Calcul dans Coq

Laurent Théry
Marelle INRIA Sophia-Antipolis France

Plan

Généralités

L'état actuel

Développements possibles

Prouveurs interactifs


Coq


Hol

Isabelle

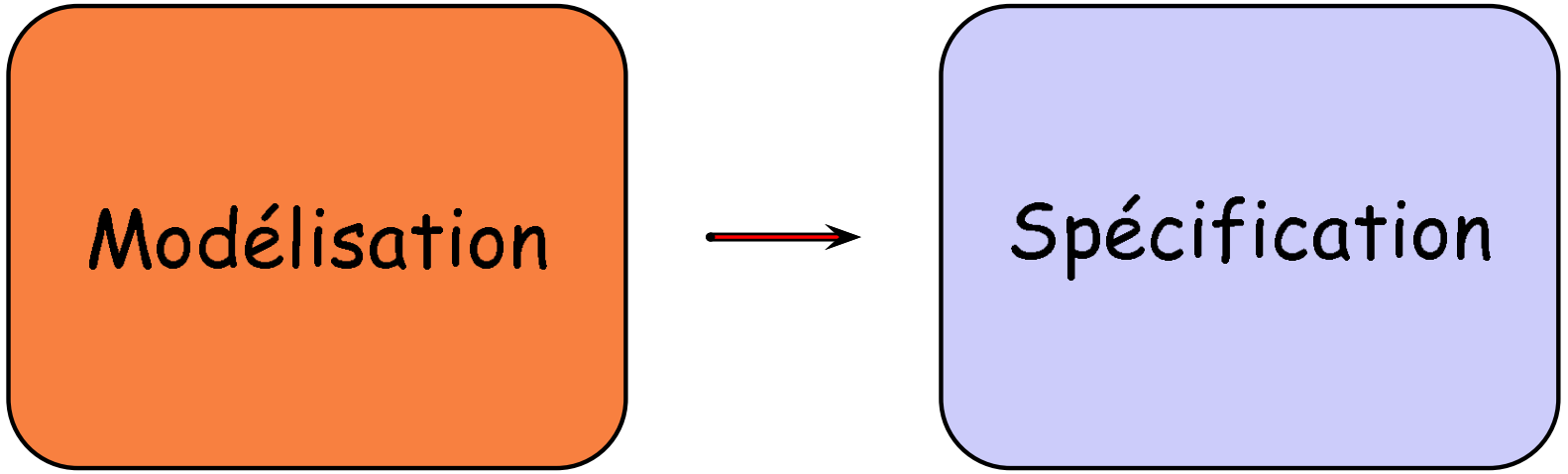

Mizar

Pvs

Acl2

INRIA

Prouveurs interactifs



Modèle syntaxique: Définitions + Propriétés initiales

Spécification: Propriétés dérivées

Confiance: minimiser le code critique

Calcul

Modélisation: $0,1,+$

$$n = \underbrace{1 + (\cdots + (1}_{n} + 0) \ldots)$$

$$\forall x,\ 0 + x = x$$

$$\forall x\, y,\ x + y = y + x$$

$$\forall x\, y\, z,\ x + y + z = x + (y + z)$$

$$\forall x\, y\, z,\ x + y + z = y + (x + z)$$

$$2 + 2 = 4 \rightsquigarrow 1 + 3 = 4 \rightsquigarrow 0 + 4 = 4 \rightsquigarrow 4 = 4$$

Calcul en Coq

En Coq, les objets sont vivants:

le test d'égalité se fait modulo l'"évaluation" des fonctions

Definition $\mathbb{N} := \mathtt{O} \mid \mathtt{S}\colon \mathbb{N} \to \mathbb{N}$.

Definition $m + n :=$
  if m is S m' then S $(m' + n)$ else n

$2 + 2 = 4$

Calcul en Coq

Les performances suivent celles d'Ocaml

[1984] `compute`
Machine de réduction

[2004] `vm_compute`
Machine virtuelle (Byte-code)

[2011] `native_compute`
Compilation (Native-code)

Nombres en Coq

[1984] `nat`,	base 1
[1994] `positive`, `Z`, `Q`	base 2
[2008] `bigN`, `bigZ` , `bigQ`	arbre binaire
[2001] `Float`	paire de `Z`
[2008] `Interval`	paramétrique
[2009] `CCorn`	calcul exact

INRIA

Quelques Tests

Fibonacci

$f_0 = 0, \quad f_1 = 1, \quad f_{n+2} = f_{n+1} + f_n$

```
Time Eval compute in fib_nat 25.
     = 75025
Finished transaction in 0.95 secs

Time Eval compute in fib_nat 26.
     Segmentation fault (core dumped)
```

Quelques Tests

Fibonacci

$$f_{2n-1} = f_{n-1}^2 + f_n^2, \quad f_{2n} = (2f_{n-1} + f_n)f_n$$

```
Time Eval compute in fib_pos 10000.
     = 3364476...366875
Finished transaction in 97.03 secs

Time Eval vm_compute in fib_pos 10000.
     = 3364476...366875
Finished transaction in 1.87 secs
```

INRIA

Quelques Tests

Fibonacci

$$f_{2n-1} = f_{n-1}^2 + f_n^2, \quad f_{2n} = (2f_{n-1} + f_n)f_n$$

```
Time Eval vm_compute in fib_pos 10000.
     = 3364476...366875
Finished transaction in 1.87 secs

Time Eval native_compute in fib_pos 10000.
     = 3364476...366875
Finished transaction in 0.66 secs
```

Quelques Tests

Fibonacci

$$f_{2n-1} = f_{n-1}^2 + f_n^2, \quad f_{2n} = (2f_{n-1} + f_n)f_n$$

```
Time Eval vm_compute in fib_bigN 100000.
     = 2597406...8746875
Finished transaction in 1.14 secs

Time Eval native_compute in fib_bigN 100000.
     = 2597406...8746875
Finished transaction in 0.50 secs
```

Quelques Tests

Fibonacci

$$f_{2n-1} = f_{n-1}^2 + f_n^2, \quad f_{2n} = (2f_{n-1} + f_n)f_n$$

```
Time Eval vm_compute in fib_bigN 1000000.
     = 1953282...2546875
Finished transaction in 35.90 secs

Time Eval native_compute in fib_bigN 1000000.
     = 1953282...2546875
Finished transaction in 16.21 secs
```

Développements possibles

Augmentation de la puissance de calcul de Coq

Besoin d'évaluer la puissance de calcul nécessaire pour Tamadi

Implantation de référence

Prouver la correction par rapport à l'implantation de référence

Augmenter la base de confiance