Some Formal Tools for Computer Arithmetic: Flocq and Gappa

Sylvie Boldo, Guillaume Melquiond

Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria
Laboratoire Méthodes Formelles

June 16, 2021

Supported by the European Research Council under the Horizon 2020 research and innovation programme (Grant agreement N°810367) and by the NuSCAP project of the French national research agency (ANR-20-CE48-0014).
What is a Correct Arithmetic Function?

Criteria

- Safety, e.g., access out of bounds, division by zero.
What is a Correct Arithmetic Function?

Criteria

- Safety, e.g., access out of bounds, division by zero.
- Arithmetic safety, e.g., round-off errors, unstable branching.
What is a Correct Arithmetic Function?

Criteria

- Safety, e.g., access out of bounds, division by zero.
- Arithmetic safety, e.g., round-off errors, unstable branching.
- Functional correctness, e.g., method error, correct rounding.
What is a Correct Arithmetic Function?

Criteria
- Safety, e.g., access out of bounds, division by zero.
- Arithmetic safety, e.g., round-off errors, unstable branching.
- Functional correctness, e.g., method error, correct rounding.

Approaches
- Validation, e.g., random sampling, code coverage, stochastic arithmetic.
What is a Correct Arithmetic Function?

Criteria

- Safety, e.g., access out of bounds, division by zero.
- Arithmetic safety, e.g., round-off errors, unstable branching.
- Functional correctness, e.g., method error, correct rounding.

Approaches

- Validation, e.g., random sampling, code coverage, stochastic arithmetic.
- Verification, e.g., abstract interpretation, model checking, deductive verification, formal proof.
What is a Correct Arithmetic Function?

Criteria
- Safety, e.g., access out of bounds, division by zero.
- Arithmetic safety, e.g., round-off errors, unstable branching.
- Functional correctness, e.g., method error, correct rounding.

Approaches
- Validation, e.g., random sampling, code coverage, stochastic arithmetic.
- Verification, e.g., abstract interpretation, model checking, deductive verification, formal proof.
Tools: Coq, Flocq, and Gappa

The Coq proof assistant

- A higher-order specification language to state theorems.
- A tactic language to interactively build proofs of theorems.
- A kernel to check that proofs are well-formed.
Tools: Coq, Flocq, and Gappa

The Coq proof assistant
- A higher-order specification language to state theorems.
- A tactic language to interactively build proofs of theorems.
- A kernel to check that proofs are well-formed.

Flocq: a formalization for Coq
- Radix 2, 10, other.
- Fixed- and floating-point arithmetic.
Tools: Coq, Flocq, and Gappa

The Coq proof assistant
- A higher-order specification language to state theorems.
- A tactic language to interactively build proofs of theorems.
- A kernel to check that proofs are well-formed.

Flocq: a formalization for Coq
- Radix 2, 10, other.
- Fixed- and floating-point arithmetic.

Gappa: decision procedure for computer arithmetic
- Analysis of ranges and round-off errors.
- Generation of formal proofs for Coq.
Outline

1. Introduction
2. Flocq
3. Gappa
4. Conclusion
Outline

1 Introduction

2 Flocq
   • Formats
   • Axiomatic rounding
   • Effective computations

3 Gappa

4 Conclusion
The most generic formats

Formats are just subsets of real numbers, $F \subseteq \mathbb{R}$, with a few properties that ensure rounding can be defined.
The most generic formats

Formats are just subsets of real numbers, $F \subseteq \mathbb{R}$, with a few properties that ensure rounding can be defined.

Radix and canonical exponents

A format is characterized by a radix $\beta$ and a function $\varphi \in \mathbb{Z} \rightarrow \mathbb{Z}$:

$$x \in F_{\varphi} \iff \exists m \in \mathbb{Z}, \; x = m \cdot \beta^{\varphi(e_x)}$$

with $e_x$ such that $|x| \in [\beta^{e_x-1}; \beta^{e_x}]$. 
Formats

The most generic formats
Formats are just subsets of real numbers, $F \subseteq \mathbb{R}$, with a few properties that ensure rounding can be defined.

Radix and canonical exponents
A format is characterized by a radix $\beta$ and a function $\varphi \in \mathbb{Z} \rightarrow \mathbb{Z}$:

$$x \in F_\varphi \Leftrightarrow \exists m \in \mathbb{Z}, \ x = m \cdot \beta^{\varphi(e_x)}$$

with $e_x$ such that $|x| \in [\beta^{e_x-1}; \beta^{e_x}]$.

