Formal Correctness of Comparison Algorithms between Binary64 and Decimal64 Floating-point Numbers

Arthur Blot

ENS Lyon, France

NSV, July 22-23, 2017

Outline

- IEEE 754-2008: binary and decimal formats
- No operation defined to compare them

- IEEE 754-2008: binary and decimal formats
- No operation defined to compare them
- Naïve method: let's compare:

  1. $y = 7205759403792794/2^{56}$ (binary64) $> 1/10$
  2. $z = 1/10$ (decimal64)

Then z converted to binary64 $= y$ and $y \leq z$!

Formal Setting

- Coq for formal proofs
- Flocq for defining floating-point numbers:

```
Record float (beta : radix) :=
  Float { Fnum : Z; Fexp : Z }.
```

- Real numbers of Coq + Coquelicot (exponential, etc.)

```
Definition F2R (f : float beta) :=
  (Fnum f) * beta ^ (Fexp f).
```

- Interval library to prove inequalities automatically:

```
Lemma bound_exp_taylor2 x :
  0 <= x <= 1 →
  Rabs ((1 + x + x ^ 2 / 2) - exp x) <= 22 / 100.
```

Continued Fractions

Used to find the best fractional approximation of real numbers:

```
Lemma halton_min (n : nat) (p q : Z) (r : R) :
   0 < q < q[r]_n.+1 → t[r]_n <= Rabs (q * r - p).
```

Where:

- $s_n = p_n / q_n$ is the sequence of convergents
- t[r]_n is $|q_n r - p_n|$

Goal

Obtaining two comparison algorithms:

```
Lemma eq_correct (f1 : float 2) (f2 : float 10) :
  Cond → eq f1 f2 = true ↔ F2R f1 = F2R f2.
```

and

```
Lemma comp_correct (f1 : float 2) (f2 : float 10) :
  Cond →    comp f1 f2 = Lt ↔ F2R f1 < F2R f2
          ∧  comp f1 f2 = Eq ↔ F2R f1 = F2R f2
          ∧  comp f1 f2 = Gt ↔ F2R f1 > F2R f2.
```

Outline

The code is available at:

`https://gitlab.com/artart78/compbindec`

- 5 files
- 6500 lines of code
- 25 minutes to be checked by Coq

Let us compare a binary64 number

$$x_2 = M_2 \cdot 2^{e_2 - 52}$$

and a decimal64 one

$$x_{10} = M_{10} \cdot 10^{e_{10} - 15}$$

where e_i and M_i are bounded.

Let us compare a binary64 number

$$x_2 = M_2 \cdot 2^{e_2 - 52}$$

and a decimal64 one

$$x_{10} = M_{10} \cdot 10^{e_{10} - 15}$$

where e_i and M_i are bounded.

We suppose $x_2 > 0$, $x_{10} > 0$.

Let us compare a binary64 number

$$x_2 = M_2 \cdot 2^{e_2 - 52}$$

and a decimal64 one

$$x_{10} = M_{10} \cdot 10^{e_{10} - 15}$$

where e_i and M_i are bounded.

We suppose $x_2 > 0$, $x_{10} > 0$.

Up to normalization:

$$2^{52} \leq M_2 < 2^{53} - 1$$
$$2^{53} \leq 2^{\nu} M_{10} \leq 2^{54} - 1$$

- We define m, h, n and $g \in \mathbb{Z}$ in order to have:

$$x_2 = m \cdot 2^h \cdot 2^{g-\nu}$$
$$x_{10} = n \cdot 5^g \cdot 2^{g-\nu}$$

- We define m, h, n and $g \in \mathbb{Z}$ in order to have:

$$x_2 = m \cdot 2^h \cdot 2^{g-\nu}$$
$$x_{10} = n \cdot 5^g \cdot 2^{g-\nu}$$

- $\Rightarrow$ Compare $m \cdot 2^h$ and $n \cdot 5^g$.

- Bounds on m, h, n, g.

