# Equations over finite algebras

Erhard Aichinger

Institute for Algebra
Johannes Kepler University Linz
Linz, Austria

AAA105, Prague, June 2024

Supported by the Austrian Science Fund (FWF) : P33878

# I. Systems of term equations

# Systems of term equations

Let $\mathbf{A}$ be an algebra.
TermSysSat($\mathbf{A}$) is the following problem:

**Given:**
Terms $s_1(x_1, \ldots, x_n), t_1(x_1, \ldots, x_n), \ldots, s_k(x_1, \ldots, x_n), t_k(x_1, \ldots, x_n)$.

**Asked:**
Is there $\boldsymbol{a} \in A^n$ with $s_1^{\mathbf{A}}(\boldsymbol{a}) = t_1^{\mathbf{A}}(\boldsymbol{a})$, $\ldots$, $s_k^{\mathbf{A}}(\boldsymbol{a}) = t_k^{\mathbf{A}}(\boldsymbol{a})$?

**Remarks:**

- The answer is always **yes** if $\mathbf{A}$ has a one-element subuniverse: groups, lattices.
- Allowing constants yields PolSysSat($\mathbf{A}$), which can be harder.

# Computational complexity of TermSysSat($\mathbf{A}$)

One can solve the equations by solving a constraint satisfaction problem.

**Idea:** (Larose, Zádori 2006)

Instead of solving

$$f(g(x_1, x_2)) = f(x_1),$$

solve

$$(x_1, x_2, y_1) \in g^\circ, (y_1, y_2) \in f^\circ, (x_1, y_2) \in f^\circ, \text{ where}$$

$$g^\circ = \{(a_1, a_2, b) \in A^3 \mid g(a_1, a_2) = b\}$$

is the graph of $g$.

This reduces TermSysSat$(A; f, g)$ to CSP$(A; f^\circ, g^\circ)$.

# Computational complexity of TermSysSat($\mathbf{A}$)

For an algebra $\mathbf{A} = (A; F)$, let $\mathbf{A}^\circ := (A; \{f^\circ \mid f \in F\})$.

As a consequence of the Bulatov-Zhuk-Dichotomy (2017) (in the form of Barto, Krokhin, Willard (2017)), one obtains:

**Theorem** (cf. [Mayr, MFCS 2023]).

(Assume $\mathbf{P} \neq \mathbf{NP}$).
Let $\mathbf{A}$ be a finite algebra. Then TermSysSat($\mathbf{A}$) $\in \mathbf{P} \iff \mathbf{A}^\circ$ has a (not necessarily idempotent) Taylor polymorphism.

Otherwise TermSysSat($\mathbf{A}$) is **NP**-complete.

# Computational complexity of TermSysSat($\mathbf{A}$)

**Question:** Algebraic description when $\mathbf{A}^\circ$ has a (not necessarily idempotent) Taylor polymorphism.

**Definition.** Let $\mathbf{A}$ be a finite algebra.
Core($\mathbf{A}$) is a minimal endomorphic image of $\mathbf{A}$ w.r.t $\subseteq$.
(Defined up to isomorphism)

**Examples.**

- $\mathbf{G}$ group. $\mathrm{Core}(\mathbf{G}) = \{1\}$.
- $\mathbf{G}$ group. $\mathbf{G}^* := (G; *, ^{-1}, (c_g)_{g\in G})$ its expansions with all constants from $G$. Then $\mathrm{Core}(\mathbf{G}^*) = G$.
- $\mathrm{Core}((S_5; \circ, ^{-1}, \underbrace{\mathrm{id}, (1\ 2)}_{\text{nullary}})) = \{\mathrm{id}, (1\ 2)\}$.

# Computational complexity of TermSysSat($\mathbf{A}$)

**Theorem** Larose, Zádori 2006

Let $\mathbf{A}$ be a finite algebra in a congruence modular variety. TFAE:

1. PolSysSat($\mathbf{A}$) = TermSysSat($\mathbf{A}^*$) $\in \mathbf{P}$.
2. $\mathbf{A}$ is abelian.

**Theorem** Mayr 2023

Let $\mathbf{A}$ be a finite algebra in a congruence modular variety. TFAE:

1. TermSysSat($\mathbf{A}$) $\in \mathbf{P}$.
2. Core($\mathbf{A}$) is abelian.

Both results also hold also if $1 \notin \mathrm{typ}(V(\mathbf{A}))$ and $5 \notin \mathrm{typ}(\{\mathbf{A}\})$.

