An experience around Hensel's lemma

Érik Martin-Dorel

École Normale Supérieure de Lyon, LIP
46 allée d'Italie, 69364 LYON CEDEX 07, France

erik.martin-dorel@ens-lyon.org
http://perso.ens-lyon.fr/erik.martin-dorel/

Journées du projet ANR TaMaDi
Mercredi 27 octobre 2010
ENS de Lyon, LIP


Outline

Outline

The IEEE 754–2008 standard for Floating-Point Arithmetic

- A *floating-point (FP) format* is specified by the following parameters:
  - the radix $\beta \in \{2, 10\}$,
  - the precision $p \in \mathbb{N}^*$,
  - a maximum exponent $e_{\max}$ and a minimum one $e_{\min} = -e_{\max} + 1$.
- In such a format, we can represent $\pm\infty$ and the numbers of the form

$$x = (-1)^s \times m \times \beta^e \quad \text{with} \quad m = d_0.d_1 \dots d_{p-1},$$

for $s \in \{0, 1\}$, $0 \leqslant d_i < \beta \ \forall i$ $(\Rightarrow 0 \leqslant m < \beta)$ and $e_{\min} \leqslant e \leqslant e_{\max}$.

- For any such finite FP number x, we define its *unit in the last place* by

$$\mathrm{ulp}(x) = \beta^{e-p+1}.$$

The IEEE 754–2008 standard for Floating-Point Arithmetic

- A *floating-point (FP) format* is specified by the following parameters:
  - the radix $\beta \in \{2, 10\}$,
  - the precision $p \in \mathbb{N}^*$,
  - a maximum exponent $e_{\max}$ and a minimum one $e_{\min} = -e_{\max} + 1$.
- In such a format, we can represent $\pm\infty$ and the numbers of the form

$$x = (-1)^s \times m \times \beta^e \quad \text{with} \quad m = d_0.d_1 \dots d_{p-1},$$

  for $s \in \{0, 1\}$, $0 \leqslant d_i < \beta \ \forall i$ $(\Rightarrow 0 \leqslant m < \beta)$ and $e_{\min} \leqslant e \leqslant e_{\max}$.
- For any such finite FP number x, we define its *unit in the last place* by

$$\mathrm{ulp}(x) = \beta^{e-p+1}.$$

The IEEE 754–2008 standard for Floating-Point Arithmetic

- A *floating-point (FP) format* is specified by the following parameters:
  - the radix $\beta \in \{2, 10\}$,
  - the precision $p \in \mathbb{N}^*$,
  - a maximum exponent $e_{\max}$ and a minimum one $e_{\min} = -e_{\max} + 1$.
- In such a format, we can represent $\pm\infty$ and the numbers of the form

$$x = (-1)^s \times m \times \beta^e \quad \text{with} \quad m = d_0.d_1 \dots d_{p-1},$$

  for $s \in \{0, 1\}$, $0 \leqslant d_i < \beta \ \forall i$ $(\Rightarrow 0 \leqslant m < \beta)$ and $e_{\min} \leqslant e \leqslant e_{\max}$.
- For any such finite FP number x, we define its *unit in the last place* by

$$\mathrm{ulp}(x) = \beta^{e-p+1}.$$

Context: correct rounding of elementary functions

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be an elementary function ($\sin, \cos, \exp, \log, \ldots$)
- Let $\mathcal{M}_\beta^p \subset \mathbb{R}$ be the set of (machine) radix-β, precision-p FP numbers, along with a rounding function $\circ_p : \mathbb{R} \longrightarrow \mathcal{M}_\beta^p$ chosen among the five rounding modes specified in IEEE 754–2008 (RU, RD, RZ; RN, RN′).
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
$$x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$$
as if we computed the exact value of $f(x)$ (with an infinite precision) before rounding to the target precision with the chosen rounding mode. The function $\widetilde{f}$ so obtained is said to be correctly rounded.

Context: correct rounding of elementary functions

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be an elementary function (sin, cos, exp, log, ...)
- Let $\mathcal{M}_\beta^p \subset \mathbb{R}$ be the set of (machine) radix-β, precision-p FP numbers, along with a rounding function $\circ_p : \mathbb{R} \longrightarrow \mathcal{M}_\beta^p$ chosen among the five rounding modes specified in IEEE 754–2008 (RU, RD, RZ; RN, RN′).
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$

$$x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$$

as if we computed the exact value of $f(x)$ (with an infinite precision) before rounding to the target precision with the chosen rounding mode. The function $\widetilde{f}$ so obtained is said to be correctly rounded.

Context: correct rounding of elementary functions

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be an elementary function ($\sin, \cos, \exp, \log, \ldots$)
- Let $\mathcal{M}_\beta^p \subset \mathbb{R}$ be the set of (machine) radix-β, precision-p FP numbers, along with a rounding function $\circ_p : \mathbb{R} \longrightarrow \mathcal{M}_\beta^p$ chosen among the five rounding modes specified in IEEE 754–2008 (RU, RD, RZ; RN, RN′).
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$

$$x \longmapsto \widetilde{f}(x) := \circ_p (f(x))$$

as if we computed the exact value of $f(x)$ (with an infinite precision) before rounding to the target precision with the chosen rounding mode. The function $\widetilde{f}$ so obtained is said to be correctly rounded.

Context: correct rounding of elementary functions

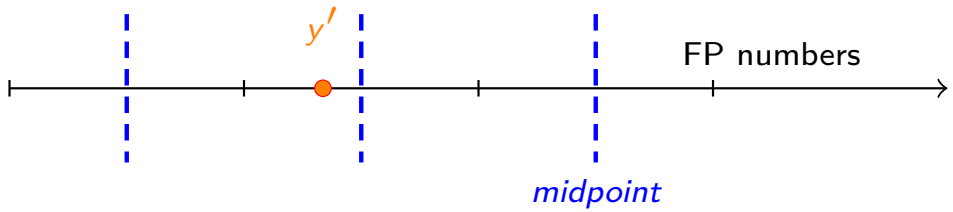
- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be an elementary function ($\sin, \cos, \exp, \log, \ldots$)
- Let $\mathcal{M}_\beta^p \subset \mathbb{R}$ be the set of (machine) radix-β, precision-p FP numbers, along with a rounding function $\circ_p : \mathbb{R} \longrightarrow \mathcal{M}_\beta^p$ chosen among the five rounding modes specified in IEEE 754–2008 (RU, RD, RZ; RN, RN′).
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$

$$x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$$

as if we computed the exact value of $f(x)$ (with an infinite precision) before rounding to the target precision with the chosen rounding mode. The function $\widetilde{f}$ so obtained is said to be correctly rounded.

