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 $C = \emptyset$.

JOHANNES KEPLER UNIVERSITÄT LINZ

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

$$p(x, z) =^? q(f(b), f(x)); R = \{(a, a'), (a', b), (b, c'), (c', c), (p, q)\}$$
$$P = \{p \simeq^? q\}$$
$$C = \{\}$$
$$\sigma = \{\}$$
$$\Rightarrow^{\text{Dec}}$$
$$P = \{x \simeq^? f, z \simeq^? f\}$$
$$C = \{p \approx^? q\}$$
$$\sigma = \{\}$$
$$\Rightarrow^{\text{VE}}$$
$$P = \{N_1(N_2) \simeq^? f(b), z \simeq^? f(N_1(N_2))\}$$
$$C = \{p \approx^? q\}$$
$$\sigma = \{x \mapsto N_1(N_2)\}$$
$$\Rightarrow^{\text{Dec}^2}$$

Simple example cont.

$$p(x,z) =^? q(f(b), f(x)); R = \{(a,a'), (a',b), (b,c'), (c',c), (p,q)\}$$
$$P = \{z \simeq^? f\}$$
$$C = \{p \approx^? q, N_1 \approx^? f, N_2 \approx^? b\}$$
$$\sigma = \{x \mapsto N_1(N_2)\}$$
$$\Rightarrow^{\text{VE}}$$
$$P = \{N_3(N_4(N_5)) \simeq^? f(N_1(N_2))\}$$
$$C = \{p \approx^? q, N_1 \approx^? f, N_2 \approx^? b\}$$
$$\sigma = \{x \mapsto N_1(N_2), z \mapsto N_3(N_4(N_5))\}$$
$$\Rightarrow^{\text{Dec}^3}$$
$$P = \{\}$$
$$C = \{p \approx^? q, N_1 \approx^? f, N_2 \approx^? b, N_3 \approx^? f, N_4 \approx^? N_1, N_5 \approx^? N_2\}$$
$$\sigma = \{x \mapsto N_1(N_2), z \mapsto N_3(N_4(N_5))\}$$

Other examples

Simple examples where the pre-unification fails:

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

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

Test 4 in Test.java

Pre - ok, CS - fail: - des ging bei mir, i hätt 5 Schritte und 2 Branches, die aber failen-

$p(x, y, x) =^? q(f(a), g(b), y); R = \{(b, c), (c, d), (f, g), (p, q)\}$

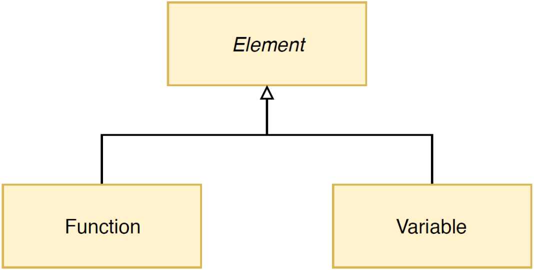
Solution: ...

System Model

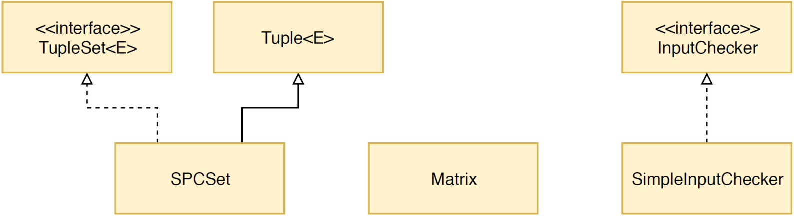
Project consists of 4 packages:

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

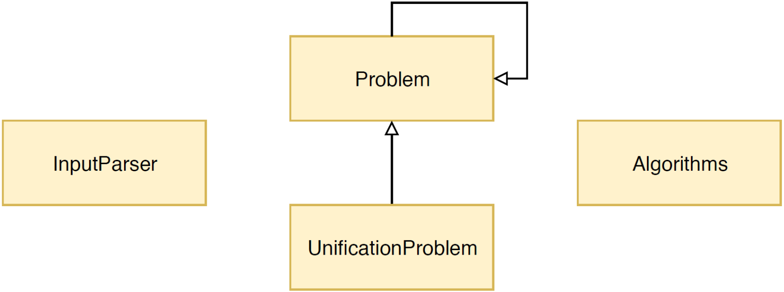
Package elements



Package tool



Package `unificationProblem`



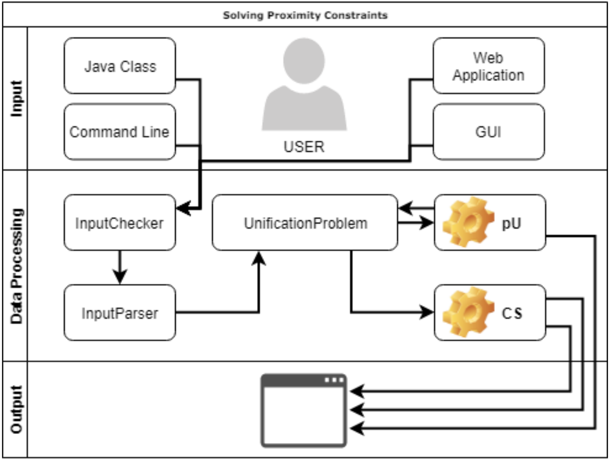
Package `userInterfaces`

1 Introduction

2 System Model

3 Workflow

Workflow


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