Faithful results about computer arithmetic: polynomial evaluation

Sylvie Boldo

MAP meeting – January, 13th 2004

LIP
(Laboratoire de l'Informatique du Parallélisme)

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

INRIA
RHÔNE-ALPES

People involved

- Arénaire project (Lyon)
  - Marc Daumas: CNRS
  - Guillaume Melquiond: PhD 2003–?
  - Sylvie Boldo: PhD sept 2001– dec 2004 ?
- Lemme project (Sophia-Antipolis)
  - Laurent Théry: INRIA
  - Laurence Rideau: INRIA

Outline

1. Introduction
2. Our own formalization: What? Why? How?
3. An example: faithfulness of Horner's rule under conditions
4. Various conclusions & learnings
5. Perspectives

Introduction

Motivations

A few expensive arithmetic-based bugs (Pentium bug. . .)

$\Rightarrow$ a will to guarantee products

$\Rightarrow$ hardware products (circuits)

$\Rightarrow$ software products (embedded, off-line)

$\Rightarrow$ need of theoretical guarantees in order to prevent bugs

State of the art

- Kaufmann, Lynch, Moore and Russinoff, 1998: ACL2

  work on AMD K5 and K7, first order logic

- Miner / Carreño, 1995: PVS/HOL

  work at the NASA, digit-oriented specifications

- Harrison, 1997 : HOL

  Cambridge, now with Intel, generic definitions and instantiations

- Berg, Jacobi and Paul, 1995: PVS

  work in Saarbrücken, specific IEEE-754 rounding developments

Our own formalization:

What? Why? How?

What?

A Coq formalization

$\approx$ 50 files

went through many Coq versions: V6.3, V7.0, V7.1, V7.2, V7.3, V7.4.

$\Rightarrow$ V8.0?

Part of it is a Coq contribution. And the rest?

The formalization

Float $=$ pair of signed integers (mantissa, exponent)

$$(n, e) \in \mathbb{Z}^2 \quad \hookrightarrow \quad n \times \beta^e \in \mathbb{R}$$

$$1.0001_2 \ \ \mathsf{E}\ 3 \quad \mapsto \quad (10001_2, -1)_2$$

We bound the floats to instantiate machine precision:

$$(n, e) \text{ bounded} \quad \Leftrightarrow \quad |n| < N = \beta^p \ \wedge \ e \geq -e_{min}$$

Instantiations

IEEE float: $\boxed{0}\boxed{101010001}\boxed{100000110100100}$

$1 \qquad\qquad\qquad\qquad p-1$

Coq float: $(16804, 210)$

$|\;| \leq \beta^p \qquad \geq -e_{min}$

Format	p	$-e_{min}$
Single	24	-149
Double	53	-1074

Main difference with IEEE-754-like behavior

$$(110_2, 1)_2 =_{\mathbb{R}} (1100_2, 0)_2 =_{\mathbb{R}} (11_2, 2)_2 =_{\mathbb{R}} 12_{10}$$

$\Rightarrow$ several possible floats share the same real value.

Why?

- generic and concise formalization
- proved results for any radix
- proved results for any bound on mantissa and amplitude
- handle denormal floats (intellectual & industrial need)
- use mathematical specifications and not hardware implementation

  $\Rightarrow$ afterwards, better expression of necessary and sufficient conditions !

How?

- pen & paper proof
- simplification/factorization of the pen & paper proof
- state and prove the lemmas
- try to prove the main theorem

(•) add more useful lemmas

(•) add proofs of forgotten subcases

An example:

