Solving Proximity Constraints

Sophie Hofmanninger & Jan-Michael Holzinger

26.06.2019

Johannes Kepler University, A-4040 Linz, Austria

Motivation

For proving theorems, a frequently occurring problem is to find common instances of formulae.

Example 1

Let f be a function, a, b constants and x a variable. The two expressions

$$f(a, x) \quad \text{and} \quad f(a, b)$$

can be unified with $\{x \mapsto b\}$.

Motivation

For proving theorems, a frequently occurring problem is to find common instances of formulae.

Example 2

Let f, g be functions, a, b constants and x a variable. The two expressions

$$f(a, x) \quad \text{and} \quad g(a, b)$$

cannot be unified as $f \neq g$.

Motivation

In 1965 Robinson presented his unification algorithm and solved this problem, his algorithm was improved for better(=faster) performance since.

If we consider now the unification problem

$$f(a, x) \simeq^? g(a, b)$$

again, we might wonder, if we could not ignore $f \neq g$, if they are "close" to each other, i.e. if they are equal in a fuzzy logic sense. Being close is represented as a proximity relation, which are symmetric and reflexive, but not necessarily transitive. C. Pau and T. Kutsia solved this problem, presenting an algorithm, which we implemented.

Introduction

The Algorithm consists of two sub-algorithms and works on (modifies) 4 sets:

- P: unification problem to be solved ,
- C: neighbourhood constraint,
- σ: set of pre-unifier,
- Φ: name-class mapping,

where Algorithm 1 modifies P, C, and σ and Algorithm 2 modifies C and Φ. If Algorithm 1 was successful, $P = \emptyset$, if Algorithm 2 was successful $C = \emptyset$.

pre-Unification rules

(Tri) $\{x \simeq^? x\} \uplus P; C; \sigma \Rightarrow P; C; \sigma$

(Dec)
$\{F(\overline{s_n}) \simeq^? G(\overline{t_n})\} \uplus P; C; \sigma \Rightarrow \{\overline{s_n \simeq^? t_n}\} \cup P; \{F \approx^? G\} \cup C; \sigma$

(VE) $\{x \simeq^? t\} \uplus P; C; \sigma \Rightarrow \{t' \simeq^? t\} \cup Px \mapsto t'; C; \sigma\{x \mapsto t'\}$

(Ori) $\{t \simeq^? x\} \uplus P; C; \sigma \Rightarrow \{x \simeq^? t\} \cup P; C; \sigma$

(Cla) $\{F(\overline{s_n}) \simeq^? G(\overline{t_n})\} \uplus P; C; \sigma \Rightarrow \bot$ if $m \neq n$

(Occ)
$\{x \simeq^? t\} \uplus P; C; \sigma \Rightarrow \bot$ if there is an occurrence cycle of x in t

(VO)
$\{x \simeq^? y, \overline{x_n \simeq^? y_n}\}; C; \sigma \Rightarrow \{\overline{x_n \simeq^? y_n}\}\{x \mapsto y\}; C; \sigma\{x \mapsto y\}$

Rules for Neighbourhood Constraints

(FFS) $\{f \approx^? g\} \uplus C; \Phi \Rightarrow C; \Phi;$ if $\mathcal{R}(f, g) \geq \lambda$

(NFS) $\{N \approx^? g\} \uplus C; \Phi \Rightarrow C; update(\Phi, N \rightarrow \mathbf{pc}(g, \mathcal{R}, \lambda))$

(FSN) $\{g \approx^? N\} \uplus C; \Phi \Rightarrow \{N \approx^? g\} \cup C; \Phi$

(NN1)
$\{N \approx^? M\} \uplus C; \Phi \Rightarrow C; update(\Phi, N \rightarrow \{f\}, M \rightarrow \mathbf{pc}(f, \mathcal{R}, \lambda))$,
where $N \in dom(\Phi)$, $f \in \Phi(N)$

(NN2) $\{M \approx^? N\} \uplus C; \Phi \Rightarrow \{N \approx^? M\} \cup C; \Phi$, where
$M \notin dom(\Phi)$, $N \in dom(\Phi)$

(Fail1) $\{f \approx^? g\} \uplus C; \Phi \Rightarrow \perp$, if $\mathcal{R}(f, g) < \lambda$

(Fail2) $C; \Phi \Rightarrow \perp$, if there exists $N \in dom(\Phi)$ such that $\Phi(N) = \emptyset$

Simple example about how both algorithms work

Example 3

Let p, q and f be functions, b, c, c' constants and x, z variables. Then the following unification problem has a solution:

$$p(x, z) =^{?} q(f(b), f(x)) \quad \text{with} \quad R = \{(b, c'), (c', c), (p, q)\}$$

Simple example - pre-Unification fails

Example [Fail pU]

Examples where the pre-Unification algorithm fails:

$$\text{(Occ)} \quad p(x) =^? q(f(x)) \tag{1}$$

$$\text{(Cla)} \quad p(a, b) =^? q(f(x)) \tag{2}$$

Simple example - Constraint Simplification fails

Example [Fail CS]

Let p and q be functions, a, b, c constants and x, y variables. Then for the following unification problem only the pre-Unification algorithm is successful:

$$p(a, x, a) =^? q(y, b, x) \quad \text{with} \quad R = \{(b, c), (p, q)\}$$

Simple example cont.

pre-Unification

...

