

PRECiSA: from real specifications to floating-point implementations



SUMMARY

- Writing floating-point programs is challenging!
- Rounding errors are silent: you still get a number, but it may be very different from the expected one.

```
Prelude> (4/3 - 1) * 3 - 1
-2.220446049250313e-16
Prelude> floor((4/3 - 1) * 3 - 1)
-1
Prelude> if (floor((4/3 - 1) * 3 - 1)) < 0 then 100 else 1
100
```

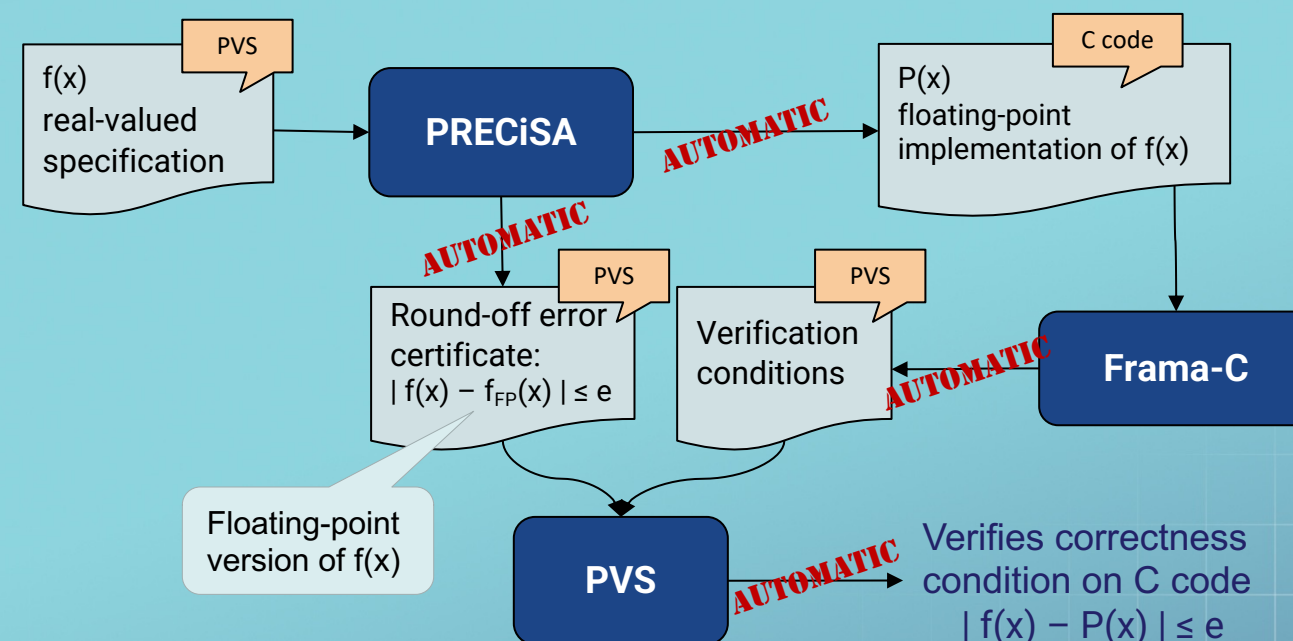
The expected result is 1
Error = 99!

- PRECiSA automatically generates a floating-point implementation from a given real number specification;
- Provides rigorous guarantees on the floating-point rounding error that may occur in this implementation;
- Instruments the code to detect discrepancies between the ideal real valued program control flow and the floating-point one.
- No expertise in theorem proving or floating-point arithmetic is required to apply the toolchain.

FUTURE WORK

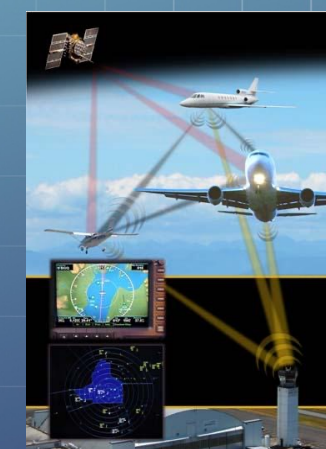
- Verification of machine learning applications
- Improvement of the scalability of the tool
- Supporting a broader variety of programs

PRECISA INTEGRATED TOOLCHAIN



PAST APPLICATIONS

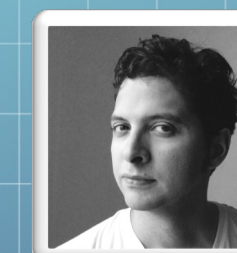
- ADS-B Compact Position Reporting (CPR)
- Geofencing UAV
- Detect-and-avoid



TEAM



Laura Titolo
Sr Research Scientist
laura.titolo@nianet.org



Mariano Moscato
Sr Research Scientist
mariano.moscato@nianet.org



Marco Feliu
Sr Research Scientist
marco.feliu@nianet.org