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

4 sets:

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

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 occurence 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 Neigborhood Constraints

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

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

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

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

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

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

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

Simple example

Pre - fail: $p(x, a) =?q(f(a), g(b))$

Test 4 in Test.java
Pre - ok, CS - fail: - des ging bei mir, i hätt 5 Schritte und 2 Branches, die aber failen-
$p(x, y, x) =?q(f(a), g(b), y); R = \{(b, c), (c, d), (f, g), (p, q)\}$

Test 3 in Test.java
Pre - ok, CS - ok: - des ging bei mir, i hätt 4 Schritte und 1 Branch-
$p(x, z) =?q(f(b), f(x)); R = \{(a, a'), (a', b), (b, c'), (c', c), (p, q)\}$

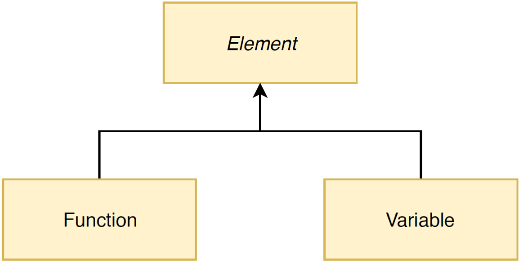
Solution: ...

System Model

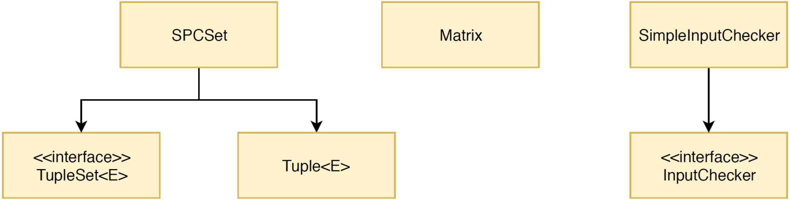
Project consists of 4 packages:

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

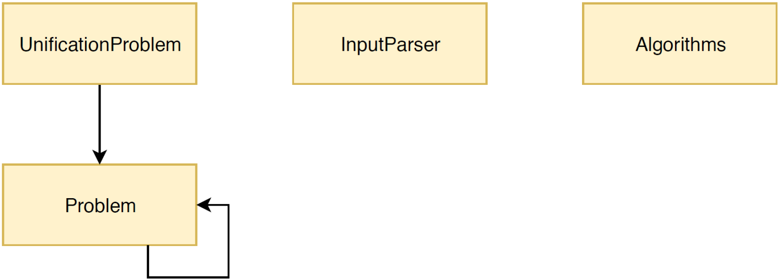
Package elements



Package tool

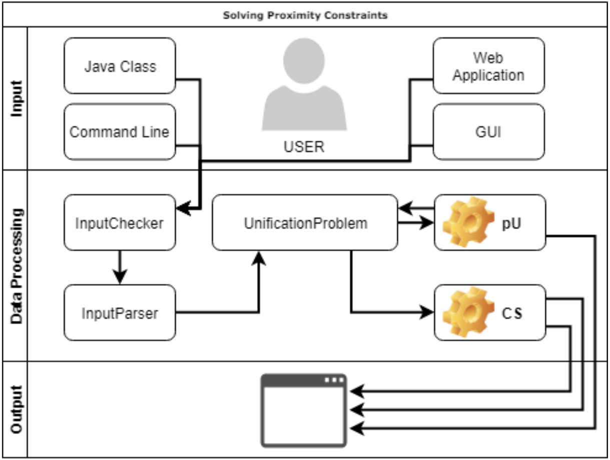


Package `unificationProblem`



Package `userInterfaces`

Workflow


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