# TermSysSat(**A**) vs. PolSysSat(**A**)

**Theorem** Mayr 2023.

Let **A** be a finite algebra of finite type. The following three problems are reducible to each other in constant time:

1. TermSysSat(**A**).
2. TermSysSat(Core(**A**)).
3. PolSysSat(Core(**A**)).

# The meta-problem for systems of term equations

# The meta-problem for TermSysSat

**Meta-problem** for TermSysSat (Assume $\mathbf{P} \neq \mathbf{NP}$)

**Given:** $\mathbf{A} = (A; f_1, \ldots, f_k)$

**Asked:** Is TermSysSat$(\mathbf{A}) \in \mathbf{P}$?

# The meta-problem for TermSysSat

**Meta-problem** for TermSysSat (Assume $\mathbf{P} \neq \mathbf{NP}$)

**Given:** $\mathbf{A} = (A; f_1, \ldots, f_k)$

**Asked:** Is TermSysSat($\mathbf{A}$) $\in \mathbf{P}$?

**Asked:** Does Core($\mathbf{A}^\circ$) have a Siggers polymorphism?

# The meta-problem for TermSysSat

**Meta-problem** for TermSysSat (Assume $\mathbf{P} \neq \mathbf{NP}$)

**Given:** $\mathbf{A} = (A; f_1, \ldots, f_k)$

**Asked:** Is TermSysSat($\mathbf{A}$) $\in \mathbf{P}$?

**Asked:** Does Core($\mathbf{A}^\circ$) have a Siggers polymorphism?

In cm varieties: **Asked:** Does $\mathbf{A}$ have an abelian core?

**Theorem** Mayr 2023

There is a quasi-polynomial algorithm that decides whether a given finite $\mathbf{A}$ in a cm variety has an abelian core.

$q(n)$ is quasi-polynomial if $\exists c, d, N > 0 \ \forall n \geq N \ : \ q(n) \leq c2^{\log(n)^d}$.

# Solving CSPs by solving term equations

# Solving CSP's through systems of equations

**Theorem**.
For every finite relational structure $\mathbb{D}$ of finite type, there is a finite algebra $\mathbf{A}(\mathbb{D})$ such that CSP($\mathbb{D}$) and TermSysSat($\mathbf{A}(\mathbb{D})$) are polynomial time reducible to each other.

1. Klíma, Tesson, Thérien 2007:
   Assume $\mathbb{D} = (D, \rho)$ is a digraph. $\mathbf{A}(\mathbb{D})$ is a semigroup with $5|D| + |\rho| + 1$ elements that satisfies $x^2 \approx x$ and $xyz \approx yxz$.
2. Broniek 2015:
   Assume $\mathbb{D} = (D, R)$ with $R \subseteq D^r$. $\mathbf{A}(\mathbb{D})$ is a unary algebra with $|D| + |R| + 2$ elements and $r + 4$ unary operations.

# Solving systems over modules

# Solving systems over modules

Let $\mathbf{A}$ be an $\mathbf{R}$-module.

- The polynomial algorithm provided by the theory uses the Bulatov-Dalmau-algorithm (2006) to solve instances of $\mathrm{CSP}(\mathbf{A}^{\circ})$, which has the Mal'cev term of $\mathbf{A}$ as a polymorphism.
- In practice, Hermite-decomposition is useful.

# Solving systems over modules

We solve

$$\begin{pmatrix} 10 & 16 & 0 \\ 15 & 24 & 30 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 66 \end{pmatrix}$$

over $\mathbb{Z}$.

# Solving systems over modules

We solve

$$\begin{pmatrix} 10 & 16 & 0 \\ 15 & 24 & 30 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 66 \end{pmatrix}$$

over $\mathbb{Z}$. To this end, we compute a $\mathbb{Z}$-Basis of the row module of

$$\begin{pmatrix} -4 & -66 & 1 & 0 & 0 & 0 \\ 10 & 15 & 0 & 1 & 0 & 0 \\ 16 & 24 & 0 & 0 & 1 & 0 \\ 0 & 30 & 0 & 0 & 0 & 1 \end{pmatrix}$$

using the Hermite normal form (1851, polynomial time since 1979).

