M2 PLS. Coq: des preuves et des nombres

Micaela Mayero

Université Paris 13,
LIPN UMR 7030-LoVe team
http://www-lipn.univ-paris13.fr/~mayero

17 octobre 2021

Problématique

- savoir compter
- savoir prouver
- quels nombres ?
- pour quoi faire ?

Qu'est ce que compter ?

Compter : "*Faire des calculs*"

Calcul : "*Opération destinée à déterminer le résultat d'une combinaison de nombres*"

Les enfants savent compter, les ordinateurs aussi...

Qu'est ce que prouver ?

Prouver : "*conclure logiquement à partir de propositions posées comme prémisses.*"

Les grands enfants savent prouver, les ordinateurs pas vraiment...

Pourquoi ? : résultat *vs* conclusion

Exemple concret

- Pas d'associativité dans le calcul :

  (1003+ -1000)+7.501 = 10.5010000000000012
  1003+ (-1000+7.501) = 10.5009999999999764

- Mais on la veut dans les preuves.

Quels entiers ?

- entiers binaires : codage machine, signés, non signés, complément à 2,...
- entiers de Peano : 0 | S n

```
Inductive nat : Set :=  O : nat | S : nat -> nat
```

Et dans les prouveurs ?

- entiers binaires : PVS, Coq, ...
- entiers de Peano : HOL, Coq, ...
- entiers relatifs dans Coq :

```
Inductive Z : Set :=
  | Z0 : Z
  | Zpos : positive -> Z
  | Zneg : positive -> Z.

Inductive positive : Set :=
  | xI : positive -> positive
  | xO : positive -> positive
  | xH : positive.
```

lecture <- xH=1 (bit de poids fort) x0=0 xI=1

```
Coq < Check 5.
Zpos (xI (x0 xH))       (101)

Coq < Check 6.
Zpos (x0 (xI xH))       (110)

Coq < Check 7.
Zpos (xI (xI xH))       (111)

Coq < Check 8.
Zpos (x0 (x0 (x0 xH))) (1000)

Coq < Check 9.
Zpos (xI (x0 (x0 xH))) (1001)

Coq < Check 10.
Zpos (x0 (xI (x0 xH)))  (1010)

Coq < Check 11.
Zpos (xI (xI (x0 xH))) (1011)
```

Quels réels ?

- axiomatisation/construction
- Cauchy, Cantor, coupure de Dedekind
- classique/intuitionniste
- 1er ordre/2d ordre

Et dans les prouveurs ?

- axiomatisation intuitionniste : Coq, ...
- axiomatisation classique : Coq, PVS, Lego, ...
- construction classique : HOL, HOL-Light, ...
- les réels de la bibliothèque standard de Coq : Corps ordonné archimédien complet

```
Parameter R : Set.
Parameter R0 R1: R.
Parameter Rplus Rmult: R -> R -> R.
Parameter Ropp Rinv : R -> R.
Parameter Rlt : R -> R -> Prop.
Parameter up : R -> Z.
```

```
Axiom Rplus_comm : forall r1 r2:R, r1 + r2 = r2 + r1.
Axiom Rmult_assoc : forall r1 r2 r3:R, r1 * r2 * r3 = r1 * (r2 * r3).
Axiom Rinv_l : forall r:R, r <> 0 -> / r * r = 1.
Axiom R1_neq_R0 : 1 <> 0.
Axiom total_order_T : forall r1 r2:R, {r1 < r2} + {r1 = r2} + {r1 > r2}.
Axiom Rlt_asym : forall r1 r2:R, r1 < r2 -> ~ r2 < r1.

Axiom archimed : forall r:R, IZR (up r) > r /\ IZR (up r) - r <= 1.

Definition is_upper_bound (E:R -> Prop) (m:R) := forall x:R, E x -> x <= m.
Definition bound (E:R -> Prop) :=  exists m : R, is_upper_bound E m.
Definition is_lub (E:R -> Prop) (m:R) :=
  is_upper_bound E m /\ (forall b:R, is_upper_bound E b -> m <= b).
Axiom
  completeness :
    forall E:R -> Prop,
      bound E -> (exists x : R, E x) -> { m:R | is_lub E m }.
```

- 17 axiomes
- L'inverse et la division (fonctions totales)
- L'ordre total
- La complétude (2d ordre)
- Attention aux axiomes !

Quels flottants ?

- norme
- étendus
- axiomatisation/construction

La norme IEEE 754 (1985, 2008)

Précision	Codage	Signe	Exposant	Mantisse	Valeur
Simple	32 bits	1 bit	8 bits	23 bits	$(-1)^S \times M \times 2^{(E-127)}$
Double	64 bits	1 bit	11 bits	52 bits	$(-1)^S \times M \times 2^{(E-1023)}$
D.étendue	80 bits	1 bit	15 bits	64 bits	$(-1)^S \times M \times 2^{(E-16383)}$

Exemple : $-15,25 = 0xC1740000$

```
Détails rapides: 15=1111 et 0,25=1/4=1/2^2=0,01
15,25=1111,001=1,111001 avec exposant=127+3=130
Signe=1 (négatif)
Exposant=130=10000010
Mantisse=111001 + (23-6)zéros
Et donc: -15,25 = 1 10000010 11100100000000000000000
                = 0xC1740000
```

Et dans les prouveurs ?

- dans Coq, HOL-Light, ...
- les flottants dans Coq :

```
Variable radix : Z.
Hypothesis radixMoreThanOne : (1 < radix)%Z.

Record float : Set := Float {Fnum : Z; Fexp : Z}.
Definition FtoR (x : float) :=
     (Fnum x * powerRZ (IZR radix) (Fexp x))%R.
```

Arithmétique réelle exacte

- L'arithmétique réelle exacte consiste à représenter un nombre réel par une fonction qui en donne une approximation rationnelle aussi précise que souhaitée.
- L'avantage majeur est la "décidabilité " de l'égalité (à un nombre de décimales près).
- L'inconvénient majeur est la forte complexité temporelle (structures de données et algorithmes).

Plusieurs arithmétiques réelles exactes

- Représentation par des nombres P-adics, fractions continues
- MPFR, Constructive Reals Calculator (Hans Boehm), $\cdots$

Test : $ln(e^{ln(e^{-36}+\pi)} - \pi)$

```
#let pi = 4.0 *. atan 1.0;;
#log(exp(log(exp(-36.)+.pi))-.pi);;
- : float = -34.657359027997263
#log(exp(log(exp(-37.)+.pi))-.pi);;
- : float = neg_infinity
```

- théorie des nombres
- analyse
- algèbre
- arithmétique
- correction et complétude d'une théorie

Lapalissade

Bien avoir en tête quels sont les objectifs pour définir et utiliser les “bons nombres”...