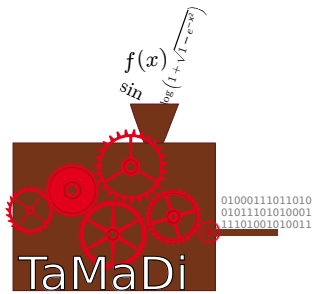
CoqHensel: from Hensel's lemma to the verification of ISValP certificates in Coq

Érik Martin-Dorel
erik.martin-dorel@lri.fr

Postdoc, team Toccata, LRI, Inria Saclay - Île-de-France

Joint works with Guillaume Hanrot, Micaela Mayero, Laurent Théry

Final meeting of the TaMaDi project
Tuesday 8th October 2013
ENS de Lyon



Overview of the Stehlé–Lefèvre–Zimmermann algorithm

$$\text{SLZ} = \text{Polynomial Approximation} + \text{Coppersmith's technique} + \text{Bivariate Hensel lifting}$$

Overview of the Stehlé–Lefèvre–Zimmermann algorithm

$$\text{SLZ} = \text{Polynomial Approximation} + \boxed{\begin{array}{l}\text{Coppersmith's technique} \\ + \text{ Bivariate Hensel lifting}\end{array}}$$

- Sophisticated algorithms, with highly optimized implementations
- Rely on many tools and libraries (SAGE, fpLLL, GMP, ...)
- Very long calculations (several years of CPU time)

Overview of the Stehlé–Lefèvre–Zimmermann algorithm

$$\text{SLZ} = \text{Polynomial Approximation} + \boxed{\begin{array}{l}\text{Coppersmith's technique} \\ + \text{ Bivariate Hensel lifting}\end{array}}$$

- Sophisticated algorithms, with highly optimized implementations
- Rely on many tools and libraries (SAGE, fpLLL, GMP, ...)
- Very long calculations (several years of CPU time)

→ **Goal:** independent verification of the results by means of certificates.

Overview of the Stehlé–Lefèvre–Zimmermann algorithm

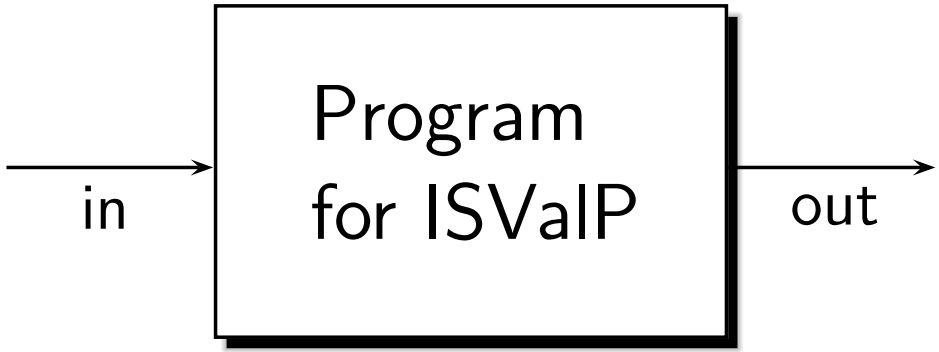
Integer Small Value Problem (ISValP)

