Closure of algebraic classes under factoring

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

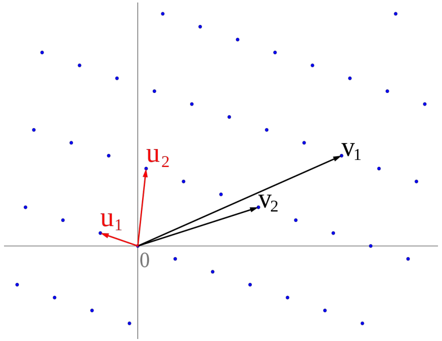
8th WACT @ RUB
Mar-Apr'25

THE PROBLEM: FACTORING POLYNOMIALS – THE BASE CASE

- Question (factor): Given f ϵ $\mathbb{F}[x]$, find a *nontrivial* factor g?
  - ➢ Depends critically on $\mathbb{F}$.
- [Cantor,Zassenhaus'81] Given f ϵ $\mathbb{F}_q[x]$, factor it in *randomized* poly-time.
  - ➢ Clever use of residuosity/ Euclid.
- [Lenstra,Lenstra,Lovasz'82] Given f ϵ $\mathbb{Q}[x]$, factor in poly-time.
  - ➢ Lattice basis reduction.
- [Cantor,Gordon'00] Given f ϵ $\mathbb{Q}_p[x]$, factor in *randomized* poly-time.
  - ➢ Newton polytope, p-adic analysis.

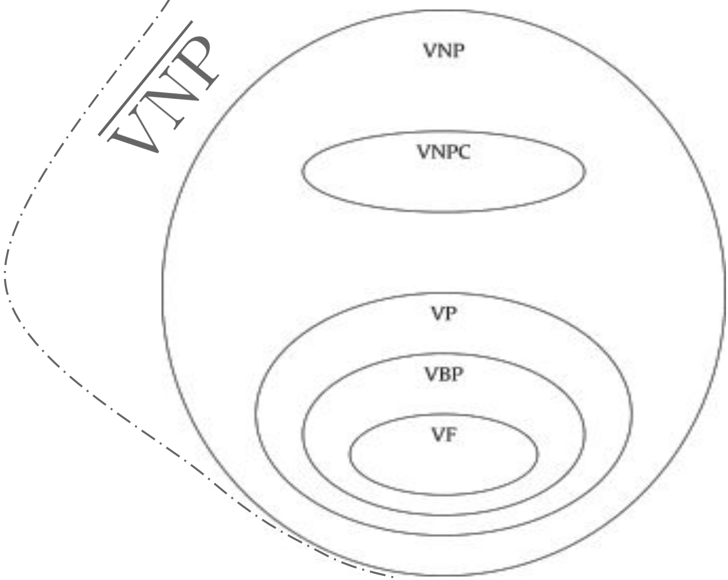
$x^2-2 \in \mathbb{Q}[x]$ is irreducible, while $x^2-2 \equiv (x-3)(x-4) \bmod 7$

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

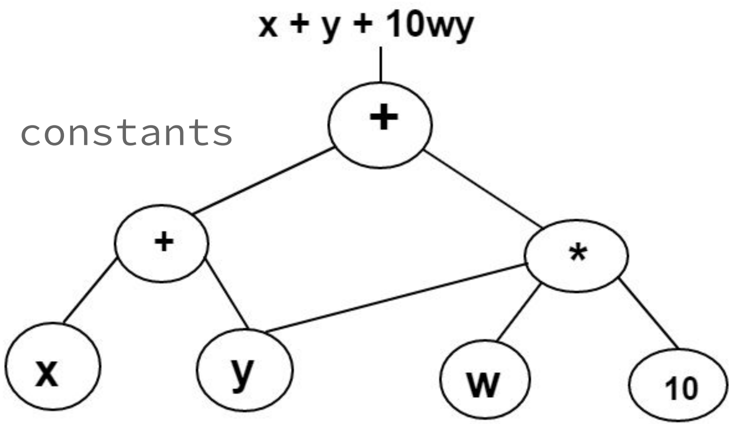


$$\sqrt{2} = 3 + 1\times 7 + 2\times 7^2 + 6\times 7^3 + \cdots$$

The model: Algebraic circuits



- ❖ Valiant (1977) formalized computation via **algebraic circuits**.
  - ➢ Giving birth to his VP ≠ VNP question.
  - ➢ Or, algebraic **hardness**!
- ❖ Algebraic circuit has constants, variables, **size**, depth.
  - ➢ Ignores the size of constants



Leslie Valiant (1949-)

Factoring Multivariates

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

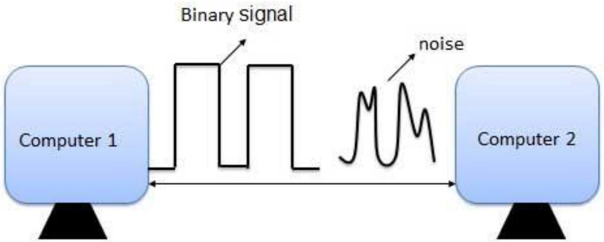
$(\sum_{i\in[n]} x_i^p) \bmod p$ has sparsity n, while its factor $(\sum_{i\in[n]} x_i)^{p-1}$ has sparsity $\approx n^p$.