Some classical formats
- Fixed-point: $\varphi_{\text{FIX}}(e) = e_{\text{min}}$.
- Float with gradual underflow: $\varphi_{\text{FLT}}(e) = \max(e - p, e_{\text{min}})$. 
Axiomatic Rounding

Rounding as relations

The rounding in $F$ of $x \in \mathbb{R}$ toward $-\infty$ is $f \in \mathbb{R}$ iff

$$(f \in F) \land (f \leq x) \land (\forall g \in \mathbb{R}, \ g \in F \Rightarrow g \leq x \Rightarrow g \leq f).$$
Axiomatic Rounding

Rounding as relations

The rounding in $\mathbb{F}$ of $x \in \mathbb{R}$ toward $-\infty$ is $f \in \mathbb{R}$ iff

$$(f \in \mathbb{F}) \land (f \leq x) \land (\forall g \in \mathbb{R}, \ g \in \mathbb{F} \Rightarrow g \leq x \Rightarrow g \leq f).$$

Rounding as functions

Rounding of $x$ toward $-\infty$:

$$\nabla(x) = \lfloor x \cdot \beta^{-\varphi(e_x)} \rfloor \cdot \beta^{\varphi(e_x)}.$$
Axiomatic Rounding

**Rounding as relations**

The rounding in \( F \) of \( x \in \mathbb{R} \) toward \(-\infty\) is \( f \in \mathbb{R} \) iff

\[
(f \in F) \land (f \leq x) \land (\forall g \in \mathbb{R}, g \in F \Rightarrow g \leq x \Rightarrow g \leq f).
\]

**Rounding as functions**

Rounding of \( x \) toward \(-\infty\):

\[
\nabla(x) = \lfloor x \cdot \beta^{-\varphi(e_x)} \rfloor \cdot \beta^{\varphi(e_x)}.
\]

**Some simple properties**

- \( x \notin F_\varphi \Rightarrow \triangle(x) = \nabla(x) + \text{ulp}(x) \) with \( \text{ulp}(x) = \beta^{\varphi(e_x)} \).
- \( |\circ \tau(x) - x| \leq \frac{1}{2} \text{ulp}(\circ \tau(x)) \).
A few Theorems of Flocq

- Sterbenz’ lemma.
A few Theorems of Flocq

- Sterbenz’ lemma.
- Error-free transformations for $+$, $\times$, $\div$, $\sqrt{}$, e.g.,

$$x \in \mathbb{F} \implies x - (\circ \tau(\sqrt{x}))^2 \in \mathbb{F}$$

for $\mathbb{F}$ without underflow.
A few Theorems of Flocq

- Sterbenz’ lemma.

- Error-free transformations for $+,	imes,\div,\sqrt{\cdot}$, e.g.,
  \[ x \in \mathbb{F} \Rightarrow x - (\circ^\tau(\sqrt{x}))^2 \in \mathbb{F} \quad \text{for } \mathbb{F} \text{ without underflow}. \]

- Innocuous double rounding, even in presence of underflow:
  \[ x, y \in \mathbb{F}_p \Rightarrow \circ^{\tau_1}_p (\circ^{\tau_2}_{p'}(x/y)) = \circ^{\tau_1}_{p'}(x/y) \quad \text{with } 2p \leq p'. \]
Effective Computations

- Effective algorithms for division and square root.
Effective Computations

- Effective algorithms for division and square root.
- Signed zeroes, infinities, Not-a-Numbers. (CompCert)
Effective Computations

- Effective algorithms for division and square root.
- Signed zeroes, infinities, Not-a-Numbers. (CompCert)
- Native binary64 numbers in proofs. (CoqInterval)
Outline

1. Introduction

2. Flocq

3. Gappa
   - Formulas and predicates
   - Database of theorems
   - User hints

4. Conclusion
Input formulas for Gappa

∀x₁, ..., xₖ ∈ ℝ, e₁ ∈ I₁ ∧ ... ∧ eₙ ∈ Iₙ ⇒ e ∈ ?

with e₁, ..., eₙ, e arithmetic expressions with rounding functions, and I₁, ..., Iₙ intervals with numerical bounds.
Formulas and Predicates

Input formulas for Gappa

\[ \forall x_1, \ldots, x_k \in \mathbb{R}, \ e_1 \in I_1 \land \ldots \land e_n \in I_n \Rightarrow e \in ? \]

with \( e_1, \ldots, e_n, e \) arithmetic expressions with rounding functions, and \( I_1, \ldots, I_n \) intervals with numerical bounds.