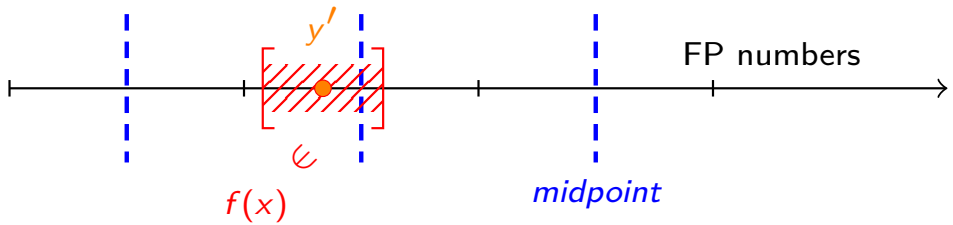
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- What we can do: for the desired value(s) of x:
- compute an approximation y' of $f(x)$, whose mantissa have an error of at most $\beta^{-p'}$ (in terms of relative dist., or in terms of ulps),
- then try to round the result.



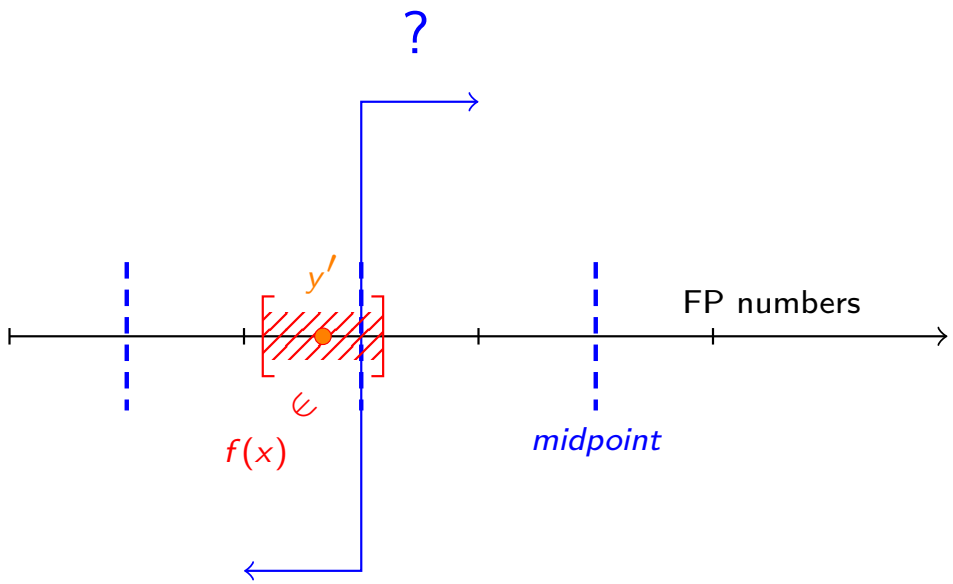
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- What we can do: for the desired value(s) of x:
- compute an approximation y' of $f(x)$, whose mantissa have an error of at most $\beta^{-p'}$ (in terms of relative dist., or in terms of ulps),
- then try to round the result.



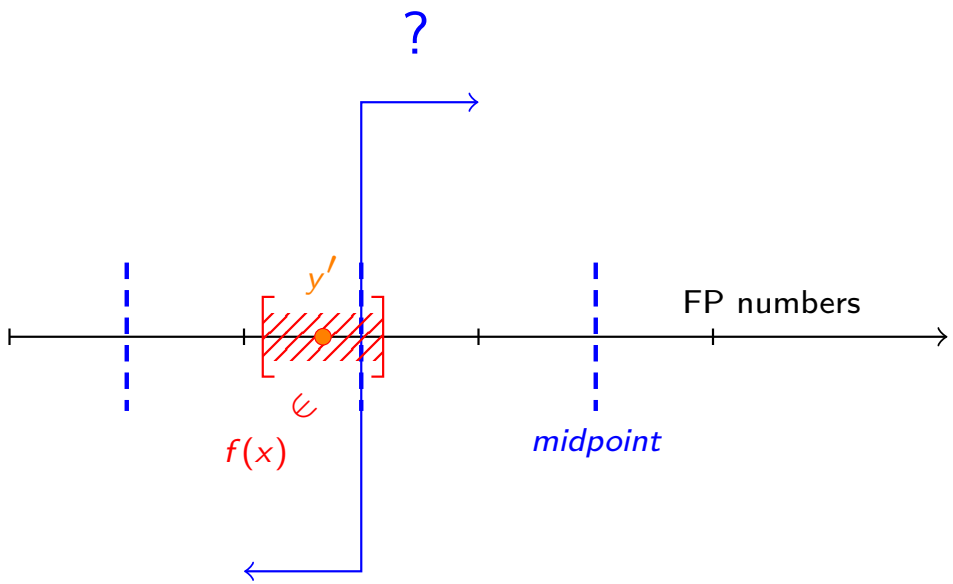
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- What we can do: for the desired value(s) of x:
- compute an approximation y' of $f(x)$, whose mantissa have an error of at most $\beta^{-p'}$ (in terms of relative dist., or in terms of ulps),
- then try to round the result.