- 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!

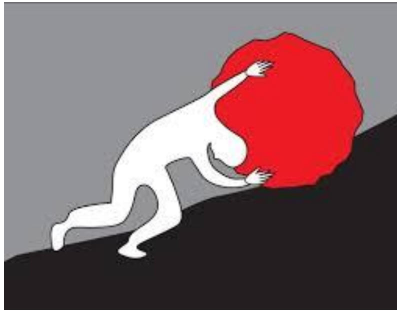
$x^{2^s} - 1 = \prod_{i\in[2^s]}(x - \zeta^i)$ has 2^{2^s} factors!

Applications of factoring



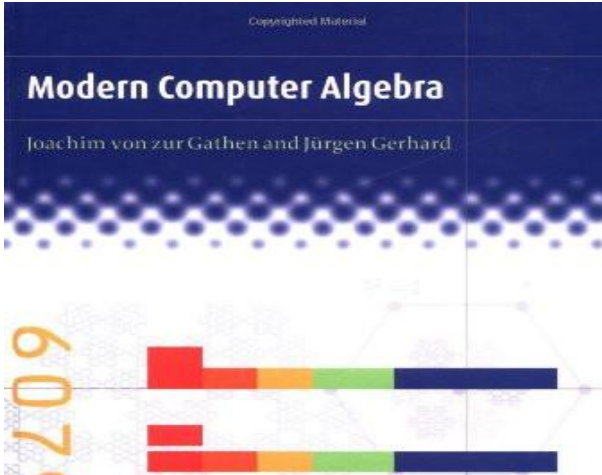
- [Sudan'97] Decoding Reed-Solomon codes.
  - [Guruswami,Sudan'06] List decoding.

- [Kabanets,Impagliazzo'04] Derandomization from hardness.
  - [Kopparty,Saraf,Shpilka'14] Identity testing (PIT) equivalence.
  - [Mulmuley'13] Geometric Complexity Theory.
  - [Forbes,Shpilka,Tzameret,Wigderson'16] Proof Complexity.



- 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:**
  $$\mathrm{size}_{ckt}(g) \leq \mathrm{poly}(s, \deg(f))\ .$$
  - [Kaltofen,Trager'91] *Blackbox* for g can be found efficiently.
- The class VP contains polynomial family $f_n(\mathbf{x}_n)$ 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 $f(y, x_1, x_2, \ldots, x_n) = gh$, where
  - $g, h \bmod y$ are univariate in x_1 and are coprime.
  - Eg. map x_1 to $(b_1x_1+a_1)$; x_2 to $yx_2+(b_2x_1+a_2)$; … ; x_n to $yx_n+(b_nx_1+a_n)$.

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):** $f(0, x_1, x_2, \ldots, x_n) =: g_1 h_1 \bmod y$.
  - ➢ Use *univariate* factoring over $\mathbb{F}$.
- **Hensel lift (2nd):** $f(y, x_1, x_2, \ldots, x_n) =: g_2 h_2 \bmod y^2$.
  - ➢ Extract coef(y) in circuit f. Use **perturbation** formula on g_1 and h_1 .
- **Hensel lift (k-th):** $f(y, x_1, x_2, \ldots, x_n) =: g_k h_k \bmod y^k$.
  - ➢ Extract coef(y^{k-1}) in circuit f. Use perturbation formula on g_{k-1} and 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 .

(error-feedback) Perturbation : $f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \bmod y^2$, 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): $f(y, x_1, x_2,\ldots, x_n) =: g_k h_k \bmod y^k$.
- Keep monic [Clean-up]: Since g is monic (in x_1), we can use monic perturbation, at each lift.
  - Divide: Reduce $ev_1 \bmod g_1$, 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 poly(s,d) .
- Fact 2: $g = g_k$ is *actual* factor of f !

QED

Trick Qn : Without the promise of g , what does g_k signify?

