On the complexity of the D5 principle

X. Dahan[⋆], M. Moreno Maza[†], É. Schost[⋆] & Y. Xie[†]

LIX, École polytechnique, Palaiseau, France.

†: ORCCA, University of Western Ontario, London, Canada.

What is the D5 principle ?

Model of computation for algebraic numbers developed by J. Della-Dora, C. Discrescenzo and D. Duval. (1985)

- *easy case: f irreducible* polynomial over $\mathbb{Q}$, $\alpha_1, \ldots, \alpha_s$ its roots.

$$\{\text{Computing over any } \mathbb{Q}(\alpha_i)\} \longleftrightarrow \{\text{Computing over } \mathbb{Q}[X]/f\}$$

What is the D5 principle ?

Model of computation for algebraic numbers developed by J. Della-Dora, C. Discrescenzo and D. Duval. (1985)

- *easy case: f irreducible* polynomial over $\mathbb{Q}$, $\alpha_1, \dots, \alpha_s$ its roots.

$$\{\text{Computing over any } \mathbb{Q}(\alpha_i)\} \longleftrightarrow \{\text{Computing over } \mathbb{Q}[X]/f\}$$

- *f is not irreducible but squarefree:* For the *addition* and the *multiplication* computing over $\mathbb{Q}[X]/f$ still works. For the *division* two possibilities:

What is the D5 principle ?

Model of computation for algebraic numbers developed by
J. Della-Dora, C. Discrescenzo and D. Duval. (1985)

- *easy case: f irreducible* polynomial over $\mathbb{Q}$, $\alpha_1, \ldots, \alpha_s$ its roots.

$$\{\text{Computing over any } \mathbb{Q}(\alpha_i)\} \longleftrightarrow \{\text{Computing over } \mathbb{Q}[X]/f\}$$

- *f is not irreducible but squarefree:* For the *addition* and the *multiplication* computing over $\mathbb{Q}[X]/f$ still works. For the *division* two possibilities:
  - factorize $f(X) = f_1(X) \ldots f_t(X)$, and work over $\mathbb{Q}[X]/f_i$ as above.

What is the D5 principle ?

Model of computation for algebraic numbers developed by J. Della-Dora, C. Discrescenzo and D. Duval. (1985)

- *easy case:* f *irreducible* polynomial over $\mathbb{Q}$, $\alpha_1, \ldots, \alpha_s$ its roots.

$$\{\text{Computing over any } \mathbb{Q}(\alpha_i)\} \longleftrightarrow \{\text{Computing over } \mathbb{Q}[X]/f\}$$

- f *is not irreducible but squarefree:* For the *addition* and the *multiplication* computing over $\mathbb{Q}[X]/f$ still works. For the *division* two possibilities:
  - factorize $f(X) = f_1(X) \ldots f_t(X)$, and work over $\mathbb{Q}[X]/f_i$ as above.
  - . . . or use the D5 principle as in this example:

Example of the D5 Principle

$$\begin{cases} f(X) = X^4 - 13X^2 + 36 \in \mathbb{Q}[X], \text{ reducible,} \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \text{ its roots.} \end{cases}$$

Problem: $g_i(Z) := Z^3 + 3\alpha_i Z^2 + 12Z + 4\alpha_i \in \mathbb{Q}(\alpha_i)[Z],\ i = 1, \ldots, 4$

Are the g_i's squarefree ?

Example of the D5 Principle

$$\begin{cases} f(X) = X^4 - 13X^2 + 36 \in \mathbb{Q}[X], \text{ reducible,} \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \text{ its roots.} \end{cases}$$

Problem: $g_i(Z) := Z^3 + 3\alpha_i Z^2 + 12Z + 4\alpha_i \in \mathbb{Q}(\alpha_i)[Z],\ i = 1, \ldots, 4$
Are the g_i's squarefree ?

- With factorization: $f(X) = (X-2)(X+2)(X-3)(X+3)$, plug each root in g_i and check if the discriminant is zero.

Example of the D5 Principle

$$\begin{cases} f(X) = X^4 - 13X^2 + 36 \in \mathbb{Q}[X], \text{ reducible,} \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \text{ its roots.} \end{cases}$$

Problem: $g_i(Z) := Z^3 + 3\alpha_i Z^2 + 12Z + 4\alpha_i \in \mathbb{Q}(\alpha_i)[Z],\ i = 1, \ldots, 4$
Are the g_i's squarefree ?