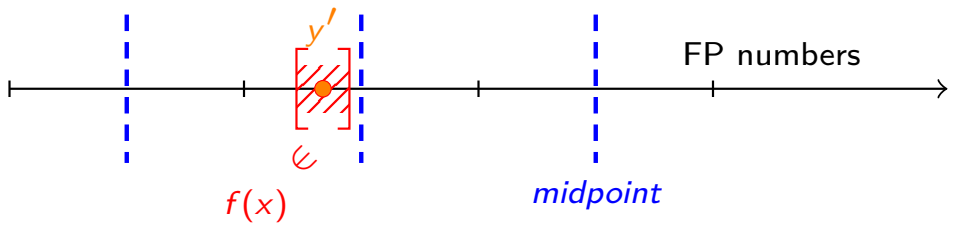
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- Definition: a *midpoint* is the middle of two consecutive FP numbers.
- Principle: if there is no midpoint in $[y' - \varepsilon y', y' + \varepsilon y']$ for $\varepsilon = \beta^{-p'}$, then we may ensure that $\circ_p(y') = \circ_p(f(x))$.
- Working precision p' needed to decide correct rounding for all x?



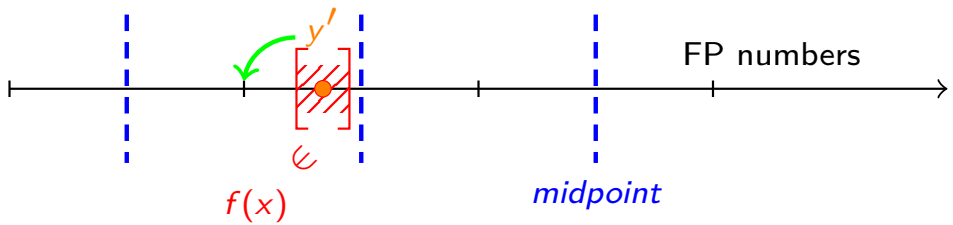
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- Definition: a *midpoint* is the middle of two consecutive FP numbers.
- Principle: if there is no midpoint in $[y' - \varepsilon y', y' + \varepsilon y']$ for $\varepsilon = \beta^{-p'}$, then we may ensure that $\circ_p(y') = \circ_p(f(x))$.
- Working precision p' needed to decide correct rounding for all x?



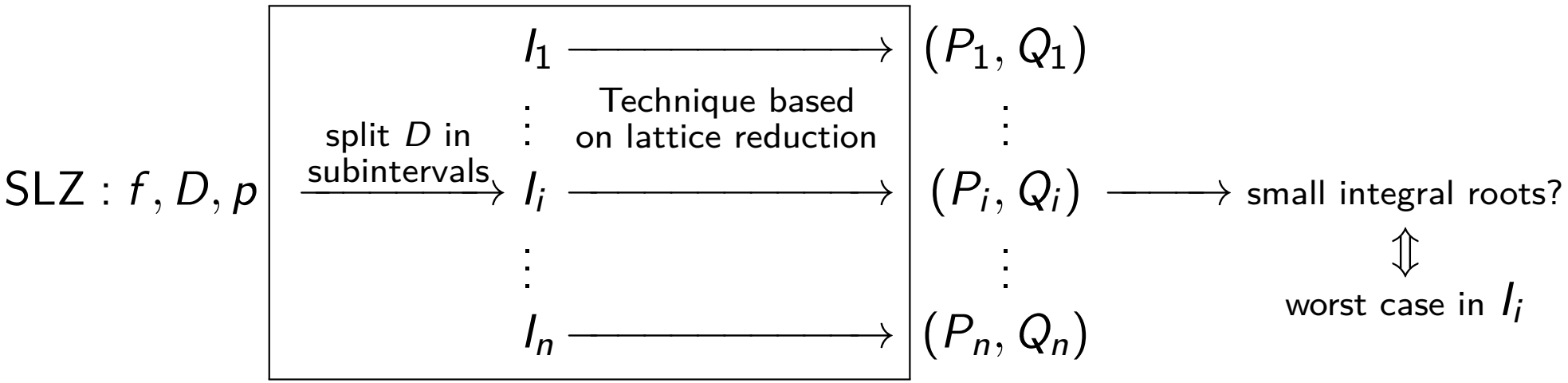
Goal: solving the Table Maker's Dilemma

- Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$.
- We want to provide $\widetilde{f} : D \cap \mathcal{M}_\beta^p \longrightarrow \mathcal{M}_\beta^p$
  $x \longmapsto \widetilde{f}(x) := \circ_p(f(x))$.
- Definition: a *midpoint* is the middle of two consecutive FP numbers.
- Principle: if there is no midpoint in $[y' - \varepsilon y', y' + \varepsilon y']$ for $\varepsilon = \beta^{-p'}$, then we may ensure that $\circ_p(y') = \circ_p(f(x))$.
- Working precision p' needed to decide correct rounding for all x?



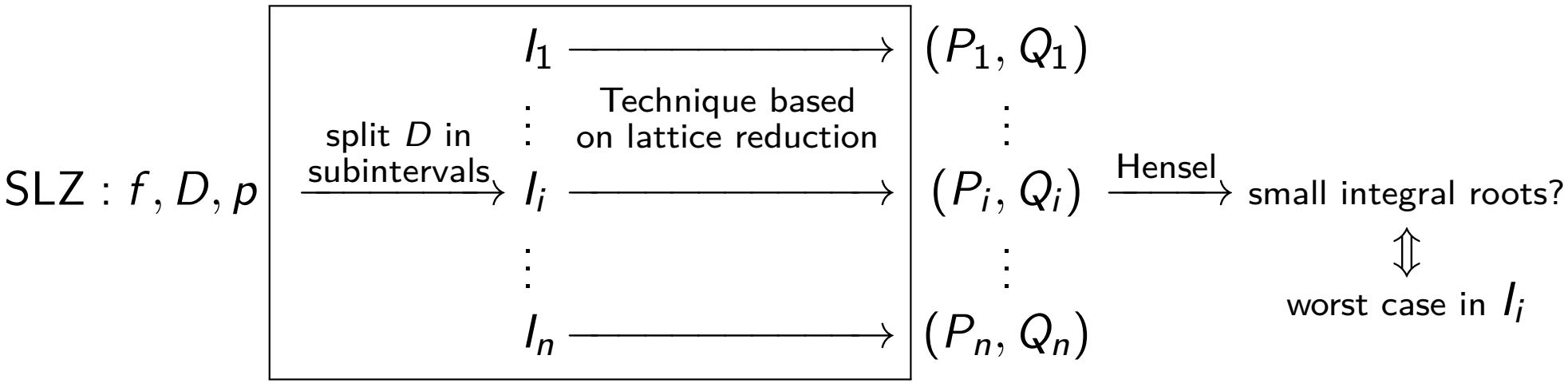
Overview of the Stehlé–Lefèvre–Zimmermann algorithm

- Goal: for $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$, know when $f(x)$ is equal, or very close to a midpoint. Thanks to these "worst cases," we can determine the working precision p' needed to decide correct rounding for all $x \in D$.



