A zero-test for σ-algebraic power series

Joris van der Hoeven and Gleb Pogudin,
MAX team, LIX, CNRS, École Polytechnique, Institut Polytechnique de Paris

IHP, Groupe de travail "Transcendance et Combinatoire"

Plan

- Introduction: what do we want?
- Our results: what can we do?
- Nuts and bolts: how do we do this?

Introduction: what do we want?

Big picture

We often deal with objects defined **implicitly** by equations, e.g:

- Algebraic equations $\rightarrow$ numbers ($\sqrt{2} :=$ positive root of $x^2 - 2 = 0$);
- Differential equations $\rightarrow$ functions ($f = e^x$ as $f' = f$ with $f(0) = 1$)

Big picture

We often deal with objects defined **implicitly** by equations, e.g:

- Algebraic equations $\to$ numbers ($\sqrt{2} :=$ positive root of $x^2 - 2 = 0$);
- Differential equations $\to$ functions ($f = e^x$ as $f' = f$ with $f(0) = 1$)

Fundamental question

Equality testing: are two such objects equal? (e.g. $\sqrt{2} - 1 = \frac{1}{1+\sqrt{2}}$)

Big picture

We often deal with objects defined **implicitly** by equations, e.g:

- Algebraic equations $\to$ numbers ($\sqrt{2} :=$ positive root of $x^2 - 2 = 0$);
- Differential equations $\to$ functions ($f = e^x$ as $f' = f$ with $f(0) = 1$)

Fundamental question

Equality testing: are two such objects equal? (e.g. $\sqrt{2} - 1 = \frac{1}{1+\sqrt{2}}$)

By $A = B \iff A - B = 0$ is often reduced to **zero-testing**.

Context

In symbolic computation:

- Polynomial equations $\rightarrow$ algebraic numbers
- Linear differential equations $\rightarrow$ D-finite power series
- Algebraic differential equations $\rightarrow$ D-algebraic power series

Context

In symbolic computation:

- Polynomial equations $\rightarrow$ algebraic numbers

- Linear differential equations $\rightarrow$ D-finite power series

- Algebraic differential equations $\rightarrow$ D-algebraic power series

- Algebraic difference equations
  (e.g., $f_{n+1} = f_n + f_{n-1}$ or $\Gamma(z+1) = z\Gamma(z)$):

Context

In symbolic computation:

- Polynomial equations $\rightarrow$ algebraic numbers

- Linear differential equations $\rightarrow$ D-finite power series

- Algebraic differential equations $\rightarrow$ D-algebraic power series

- Algebraic difference equations
(e.g., $f_{n+1} = f_n + f_{n-1}$ or $\Gamma(z+1) = z\Gamma(z)$):
    1. σ-algebraic sequences

    2. σ-algebraic power series

Context

In symbolic computation:

- Polynomial equations $\to$ algebraic numbers
  Zero-test: Liouville's theorem
- Linear differential equations $\to$ D-finite power series
  Zero-test: folklore?
- Algebraic differential equations $\to$ D-algebraic power series
  Zero-test: Denef & Lipshitz (1984), Shackell (1993), van der Hoeven (2002, 2019) $\to$ **Two weeks ago**
- Algebraic difference equations
  (e.g., $f_{n+1} = f_n + f_{n-1}$ or $\Gamma(z+1) = z\Gamma(z)$):
  1. σ-algebraic sequences
     Zero-test: Kauers (2007) for a large class
  2. σ-algebraic power series
     Zero-test: **This talk!**

Background: computable power series

Let K be a computable ground field (e.g., $\mathbb{Q}$).

Background: computable power series

Let K be a computable ground field (e.g., $\mathbb{Q}$).

Computable power series

A power series $f = \sum_{i=0}^{\infty} f_i t^i \in K[\![z]\!]$ is called computable if there exists an algorithm computing f_n given n as input.

Background: computable power series

Let K be a computable ground field (e.g., $\mathbb{Q}$).

Computable power series

A power series $f = \sum_{i=0}^{\infty} f_i t^i \in K[\![z]\!]$ is called computable if there exists an algorithm computing f_n given n as input.

Remark

- sum, product, quotient, and composition (if well-defined) of computable power series are computable;

Background: computable power series

Let K be a computable ground field (e.g., $\mathbb{Q}$).

Computable power series

A power series $f = \sum_{i=0}^{\infty} f_i t^i \in K[\![z]\!]$ is called computable if there exists an algorithm computing f_n given n as input.

Remark

- sum, product, quotient, and composition (if well-defined) of computable power series are computable;
- BUT given computable f and g, it is not decidable whether $f = g$ (or, equivalently, whether $f - g = 0$)

Background: computable power series

Let K be a computable ground field (e.g., $\mathbb{Q}$).