- With factorization: $f(X) = (X-2)(X+2)(X-3)(X+3)$, plug each root in g_i and check if the discriminant is zero.
- With D5 principle: represent all $\mathbb{Q}(\alpha_i)$ using $\mathbb{Q}[X]/f$

Example of the D5 Principle

$$\begin{cases} f(X) = X^4 - 13X^2 + 36 \in \mathbb{Q}[X], \text{ reducible,} \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \text{ its roots.} \end{cases}$$

Problem: $g_i(Z) := Z^3 + 3\alpha_i Z^2 + 12Z + 4\alpha_i \in \mathbb{Q}(\alpha_i)[Z], \; i = 1, \ldots, 4$

Are the g_i's squarefree ?

- With factorization: $f(X) = (X-2)(X+2)(X-3)(X+3)$, plug each root in g_i and check if the discriminant is zero.
- With D5 principle: represent all $\mathbb{Q}(\alpha_i)$ using $\mathbb{Q}[X]/f$
  - define $g(X, Z) := Z^3 + 3XZ^2 + 12Z + 4X$
  - $\mathsf{Discr}(g) = -432(X-2)^2(X+2)^2$

Example of the D5 Principle

$$\begin{cases} f(X) = X^4 - 13X^2 + 36 \in \mathbb{Q}[X], \text{ reducible,} \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \text{ its roots.} \end{cases}$$

Problem: $g_i(Z) := Z^3 + 3\alpha_i Z^2 + 12Z + 4\alpha_i \in \mathbb{Q}(\alpha_i)[Z],\ i = 1, \ldots, 4$
Are the g_i's squarefree ?

- With factorization: $f(X) = (X-2)(X+2)(X-3)(X+3)$, plug each root in g_i and check if the discriminant is zero.
- With D5 principle: represent all $\mathbb{Q}(\alpha_i)$ using $\mathbb{Q}[X]/f$
  - define $g(X, Z) := Z^3 + 3XZ^2 + 12Z + 4X$
  - $\text{Discr}(g) = -432(X-2)^2(X+2)^2$
  - $\alpha_i^2 \neq 4 \Leftrightarrow g_i$ is squarefree

From one to many variables

Previous example $\longrightarrow$ the problem has actually led to:

$$\begin{array}{ccccc} \mathbb{Q}[X]/f & \simeq & \mathbb{Q}[X]/X^2-4 & \times & \mathbb{Q}[X]/(\frac{f}{X^2-4}) \\ A & \simeq & B & \times & C \\ & & g \text{ is not squarefree} & & g \text{ is squarefree} \end{array}$$

From one to many variables

Previous example $\longrightarrow$ the problem has actually led to:

$$\begin{array}{ccccc} \mathbb{Q}[X]/f & \simeq & \mathbb{Q}[X]/X^2-4 & \times & \mathbb{Q}[X]/(\frac{f}{X^2-4}) \\ A & \simeq & B & \times & C \\ A[Y]/(h \bmod f) & \simeq & B[Y]/(h \bmod X^2-4) & \times & C[Y]/(h \bmod \frac{f}{X^2-4}) \end{array}$$

where $h \in \mathbb{Q}[X, Y]$.

From one to many variables

Previous example $\longrightarrow$ the problem has actually led to:

$$
\begin{array}{ccccc}
\mathbb{Q}[X]/f & \simeq & \mathbb{Q}[X]/X^2-4 & \times & \mathbb{Q}[X]/(\frac{f}{X^2-4}) \\
A & \simeq & {\color{darkred}B} & \times & {\color{darkblue}C} \\
A[Y]/(h \bmod f) & \simeq & {\color{darkred}B}[Y]/(h \bmod X^2-4) & \times & {\color{darkblue}C}[Y]/(h \bmod \frac{f}{X^2-4})
\end{array}
$$

where $h \in \mathbb{Q}[X, Y]$.

$\Rightarrow$ Generalization: use of *Triangular sets*...

Triangular sets

- Family of n monic polynomials
- Finite number of solutions, which are simple

$$\begin{array}{|l} T_n(X_1, X_2, \ldots, X_n), \\ \vdots \\ T_2(X_1, X_2), \\ T_1(X_1) \end{array}$$

- Useful in polynomial systems solving (decomposition of varieties, since the 90's): Lazard, Kalkbrener, Moreno Maza, Wang, Aubry, Dahan-Moreno Maza-Schost-Wu-Xie etc.

There is the following isomorphism:

- $k[X_1, \ldots, X_n]/T \simeq k[X_1, \ldots, X_n]/\mathfrak{m}_1 \times \cdots \times k[X_1, \ldots, X_n]/\mathfrak{m}_s$, where $\mathfrak{m}_1, \ldots, \mathfrak{m}_s$ are the primary ideals of T.
- Primary decomposition $\longrightarrow$ factorization. Again, the D5 principle avoids the factorization.

Motivation

Newton algorithm modulo a triangular set:

- $F = f_1, \ldots, f_n$ a zero-dimensional radical polynomial system over $\mathbb{Q}$
- Obtain a triangular decomposition modulo a prime p
- Lift each triangular set with Newton-Hensel.

