On the intrinsic complexity of elimination problems in effective algebraic geometry

Joos Heintz

Universidad de Cantabria
Universidad de Buenos Aires

Joint work with
Bart Kuijpers (Hasselt University)
Andrés Rojas Paredes (Universidad de Buenos Aires)

Escuela Lluis Santaló, RSME and UIMP,
July 16–20, 2012, Santander, España

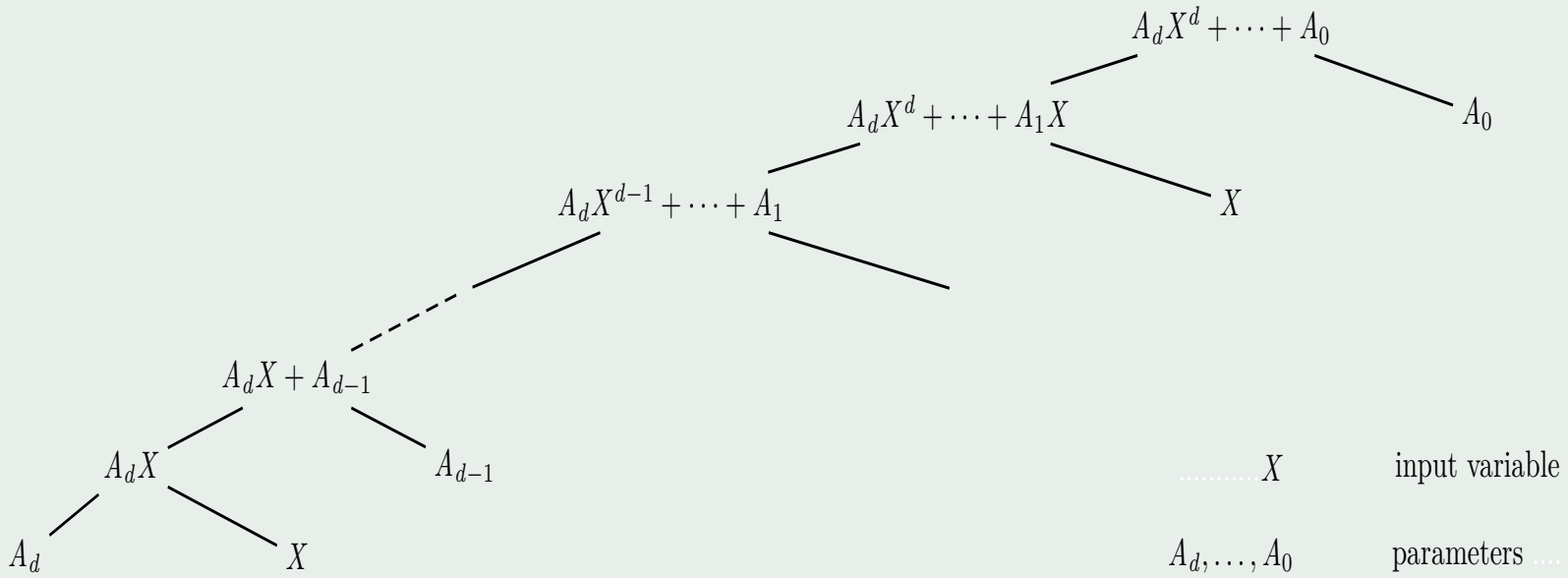
Previous lecture:

- Geometrically robust constructible maps.
- Parameterized arithmetic circuit.
- A family of hard elimination polynomials.

Generic computations

A *generic computation* is a parameterized arithmetic circuit with domain of parameters an *affine space*.

Equivalently:
labelled DAG with *indeterminates* as parameters.
Example: Horner Scheme



Computation model: Routines

Routines will transform
a given robust parameterized arithmetic (input) circuit
into another robust parameterized arithmetic (output) circuit.

Types of variables:

$U, U', U'', \ldots$ individual parameters, W, W', W'' (vectors of) parameters,
$Y, Y', Y'', \ldots$ and Z, Z', Z'' (vectors of) argument variables,
$X, X', X'', \ldots$ (vectors of) standard input variables.

