Flocq: A Unified Library for Proving Floating-point Algorithms in Coq

Sylvie Boldo    Guillaume Melquiond

INRIA, LRI, ANR FjST

2011-07-27
Floating-point arithmetic: a widely-used approach for approximating computations on real numbers.

Numerical issues: exceptional behaviors, inaccurate results. Usually out of the reach of exhaustive testing.
Floating-point arithmetic: a widely-used approach for approximating computations on real numbers.

Numerical issues: exceptional behaviors, inaccurate results. Usually out of the reach of exhaustive testing.

High level of safety thanks to formal methods: model checking, satisfiability, temporal logic, abstract interpretation, and so on. Automated and suitable for large codes.
**Computer Arithmetic and Formal Proofs**

Floating-point arithmetic: a widely-used approach for approximating computations on real numbers.

Numerical issues: exceptional behaviors, inaccurate results. Usually out of the reach of exhaustive testing.

High level of safety thanks to formal methods: model checking, satisfiability, temporal logic, abstract interpretation, and so on. Automated and suitable for large codes.

What about correctness? Intricate algorithms require formal proofs. (Hopefully they are short.)
Some Prior Work on Formal Proofs for FP Arithmetic

**Formal proof**: a proof that can be checked automatically by a computer.

- **Formalization of standards**: Barrett (Z), Carreño, Miner (PVS), Loiseleur (Coq).

- **Certification of low-level designs**: Kaufmann, Lynch, Moore, Russinoff (ACL2), Kaivola, Kohatsu (Forte), Berg, Jacobi (PVS).

- **Certification of high-level algorithms**: Harrison (HOL Light), Boldo (Coq).
Some Prior Work in Coq

- **Float** → **Pff** theorems about FP arithmetic
  - any radix, only FLT format (with subnormal numbers),
  - axiomatized rounding operators,
  - comprehensive library.

- **Gappa** verification of FP algorithms
  - radix 2, any format,
  - effective rounding for dyadic numbers (+, ×),
  - dedicated library.

- **Coq.Interval** proofs automated by FP computations
  - any radix, only FLX format (normal numbers only),
  - effective FP operations (+, ×, ÷, √·, etc),
  - dedicated library, some incomplete proofs.
Some Prior Work in Coq

- **Float → Pff**
  - theorems about FP arithmetic
  - any radix, only FLT format (with subnormal numbers),
  - axiomatized rounding operators,
  - comprehensive library.

- **Gappa**
  - verification of FP algorithms
  - radix 2, any format,
  - effective rounding for dyadic numbers (+, ×),
  - dedicated library.

- **Coq.Interval**
  - proofs automated by FP computations
  - any radix, only FLX format (normal numbers only),
  - effective FP operations (+, ×, ÷, √, etc),
  - dedicated library, some incomplete proofs.
Motivations

- Ease the **combined usage** of several formalisms:
  - proof obligations generated by the Why tool,
  - theorems already proved in the Pff library,
  - automation provided by the gappa tactic.
Motivations

- Ease the **combined usage** of several formalisms:
  - proof obligations generated by the Why tool,
  - theorems already proved in the Pff library,
  - automation provided by the gappa tactic.

- Design a formalization:
  - as **generic** as possible,
  - that avoids earlier shortcomings,
  - that scales better with future works.
Motivations

- Ease the *combined usage* of several formalisms:
  - proof obligations generated by the Why tool,
  - theorems already proved in the Pff library,
  - automation provided by the gappa tactic.

- Design a formalization:
  - as *generic* as possible,
  - that avoids earlier shortcomings,
  - that scales better with future works.

- Explore properties of usual and *exotic* formats.

⇒ **Flocq**: a Coq formalization for computer arithmetic.
Core Library

1. Introduction

2. Core library
   - Axiomatic rounding and formats
   - Generalizing formats
   - Rounding operators

3. Auxiliary libraries

4. Conclusion
Predefined Axiomatic Rounding

Axiomatic rounding: relation $Q(x, f)$ “real $x$ rounds to $f$.”

Predefined relations: rounding downward, upward, toward zero, to nearest, for any format $F$. 
Predefined Axiomatic Rounding

Axiomatic rounding: relation \( Q(x, f) \) “real \( x \) rounds to \( f \).”

Predefined relations: rounding downward, upward, toward zero, to nearest, for any format \( F \).

Example (\( \nabla_F \), rounding toward \(-\infty\) on \( F \))

\[
\nabla_F(x, f) \equiv f \in F \land f \leq x \land (\forall g \in F, g \leq x \Rightarrow g \leq f).
\]
Predefined Axiomatic Rounding

Axiomatic rounding: relation $Q(x, f)$ “real $x$ rounds to $f$.”

Predefined relations: rounding downward, upward, toward zero, to nearest, for any format $F$.

