JOHANNES KEPLER
UNIVERSITÄT LINZ

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

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

(NN1)
$\{N \approx^? M\} \uplus C; \Phi \Rightarrow C; update(\Phi, N \to f, M \to \mathbf{pc}(f, \mathcal{R}, \lambda)),$
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 - both algorithms work

Example

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 - PU fails

Examples where the Pre Unification algorithm fails:

(Occ) $p(x) =^? q(f(x))$

(Cla) $p(a, b) =^? q(f(x))$

Simple example - Constrains Simplification fails

Example

Let p and f be functions, a, b 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

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

$\Phi = \{\}$

$\Rightarrow^{\mathsf{FFS}}$

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

$\Phi = \{\}$

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

Simple example cont.

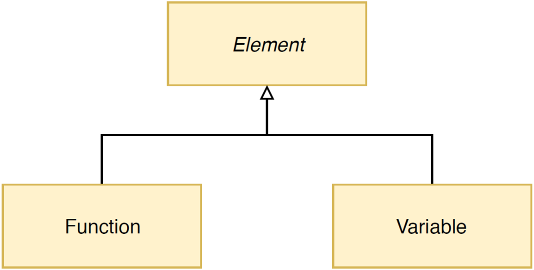
$$C = \{a \approx^? N_2\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \{b, c\}\}$$
$$\Rightarrow^{\text{FSN}}$$
$$C = \{N_2 \approx^? a\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \{b, c\}\}$$
$$\Rightarrow^{\text{NFS}}$$
$$C = \{\}$$
$$\Phi = \{N_1 \mapsto \{a\}, N_2 \mapsto \{b, c\}, N_2 \mapsto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

System Model

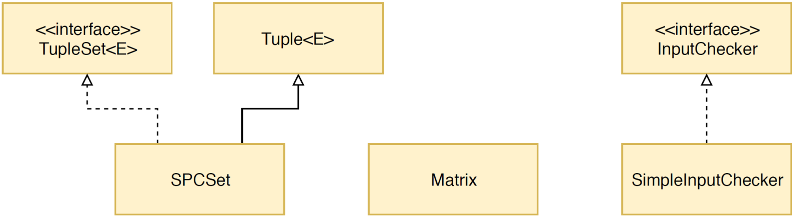
Project consists of 4 packages:

- elements : deals with elements
- tool : offers important tools (e.g. read input)
- unificationProblem : treats the unification problem
- userInterfaces : allow user interfaces

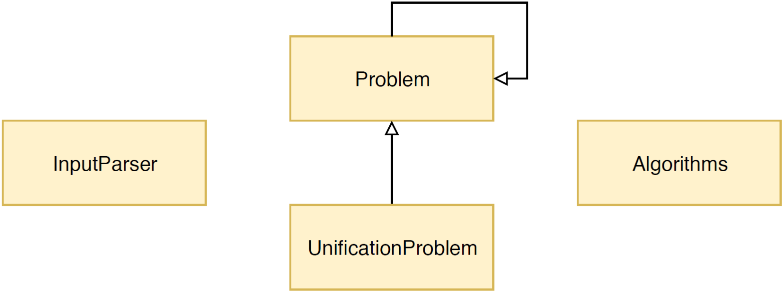
Package elements



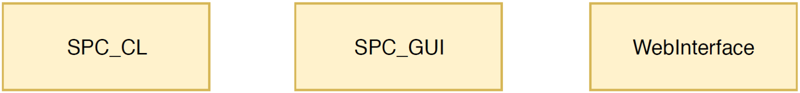
Package tool



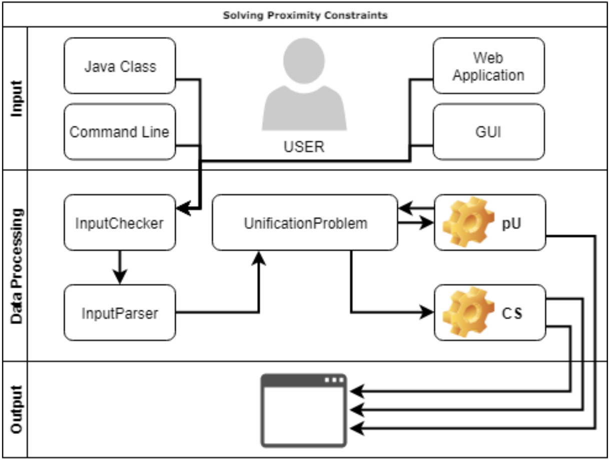
Package `unificationProblem`



Package userInterfaces



Workflow


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