Unification de termes schématisés à exposants entiers

Mathieu Guillame-Bert, Gabriel Synnaeve

Plan

Plan

I. Présentation du sujet :

1. Unification logique
2. Termes schématisés
3. Cahier des charges

Plan

I. Présentation du sujet :
   1. Unification logique
   2. Termes schématisés
   3. Cahier des charges

II. El Unificator :
   1. Implémentation
   2. Architecture
   3. Démonstrations

Plan

I. Présentation du sujet :
   1. Unification logique
   2. Termes schématisés
   3. Cahier des charges

II. El Unificator :
   1. Implémentation
   2. Architecture
   3. Démonstrations

III. Bilan :
   1. Résultats
   2. Travail de recherche
   3. Bilan moral

Unification logique

- Def. : Deux termes $\mathbf{t_1}$ et $\mathbf{t_2}$: trouver la substitution $\boldsymbol{\sigma}$ (un **u**nificateur le **p**lus **g**énéral)
  - $\boldsymbol{\sigma}(\mathbf{t_1}) = \boldsymbol{\sigma}(\mathbf{t_2})$ (même notation que l'égalité)
  - $\boldsymbol{\sigma}(\mathbf{t_1}) = \mathbf{t_2}$, si seul t_1 contient des variables
- "Résolution syntaxique d'équation $\subset$ Unification"
- L'unification "simple" (variables, fonctions d'arité fixe, constantes) est décidable.

Exemples

Exemples

- f(X, a) = f(b, Y)

Exemples

- $f(X, a) = f(b, Y)$
  - ✓ $X = b \wedge Y = a$

Exemples

- $f(X, a) = f(b, Y)$
  - ✓ $X = b \wedge Y = a$
- $f(X, a) = g(b, Y)$

Exemples

- $f(X, a) = f(b, Y)$
  - ✓ $X = b \wedge Y = a$
- $f(X, a) = g(b, Y)$
  - ✦ impossible : $f = g$?

Exemples

- $f(X, a) = f(b, Y)$
  - ✓ $X = b \wedge Y = a$
- $f(X, a) = g(b, Y)$
  - ✦ impossible : $f = g$?
- $f(X, Z) = f(f(Y), h(X))$

Exemples

- $f(X, a) = f(b, Y)$
  - ✓ $X = b \wedge Y = a$
- $f(X, a) = g(b, Y)$
  - ✦ impossible : $f = g$?
- $f(X, Z) = f(f(Y), h(X))$
  - ✓ $X = f(Y) \wedge Z = h(X) \wedge Y$ libre

Termes Schématisés

- Un moyen d'augmenter le pouvoir d'expression
- $(\forall x)(P(x) \Rightarrow P(f(x))) \wedge P(a)$ engendre :

  $P(f(a)), P(f(f(a))), P(f(f(f(a))))$ ...
- "méta-terme" : $f(\diamond)^N \cdot a$ (avec N variable)
- prop. : $u(\diamond)^N \cdot v = u(\diamond)^{N-1} \cdot (u(\diamond)^1 \cdot v)$
- Exposants naturels non-nuls

Interprétation

Interprétation

$$f(\diamond,a)^2 \cdot X =$$

Interprétation

$$f(\diamond,a)^2 \cdot X =$$

$$f(f(\diamond,a),a)^1 \cdot X =$$

Interprétation

$$f(\diamond,a)^2 \cdot X =$$

$$f(f(\diamond,a),a)^1 \cdot X =$$

$$f(f(f(\diamond,a),a),a)^0 \cdot X =$$

Interprétation

$$f(\diamond,a)^2 \cdot X =$$

$$f(f(\diamond,a),a)^1 \cdot X =$$

$$f(f(f(\diamond,a),a),a)^0 \cdot X =$$

$$f(f(f(X,a),a),a)$$

Exemples

Exemples

- $f(\diamond)^{N1} \cdot Y = f(\diamond)^{N2} \cdot Y$

Exemples

- $f(\diamond)^{N1} \cdot Y = f(\diamond)^{N2} \cdot Y$
  - ✓ $N1 = N2 \wedge Y$ libre

Exemples

- $f(\Diamond)^{N1} \cdot Y = f(\Diamond)^{N2} \cdot Y$

  ✓ $N1 = N2 \wedge Y$ libre

- $f(\Diamond)^{N} \cdot a = f(f(\Diamond))^{M} \cdot a$

Exemples

- $f(\diamond)^{N1} \cdot Y = f(\diamond)^{N2} \cdot Y$
  - ✓ $N1 = N2 \wedge Y$ libre
- $f(\diamond)^{N} \cdot a = f(f(\diamond))^{M} \cdot a$
  - ✓ $N = 2*M$