For each variable $X_1, X_2, \ldots, X_n$ there are given generic computations:

$$\begin{array}{lll} R_{X_1}(W_1; X^{(1)}), & R_{X_2}(W_2; X^{(2)}), & \ldots \\ R'_{X_1}(W_{1'}; X^{(1')}), & R'_{X_2}(W_{2'}; X^{(2')}), & \ldots \\ \ldots & \ldots & \ldots \end{array}$$

There are given families of generic computations of the form:

$$\begin{array}{llll} R_{+}(W; U, Y; X), & R'_{+}(W'; U', Y'; X'), & R''_{+}(W''; U'', Y''; X'') & \ldots \\ R_{./}(W; U, Y; X), & R'_{./}(W'; U', Y'; X'), & R''_{./}(W''; U'', Y''; X'') & \ldots \\ R_{add}(W; Y, Z; X), & R'_{add}(W'; Y', Z'; X'), & R''_{add}(W''; Y'', Z''; X'') & \ldots \\ R_{mult}(W; Y, Z; X), & R'_{mult}(W'; Y', Z'; X'), & R''_{mult}(W''; Y'', Z''; X'') & \ldots \\ R_{div}(W; Y, Z; X), & R'_{div}(W'; Y', Z'; X'), & R''_{div}(W''; Y'', Z''; X'') & \ldots . \end{array}$$

The subscripts refer to addition of, and multiplication or division by a parameter (or scalar) and to essential addition, multiplication and division.

Recursive routine: Well behavedness under restrictions

Input: a robust parameterized arithmetic circuit β (parameter domain $\mathcal{M}$).
Output: another robust parameterized arithmetic circuit (same parameter domain)

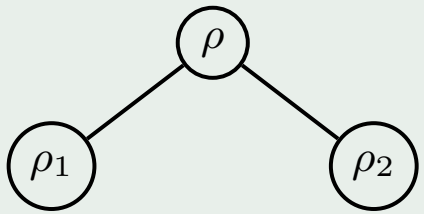
Recursive routine $\mathcal{A}$ (on input β).

For any essential node ρ of β
precompute w_ρ, vector of length $m :=$ length W_ρ of geometrically robust constructible functions defined on $\mathcal{M}$.

If ρ input node: w_ρ vector of complex numbers.

w_ρ are the *parameters* at the node ρ.

Recursive routine step by step



ρ internal node of β.

ρ_1, ρ_2 two ingoing edges.

F_{ρ_1}, F_{ρ_2} already computed vectors of polynomials whose coefficients constitute the entries of a geometrically robust, constructible map defined on $\mathcal{M}$.

$\mathcal{K}_\rho$ image of w_ρ in $\mathbb{C}^m$.
κ_ρ vector of the restrictions to $\mathcal{K}_\rho$ of the canonical projections of $\mathbb{C}^m$.

$\mathcal{K}_\rho$ new parameter domain with basic parameters κ_ρ.

Recursive routine step by step (2)

W_ρ, Y_ρ, Z_ρ vectors of variables.
$X^{(\rho)}$ standard input variables.

Suppose that the node ρ is labelled by an essential multiplication. Let $R^{(\rho)}_{mult}(W_\rho; Y_\rho, Z_\rho; X^{(\rho)})$ be the corresponding generic computation and $R^{(\rho)}_{mult}(\kappa_\rho, Y_\rho, Z_\rho, X^{(\rho)})$ be the specialized generic computation.

Assume that the routine $\mathcal{A}$ satisfies at the node ρ the requirement:

(A) *The by $\mathcal{K}_\rho$ parameterized arithmetic circuit of $R^{(\rho)}_{mult}(\kappa_\rho; Y_\rho, Z_\rho; X^{(\rho)})$, should be consistent and robust.*

Recursive routine step by step (3)

Join the generic computation $R^{(\rho)}_{mult}(W_\rho; Y_\rho, Z_\rho; X^{(\rho)})$ with the previous computations of w_ρ, F_{ρ_1}, F_{ρ_2}.

We obtain a new parameterized arithmetic circuit with final result F_ρ.

By assumptions on w_ρ, F_{ρ_1}, F_{ρ_2}
and requirement (A)
the new circuit is robust if it is consistent.