Outline

$$\phi(h) = \lfloor h \cdot \log_5 2 \rfloor \quad \psi(g) = \lfloor g \cdot \log_2 5 \rfloor$$

$$g < \phi(h) \Rightarrow x_2 > x_{10}$$
$$g > \phi(h) \Rightarrow x_2 < x_{10}$$

and:

$$\phi(h) = \left\lfloor h \cdot L_\phi \cdot 2^{-19} \right\rfloor \quad \text{for } |h| \leq 1831$$
$$\psi(g) = \left\lfloor L_\psi \cdot g \cdot 2^{-12} \right\rfloor \quad \text{for } |g| \leq 204$$
$$\psi(16q) = \left\lfloor L_\psi \cdot q \cdot 2^{-8} \right\rfloor \quad \text{for } |q| \leq 32$$

$$\phi(h) = \left\lfloor h \cdot L_\phi \cdot 2^{-19} \right\rfloor \quad \text{for } |h| \leq 1831$$

is outside the reach of the `interval` tactic.

We prove an enclosure I over $\log_5 2$ and use:

$$\lfloor x \rfloor = y \qquad \text{iff} \qquad y \leq x < y + 1$$

which is proved automatically with `interval`.

```
Definition easycomp : comparison :=
  let h := v + e2 - e10 - 37 in
  let g := e10 - 15 in
  let phih := (225799 * h) / (2 ^ 19) in
  if g < phih then Gt
  else if g > phih then Lt
  else Eq.
```

and its associated theorem of correctness:

```
Theorem easycomp_correct:
  (easycomp = Lt → F2R x2 < F2R x10) ∧
  (easycomp = Gt → F2R x2 > F2R x10) ∧
  (easycomp = Eq ↔ g = phi h).
```

```
Compute easycomp {|Fnum := 7205759403792794; Fexp := -56|}
                 {|Fnum :=                1; Fexp :=  -1|}.
  = Eq
  :  comparison
```

Outline

Sommaire

- We suppose $g = \phi(h)$.
- The function:

$$f(h) = 5^{\phi(h)} \cdot 2^{-h} = 2^{\lfloor h \log_5 2 \rfloor \cdot log_2 5 - h}$$

verifies:

$$f(h) \cdot n > m \Rightarrow x_{10} > x_2$$
$$f(h) \cdot n < m \Rightarrow x_{10} < x_2$$
$$f(h) \cdot n = m \Rightarrow x_{10} = x_2$$

- We suppose $g = \phi(h)$.
- The function:

$$f(h) = 5^{\phi(h)} \cdot 2^{-h} = 2^{\lfloor h \log_5 2 \rfloor \cdot log_2 5 - h}$$

verifies:

$$f(h) \cdot n > m \Rightarrow x_{10} > x_2$$
$$f(h) \cdot n < m \Rightarrow x_{10} < x_2$$
$$f(h) \cdot n = m \Rightarrow x_{10} = x_2$$

- $\Rightarrow$ need a good precision on $f(h) \cdot n$.
- η: smallest error of $|m/n - f(h)|$
- $\eta > 2^{-113.7}$.

How to prove the bound on η?

- For $-53 \leq h \leq 53$, immediate
- Otherwise, for a given h:
    - Enumerate the convergents p_l/q_l approximating $f(h)$ with $q_l < 2^{53} - 1$
    - Check the bound for them only

Sommaire

If $|\mu - f(h) \cdot n| < \epsilon n < \eta n/4$ then:

$$\mu > m + \epsilon \cdot 2^{54} \Rightarrow x_{10} > x_2$$
$$\mu < m - \epsilon \cdot 2^{54} \Rightarrow x_{10} < x_2$$
$$|m - \mu| \leq \epsilon \cdot 2^{54} \Rightarrow x_{10} = x_2$$

Problem: many values to tabulate.

Build the table `f_tbl` of $\lceil f(h) \cdot 2^{127} \rceil$, then:

```
Definition direct_method_alg :=
  let v := n * (f_tbl h) - m * 2 ^ 127 in
  if v < - 2 ^ 57 then Gt
  else if v > 2 ^ 57 then Lt
  else Eq.
```

giving

```
Compute direct_method_alg
          {| Fnum := 72057594037927934; Fexp := -56 |}
          {| Fnum :=                 1; Fexp :=  -1 |}.
  = Gt
  :  comparison
```

Sommaire

- We use:

$$g = 16q - r \quad q = \left\lfloor \frac{g}{16} + 1 \right\rfloor$$

  such that $f(h) = 5^{16q} \cdot 5^{-r} \cdot 2^{-h}$.