Exemples

- $f(\diamond)^{N1} \cdot Y = f(\diamond)^{N2} \cdot Y$
  - ✓ $N1 = N2 \wedge Y$ libre
- $f(\diamond)^{N} \cdot a = f(f(\diamond))^{M} \cdot a$
  - ✓ $N = 2*M$
- $f(g(f(\diamond))^{N} \cdot g(a)) = f(g(f(g(\diamond))))^{M} \cdot a$

Exemples

- $f(\diamond)^{N1} \cdot Y = f(\diamond)^{N2} \cdot Y$
  - ✓ $N1 = N2 \wedge Y$ libre
- $f(\diamond)^{N} \cdot a = f(f(\diamond))^{M} \cdot a$
  - ✓ $N = 2*M$
- $f(g(f(\diamond))^{N} \cdot g(a)) = f(g(f(g(\diamond))))^{M} \cdot a$
  - ✓ $N = M = 1$
    $\vee\ N = 2*H + 1 \wedge M = H + 1 \wedge H$ libre

Arithmétique de Presburger

- Pour la résolution des contraintes sur les variables numériques
- 1er ordre, dans le langage de l'arithmétique de Peano mais sans la *multiplication* ie. : $\{\mathbb{N}, \mathbf{+}, \mathbf{0}, succ()\}$ ➡ **Complète et décidable**
- Construction d'automates pour résoudre des équations linéaires, exemple : $aX + bY = c$

Travail d'application

- Basé principalement sur 2 papiers :
  - "On unification of terms with integer exponents" d'Hubert Comon, 1993
  - "On the construction of Automata from Linear Arithmetic Constraints" de Pierre Wolper et Bernard Boigelot, 2000

