\otimes,\lambda)$-proximity classes of the terms $f(a,b)$ and $h(c,d)$ are the following fuzzy sets:

$$\begin{aligned}\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a,b)) = \\ \{&(f(a,b),1),\ (f(b,b),0.9),\ (f(d,b),0.8),\\ &(f(a,a),0.9),\ (f(b,a),0.81),\\ &(g(a,b),0.95),\ (g(b,b),0.895),\\ &(g(a,a),0.895)\}.\end{aligned}$$

$$\begin{aligned}\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(h(d,b)) = \\ \{&(h(d,b),1),\ (h(a,b),0.8),\ (h(d,a),0.9),\\ &(g(d,b),0.85)\}.\end{aligned}$$

The $\otimes$-intersection of $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a,b))$ and $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(h(d,b))$ gives

$$\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a,b)) \cap_\otimes \mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(h(d,b)) = \emptyset.$$

2.4 Extended terms

Based on the assumptions on $\mathcal{R}$, the definition of proximity relations over terms, and the notion of proximity class, we can conclude that for any term t, proximity relation $\mathcal{R}$, and a cut value $\lambda \in (0,1]$, if $(s,\alpha) \in \mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(t)$, then

- $Pos(t) = Pos(s)$,
- $\mathcal{R}_\otimes(t|_p, s|_p) \geq \alpha$ for any $p \in Pos(t)$.
- if $f = t[p]$ and $g = s[p]$, then $(g,\gamma) \in \mathbf{pc}_{\mathcal{R},\lambda}(f)$ for some $\gamma \geq \alpha$.

By bold-face lowercase letters $\mathbf{f}, \mathbf{g}, \mathbf{a}, \mathbf{b}$ we denote finite fuzzy sets of function symbols of the same arity (where $\mathbf{a}$ and $\mathbf{b}$ are reserved for finite fuzzy sets of constants). We call them *extended function symbols* (resp. *extended constants*). Likewise, we use $\mathbf{x}$ and $\mathbf{y}$ to denote singleton fuzzy sets containing a variable with degree 1 and call them *extended variables*. We use $\mathit{arity}(\mathbf{f})$ to denote the arity of symbols appearing in $\mathbf{f}$. For variables, $\mathit{arity}(\mathbf{x})$ is defined as 0.

Definition 8 (Extended terms)**.** *Extended terms* are trees whose nodes are labeled by extended function symbols or variables such that the number of children for each node equals the arity of the label of the node.

The empty extended term (the empty tree) is denoted by $\perp$.

We use bold-face letters $\mathbf{t}, \mathbf{s}, \mathbf{r}$ to denote extended terms and denote by $\mathit{Pos}(\mathbf{t})$ the set of positions of $\mathbf{t}$ (as the set of positions of a tree, defined as usual).

Example 5 (Extended-term version of Example 1)**.** Let

$$\mathbf{t} = \{(f,1),(h,0.8)\}(\{(g,0.7),(h,0.7)\}(\{(a,1),(c,0.5)\}),\{(b,0.7),(d,0.6)\}).$$

Then

- $\mathbf{t}|_\epsilon = \mathbf{t}$, but $\mathbf{t}[\epsilon] = \{(f,1),(h,0.8)\}$.
- $\mathbf{t}|_1 = \{(g,0.7),(h,0.7)\}(\{(a,1),(c,0.5)\})$, but $\mathbf{t}[1] = \{(g,0.7),(h,0.7)\}$.
- $\mathbf{t}|_{1.1} = \{(a,1),(c,0.5)\}$, and $\mathbf{t}[1.1] = \{(a,1),(c,0.5)\}$.
- $\mathbf{t}|_2 = \{(b,0.7),(d,0.6)\}$, and $\mathbf{t}[2] = \{(b,0.7),(d,0.6)\}$.

Definition 9. Given $\mathcal{R}$, a cut value $\lambda \in (0,1]$, and a term t, we define an $(\mathcal{R},\lambda)$-*extension* of t as an extended term $\mathbf{ext}_{\mathcal{R},\lambda}(t)$ as follows:

$$\mathbf{ext}_{\mathcal{R},\lambda}(x) = \{(x,1)\},$$
$$\mathbf{ext}(f(t_1,\ldots,t_n)) = \mathbf{pc}_{\mathcal{R},\lambda}(f)(\mathbf{ext}_{\mathcal{R},\lambda}(t_1,\ldots,t_n)).$$

Definition 10. For an extended term $\mathbf{t}$ and a T-norm $\otimes$, we define the fuzzy set of terms $[\![\mathbf{t}]\!]_\otimes$ represented by $\mathbf{t}$ as follows:

$$
\begin{aligned}
&[\![\bot]\!]_\otimes = \emptyset,\\
&[\![\mathbf{x}]\!]_\otimes = \mathbf{x},\\
&[\![\mathbf{f}(\mathbf{t}_1, \ldots, \mathbf{t}_n)]\!]_\otimes =\\
&\quad \{(f(t_1, \ldots, t_n), \alpha \otimes \delta_1 \otimes \cdots \otimes \delta_n) \mid (f, \alpha) \in \mathbf{f}, (t_i, \delta_i) \in [\![\mathbf{t}_i]\!]_\otimes, 1 \leq i \leq n\}.
\end{aligned}
$$

The following theorem can be proved using the definitions above:

Theorem 1. For a proximity relation $\mathcal{R}$, a term t, a T-norm $\otimes$, and $\lambda \in (0, 1]$,

$$[[\![\mathbf{ext}_{\mathcal{R},\lambda}(t)]\!]_\otimes]_\lambda = \mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(t).$$

If $\otimes = \min$, then we also have

$$[\![\mathbf{ext}_{\mathcal{R},\lambda}(t)]\!]_\otimes = [[\![\mathbf{ext}_{\mathcal{R},\lambda}(t)]\!]_\otimes]_\lambda = \mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(t).$$

Since the size of $\mathbf{ext}_{\mathcal{R},\lambda}(t)$ is exponentially smaller than the size of $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(t)$, this theorem states that we can compactly represent $\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(t)$ using $\mathbf{ext}_{\mathcal{R},\lambda}(t)$.

Example 6. Consider the relation $\mathcal{R}$ from Example 4. Let again $\lambda = 0.8$, then

$$\mathbf{ext}_{\mathcal{R},\lambda}(f(a, b)) = \{(f, 1), (g, 0.95)\}(\{(a, 1), (b, 0.9), (d, 0.8)\}, \{(b, 1), (a, 0.9)\}).$$

- Take $\otimes = \min$.

$$
\begin{aligned}
&[[\![\mathbf{ext}_{\mathcal{R},\lambda}(f(a, b))]\!]_\otimes]_\lambda = [\![\mathbf{ext}_{\mathcal{R},\lambda}(f(a, b))]\!]_\otimes =\\
&\qquad \{(f(a, b), 1),\ (f(b, b), 0.9),\ (f(d, b), 0.8),\\
&\qquad\ (f(a, a), 0.9),\ (f(b, a), 0.9),\ (f(d, a), 0.8),\\
&\qquad\ (g(a, b), 0.95),\ (g(b, b), 0.9),\ (g(d, b), 0.8),\\
&\qquad\ (g(a, a), 0.9),\ (g(b, a), 0.9),\ (g(d, a), 0.8)\} =\\
&\mathbf{pc}_{\mathcal{R}_\otimes,\lambda}(f(a, b)).
\end{aligned}
$$

- Take $\otimes = *$.

$$
\begin{aligned}
&[[\![\mathbf{ext}_{\mathcal{R},\lambda}(f(a,b))]\!]_{\otimes}]_{\lambda} = \\
&\quad \{(f(a,b),1),\ (f(b,b),0.9),\ (f(d,b),0.8),\\
&\quad\ (f(a,a),0.9),\ (f(b,a),0.81),\\
&\quad\ (g(a,b),0.95),\ (g(b,b),0.895),\\
&\quad\ (g(a,a),0.895)\} = \\
&\mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(f(a,b)).
\end{aligned}
$$