Supported predicates

\[
\begin{align*}
    \text{BND}(x, I) & \triangleq x \in I, \\
    \text{FIX}(x, n) & \triangleq \exists m \in \mathbb{Z}, \ x = m \cdot 2^n, \\
    \text{FLT}(x, n) & \triangleq \exists m, e \in \mathbb{Z}, \ x = m \cdot 2^e \land |m| < 2^n, \\
    \text{REL}(\tilde{x}, x, I) & \triangleq \exists \epsilon \in \mathbb{R}, \ \tilde{x} = x \cdot (1 + \epsilon) \land \epsilon \in I \\
    & \approx \text{BND}((\tilde{x} - x)/x, I).
\end{align*}
\]
Database of Theorems

Gappa’s process

Saturate the set of arithmetic facts using theorems, until a fixed point is reached or a contradiction is found.
Database of Theorems

**Gappa’s process**
Saturate the set of arithmetic facts using theorems, until a fixed point is reached or a contradiction is found.

**Some theorems known by Gappa**
- Real arithmetic: $\text{BND}(u) \land \text{BND}(v) \Rightarrow \text{BND}(u \diamond v)$. 
Database of Theorems

Gappa’s process
Saturate the set of arithmetic facts using theorems, until a fixed point is reached or a contradiction is found.

Some theorems known by Gappa
- Real arithmetic: \( \text{BND}(u) \land \text{BND}(v) \Rightarrow \text{BND}(u \diamond v) \).
- Rounding functions:
  \( \text{FIX}(\circ(u)), \text{FLT}(\circ(u)), \text{BND}(\circ(u) - u), \text{REL}(\circ(u), u) \).
Database of Theorems

Gappa’s process

Saturate the set of arithmetic facts using theorems, until a fixed point is reached or a contradiction is found.

Some theorems known by Gappa

- Real arithmetic: \( BND(u) \land BND(v) \Rightarrow BND(u \diamond v) \).
- Rounding functions:
  
  - \( \text{FIX}(\circ(u)) \), \( \text{FLT}(\circ(u)) \), \( BND(\circ(u) - u) \), \( \text{REL}(\circ(u), u) \).
- Error propagation:
  
  - \( \text{REL}(u, v) \land \text{REL}(v, w) \Rightarrow \text{REL}(u, w) \);
  
  - \( \text{REL}(\tilde{u}, u) \land \text{REL}(\tilde{v}, v) \land BND(u/(u + v)) \Rightarrow \text{REL}(\tilde{u} + \tilde{v}, u + v) \).
Newton iteration for $y_n \rightarrow a^{-1}$

\[
\begin{align*}
\delta_n & \leftarrow \circ (1 - a \cdot y_n), \\
y_{n+1} & \leftarrow \circ (y_n + y_n \cdot \delta_n).
\end{align*}
\]
User Hints

Newton iteration for $y_n \rightarrow a^{-1}$

$$\delta_n \leftarrow \circ(1 - a \cdot y_n),$$
$$y_{n+1} \leftarrow \circ(y_n + y_n \cdot \delta_n).$$

Helping Gappa

1. Define $\bar{y}_{n+1} = y_n + y_n \cdot (1 - a \cdot y_n)$ and $\varepsilon_n = (y_n - a^{-1})/a^{-1}$.
2. State $(\bar{y}_{n+1} - a^{-1})/a^{-1} = -\varepsilon_n^2.$
Outline

1. Introduction
2. Flocq
3. Gappa
4. Conclusion
   - Tools
   - Applications
Some Formal Tools

- **Flocq**: formalization for computer arithmetic.
- **Gappa**: decision procedure for computer arithmetic.
- **Coquelicot**: formalization of real analysis.
- **CoqInterval**: decision procedure for real analysis.
Applications

- Order-2 discriminant; area of a triangle; correctly-rounded average.
- FastTwoSum, TwoSum; error of an FMA.
- Some elementary functions.
- CompCert C compiler.
- 3-point numerical scheme for the 1D wave equation.