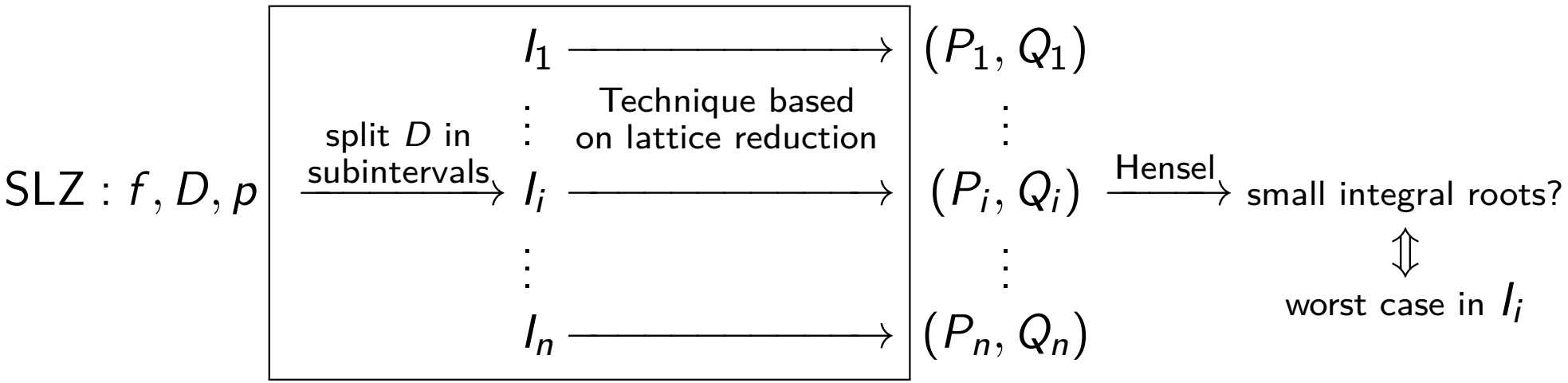
Overview of the Stehlé–Lefèvre–Zimmermann algorithm

- Goal: for $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$, know when $f(x)$ is equal, or very close to a midpoint. Thanks to these "worst cases," we can determine the working precision p' needed to decide correct rounding for all $x \in D$.



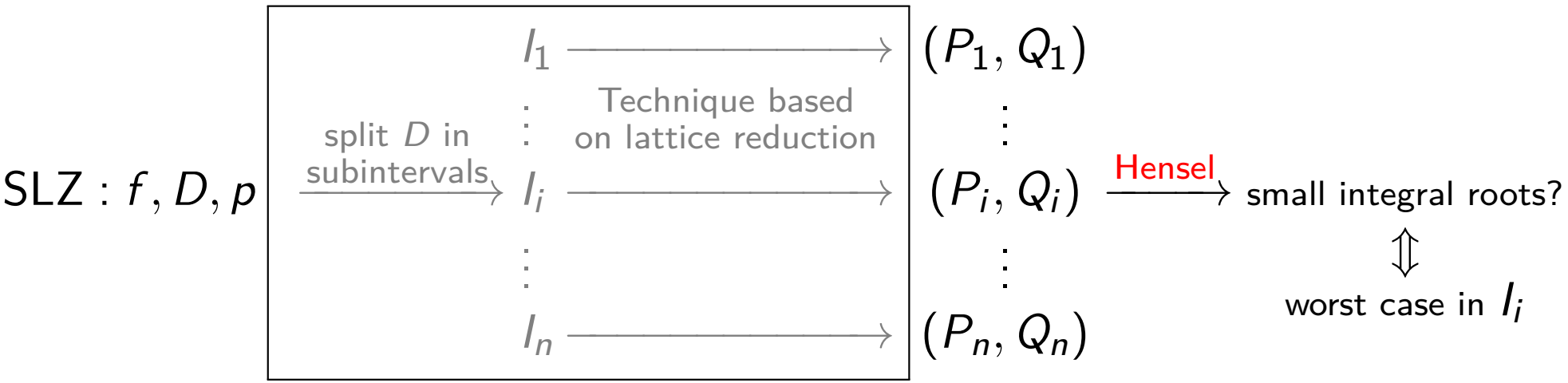
where for each i, P_i and Q_i are in $\mathbb{Z}[X, Y]$.

Overview of the Stehlé–Lefèvre–Zimmermann algorithm



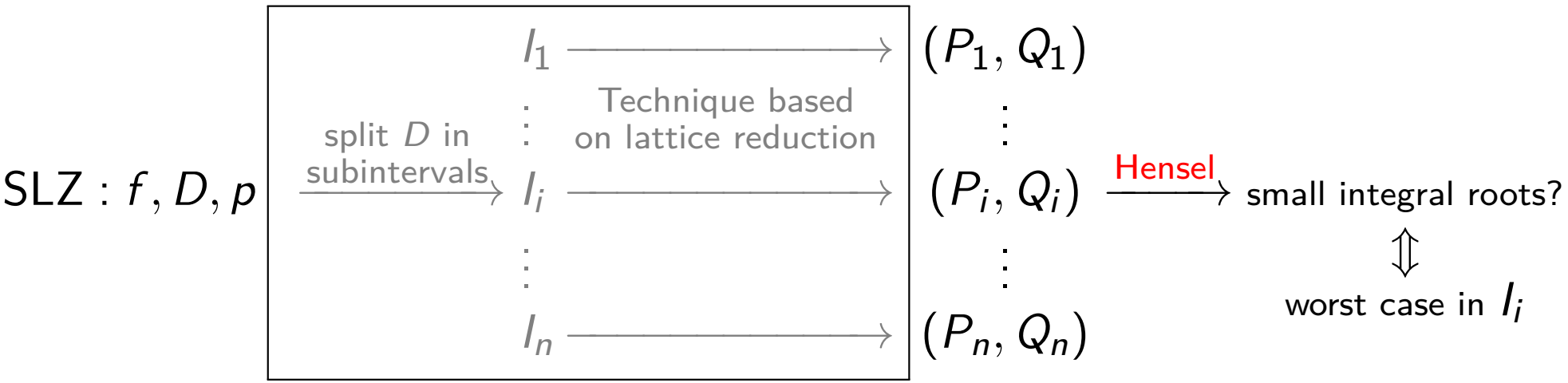
- The SLZ algorithm succeeds in enumerating the hardest-to-round cases of an elementary function like exp after months of CPU-time. We need to ensure high confidence in theses worst-cases tables.
- Thus, a formal specification and proof of Hensel's lifting method that is part of SLZ contributes to the validation of the overall algorithm.
- As with any algorithm, there is a trade-off between "performance" and "provability". But the certificate-based approach can satisfy both criteria...

