Certification d'un algorithme d'élimination des quantificateurs sur $\mathbb{R}$

Assia Mahboubi

INRIA Sophia Antipolis

Manipulation d'expressions symboliques

Deux approches :

Logiciels de Calcul Formel	*Assistants à la preuve*
communauté vaste et diverse	preuves formelles
simplicité d'utilisation	sémantique non ambigüe
optimisation, efficacité	expressivité

Certification d'algorithmes

Différentes approches possibles

- Choix d' un assistant à la preuve :
  - formalisme logique sous-jacent (Coq, HOL, PVS,...)
  - degré d'automatisation nécessaire
  - ...
- Choix d' une méthode :
  - Conception d'un nouveau logiciel ce calcul formel (projet Foc)
  - Interfaçage (ex : avec Maple)
  - Programmation d'une nouvelle librairie pour l'assistant

Dans ce qui va suivre ...

- Choix d' un assistant à la preuve :
  - formalisme logique sous-jacent (Coq,HOL, PVS ...)
  - degré d'automatisation nécessaire
  - . . .
- Choix d' une méthode :
  - Conception dun nouveau logiciel ce calcul formel (projet Foc)
  - Interfaçage (ex : avec Maple)
  - Programmation d'une nouvelle librairie pour l'assistant

Calcul des Constructions (Inductives)

Caractéristiques du formalisme :

- Théorie des types
- Extension de la logique d'orde supérieur
- Imprédicative
- Basé sur l'isomorphisme de Curry-Howard

Curry-Howard

Rappels :

- preuves = objets
- propositions = types
- une preuve de la proposition A est un terme a de type A
- une une preuve de la proposition $A \Rightarrow B$ est une application de type $A \rightarrow B$
- ...

Conséquences :

- Homogénéité entre les preuves et les objets
- Articulation calcul/déduction

$\hookrightarrow$ facilité et clarté dans l'implantation

Automatisation en Coq

Dans Coq :

- Enoncer un théorème = declarer un type bien former
- Le prouver = Construire de façon interactive un terme de ce type
- Outils d'aide à la construction = Tactiques

Nécéssité absolue de développer outils d'automatisation!

Ex : preuves par réécritures fastidieuses ...

Élimination des quantificateurs dans $\mathbb{R}$

Problème équivalent :

- Soit $f_1 \dots f_s$ des polynômes de $\mathbb{Z}[X_1 \dots X_n]$
- Soit $\sharp_1 \dots \sharp_n$ des conditions de signes :
  $\forall 1 \leq i \leq s, \quad \sharp_i \in \{<, >, =\}$

On peut déterminer (algorithmiquement) si le système :

$$\begin{cases} f_1(x_1, \dots, x_n) \ \sharp_1 0 \\ \vdots \\ f_s(x_1, \dots, x_n) \ \sharp_n 0 \end{cases}$$

a une solution dans $\mathbb{R}^n$.

Tarski(1951)

But

Supposons d'abord $f_1, \ldots, f_s$ sont dans $\mathbb{Z}[X]$:

- Construire le tableau de signe exhaustif de la famille de départ $(f_1, \ldots, f_s)$:
  - «trouver» l'ensemble des racines $x_1, \ldots x_m$ des polynômes de la famille : partition adaptée de l'espace
  - caculer les signes de chaque f_i sur chacun de ces intervalles
- Parcourir les colonnes du tableau pour trouver celle correspondant à la condition recherchée

Méthode de Hörmander(1990)

$$f = (f_1, \ldots, f_s) \xrightarrow{TRANSFORMATION} f' = (f_1, \ldots, f_{s-1}, f'_s, g_1, \ldots, g_s)$$

$$\begin{array}{ccc} \left[\begin{array}{c} \\ ? \\ \\ \end{array}\right] & \xleftarrow{REMONTEE\ SIGNES} & \left[\begin{array}{c} signes \\ de \\ f' \end{array}\right] \\ w & & w' \end{array}$$