faithfulness of Horner's rule under conditions

Motivations

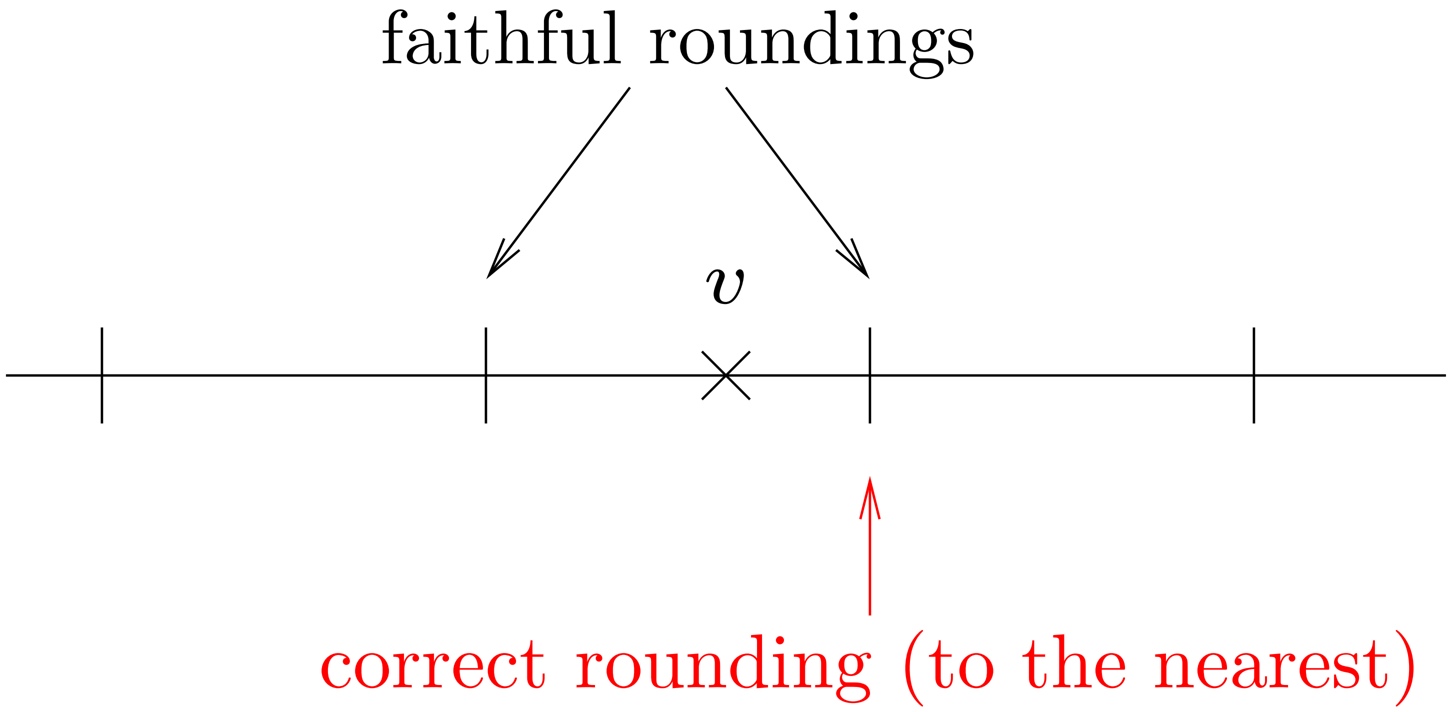
We want to compute $a \times x + y$ and we know that

$$a \times x \ll y$$

We are then convinced that $\circ(y + \circ(a \times x))$ is "very near" the correct result, even if a and x are not exact.

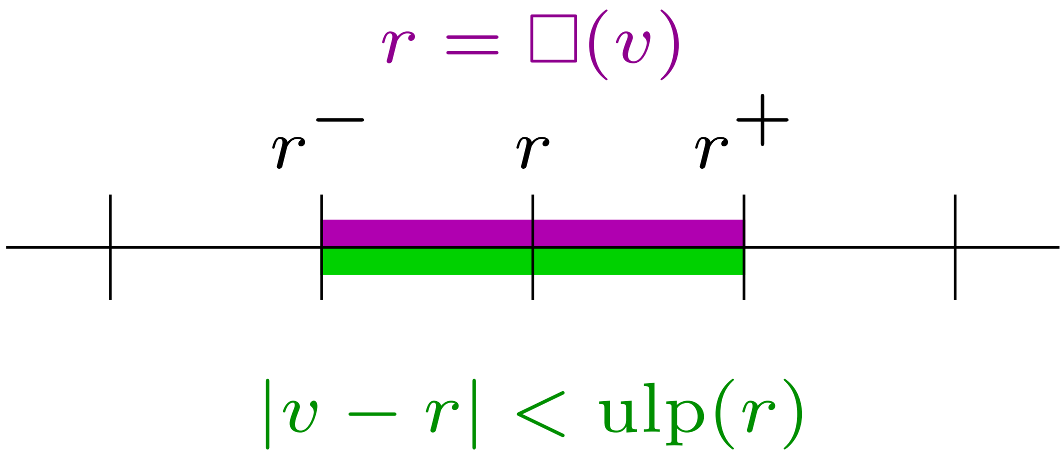
Faithful?