The Newton iterator uses the following computation:

$$\mathsf{Jac}(T)\mathsf{Jac}(F)^{-1}F \mod T,$$

which requires at least one division modulo a triangular set. Estimation of the complexity ?

Splitting during the D5 process

Quasi-inversion modulo a triangular set T:

- T is split into triangular sets $R_1, \ldots, R_\ell$ and $R_{\ell+1}, \ldots, R_m$, with

  $\alpha = 0$ mod each $R_i,\ i \leq \ell$ and α is a unit mod each $R_i,\ i > \ell$

- Then one can continue the computations in "parallel", modulo $R_1, \ldots, R_m$ (less costly than primary decomposition)

Splitting during the D5 process

Quasi-inversion modulo a triangular set T:

- T is split into triangular sets $R_1, \dots, R_\ell$ and $R_{\ell+1}, \dots, R_m$, with

  $\alpha = 0$ mod each $R_i, \ i \leq \ell$ and α is a unit mod each $R_i, \ i > \ell$

- Then one can continue the computations in "parallel", modulo $R_1, \dots, R_m$ (less costly than primary decomposition)

Problem raised: After a splitting, one needs to compute the map:

$$k[X_1, \dots, X_n]/T \to k[X_1, \dots, X_n]/R_1 \times \cdots \times k[X_1, \dots, X_n]/R_m$$

In general, a good complexity estimate for that is not always obvious. . .

Main result

- No complexity results for algorithms relying on this principle.
- What is missing: complexity of the *quasi-inverse* modulo a triangular set T. (all algorithms rely on it)
- Complexity measures used:
  - let $T = (T_1, \ldots, T_n)$, denote $d_i := \deg_{X_i}(T_i)$,
  - $\mathsf{M}(d)$ is an upper bound for the cost of the multiplication of two polynomials of degree at most d.
    $\mathsf{M}(d) \in O(d \log(d) \log\log(d))$.

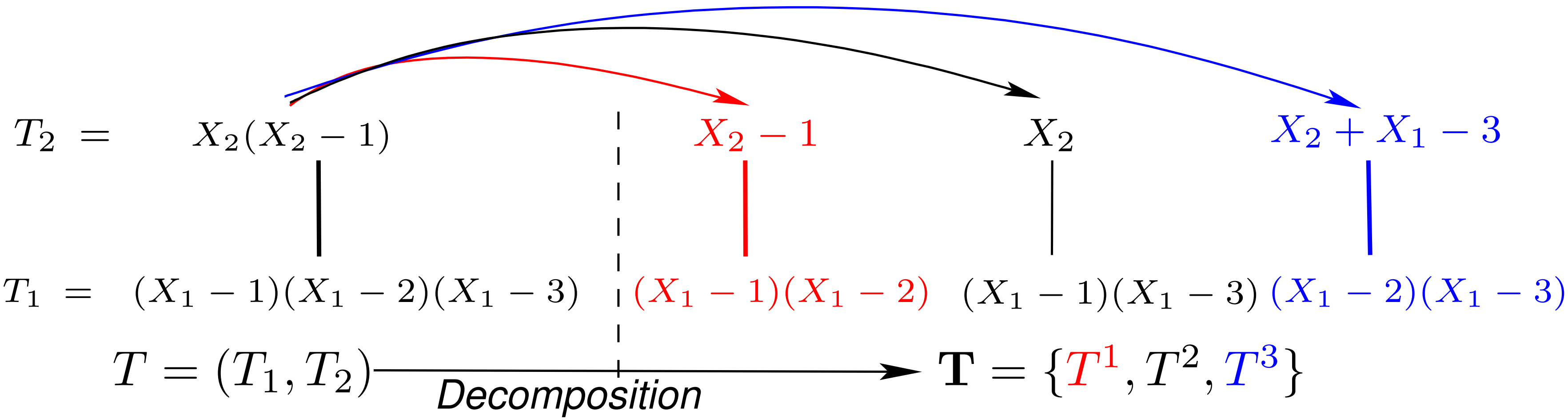
Theorem 0 *There exists a* $\mathsf{C} > 0$*, such that for all triangular set* T *and for all* $f \in k[X_1, \ldots, X_n]$ *with* $\deg_{X_i}(f) < d_i$*, the computation of the quasi-inverse of* f *modulo* T *requires at most:*

$$\mathsf{C}^n \prod_{i=1}^n \mathsf{M}(d_i) \log^3(d_i)\,, \qquad (\text{quasi} - \text{linear in } d_1 \ldots d_n)$$

operations over k.

The splitting problem

A quasi-inverse computation may lead to the following split:



Splitting an element p from T to $\mathbf{T}$ requires then to compute:

$$p \bmod (X_1-1)(X_1-2)\,,\; p \bmod (X_1-1)(X_1-3)\,,\; p \bmod (X_1-2)(X_1-3),$$

there are some *redundancies* $\rightarrow$ bad for complexity estimation.