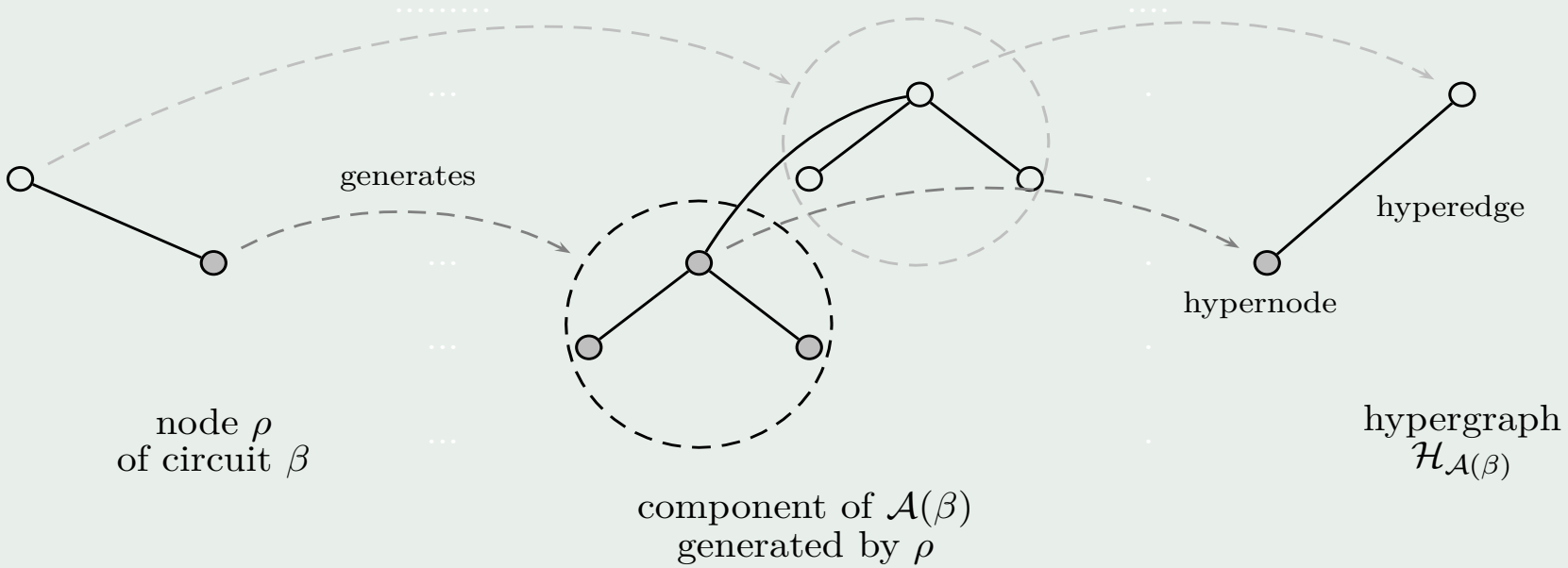
Routine $\mathcal{A}$ *well behaved under restrictions*:

condition (A) is satisfied at any essential node ρ of β
and $\mathcal{A}$ transforms step by step the input circuit β into another *consistent* arithmetic circuit $\mathcal{A}(\beta)$

$\Rightarrow \mathcal{A}(\beta)$ is robust.

Hypergraph $\mathcal{H}_{\mathcal{A}(\beta)}$

Each node ρ of β generates a subcircuit of $\mathcal{A}(\beta)$ which we call the component of $\mathcal{A}(\beta)$ generated by ρ.



The output nodes of each component of $\mathcal{A}(\beta)$ form the hypernodes of a hypergraph $\mathcal{H}_{\mathcal{A}(\beta)}$.

Isoparametricity and well behavedness under reductions

G_ρ intermediate result of β associated with the node ρ.

G_ρ polynomial in $X_1, \ldots, X_n$ whose coefficients form the entries of a geometrically robust, constructible map $\theta_\rho : \mathcal{M} \to \mathcal{T}_\rho$.

The intermediate results of the circuit $\mathcal{A}(\beta)$ at the elements of the hypernode ρ of $\mathcal{H}_{\mathcal{A}(\beta)}$ constitute a polynomial vector which we denote by F_ρ.

We shall now make another requirement on the routine $\mathcal{A}$ at the node ρ of β.

(B) *There exists a geometrically robust, constructible map σ_ρ defined on $\mathcal{T}_\rho$ such that $\sigma_\rho \circ \theta_\rho$ constitutes the coefficient vector of F_ρ.*

Recursive routine $\mathcal{A}$ *isoparametric* (on input β):
requirements (A) and (B) are satisfied at any essential node ρ of β.

Let the recursive routine $\mathcal{A}$ be well behaved under restrictions.

$\mathcal{A}$ *well behaved under reductions* (on input β) if $\mathcal{A}(\beta)$ satisfies the following requirement:

> *Let ρ and ρ' be distinct nodes of β which compute the same intermediate result. Then the intermediate results at the hypernodes ρ and ρ' of $\mathcal{H}_{\mathcal{A}(\beta)}$ are identical.*

Well behavedness under reductions is a well motivated quality attribute of recursive routines.

$\mathcal{A}$ well behaved under reductions $\Rightarrow$ $\mathcal{A}$ isoparametric.

Isoparametricity is computationally meaningful concept.

Operations with recursive routines

Let $\mathcal{A}$, $\mathcal{B}$ be recursive routines, well behaved under restrictions and isoparametric.

Let $\mathcal{B} \circ \mathcal{A}$ be the composed routine.

If $\mathcal{A}$ and $\mathcal{B}$ are isoparametric
$\Rightarrow \mathcal{B} \circ \mathcal{A}$ is isoparametric.