# Solving systems over modules

We solve

$$\begin{pmatrix} 10 & 16 & 0 \\ 15 & 24 & 30 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 66 \end{pmatrix}$$

over $\mathbb{Z}$. We have

$$\mathrm{row}\left(\begin{pmatrix} -4 & -66 & 1 & 0 & 0 & 0 \\ 10 & 15 & 0 & 1 & 0 & 0 \\ 16 & 24 & 0 & 0 & 1 & 0 \\ 0 & 30 & 0 & 0 & 0 & 1 \end{pmatrix}\right) = \mathrm{row}\left(\begin{pmatrix} 2 & 3 & 0 & 5 & -3 & 0 \\ 0 & 30 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 & -1 & 2 \\ 0 & 0 & 0 & 8 & -5 & 0 \end{pmatrix}\right)$$

and thus $S = \{(2,-1,2) + t\,(8,-5,0) \mid t \in \mathbb{Z}\}$.

# Solving systems over modules

**Problem:** Find all $(z_1, z_2, z_3) \in \mathbb{Z}[x, y]^3$ with

$$(10y)z_1 + 0z_2 + (4x)z_3 = 4x^3.$$

**Solution:** Compute the (reduced strong) Gröbner basis (with respect to a certain order) of the row module of

$$A' := \begin{pmatrix} -4x^3 & 1 & 0 & 0 & 0 \\ 10y & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 4x & 0 & 0 & 0 & 1 \end{pmatrix}.$$

# Solving systems over modules

We solve $(10y)z_1 + 0z_2 + (4x)z_3 = 4x^3$.

$$\mathrm{row}\left(\begin{pmatrix} -4x^3 & 1 & 0 & 0 & 0 \\ 10y & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 4x & 0 & 0 & 0 & 1 \end{pmatrix}\right) = \mathrm{row}\left(\begin{pmatrix} 2xy & 0 & x & 0 & -2y \\ 4x & 0 & 0 & 0 & 1 \\ 10y & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & x^2 \\ 0 & 0 & 2x & 0 & -5y \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}\right).$$

Hence $S = (0, 0, x^2) + \langle (2x, 0, -5y), (0, 1, 0) \rangle$.

# II. One equation

# Supernilpotent algebras

**Theorem** (Coordinatization, EA 2019).

Let $\mathbf{A}$ be supernilpotent of order $p^\alpha$ in a cm variety. Then there are operations $+, \times$ on $A$ and $D \in \mathbb{N}$ such that

1. $\mathbb{F} = (A; +, \times)$ is a field,
2. for all $n \in \mathbb{N}$ and $p \in \mathrm{Pol}_n(\mathbf{A})$, there is by $P \in \mathbb{F}[X_1, \ldots, X_n]$ with $P^{\mathbb{F}} = p^{\mathbf{A}}$ and $\deg(f) \leq D$.

**Corollary** Kompatscher 2018.

Let $\mathbf{A}$ be supernilpotent in a cm variety. Then one equation

$$s(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)$$

can be solved in polynomial time.

# Supernilpotent algebras

**Corollary** Kompatscher 2018.

Let $\mathbf{A}$ be supernilpotent in a cm variety. Then one equation $s(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)$ can be solved in polynomial time.

- Assume $|A| = p^\alpha =: q$.
- $\exists S, T \in \mathbb{F}_q[X_1, \ldots, X_n] : S^{\mathbb{F}} = s^{\boldsymbol{A}}$ and $T^{\mathbb{F}} = t^{\boldsymbol{A}}$ with $\deg(S) \leq D$, $\deg(T) \leq D$.
- $\deg(1 - (S - T)^{q-1}) \leq (q-1)D$.
- Let $(a_1, \ldots, a_n)$ be the nonzero of $P := 1 - (S - T)^{q-1}$ with the smallest number of nonzero entries. WLOG $(a_1, \ldots, a_n) = (a_1, \ldots, a_k, 0, \ldots, 0)$.
- $Q(X_1, \ldots, X_k) := P(X_1, \ldots, X_k, 0, \ldots, 0)$ satisfies $\deg(Q) \geq k$.
- Hence $k \leq (q-1)D$.
- Look for a solution of $s = t$ with at most $(q-1)D$ entries different from $a \in A$.

# Supernilpotent algebras

Why was it easy to solve equations?

- We could reduce the search space from $|A|^n$ to a hitting set of size $c(|A|)\, n^k$.
- There are either no or many solutions.

**Theorem** Warning 1935.

$f \in \mathbb{F}_q[X_1, \ldots, X_n]$.

If the polynomial $f$ with $\deg(f) < n$ has a zero, then the number of zeros of $f$ is at least $q^{n-\deg(f)}$.

**Consequence:** if there is a solution, picking $\boldsymbol{a}$ at random yields a solution with probability at least $q^{-\deg(f)} \rightsquigarrow$ linear time Monte Carlo-algorithm (Kawałek, Krzackowski 2020).