Example ($\bigtriangledown_F$, rounding toward $-\infty$ on $F$)

$$\bigtriangledown_F(x, f) \equiv f \in F \land f \leq x \land (\forall g \in F, g \leq x \Rightarrow g \leq f).$$

All these relations describe monotone total functions when format $F$ satisfies:

- $0 \in F, \forall x \in \mathbb{R}, x \in F \Rightarrow -x \in F$, (zero, symmetry)
- $\forall x \in \mathbb{R}, \exists f \in \mathbb{R}, \bigtriangledown_F(x, f)$. (existence of rounding down)
Predefined Formats

**Definition (Number in radix $\beta$)**

A floating-point number is a pair $(m, e) \in \mathbb{Z}^2$ that represents the real number $m \cdot \beta^e$.

Note: no signed zeros, no infinities, no NaN.
**Predefined Formats**

**Definition (Number in radix $\beta$)**

A floating-point number is a pair $(m, e) \in \mathbb{Z}^2$ that represents the real number $m \cdot \beta^e$.

Note: no signed zeros, no infinities, no NaN.

<table>
<thead>
<tr>
<th>Format</th>
<th>is the set of all reals $x = m \cdot \beta^e$ such that</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{FIX}<em>{e</em>{\min}}$</td>
<td>$e = e_{\min}$ $</td>
</tr>
<tr>
<td>$\text{FLX}_p$</td>
<td>$x \neq 0 \Rightarrow \beta^{p-1} \leq</td>
</tr>
<tr>
<td>$\text{FLXN}_p$</td>
<td>$e_{\min} \leq e \land</td>
</tr>
<tr>
<td>$\text{FLT}<em>{p, e</em>{\min}}$</td>
<td>$x \neq 0 \Rightarrow e_{\min} \leq e \land \beta^{p-1} \leq</td>
</tr>
<tr>
<td>$\text{FTZ}<em>{p, e</em>{\min}}$</td>
<td>$x \neq 0 \Rightarrow e_{\min} \leq e \land \beta^{p-1} \leq</td>
</tr>
</tbody>
</table>
Generalizing Formats

Single parameter: \( \varphi : \mathbb{Z} \rightarrow \mathbb{Z} \).

Definition (Slice, canonical exponent, normalized mantissa)

- \( \text{slice}(x) = \lfloor \log_\beta |x| \rfloor + 1 \),
- \( \text{cexp}(x) = \varphi(\text{slice}(x)) \),
- \( \text{smant}(x) = x \cdot \beta^{-\text{cexp}(x)} \),
- \( \beta^{\text{slice}(x)-1} \leq |x| < \beta^{\text{slice}(x)}. \)
- \( x = \text{smant}(x) \cdot \beta^{\text{cexp}(x)} \).
Generalizing Formats

Single parameter: \( \varphi : \mathbb{Z} \to \mathbb{Z} \).

**Definition (Slice, canonical exponent, normalized mantissa)**

- \( \text{slice}(x) = \lfloor \log_\beta |x| \rfloor + 1 \), \( \beta^{\text{slice}(x) - 1} \leq |x| < \beta^{\text{slice}(x)} \).
- \( \text{cexp}(x) = \varphi(\text{slice}(x)) \).
- \( \text{smant}(x) = x \cdot \beta^{-\text{cexp}(x)} \), \( x = \text{smant}(x) \cdot \beta^{\text{cexp}(x)} \).

**Definition (Generic format)**

Format \( \mathbb{F}_\varphi \) is a subset of \( \mathbb{R} \) described by \( \varphi \):

\[ x \in \mathbb{F}_\varphi \iff x = \mathbb{Z}(\text{smant}(x)) \cdot \beta^{\text{cexp}(x)}. \]

Alternatively: \( x \in \mathbb{F}_\varphi \iff \text{smant}(x) \in \mathbb{Z} \).
Lemma (Validity of $\varphi$)

If the following properties hold $\forall e \in \mathbb{Z}$

$$
\varphi(e) < e \implies \varphi(e + 1) \leq e
$$

$$
e \leq \varphi(e) \implies \begin{cases} 
\varphi(\varphi(e) + 1) \leq \varphi(e), \\
\forall e', \; e' \leq \varphi(e) \implies \varphi(e') = \varphi(e),
\end{cases}
$$

then for all real $x$,

- $f = \lfloor \text{smant}(x) \rfloor \cdot \beta^{\text{cexp}(x)}$ is in $\mathbb{F}_{\varphi}$,
- any element of $\mathbb{F}_{\varphi}$ bigger than $f$ is bigger than $x$.

**Consequence:** all the usual rounding relations are meaningful!
Usual Formats

Definition (FIX)

Fixed-point format with exponent $e_{\text{min}}$: $\varphi(e) = e_{\text{min}}$. 