If $\mathcal{A}$ and $\mathcal{B}$ are well behaved under reductions
$\Rightarrow \mathcal{B} \circ \mathcal{A}$ is well behaved under reductions.

Let $\mathcal{A}$ and $\mathcal{B}$ be recursive routines which are well behaved under reductions
$\Rightarrow$ the join of $\mathcal{A}$ with $\mathcal{B}$ is well behaved under reductions.

The join of two isoparametric recursive routines $\mathcal{A}$ and $\mathcal{B}$ is not necessarily isoparametric.
However, condition (B) is still satisfied between the output nodes.

A routine with this property is called *output isoparametric*.

Elementary routines

An *elementary routine* $\mathcal{A}$ of our computation model is obtained by the iterated application of isoparametric recursion, composition, join and union.

We allow also broadcastings and reductions at the interface of two constructions.

We formulate now the property of an elementary routine $\mathcal{A}$ to be output isoparametric:

Let β be a robust, parameterized arithmetic circuit with parameter domain $\mathcal{M}$ and suppose
β admissible input for $\mathcal{A}$

Let θ be the geometrically robust, constructible map defined on $\mathcal{M}$ which represents the coefficient vector of the final results of β.

Let $\mathcal{T}$ be the image of θ.

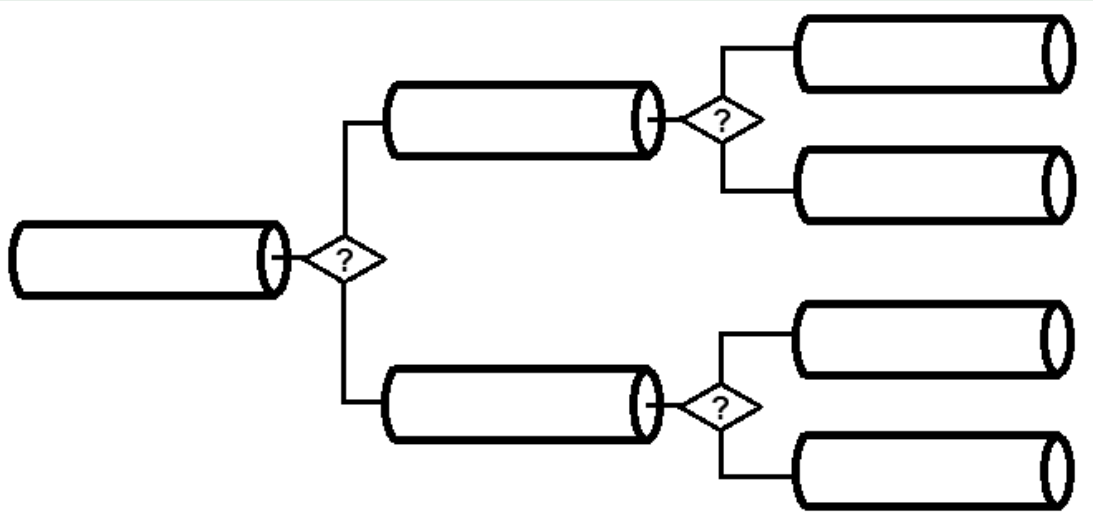
Proposition.

$\mathcal{T}$ is a constructible subset of a suitable affine space and there exists a geometrically robust, constructible map σ defined on $\mathcal{T}$ such that the composition map $\sigma \circ \theta$ represents the coefficient vector of the final results of $\mathcal{A}(\beta)$.

Elementary routines do not contain *branchings*.

A branching parsimonious computation model

An *algorithm* is a dynamic DAG of elementary routines which will be interpreted as pipes.



At the end points of the pipes, *equality tests* between functions of $\mathcal{M}$ determine the next elementary routine (i.e., pipe).

This gives rise to a computation model
which contains branchings.

Branchings depend on a limited type of decisions (equality tests).

Because of this limitation of branchings, we call the algorithms of our model *branching parsimonious*.

Invariants

A tuple of natural numbers determines
the generic computations of our shape list
that intervene in the elementary routine under consideration.

The entries of this tuple are called *invariants* of the circuit β and are denoted by $\mathrm{inv}(\beta)$.

$\mathrm{inv}(\beta)$ determines the architecture of a first elementary routine, say $\mathcal{A}_{\mathrm{inv}(\beta)}$, which admits β as input.

Low level program

A *low level program* of our extended computation model is the transition table of a deterministic Turing machine, which computes a function ψ such that:

- ψ returns on $\mathrm{inv}(\beta)$ the index of an elementary routine $\mathcal{A}_{\mathrm{inv}(\beta)}$, which admits β as input.
- ψ determines the equality tests to be realized with the final results of $\mathcal{A}_{\mathrm{inv}(\beta)}(\beta)$.