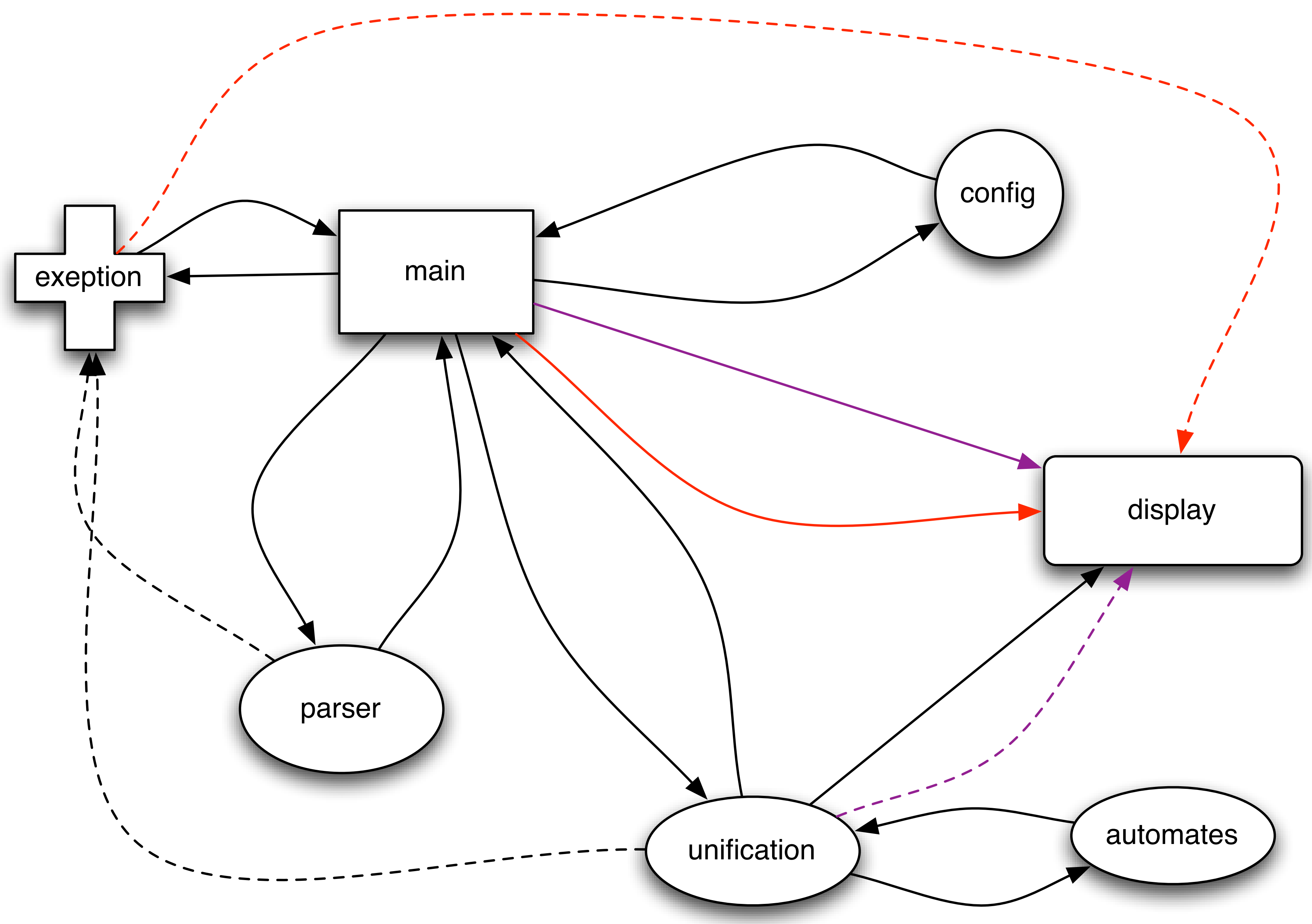
Cahier des charges

- Implémenter l'algorithme d'unification d'Hubert Comon
- Vérifier la validité des solutions par rapport aux contraintes arithmétiques sur les variables numériques
- Architecture propice à l'évolution

Implémentation

- C++
- Facilement interfaçable (programme & librairie)
- Fonctions et système d'option standard (testé sous WindowS, GNU/Linux, Mac OS X)
- Librairie de parsing : Flex / Bison
- Structure des équations en arbre
- Automates pour Presburger en interne

Architecture



Déroulement de l'unification

- Ensemble de règles de réécriture
- But : retrouver une forme du type

$$\exists \vec{n}.N_1 = E_1 \wedge \cdots \wedge N_k = E_k \wedge x_1 = t_1 \wedge \cdots \wedge x_n = t_n$$

- Exemple de règle

  Idée : décomposer en sous cas

$$s = t[\diamond]_p^N.u \rightarrow (N = 1 \wedge s = t[u]_p) \vee (\exists M.N = M + 1 \wedge s = t[t^M.u]_p)$$

Déroulement de l'unification

- Complexité variable
- Exemple (Decompose 2) :

$$s[(v_1[w_1[\diamond]_p]_{q'})^{N_1}.u_1]_p = (v_2[w_2[\diamond]_{q'}]_p)^{N_2}.u_2 \rightarrow$$

$$\begin{aligned}
&\mathrm{s[a]}_p = w_1[a]_p \wedge s[a]_p = v_2[a]_p \wedge v_1[a]_{q'} = w_2[a]_{q'} \\
&\wedge((N_1 = N_2 \wedge s[u_1]_p = u_2) \\
&\vee(\exists M_1.N_1 = N_2 + M_1 \wedge s[(v_1[w_1[\diamond]_p]_{q'})^{M_1}.u_1]_p = u_2) \\
&\vee(\exists M_2.N_2 = N_1 + M_2 \wedge w_1[u_1]_p = (v_2[w_2[\diamond]_{q'}]_p)^{M_2}.u_2) \\
&)
\end{aligned}$$

Démonstration**S**

DémonstrationS

- Unification simple

Démonstration**S**

- Unification simple
- Unification avec termes à exposants entiers

Démonstration**S**

- Unification simple
- Unification avec termes à exposants entiers
- Filtrage des solutions avec les automates

Démonstration**S**

- Unification simple
- Unification avec termes à exposants entiers
- Filtrage des solutions avec les automates
- Exemple plus conséquent

Résultats

- Implémentation fonctionnelle et complète
- + option *-latex* qui sort du latex

1. Tests automatisés de non régression
2. Code documenté (doxygen + UML)

   ➡ Reprise extérieure facile

- Extension possible : plusieurs “holes” ($\diamond$)

Travail de recherche

- Recherche d’un algorithme d’existence de solutions et de réécriture pour un sous-ensemble de l’arithmétique de Presburger, afin de résoudre les contraintes (1 système d’équations diophantines par solution)
- Échec

Bilan moral

- Sujet intéressant
- Cadre de travail motivant

➡ Beaucoup de travail fourni dans la bonne humeur

- Initiation à un travail de recherche

Merci de votre attention