Computable power series

A power series $f = \sum_{i=0}^{\infty} f_i t^i \in K[\![z]\!]$ is called computable if there exists an algorithm computing f_n given n as input.

Remark

- sum, product, quotient, and composition (if well-defined) of computable power series are computable;
- BUT given computable f and g, it is not decidable whether $f = g$ (or, equivalently, whether $f - g = 0$)

Wanted: zero-test

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma\colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma \colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\} \colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma\colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\}\colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

Examples

- $g = z + z^2,\ f = z \implies \sigma(f) - f - f^2 = 0$

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma\colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\}\colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

Examples

- $g = z + z^2,\ f = z \implies \sigma(f) - f - f^2 = 0$
- Γ-function satisfies $\Gamma(n+1) = n\Gamma(n)$

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma \colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\} \colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

Examples

- $g = z + z^2,\ f = z \implies \sigma(f) - f - f^2 = 0$
- Γ-function satisfies $\Gamma(n+1) = n\Gamma(n)$,
  BUT after $z := \frac{1}{n}$ the shift $n \to n+1$ becomes $z \to \frac{z}{1+z} = z - \ldots$

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma\colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\}\colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

Computability

σ-algebraic $f \in K[\![z]\!]$ is computable if represented by

- $P(X_0, \ldots, X_r)$ with $\frac{\partial P}{\partial X_r}(f, \sigma(f), \ldots, \sigma^r(f)) \neq 0$

;

- sufficiently many initial terms.

σ-algebraic power series

Fix $g = z + \mathcal{O}(z^2) \in K[\![z]\!]$ and consider difference operator

$$\sigma\colon f(z) \to f(g(z)) \text{ for every } f(z) \in K[\![z]\!]$$

Definition

f is σ-algebraic of order r

$$\iff \exists P \in K[X_0, \ldots, X_r] \setminus \{0\}\colon P(f, \sigma(f), \ldots, \sigma^r(f)) = 0$$

Computability

σ-algebraic $f \in K[\![z]\!]$ is computable if represented by

- $P(X_0, \ldots, X_r)$ with $\frac{\partial P}{\partial X_r}(f, \sigma(f), \ldots, \sigma^r(f)) \neq 0$
  Differential case: the latter not required but achieved;
- sufficiently many initial terms.

Our results: what can we do?

Our algorithm

We give the first algorithm such that

Input:
- a σ-algebraic power series f defined as above (annihilator P + terms)
- polynomial Q in $f, \sigma(f), \sigma^2(f), \ldots$

Output: `True` if $Q = 0$ and `False` otherwise

Our algorithm

We give the first algorithm such that

Input:
- a σ-algebraic power series f defined as above (annihilator P + terms)
- polynomial Q in $f, \sigma(f), \sigma^2(f), \ldots$

Output: `True` if $Q = 0$ and `False` otherwise

Important remark

The annihilator P may be not over K but over $A \subset K[\![z]\!]$ such that

- A is a subalgebra closed under σ
- A has a zero test

Our algorithm

We give the first algorithm such that

Input:
- a σ-algebraic power series f defined as above (annihilator P + terms)
- polynomial Q in $f, \sigma(f), \sigma^2(f), \dots$

Output: `True` if $Q = 0$ and `False` otherwise

Important remark

The annihilator P may be not over K but over $A \subset K[\![z]\!]$ such that

- A is a subalgebra closed under σ
- A has a zero test

$\Longrightarrow$ One can build towers of extensions with zero-test (later in example)

Our algorithm

We give the first algorithm such that

Input:
- a σ-algebraic power series f defined as above (annihilator P + terms)
- polynomial Q in $f, \sigma(f), \sigma^2(f), \ldots$

Output: `True` if $Q = 0$ and `False` otherwise

Important remark

The annihilator P may be not over K but over $A \subset K[\![z]\!]$ such that

- A is a subalgebra closed under σ
- A has a zero test

$\Longrightarrow$ One can build towers of extensions with zero-test (later in example)

We provide a proof-of-concept Julia implementation
`https://github.com/pogudingleb/DifferenceZeroTest`
(gives a good idea how one should not implement this)

Example (Legendre's duplication formula)

$$\Gamma(n)\Gamma\left(n+\frac{1}{2}\right) = 2^{1-2n}\sqrt{\pi}\,\Gamma(2n)$$

Example (Legendre's duplication formula)

$$\Gamma(n)\Gamma\left(n+\frac{1}{2}\right) = 2^{1-2n}\sqrt{\pi}\Gamma(2n)$$

How we represent Γ-function?

$$\text{Stirling's series:}\quad \log\Gamma(n+1) = n\log n - n + \frac{1}{2}\log(2\pi n) + \boxed{\sum_{k=1}^{\infty}\frac{s_k}{n^k}}$$

Example (Legendre's duplication formula)

$$\Gamma(n)\Gamma\left(n+\frac{1}{2}\right) = 2^{1-2n}\sqrt{\pi}\Gamma(2n)$$

How we represent Γ-function?