Definition 11. Let S be a set of pairs consisting of a variable and an extended term, written in the form of equations:

$$S = \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}.$$

Assume that all x_j's are pairwise distinct. We call such a set a *solved set of extended equations*. For a given T-norm $\otimes$, we associate to S the set of graded substitutions

$$
\begin{aligned}
&\mathsf{Subst}_{\otimes}(S) := \\
&\{(\{x_1 \mapsto r_1, \ldots, x_m \mapsto r_m\}, \bigotimes_{j=1}^{m} \alpha_j) \mid (r_j, \alpha_j) \in [\![\mathbf{r}_j]\!]_{\otimes}, 1 \leq j \leq m\}.
\end{aligned}
$$

Definition 12 ($\otimes$-intersection of extended terms)**.** Given a T-norm $\otimes$ and two extended terms $\mathbf{t}$ and $\mathbf{s}$ such that $Pos(\mathbf{t}) = Pos(\mathbf{s})$, the $\otimes$-intersection of $\mathbf{t}$ and $\mathbf{s}$, denoted $\mathbf{t} \sqcap_{\otimes} \mathbf{s}$, is defined as follows:

- $\mathbf{x} \sqcap_{\otimes} \mathbf{x} = \mathbf{x}$,
- $\mathbf{f}(\mathbf{t}_1, \ldots, \mathbf{t}_n) \sqcap_{\otimes} \mathbf{g}(\mathbf{s}_1, \ldots, \mathbf{s}_n) = \bot$ (the empty tree) if $\mathbf{f} \cap_{\otimes} \mathbf{g} = \emptyset$ or $\mathbf{t}_i \sqcap_{\otimes} \mathbf{s}_i = \bot$ for some $1 \leq i \leq n$;

  otherwise, $\mathbf{f}(\mathbf{t}_1, \ldots, \mathbf{t}_n) \sqcap_{\otimes} \mathbf{g}(\mathbf{s}_1, \ldots, \mathbf{s}_n) = (\mathbf{f} \cap_{\otimes} \mathbf{g})(\mathbf{t}_1 \sqcap_{\otimes} \mathbf{s}_1, \ldots, \mathbf{t}_n \sqcap_{\otimes} \mathbf{s}_n)$.
- $\mathbf{t} \sqcap_{\otimes} \mathbf{s} = \bot$ in any other case.

The following theorem holds:

Theorem 2. For a proximity relation $\mathcal{R}$, terms t and s, a T-norm $\otimes$, and a cut value $\lambda \in (0, 1]$,

$$[[\![\mathbf{ext}_{\mathcal{R},\lambda}(t) \sqcap_{\otimes} \mathbf{ext}_{\mathcal{R},\lambda}(s)]\!]_{\otimes}]_{\lambda} = [\mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(t) \cap_{\otimes} \mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(s)]_{\lambda}.$$

2.5 Matching: the problem statement

Given a proximity relation $\mathcal{R}$ and a T-norm $\otimes$, a substitution σ is called an $(\mathcal{R}_{\otimes}, \lambda)$-*matcher* of a term t to a term s iff $t\sigma \simeq_{\mathcal{R}_{\otimes},\lambda} s$, i.e., iff $\mathcal{R}_{\otimes}(t\sigma, s) \geq \lambda$. The value $\mathcal{R}_{\otimes}(t\sigma, s)$ is called *the approximation degree of* the matcher σ of t to s. We also say that σ *solves* the $(\mathcal{R}_{\otimes}, \lambda)$-*matching equation* $t \preceq^{?}_{\mathcal{R}_{\otimes},\lambda} s$ (with degree $\mathcal{R}_{\otimes}(t\sigma, s)$). Below we usually skip the subscript and the question mark when we write matching equations, if the proximity relation and the cut value are obvious from the context.

Moreover, σ is called a *relevant* $(\mathcal{R}_{\otimes}, \lambda)$-matcher of t to s iff $dom(\sigma) = \mathcal{V}(t)$.

Using these definitions, a matching problem is formulated as follows:

Given: A term t, a ground term s, a proximity relation $\mathcal{R}$, a T-norm $\otimes$, and a cut value λ.

Find: A relevant $(\mathcal{R}_{\otimes}, \lambda)$-matcher of t to s and its approximation degree.

We may be interested in finding one (preferably, with the best approximation degree) or all solutions to a given matching problem.

3 The matching inference system

In this chapter, we present an algorithm that is the basis of our work: the class-based matching algorithm for proximity relations developed for the minimum T-norm in [4, 5] and state its correctness properties. In this special case, the algorithm consists of an inference system that terminates on a configuration from which we can obtain matchers without further calculations. We will then demonstrate where the key difference lies between the algorithm using the minimum T-norm and using other, arbitrary T-norms and adjust the inference system for this more general setting while still maintaining the correctness properties, which are proven in the next chapter. The algorithm is reformulated using our notation.

3.1 Matching with the minimum T-norm

In this section, we assume $t \otimes s = \min(t, s)$. Also, the proximity relation $\mathcal{R}$ and the cut value λ are assumed to be fixed.

The matching algorithm in [4, 5], here denoted by $\mathfrak{M}_{\min}$, works as an inference system on tuples of the form $M; S; \delta$, called *configurations*, where M denotes the set of $(\mathcal{R}_{\min}, \lambda)$-matching equations to be solved (slightly abusing the notions, we refer to M as a matching problem as well), S denotes the set of equations of the form $x \approx \mathbf{r}$, and δ gives the current value of the upper bound for the approximation degree of a possible matcher.

For a given problem of matching t to s (with respect to $\mathcal{R}$ and λ), the algorithm starts with the configuration $\{t \preceq s\}; \emptyset; 1$ and by applying the inference rules below, transforms configurations into configurations. When no more inference rule can be applied and the process does not fail, matchers for the initial problem can be obtained from the second

component of the final configuration, while the third component provides the upper bound for the approximation degrees of those matchers.

The matching inference system consists of the following inference rules, where $\uplus$ denotes the disjoint union:

DEC-MIN: **Decomposition**

$$\{f(t_1, \ldots, t_n) \preceq g(s_1, \ldots, s_n)\} \uplus M; S; \delta \Longrightarrow \\ M \cup \{t_i \preceq s_i \mid 1 \leq i \leq n\}; S; \min(\delta, \mathcal{R}(f, g)),$$

if $\mathcal{R}(f, g) \geq \lambda$.

CLA-MIN: **Clash**

$$\{f(t_1, \ldots, t_n) \preceq g(s_1, \ldots, s_m)\} \uplus M; S; \delta \Longrightarrow \mathsf{Fail}, \qquad \text{if } \mathcal{R}(f, g) < \lambda.$$

SOL-MIN: **Solve**

$$\{x \preceq r\} \uplus M; S; \delta \Longrightarrow M; S \cup \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(r)\}; \delta$$

MER-MIN: **Merge**

$$M; S \uplus \{x \approx \mathbf{t}, x \approx \mathbf{s}\}; \delta \Longrightarrow M; S \cup \{x \approx \mathbf{t} \sqcap_{\min} \mathbf{s}\}; \delta \qquad \text{if } \mathbf{t} \sqcap_{\min} \mathbf{s} \neq \bot.$$

INC-MIN: **Inconsistency**

$$M; S \uplus \{x \approx \mathbf{t}, x \approx \mathbf{s}\}; \delta \Longrightarrow \mathsf{Fail} \qquad \text{if } \mathbf{t} \sqcap_{\min} \mathbf{s} = \bot.$$

Final configurations of the form

$$\emptyset; S; \delta, \text{ where } S = \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}, \tag{3.1}$$

are called *successful* configurations. Note that S does not contain two equations with the same variable on the left-hand side. Therefore, $\mathsf{Subst}_{\min}(S)$ is well-defined. As it was shown in [4, 5], if $\emptyset; S; \delta$ is a successful configuration produced by $\mathfrak{M}_{\min}$ for an $(\mathcal{R}_{\min}, \lambda)$-matching problem $\{t \preceq s\}$, then $\mathsf{Subst}_{\min}(S)$ is the set of all relevant $(\mathcal{R}_{\min}, \lambda)$-matchers of t to s and their approximation degrees:

Theorem 3. Given $\mathcal{R}, \lambda$ and a matching problem $t \preceq s$, the matching algorithm $\mathfrak{M}_{\min}$ terminates and computes a set of all relevant $(\mathcal{R}_{\min})$-matchers of t to s together with their approximation degrees.