Overview of the Stehlé–Lefèvre–Zimmermann algorithm



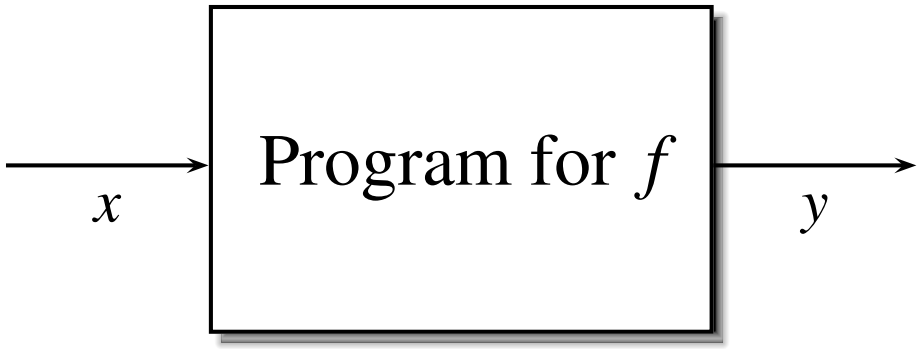
- The SLZ algorithm succeeds in enumerating the hardest-to-round cases of an elementary function like exp after months of CPU-time. We need to ensure high confidence in theses worst-cases tables.
- Thus, a formal specification and proof of Hensel's lifting method that is part of SLZ contributes to the validation of the overall algorithm.
- As with any algorithm, there is a trade-off between "performance" and "provability". But the certificate-based approach can satisfy both criteria...

Overview of the Stehlé–Lefèvre–Zimmermann algorithm



- The SLZ algorithm succeeds in enumerating the hardest-to-round cases of an elementary function like exp after months of CPU-time. We need to ensure high confidence in theses worst-cases tables.
- Thus, a formal specification and proof of Hensel's lifting method that is part of SLZ contributes to the validation of the overall algorithm.
- As with any algorithm, there is a trade-off between "performance" and "provability". But the certificate-based approach can satisfy both criteria...

Overview of the certificate-based approach



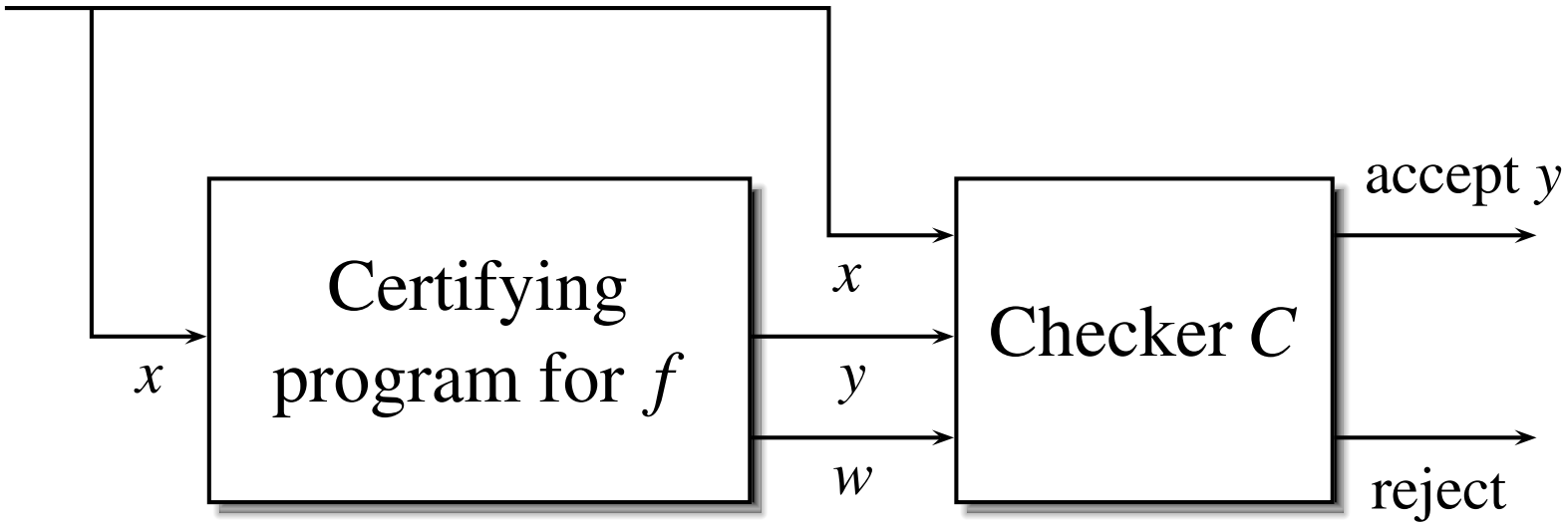


Figure taken from *Certifying Algorithms* [Mehlhorn et al., 2010].