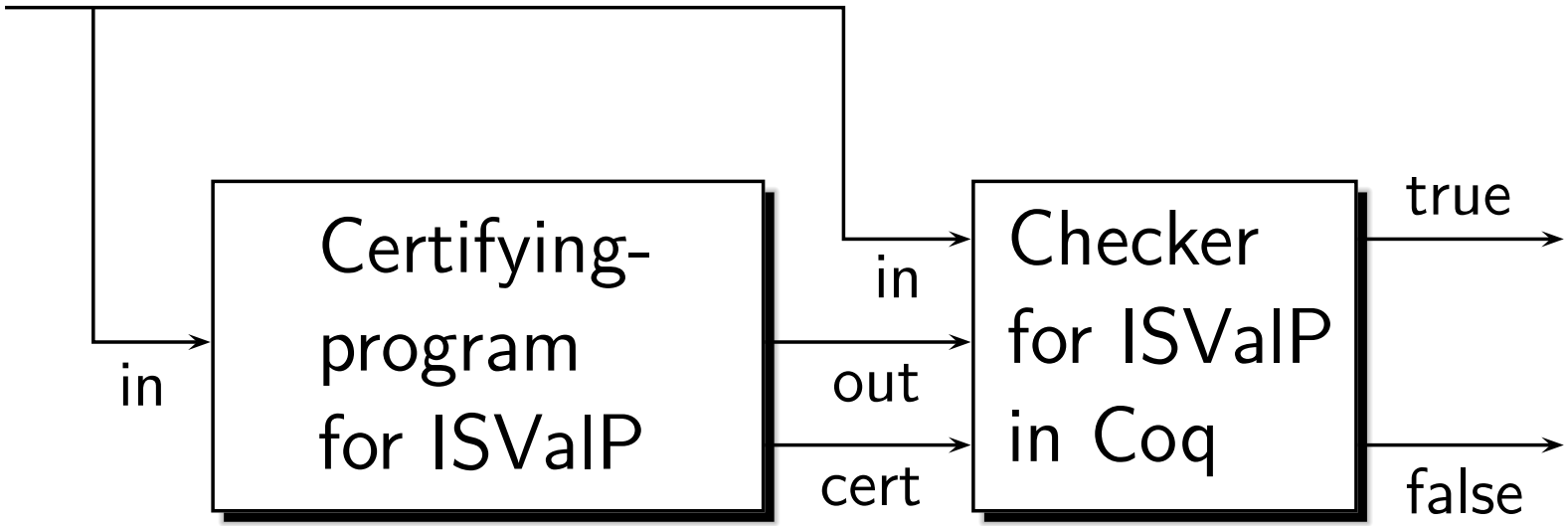
SLZ = Polynomial Approximation + Coppersmith's technique + Bivariate Hensel lifting

- Sophisticated algorithms, with highly optimized implementations
- Rely on many tools and libraries (SAGE, fpLLL, GMP, ...)
- Very long calculations (several years of CPU time)

→ **Goal:** independent verification of the results by means of certificates.

From SLZ to certificates for Integer Small Value Problems





Goal: Solving the Integer Small Value Problem (ISValP)

Integer Small Value Problem (ISValP)

For given $P \in \mathbb{Z}[X]$ and $A, B, M \in \mathbb{N}$, find *all* solutions of the system

$$\begin{cases} (x, y) \in \mathbb{Z}^2, \\ |x| \leqslant A, \\ |y| \leqslant B, \\ P(x) \equiv y \pmod{M}. \end{cases}$$

Goal: Solving the Integer Small Value Problem (ISValP)

Heuristically, Coppersmith's technique reduces instances of ISValP to:

Bivariate Small-Integral-Roots Problem

For given $v_1, v_2 \in \mathbb{Z}[X, Y]$ (and $A, B \in \mathbb{N}$), find *all* solutions of the system

$$\begin{cases} (x, y) \in \mathbb{Z}^2, \\ |x| \leqslant A, \\ |y| \leqslant B, \\ v_1(x, y) = 0 = v_2(x, y) \end{cases}$$

Solving ISValP: Coppersmith's technique and Hensel lifting

Heuristically, Coppersmith's technique reduces instances of ISValP to:

Bivariate Small-Integral-Roots Problem

For given $v_1, v_2 \in \mathbb{Z}[X, Y]$ (and $A, B \in \mathbb{N}$), find *all* solutions of the system

$$\begin{cases} (x, y) \in \mathbb{Z}^2, \\ |x| \leqslant A, \\ |y| \leqslant B, \\ v_1(x, y) = 0 = v_2(x, y) \end{cases}$$

This is where bivariate Hensel lifting comes into play.

Outline

Outline

Focus on the Univariate Small-Integral-Roots Problem

Univariate Small-Integral-Roots Problem

For given $v \in \mathbb{Z}[X]$ and $B \in \mathbb{N}$, find *all* solutions of the system

$$\begin{cases} x \in \mathbb{Z}, \\ |x| \leqslant B, \\ v(x) = 0 \end{cases}$$

Focus on the Univariate Small-Integral-Roots Problem

Univariate Small-Integral-Roots Problem

For given $v \in \mathbb{Z}[X]$ and $B \in \mathbb{N}$, find *all* solutions of the system

$$\begin{cases} x \in \mathbb{Z}, \\ |x| \leqslant B, \\ v(x) = 0 \end{cases}$$

Use Hensel lifting ($= p$-adic Newton iteration) to compute the solutions.