Stirling's series: $\log\Gamma(n+1) = n\log n - n + \frac{1}{2}\log(2\pi n) + \boxed{\sum_{k=1}^{\infty}\frac{s_k}{n^k}}$

For $S(z) := \sum_{k=1}^{\infty} s_k z^k$ and $\sigma(f(z)) = f\left(\frac{z}{1+z}\right)$, we have

$$z\sigma(S) - zS - z + \left(1+\frac{z}{2}\right)\log(1+z) = 0.$$

Example (Legendre's duplication formula)

$$\Gamma(n)\Gamma\left(n+\frac{1}{2}\right) = 2^{1-2n}\sqrt{\pi}\Gamma(2n)$$

How we represent Γ-function?

$$\text{Stirling's series:}\quad \log\Gamma(n+1) = n\log n - n + \frac{1}{2}\log(2\pi n) + \boxed{\sum_{k=1}^{\infty}\frac{s_k}{n^k}}$$

For $S(z) := \sum_{k=1}^{\infty} s_k z^k$ and $\sigma(f(z)) = f\left(\frac{z}{1+z}\right)$, we have

$$z\sigma(S) - zS - z + \left(1+\frac{z}{2}\right)\log(1+z) = 0.$$

Legendre's formula turns into

$$z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1+\frac{z}{2}) - \frac{z}{2}$$

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1 + \frac{z}{2}) - \frac{z}{2}$

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1 + \frac{z}{2}) - \frac{z}{2}$

Building a zero-test

Starting with $\mathbb{Q}[z]$ (has zero-test!)

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1+\frac{z}{2}) - \frac{z}{2}$

Building a zero-test

Starting with $\mathbb{Q}[z]$ (has zero-test!)

1. Adjoin $\log(1+z)$ and $\log(1+\frac{z}{1+z/2})$ (σ-transcendental)

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1+\frac{z}{2}) - \frac{z}{2}$

Building a zero-test

Starting with $\mathbb{Q}[z]$ (has zero-test!)

1. Adjoin $\log(1+z)$ and $\log(1+\frac{z}{1+z/2})$ (σ-transcendental)
2. Adjoin $\log(1+z/2)$ (σ-algebraic)

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1+\frac{z}{2}) - \frac{z}{2}$

Building a zero-test

Starting with $\mathbb{Q}[z]$ (has zero-test!)

1. Adjoin $\log(1+z)$ and $\log(1+\frac{z}{1+z/2})$ (σ-transcendental)
2. Adjoin $\log(1+z/2)$ (σ-algebraic)
3. Adjoin $S(z)$, $S(\frac{z}{2})$, and $S(\frac{z}{1+z/2})$ (σ-algebraic)

Example (Legendre's duplication formula)

Setup

- $S(z)$ is given by $z\sigma(S) - zS - z + \left(1 + \frac{z}{2}\right)\log(1+z) = 0$ and enough terms;
- We want to check $z\left(S\left(\frac{z}{2}\right) - S(z) - S\left(\frac{z}{1+z/2}\right)\right) = \log(1+\frac{z}{2}) - \frac{z}{2}$

Building a zero-test

Starting with $\mathbb{Q}[z]$ (has zero-test!)

1. Adjoin $\log(1+z)$ and $\log(1+\frac{z}{1+z/2})$ (σ-transcendental)
2. Adjoin $\log(1+z/2)$ (σ-algebraic)
3. Adjoin $S(z)$, $S(\frac{z}{2})$, and $S(\frac{z}{1+z/2})$ (σ-algebraic)

And now we can perform the desired zero-test (well, implementation can).

Nuts and bolts: how do we do this?

Difference reduction

Main notions

- Difference polynomial over a difference ring A is an element of $A[X, \sigma(X), \sigma^2(X), \ldots]$.
- Let P be difference polynomial:
  - Leader is $\sigma^\ell X$ appearing in P s.t. ℓ is maximal;
  - let d be the degree of P in $\sigma^\ell X$;
  - $(\ell, d) :=$ Ritt rank of P.