Outline

Statement of univariate Hensel's lemma

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies:

Statement of univariate Hensel's lemma

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.
[Combined with $P(x) = 0$, this implies $P(u_k) \equiv 0 \pmod{p^{2^k}}$ for all $k \in \mathbb{N}$.]
Furthermore, as soon as p^{2^k} exceeds a known bound on the roots of P, we can deduce the value of the sought root: $x = u_k$ or $x = u_k - p^{2^k}$.

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

For $u_0 = 0$, we compute:

$u_1 = u_0 - P(u_0)/P'(u_0) \bmod 2^{2^1} = 0 - 42/(-13) \bmod 4 = 2$;

$u_2 = u_1 - P(u_1)/P'(u_1) \bmod 2^{2^2} = 2 - 20/(-9) \bmod 16 = 6$.

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

$u_0 = 0 \implies u_1 = 2 \implies u_2 = 6$.

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

$u_0 = 0 \implies u_1 = 2 \implies u_2 = 6$.

For $u_0 = 1$, we compute:

$u_1 = u_0 - P(u_0)/P'(u_0) \bmod 2^{2^1} = 1 - 30/(-11) \bmod 4 = 3$;

$u_2 = u_1 - P(u_1)/P'(u_1) \bmod 2^{2^2} = 3 - 12/(-7) \bmod 16 = 7$.

Statement of univariate Hensel's lemma & example of use

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

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

If $x \in \mathbb{Z}$ is an integral root of P, then for $u_0 := x \bmod p$, the sequence (u_k) defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad u_{k+1} := u_k - \frac{P(u_k)}{P'(u_k)} \bmod p^{2^{k+1}}$$

satisfies: $\forall k \in \mathbb{N}, \quad u_k = x \bmod p^{2^k}$.

E.g., let us consider $P := X^2 - 13 \cdot X + 42 \in \mathbb{Z}[X]$ (its roots are 6 and 7).

Its formal derivative is $P'(X) = 2 \cdot X - 13$.

For $p = 2$, the main hypothesis is fulfilled for both roots mod p (0 and 1).

$u_0 = 0 \implies u_1 = 2 \implies u_2 = 6.$

$u_0 = 1 \implies u_1 = 3 \implies u_2 = 7.$

We did find each root in $\mathbb{Z}$ after 2 iterations only. *In practice, we can stop as soon as the modulus exceeds a known bound on the roots.*

Summary of the required concepts

- types: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Z}/q\mathbb{Z}$, $\mathbb{Z}[X]$;
- predicates/functions: `prime`, powers, `Zmod`, polynomial evaluation;
- and some related support theorems.

Summary of the required concepts

- types: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Z}/q\mathbb{Z}$, $\mathbb{Z}[X]$;
- predicates/functions: `prime`, powers, `Zmod`, polynomial evaluation;
- and some related support theorems,

  including Taylor's theorem for polynomials:

```
Lemma nderiv_taylor :
  forall (R : ringType) (p : {poly R}) (x h : R),
  GRing.comm x h ->
  p.[x + h] = \sum_(i < size p) (p^`N(i)).[x] * h ^+ i.
```

$$\left(\text{where p\^{}`N(i) stands for } \frac{p^{(i)}}{i!}\right)$$

We need to handle Polynomials on Z

```
Require Import ZArith.
Require Import ssreflect ssrfun ssrbool eqtype ssrnat.
Require Import poly.


Check {poly Z}.
```

We need to handle Polynomials on Z

```
Require Import ZArith.
Require Import ssreflect ssrfun ssrbool eqtype ssrnat.
Require Import poly.


Check {poly Z}.

(*
 * Toplevel input, characters 25-33:
 * > Check {poly Z}.
 * >         ^^^^^^^^
 * Error: The term "Phant Z" has type "phant Z"
 *  while it is expected to have type "phant ?4".
 *)
```

We need to handle Polynomials on Z

```
Require Import ZArith.
Require Import ssreflect ssrfun ssrbool eqtype ssrnat.
Require Import poly.

Set Printing All.
Check {poly Z}.

(*
 * Toplevel input, characters 25-33:
 * > Check {poly Z}.
 * >       ^^^^^^^^
 * Error: The term "Phant Z" has type "phant Z"
 *  while it is expected to have type "phant
 *    (ssralg.GRing.Ring.sort ?5)".
 *)
```

Overview of the Coq theories at stake

- `ssrz.v`:
  - based on the binary integers from `ZArith`,
  - it provides the `Z_eqType`, `Z_choiceType`, `Z_countType`, `Z_zmodType`, and `Z_ringType` Canonical Structures (cf. the SSReflect hierarchy),
  - and gathers some other results useful for the formalization, including:

```
Lemma Z_of_nat_moduli : forall n m : nat, (0 < m)%N ->
  Z_of_nat (n %% m) = ((Z_of_nat n) mod (Z_of_nat m))%Z.
Definition ZtoI :  forall q : nat, 1 < q -> Z -> 'Z_q.
Lemma ZtoI_morph : forall q (q_gt1 : 1 < q),
  GRing.morphism (ZtoI q_gt1).
Lemma ZtoI_Zmod :  forall q (q_gt1 : 1 < q) z,
  Z_of_nat (ZtoI q_gt1 z) = (z mod (Z_of_nat q))%Z.
Lemma commut_ZmodN_horner :
  forall (q : nat) (z : Z) (P : {poly Z}),
  P.[z modN q] = P.[z] %[ZmodN q].
```

- `hensel.v`:
  - on top of `ssrz.v` and `poly.v`,
  - it provides a proof of univariate Hensel's lemma, whose type is...

Overview of the Coq theories at stake

- `ssrz.v`:
  - based on the binary integers from `ZArith`,
  - it provides the `Z_eqType`, `Z_choiceType`, `Z_countType`, `Z_zmodType`, and `Z_ringType` Canonical Structures (cf. the SSReflect hierarchy),
  - and gathers some other results useful for the formalization, including:

```
Lemma Z_of_nat_moduli : forall n m : nat, (0 < m)%N ->
  Z_of_nat (n %% m) = ((Z_of_nat n) mod (Z_of_nat m))%Z.
Definition ZtoI :  forall q : nat, 1 < q -> Z -> 'Z_q.
Lemma ZtoI_morph : forall q (q_gt1 : 1 < q),
  GRing.morphism (ZtoI q_gt1).
Lemma ZtoI_Zmod :  forall q (q_gt1 : 1 < q) z,
  Z_of_nat (ZtoI q_gt1 z) = (z mod (Z_of_nat q))%Z.
Lemma commut_ZmodN_horner :
  forall (q : nat) (z : Z) (P : {poly Z}),
  P.[z modN q] = P.[z] %[ZmodN q].
```

- `hensel.v`:
  - on top of `ssrz.v` and `poly.v`,
  - it provides a proof of univariate Hensel's lemma, whose type is...

