Simplification des Entiers Symboliques Conditionnels

Hanane Naciri

INRIA
Sophia Antipolis

Email: hanane.naciri@sophia.inria.fr

Mai 1999

PLAN

-
-
- Réalisation et approche suivie
-

CONTENU

☞ •
  – Problème général

•

• Réalisation et approche suivie

•

Problème général

- Calcul formel "générique"

```
> solve(a*x+b=0,x);
-b/a

> simplify((n+1)/(n^2-1));
   1
 -----
 n - 1
```

- Manipulation formelle d'expressions associées à des conditions.

$$\begin{cases} \sqrt{x+y} & , \quad x+y \geq 0 \\ \log xy = \log x + \log y & , \quad x > 0 \wedge y > 0 \\ \cos n\pi = 1 & , \quad n \equiv 0 \pmod 2 \end{cases}$$

Travaux existants

- **Systèmes de calcul formel**

assume de MAPLE

```
> assume(n, integer); sin(n*Pi);
  0

> assume(x>5, y<-10); is(x*y < 50);
 true
```

- **Objets inconnus et objets conditionnels**

C.Faure et J.H.Davenport

$$\begin{cases} |n| = n & si \quad n \geq 0 \\ |n| = -n & si \quad n < 0 \end{cases}$$

- **Évaluation dynamique**

D.Duval et L.Gonzalez-Vega

Problème traité durant le stage

- Manipulation d'entiers symboliques associés à des conditions de congruence et simplification de ces conditions.

$$\{n, \quad n \equiv 2 \pmod 4\}$$

$$\left\{ \begin{array}{l} n \equiv 2 \pmod 4 \\ \vee \\ m \equiv 2 \pmod 4 \end{array} \right. \quad \Rightarrow \quad n + m \equiv 0 \pmod 4$$

Exemple

- La méthode de recherche de contre exemples dans les tests de pseudo-primalité

$n = (k+1)(5k+1)(9k+1)$, nombre de carmichael

$(k+1)$, $(5k+1)$ et $(9k+1)$ premiers

$k \equiv 0 \pmod{7} \wedge k \equiv 2 \pmod{4} \wedge k \equiv 0 \pmod{3}$
$\wedge (k \equiv 0 \pmod{11} \vee k \equiv 1 \pmod{11})$

$$\left\{ \begin{array}{l} k \equiv 2310 \pmod{13860} \\ \vee \\ k \equiv 4830 \pmod{13860} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} k = 13860k' + 2310 \\ \vee \\ k = 13860k' + 4830 \end{array} \right.$$

CONTENU

- ✓
- ✓
- ☞ Réalisation et approche suivie
  - Le langage Aldor
  - Exemple de programme en Aldor
  - Expressions et conditions
-

Le langage Aldor

- Dédié aux algorithmes mathématiques.

  Système du calcul formel AXIOM

- Sépare l'interface de l'implémentation

  Catégorie: signature des opérateurs

  Domaine : Implémentation

- Générique.

  CAT: Category == with{ ... }

  ACategory(E : CAT) : Category == with{ ... }

- Bibliothèque SALLI

  Entiers, Booléens, Tableaux, Listes, chaînes de caractères

Exemple de programme en Aldor

- Catégorie des expressions algébriques

```
#include "exprlib"
define AlgExprCategory:  Category  == with{
    coerce : Symbol                -> %;
    coerce : %                     -> Symbol;
    ....
    one    : %;
    zero   : %;
    zero?   :  %                    -> Boolean;
    ....
    +: (%, %)                      -> %;
    *: (%, %)                      -> %;
    ....
    integer?: %                    -> Boolean;
    symbole?: %                    -> Boolean;
    power?:   %                    -> Boolean;
    ....
    operandes: %                   -> List %;
    ....
    =:(%, %)                        -> Boolean;
    << : (OutputStream, %)          -> OutputStream;
    default{ (e1: % ) - (e2: %) : % == { ... }
    ....
}
```

Exemple de programme en Aldor *(suite)*

- Domaine des expressions algébriques

```
#include "exprlib";
AlgExpression: AlgExprCategory with{
   power : (%, Integer) -> %;
   somme : List %    -> %;
   prod  : List %    -> %;
} == add {
   macro Rep == Union(i :Integer, s :Symbol,
                      tree:Record(op: Operator, args:List %));
   import from Rep;

   coerce(e:%): Symbol      == (rep e).s;
   coerce(s : Symbol):%     == per [s];
   ....
   one: %  == per [1];
   zero: % == per [0];
   zero?(t: %): Boolean == (t = zero);
   ....

   (e1: %) + (e2: %) :%           == {
   import from NormalAlgSum(%);
   ...
   }
   ....
```

8 Approche suivie

```
   integer?(e: %): Boolean == (rep e) case i;
   symbole?(e: %): Boolean == (rep e) case s;
   ...
   operandes(e: %): List % == (rep e).tree.args;

   (e1:%) = (e2: %): Boolean == {
   integer? e1 and integer? e2  => e1::Integer = e2::Integer;
   symbole? e1 and symbole? e2  => e1::Symbol = e2::Symbol;
   ....
   }
   ....
}
```

Expressions et conditions

- **Simplification des expressions algébriques**

$$4 + n + 5 + m^2 + 2 * n \rightarrow 9 + 3 * n + m^2$$

- **Simplification des conditions**

  - Algèbre booléen

$$(c1 \vee c2) \wedge c3 \rightarrow (c1 \wedge c3) \vee (c2 \wedge c3)$$

  - Conditions de congrunences

("Théorème des restes chinois généralisé")

$$\left\{ \begin{array}{lll} x \equiv & 1 & \pmod{180} \\ x \equiv & 61 & \pmod{600} \end{array} \right. \; ssi \quad x \equiv 1261 \pmod{1800}$$

Conclusion

■ Travail réalisé

– Définition des objets de bases (Expression, Condition)
– Simplification des expressions entières
– Simplification des conditions (le théorème des restes chinois généralisé)
– Étude de la simplification des expressions conditionnelles

■ Travail restant à effectuer

– Simplification des expressions conditionnelles.
– Test sur des exemples.