We have $r = \square(v)$ iff r is either the rounded towards $+\infty$ or towards $-\infty$ of v.



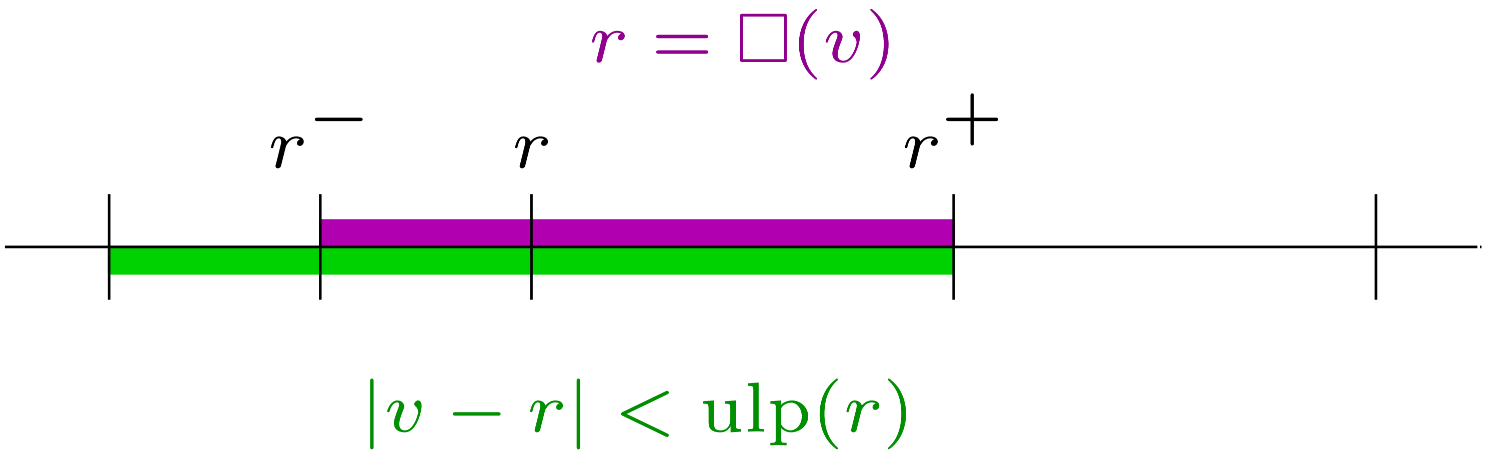
Faithful/ulp?

We could use $|v - r| < \mathrm{ulp}(r)$



ulp: unit in the last place $\left(\mathrm{ulp}(x) = 2^{\lfloor log_2(x) \rfloor - p + 1}\right)$

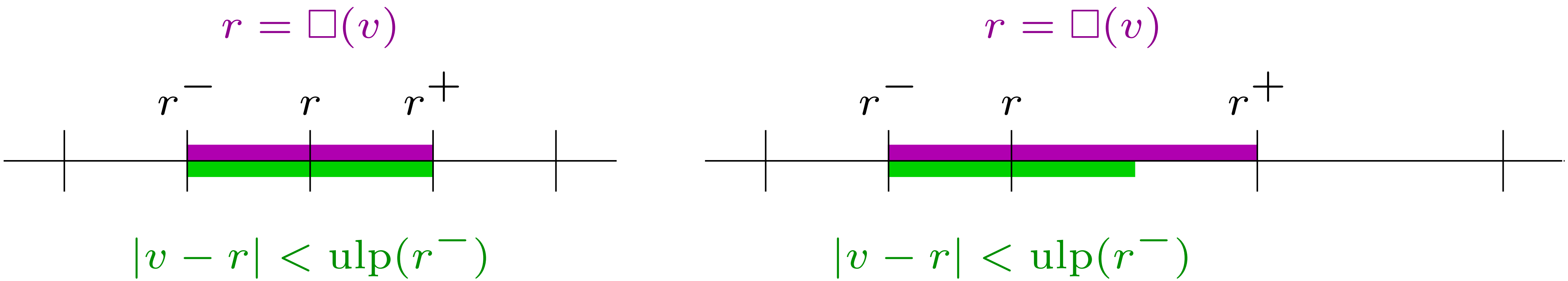
But it is not equivalent: (if $r = 2^k$)