(error-feedback) Perturbation : $f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \bmod y^2$, 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:** $\mathrm{size}_{abp}(g) \le \mathrm{poly}(s)$.
  - ABP is a matrix-product expression, or equivalently, the determinant model.
- The class VBP contains polynomial family $f_n(\mathbf{x}_n)$ 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 $f(y, x_1, x_2, \dots, x_n) = gh$, where
  - $g, h \bmod y$ are univariate in x_1 and are coprime.
  - Eg. map x_1 to $(b_1x_1+a_1)$; x_2 to $yx_2+(b_2x_1+a_2)$; … ; x_n to $yx_n+(b_nx_1+a_n)$.

Efficient Factoring in VBP – Fast Hensel lifts

- **Given:** size-s degree<s ABP $f(y, x_1, x_2, \dots, x_n)$ as before. **Find g, h.**
- **Hensel lift (1st):** $f(0, x_1, x_2, \dots, x_n) =: g_1 h_1 \bmod y$.
  - Use *univariate* factoring over $\mathbb{F}$.
- **Hensel lift (2nd):** $f(y, x_1, x_2, \dots, x_n) =: g_2 h_2 \bmod y^2$.
  - Extract coef(y) in circuit f. Use **perturbation** formula on g_1 and h_1 .
- **Hensel lift ($\log_2(D)$-th):** $f(y, x_1, x_2, \dots, x_n) =: g_D h_D \bmod y^D$.
  - Extract $\mathrm{coef}(y^{D-1})$ 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.

(error-feedback) Perturbation : $f \equiv (g_1 + e \cdot v_1) \cdot (h_1 + e \cdot u_1) \bmod y^2$, 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 ($D=2s^2+1$) : $f(y, x_1, x_2,\dots, x_n) =: g_D\, h_D \bmod y^D$.
- Solve linear-system [Clean-up]: $g' = g_D\,\ell \bmod y^D$, where
  - $\deg_{x_1}(g') \le \deg_{x_1}(g)$, $\deg_y(g') \le \deg_y(g)$,
  - $\deg_{x_1}(\ell) \le \deg_{x_1}(h_D)$, $\deg_y(\ell) < D$.
  - It's ABP friendly.
- <u>Fact 3</u>: g' is ABP of size poly(s) .
  - So is its leading-coeff (wrt x_1), say $c = c(y, x_2,\dots, x_n)$.
- <u>Fact 4</u>: $g = g'/c$.
- *Eliminating* division (merely once!), finishes the proof.

QED

Prior Trick Qn:
Answered now!

"Efficient" Factoring in VNP – Witness/formula trick

- Proof similar to factoring in VP. Except,
  - $f(y,\mathbf{x}) =: \sum_{\mathbf{w} \in \{0,1\}^m} V(\mathbf{w},y,\mathbf{x})$, where V is verifier-circuit on witness **w**.
- In VP proof: $f(y,\mathbf{x}) =: g_k h_k \bmod y^k$, gives *circuit* C(f) for $g_k = g$.
- [Valiant'82] There is small verifier-formula F: $C(f) =: \sum_{\mathbf{w}' \in \{0,1\}^{m'}} F(\mathbf{w}',f)$.
- Composition gives: $g = \sum_{(\mathbf{w},\mathbf{w}') \in \{0,1\}^{(m+m')}} F(\mathbf{w}', V(\mathbf{w},y,\mathbf{x}))$, thus proving–

- Fact 5: g in VNP , with size-parameter poly(s,d).
- [Chou,Kumar,Solomon'18] **VNP is closed under factoring.**
  - [Bhargav,Dwivedi,S.'24] made it general.

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:** $\mathrm{size}(g) \leq \mathrm{poly}(s^{r})$.
  - Constant-ind.degree, constant-depth model is closed under factoring.

- Tools: Newton-iteration.

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

- Newton-iteration specifies the *simple*-root φ of f .

- <u>Requires</u> : one derivation, many compositions.

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

Factoring in shallow depths? – introducing Newton

- Newton-iteration: The coefficients of f are $C_0(y, x_2,\dots, x_n)$,..., $C_r(y, x_2,\dots, x_n)$.
- *Inductively*, φ_{m+1} can be written as degree-m function in these.
- Fact 6: φ_{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 poly(s^r), on top of f.