The main topic of this work is to *remove* this difficulty...

Solving the splitting problem 1/2

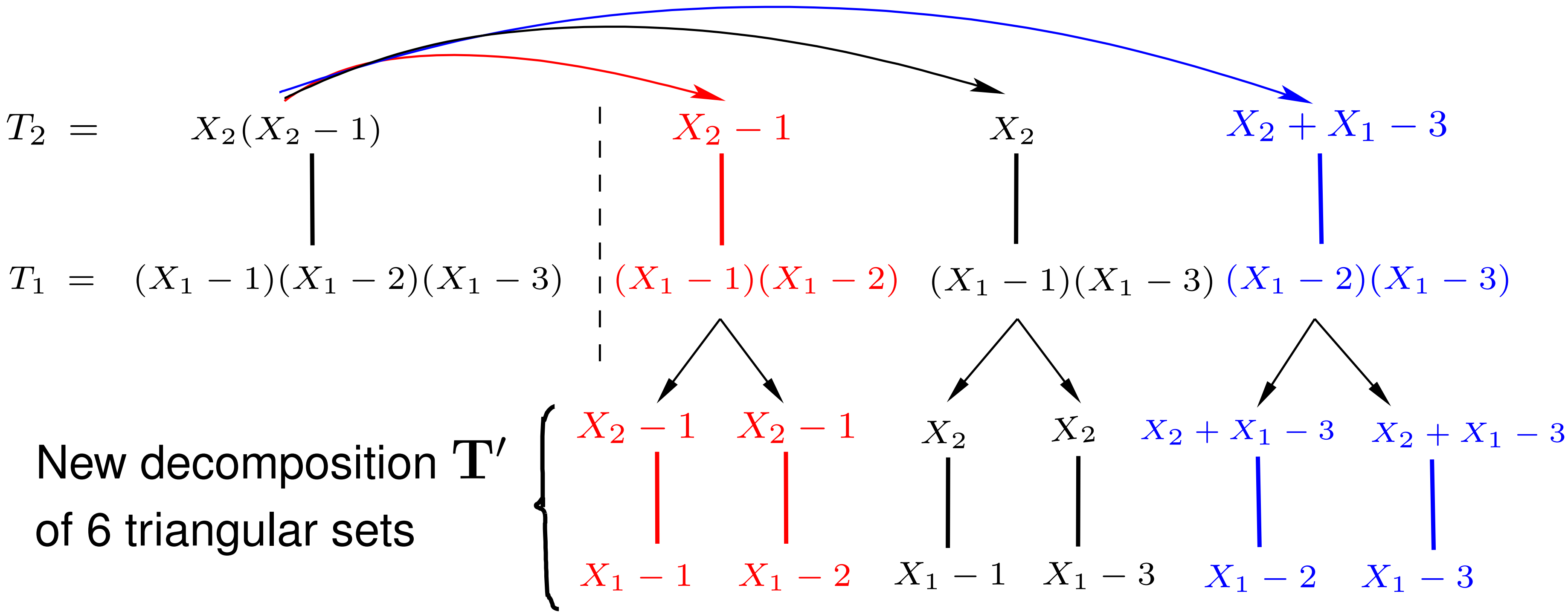
A solution is to *refine* the triangular decomposition as follows:

$$
\begin{array}{lll|llll}
T_2 & = & X_2(X_2-1) & X_2-1 & X_2 & X_2+X_1-3 \\
T_1 & = & (X_1-1)(X_1-2)(X_1-3) & (X_1-1)(X_1-2) & (X_1-1)(X_1-3) & (X_1-2)(X_1-3)
\end{array}
$$

$$T=(T_1,T_2) \xrightarrow{\textit{Decomposition}} \mathbf{T}=\{T^1,T^2,T^3\}$$

Solving the splitting problem 1/2

A solution is to *refine* the triangular decomposition as follows:



Now splitting p from T to $\mathbf{T}'$ requires only to compute:

$$p \bmod X_1 - 1\,, \qquad p \bmod X_1 - 3\,, \qquad p \bmod X_1 - 2$$

No more redundancy.

Solving the splitting problem 2/2

The previous decomposition has the following property:

Definition 0 *$T \neq T'$ two triangular sets.*

Let ℓ the least integer s.t. $T_\ell \neq T'_\ell$.

The pair T, T' is critical if $(T_\ell, T'_\ell) \neq (1)$ in $k[X_1, \ldots, X_{\ell-1}]/(T_1, \ldots, T_{\ell-1})[X_\ell]$.

A decomposition $\mathbf{T}$ of T is non-critical if $\mathbf{T}$ has no critical pairs.