Depending on these equality tests,
ψ determines an index value corresponding to a new elementary routine which admits $\mathcal{A}_{\mathrm{inv}(\beta)}(\beta)$ as input, etc.

Procedures

A given algorithm $\mathcal{A}$ of our branching parsimonious computation model *computes* (only) *parameters* if for any admissible input β the final results of $\mathcal{A}(\beta)$ are all parameters.

Let $\mathcal{A}$ be such an algorithm. The input and auxiliary variables become eliminated.

In order to reduce $\mathcal{A}(\beta)$ to a *final output circuit* $\mathcal{A}_{\text{final}}(\beta)$ whose intermediate results are only parameters, we may *collect garbage*.

If we consider $\mathcal{A}$ as a partial map $\beta \to \mathcal{A}_{\text{final}}(\beta)$, we call $\mathcal{A}$ a *procedure*.

Procedures

A procedure $\mathcal{A}$ of our model is the composition of two algorithms $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$such that:

(i) *$\mathcal{A}^{(1)}$ computes only parameters, β is admissible for $\mathcal{A}^{(1)}$ and none of the indeterminates $Y_1, \ldots, Y_s$ is introduced in $\mathcal{A}^{(1)}(\beta)$ as auxiliary variable (all other auxiliary variables become eliminated during the execution of $\mathcal{A}^{(1)}$).*

(ii) *$\mathcal{A}^{(1)}_{final}(\beta)$ is an admissible input for $\mathcal{A}^{(2)}$, the indeterminates $Y_1, \ldots, Y_s$ occur as auxiliary variables in $\mathcal{A}^{(2)}(\mathcal{A}^{(1)}_{final}(\beta))$ and the final results of $\mathcal{A}^{(2)}(\mathcal{A}^{(1)}_{final}(\beta))$ depend only on $\pi_1, \ldots, \pi_r$ and $Y_1, \ldots, Y_s$.*

The subalgorithm $\mathcal{A}^{(1)}$ is output isoparametric.

Conditions (i) and (ii) represent an architectural restriction which is justified when it makes sense to require that

on input β the number of essential additions and multiplications contained in $\mathcal{A}_{\text{final}}(\beta)$ is bounded by a function which depends only on $\text{inv}(\beta)$.

A hard elimination problem

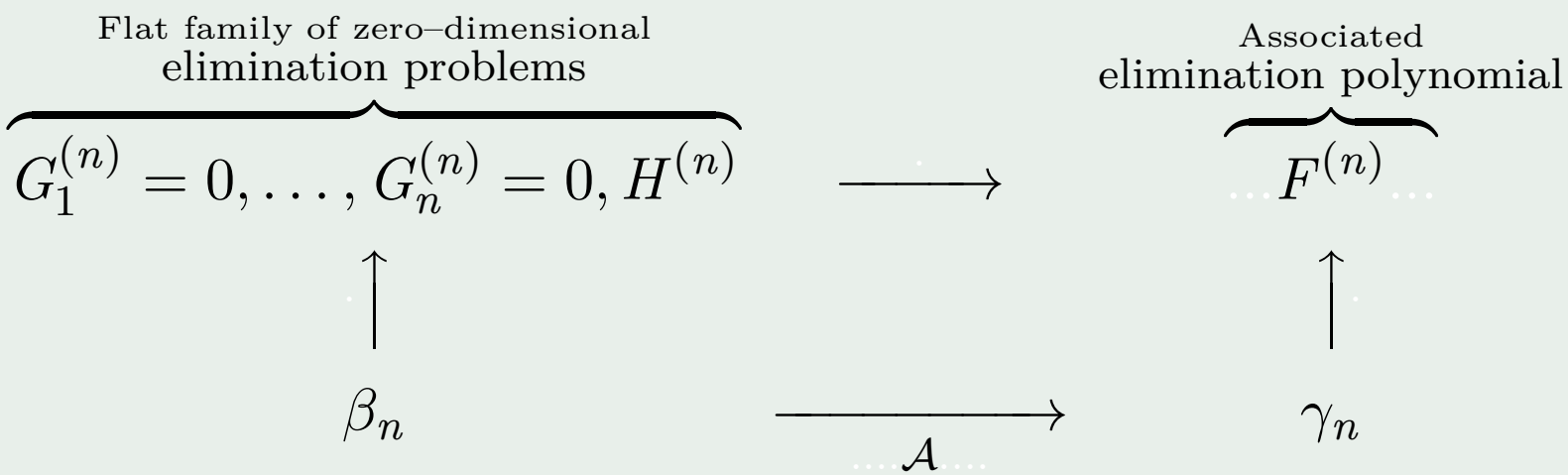
$n \in \mathbb{N}$,
$S_1, \ldots, S_n, T, U_1, \ldots, U_n, X_1, \ldots, X_n$ indeterminates.

