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

Outline

Formats

The most generic formats

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

Formats

The most generic formats

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

Radix and canonical exponents

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

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

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

Formats

The most generic formats

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

Radix and canonical exponents

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

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

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

Some classical formats

- Fixed-point: $\varphi_{\mathsf{FIX}}(e) = e_{\min}$.
- Float with gradual underflow: $\varphi_{\mathsf{FLT}}(e) = \max(e - p, e_{\min})$.

Axiomatic Rounding

Rounding as relations

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

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

Axiomatic Rounding

Rounding as relations

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

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

Rounding as functions

Rounding of x toward $-\infty$:

$$\bigtriangledown(x) = \left\lfloor x \cdot \beta^{-\varphi(e_x)} \right\rfloor \cdot \beta^{\varphi(e_x)}.$$

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 $-\infty$:

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

Some simple properties

- $x \notin \mathbb{F}_\varphi \Rightarrow \triangle(x) = \triangledown(x) + \mathrm{ulp}(x)$ with $\mathrm{ulp}(x) = \beta^{\varphi(e_x)}$.
- $|\circ^\tau(x) - x| \leq \frac{1}{2}\,\mathrm{ulp}(\circ^\tau(x))$.

A few Theorems of Flocq

- Sterbenz' lemma.

A few Theorems of Flocq

- Sterbenz' lemma.
- Error-free transformations for $+$, $\times$, $\div$, $\sqrt{\cdot}$, e.g.,
  $x \in \mathbb{F} \Rightarrow x - (\circ^\tau(\sqrt{x}))^2 \in \mathbb{F}$ for $\mathbb{F}$ without underflow.

A few Theorems of Flocq

- Sterbenz' lemma.
- Error-free transformations for $+$, $\times$, $\div$, $\sqrt{\cdot}$, e.g.,
  $x \in \mathbb{F} \Rightarrow x - (\circ^{\tau}(\sqrt{x}))^2 \in \mathbb{F}$ for $\mathbb{F}$ without underflow.
- Innocuous double rounding, even in presence of underflow:
  $x, y \in \mathbb{F}_p \Rightarrow \circ_p^{\tau_1}(\circ_{p'}^{\tau_2}(x/y)) = \circ_p^{\tau_1}(x/y)$ 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

Formulas and Predicates

Input formulas for Gappa

$$\forall x_1, \ldots, x_k \in \mathbb{R},\ e_1 \in I_1 \wedge \ldots \wedge e_n \in I_n \Rightarrow e \in \text{?}$$

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

Formulas and Predicates

Input formulas for Gappa

$$\forall x_1, \ldots, x_k \in \mathbb{R},\ e_1 \in I_1 \wedge \ldots \wedge e_n \in I_n \Rightarrow e \in \text{?}$$

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

Supported predicates

$$\begin{aligned}
\mathrm{BND}(x, I) &\triangleq x \in I, \\
\mathrm{FIX}(x, n) &\triangleq \exists m \in \mathbb{Z},\ x = m \cdot 2^n, \\
\mathrm{FLT}(x, n) &\triangleq \exists m, e \in \mathbb{Z}, x = m \cdot 2^e \wedge |m| < 2^n, \\
\mathrm{REL}(\tilde{x}, x, I) &\triangleq \exists \varepsilon \in \mathbb{R},\ \tilde{x} = x \cdot (1 + \varepsilon) \wedge \varepsilon \in I \\
&\simeq \mathrm{BND}((\tilde{x} - x)/x, I).
\end{aligned}$$

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: $\text{BND}(u) \wedge \text{BND}(v) \Rightarrow \text{BND}(u \diamond v)$.

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: $\text{BND}(u) \wedge \text{BND}(v) \Rightarrow \text{BND}(u \diamond v)$.
- 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: $\mathrm{BND}(u) \wedge \mathrm{BND}(v) \Rightarrow \mathrm{BND}(u \diamond v)$.
- Rounding functions:
  $\mathrm{FIX}(\circ(u))$, $\mathrm{FLT}(\circ(u))$, $\mathrm{BND}(\circ(u) - u)$, $\mathrm{REL}(\circ(u), u)$.
- Error propagation: $\mathrm{REL}(u, v) \wedge \mathrm{REL}(v, w) \Rightarrow \mathrm{REL}(u, w)$;
  $\mathrm{REL}(\tilde{u}, u) \wedge \mathrm{REL}(\tilde{v}, v) \wedge \mathrm{BND}(u/(u+v)) \Rightarrow \mathrm{REL}(\tilde{u}+\tilde{v}, u+v)$.

User Hints

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

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

User Hints

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

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

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

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.