Use Hensel's lemma to certify that all solutions have been found.

Generating a certificate using univariate Hensel lifting

$$v(X) := X^4 + 265X^3 - 6005X^2 + 265X - 6006 \in \mathbb{Z}[X], \quad B := 1024.$$

Let us choose a small prime, for example $p := 5$.
The modulo-5 roots of v are $\{1, 2, -2, -1\}$, that is $v(1) \equiv 0 \pmod 5$, etc.

Generating a certificate using univariate Hensel lifting

$$v(X) := X^4 + 265X^3 - 6005X^2 + 265X - 6006 \in \mathbb{Z}[X], \quad B := 1024.$$

Let us choose a small prime, for example $p := 5$.
The modulo-5 roots of v are $\{1, 2, -2, -1\}$, that is $v(1) \equiv 0 \pmod 5$, etc.

$p^{2^3} > 2\,B \rightsquigarrow$ for each modulo-5 root, we iterate Hensel lifting $k := 3$ times

a_0	a_1	a_2	a_3
1	-4	21	21
2	7	182	-110443
-2	-7	-182	110443
-1	-11	-286	-286

$$a_{i+1} := a_i - \frac{v(a_i)}{v'(a_i)} \operatorname{cmod} p^{2^{i+1}}$$

Generating a certificate using univariate Hensel lifting

$$v(X) := X^4 + 265X^3 - 6005X^2 + 265X - 6006 \in \mathbb{Z}[X], \quad B := 1024.$$

Let us choose a small prime, for example $p := 5$.
The modulo-5 roots of v are $\{1, 2, -2, -1\}$, that is $v(1) \equiv 0 \pmod 5$, etc.

$p^{2^3} > 2\,B \rightsquigarrow$ for each modulo-5 root, we iterate Hensel lifting $k := 3$ times

a_0	a_1	a_2	a_3
1	-4	21	21
2	7	182	-110443
-2	-7	-182	110443
-1	-11	-286	-286

$$a_{i+1} := a_i - \frac{v(a_i)}{v'(a_i)} \operatorname{cmod} p^{2^{i+1}}$$

$$y = x \operatorname{cmod} m \Leftrightarrow \begin{cases} y \equiv x \pmod m \\ |y| \leqslant \frac{m}{2} \end{cases}$$

Generating a certificate using univariate Hensel lifting

$v(X) := X^4 + 265X^3 - 6005X^2 + 265X - 6006 \in \mathbb{Z}[X], \quad B := 1024.$

Let us choose a small prime, for example $p := 5$.
The modulo-5 roots of v are $\{1, 2, -2, -1\}$, that is $v(1) \equiv 0 \pmod 5$, etc.

$p^{2^3} > 2\,B \rightsquigarrow$ for each modulo-5 root, we iterate Hensel lifting $k := 3$ times

a_0	a_1	a_2	a_3
1	-4	21	21
2	7	182	-110443
-2	-7	-182	110443
-1	-11	-286	-286

$$a_{i+1} := a_i - \frac{v(a_i)}{v'(a_i)} \operatorname{cmod} p^{2^{i+1}}$$

$$y = x \operatorname{cmod} m \Leftrightarrow \begin{cases} y \equiv x \pmod m \\ |y| \leqslant \frac{m}{2} \end{cases}$$

We provide the certificate (v, B, p, k, L) with
$L := ((21, \text{true}); (-110443, \text{false}); (110443, \text{false}); (-286, \text{true}))$.

Soundness of the Small-Integral-Roots checker

Univariate small-integral-roots checker: a Coq function that takes (v, B, p, k, L) as input and returns a Boolean (accept/reject the certificate)

Theorem

For any certificate (v, B, p, k, L) that is accepted, we have

$$\forall x \in \mathbb{Z}, \quad \big(|x| \leqslant B \text{ and } v(x) = 0\big) \iff (x, \mathsf{true}) \in L.$$

The proof of $(\Longrightarrow)$ relies on the uniqueness statement of Hensel's lemma, whereas the $(\Longleftarrow)$ step is trivial.