- Need to remove the critical pairs from any decomposition of triangular sets.
- Achieve this task with a good complexity $\rightarrow$ use of "coprime factorization" algorithm.
- By definition, this requires to compute gcd modulo a triangular set.

The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

split in $\mathbb{L}_n$

gcd in $\mathbb{L}_{n-1}[X_n]$ in degree d $\longrightarrow$ coprime factorization in $\mathbb{L}_{n-1}[X_n]$ in degree d

The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

$$\begin{tikzcd}
 & \text{split in } \mathbb{L}_n \arrow[dl] & \\
\text{quasi-inverse in } \mathbb{L}_n & & \\
\text{gcd in } \mathbb{L}_{n-1}[X_n] \text{ in degree } d \arrow[u] \arrow[rr] & & \text{coprime factorization in } \mathbb{L}_{n-1}[X_n] \text{ in degree } d
\end{tikzcd}$$

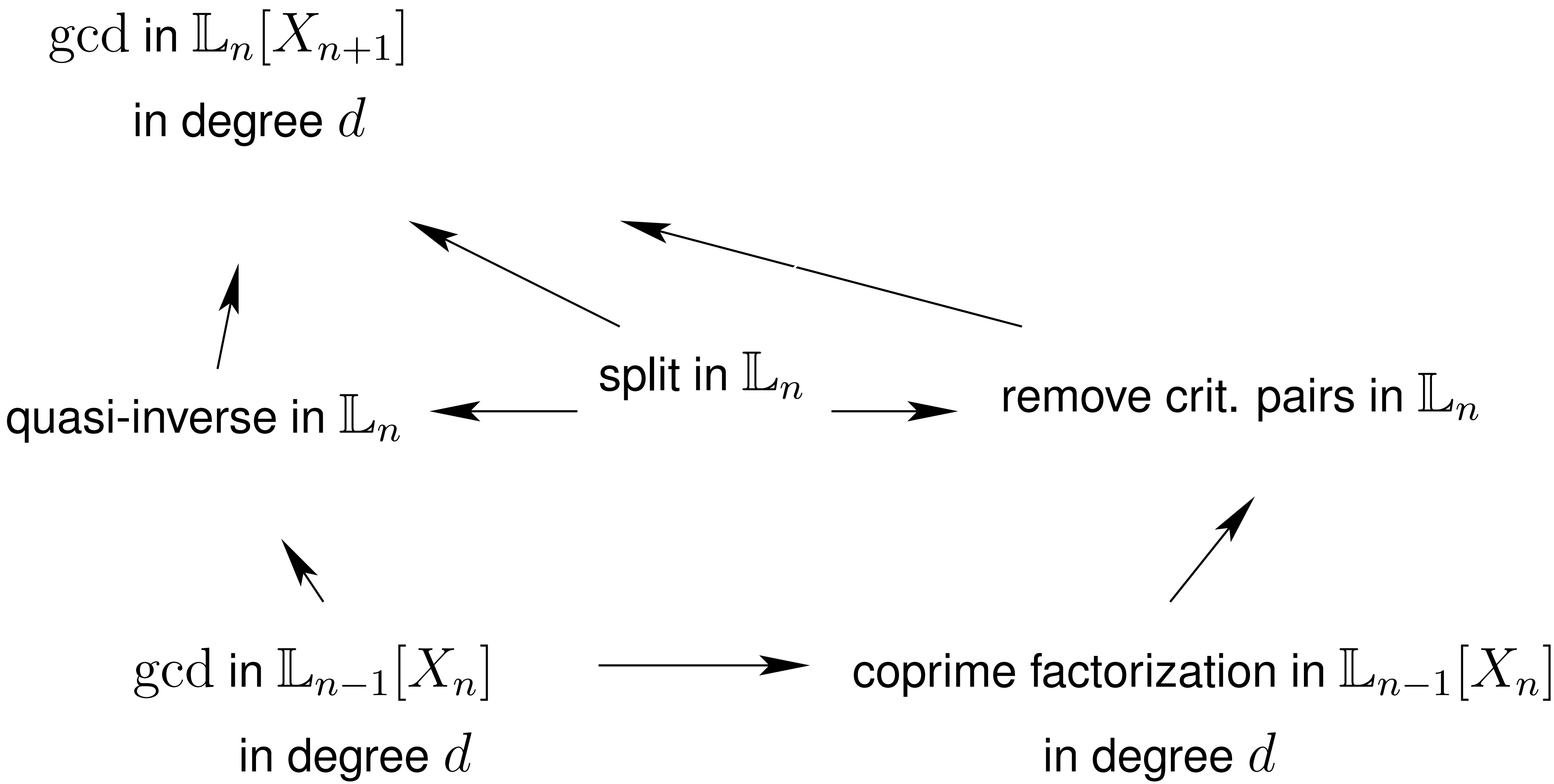
The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

$$\begin{tikzcd}
\text{quasi-inverse in } \mathbb{L}_n & \text{split in } \mathbb{L}_n \arrow[l] \arrow[r] & \text{remove crit. pairs in } \mathbb{L}_n \\
\begin{array}{c}\mathrm{gcd} \text{ in } \mathbb{L}_{n-1}[X_n] \\ \text{in degree } d\end{array} \arrow[u] \arrow[rr] & & \begin{array}{c}\text{coprime factorization in } \mathbb{L}_{n-1}[X_n] \\ \text{in degree } d\end{array} \arrow[u]
\end{tikzcd}$$

