Closure of algebraic classes under factoring

Nitin Saxena
CSE @ IIT Kanpur
[*Based on many works* / *Thanks to the artists*]

Institut Henri Poincaré @ Paris
Septembre 2023
The problem: Factoring polynomials — the base case

❖ **Question (factor)**: Given \( f \in \mathbb{F}[x] \), find a **nontrivial** factor \( g \)?
   ➢ Depends critically on \( \mathbb{F} \).

❖ \[ \text{[Cantor, Zassenhaus’81]} \] Given \( f \in \mathbb{F}_q[x] \), factor it in **randomized** poly-time.
   ➢ Clever use of residuosity/ Euclid.

❖ \[ \text{[Lenstra, Lenstra, Lovasz’82]} \] Given \( f \in \mathbb{Q}[x] \), factor in poly-time.
   ➢ Lattice basis reduction.

❖ \[ \text{[Cantor, Gordon’00]} \] Given \( f \in \mathbb{Q}_p[x] \), factor in **randomized** poly-time.
   ➢ Newton polytope, p-adic analysis.

\[ x^2 - 2 \in \mathbb{Q}[x] \text{ is irreducible, while } x^2 - 2 \equiv (x - 3)(x - 4) \mod 7 \]

\[ X^{(q-1)/2} - 1 \equiv \prod_{\text{square } a \in \mathbb{F}_q^*} (X - a) \]

\[ \sqrt{2} = 3 + 1 \times 7 + 2 \times 7^2 + 6 \times 7^3 + \ldots \]
**The Model: Algebraic Circuits**

- Valiant (1977) formalized computation via **algebraic circuits**.
  - Giving birth to his $\text{VP} \neq \text{VNP}$ question.
  - Or, algebraic **hardness**!

- Algebraic circuit has constants, variables, **size**, depth.
  - Ignores the size of constants
Factoring Multivariates

❖ Qn. (class): Given $f \in \mathbb{F}[x] := \mathbb{F}[x_1, ..., x_n]$ in class $\mathcal{C}$, find nontrivial factor $g$ in $\mathcal{C}$?
   ➢ Is there an efficient algorithm?

❖ Class $\mathcal{C}$ has to be strong enough to afford factoring techniques.

❖ Circuit of size-$s$ can have $\exp(s)$ degree.
   ➢ Its high-degree factors can be hard.
   ➢ We’ll choose our closure questions carefully!

\[
\left( \sum_{i \in [n]} x_i^p \right) \mod p \text{ has sparsity } n, \text{ while its factor } \\
\left( \sum_{i \in [n]} x_i \right)^{p-1} \text{ has sparsity } \approx n^p.
\]

\[
x^{2^s} - 1 = \prod_{i \in [2^s]} (x - \zeta^i) \text{ has } 2^{2^s} \text{ factors!}
\]
Applications of factoring

❖ [Sudan’97] Decoding Reed-Solomon codes.
  ➢ [Guruswami, Sudan’06] List decoding.

❖ [Kabanets, Impagliazzo’04] Derandomization from hardness.
  ➢ [Kopparty, Saraf, Shpilka’14] Identity testing (PIT) equivalence.

❖ Cryptography.
  ➢ Cryptanalysis,
  ➢ Constructing fields; factoring integers.

❖ Computer Algebra.
  ➢ System solvers; Gröbner bases; Numerical methods.
  ➢ Cornerstone problem!
BIG IDEAS
(POLY-DEGREE)
**Efficiently Factoring VP circuits**

- [Kaltofen’86] Any factor $g$ of size-$s$ circuit $f$ satisfies:
  \[ \text{size}_{\text{ckt}}(g) \leq \text{poly}(s, \text{deg}(f)). \]
  - [Kaltofen, Trager’91] Blackbox for $g$ can be found efficiently.

- The class VP contains polynomial family $f_n(x_n)$ of poly(n)-size and poly(n)-degree.
  - [Kaltofen’86] VP is closed under factoring.
  - **Corollary:** Any nonzero multiple of hard polynomial $(g)$ is hard!