Let $U := (U_1, \ldots, U_n)$, $S := (S_1, \ldots, S_n)$, $X := (X_1, \ldots, X_n)$ and $G_1^{(n)} := X_1^2 - X_1 - S_1, \ldots, G_n^{(n)} := X_n^2 - X_n - S_n$,
$H^{(n)} := \sum_{1 \leq i \leq n} 2^{i-1} X_i + T \prod_{1 \leq i \leq n} (1 + (U_i - 1) X_i)$.

$G_1^{(n)} = 0, \ldots, G_n^{(n)} = 0$ and $H^{(n)}$ represent a *flat family of zero–dimensional elimination problems* with associated elimination polynomial $F^{(n)}$

There exists an ordinary division–free arithmetic circuit β_n of size $O(n)$ over $\mathbb{C}$ with inputs $S_1, \ldots, S_n,\ T,\ U_1, \ldots, U_n,$ $X_1, \ldots, X_n$ and final results $G_1^{(n)}, \ldots, G_n^{(n)}, H^{(n)}$.

Let $\mathcal{A}$ be an essentially division–free procedure, such that $\gamma_n := \mathcal{A}_{\text{final}}(\beta_n)$ essentially division–free, robust parameterized arithmetic circuit which depends on the basic parameters $S_1, \ldots, S_n,\ T,\ U_1, \ldots, U_n$ and the input Y and its final result is a power of $F^{(n)}$.



Theorem 7.

The circuit γ_n performs at least $\Omega(2^{\frac{n}{2}})$ essential multiplications and at least $\Omega(2^n)$ multiplications with parameters. The circuit γ_n has non–scalar size at least $\Omega(2^n)$.

Theorem 7 implies the asymptotic optimality of the Kronecker algorithm within our computation model.

KRONECKER

Polynomial Equation System Solver

`http://lecerf.perso.math.cnrs.fr/software/kronecker/index.html`
`http://www.mathemagix.org`

Approximative computations

β essentially division-free, robust parameterized arithmetic circuit with parameter domain $\mathcal{M}$, basic parameters $\pi_1, \ldots, \pi_r$, inputs $X_1, \ldots, X_n$ and single final result G,
$U_1, \ldots, U_r$ parameter variables, $U := (U_1, \ldots, U_r)$, $\pi := (\pi_1, \ldots, \pi_r)$, $X := (X_1, \ldots, X_n)$.

$\mathfrak{a}$ vanishing ideal of $\overline{\mathcal{M}}$ in $\mathbb{C}[U]$,
$P \in \mathbb{C}[U]$ fixed polynomial such that $\overline{\mathcal{M}}_P$ Zariski open and dense in $\mathcal{M}$, ϵ a new indeterminate.

Approximative parameter instance

An *approximative parameter instance* for β is a vector $u(\epsilon) = (u_1(\epsilon), \ldots, u_r(\epsilon)) \in \mathbb{C}((\epsilon))^r$,
meromorphic map germ at 0, such that
$\mathfrak{a}$ vanishes at $u(\epsilon)$ and $P(u(\epsilon)) \neq 0$.

Let $u(\epsilon)$ approximative parameter instance for β.
$\exists\ \Delta$ open disc around 0 such that for any $c \in \Delta - \{0\}$ the germ $u(\epsilon)$ is holomorphic at c and $P(u(c)) \neq 0$.
$\Rightarrow \mathfrak{a}$ vanishes at $u(c)$, $u(c) \in \mathcal{M}$.

Let $\phi : \mathcal{M} \to \mathbb{C}^m$ geometrically robust constructible map.

Lemma

There exists an open disc Δ of $\mathbb{C}$ around the origin and a germ ψ of meromorphic functions at the origin such that $u(\epsilon)$ and ψ are holomorphic on $\Delta - \{0\}$ and such that any complex number $c \in \Delta - \{0\}$ satisfies the conditions $P(u(c)) \neq 0$ and $\psi(c) = \phi(u(c))$.

There exists an open disc Δ of $\mathbb{C}$ around 0 such that for any node ρ of β with intermediate result $G_\rho(\pi, X)$ the expression $G_\rho(u(\epsilon), X)$ defines
a polynomial in $X_1, \ldots, X_n$ whose coefficients are meromorphic functions on Δ, holomorphic on $\Delta - \{0\}$.