Checking the type of (univariate) Hensel's lemma

```
Lemma uni_hensel_lemma :
  forall (P : {poly Z}) (p : nat) (p_prime : prime p),
  (forall z : Z,
    P.[z] = 0 %[ZmodN p] -> (P^`()).[z] <> 0 %[ZmodN p]) ->
  forall x : Z,
  P.[x] = 0 ->
  forall k : nat,
  uni_hensel_iter P p_prime (x modN p) k = x modN p ^ 2 ^ k.
```

Structure of the certificate

```
Record certif := Certif {
  c_pol : {poly Z};
  c_p : nat;
  c_bnd : Z;
  c_k : nat;
  c_l : nat;
  c_lst : seq (Z * Z * bool)
}.
```

Bivariate version of Hensel's lemma

Let $P_1, P_2 \in \mathbb{Z}[X, Y]$ and let p be a prime satisfying

$$\forall x, y \in \mathbb{Z}, P_1(x, y) \equiv 0 \equiv P_2(x, y) \pmod{p} \Rightarrow \det J_{P_1,P_2}(x, y) \not\equiv 0 \pmod{p}.$$

If $(a, b) \in \mathbb{Z}^2$ is a root for both P_1 and P_2, then for

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} := \begin{pmatrix} a \bmod p \\ b \bmod p \end{pmatrix},$$

the sequence $(u_k, v_k)_{k\in\mathbb{N}}$ defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} := \begin{pmatrix} u_k \\ v_k \end{pmatrix} - \left[J_{P_1,P_2}(u_k, v_k) \right]^{-1} \begin{pmatrix} P_1(u_k, v_k) \\ P_2(u_k, v_k) \end{pmatrix} \bmod p^{2^{k+1}}$$

satisfies:

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} a \bmod p^{2^k} \\ b \bmod p^{2^k} \end{pmatrix}.$$

[This implies $P_1(u_k, v_k) \equiv 0 \equiv P_2(u_k, v_k) \pmod{p^{2^k}}$ for all $k \in \mathbb{N}$.]

Bivariate version of Hensel's lemma

Let $P_1, P_2 \in \mathbb{Z}[X, Y]$ and let p be a prime satisfying

$$\forall x, y \in \mathbb{Z}, P_1(x, y) \equiv 0 \equiv P_2(x, y) \pmod{p} \Rightarrow \det J_{P_1,P_2}(x, y) \not\equiv 0 \pmod{p}.$$

If $(a, b) \in \mathbb{Z}^2$ is a root for both P_1 and P_2, then for

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} := \begin{pmatrix} a \bmod p \\ b \bmod p \end{pmatrix},$$

the sequence $(u_k, v_k)_{k\in\mathbb{N}}$ defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} := \begin{pmatrix} u_k \\ v_k \end{pmatrix} - \left[J_{P_1,P_2}(u_k, v_k)\right]^{-1} \begin{pmatrix} P_1(u_k, v_k) \\ P_2(u_k, v_k) \end{pmatrix} \bmod p^{2^{k+1}}$$

satisfies:

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} a \bmod p^{2^k} \\ b \bmod p^{2^k} \end{pmatrix}.$$

[This implies $P_1(u_k, v_k) \equiv 0 \equiv P_2(u_k, v_k) \pmod{p^{2^k}}$ for all $k \in \mathbb{N}$.]

Bivariate version of Hensel's lemma

Let $P_1, P_2 \in \mathbb{Z}[X, Y]$ and let p be a prime satisfying

$$\forall x, y \in \mathbb{Z}, P_1(x, y) \equiv 0 \equiv P_2(x, y) \pmod{p} \Rightarrow \det J_{P_1,P_2}(x, y) \not\equiv 0 \pmod{p}.$$

If $(a, b) \in \mathbb{Z}^2$ is a root for both P_1 and P_2, then for

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} := \begin{pmatrix} a \bmod p \\ b \bmod p \end{pmatrix},$$

the sequence $(u_k, v_k)_{k\in\mathbb{N}}$ defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} := \begin{pmatrix} u_k \\ v_k \end{pmatrix} - \left[J_{P_1,P_2}(u_k, v_k)\right]^{-1} \begin{pmatrix} P_1(u_k, v_k) \\ P_2(u_k, v_k) \end{pmatrix} \bmod p^{2^{k+1}}$$

satisfies:

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} a \bmod p^{2^k} \\ b \bmod p^{2^k} \end{pmatrix}.$$

[This implies $P_1(u_k, v_k) \equiv 0 \equiv P_2(u_k, v_k) \pmod{p^{2^k}}$ for all $k \in \mathbb{N}$.]

Bivariate version of Hensel's lemma

Let $P_1, P_2 \in \mathbb{Z}[X, Y]$ and let p be a prime satisfying

$$\forall x, y \in \mathbb{Z}, P_1(x, y) \equiv 0 \equiv P_2(x, y) \pmod{p} \Rightarrow \det J_{P_1,P_2}(x, y) \not\equiv 0 \pmod{p}.$$

If $(a, b) \in \mathbb{Z}^2$ is a root for both P_1 and P_2, then for

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} := \begin{pmatrix} a \bmod p \\ b \bmod p \end{pmatrix},$$

the sequence $(u_k, v_k)_{k\in\mathbb{N}}$ defined by the recurrence relation

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} := \begin{pmatrix} u_k \\ v_k \end{pmatrix} - \left[J_{P_1,P_2}(u_k, v_k) \right]^{-1} \begin{pmatrix} P_1(u_k, v_k) \\ P_2(u_k, v_k) \end{pmatrix} \bmod p^{2^{k+1}}$$

satisfies:

$$\forall k \in \mathbb{N}, \quad \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} a \bmod p^{2^k} \\ b \bmod p^{2^k} \end{pmatrix}.$$

[This implies $P_1(u_k, v_k) \equiv 0 \equiv P_2(u_k, v_k) \pmod{p^{2^k}}$ for all $k \in \mathbb{N}$.]