S. Boldo, G. Melquiond

Flocq: Floats in Coq

ARITH’20
Usual Formats

**Definition (FIX)**

Fixed-point format with exponent $e_{\text{min}}$: $\varphi(e) = e_{\text{min}}$.

**Definition (FL*)**

Floating-point format with precision $p$:
- unbounded (FLX): $\varphi(e) = e - p$, 

Usual Formats

Definition (FIX)
Fixed-point format with exponent $e_{\text{min}}$: $\varphi(e) = e_{\text{min}}$.

Definition (FL*)
Floating-point format with precision $p$:
- unbounded (FLX): $\varphi(e) = e - p$,
- bounded with subnormal numbers (FLT):
  $\varphi(e) = \max(e - p, e_{\text{min}})$,
Usual Formats

**Definition (FIX)**
Fixed-point format with exponent $e_{\text{min}}$: $\varphi(e) = e_{\text{min}}$.

**Definition (FL*)**
Floating-point format with precision $p$:
- unbounded (FLX): $\varphi(e) = e - p$,
- bounded with subnormal numbers (FLT):
  $\varphi(e) = \max(e - p, e_{\text{min}})$,
- bounded without subnormal numbers (FTZ):
  $\varphi(e) = \begin{cases} e - p & \text{if } e - p \geq e_{\text{min}}, \\ e_{\text{min}} + p - 1 & \text{otherwise}. \end{cases}$
Rounding Operators

Lemma (Rounding operators)

Let $Z \text{rnd} : \mathbb{R} \rightarrow \mathbb{Z}$ increasing such that $\forall x \in \mathbb{Z}, \ Z \text{rnd}(x) = x$.
Assuming $\varphi$ is valid, the following function is a rounding for $F_{\varphi}$:

$$x \in \mathbb{R} \mapsto Z \text{rnd}(\text{smant}(x)) \cdot \beta^{\text{cexp}(x)} \in F_{\varphi}.$$
Rounding Operators

**Lemma (Rounding operators)**

Let \( \text{Zrnd} : \mathbb{R} \rightarrow \mathbb{Z} \) increasing such that \( \forall x \in \mathbb{Z}, \text{Zrnd}(x) = x \).

Assuming \( \varphi \) is valid, the following function is a rounding for \( F_\varphi \):

\[
x \in \mathbb{R} \mapsto \text{Zrnd}(\text{smant}(x)) \cdot \beta^{\text{cexp}(x)} \in F_\varphi.
\]

**Example (Usual rounding modes)**

Toward \(-\infty\): \( \lfloor \cdot \rfloor \). Toward \(+\infty\): \( \lceil \cdot \rceil \).

Toward zero: \( \mathbb{Z}(\cdot) \). To nearest: \( \lfloor \cdot \rfloor_{\text{even}}, \lceil \cdot \rceil_{\text{away}} \).
Flushing to Zero

Example (Flush-to-zero)

Rounding to nearest number with $p$ digits but with subnormal numbers flushed to zero:

\[
\varphi(e) = \begin{cases} 
  e - p & \text{if } e - p \geq e_{\min}, \\
  e_{\min} + p - 1 & \text{otherwise}.
\end{cases}
\]

\[
Z\text{rnd}(x) = \begin{cases} 
  \lfloor x \rfloor & \text{if } |x| \geq 1, \\
  0 & \text{otherwise}.
\end{cases}
\]
Auxiliary Libraries

1. Introduction

2. Core library

3. Auxiliary libraries
   - High-level properties
   - Computable operators

4. Conclusion
High-Level Properties: Addition

**Theorem (Sterbenz)**

*Assuming that $\varphi$ is valid and nondecreasing,*

$$\forall x, y \in \mathbb{F}_\varphi, \quad \frac{y}{2} \leq x \leq 2y \Rightarrow x - y \in \mathbb{F}_\varphi.$$
High-Level Properties: Addition

**Theorem (Sterbenz)**

*Assuming that $\varphi$ is valid and nondecreasing,*

$$\forall x, y \in \mathbb{F}_\varphi, \quad \frac{y}{2} \leq x \leq 2y \Rightarrow x - y \in \mathbb{F}_\varphi.$$  

**Theorem (plus_error)**

*Assuming that $\varphi$ is valid and nondecreasing and that round$_{\varphi}^N$ rounds to nearest,*

$$\forall x, y \in \mathbb{F}_\varphi, \quad \text{round}_\varphi^N(x + y) - (x + y) \in \mathbb{F}_\varphi.$$
High-Level Properties: Addition

**Theorem (Sterbenz)**

Assuming that $\varphi$ is valid and nondecreasing,

$$\forall x, y \in F_\varphi, \quad \frac{y}{2} \leq x \leq 2y \implies x - y \in F_\varphi.$$ 