- **Tools:** Hensel lifting and division.

- **Preprocessing (monic in $x_1$):** Write $f(y, x_1, x_2, ..., x_n) = gh$, where
  - $g, h \text{ mod } y$ are univariate in $x_1$ and are coprime.
  - Eg. map $x_1$ to $(b_1 x_1 + a_1)$; $x_2$ to $yx_2 + (b_2 x_1 + a_2)$; ...; $x_n$ to $yx_n + (b_n x_1 + a_n)$. 
Efficiently Factoring VP circuits — Hensel lifts

Given: size-\(s\) degree-\(d\) circuit \(f(y, x_1, x_2, \ldots, x_n)\) as before. Find \(g, h\).

Hensel lift (1st): \(f(0, x_1, x_2, \ldots, x_n) =: g_1 h_1 \mod y\).

➢ Use univariate factoring over \(\mathbb{F}\).

Hensel lift (2nd): \(f(y, x_1, x_2, \ldots, x_n) =: g_2 h_2 \mod y^2\).

➢ Extract coef(y) in circuit \(f\). Use perturbation formula on \(g_1, h_1\).

Hensel lift (\(k\)-th): \(f(y, x_1, x_2, \ldots, x_n) =: g_k h_k \mod y^k\).

➢ Extract coef\((y^{k-1})\) in circuit \(f\). Use perturbation formula on \(g_{k-1}, h_{k-1}\).

Go up to \(k := d+1\).

Question: Is \(g_k\) factor of \(f\)?

➢ Lift is messy: \(g_k\) may’ve extra degree in \(y, x_1\).

Perturbation: \(f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \mod y^2, \) where \(e := (f - g_1 \cdot h_1)\) and \(1 =: u_1 \cdot g_1 + v_1 \cdot h_1\).
Efficiently Factoring VP circuits — monic lifts

- Given \((k=d+1)\): \[ f(y, x_1, x_2, \ldots, x_n) = g_k h_k \mod y^k \]

- Keep monic [Clean-up]: Since \(g\) is monic (in \(x_1\)), we can use monic perturbation, at each lift.
  - Divide: Reduce \(ev_1 \mod g_1\), before adding to \(g_1\), to get \(g_2\). [Strassen’73]

- \(g_k, h_k\) are monic (in \(x_1\)).
  - \(\deg_{x_1}(g) = \deg_{x_1}(g_k)\).

- Fact 1: \(g_k\) is circuit of size \(\text{poly}(s,d)\).

- Fact 2: \(g = g_k\) is actual factor of \(f\)!

- Trick Qn: Without the promise of \(g\), what does \(g_k\) signify?

Perturbation: \(f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \mod y^2\), where \(e := (f - g_1 \cdot h_1)\) and \(1 =: u_1 \cdot g_1 + v_1 \cdot h_1\).
Efficient Factoring in VBP

- [Sinhababu,Thierauf’21] Any factor $g$ of size-$s$ algebraic branching program (ABP) $f$ satisfies: $\text{size}_{\text{abp}}(g) \leq \text{poly}(s)$.
  - ABP is a matrix–product expression, or equivalently, the determinant model.

- The class VBP contains polynomial family $f_n(x_n)$ of poly(n)-size ABP.
  - [Sinhababu,Thierauf’21] VBP is closed under factoring.
  - Corollary: Any nonzero multiple of ABP-hard $g$ is ABP-hard!

- **Tools:** Fast Hensel-lifting and Linear-system solving.

- **Preprocessing (monic in $x_1$):** Write $f(y,x_1,x_2,\ldots,x_n)=gh$, where
  - $g,h$ mod $y$ are univariate in $x_1$ and are coprime.
  - Eg. map $x_1$ to $(b_1x_1+a_1)$; $x_2$ to $yx_2+(b_2x_1+a_2)$; $\ldots$; $x_n$ to $yx_n+(b_nx_1+a_n)$. 

Efficient Factoring in VBP — Fast Hensel lifts

- **Given:** size-s degree<s ABP \( f(y, x_1, x_2, \ldots, x_n) \) as before. Find \( g, h \).