QED

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

Big ideas (Exp-degree)

Factoring exponential degree circuits? – More Newton

- [Dutta,S.,Sinhababu'18] **Any factor g of size-s circuit f satisfies:** $\mathrm{size}_{ckt}(g) \le \mathrm{poly}(s, \deg(\mathrm{rad}(f)))$.
  - Radical $\mathrm{rad}(f)$ is the squarefree part. May have $\deg > 2^s$!
- Tools: Modified Newton-iteration.
- Preprocessing (monic in x_1): Write $f(y, x_1, x_2, \ldots, x_n) = \prod_{i\in[k]} (x_1 - \varphi_i(yx_2, \ldots, yx_n))^{e\_i}$, where
  - φ_i is power-series in $\mathbb{F}[[yx_2, \ldots, yx_n]]$ and $\varphi_i(y=0)$ 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 φ_i is $\varphi_{i,m+1} := \varphi_{i,m} - f(\varphi_{i,m})/\partial_{x_1} f(\varphi_{i,m}(\mathbf{0}))$.

Factoring exponential degree circuits? – More Newton

- Consider $F := f + yz\cdot\partial_{x\_1} f$, where z is new. Then,
- $F =: \prod_{i\in[k]} (x_1 - \varphi_i(yx_2 ,\dots, yx_n))^{e\_i-1} \cdot (rad(f) + yz\cdot Q) =: u\cdot v$, where
  - u, v are coprime, monic and $k = deg_{x\_1}(v) = deg_{x\_1}(rad(f)) > deg_{x\_1}(Q)$.
- Newton-iteration finds (distinct) simple root ψ_i of v in $\mathbb{F}[[yz, yx_2 ,\dots, yx_n]]$.
- Setting z=0, we get circuit for rad(f).
  - of size poly(s, k) .
  - Though F is *very-high* deg, we only use its deg(rad(f)) part.

QED

Newton-iteration: Approximant up to degree m of ψ_i is $\psi_{i,m+1} := \psi_{i,m} - F(\psi_{i,m})/\partial_{x_1} F(\psi_{i,m}(\mathbf{0}))$.

Factoring approximatively – introducing ε

- [Bürgisser'01] **Any factor g of size-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 *precision is exponential*!
- Tools: Perturb by ε, and Newton-iteration over $\mathbb{F}(\varepsilon)$.
- Preprocessing (monic in x_1): Write $f(y, x_1, x_2, \ldots, x_n) = (x_1 - \varphi(yx_2, \ldots, yx_n))^e \cdot h$, where
  - φ 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 $\mathbf{yx_2}+(b_2x_1+a_2)$; … ; x_n to $\mathbf{yx_n}+(b_nx_1+a_n)$.
- Root φ is very far from *simple*, as e is exponential.
  - Can't run Newton iteration. [Division by 0 !]

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

Factoring approximatively – introducing ε

- Consider $F(y, x_1, x_2, ..., x_n) := f(y, x_1+\varepsilon, x_2, ..., x_n) - f(0, \varphi(y=0)+\varepsilon, x_2, ..., x_n)$. Then,
  - $F(y=0, x_1=\varphi) = 0$, $F_{\varepsilon=0} = f$,
  - $\partial_{x\_1} F(y=0, x_1=\varphi) = \varepsilon^{e-1} \cdot (e \cdot h(y=0, x_1=\varphi) + \varepsilon \cdot \partial_{x\_1} h(y=0, x_1=\varphi)) \neq 0$.
- Fact 8: φ is *simple* root of $F(y=0)$.
- Initializing: $x_1 \leftarrow \varphi(y=0)$, Newton-iteration finds simple root ψ of F, in $\mathbb{F}(\varepsilon)[[yx_2, ..., yx_n]]$.
- Fact 9: $\psi_{\varepsilon=0} \to \varphi$ is required root of f .
  - No way known to find φ exactly. QED
- [Bhargav,Dwivedi,S.'24] **made it more explicit: "g is in VNP".**

Newton-iteration: Approximant up to degree m of ψ is $\psi_{m+1} := \psi_m - F(\psi_m)/\partial_{x_1} F(\psi_m(\mathbf{0}))$.

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 $f(y, x_1, x_2, \ldots, x_n) = (x_1 - \varphi(yx_2, \ldots, yx_n))^e \cdot h$, where
  - φ is **power-series** in $\mathbb{F}[[yx_2, \ldots, yx_n]]$.
- Question (any-root): $\text{size}(\varphi_m) \leq \text{poly}(s, m)$?
  - Implies [Bürgisser'01]'s factor conjecture.
  - Is φ_m in VNP? [general case is OPEN]
- **Characteristic issues:** Say, $\text{char}(\mathbb{F}) =: p$ and $p | e$.
- VP/VBP/approximative results for **bad multiplicity** ?
- Question (inverse-Frobenius): Given g^p , find g ?
- Question (non-Fields): Factor $\bmod\ p^2, p^3, \ldots, p^k, \ldots, p^\infty$?

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

THANKS!