Proof. See [4] or [5]. □

Besides, for each $(\sigma, \alpha) \in \mathsf{Subst}_{\min}(S)$, we have $\alpha = \mathcal{R}(t\sigma, s)$ (i.e., $\mathcal{R}(t\sigma, s)$ is the approximation degree for σ) and $\lambda \leq \alpha \leq \delta$.

Example 7. We want to solve the matching problem $\{f(x, x) \preceq g(a, c)\}$ with the $\mathcal{R}$ from Example 4 and $\lambda = 0.52$.

$$\{f(x,x) \preceq g(a,c)\}; \emptyset; 1 \Longrightarrow_{\text{DEC-MIN}}$$
$$\{x \preceq a,\ x \preceq c\}; \emptyset; 0.95 \Longrightarrow_{\text{SOL-MIN}}$$
$$\{x \preceq c\}; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a)\}; 0.95 \Longrightarrow_{\text{SOL-MIN}}$$
$$\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ x \approx \mathbf{ext}_{\mathcal{R},\lambda}(c)\}; 0.95 \Longrightarrow_{\text{MER-MIN}}$$
$$\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_{\min} \mathbf{ext}_{\mathcal{R},\lambda}(c)\}; 0.95$$

By Theorem 2, $\mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_{\min} \mathbf{ext}_{\mathcal{R},\lambda}(c)$ is a compact representation of $\mathbf{pc}_{\mathcal{R}_{\min},\lambda}(a) \cap_{\min} \mathbf{pc}_{\mathcal{R}_{\min},\lambda}(c)$. The minimum T-norm ensures that for each element (r, α) of $\mathbf{pc}_{\mathcal{R}_{\min},\lambda}(a) \cap_{\min} \mathbf{pc}_{\mathcal{R}_{\min},\lambda}(c)$, we have $\alpha \geq \lambda$. An $(\mathcal{R}_{\min})$-matcher of $f(x, x)$ to $g(a, c)$ is now any substitution that maps x to a term r with corresponding pair $(r, \alpha) \in \mathbf{pc}_{\mathcal{R}_{\min},\lambda}(a) \cap_{\min} \mathbf{pc}_{\mathcal{R}_{\min},\lambda}(c) = \{(b, 0.6), (d, 0.7)\}$.

So $\{x \mapsto b\}$ with approximation degree $\min(0.6, 0.95) = 0.6$ is a solution as well as $\{x \mapsto d\}$ with approximation degree $\min(0.7, 0.95) = 0.7$.

3.2 Matching with arbitrary T-norms

Now we adjust the inference system for arbitrary T-norms. In fact, syntactically we only accommodate those rules where the T-norm appears explicitly:

DEC: **Decomposition**

$$\{f(t_1, \ldots, t_n) \preceq g(s_1, \ldots, s_n)\} \uplus M; S; \delta \Longrightarrow$$
$$M \cup \{t_i \preceq s_i \mid 1 \leq i \leq n\}; S; \delta \otimes \mathcal{R}(f, g)$$

if $\mathcal{R}(f, g) \geq \lambda$.

CLA: **Clash**

$$\{f(t_1, \ldots, t_n) \preceq g(s_1, \ldots, s_m)\} \uplus M; S; \delta \Longrightarrow \mathsf{Fail}, \qquad \text{if } \mathcal{R}(f, g) < \lambda.$$

SOL: **Solve**

$$\{x \preceq r\} \uplus M; S; \delta \Longrightarrow M; S \cup \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(r)\}; \delta$$

MER: **Merge**

$$M; S \uplus \{x \approx \mathbf{t},\, x \approx \mathbf{s}\}; \delta \Longrightarrow M; S \cup \{x \approx \mathbf{t} \sqcap_{\otimes} \mathbf{s}\}; \delta, \qquad \text{if } \mathbf{t} \sqcap_{\otimes} \mathbf{s} \neq \bot.$$

INC: **Inconsistency**

$$M; S \uplus \{x \approx \mathbf{t},\, x \approx \mathbf{s}\}; \delta \Longrightarrow \mathsf{Fail}, \qquad \text{if } \mathbf{t} \sqcap_{\otimes} \mathbf{s} = \bot.$$

To solve an $(\mathcal{R}_{\otimes}, \lambda)$-matching problem between terms t and s, the algorithm $\mathfrak{M}_{\otimes}$ starts with the initial configuration $\{t \preceq s\}; \emptyset; 1$ and by applying these inference rules, transforms configurations into configurations. However, in general, extracting a solution from this final configuration for $\mathfrak{M}_{\otimes}$ (if the algorithm does not fail and produces a successful configuration (3.1)) is not as straightforward as with the minimum T-norm. Even if we assume that for any $x_j \approx \mathbf{r}_j \in S$, all elements $(r_j, \alpha_j) \in [\![\mathbf{r}_j]\!]_{\otimes}$ satisfy $\alpha_j \geq \lambda$, or instead impose the constraint $(r_j, \alpha_j) \in [[\![\mathbf{r}_j]\!]_{\otimes}]_{\lambda}$, the final approximation degree of any substitution still only meets the cut value if the inequality

$$\delta \otimes \alpha_1 \otimes \cdots \otimes \alpha_m \geq \lambda, \tag{3.2}$$

holds, which is not guaranteed for $\otimes \neq \min$.

Example 8. We revisit the problem of matching $f(x, x)$ to $g(a, c)$ in Example 7 for $\lambda = 0.52$ and $\otimes = *$. The computation proceeds as in Example 7 and in the final configuration we get $\{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_{*} \mathbf{ext}_{\mathcal{R},\lambda}(c)\}; 0.95$.

The concrete value of $\mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_{*} \mathbf{ext}_{\mathcal{R},\lambda}(c)$ is $\{(b, 0.54), (d, 0.56)\}$. For the total approximation degree of a substitution, we take the computed $\delta = 0.95$ and see that for $\{x \mapsto d\}$, its approximation degree equals $0.95 * 0.56 = 0.532 \geq 0.52$, but for $\{x \mapsto d\}$, we have $0.95 * 0.54 = 0.513 < 0.52$. Thus, $\{x \mapsto b\}$ is not an $(\mathcal{R}_{\otimes}, \lambda)$-matcher of $f(x, x)$ to $g(a, c)$ for the product T-norm.

This example shows that one still has to post-process the result computed by $\mathfrak{M}_{\otimes}$ to compute real solutions. The postprocessing amounts into solving constraints between

degree expressions, which should tell us which one from the substitutions extracted from the final configuration should be actually a solution and by which degree. Namely, we should extract all substitutions $\{x_1 \mapsto r_1, \ldots, x_m \mapsto r_m\}$ with $(r_j, \alpha_j) \in [\![\mathbf{r}_j]\!]_{\otimes}$ from final configuration (3.1) such that inequality (3.2) holds. In a later chapter, such a post-processing mechanism, denoted by $\mathfrak{C}$ for constraint solving, will be developed.

4 Properties of the rule-based part

As outlined in the previous chapter, solving proximity-based matching problems with arbitrary T-norms involves two phases: the rule-based algorithm $\mathfrak{M}_{\otimes}$ and the subsequent constraint solving procedure $\mathfrak{C}$. We denote this two-phase generic algorithm by $\mathfrak{M}$. If the constraint solving part is sound, complete, and terminating, then the correctness of $\mathfrak{M}$ depends only on the correctness of the rule-based part $\mathfrak{M}_{\otimes}$. In this chapter, we concentrate on the latter.

4.1 Termination

Termination of a rule-based system can be shown by

- defining a well-founded ordering on the expressions the system operates on, and
- proving that each rule strictly decreases the ordering.