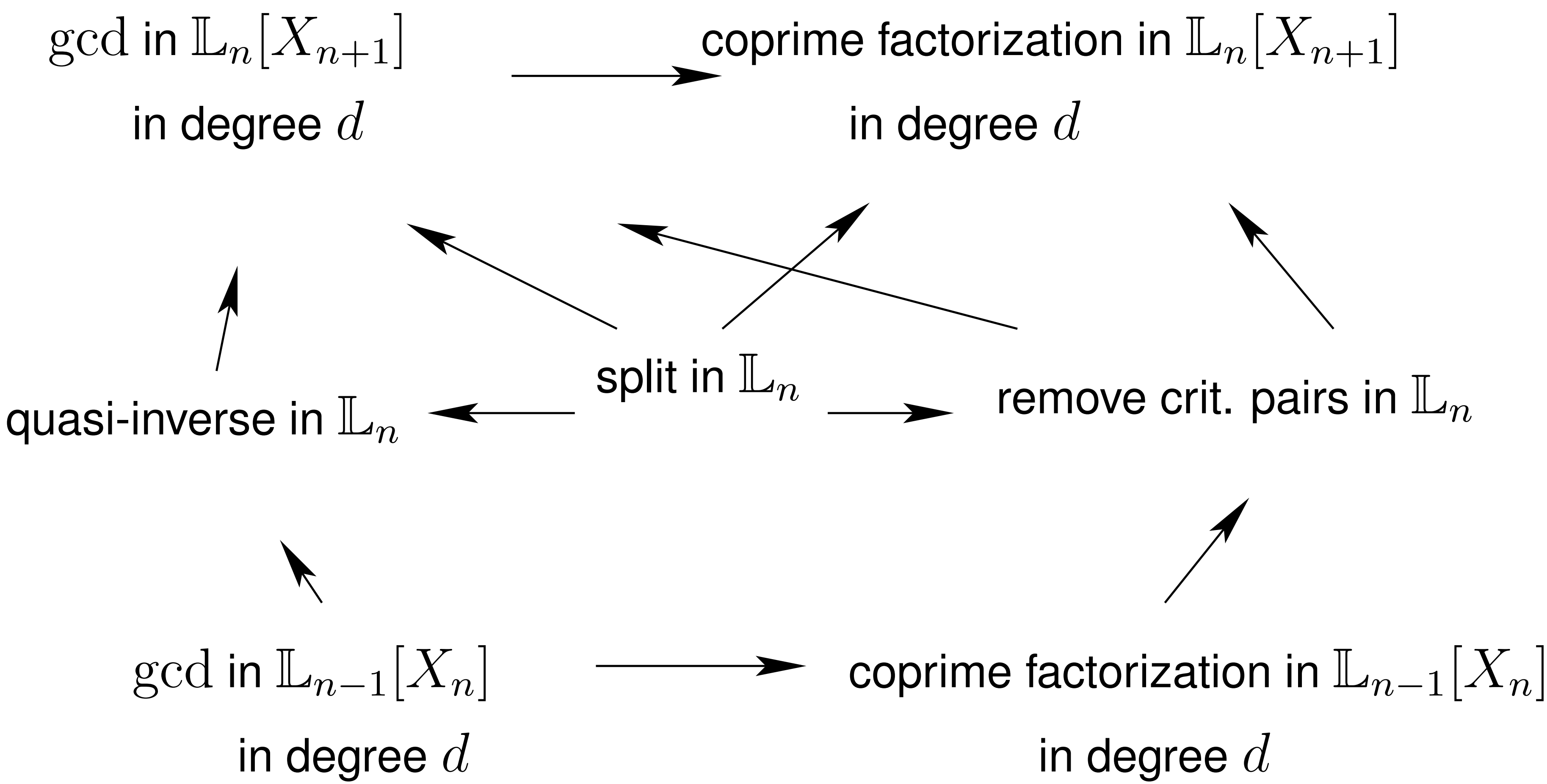
The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$