- **Hensel lift (1st):** \( f(0, x_1, x_2, \ldots, x_n) =: g_1 h_1 \mod y \).
  - Use univariate factoring over \( \mathbb{F} \).

- **Hensel lift (2nd):** \( f(y, x_1, x_2, \ldots, x_n) =: g_2 h_2 \mod y^2 \).
  - Extract coef\((y)\) in circuit \( f \). Use perturbation formula on \( g_1 \) and \( h_1 \).

- **Hensel lift \((\log_2(D)-th)\):** \( f(y, x_1, x_2, \ldots, x_n) =: g_D h_D \mod y^D \).
  - Extract coef\((y^{D-1})\) in circuit \( f \). Use perturbation formula on \( g_{D/2}, h_{D/2} \).

- **Go up to** \( D := (2s^2 + 1) \). [ABP-size grows 4-times per lift.]

- **Question:** Is \( g_D \) factor of \( f \)?
  - **Lift is messy:** Non-monic \( g_D \) may’ve extra degree in \( y, x_1 \).

  **Perturbation:** \( f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \mod y^2 \), where \( e := (f - g_1 \cdot h_1) \) and \( 1 =: u_1 \cdot g_1 + v_1 \cdot h_1 \).
Efficient Factoring in VBP – Linear-system

- Given \((D=2s^2+1)\): \(f(y,x_1,x_2,...,x_n) =: g_D h_D \mod y^D\).
- Solve linear-system [Clean-up]: \(g' = g_D \ell \mod y^D\), where
  - \(\deg_{x_1}(g') \leq \deg_{x_1}(g), \deg_y(g') \leq \deg_y(g)\),
  - \(\deg_{x_1}(\ell) \leq \deg_{x_1}(h_D), \deg_y(\ell) < D\).
  - It’s ABP friendly.

- **Fact 3**: \(g'\) is ABP of size \(\text{poly}(s)\).
  - So is its leading-coeff (wrt \(x_1\)), say \(c = c(y,x_2,...,x_n)\).
- **Fact 4**: \(g = g'/c\).
- **Eliminating** division (merely once!), finishes the proof.

\[\text{QED}\]
“Efficient” Factoring in VNP — Witness/formula trick

Proof similar to factoring in VP. Except,

\[ f(y,x) = \sum_{w \in \{0,1\}^m} V(w,y,x), \] where \( V \) is verifier-circuit on witness \( w \).

In VP proof: \( f(y,x) =: g_k h_k \mod y^k \), gives circuit \( C(f) \) for \( g_k = g \).

[Valiant’82] There is small verifier-formula \( F \): \( C(f) =: \sum_{w \in \{0,1\}^m} F(w',f) \).

Composition gives: \( g = \sum_{(w,w') \in \{0,1\}^{m+m'}} F(w',V(w,y,x)) \), thus proving—

Fact 5: \( g \) in VNP, with size-parameter poly(s,d).

[Chou,Kumar,Solomon’18] VNP is closed under factoring.

QED

Overlooked: need large field; characteristic? OK for coprime \( g,h \).
Factoring in shallow depths? — Introducing Newton

- [Oliveira’15] Let $f$ has individual-degree $r$ and size-$s$. In just depth+4, any factor $g$ of $f$ has: $\text{size}(g) \leq \text{poly}(s^r)$.
  - Constant-ind.degree, constant-depth model is closed under factoring.

Tools: Newton-iteration.

Preprocessing (monic in $x_1$): Write $f(y, x_1, x_2, \ldots, x_n) = (x_1 - \varphi(yx_2, \ldots, yx_n)) \cdot h$, where
  - $\varphi$ is power-series in $\mathbb{F}[[yx_2, \ldots, yx_n]]$ and $h(y=0, x_1=\varphi) \neq 0$ [coprime].
  - Eg. map $x_1$ to $(b_1x_1+a_1)$; $x_2$ to $yx_2+(b_2x_1+a_2)$; $\ldots$; $x_n$ to $yx_n+(b_nx_1+a_n)$.