Definition 13. With

- $size(t)$: the number of symbols in a term t,
- $size(M) \coloneqq \sum_{t \preceq s \in M} (size(t) + size(s))$,
- $|S|$ is the cardinality of S, i.e. the number of equations of the form $x \approx \mathbf{r}$,

the ordering $\rhd$ for our configurations is defined as:

$$M; S; \delta \rhd M'; S'; \delta' \text{ iff } size(M) + |S| > size(M') + |S'|$$

The ordering $\rhd$ is obviously well-founded.

Theorem 4 (Termination of $\mathfrak{M}_{\otimes}$)**.** The algorithm $\mathfrak{M}_{\otimes}$ terminates on any input.

Proof. For each rule performing

$$M; S; \delta \Longrightarrow M'; S'; \delta',$$

we have

$$M; S; \delta \ \rhd \ M'; S'; \delta'$$

because

- DEC reduces the size of M without affecting S,
- SOL increases $|S|$ by one, but decreases the size of M by at least two,
- MER decreases $|S|$ without affecting M,
- CLA and INC stop the algorithm immediately.

Thus, the each rule used in $\mathfrak{M}_{\otimes}$ terminates, which implies the termination of $\mathfrak{M}_{\otimes}$. □

4.2 Soundness and completeness

First, we introduce the notion of a matcher for a configuration.

Definition 14. Given a proximity relation $\mathcal{R}$ and a cut value λ, let $M; S; \delta$ be a configuration, where

- $M = \{t_1 \preceq s_1, \ldots, t_n \preceq s_n\}$,
- $S = \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}$,
- $\delta \geq \lambda$.

Then a substitution σ is called an $(\mathcal{R}_\otimes, \lambda)$-matcher of the configuration $M; S; \delta$ iff

$$\delta \otimes \bigotimes_{i=1}^{n} \mathcal{R}_\otimes(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda, \tag{4.1}$$

where $(x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes]_\lambda$ for all $1 \leq j \leq m$.

This definition coincides for the initial step $M; \emptyset; 1$ with the definition of an $(\mathcal{R}_\otimes, \lambda)$-matcher of the original problem, and for a final configuration $\emptyset; S; \delta$ it coincides with inequality (3.2).

Lemma 1. If $M_1; S_1; \delta_1 \implies M_2; S_2; \delta_2$ is a step of the generalized matching algorithm, then σ is an $(\mathcal{R}_\otimes, \lambda)$-matcher of $M_1; S_1; \delta_1$ iff it is an $(\mathcal{R}_\otimes, \lambda)$-matcher of $M_2; S_2; \delta_2$.

Proof. A step of the algorithm is an application of one of the rules, thus we have to prove the invariance of Definition 14 for each rule individually.

- DEC: Let

  $$M_1 = \{t_1 \preceq s_1, \ldots, t_n \preceq s_n, f(t_{n+1}, \ldots, t_{n+k}) \preceq g(s_{n+1}, \ldots, s_{n+k})\},$$
  $$S_1 = \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}.$$

  Then, after the application of DEC, we get

  $$M_2 = \{t_1 \preceq s_1, \ldots, t_{n+k} \preceq s_{n+k}\},$$
  $$S_2 = S_1,$$
  $$\delta_2 = \delta_1 \otimes \mathcal{R}(f, g).$$

  By Definition 14, a substitution σ is an $(\mathcal{R}_\otimes, \lambda)$-matcher of $M_1; S_1; \delta_1$ iff

  $$\delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_\otimes(t_i\sigma, s_i) \otimes \mathcal{R}_\otimes(f(t_{n+1}, \ldots, t_{n+k})\sigma, g(s_{n+1}, \ldots, s_{n+k})) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda$$
  $$\text{with } (x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes]_\lambda$$

$$
\begin{aligned}
\text{iff}\quad & \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{i=n+1}^{n+k} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \mathcal{R}(f,g) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda \\
& \text{with } (x_j\sigma, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda} \\
\text{iff}\quad & \delta_1 \otimes \bigotimes_{i=1}^{n+k} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \mathcal{R}(f,g) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda \\
& \text{with } (x_j\sigma, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda} \\
\text{iff}\quad & \delta_2 \otimes \bigotimes_{i=1}^{n+k} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda \\
& \text{with } (x_j\sigma, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda}
\end{aligned}
$$

which is exactly inequality (4.1) for $M_2; S_2; \delta_2$.

- SOL: Let

$$
\begin{aligned}
M_1 &= \{t_1 \preceq s_1, \ldots, t_n \preceq s_n, x_{m+1} \preceq r_{m+1}\}, \\
S_1 &= \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}.
\end{aligned}
$$

Then, after the application of SOL, we get

$$
\begin{aligned}
M_2 &= \{t_1 \preceq s_1, \ldots, t_n \preceq s_n\}, \\
S_2 &= \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m, x_{m+1} \approx \mathbf{ext}_{\mathcal{R},\lambda}(r_{m+1})\}, \\
\delta_2 &= \delta_1.
\end{aligned}
$$

By Definition 14, a substitution σ is an $(\mathcal{R}_{\otimes}, \lambda)$-matcher of $M_1; S_1; \delta_1$ iff

$$
\begin{aligned}
& \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}(t_i\sigma, s_i) \otimes \mathcal{R}_{\otimes}(x_{m+1}\sigma, r_{m+1}) \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda \\
& \text{with } (x_j\sigma, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda} \\
\text{iff}\quad & \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \alpha_{m+1} \otimes \bigotimes_{j=1}^{m} \alpha_j \geq \lambda \\
& \text{with } (x_j\sigma, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda} \text{ and } \alpha_{m+1} = \mathcal{R}_{\otimes}(x_{m+1}\sigma, r_{m+1})
\end{aligned}
$$

$$\begin{aligned}
\text{iff}\quad & \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m+1} \alpha_j \geq \lambda \\
& \quad \text{with } (x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_{\otimes}]_{\lambda} \text{ and} \\
& \quad (x_{m+1}\sigma, \alpha_{m+1}) \in \mathbf{pc}_{\mathcal{R}_{\otimes},\lambda}(r_{m+1}) \\
\text{iff}\quad & \text{(by Theorem 1)} \\
& \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m+1} \alpha_j \geq \lambda \\
& \quad \text{with } (x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_{\otimes}]_{\lambda} \text{ and} \\
& \quad (x_{m+1}\sigma, \alpha_{m+1}) \in [[\![\mathbf{ext}_{\mathcal{R},\lambda}(r_{m+1})]\!]_{\otimes}]_{\lambda}
\end{aligned}$$

which is exactly inequality (4.1) for $M_2; S_2; \delta_2$.

- MER: Let

$$\begin{aligned}
M_1 &= \{t_1 \preceq s_1, \ldots, t_n \preceq s_n\}, \\
S_1 &= \{x_1 \approx \mathbf{r}_1, \ldots, x_{m-1} \approx \mathbf{r}_{m-1}, x_m \approx \mathbf{r}_{m_1},\ x_m \approx \mathbf{r}_{m_2}\}.
\end{aligned}$$

Then, after the application of MER, we get

$$\begin{aligned}
M_2 &= M_1, \\
S_2 &= \{x_1 \approx \mathbf{r}_1, \ldots, x_{m-1} \approx \mathbf{r}_{m-1},\ x_m \approx \mathbf{r}_{m_1} \sqcap_{\otimes} \mathbf{r}_{m_2}\}, \\
\delta_2 &= \delta_1.
\end{aligned}$$

By Definition 14, a substitution σ is an $(\mathcal{R}_{\otimes}, \lambda)$-matcher of $M_1; S_1; \delta_1$ iff

$$\begin{aligned}
& \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m-1} \alpha_j \otimes \alpha_{m_1} \otimes \alpha_{m_2} \geq \lambda \\
& \quad \text{with } (x_m\sigma, \alpha_{m_1}) \in [[\![\mathbf{r}_{m_1}]\!]_{\otimes}]_{\lambda}, (x_m\sigma, \alpha_{m_2}) \in [[\![\mathbf{r}_{m_2}]\!]_{\otimes}]_{\lambda}, \\
& \quad \text{and } (x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_{\otimes}]_{\lambda} \\
\text{iff}\quad & \text{(by Definition 12, Theorem 2, and Definition 3)} \\
& \delta_1 \otimes \bigotimes_{i=1}^{n} \mathcal{R}_{\otimes}(t_i\sigma, s_i) \otimes \bigotimes_{j=1}^{m-1} \alpha_j \otimes (\alpha_{m_1} \otimes \alpha_{m_2}) \geq \lambda
\end{aligned}$$

with $(x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes]_\lambda$ and
$(x_m\sigma, \alpha_{m_1} \otimes \alpha_{m_2}) \in [[\![\mathbf{r}_{m_1} \sqcap_\otimes \mathbf{r}_{m_2}]\!]_\otimes]_\lambda$,

which is exactly inequality (4.1) for $M_2; S_2; \delta_2$.