$\Rightarrow$ faithful is the "tightest" non-perfect condition

Lemma I

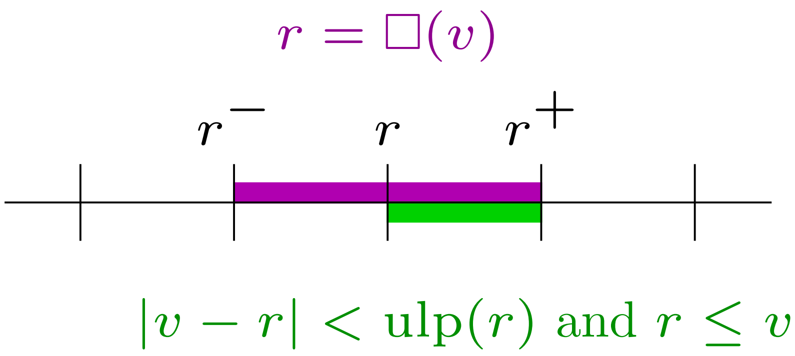
If $r > 0$ and $|v - r| < \mathrm{ulp}\,(r^{-})$ then $r = \square(v)$.

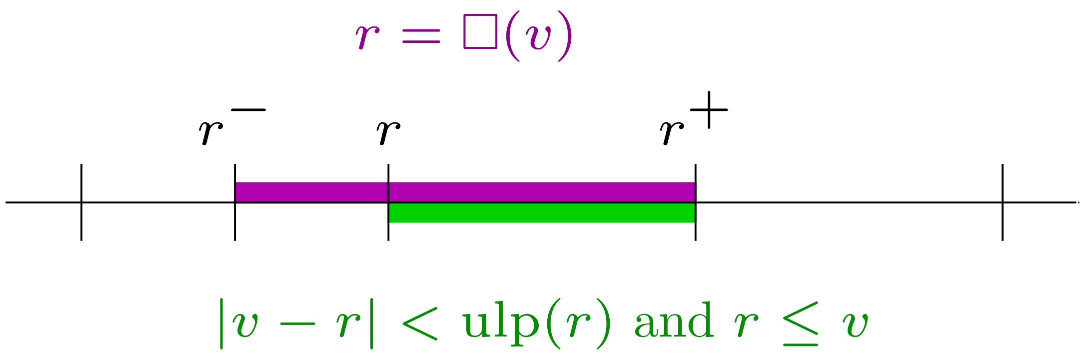


r^{-} is the floating-point number predecessor of r and r^{+} is the successor.

Lemma II

If $r > 0$ and $|v - r| < \mathrm{ulp}(r)$ and $r \leq v$ then $r = \square(v)$.





Notations

a, x, y, t, u are floating-point numbers such that

$$t = a \otimes x \qquad \text{and} \qquad u = t \oplus y.$$

a_1, x_1, y_1 are the exact real values of a, x and y.

We want to give sufficient hypotheses to ensure that $u = \square(a_1 \times x_1 + y_1)$.

The radix is 2.

`Axpy_opt`

Tight condition using the inputs (including denormal cases):

- if $\frac{5+5\times 2^{-p}}{1-2^{-p}} \times \left(|a \times x| + 2^{-e_{min}-1}\right) \leq |y|$,
- if $|y_1 - y| + |a_1 \times x_1 - a \times x|$
  $\leq 2^{-p-2} \times \left(\left(1 - 2^{1-p}\right) \times |y| - |a \times x|\right) - 2^{-e_{min}-2}$,

then $u = \square(a_1 \times x_1 + y_1)$.

Axpy_Simpl2

User-friendly condition (including denormal cases):

- if $p \geq 4$,
- if $6\ |a \times x| \leq |y|\ -3 \times 2^{-e_{min}}$,
- if $|y_1 - y| + |a_1 \times x_1 - a \times x| \leq \frac{2^{-p}}{6} \times |y|\ -2^{-e_{min}-2}$,

then $u = \square(a_1 \times x_1 + y_1)$.

