Univariate and Bivariate Integral Roots Certificates Based on Hensel Lifting[⋆]

Érik Martin-Dorel

École Normale Supérieure de Lyon
LIP (UMR 5668 CNRS, ENSL, INRIA, UCBL)
46 allée d'Italie, F-69364 Lyon Cedex 07

erik.martin-dorel@ens-lyon.org

1 Introduction and Motivations

The newly revised IEEE 754–2008 standard for *floating-point* (FP) arithmetic recommends that some mathematical functions (exp, log, $x \mapsto 2^x$, ...) should be correctly rounded (roughly speaking, the system must always return the FP number nearest to the exact mathematical result of the operation). Requiring correctly rounded functions has a number of advantages: among them, it greatly improves the portability of numerical software and it allows one to design algorithms and formal proofs of software that use this requirement. To be able to design fast programs for correctly rounded functions, we must address a problem called the *Table Maker's Dilemma* (TMD) [1, chap. 12]. We need to locate, for each considered function f and for each considered FP format and rounding mode, the *hardest-to-round* (HR) points, that is, in rounding-to-nearest, what are the FP numbers x such that $f(x)$ is closest to the exact middle of two consecutive FP numbers. The naive method of finding these points (evaluating the function with large precision at each FP number) is far too impractical.

Two algorithms (Lefèvre [2] and SLZ [3]) have been designed to enumerate these HR points, however; however, they are based on complex and very long calculations (years of cumulated CPU time) that inevitably cast some doubt on the correctness of their results. In the French ANR project entitled TaMaDi, we thus undertake to fully reconsider the methods used to get HR points, with a special focus on their formal validation (by enabling our programs to generate certificates that guarantee the validity of their results).

In the sequel, we focus on the certification of the last step of the SLZ algorithm, namely finding all the solutions of the systems

$$\begin{cases} P(x,y) = 0, \\ Q(x,y) = 0, \\ |x| \leqslant A \text{ and } |y| \leqslant B, \end{cases}$$

for each (P, Q, A, B) generated in previous steps ($P, Q \in \mathbb{Z}[X, Y]$; $x, y, A, B \in \mathbb{Z}$).

[⋆] This work was partially funded by the TaMaDi project of the French ANR.

For this purpose, we devise a type of integral roots certificates and the corresponding checker specification, which are based on Hensel lifting, also known as the p-adic Newton iteration [4].

In this talk, we present a formalization within the Coq proof assistant along with the SSReflect extension [5] of both univariate and bivariate Hensel lifting with a uniqueness property from which we deduce a formal proof of correctness of the integral-roots-certificates checkers.

2 Univariate Integral-Roots Certificates

For the sake of simplicity, we present below only the univariate version of our certificates.

We define a univariate integral-roots certificate as a 5-tuple (P, B, p, k, L) whose type is given by the following Coq `Record`:

```
Record univCertif := UnivCertif {
  uc_P : {poly Z};
  uc_B : N;
  uc_p : nat;
  uc_k : nat;
  uc_L : seq (Z * bool)
}.
```

We will say one such certificate (P, B, p, k, L) is valid if the following conditions are fulfilled:

$$p \text{ is a prime number,} \tag{1}$$

$$\text{the integer } k \geqslant 0 \text{ satisfies } p^{2^k} > 2 \cdot B, \tag{2}$$

and denoting $L_p = \{u \bmod p \mid \exists b \in \texttt{bool},\ (u, b) \in L\}$, we have:

$$\forall s \in \{0, 1, \ldots, p-1\},\ \ s \in L_p \iff P(s) \equiv 0 \pmod{p}, \tag{3}$$

$$\text{the } \ell \text{ elements of } L_p \text{ are pairwise distinct,} \tag{4}$$

and for all $(u, b) \in L$, we have:

$$P'(u) \not\equiv 0 \pmod{p}, \tag{5}$$

$$|2 \cdot u| \leqslant p^{2^k}, \tag{6}$$

$$P(u) \equiv 0 \pmod{p^{2^k}}, \tag{7}$$

$$b = \texttt{true} \iff \big(|u| \leqslant B \ \wedge\ P(u) = 0\big). \tag{8}$$

Intuitively, the Boolean values "b" indicate whether the *modular roots* of the polynomial are effectively some *integral roots* or not.

All these Boolean conditions are formalized in the form of a Coq function `univ_check : univCertif -> bool` representing the univariate integral-roots-certificates checker.

3 Formal Verification of the Univariate Checker

The correctness proof of the univariate checker consists of proving that any certificate that is accepted by the checker contains all the integral roots of the considered polynomial in the considered range.