- The conditions of CLA and INC rule out the existence of an $(\mathcal{R}_\otimes, \lambda)$-matcher for the matching problem.

□

Theorem 5 (Soundness and Completeness of $\mathfrak{M}_\otimes$)**.** Let $\mathcal{R}$ be a proximity relation, λ be a cut value, $\otimes$ be a T-norm, and $t \preceq s$ be a matching problem.

- Soundness: If $\mathfrak{M}_\otimes$ produces a derivation $\{t \preceq s\}; \emptyset; 1 \Longrightarrow^+ \emptyset; S; \delta$, then for each $(\sigma, \alpha) \in \mathsf{Subst}_\otimes(S)$ with $\alpha \otimes \delta \geq \lambda$, the substitution σ is a relevant $(\mathcal{R}_\otimes, \lambda)$-matcher of $t \preceq s$ with the approximation degree $\alpha \otimes \delta$.
- Completeness: If σ is a relevant $(\mathcal{R}_\otimes, \lambda)$-matcher of $t \preceq s$ with the approximation degree β, then $\mathfrak{M}_\otimes$ produces a derivation $\{t \preceq s\}; \emptyset; 1 \Longrightarrow^+ \emptyset; S; \delta$ such that $(\sigma, \alpha) \in \mathsf{Subst}_\otimes(S)$ for some α and $\beta = \delta \otimes \alpha$.

Proof. The core of the proof of both properties is induction on Lemma 1.

- Soundness: If the algorithm $\mathfrak{M}_\otimes$ produces the output Fail, the statement trivially holds.

  Let now $\{t \preceq s\}; \emptyset; 1 \Longrightarrow^+ \emptyset; S; \delta$ be the derivation in $\mathfrak{M}_\otimes$ with a successful final configuration, where S is of the form $\{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}$. Let $(\sigma, \alpha) \in \mathsf{Subst}_\otimes(S)$ and $\alpha = \bigotimes_{j=1}^m \alpha_j$ such that $(x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes$ for all $1 \leq j \leq m$.

  Inequality $\alpha \otimes \delta \geq \lambda$ together with the monotonicity of the T-norm implies that we also have $(x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes]_\lambda$, hence σ is an $(\mathcal{R}_\otimes, \lambda)$-matcher of the final configuration by Definition 14. By induction on the length of the derivation, Lemma 1, it is also an $(\mathcal{R}_\otimes, \lambda)$-matcher of t to s.

  All variables in S come from the original problem, so σ is also a relevant matcher of t to s.

- Completeness: Let σ be a relevant $(\mathcal{R}_\otimes, \lambda)$-matcher of $t \preceq s$ with the approximation degree β. Since this matching problem is solvable, we would never apply CLA or INC, and thus arrive at a successful configuration $\emptyset; S; \delta$ with S of the form $\{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}$.

  Lemma 1 implies that σ is an $(\mathcal{R}_\otimes, \lambda)$-matcher of this final configuration. By Definition 14, we have $\alpha_1, \ldots, \alpha_m$ with $(x_j\sigma, \alpha_j) \in [[\![\mathbf{r}_j]\!]_\otimes]_\lambda$ for all $1 \leq j \leq m$ and $\bigotimes_{j=1}^m \alpha_j \otimes \delta = \beta$. Because $[[\![\mathbf{r}_j]\!]_\otimes]_\lambda \subset [\![\mathbf{r}_j]\!]_\otimes$, we have $(x_j\sigma, \alpha_j) \in [\![\mathbf{r}_j]\!]_\otimes$, and finally $(\sigma, \bigotimes_{j=1}^m \alpha_j) \in \mathsf{Subst}_\otimes(S)$.

□

5 Constraint solving

The second phase of the generic proximity-based algorithm $\mathfrak{M}$ concerns extracting matching substitutions from the final configuration computed in the first phase by the rule-based algorithm $\mathfrak{M}_{\otimes}$. Not all substitutions represented by such final configurations are valid matchers (unlike the $\mathfrak{M}_{\text{min}}$ case): only those that satisfy the conditions of Definition 14 are. In this chapter, we outline challenges and methods of extracting such substitutions from the computed final configurations. As we said earlier, this phase of the algorithm is referred to by $\mathfrak{C}$.

5.1 Motivating examples

If a derivation using the rules from $\mathfrak{M}_{\otimes}$ does not fail, it has the form

$$\{t \preceq s\}; \emptyset; 1 \Longrightarrow^{*} \emptyset; \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}; \delta.$$

Now, our task is to find those substitutions $\{x_1 \mapsto r_1, \ldots, x_m \mapsto r_m\}$, where $(r_j, \alpha_j) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda}$ and $\delta \otimes \alpha_1 \otimes \cdots \otimes \alpha_m \geq \lambda$ for the given λ.

A straightforward way would be to express this problem in the form of the following constraint, formulated as a formula over inequalities and equalities where $\chi_1, \ldots, \chi_m$ are degree variables, and then solve it for χ's and x's:

$$\left(\delta \otimes \bigotimes_{j=1}^{m} \chi_j \geq \lambda\right) \wedge \left(\bigwedge_{j=1}^{m} \bigvee_{(r,\alpha) \in [[\mathbf{r}_j]]_{\otimes}]_{\lambda}} (\chi_j = \alpha \wedge x_j\sigma = r)\right). \tag{5.1}$$

Its solution would provide a substitution σ with the domain $x_1, \ldots, x_m$ together with its degree as the value γ of $\delta \otimes \bigotimes_{j=1}^{m} \chi_j$. Then σ would be an $(\mathcal{R}_\otimes, \lambda)$-matcher of t to s with degree γ.

Example 9 (Continuation of Example 8)**.** From final configuration $\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_\otimes \mathbf{ext}_{\mathcal{R},\lambda}(c)\}; 0.95$ with $[[\![\mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_\otimes \mathbf{ext}_{\mathcal{R},\lambda}(c)]\!]_\otimes]_\lambda = \{(b, 0.54), (d, 0.56)\}$, we see that the approximation degree α for a substitution of x can only be $0.95 \otimes 0.54$ or $0.95 \otimes 0.56$. Using formula (5.1), we get:

$$
\begin{aligned}
&0.95 * \chi \geq 0.52 \wedge ((\chi = 0.54 \wedge x\sigma = b) \vee (\chi = 0.56 \wedge x\sigma = d)) \equiv \\
&(0.95 * \chi \geq 0.52 \wedge \chi = 0.54 \wedge x\sigma = b) \vee \\
&\qquad (0.95 * \chi \geq 0.52 \wedge \chi = 0.56 \wedge x\sigma = d) \equiv \\
&(0.95 * \chi \geq 0.52 \wedge \chi = 0.56 \wedge x\sigma = d) \equiv \\
&\chi = 0.56 \wedge x\sigma = d.
\end{aligned}
$$

Hence, from the obtained solution to the constraint, we conclude that the substitution $\sigma = \{x \mapsto d\}$ is a solution of the original problem with degree $0.95 * \chi = 0.95 * 0.56 = 0.532$.

However, the problem with this approach is that the constraints are formulated on $[\![\mathbf{r}]\!]_\otimes$'s, which are proximity classes of terms: fuzzy sets listed explicitly. They can be exponentially large compared to $\mathbf{r}$.