Quantity of Coq

- a file of lemmas about faithfulness (260 lines)
- a file for the rest (3 400 lines)
  - 4 generic lemmas
  - 10 specific lemmas
  - 5 usable theorems
  - 3 theorems when using a FMAC
- written using ProofGeneral

Structure of the formal proof

- $u > 0$
  - the float $a \otimes x$ is normal (main case)
  - the float $a \otimes x$ is denormal
    - $u = a \otimes x + y$ (everybody is denormal)
    - $-e_{min} + 1 \leq e_{u^-}$ (u is big enough)
    - $e_{u^-} = -e_{min}$ and $e_u = -e_{min} + 1$

      (awful case where no other theorem applies !)

- $u < 0$ (immediate from the positive case)

- $u = 0$ (some work here)

Various conclusions & learnings

An often-used quantity: the ulp

$$\mathrm{ulp}(f) = \beta^{e_{\mathcal{N}(f)}}$$

$\Rightarrow$ we must consider the normalized float

$\Rightarrow$ back to the definition (pair of integers) of the float

$\Rightarrow$ complex because of the multiple representations,

$\Rightarrow$ the rest is handled using middle-level lemmas

$\Rightarrow$ the user is not always protected from the definitions

Drawbacks

- far from internal representations (also an advantage)
- computations with real numbers are a pain
  $\Rightarrow$ conditions apply only when the radix is 2
- some silly cases cannot be automatically handled
- time-consuming

Advantages

- representation near our trends of thought and easy to use
- certified result (no forgotten subcase)
- reusable lemmas

  with no hidden implicit condition
- nobody has to check the computations inside my proof
- trust in the result
- tightness of the result: any weakening is clear in the proof

Example where faithfulness is proved

We approximate $\exp(x)$ with x in $[-2^{-3}, 2^{-3}]$.

$$P(x) = 1 + x + \frac{1}{2}x^2 + \frac{6004799503158175}{36028797018963968}x^3 + \frac{6004799503152767}{144115188075855872}x^4$$
$$+ \frac{4803839629518891}{576460752303423488}x^5 + \frac{1601279914265145}{1152921504606846976}x^6$$
$$+ \frac{3660108472028231}{18446744073709551616}x^7 + \frac{7318367436494265}{295147905179352825856}x^8$$

The truncation error $|\exp(x) - P(x)|/|x|$ is bounded by

$$\frac{5509901405496691}{81129638414606681695789005144064}.$$

Extensions

- similar result using a FMAC (fused multiply-and-accumulate) in the IA-64 or PowerPC

- programs in Maple, Java and C that check criteria for a polynomial associated with a domain for the indeterminate and a possible truncation error.

  (See RR INRIA 4707 or the Web)

Conclusion

- a formally-proved result both complex and useful, tediously covering all cases
- already many results using this formalization:
  - expansions (multi-precision via floating-point numbers)
  - argument reduction
  - representable correcting terms (FMAC?)
  - properties of a generalized floating-point system (2's complement – Airbus)

Perspectives

All the IEEE-754 standard?

- normal and denormal floats
- variable formats
- rounding modes
- arithmetic operations and comparisons
- vector of bits representation
- special numbers (0^-, $NaN\ldots$)
- exceptions, traps
- conversions (with decimal or integers)

All the IEEE-754 standard?

- vector of bits representation

  $\Rightarrow$ either radix-2 vector: simple and usable

  $\Rightarrow$ or a radix-generic vector: powerful but probably useless

- special numbers (0^-, $NaN\ldots$): natural extension

- exceptions: possible

- traps: very complex – linked to semantics

- conversions (with decimal or integers): useless

Less sweat, less tight

We want to handle easy goals or loose results (includes many industrial applications)

but without overdoing it

$\Rightarrow$ automatization of interval-like techniques

see G. Melquiond

A few references

S. Boldo et M. Daumas, **A simple test qualifying the accuracy of Horner's rule for polynomials**, *Numerical Algorithms*, 2004, (also in conference).

S. Boldo et M. Daumas, **Properties of two's complement floating-point notations**, *Software Tools for Technology Transfer*, 2004, (also in conference).

R.-C. Li, S. Boldo et M. Daumas, **Theorem on efficient argument reductions**, *ARITH-16*, Santiago de Compostella, 2003, (submitted to TCS).

Links

http://perso.ens-lyon.fr/Sylvie.Boldo

Sylvie.Boldo@ens-lyon.fr