Difference reduction

Main notions

- Difference polynomial over a difference ring A is an element of $A[X, \sigma(X), \sigma^2(X), \ldots]$.
- Let P be difference polynomial:
  - Leader is $\sigma^\ell X$ appearing in P s.t. ℓ is maximal;
  - let d be the degree of P in $\sigma^\ell X$;
  - $(\ell, d) :=$ Ritt rank of P.

Reduction

- P and Q have Ritt ranks (ℓ_P, d_P) and (ℓ_Q, d_Q);
- if $\ell_P \leqslant \ell_Q$ and $d_P \leqslant d_Q$, Q is reducible w.r.t. P
  $\Longleftarrow$ pseudo-Euclidean division of Q by $\sigma^{\ell_Q - \ell_P} P$ w.r.t $\sigma^{\ell_Q} X$.

Difference reduction

Main notions

- Difference polynomial over a difference ring A is an element of $A[X, \sigma(X), \sigma^2(X), \ldots]$.
- Let P be difference polynomial:
  - Leader is $\sigma^\ell X$ appearing in P s.t. ℓ is maximal;
  - let d be the degree of P in $\sigma^\ell X$;
  - $(\ell, d) :=$ Ritt rank of P.

Reduction

- P and Q have Ritt ranks (ℓ_P, d_P) and (ℓ_Q, d_Q);
- if $\ell_P \leqslant \ell_Q$ and $d_P \leqslant d_Q$, Q is reducible w.r.t. P
  $\Longleftarrow$ pseudo-Euclidean division of Q by $\sigma^{\ell_Q - \ell_P} P$ w.r.t $\sigma^{\ell_Q} X$.

Oh lá lá!

differential reducibility $\to$ total ordering

difference reducibility $\to$ partial ordering

Coherent autoreduced set

Autoreduced set

For $\{Q_1, \ldots, Q_s\}$, every Q_i is not reducible w.r.t. the rest
$\implies$ autoreduced set

Coherent autoreduced set

Autoreduced set

For $\{Q_1, \ldots, Q_s\}$, every Q_i is not reducible w.r.t. the rest
$\implies$ autoreduced set

Coherent autoreduced set ("minimal annihilator")

Autoreduced $\{Q_1, \ldots, Q_s\}$ is coherent if $\Delta(Q_i, Q_j)$ reducible to zero $\forall i, j$.
(for a suitable notion of Δ-polynomial)

Coherent autoreduced set

Autoreduced set

For $\{Q_1, \dots, Q_s\}$, every Q_i is not reducible w.r.t. the rest
$\implies$ autoreduced set

Coherent autoreduced set ("minimal annihilator")

Autoreduced $\{Q_1, \dots, Q_s\}$ is coherent if $\Delta(Q_i, Q_j)$ reducible to zero $\forall i, j$.
(for a suitable notion of Δ-polynomial)

Issue

For a coherent autoreduced $Q_1, \dots, Q_s$ in a single indeterminate:

- differential case $\implies s = 1$;
- difference case: can be $s > 1$.

Example

Let $\sigma\colon f(z) \to f(z - z^2)$ and we consider $f = z^2$.

Example

Let $\sigma\colon f(z) \to f(z - z^2)$ and we consider $f = z^2$.

The annihilator of the minimal order is:

$$P_1 = X^4 - 2X^3 - 2X^2\sigma(X) + X^2 - 2X\sigma(X) + \sigma(X)^2$$

Example

Let $\sigma\colon f(z) \to f(z - z^2)$ and we consider $f = z^2$.

The annihilator of the minimal order is:

$$P_1 = X^4 - 2X^3 - 2X^2\sigma(X) + X^2 - 2X\sigma(X) + \sigma(X)^2$$

But there is also:

$$P_2 = X\sigma(X)^3 - 2X\sigma(X)^2 + X\sigma(X) + (-2X + \sigma(X) + X^2 - X\sigma(X))\sigma^2(X)$$

Example

Let $\sigma\colon f(z) \to f(z-z^2)$ and we consider $f = z^2$.

The annihilator of the minimal order is:

$$P_1 = X^4 - 2X^3 - 2X^2\sigma(X) + X^2 - 2X\sigma(X) + \sigma(X)^2$$

But there is also:

$$P_2 = X\sigma(X)^3 - 2X\sigma(X)^2 + X\sigma(X) + (-2X + \sigma(X) + X^2 - X\sigma(X))\sigma^2(X)$$

None of P_1 and P_2 is reducible w.r.t. another!