$C = \{p \approx^? q, N_1 \approx^? a, N_2 \approx^? b, a \approx^? N_2\}$

Constraint Simplification

$C = \{p \approx^? q, N_1 \approx^? a, N_2 \approx^? b, a \approx^? N_2\}$

$\Phi = \{\}$

$\Rightarrow^{\text{FFS}}$

$C = \{N_1 \approx^? a, N_2 \approx^? b, a \approx^? N_2\}$

$\Phi = \{\}$

$\Rightarrow^{\text{NFS}^2}$

Simple example cont.

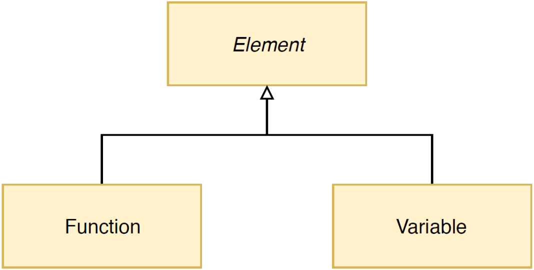
$$C = \{a \approx^? N_2\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \{b, c\}\}$$
$$\Rightarrow^{\text{FSN}}$$
$$C = \{N_2 \approx^? a\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \{b, c\}\}$$
$$\Rightarrow^{\text{NFS}}$$
$$C = \{\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \emptyset\}$$
$$\Rightarrow^{\text{Fail2}}$$
$$\perp$$

System Model

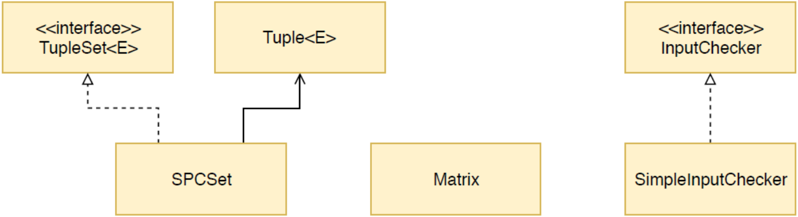
Project consists of 4 packages:

- elements : contains all needed types
- tool : offers important tools (e.g. store proximity relations)
- unificationProblem : has the core features
- userInterfaces : provide user interfaces

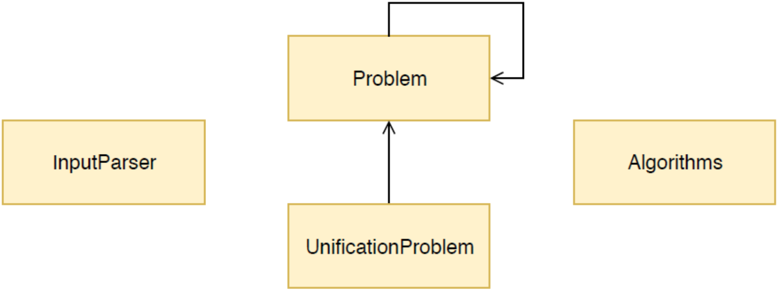
Package elements



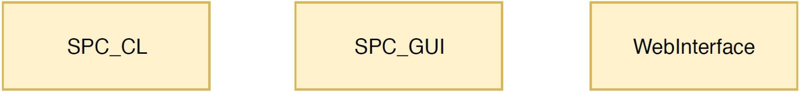
Package tool



Package unificationProblem



Package `userInterfaces`



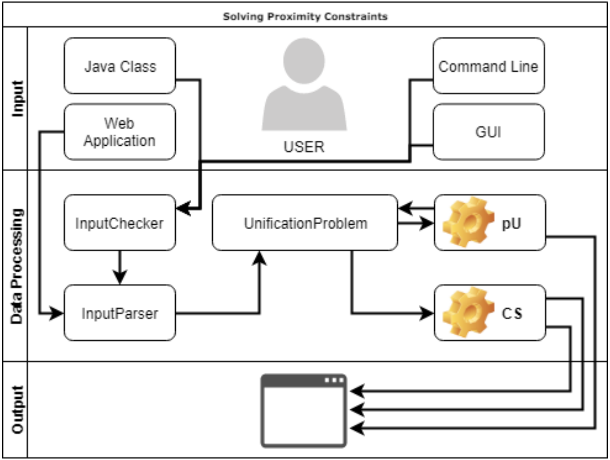
1 Introduction

2 System Model

3 Workflow

4 Usage and Experience with the presented Tools

Workflow


Solving Proximity Constraints
Input
Data Processing
Output
Java Class
Web Application
USER
Command Line
GUI
InputChecker
UnificationProblem
pU
InputParser
CS

Redmine/UML

- Redmine - useful feature
- Communication:
  - Redmine forum
  - Whatsapp
  - meetings before the lectures
- UML
  - used it from the beginning
  - to express and communicate our ideas

Git/Javadoc

- Git
  - own Git repository for the project
  - merged branches
  - commited continuously
- Javadoc
  - used it from the beginning
  - displaying it as a tooltip

JUnit/Jenkins

- JUnit 5
- good to find bugs
- not easy to call from the CMD/Jenkins
- JUnit 5 - strict naming of test classes