Transformation/Remontée

- Transformation :
  - par rapport à $f_s \neq 0$
  - f'_s est le polynôme dérivé de f_s
  - pour $i = 1 \ldots s-1$, g_i est le reste positif de la division euclidienne de f_s par f_i
  - g_s est le reste positif de la division euclidienne de f_s par f'_s
- Remontée :
  - Théorème des valeurs intermédiaires
  - Nombre fini de racines pour un polynôme
  - Arguments de monotonie

Exemple de calcul

$F_1 = \{X^2 - 1\}$

$\downarrow$

$F_2 = \{X, -2\}$

$\downarrow$

$F_3 = \{X\}$

$\downarrow$

$F_4 = \{1, 0\}$ $\longrightarrow$

	$]-\infty; y_1[$	$[y_1]$	$]y_1; y_2[$	$[y_2]$	$]y_2; +\infty[$
X	$+$	0	$-$	0	$+$

$\uparrow$

	$]-\infty; y_1[$	$[y_1]$	$]y_1; +\infty[$
X	$-$	0	$+$
-2	$-$	$-$	$-$

$\uparrow$

	$]-\infty; y_1[$	$[y_1]$	$]y_1; +\infty[$
X	$-$	0	$+$

$\uparrow$

	$]-\infty; +\infty[$
1	$+$
0	0

En plusieurs indéterminées

Le cas général : $f_1 \dots f_s \in \mathbb{Z}[X_1, \dots, X_n]$

- On considère $f_1 \dots f_s \in \mathbb{Z}[X_1, \dots, X_{n-1}][X_n]$
- On élimine les variables successivement
- On partitionne $\mathbb{R}^n$ en cellules, partition adaptée a la famille $f_1 \dots f_s$
- $\hookrightarrow$ calcul d' une liste de tableaux au lieu d'un tableau
- $\hookrightarrow$ coefficients polynômiaux $\Rightarrow$ nouvelles difficultés . . .

Pseudo Divisions

f, g sont dans $R[X_1] \ldots R[X_n]$.
a est le coefficient dominant de g.
r est le reste positif de la pseudo-division de f par g si :

$$cf = sg + r$$

avec :

- $deg(r) < deg(g)$
- $c > 0$ (ne prend que des valeurs strictement positives)
- c divise une puissance de a

Détecter la possible nullité de polyômes

... est la principale source de difficulté :

- Application de la transformation seulement si f_s est non constant
- Propriétés de signe de pseudo-restes ($c > 0$)

Conséquence :

- Maintenir des conditions de validité de la correspondance tableau/famille
- $\hookrightarrow$ Explosion combinatoire pour garder un calcul exhaustif

Nouvelle transformation

La transformation est maintenant :

$$\mathcal{F} = (f_1, \ldots, f_s),\ \mathcal{Z},\ \mathcal{NZ}$$

$$\mathcal{F}' = \{(f_1, \ldots, f_{s-1}, f'_s, g_1, \ldots, g_s)_i,\ \mathcal{Z}_i,\ \mathcal{NZ}_i \mid i \in I\}$$

Et à chaque étape on manipule des listes de tableaux

Certification de l'algorithme

- Programation en Coq :
  - lemmes sur les fonctions polynômiales sur $\mathbb{R}$
  - fonction de «remontée» des tableaux de signes
  - preuve de correction de cette fonction générée simultanément
- En OCaml
  - opérations sur les polynômes
  - transformation des familles de polynômes
  - optimisation ...

Objet final

- But courant dans une preuve Coq = pas de solution à un système d'inégalités polynômiales donné
- L'appel de la tactique génère la preuve demandée

Pourquoi cet algorithme?

- Une complexité énorme ...
- ... mais un algorithme adapté à la certification :
  - peu de formalisation mathématique en Coq
  - structures de données élémentaires

Travail à plusieurs niveaux

- Algorithmique sur les polynômes (introduction de sous-résultants ?)
- Autres méthodes de CAD (Collins)?
- Conception d'outils adaptés en Coq (ex: preuves d'égalités dans les anneaux)