Newton-iteration specifies the simple-root $\varphi$ of $f$.

Requires: one derivation, many compositions.

Newton-iteration: Approximant up to degree $m$ of $\varphi$ is $\varphi_{m+1} := \varphi_m - f(\varphi_m)/\partial_{x_1} f(\varphi_m(0))$. 
Factoring in shallow depths? — Introducing Newton

- Newton-iteration: The coefficients of $f$ are $C_0(y, x_2, ..., x_n), ..., C_r(y, x_2, ..., x_n)$.
- Inductively, $\phi_{m+1}$ can be written as degree-$m$ function in these.
- **Fact 6**: $\phi_{m+1}$ is depth-2 circuit of size $m^r$, in $C_i$'s.
- Once we’ve roots, we’ve factors!
- **Fact 7**: $g$ requires depth-4 circuit, of size $\text{poly}(s^r)$, on top of $f$.

QED

Newton-iteration: Approximant up to degree $m$ of $\phi$ is $\phi_{m+1} := \phi_m - f(\phi_m)/\partial_{x_1} f(\phi_m(0))$. 
Big ideas
(Exp-degree)
Factoring exponential degree circuits? — More Newton

[Dutta, S., Sinhababu’18] Any factor $g$ of size-$s$ circuit $f$ satisfies:

$$\text{size}_{\text{ckt}}(g) \leq \text{poly}(s, \deg(\text{rad}(f)))$$

- Radical $\text{rad}(f)$ is the squarefree part. May have $\deg > 2^s$!

**Tools:** Modified Newton-iteration.

**Preprocessing (monic in $x_1$):** Write $f(y, x_1, x_2, ..., x_n) = \prod_{i \in [k]} (x_1 - \phi_i(yx_2, ..., yx_n))^{e_i}$, where

- $\phi_i$ is power-series in $\mathbb{F}[[yx_2, ..., yx_n]]$ and $\phi_i(y=0)$ are distinct [coprime].
- Eg. map $x_1$ to $(b_1x_1 + a_1)$; $x_2$ to $yx_2 + (b_2x_1 + a_2)$; … ; $x_n$ to $yx_n + (b_nx_1 + a_n)$.

- Roots are very far from simple.
  - Can’t run Newton iteration. [Division by 0!]

**Newton-iteration:** Approximant up to degree $m$ of $\varphi_i$ is $\varphi_{i,m+1} := \varphi_{i,m} - f(\varphi_{i,m})/\partial_{x_1}f(\varphi_{i,m}(0))$. 

}$\square$

$\square$
Factoring exponential degree circuits? — More Newton

❖ Consider $F := f + yz \cdot \partial_{x_1} f$, where $z$ is new. Then,

❖ $F =: \prod_{i \in [k]} (x_1 - \psi_i(yx_2, ..., yx_n))^{e_i^{-1}} \cdot (\text{rad}(f) + yz \cdot Q) =: u \cdot v$, where
  ➢ $u, v$ are coprime, monic and $k = \deg_{x_1}(v) = \deg_{x_1}(\text{rad}(f)) > \deg_{x_1}(Q)$.

❖ Newton-iteration finds (distinct) simple root $\psi_i$ of $v$ in $\mathbb{F}[[yz, yx_2, ..., yx_n]]$.

❖ Setting $z=0$, we get circuit for rad($f$).
  ➢ of size $\text{poly}(s, k)$.
  ➢ Though $F$ is very-high deg, we only use its $\deg(\text{rad}(f))$ part. 
  \[ \text{QED} \]

Newton-iteration: Approximant up to degree $m$ of $\psi_i$ is $\psi_{i,m+1} := \psi_{i,m} - F(\psi_{i,m})/\partial_{x_1} F(\psi_{i,m}(0))$. 
Factoring approximatively — introducing $\varepsilon$

- [Bürgisser’01] Any factor $g$ of size-\textit{s} circuit $f$ satisfies:
  \[
  \text{size}_{\text{approx}}(g) \leq \text{poly}(s, \deg(g)).
  \]
  - Works over $\mathbb{F}(\varepsilon)$, with $\varepsilon \to 0$, where \textit{precision is exponential}!