Generalization to the bivariate case

Major changes:

- Deal with the simultaneous roots of two bivariate polynomials over $\mathbb{Z}$;

Generalization to the bivariate case

Major changes:

- Deal with the simultaneous roots of two bivariate polynomials over $\mathbb{Z}$;
- The iteration becomes:

$$\begin{pmatrix} a_{i+1} \\ b_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} a_i \\ b_i \end{pmatrix} - \left[J_{v_1,v_2}(a_i, b_i) \right]^{-1}_{p^{2^{i+1}}} \begin{pmatrix} v_1(a_i, b_i) \\ v_2(a_i, b_i) \end{pmatrix} \bmod p^{2^{i+1}}$$

Generalization to the bivariate case

Major changes:

- Deal with the simultaneous roots of two bivariate polynomials over $\mathbb{Z}$;
- The iteration becomes:

$$\begin{pmatrix} a_{i+1} \\ b_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} a_i \\ b_i \end{pmatrix} - \left[J_{v_1,v_2}(a_i, b_i) \right]^{-1}_{p^{2^{i+1}}} \begin{pmatrix} v_1(a_i, b_i) \\ v_2(a_i, b_i) \end{pmatrix} \operatorname{cmod} p^{2^{i+1}}$$

- The condition $v'(a) \not\equiv 0 \pmod{p}$ is replaced with:

$$\det J_{v_1,v_2}(a, b) \not\equiv 0 \ (\text{mod } p).$$

Generalization to the bivariate case

Major changes:

- Deal with the simultaneous roots of two bivariate polynomials over $\mathbb{Z}$;
- The iteration becomes:

$$\begin{pmatrix} a_{i+1} \\ b_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} a_i \\ b_i \end{pmatrix} - \left[J_{v_1,v_2}(a_i, b_i) \right]^{-1}_{p^{2^{i+1}}} \begin{pmatrix} v_1(a_i, b_i) \\ v_2(a_i, b_i) \end{pmatrix} \operatorname{cmod} p^{2^{i+1}}$$

- The condition $v'(a) \not\equiv 0 \pmod{p}$ is replaced with:

$$\det J_{v_1,v_2}(a, b) := \begin{vmatrix} \frac{\partial v_1}{\partial x}(a, b) & \frac{\partial v_1}{\partial y}(a, b) \\ \frac{\partial v_2}{\partial x}(a, b) & \frac{\partial v_2}{\partial y}(a, b) \end{vmatrix} \not\equiv 0 \pmod{p}.$$

Outline

Outline of the ISValP certificate checker

$\texttt{check\_ISValP}\,(P, A, B, M, \alpha, u_1, u_2, p, k, L) \in \{\text{true}, \text{false}\}$

- Compute $Q(X, Y) := P(Y) - X \in \mathbb{Z}[X, Y]$
- For $l \in \{1, 2\}$, Compute $v_l := M^\alpha \cdot u_l(Q(X, Y)/M, Y)$
- Check if

$$\begin{cases} \displaystyle\sum_{i,j} |v_{1;i,j}|\, A^i B^j < M^\alpha, \\ \displaystyle\sum_{i,j} |v_{2;i,j}|\, A^i B^j < M^\alpha, \\ (v_1, v_2, A, B, p, k, L) \text{ is a valid bivariate SIntRootP certificate.} \end{cases}$$

Soundness of the ISValP checker

Theorem

For any certificate $(P, A, B, M, \alpha, u_1, u_2, p, k, L)$ that is accepted, we have

$$\forall (x, y) \in [\![-A, A]\!] \times [\![-B, B]\!], \ P(y) \equiv x \ (\mathsf{mod}\ M) \implies (x, y, \mathsf{true}) \in L.$$

Outline

Tools and libraries used to get efficient certificate checkers

- Implement generic effective checkers that are proved correct w.r.t. the reference (non-computational) implementation
- Rely on the module system of Coq
- Functor `CalcISValP` parameterized by an implementation of integers: we can use
  - `ZArith`,
  - `BigZ`,
  - or IPPE (Integers Plus Positive Exponent): unevaluated dyadic numbers $m \times 2^e$ with $e \geqslant 0$
    $\rightsquigarrow$ functor that relies on the Flocq and Coq.Interval libraries

Benchmarks for the ISValP checker (for function $\exp$)

Summary of the test-suite

- $n := 53$, $n' := 300$, $\alpha := 13$,
- 4096 ISValP certificates addressing the full range of an exponent,
- ≈ 100 MB of data

The generation of each certificate took $\approx 6\,\text{mn}$.