**Theorem (plus\_error)**

Assuming that $\varphi$ is valid and nondecreasing and that $\text{round}_\varphi^N$ rounds to nearest,

$$\forall x, y \in F_\varphi, \quad \text{round}_\varphi^N(x + y) - (x + y) \in F_\varphi.$$ 

Weak constraint on $\varphi \Rightarrow$ Valid for FIX, FLX, FLT.
High-Level Properties: Relative Error

**Theorem (generic_relative_error_ex)**

Assuming that $\varphi$ is valid and that there exists $p$ and $e_{\text{min}}$ such that

$$\forall k \in \mathbb{Z}, \quad e_{\text{min}} < k \Rightarrow p \leq k - \varphi(k).$$

Then, for any rounding operator $\text{round}_\varphi$ and for any real $x$ such that $\beta^{e_{\text{min}}} \leq |x|$, there exists $\varepsilon$ such that

$$|\varepsilon| < \beta^{1-p} \quad \text{and} \quad \text{round}_\varphi(x) = x \cdot (1 + \varepsilon).$$
High-Level Properties: Relative Error

**Theorem (generic_relative_error_ex)**

Assuming that $\varphi$ is valid and that there exists $p$ and $e_{\text{min}}$ such that

$$\forall k \in \mathbb{Z}, \quad e_{\text{min}} < k \Rightarrow p \leq k - \varphi(k).$$

Then, for any rounding operator $\text{round}_\varphi$ and for any real $x$ such that

$$\beta^{e_{\text{min}}} \leq |x|,$$

there exists $\varepsilon$ such that

$$|\varepsilon| < \beta^{1-p} \quad \text{and} \quad \text{round}_\varphi(x) = x \cdot (1 + \varepsilon).$$

Valid for FLX, FLT, FTZ.

Special case: $\beta^{1-p}/2$ when rounding to nearest.
High-Level Properties

Lots of other theorems:

- Error of multiplication is representable in FLX.
- Remainders of $\div$ and $\sqrt{\cdot}$ are representable in FLX.
- Some facts about ulp.
- ...
Computable Floating-point Operators

- How to round a real number \( x > 0 \)?
  - an accurate approximation \( f \) of \( x \), (guard bits if needed)
  - the relative location of \( x \) wrt \( f \). (round, sticky)
Computable Floating-point Operators

- How to **round** a real number $x > 0$?
  - an accurate approximation $f$ of $x$, (guard bits if needed)
  - the relative location of $x$ wrt $f$. (round, sticky)

- **Addition** and **multiplication**:
  - naive algorithm: compute the exact result first.

- **Division** and **square root**:
  - scale inputs so that the integer result has enough digits,
  - use the integer remainder to get the relative location.
Computable Floating-point Operators

- How to **round** a real number $x > 0$?
  - an accurate approximation $f$ of $x$, (guard bits if needed)
  - the relative location of $x$ wrt $f$. (round, sticky)

- **Addition and multiplication:**
  - naive algorithm: compute the exact result first.

- **Division and square root:**
  - scale inputs so that the integer result has enough digits,
  - use the integer remainder to get the relative location.

Generic radix, independent of formats ($\simeq$ FLX).
Format $\varphi$ is needed only at round time, if there are enough digits.
Exceptional values: signed zeros, infinities, NaN (no payload).

Conversion from/to binary interchange formats.

Computable arithmetic operators (+, −, ×, ÷, √),

“performed as if it first produced an intermediate result correct to infinite precision and with unbounded range, and then rounded that intermediate. . .”
Application: IEEE-754 Binary Arithmetic

- **Exceptional values**: signed zeros, infinities, NaN (no payload).
- Conversion from/to binary **interchange formats**.
- **Computable** arithmetic operators (+, −, ×, ÷, √),
  
  “performed as if it first produced an intermediate result correct to infinite precision and with unbounded range, and then rounded that intermediate. . .”

**Example (Square root tests from FPAccuracy)**

1055 binary64 tests.

```
Definition check inp out :=
  bits_of_b64 (b64_sqrt mode_NE (b64_of_bits inp))
  == out.
```

Whole testsuite checked in 3 seconds by Coq.
**Flocq**: 15,000 lines of Coq, 600 theorems,
- any radix, any format,
- both axiomatic and computable definitions of rounding,
- effective arithmetic operators,
- numerous theorems.
Conclusion

**Flocq**: 15 000 lines of Coq, 600 theorems,
- any radix, any format,
- both axiomatic and computable definitions of rounding,
- effective arithmetic operators,
- numerous theorems.

**Applications:**
- Frama-C/Jessie: C code certifier
- CompCert: certified C compiler
Questions?

Flocq: http://flocq.gforge.inria.fr/.