To be more precise, we need to prove that *for all* `uc` $= (P, B, p, k, L)$ *such that* `(univ_check uc)` *holds, for all* $x \in \mathbb{Z}$ *we have the equivalence*

$$\big(|x| \leqslant B \ \wedge\ P(x) = 0\big) \iff x \in L' := \{u \mid (u, \mathsf{true}) \in L\}\,. \tag{9}$$

The proof of "$\Longrightarrow$" relies on the following lemma:

Lemma 1. *Let* $P \in \mathbb{Z}[X]$ *and* $p \in \mathbb{P}$ *that satisfies*

$$\forall z \in \mathbb{Z}, \quad P(z) \equiv 0 \pmod{p} \implies P'(z) \not\equiv 0 \pmod{p}, \tag{10}$$

where P' *is the derivative of the polynomial* P*. If* $x \in \mathbb{Z}$ *is such that*

$$P(x) \equiv 0 \pmod{p^{2^m}} \tag{11}$$

for a given $m \in \mathbb{N}$*, then for*

$$u_0 := x \bmod p, \tag{12}$$

the sequence (u_k) *defined by the recurrence relation*

$$\forall k \in \{0, 1, \ldots, m-1\}\,, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}} \tag{13}$$

satisfies:

$$\forall k \in \{0, 1, \ldots, m\}\,, \quad u_k = x \bmod p^{2^k}\,. \tag{14}$$

First, we prove x is a root modulo p, from which we deduce $x \bmod p \in L_p$, hence there exists some $(u, b) \in L$ such that $x \bmod p = u \bmod p$, so that we can apply Lemma 1 that can be viewed as a *uniqueness* result for Hensel lifting: it says the roots of P modulo powers of p are fully determined from the knowledge of the roots modulo p.

As regards the proof of Lemma 1, it uses Taylor's theorem for polynomials and it is fully described in [6]. Note that there is no constraint on the degree of P. Yet Hypothesis (10) assumes the prime p has been chosen so that P has no repeated roots modulo p.

For the "$\Longrightarrow$" part of (9), we need to verify the values stored in L' are true *integral roots,* which is indeed the case given the definition of our verifier.

4 Some Remarks on the Bivariate Case

As we focus on the certification of SLZ that deals with pairs of bivariate polynomials (P_1, P_2) with integer coefficients, we devise some *bivariate* integral-roots certificates whose structure is very similar to the one presented in Sections 2 and 3. A major change in the bivariate version is that the condition (5) is replaced with

$$\det J_{P_1,P_2}(u,v) \not\equiv 0 \pmod{p},$$

where $J_{P_1,P_2}(a,b)$ denotes the Jacobian matrix of $(P_1,P_2) \in (\mathbb{Z}[X,Y])^2$ evaluated at $(a,b) \in \mathbb{Z}^2$. In other words,

$$J_{P_1,P_2}(a,b) = \begin{pmatrix} \frac{\partial P_1}{\partial X}(a,b) & \frac{\partial P_1}{\partial Y}(a,b) \\ \frac{\partial P_2}{\partial X}(a,b) & \frac{\partial P_2}{\partial Y}(a,b) \end{pmatrix}.$$

Furthermore, we state and prove the lemma corresponding to the bivariate version of Lemma 1 by using some new material specific to 2-by-2 matrices, as well as the Taylor theorem for bivariate polynomials that Laurent Théry has formalized in Coq/SSReflect. Again, we use this uniqueness lemma to derive the proof of correctness of our (bivariate) integral-roots-certificates checker.

5 Conclusion and Perspectives

The approach we followed amounts to viewing Hensel lifting as a so-called *certifying algorithm* [7]. Compared to the formal verification of a traditional algorithm (for providing what we can call a *certified* algorithm), the certificate-based approach that relies on a *certifying* algorithm ensures that the computed result has not been compromised by any bug. Moreover, with this approach we do not need to verify the implemented program nor the algorithm itself. This means that we could even use a “fast-and-dirty program” to do the job, since the result can be easily checked by the certificate verifier that has been formally proved in Coq. In compensation the checker itself has to be somewhat efficient since the approach rely on the individual verification of each result.

In particular, we had to work around the fact some data structures chosen for our proofs are somewhat not computational. Now we can for instance check a certificate for a degree-9 dense univariate polynomial with Coq 8.2 in less than 5 ms. But since the ultimate goal of this work is to provide a component that will be involved in a full certification chain for SLZ, we might need to use more efficient data structures in order to facilitate the verification of our certificates within Coq.

References

1. J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres, *Handbook of Floating-Point Arithmetic.* Birkhäuser, 2009.
2. V. Lefèvre and J.-M. Muller, “Worst Cases for Correct Rounding of the Elementary Functions in Double Precision,” in *Proceedings of the 15th IEEE Symposium on Computer Arithmetic* (N. Burgess and L. Ciminiera, eds.), (Vail, CO), pp. 111–118, June 2001.

3. D. Stehlé, V. Lefèvre, and P. Zimmermann, “Searching Worst Cases of a One-Variable Function Using Lattice Reduction,” *IEEE Transactions on Computers*, vol. 54, pp. 340–346, Mar. 2005.
4. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge University Press, 2003.
5. G. Gonthier and A. Mahboubi, “A Small Scale Reflection Extension for the Coq system,” Research Report RR-6455, INRIA, 2009.
6. Érik Martin-Dorel, “Univariate and bivariate integral roots certificates based on Hensel’s lifting,” Research Report RRLIP2011-1, LIP, ENS de Lyon, 2011.
7. R. M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer, “Certifying Algorithms.” To appear in Computer Science Review, June 2010.