Class of certificates	Average CPU time $\pm$ std deviation, using native-coq with IPPE integers
4090 certs. s.t. $\texttt{size}(L) = 0$	$34.6\,\text{s} \pm 6.6\,\text{s}$
6 certs. s.t. $\texttt{size}(L) = 1$	$221.3\,\text{s} \pm 49.6\,\text{s}$
all the 4096 certificates	$34.9\,\text{s} \pm 9.9\,\text{s}$

Outline

Conclusion and Perspectives

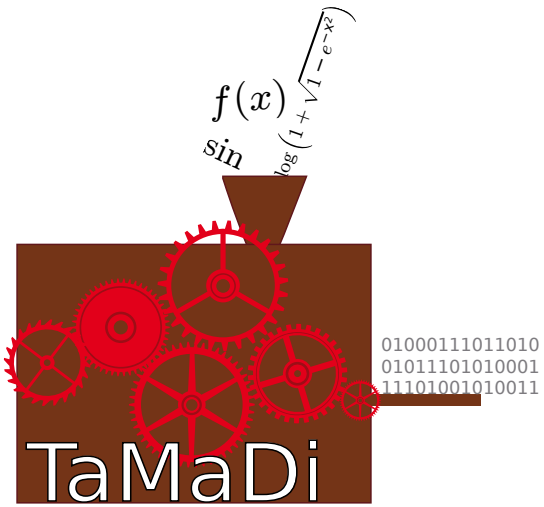
CoqHensel: a formalization of effective certificate checkers in Coq

- using Coppersmith's technique & Hensel lifting as certifying programs.
- Ultimate goal: Ensure that no hard-to-round case has been forgotten.

Perspectives

- Increase the performances of computation in Coq by implementing faster algorithms for the operations of polynomials.
- Combine CoqHensel & CoqApprox to provide a complete certificate checker for the SLZ algorithm.

End of the talk



Thank you for your attention!

The TaMaDi project homepage:
http://tamadi.gforge.inria.fr/

Focus on Coppersmith's technique

Outline of the reduction (from an ISValP instance)

- Assume $(x, y) \in [\![-A, A]\!] \times [\![-B, B]\!]$ satisfies $P(x) \equiv y \pmod{M}$, and pose $Q(X, Y) := P(X) - Y \in \mathbb{Z}[X, Y]$. This implies
  (1) $\quad Q(x, y) \equiv 0 \pmod{M}$
- Choose a parameter $\alpha > 0$ and consider the family of polynomials
  $$Q_{i,j}(X, Y) = Q(X, Y)^j M^{\alpha - j} X^i \quad (j \leqslant \alpha).$$
- Heuristically, find two $\mathbb{Z}$-linear combinations v_1, v_2 of $(Q_{i,j})$ such that
  (2) $\quad \forall (z, t) \in [\![-A, A]\!] \times [\![-B, B]\!], \quad |v_l(z, t)| < M^\alpha \quad (l = 1, 2).$
- By definition of $(Q_{i,j})$, v_1 and v_2, Equation (1) implies
  (3) $\quad v_1(x, y) \equiv 0 \equiv v_2(x, y) \pmod{M^\alpha}.$
- Combining (2) and (3) leads to
  (4) $\quad v_1(x, y) = 0 = v_2(x, y).$

Univariate Hensel's lemma

Lemma (Hensel)

Let $v \in \mathbb{Z}[X]$ and p be a prime satisfying

$$\forall x \in \mathbb{Z}, \quad v(x) \equiv 0 \pmod{p} \implies v'(x) \not\equiv 0 \pmod{p}. \tag{5}$$

For any $a, b \in \mathbb{Z}$, and $k \in \mathbb{N}$ s.t. $v(a) \equiv 0 \equiv v(b) \pmod{p^{2^k}}$, we have

$$a \equiv b \pmod{p} \implies a \equiv b \pmod{p^{2^k}}.$$

The proof notably relies on Taylor's theorem for polynomials. Return