Approximative β–computation

$\beta^{(u(\epsilon))}$ labelled DAG of β where we assign to each node ρ of β the polynomial $G_\rho(u(\epsilon), X)$.
We call $\beta^{(u(\epsilon))}$ an *approximative β–computation* and denote by $G^{(u(\epsilon))}$ the final result of $\beta^{(u(\epsilon))}$.

The approximative β–computation $\beta^{(u(\epsilon))}$ *represents* the polynomial $H \in \mathbb{C}[X]$ if
$\exists\ H^{(u(\epsilon))} \in \mathbb{C}[[\epsilon]][X]$ with $G^{(u(\epsilon))} = H + \epsilon H^{(u(\epsilon))}$.
W_β set of coefficient vectors of the final results of $\beta^{(u)}$, $u \in \mathcal{M}$.

Theorem (Alder '84, Lickteig '90)

The following three conditions are equivalent:

(i) there exists an approximative β–computation that represents $H \in \mathbb{C}[X]$.

(ii) there exists a sequence $(u_k)_{k\in\mathbb{N}}$ with $u_k \in \mathcal{M}$ such that the final results of the sequence $(\beta^{(u_k)})_{k\in\mathbb{N}}$ of ordinary circuits converge to H in $\mathbb{C}[X]$.

(iii) the coefficient vector of H belongs to $\overline{W}_\beta$.

An essentially division–free procedure

Let $u(\epsilon)$ approximative parameter instance for β such that $\beta^{(u(\epsilon))}$ represents a polynomial $H \in \mathbb{C}[X]$ with coefficient vector $h \in W_\beta$.

Essentially division–free procedure $(\mathcal{A}^{(1)}, \mathcal{A}^{(2)})$:

- (i) compute the coefficient vector $\theta(u(\epsilon))$ of $G^{(u(\epsilon))}$ by interpolation $(\mathcal{A}^{(1)})$
- (ii) compute $G^{(u(\epsilon))}$ from $\theta(u(\epsilon))$ and $X_1, \ldots, X_n$ $(\mathcal{A}^{(2)})$

Finally:
Compute H as in (ii) from $h = \lim_{\epsilon \to 0} \theta(u(\epsilon))$ and $X_1, \ldots, X_n$.

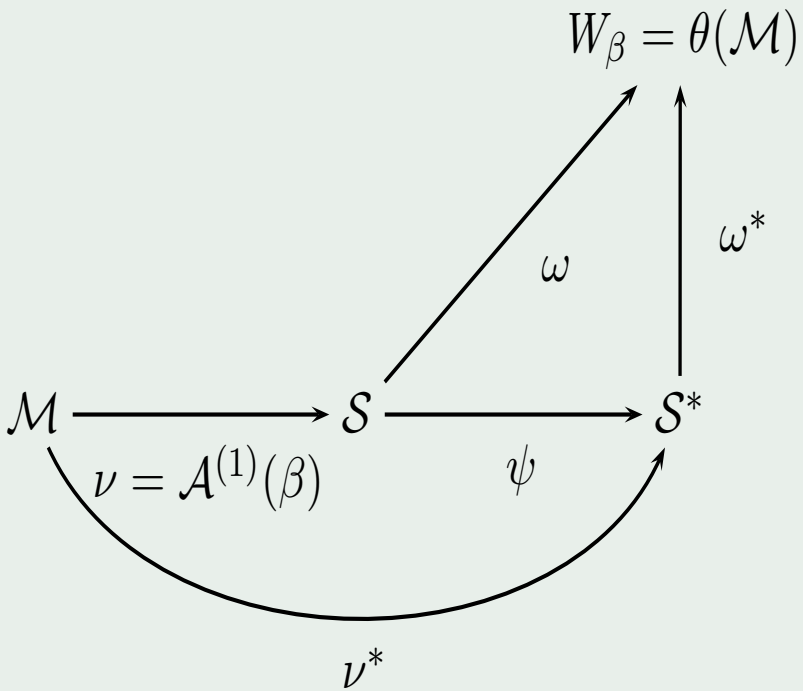
Robust encodings

Let $\mathcal{A} = (\mathcal{A}^{(1)}, \mathcal{A}^{(2)})$ essentially division–free procedure,
$\mathcal{A}$ accepts β as input and returns $\mathcal{A}_{\text{final}}(\beta)$ with final result G,

$\gamma := \mathcal{A}_{\text{final}}(\beta)$ essentially division–free, robust arithmetic circuit with parameter domain $\mathcal{M}$,
ν output of $\mathcal{A}^{(1)}(\beta)$, $\nu : \mathcal{M} \to \mathcal{S}$ geometrically robust, constructible
$\theta :=$ coefficient vector of G.