- We normalize 5^{16q} and 5^r:

$$\theta_1(q) = 5^{16q} \cdot 2^{-\psi(16q)+127}$$
$$\theta_2(r) = 5^r \cdot 2^{-\psi(r)+63}$$

- We then have:

$$f(h) = \frac{\theta_1(q)}{\theta_2(r)} 2^{-64-\sigma(h)}$$

  with:

$$\sigma(h) = \psi(r) - \psi(16q) + h$$

- We define:

$$\Delta = \theta_1(q) \cdot n \cdot 2^{-64+8} - \theta_2(r) \cdot m \cdot 2^{8+\sigma(h)}$$

whose sign gives the comparison between the x_i.

- We have: $|\Delta| \geq 2^{124}\eta$ si $\neq 0$

- We define:

$$\Delta = \theta_1(q) \cdot n \cdot 2^{-64+8} - \theta_2(r) \cdot m \cdot 2^{8+\sigma(h)}$$

whose sign gives the comparison between the x_i.

- We have: $|\Delta| \geq 2^{124}\eta$ si $\neq 0$

- We compute an approached version:

$$\tilde{\Delta} = \left\lfloor \lceil \theta_1(q) \rceil \cdot n \cdot 2^{8-64} \right\rfloor - \theta_2(r) \cdot m \cdot 2^{8+\sigma(h)}$$

whose sign gives us again what we want.

We make tables for $\lceil \theta_1(q) \rceil$ and $\theta_2(r)$, then:

```
Definition ineq_alg : comparison :=
  let q := g / 16 + 1 in
  let r := 16 * q - g in
  let psir := (9511 * r) / 2 ^ 12 in
  let psiq := (9511 * q) / 2 ^ 8 in
  let s := psir - psiq + h in
  let a := ((theta1_tbl q) * (n * 2 ^ 8)) / (2 ^ 64) in
  let b := (theta2_tbl r) * (m * 2 ^ (8 + s)) in
  let D := a - b in (0 =?= D)
```

gives

```
Compute ineq_alg {|Fnum := 7205759403792794; Fexp := -56|}
                 {|Fnum :=                1; Fexp :=  -1|}.
  = Gt
  : comparison
```

Outline

- Testing equality ⇔ testing $m \cdot 2^h = n \cdot 5^g$.
- We prove:
  - either $0 \leq g \leq 22$, $0 \leq h \leq 53$, $5^g \mid m$ and $2^h \mid n$
  - or $-22 \leq g \leq 0$, $-51 \leq h \leq 0$, $2^{-h} \mid m$ and $5^{-g} \mid n$.
- We then check that $5^g \cdot (n2^{-h}) = m$ or $5^{-g} \cdot (m2^h) = n$

```
Definition eq_alg : bool :=
  if (0 <= h) && (h <= 53) && (0 <= g) &&
     (g <= 22) && (n mod (2 ^ h) == 0) then
    let m' := 5 ^ g * (n / (2 ^ h)) in m' == m
  else if (h >= -51) && (-22 <= g) &&
         (g <= 0) && (m mod (2 ^ (-h)) = 0) then
    let n' := 5 ^ (- g) * (m / (2 ^ (- h))) in n' == n
  else
    false.
```

gives

```
Theorem eq_alg_correct : eq_alg = true ↔ (F2R x2 = F2R x10).
```

and

```
Compute eq_alg {| Fnum := 7205759403792794; Fexp := -56 |}
               {| Fnum :=                1; Fexp :=  -1 |}.
  = false
  : comparison
```

Outline

Conclusion

- Full formalization of the algorithms
- Some minor mistakes detected
- Some proofs slightly changed
- Life would have been easier with a better `interval` tactic

Future Work

Some small chunks are missing:

- Counting how many FP numbers only need easy comparison
- Proving an actual software implementation

Future Work

Some small chunks are missing:

- Counting how many FP numbers only need easy comparison
- Proving an actual software implementation

The big challenge would be to prove the algorithm for all the formats!