Example 10. Consider the matching problem $\{f(x, x) \preceq g(f(a, b), h(c, d))\}$. Similar to Example 8, the inference system terminates on configuration

$$\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(f(a, b)) \sqcap_\otimes \mathbf{ext}_{\mathcal{R},\lambda}(h(c, d))\}; \mathcal{R}(f, g).$$

The constraint solving via formula (5.1) requires the computation of the sets

$$
\begin{aligned}
&[[\![\mathbf{ext}_{\mathcal{R},\lambda}(f(a, b))]\!]_\otimes]_\lambda, \\
&[[\![\mathbf{ext}_{\mathcal{R},\lambda}(h(c, d))]\!]_\otimes]_\lambda,
\end{aligned}
$$

and their $\otimes$-intersection. With the relation $\mathcal{R}$ from Example 4, $\lambda = 0.28$ and $\otimes = *$, the first expression, for instance, results in

$$
\begin{aligned}
&[[\![\mathbf{ext}_{\mathcal{R},\lambda}(f(a,b))]\!]_{\otimes}]_{\lambda} = \\
&\quad \{(f(a,b),1),\ (f(b,b),0.9),\ (f(d,b),0.8),\\
&\quad\ (f(a,c),0.6),\ (f(b,c),0.54),\ (f(d,c),0.48),\\
&\quad\ (f(a,a),0.9),\ (f(b,a),0.81),\ (f(d,a),0.72),\\
&\quad\ (g(a,b),0.95),\ (g(b,b),0.855),\ (g(d,b),0.76),\\
&\quad\ (g(a,c),0.57),\ (g(b,c),0.513),\ (g(d,c),0.456),\\
&\quad\ (g(a,a),0.855),\ (g(b,a),0.7695),\ (g(d,a),0.684)\\
&\quad\ (h(a,b),0.75),\ (h(b,b),0.675),\ (h(d,b),0.6),\\
&\quad\ (h(a,c),0.45),\ (h(b,c),0.405),\ (h(d,c),0.36),\\
&\quad\ (h(a,a),0.675),\ (h(b,a),0.6075),\ (h(d,a),0.54)\},
\end{aligned}
$$

which is much bigger in size than its compact counterpart

$$
\begin{aligned}
&\mathbf{ext}_{\mathcal{R},\lambda}(f(a,b)) = \\
&\quad \{(f,1),(g,0.95),(h,0.75)\}\\
&\qquad (\{(a,1),(b,0.9),(d,0.8)\},\{(b,1),(c,0.6),(d,0.7)\}).
\end{aligned}
$$

This motivates attempting to work directly with $\mathbf{r}$'s instead of $[\![\mathbf{r}]\!]_{\otimes}$'s, an approach which is discussed in the next section.

5.2 Compact constraints

Using compact representation, we can solve degree variable constraints position-wise.

Consider again a final configuration $\emptyset; S; \delta$, where $S = \{x_1 \approx \mathbf{r}_1, \ldots, x_m \approx \mathbf{r}_m\}$.

We rewrite formula (5.1) as

$$\begin{aligned}&\left(\delta \otimes \bigotimes_{j=1}^{m} \bigotimes_{p \in Pos(\mathbf{r}_j)} \chi_{j,p} \geq \lambda\right) \wedge \\ &\left(\bigwedge_{j=1}^{m} \bigwedge_{p \in Pos(\mathbf{r_j})} \bigvee_{(r,\alpha) \in \mathbf{r}_j[p]} \left(\chi_{j,p} = \alpha \wedge (x_j\sigma)[p] = r\right)\right),\end{aligned} \tag{5.2}$$

where $\chi_{j,p}$ and σ are the variables we solve for.

Example 11 (Compact representation of Example 10)**.** Recall: $\otimes = *$, $\lambda = 0.28$ and

$$\mathcal{R} = \{f \approx_{0.95} g, g \approx_{0.85} h, h \approx_{0.75} f, a \approx_{0.9} b, b \approx_{0.6} c, c \approx_{0.7} d, d \approx_{0.8} a\}.$$

Applying the rules of $\mathfrak{M}_{\otimes}$:

$$\begin{aligned}&\{f(x,x) \preceq g(f(a,b), h(c,d))\}; \emptyset; 1 \Longrightarrow^{+} \\ &\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(f(a,b)) \sqcap_{\otimes} \mathbf{ext}_{\mathcal{R},\lambda}(h(c,d))\}; 0.95,\end{aligned}$$

where

$$\begin{aligned}&\mathbf{ext}_{\mathcal{R},\lambda}(f(a,b)) \sqcap_{\otimes} \mathbf{ext}_{\mathcal{R},\lambda}(h(c,d)) = \\ &\{(f, 1 * 0.75), (g, 0.0.95 * 0.85), (h, 0.75 * 1)\}(\\ &\quad \{(b, 0.9 * 0.6), (d, 0.8 * 0.7)\}, \{(a, 0.9 * 0.8), (c, 0.6 * 0.7)\} \\ &\quad).\end{aligned}$$

Using formula (5.2), the constraints amount in

$$\begin{aligned}&\Big(0.95 * \chi_\epsilon * \chi_1 * \chi_2 \geq 0.3\Big) \wedge \\ &\Big((\chi_\epsilon = 0.75 \wedge (x\sigma)[\epsilon] = f) \vee (\chi_\epsilon = 0.8075 \wedge (x\sigma)[\epsilon] = g) \vee \\ &\quad (\chi_\epsilon = 0.75 \wedge (x\sigma)[\epsilon] = h)\Big) \wedge \\ &\Big((\chi_1 = 0.54 \wedge (x\sigma)[1] = b) \vee (\chi_1 = 0.56 \wedge (x\sigma)[1] = d)\Big) \wedge \\ &\Big((\chi_2 = 0.72 \wedge (x\sigma)[2] = a) \vee (\chi_2 = 0.42 \wedge (x\sigma)[2] = c)\Big).\end{aligned}$$

This system is equivalent to the DNF

$$\begin{aligned}
&\Big(\chi_\epsilon = 0.8075 \wedge (x\sigma)[\epsilon] = g \wedge \chi_1 = 0.56 \wedge (x\sigma)[1] = d \wedge \chi_2 = 0.72 \wedge (x\sigma)[2] = a\Big) \\
&\quad \vee \\
&\Big(\chi_\epsilon = 0.8075 \wedge (x\sigma)[\epsilon] = g \wedge \chi_1 = 0.54 \wedge (x\sigma)[1] = b \wedge \chi_2 = 0.72 \wedge (x\sigma)[2] = a\Big) \\
&\quad \vee \\
&\Big(\chi_\epsilon = 0.75 \wedge (x\sigma)[\epsilon] = f \wedge \chi_1 = 0.56 \wedge (x\sigma)[1] = d \wedge \chi_2 = 0.72 \wedge (x\sigma)[2] = a\Big) \\
&\quad \vee \\
&\Big(\chi_\epsilon = 0.75 \wedge (x\sigma)[\epsilon] = h \wedge \chi_1 = 0.56 \wedge (x\sigma)[1] = d \wedge \chi_2 = 0.72 \wedge (x\sigma)[2] = a\Big),
\end{aligned}$$

where each disjunct corresponds to one solution of the matching problem with approximation degree $\chi_\epsilon * \chi_1 * \chi_2 * \delta$. Hence, we get as the set of graded substitutions:

$$\begin{aligned}
&\Big\{(\{x \mapsto g(d,a)\}, 0.3093048), \\
&\ (\{x \mapsto g(b,a)\}, 0.2982582), \\
&\ (\{x \mapsto f(d,a)\}, 0.28728), \\
&\ (\{x \mapsto h(d,a)\}, 0.28728)\Big\}.
\end{aligned}$$

Remark 1. There would be not much benefit in using the compact representation if during constraint solving we have to always expand it again to the explicit form. The expansion to the DNF was done here to illustrate a manual extraction of the set of solutions. In general, providing a practically efficient systematic method of solving such constraints goes beyond the scope of this thesis.