$\mathcal{A}_{\text{final}}(\beta)$ yields:
$\psi : \mathcal{S} \to \mathbb{C}^m$ geometrically robust constructible map,
ω^* vector of m–variate polynomials such that $\theta = \omega^* \circ \psi \circ \nu$ holds.

$\mathcal{S}, \mathcal{S}^* := \psi(\mathcal{S})$ *data structures*
W_β an (abstract) *object class* (W_β represents $\{G^{(u)}, u \in \mathcal{M}\}$),
ω^* *holomorphic encoding* of W_β by $\mathcal{S}^*$ ($\omega^* : \mathcal{S}^* \to W_\beta$ surjective polynomial map),
ω *continuous encoding* of the object class W_β by the data structure $\mathcal{S}$, and ω^*, ω robust *encodings*:



Evaluating the polynomial H

We wish to evaluate the polynomial H

The approximative β–computation $\beta^{(u(\epsilon))}$ represents the polynomial $H \Rightarrow$
the sequence $(G^{(u_k)})_{k\in\mathbb{N}}$ converges to H.

The sequences $(\nu(u_k))_{k\in\mathbb{N}}$ and $(\nu^*(u_k))_{k\in\mathbb{N}}$ converge to points s and s^* of $\mathcal{S}$ and $\mathcal{S}^* \Rightarrow$
$\omega(s) = \omega^*(s^*)$ forms the coefficient vector of H.

Reinterpret γ as a robust parameterized arithmetic circuit with parameter domain $\mathcal{S}^* \Rightarrow$
$\gamma^{(s^*)}$ becomes an ordinary division–free arithmetic circuit in $\mathbb{C}[X]$ whose single final result is H.

Example

L, n natural numbers, $r := (L+n+1)^2$, $X_1, \ldots, X_n$ input variables, $\mathcal{M}_{L,n} := \mathbb{C}^r$ and $\pi_1, \ldots, \pi_r$ the canonical projections of $\mathcal{M}_{L,n}$ onto $\mathbb{C}^1$.

There exists a totally division–free generic computation $\beta_{L,n}$ with a single final result $G_{L,n}$ such that any polynomial $H \in \mathbb{C}[X_1, \ldots, X_n]$ is evaluable by at most L essential multiplications iff $\exists u \in \mathcal{M}_{L,n}$ such that $H = G_{L,n}^{(u)}$ holds.

Interpret $\beta_{L,n}$ as a robust parameterized arithmetic circuit with parameter domain $\mathcal{M}_{L,n}$, basic parameters $\pi_1, \ldots, \pi_r$ and inputs $X_1, \ldots, X_n$.

Let $\mathcal{A} = (\mathcal{A}^{(1)}, \mathcal{A}^{(2)})$ essentially division–free procedure which on input $\beta_{L,n}$ returns a robust parameterized arithmetic circuit $\mathcal{A}_{\text{final}}(\beta_{L,n})$ whose single final result is $G_{L,n}$.

$\nu_{L,n}$ output of $\mathcal{A}^{(1)}(\beta_{L,n})$ and $\mathcal{S}_{L,n}$ image of $\nu_{L,n}$. We think $\mathcal{S}_{L,n}$ as a constructible subset of an affine space $\mathbb{C}^{p_{L,n}}$.

$\mathcal{A}^{(2)}(\mathcal{A}^{(1)}(\beta))$ yields:
$\psi_{L,n} : \mathcal{S}_{L,n} \to \mathbb{C}^{m_{L,n}}$ geometrically robust constructible map
$\omega^*_{L,n}$ vector of $m_{L,n}$–variate polynomials such that for $\nu^*_{L,n} := \psi_{L,n} \circ \nu_{L,n}$ the vector of coefficients of $G_{L,n}$ with respect to the variables $X_1, \ldots, X_n$ can be written as $\omega^*_{L,n} \circ \nu^*_{L,n}$.

Lower bound

The size $p_{L,n}$ of the continuous encoding $\omega^*_{L,n} \circ \psi_{L,n} : \mathcal{S}_{L,n} \to W_{\beta_{L,n}}$ of $W_{\beta_{L,n}}$ is $4(L+n+1)^2+2$ whereas for $\mathcal{S}^*_{L,n} := \psi_{L,n}(\mathcal{S}) = \nu^*_{L,n}(\mathcal{M})$ the map $\omega^*_{L,n} : \mathcal{S}^*_{L,n} \to W_{\beta_{L,n}}$ represents a holomorphic encoding of $W_{\beta_{L,n}}$ of size $m_{L,n} = 2^{\Omega(Ln)}$.

Thus, there are natural classes of polynomials which have continuous encodings of "small size" whereas their holomorphic encodings may become necessarily "large".