# Solving equations over groups

# Solving equations over groups

**G** a finite group. PolSat(**G**) asks whether

$$s(x_1, \ldots, x_k, g_1, \ldots, g_l) = t(x_1, \ldots, x_k, g_1, \ldots, g_l)$$

has a solution in $G^k$.

Input size: lengths of terms $s$ and $t$.

- **G** nilpotent $\Rightarrow$ PolSat(**G**) $\in$ **P** (Horváth, 2011).
- **G** not solvable $\Rightarrow$ PolSat(**G**) is **NP**-complete (Goldmann and Russell, 1999).
- **G** not solvable $\Rightarrow$ there exists $e \in \mathrm{Pol}_1(\mathbf{A})$ such that PolSat($\mathbf{A} + e$) has no subexponential algorithm, or the exponential time hypothesis fails (Rossi, EA 2024).
- $\mathbf{G} = P \rtimes A$ with $P$ $p$-group and $A$ abelian $\Rightarrow$ PolSat(**G**) $\in$ **P**. (Földvári, Horváth 2019).
  Even solving a fixed number of equations is in **P** (Nuspl 2021).

# Solving equations over groups

**Theorem** Idziak, Kawałek, Krzaczkowski; Weiß 2020

If $\textsc{PolSat}(S_4) \in \mathbf{P}$, then for every $\varepsilon > 0$, we can solve 3-$\textsc{Sat}$ in time $O(2^{\varepsilon n})$, contradicting the Exponential Time Hypothesis by Impagliazzo and Paturi from 1999.

- The result is not just about $S_4$, but about all groups of Fitting length (length of shortest composition series with nilpotent quotients) at least 3.
- An algorithm with running time $O(n^{c(\log(n))^d})$ is consistent with ETH, where $n$ is the length of the input terms.

# Solving equations over groups

- We make a (still unsuccessful) attempt to find an $O(n^{c(\log(n))^d})$ algorithm for $\textsc{PolSat}(S_4)$.
- Easier problem: $\textsc{PolEqv}(S_4)$:
  Input: $p, q$ polynomial terms over $S_4$.
  Output: Is $\forall \boldsymbol{a} \in {S_4}^n : p(\boldsymbol{a}) = q(\boldsymbol{a})$ true?

# Identity checking for groups

We want to check whether $p(x_1, \ldots, x_n) = 1$ for all $\boldsymbol{x} \in {S_4}^n$.

**Definition.** $p \in \mathrm{Pol}_k(S_4)$ is absorbing if for all $\boldsymbol{x}$: $1 \in \{x_1, \ldots, x_k\} \Rightarrow$ $p(x_1, \ldots, x_k) = 1$.

- Suppose we can prove: every nonconstant absorbing polynomial has length at least $2^{ck^{1/d}}$.
- Then pick $\boldsymbol{a}$ with $p(\boldsymbol{a}) \neq 1$ with maximal amount of 1's. WLOG $\boldsymbol{a} = (a_1, \ldots, a_k, 1, \ldots, 1)$.
- Then $q(x_1, \ldots, x_k) := p(x_1, \ldots, x_k, 1, \ldots, 1)$ is absorbing.
- Hence $\mathrm{length}(p) \geq 2^{ck^{1/d}}$.
- Then $k \leq \frac{1}{c}(\log_2(\mathrm{length}(p)))^d$.
- Input size $m$ : greater than $\max(n, \mathrm{length}(p))$.
- There are at most $m^{c_2 \log^d(m)}$ tuples with at most $k$ entries $\neq 1 \rightsquigarrow$ algorithm of complexity $O(m^{c \log^d(m)})$.

# Finite fields

# Absorbing polynomial functions over finite fields

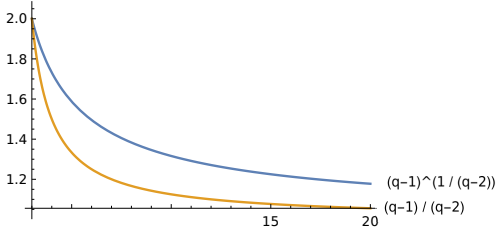
**Definition.** Let $\mathbb{F}$ be a field, $f \in \mathbb{F}[X_1, \ldots, X_n]$ is absorbing at $a$ for $S :\Leftrightarrow$ $\forall (x_1, \ldots, x_n) \in S^n : a \in \{x_1, \ldots, x_n\} \Rightarrow f(x_1, \ldots, x_n) = 0$.