Remark 2. If we are only interested in the solution with the highest approximation degree, we do not have to solve the whole constraint completely. Selecting the pair with the biggest number from each position for each variable returns the best solution candidate due to the monotonicity of the T-norm, as long as the approximation degree of this candidate guarantees that it is actually a solution. In Example 11 above, this would correspond to selecting

$$\{(0.8075, g)\}(\{(0.56, d)\}, \{(0.72, a)\})$$

as the candidate for instantiation x. Moreover, if this is not a solution, then the problem is not solvable. Hence, this procedure serves as a satisfiability check. Similarly, we can check whether the constraint is valid if selecting the smallest pair from each position for each variable gives a solution.

5.3 An illustrative example

We present a final example with more than one variable.

Example 12. Let

- $f \in \mathcal{F}^4$,
- $g, h, k \in \mathcal{F}^1$,
- $a, b, c, d \in \mathcal{F}^0$,
- $x, y \in \mathcal{V}$.

We define a proximity relation on this alphabet as

$$\begin{aligned}\mathcal{R} = \{&g \approx_{0.95} h,\ h \approx_{0.92} k,\ k \approx_{0.6} g,\\ &a \approx_{0.98} d,\ a \approx_{0.87} b,\ a \approx_{0.69} c,\\ &b \approx_{0.93} d,\ c \approx_{0.75} d,\ b \approx_{0.88} c\}.\end{aligned}$$

Let $\lambda = 0.4$. We solve the matching problem

$$\{f(x, y, g(y), k(x)) \preceq f(a, g(c), g(k(a)), h(b))\}$$

with the Łukasiewicz T-norm $a \otimes_L b = \max(0, a + b - 1)$ using the algorithm $\mathfrak{M}$.

Phase 1: using the algorithm $\mathfrak{M}_\otimes$. The derivation proceeds as follows:

$$\begin{aligned}&\{f(x, y, g(y), k(x)) \preceq f(a, g(c), g(k(a)), h(b))\}; \emptyset; 1 \Longrightarrow_{\text{DEC}}\\ &\{x \preceq a,\ y \preceq g(c), g(y) \preceq g(k(a)),\ k(x) \preceq h(b)\}; \emptyset; 1 \Longrightarrow_{\text{SOL}}\\ &\{y \preceq g(c),\ g(y) \preceq g(k(a)),\ k(x) \preceq h(b)\};\ \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a)\}; 1 \Longrightarrow_{\text{SOL}}\end{aligned}$$

$\{g(y) \preceq g(k(a)),\ k(x) \preceq h(b)\}; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c))\}; 1 \Longrightarrow_{\text{DEC}}$
$\{y \preceq k(a),\ k(x) \preceq h(b)\}; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c))\}; 1 \Longrightarrow_{\text{SOL}}$
$\{k(x) \preceq h(b)\}; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c)),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(k(a))\}; 1 \Longrightarrow_{\text{DEC}}$
$\{x \preceq b\}; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c)),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(k(a))\}; 0.92 \Longrightarrow_{\text{SOL}}$
$\emptyset; \{x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c)),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(k(a)),\ x \approx \mathbf{ext}_{\mathcal{R},\lambda}(b)\}; 0.92.$

At this stage, we perform two steps using the MER rule.

First, merging $x \approx \mathbf{ext}_{\mathcal{R},\lambda}(a)$ and $x \approx \mathbf{ext}_{\mathcal{R},\lambda}(b)$: Since

$\mathbf{ext}_{\mathcal{R},\lambda}(a) = \{(a, 1), (b, 0.87), (c, 0.69), (d, 0.98)\}$ and
$\mathbf{ext}_{\mathcal{R},\lambda}(b) = \{(a, 0.87), (b, 1), (c, 0.88), (d, 0.93)\},$

we have

$$\mathbf{ext}_{\mathcal{R},\lambda}(a) \sqcap_{\otimes_L} \mathbf{ext}_{\mathcal{R},\lambda}(b) = \{(a, 0.87), (b, 0.87), (c, 0.57), (d, 0.89)\},$$

which gives the configuration

$$\emptyset; \{x \approx \{(a, 0.87), (b, 0.87), (c, 0.57), (d, 0.89)\},\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c)),\ y \approx \mathbf{ext}_{\mathcal{R},\lambda}(k(a))\}; 0.92$$

Next, merging $y \approx \mathbf{ext}_{\mathcal{R},\lambda}(g(c))$ and $y \approx \mathbf{ext}_{\mathcal{R},\lambda}(k(a))$: Since

$\mathbf{ext}_{\mathcal{R},\lambda}(g(c)) =$
$\{(g, 1), (h, 0.95), (k, 0.6)\}\Big(\{(a, 0.69), (b, 0.88), (c, 1), (d, 0.75)\}\Big)$ and
$\mathbf{ext}_{\mathcal{R},\lambda}(k(a)) =$
$\{(g, 0.6), (h, 0.92), (k, 1)\}\Big(\{(a, 1), (b, 0.87), (c, 0.69), (d, 0.98)\}\Big),$

we have

$\mathbf{ext}_{\mathcal{R},\lambda}(g(c)) \sqcap_{\otimes_L} \mathbf{ext}_{\mathcal{R},\lambda}(k(a)) =$
$\{(g, 0.6), (h, 0.87), (k, 0.6)\}\Big(\{(a, 0.69), (b, 0.75), (c, 0.69), (d, 0.73)\}\Big),$

which gives the final configuration

$$\emptyset; \Big\{ x \approx \{(a, 0.87), (b, 0.87), (c, 0.57), (d, 0.89)\}, \\ y \approx \{(g, 0.6), (h, 0.87), (k, 0.6)\} \\ \Big(\{(a, 0.69), (b, 0.75), (c, 0.69), (d, 0.73)\}\Big) \Big\};\ 0.92.$$

Phase 2: using $\mathfrak{C}$. Now, to compute matchers, from this configuration, using formula (5.2), we generate the constraint below and try to solve it:

$$\begin{aligned} &\Big(0.92 \otimes_L \chi_x \otimes_L \chi_{y,\epsilon} \otimes_L \chi_{y,1} \geq 0.4\Big) \wedge \\ &\Big((\chi_x = 0.87 \wedge x\sigma[\epsilon] = a) \vee (\chi_x = 0.87 \wedge x\sigma[\epsilon] = b) \vee \\ &\ (\chi_x = 0.57 \wedge x\sigma[\epsilon] = c) \vee (\chi_x = 0.89 \wedge x\sigma[\epsilon] = d)\Big) \wedge \\ &\Big((\chi_{y,\epsilon} = 0.6 \wedge (y\sigma)[\epsilon] = g) \vee (\chi_{y,\epsilon} = 0.87 \wedge (y\sigma)[\epsilon] = h) \vee \\ &\ (\chi_{y,\epsilon} = 0.6 \wedge (y\sigma)[\epsilon] = k)\Big) \wedge \\ &\Big((\chi_{y,1} = 0.69 \wedge (y\sigma)[1] = a) \vee (\chi_{y,1} = 0.75 \wedge (y\sigma)[1] = b) \vee \\ &\ (\chi_{y,1} = 0.69 \wedge (y\sigma)[1] = c) \vee (\chi_{y,1} = 0.73 \wedge (y\sigma)[1] = d)\Big). \end{aligned} \tag{5.3}$$

As we mentioned above, we do not aim at providing a practical solving method for such constraints, but it is pretty straightforward to check whether it is solvable. For this, in each equational conjunct of the formula we keep only one disjunct with the maximal number, which gives a simpler constraint

$$\begin{aligned} &\Big(0.92 \otimes_L \chi_x \otimes_L \chi_{y,\epsilon} \otimes_L \chi_{y,1} \geq 0.4\Big) \wedge \\ &\Big(\chi_x = 0.89 \wedge x\sigma[\epsilon] = d\Big) \wedge \\ &\Big(\chi_{y,\epsilon} = 0.87 \wedge (y\sigma)[\epsilon] = h\Big) \wedge \\ &\Big(\chi_{y,1} = 0.75 \wedge (y\sigma)[1] = b\Big). \end{aligned} \tag{5.4}$$

Now, we have to check whether $\chi_x = 0.89$, $\chi_{y,\epsilon} = 0.87$, and $\chi_{y,1} = 0.75$ satisfy the inequality $0.92 \otimes_L \chi_x \otimes_L \chi_{y,\epsilon} \otimes_L \chi_{y,1} \geq 0.4$. For the Łukasiewicz T-norm we have

$0.92 \otimes_L 0.89 \otimes_L 0.87 \otimes_L 0.75 = 0.43 \geq 0.4$, meaning that the inequality is satisfied. Therefore, those values for χ's and the substitution

$$\sigma = \{x \mapsto d,\ y \mapsto h(b)\}$$

give the solution of constraint (5.4). This substitution is a solution of the $(\mathcal{R}_{\otimes_L}, \lambda)$-matching problem $\{f(x, y, g(y), k(x)) \preceq f(a, g(c), g(k(a)), h(b))\}$ with degree 0.43, which is also the maximal possible approximation degree among all the solutions of this problem.