Coq objects available for the proof

```
Record mat2by2 (R : ringType) : Type := Mat2by2
  { m11 : GRing.Ring.sort R; m12 : GRing.Ring.sort R;
    m21 : GRing.Ring.sort R; m22 : GRing.Ring.sort R }.
MatVec : forall R : ringType, mat2by2 R -> R * R -> R * R.
Jmatev : forall R : ringType, {bipoly R} -> {bipoly R} ->
  R -> R -> mat2by2 R.
Jdet  :  forall R : ringType, {bipoly R} -> {bipoly R} ->
  {bipoly R}.
Cramer_x : mat2by2 Z_ringType -> Z_ringType -> Z_ringType ->
  forall q : nat, (1 < q)%N -> Z_ringType.

Lemma bieval_mod : forall (P : {bipoly Z}) (u v : Z) q,
  (1 < q)%N -> P.2[u modN q, v modN q] = P.2[u,v] %[ZmodN q].
```

& `nderiv2_taylor`, which has been proved by Laurent Théry in `bipoly.v`

Checking the type of bivariate Hensel's lemma (with holes)

```
Lemma biv_hensel_lemma
  forall (P1 P2 : {bipoly Z}) (p : nat) (p_prime : prime p),
  (forall x y : Z,
    0 <= x < Z_of_nat p ->
    0 <= y < Z_of_nat p ->
    P1.2[x, y] = 0 %[ZmodN p] ->
    P2.2[x, y] = 0 %[ZmodN p] ->
    (Jdet P1 P2).2[x, y] <> 0 %[ZmodN p]) ->
  forall a b : Z,
  P1.2[a, b] = 0 ->
  P2.2[a, b] = 0 ->
  forall k : nat,
  biv_hensel_iter P1 P2 p_prime (a modN p, b modN p) k =
  ((fun k0 : nat => a modN p ^ 2 ^ k0) k,
   (fun k0 : nat => b modN p ^ 2 ^ k0) k).
```

Outline

Conclusion

- Hensel's lemma is a usual result in valuation theory that
  - can be fully expressed in modular arithmetic on $\mathbb{Z}$
  - provides a correctness proof for Hensel's lifting method
- This method provides an efficient root-finding algorithm which
  - can be viewed as the p-adic variant of the Newton–Raphson's iteration
  - has a significant application in Computer Arithmetic through SLZ
- Nonetheless
  - Hensel's lifting has many other applications
  - SLZ can use other methods, such as resultant (former implementation)
- Key point of the proof: Taylor's theorem for polynomials
- The obtained proofs will be ultimately included in a full certification chain to solve the Table Maker's Dilemma in an exact way.

Conclusion

- Hensel's lemma is a usual result in valuation theory that
  - can be fully expressed in modular arithmetic on $\mathbb{Z}$
  - provides a correctness proof for Hensel's lifting method
- This method provides an efficient root-finding algorithm which
  - can be viewed as the p-adic variant of the Newton–Raphson's iteration
  - has a significant application in Computer Arithmetic through SLZ
- Nonetheless
  - Hensel's lifting has many other applications
  - SLZ can use other methods, such as resultant (former implementation)
- Key point of the proof: Taylor's theorem for polynomials
- The obtained proofs will be ultimately included in a full certification chain to solve the Table Maker's Dilemma in an exact way.

Conclusion

- Hensel's lemma is a usual result in valuation theory that
  - can be fully expressed in modular arithmetic on $\mathbb{Z}$
  - provides a correctness proof for Hensel's lifting method
- This method provides an efficient root-finding algorithm which
  - can be viewed as the p-adic variant of the Newton–Raphson's iteration
  - has a significant application in Computer Arithmetic through SLZ
- Nonetheless
  - Hensel's lifting has many other applications
  - SLZ can use other methods, such as resultant (former implementation)
- Key point of the proof: Taylor's theorem for polynomials
- The obtained proofs will be ultimately included in a full certification chain to solve the Table Maker's Dilemma in an exact way.

Conclusion

- Hensel's lemma is a usual result in valuation theory that
  - can be fully expressed in modular arithmetic on $\mathbb{Z}$
  - provides a correctness proof for Hensel's lifting method
- This method provides an efficient root-finding algorithm which
  - can be viewed as the p-adic variant of the Newton–Raphson's iteration
  - has a significant application in Computer Arithmetic through SLZ
- Nonetheless
  - Hensel's lifting has many other applications
  - SLZ can use other methods, such as resultant (former implementation)
- Key point of the proof: Taylor's theorem for polynomials
- The obtained proofs will be ultimately included in a full certification chain to solve the Table Maker's Dilemma in an exact way.

Future Work

1. Prove the few remaining subgoals for the bivariate proof
2. Extract a verifier for Hensel-based bivariate integral-roots certificates
3. Goal: provide a full certification of many correct-rounding worst-cases

Future Work

1. Prove the few remaining subgoals for the bivariate proof
2. Extract a verifier for Hensel-based bivariate integral-roots certificates
3. Goal: provide a full certification of many correct-rounding worst-cases

End of the talk

- Many thanks for your attention
- Any question?

Checking the type of nderiv2_taylor

```
Lemma nderiv2_taylor :
  forall (R : ringType) (p : {bipoly R}) (x h y k : R),
  GRing.comm x h ->
  GRing.comm x k ->
  GRing.comm x y ->
  GRing.comm y h ->
  GRing.comm y k ->
  GRing.comm h k ->
    p.2[x + h, y + k] =
      \sum_(i < size p)
        \sum_(j < size (p ^x`N(i)).[x%:P])
          ((p ^x`N(i)) ^y`N(j)).2[x, y] * (h ^+ i * k ^+ j).
```