The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$



Algorithm for Split

- *Input:* A triangular set T,
  a non-critical decomposition $\mathbf{T} = \{T^1, \ldots, T^s\}$ of T,
  a polynomial $f \in k[X_1, \ldots, X_n]$ with $\deg_{X_i}(f) < d_i$.
- *Output:* The family of polynomials $\{f \bmod T^i\,, i = 1 \ldots s\}$.
- *Main idea:* Works recursively on the number of variables, generalizing the well-known *"multi-reduction"* algorithm in one variable.
- *Complexity:* $\mathsf{C}^n \prod_{i=1}^n \mathsf{M}(d_i) \log(d_i)$.
- *References:* Borodin & Moenck

Quasi-inverse

Notation: $\mathbb{K}(T)$ will denote the ring $k[X_1, \ldots, X_n]/(T)$

- *Input:* T a triangular set, $f \in k[X_1, \ldots, X_n]$ with $\deg_{X_i}(f) < d_i$.
- *Output:* A non-critical decomposition $\mathbf{T} = \{T^1, \ldots, T^s\}$, a family of elements $h_i \in \mathbb{K}(T^i)$, s.t.:
  - $f \bmod T^i \equiv 0 \Rightarrow h_i = 0$, and
  - $(f \bmod T^i) \,.\, h_i = 1$ in $\mathbb{K}(T^i)$ else.
- *Main idea:* Use the $\gcd$, "remove critical pairs" and "split" algorithms modulo triangular sets in $n-1$ variables.
- *Complexity:* CM$(d_n)\log(d_n)$ operations modulo $(T_1, \ldots, T_{n-1})$.

Half-GCD

- *Input:* A triangular set T. Two polynomials $a, b \in \mathbb{K}(T)[Y]$, of degree at most d.
- *Output:* A non-critical decomposition $\mathbf{T} = \{T^1, \ldots, T^s\}$ of T, a family $\{g_i\ ,\ i = 1 \ldots s\}$ of monic polynomials, where $g_i \in \mathbb{K}(T^i)[Y]$, s.t. g_i is a gcd of $a \bmod T^i$ and $b \bmod T^i$
- *Main idea (over a field):* The quotient of two polynomials of high degree depends only on their high degree part $\Rightarrow$ relies on *divide and conquer* and is recursive.
- *... and modulo a triangular set:* At each recursive call, there is an Euclidean Division $\Rightarrow$ quasi-inversion of the leading coefficient $\Rightarrow$ splittings, and “remove the critical pairs”.
- *Complexity:* $\mathsf{CM}(d)\log(d)$ operations modulo T
- *References:* Knuth-Schönhage-Moenck

Coprime factorization

- *Input (over a field):* $A = a_1, \ldots, a_s$ family of squarefree polynomials in $k[y]$. $d := \sum_{i=1}^{n} \deg(a_i)$
- *Output (over a field):* $B = b_1, \ldots, b_t \in k[y]$ s.t.:
  - $\gcd(b_i, b_j) = 1$ for $i \neq j$
  - each a_i is a product of some b_j's.
  - each b_j divides one a_i.

- *Main idea:* Divide the family A in 2 sets, compute a coprime factorization of the both two sets, take all the pairs of $\gcd$ between the two coprime factorizations.
- *Complexity:* CM$(d)\log^3(d)$

Coprime factorization

- *Input (over a field):* $A = a_1, \ldots, a_s$ family of squarefree polynomials in $k[y]$. $d := \sum_{i=1}^{n} \deg(a_i)$
- *Output (over a field):* $B = b_1, \ldots, b_t \in k[y]$ s.t.:
  - $\gcd(b_i, b_j) = 1$ for $i \neq j$
  - each a_i is a product of some b_j's.
  - each b_j divides one a_i.
- *Modulo a triangular set:* The input family A is in $\mathbb{K}(T)[y]$. The algorithm uses $\gcd$ computations, hence splittings appear. The output is a non-critical decomposition of T, where the three conditions above make sense.
- *Main idea:* Divide the family A in 2 sets, compute a coprime factorization of the both two sets, take all the pairs of $\gcd$ between the two coprime factorizations.
- *Complexity:* CM$(d) \log^3(d)$ operations modulo T.