Solution: one polynomial to rule them all

Key theoretical lemma

Let $Q_1, \ldots, Q_s$ be coherent and autoreduced and Q_1 be of minimal order. Then there exists M:

$$\Big(Q_1(\widetilde{f}) = 0 \;\&\; \forall i > 2\; Q_i(\widetilde{f}) = \mathcal{O}(z^M)\Big) \implies Q_1(\widetilde{f}) = \ldots = Q_s(\widetilde{f}) = 0$$

Solution: one polynomial to rule them all

Key theoretical lemma

Let $Q_1, \ldots, Q_s$ be coherent and autoreduced and Q_1 be of minimal order. Then there exists M:

$$\left(Q_1(\widetilde{f}) = 0 \ \& \ \forall i > 2 \ Q_i(\widetilde{f}) = \mathcal{O}(z^M) \right) \implies Q_1(\widetilde{f}) = \ldots = Q_s(\widetilde{f}) = 0$$

So what?

We can focus on Q_1 and mimic the strategy from the differential algorithm presented by Joris.

Outline of the algorithm

Fix σ-algebraic f. Describe algorithm ZeroTest($Q_1, \ldots, Q_s$)

Input $Q_1, \ldots, Q_s$ — difference polynomials

Output YES if $Q_1(f) = \ldots = Q_s(f) = 0$ and NO otherwise

Outline of the algorithm

Fix σ-algebraic f. Describe algorithm $\mathrm{ZeroTest}(Q_1, \ldots, Q_s)$

Input $Q_1, \ldots, Q_s$ — difference polynomials

Output YES if $Q_1(f) = \ldots = Q_s(f) = 0$ and NO otherwise

Steps (simplified)

1. If there exists Q — initial or a separant of $Q_1, \ldots, Q_s$ not reducible to zero
   - 1.1 if $\mathrm{ZeroTest}(Q, Q_1, \ldots, Q_s)$, return YES
   - 1.2 find who among $Q, Q_1, \ldots, Q_s$ does not vanish at f
   - 1.3 if one of $Q_1, \ldots, Q_s$, return NO

   (by this line, none of the initials and separants vanish at f)

Outline of the algorithm

Fix σ-algebraic f. Describe algorithm $\mathrm{ZeroTest}(Q_1, \ldots, Q_s)$

Input $Q_1, \ldots, Q_s$ — difference polynomials
Output YES if $Q_1(f) = \ldots = Q_s(f) = 0$ and NO otherwise

Steps (simplified)

1. If there exists Q — initial or a separant of $Q_1, \ldots, Q_s$ not reducible to zero
   1.1 if $\mathrm{ZeroTest}(Q, Q_1, \ldots, Q_s)$, return YES
   1.2 find who among $Q, Q_1, \ldots, Q_s$ does not vanish at f
   1.3 if one of $Q_1, \ldots, Q_s$, return NO

   (by this line, none of the initials and separants vanish at f)
2. If a pairwise reminder or a Δ-polynomial Q is not reducible to zero, return $\mathrm{ZeroTest}(Q, Q_1, \ldots, Q_s)$
   (by this line, $Q_1, \ldots, Q_s$ can be assumed coherent autoreduced)

Outline of the algorithm

Fix σ-algebraic f. Describe algorithm $\mathrm{ZeroTest}(Q_1, \ldots, Q_s)$

Input $Q_1, \ldots, Q_s$ — difference polynomials

Output YES if $Q_1(f) = \ldots = Q_s(f) = 0$ and NO otherwise

Steps (simplified)

1. If there exists Q — initial or a separant of $Q_1, \ldots, Q_s$ not reducible to zero
   - 1.1 if $\mathrm{ZeroTest}(Q, Q_1, \ldots, Q_s)$, return YES
   - 1.2 find who among $Q, Q_1, \ldots, Q_s$ does not vanish at f
   - 1.3 if one of $Q_1, \ldots, Q_s$, return NO

   (by this line, none of the initials and separants vanish at f)
2. If a pairwise reminder or a Δ-polynomial Q is not reducible to zero, return $\mathrm{ZeroTest}(Q, Q_1, \ldots, Q_s)$
   (by this line, $Q_1, \ldots, Q_s$ can be assumed coherent autoreduced)
3. Compute special N (Joris talk + lemma from prev slide)
4. If $Q_1(f) = \ldots = Q_s(f) = \mathcal{O}(z^N)$, return YES. Otherwise, NO.

Summary and outlook

We have

- the first zero-test algorithm for σ-algebraic power series
- and it actually works

Summary and outlook

We have

- the first zero-test algorithm for σ-algebraic power series
- and it actually works

We do not have (yet)

- implementation handling both σ and differential equations (we have the theory)
- automatic transform of shift into σ (like $\Gamma \to S$ in the example)
- more examples (e.g., fractional special functions)
- other σ's like $z \to z^k$