- **Tools:** Perturb by $\varepsilon$, and Newton-iteration over $\mathbb{F}(\varepsilon)$.

- **Preprocessing (monic in $x_1$):** Write $f(y, x_1, x_2, \ldots, x_n) = (x_1 - \varphi(yx_2, \ldots, yx_n))^{\varepsilon} \cdot h$, where
  - $\varphi$ is power-series in $\mathbb{F}[[yx_2, \ldots, yx_n]]$ and $h(y=0, x_1=\varphi) \neq 0$ [coprime].
  - Eg. map $x_1$ to $(b_1x_1+a_1)$; $x_2$ to $yx_2+(b_2x_1+a_2)$; $\ldots$; $x_n$ to $yx_n+(b_nx_1+a_n)$.

- Root $\varphi$ is very far from simple, as $\varepsilon$ is exponential.
  - Can’t run Newton iteration. [Division by 0 !]

**Newton-iteration:** Approximant up to degree $m$ of $\varphi$ is $\varphi_{m+1} := \varphi_m - f(\varphi_m)/\partial_{x_1} f(\varphi_m(0))$. 


Factoring approximatively — Introducing \( \varepsilon \)

- Consider \( F(y, x_1, x_2, \ldots, x_n) := f(y, x_1 + \varepsilon, x_2, \ldots, x_n) - f(0, \varphi(y=0) + \varepsilon, x_2, \ldots, x_n) \). Then,
  - \( F(y=0, x_1=\varphi) = 0 \), \( F_{\varepsilon=0} = f \),
  - \( \partial_{x_1} F(y=0, x_1=\varphi) = \varepsilon^{-1} \cdot (e \cdot h(y=0, x_1=\varphi) + \varepsilon \cdot \partial_{x_1} h(y=0, x_1=\varphi)) \neq 0 \).

- **Fact 8**: \( \varphi \) is simple root of \( F(y=0) \).

- **Initializing**: \( x_1 \leftarrow \varphi(y=0) \), Newton-iteration finds simple root \( \psi \) of \( F \), in \( \mathbb{F}(\varepsilon)[[y x_2, \ldots, y x_n]] \).

- **Fact 9**: \( \psi_{\varepsilon=0} \rightarrow \varphi \) is required root of \( f \).
  - No way known to find \( \varphi \) exactly.

QED

Newton-iteration: Approximant up to degree \( m \) of \( \psi \) is \( \psi_{m+1} := \psi_m - F(\psi_m)/\partial_{x_1} F(\psi_m(0)) \).
Open questions (tricky models)
Factoring ‘weak’ models?

- **Question (formula):** Factor formulas?
  - Is VF closed under factoring?
  - Only known for constant-individual-degree. [Oliveira’15]

- Could sparse-polynomials be factored? **No.**
  - Depth-2 not closed under factoring.

- **Question (depth-2):** Factor constant-individual-degree depth-2?
  - Partial results known. [Bhargava,Saraf,Volkovich’18] [Bisht,S.’22]
Roots in general?

- Given size-\( s \) circuit \( f \), apply the random map to see roots:
  - Write \( f(y, x_1, x_2, \ldots, x_n) = (x_1 - \varphi(x_2, \ldots, x_n))^e \cdot h \), where
    - \( \varphi \) is power-series in \( \mathbb{F}[[x_2, \ldots, x_n]] \).

- Question (any-root): size(\( \varphi_m \)) \( \leq \) poly(\( s, m \))?
  - Implies [Bürgisser’01]’s factor conjecture.
  - Is \( \varphi_m \) in VNP?

- Characteristic issues: Say, char(\( \mathbb{F} \)) = \( p \) and \( p \mid e \).
- VP/VBP/VNP/approximative results for bad multiplicity?
- Question (inverse-Frobenius): Given \( g^p \), find \( g \)?
- Question (Galois-ring): Factor mod \( p^2, p^3, \ldots, p^k, \ldots, p^\infty \)?

www.cse.iitk.ac.in/users/nitin/