It is also not difficult to check whether constraint (5.3) is valid, i.e., if all the substitutions induced by it are solutions of the original matching problem. For this, we select a subconstraint in a similar way as we did above for constraint (5.4), with the difference that instead of maximal number equations we choose those with minimal numbers (nondeterministically choosing one when there are several with the same number). It gives the constraint

$$\begin{aligned}
&\Big(0.92 \otimes_L \chi_x \otimes_L \chi_{y,\epsilon} \otimes_L \chi_{y,1} \geq 0.4\Big) \wedge \\
&\Big(\chi_x = 0.57 \wedge x\sigma[\epsilon] = c\Big) \wedge \\
&\Big(\chi_{y,\epsilon} = 0.6 \wedge (y\sigma)[\epsilon] = g\Big) \wedge \\
&\Big(\chi_{y,1} = 0.69 \wedge (y\sigma)[1] = a\Big).
\end{aligned} \tag{5.5}$$

However, this is not solvable, because $0.92 \otimes_L 0.57 \otimes_L 0.6 \otimes_L 0.69 = 0$ and the inequality is not satisfied. It means that not all substitutions induced by the final configuration above are solutions. To find out which ones are, we need to solve constraint (5.3) completely, which gives rise to the following substitutions and the corresponding approximation degrees:

$$\begin{aligned}
\Big\{&(\{x \mapsto d, y \mapsto h(b)\}, 0.89 \otimes_L 0.62 \otimes_L 0.92 = 0.43), \\
&(\{x \mapsto d, y \mapsto h(d)\}, 0.89 \otimes_L 0.60 \otimes_L 0.92 = 0.41), \\
&(\{x \mapsto a, y \mapsto h(b)\}, 0.87 \otimes_L 0.62 \otimes_L 0.92 = 0.41), \\
&(\{x \mapsto b, y \mapsto h(b)\}, 0.87 \otimes_L 0.62 \otimes_L 0.92 = 0.41)\Big\}.
\end{aligned}$$

6 Further outlook and concluding remarks

6.1 Early failure detection

In Example 11, we see that the inequality $0.95 * \chi_\epsilon * \chi_1 * \chi_2 \geq 0.3$ is never satisfied with the instantiation $\chi_2 = 0.42$. This begs the question if there was a point during the application of the inference rules where this could have been detected.

One such point is after the application of the MER rule of two extended terms $\mathbf{t}$ and $\mathbf{s}$, where elements $(r, \alpha) \in (\mathbf{t} \sqcap_\otimes \mathbf{s})[p]$ for some position p with $\alpha < \lambda$, or even $\alpha \otimes \delta < \lambda$, are omitted from $(\mathbf{t} \sqcap_\otimes \mathbf{s})[p]$. In general, one could check all $(x \approx \mathbf{r}) \in S$ after applying DEC if there is an $(r, \alpha) \in \mathbf{r}[p]$ for some position p which does not meet $\alpha \otimes \delta \geq \lambda$ anymore.

Thinning out the set S in the final configuration $\emptyset; S; \delta$ requires more computation, but it speeds up the constraint solving part and vice versa. There may be an optimal balance between these two sides.

6.2 Fully fuzzy signatures

In this thesis, we defined the approximation degree between function symbols of different arities to be 0. Such signatures are called basic fuzzy signatures, see [5].

Proximity relations can be extended to function symbols of different arities, in what is called a fully fuzzy signature. This can be done by so-called *argument relations*. Solving equations in fully fuzzy signatures has been addressed in [6] (for similarity relations), [7] (for proximity relations based on proximity blocks), and in [8] (for proximity relations based on proximity classes). The algorithms in these papers use the minimum T-norm. Since our work is based on proximity classes, we expect that the matching algorithm from

[8] can be extended to generic T-norms similarly to how we extended the class-based proximity matching algorithm for basic fuzzy signatures ([4]) from the minimum T-norm to arbitrary ones.

6.3 Conclusion

In this thesis, we extended a class-based matching algorithm for proximity relations that only uses the minimum T-norm to accommodate for arbitrary T-norms.

The existing algorithm works as an inference system and terminates on set of constraints where the set of solutions is straightforward to extract. We generalized these inference rules for arbitrary T-norms and proved its correctness properties under these new conditions.

We showed that if the system terminates on a successful final configuration with a set of proximity constraints, the set of solutions is not straightforward to extract, but requires a step for solving the satisfiability problem for such constraints.

An important feature is the ability to work with extended terms as compact representations of proximity classes and their intersections. Such a representation makes it extremely easy to check the satisfiability of proximity constraints and extract a matcher with the best approximation degree.

Bibliography

[1] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998. ISBN: 978-0-521-45520-6 (cit. on pp. 1, 2, 3).

[2] Maria I. Sessa. "Approximate reasoning by similarity-based SLD resolution". In: *Theor. Comput. Sci.* 275.1-2 (2002), pp. 389–426. DOI: 10.1016/S0304-3975(01)00188-8 (cit. on p. 2).

[3] Pascual Julián Iranzo and Clemente Rubio-Manzano. "Proximity-based unification theory". In: *Fuzzy Sets Syst.* 262 (2015), pp. 21–43. DOI: 10.1016/J.FSS.2014.07.006 (cit. on pp. 2, 3).

[4] Temur Kutsia and Cleo Pau. "Matching and Generalization Modulo Proximity and Tolerance Relations". In: *Language, Logic, and Computation - 13th International Tbilisi Symposium, TbiLLC 2019, Batumi, Georgia, September 16-20, 2019, Revised Selected Papers*. Ed. by Aybüke Özgün and Yulia Zinova. Vol. 13206. Lecture Notes in Computer Science. Springer, 2019, pp. 323–342. DOI: 10.1007/978-3-030-98479-3_16 (cit. on pp. 2, 3, 14, 15, 16, 36).

[5] Cleo Pau. "Symbolic Techniques for Approximate Reasoning". PhD thesis. RISC, Johannes Kepler University Linz, 2022 (cit. on pp. 3, 14, 15, 16, 35).

[6] Hassan Aït-Kaci and Gabriella Pasi. "Fuzzy lattice operations on first-order terms over signatures with similar constructors: A constraint-based approach". In: *Fuzzy Sets Syst.* 391 (2020), pp. 1–46. DOI: 10.1016/J.FSS.2019.03.019 (cit. on p. 35).

[7] Maria Eugenia Cornejo, Jesús Medina-Moreno, and Clemente Rubio-Manzano. "Towards a Full Fuzzy Unification in the Bousi Prolog system". In: *2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2018, Rio de Janeiro, Brazil, July 8-13, 2018*. IEEE, 2018, pp. 1–7. DOI: 10.1109/FUZZ-IEEE.2018.8491514 (cit. on p. 35).

[8] Cleo Pau and Temur Kutsia. “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*. IEEE, 2021, pp. 1–6. DOI: 10.1109/FUZZ45933.2021.9494438 (cit. on pp. 35, 36).