Remove critical pairs

- *Input:* A decomposition $\mathbf{T} = \{T^1, \ldots, T^e\}$ of a triangular set T.
- *Output:* Family of decompositions $\mathbf{T}^1, \ldots, \mathbf{T}^e$ s.t.:
  - $\mathbf{T}^i$ is a decomposition of T^i
  - the total family $\mathbf{T}^1 \cup \ldots \cup \mathbf{T}^e$ has no critical pairs.
- *Main idea:* coprime factorization of course…but it is tricky

The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

gcd in $\mathbb{L}_n[X_n]$ in degree d

coprime factorization in $\mathbb{L}_n[X_{n+1}]$ in degree d

quasi-inverse in $\mathbb{L}_n$

split in $\mathbb{L}_n$

$\mathsf{C}^n \prod_{i=1}^{n} \mathsf{M}(d_i) \log(d_i)$

remove crit. pairs in $\mathbb{L}_n$

gcd in $\mathbb{L}_{n-1}[X_n]$ in degree d

$\mathsf{C}^n \mathsf{M}(d) \log(d) \prod_{i=1}^{n-1} \mathsf{M}(d_i) \log^3(d_i)$

coprime factorization in $\mathbb{L}_{n-1}[X_n]$ in degree d

$\mathsf{C}^n \mathsf{M}(d) \log^3(d) \prod_{i=1}^{n-1} \mathsf{M}(d_i) \log^3(d_i))$

The inductive process

$$\boxed{\mathbb{L}_i := k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

$\gcd$ in $\mathbb{L}_n[X_n]$
in degree d

coprime factorization in $\mathbb{L}_n[X_{n+1}]$
in degree d

quasi-inverse in $\mathbb{L}_n$
$\mathsf{C}^n \prod_{i=1}^{n} \mathsf{M}(d_i) \log^3(d_i)$

split in $\mathbb{L}_n$
$\mathsf{C}^n \prod_{i=1}^{n} \mathsf{M}(d_i) \log(d_i)$

remove crit. pairs in $\mathbb{L}_n$
$\mathsf{C}^n \prod_{i=1}^{n} \mathsf{M}(d_i) \log^3(d_i)$

$\gcd$ in $\mathbb{L}_{n-1}[X_n]$
in degree d
$\mathsf{C}^n \mathsf{M}(d) \log(d) \prod_{i=1}^{n-1} \mathsf{M}(d_i) \log^3(d_i)$

coprime factorization in $\mathbb{L}_{n-1}[X_n]$
in degree d
$\mathsf{C}^n \mathsf{M}(d) \log^3(d) \prod_{i=1}^{n-1} \mathsf{M}(d_i) \log^3(d_i))$

The inductive process

$$\boxed{\mathbb{L}_i \; := \; k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

gcd in $\mathbb{L}_n[X_n]$

in degree d

$\mathsf{C}^{n+1}\mathsf{M}(d)\log(d)\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

coprime factorization in $\mathbb{L}_n[X_{n+1}]$

in degree d

quasi-inverse in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

split in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log(d_i)$

remove crit. pairs in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

gcd in $\mathbb{L}_{n-1}[X_n]$

in degree d

$\mathsf{C}^{n}\mathsf{M}(d)\log(d)\prod_{i=1}^{n-1}\mathsf{M}(d_i)\log^3(d_i)$

coprime factorization in $\mathbb{L}_{n-1}[X_n]$

in degree d

$\mathsf{C}^{n}\mathsf{M}(d)\log^3(d)\prod_{i=1}^{n-1}\mathsf{M}(d_i)\log^3(d_i))$

The inductive process

$$\boxed{\mathbb{L}_i \ := \ k[X_1, \ldots, X_i]/(T_1, \ldots, T_i)}$$

gcd in $\mathbb{L}_n[X_n]$ in degree d

$\mathsf{C}^{n+1}\mathsf{M}(d)\log(d)\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

coprime factorization in $\mathbb{L}_n[X_{n+1}]$ in degree d

$\mathsf{C}^{n+1}\mathsf{M}(d)\log^3(d)\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i))$

quasi-inverse in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

split in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log(d_i)$

remove crit. pairs in $\mathbb{L}_n$

$\mathsf{C}^{n}\prod_{i=1}^{n}\mathsf{M}(d_i)\log^3(d_i)$

gcd in $\mathbb{L}_{n-1}[X_n]$ in degree d

$\mathsf{C}^{n}\mathsf{M}(d)\log(d)\prod_{i=1}^{n-1}\mathsf{M}(d_i)\log^3(d_i)$

coprime factorization in $\mathbb{L}_{n-1}[X_n]$ in degree d

$\mathsf{C}^{n}\mathsf{M}(d)\log^3(d)\prod_{i=1}^{n-1}\mathsf{M}(d_i)\log^3(d_i))$

Conclusion

The next goal is to obtain complexity statements for more general dynamic evaluation questions:

- If $\mathbf{A}$ is an "algorithm" that works over a field k in time $T(\mathbf{A})$,
- then one can deduce an "algorithm" that works over the product of fields $k[X_1, \ldots, X_n]/(T_1, \ldots, T_n)$ in time

$$\mathsf{C}^n (\mathsf{M}(d_1) \log^3(d_1) \cdots \mathsf{M}(d_n) \log^3(d_n))\, T(\mathbf{A}).$$

The algorithm for $\gcd$, coprime factorization rely on such results, but the proofs are still ad-hoc.