**Theorem** (Grünbacher, Hametner, EA 2024).

$S \subseteq \mathbb{F}_q \setminus \{0\}$, $a \in S$. If $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ is absorbing at $a$ for $S$ and $f$ is not identically 0 on $S$, then $f$ contains at least $(\frac{q-1}{q-2})^n$ monomials.

**Examples:**

1. $\omega :=$ primitive element of $\mathbb{F}_4$, $S := \{1, \omega\}$. Then $f := \prod_{i=1}^{n}(X_i - \omega)$ is absorbing at $\omega$ for $S$ and has $2^n$ monomials.
2. (Grünbacher) There is an absorbing function with at most $(\sqrt[q-2]{q-1})^n$ monomials (if $q - 2 \mid n$).



# Equation solving over finite fields

**Intuition:** If there is a solution, there is one in the neighborhood.

**Theorem** Grünbacher, Hametner, EA 2024

Let $q > 2$, $n \in \mathbb{N}$, $f \in \mathbb{F}_q[X_1, \ldots, X_n] \setminus \{0\}$ with $M(f)$ monomials, $S \subseteq \mathbb{F}_q \setminus \{0\}$, $t := \frac{q-1}{q-2}$. Let

$$V(f) := \{\boldsymbol{x} \in S^n \mid f(\boldsymbol{x}) = 0\}.$$

If $V(f) \neq \varnothing$, then for every $\boldsymbol{a} \in S^n$, there is $\boldsymbol{b} \in V(f)$ with

$$d_H(\boldsymbol{a}, \boldsymbol{b}) \leq \frac{1}{\log_2(t)}\big(1 + (q-1)\log_2(M(f))\big).$$

This gives quasi-polynomial time algorithms for the following questions:

1. Does $f$ have a zero in $S^n$?
2. Does $f$ vanish identically on $S^n$?

# III. Quasi-identities

# Quasi-identities in universal algebra

- $\mathbf{A}$ algebra, $s_i, t_i, u, v$ terms.
- We ask whether $S = \{\boldsymbol{x} \in A^n \mid \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x})\}$ is contained in $U = \{\boldsymbol{x} \in A^n \mid u(\boldsymbol{x}) = v(\boldsymbol{x})\}$.
- This holds if the formula

$$\forall \boldsymbol{x} \; : \; \Big( \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}) \Big) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$$

  holds in $\mathbf{A}$.
- Such a formula is called a conditional identity or quasi-identity.
- We want to determine the validity of this formula.

# Quasi-identities in universal algebra

- $\mathbf{A}$ algebra, $s_i, t_i, u, v$ terms.
- We ask whether $S = \{\boldsymbol{x} \in A^n \mid \bigwedge_{i \in \underline{k}} s_i^{\mathbf{A}}(\boldsymbol{x}) = t_i^{\mathbf{A}}(\boldsymbol{x})\}$ is contained in $U = \{\boldsymbol{x} \in A^n \mid u^{\mathbf{A}}(\boldsymbol{x}) = v^{\mathbf{A}}(\boldsymbol{x})\}$.
- This holds if the formula

$$\forall \boldsymbol{x} \; : \; \Big( \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}) \Big) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$$

  holds in $\mathbf{A}$.
- Such a formula is called a conditional identity or quasi-identity.
- We want to determine the validity of this formula.

# Algebras that satisfy the same quasi-identities

Some facts on quasi-identities:

- Classes of algebras defined by quasi-identities are called quasivarieties. The quasivariety generated by $K$ is $ISP_{\mathrm{u}}P_{\mathrm{fin}}\,K$.
- Generalization: infinite pre-condition, finitely many variables:
  Considered in Universal Algebraic Geometry. Closure operator: $LSP$.
- Generalization: infinite pre-condition, arbitrary many variables:
  Closure operator $ISP$.
- $\mathbf{A}, \mathbf{B}$ finite, of finite type, $|\mathbf{A}| = n$. Then $\mathbf{A} \in Q(\mathbf{B}) \Leftrightarrow \mathbf{A} \in IS\,\{\mathbf{B}^{\binom{n}{2}}\}$.

# Quasi-identity validity

Let $\mathbf{A}$ be an algebra. QuasiIdVal($\mathbf{A}$) is the problem:

**Given:** A quasi-identity $\Phi := \forall \boldsymbol{x} : \left( \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}) \right) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$.
Here, $s_i, t_i, u, v$ are terms in the language of $\mathbf{A}$ over the variables $\boldsymbol{x}$.

**Asked:** Does $\Phi$ hold in $\mathbf{A}$?

# Quasi-identity validity

Let $\mathbf{A}$ be an algebra. QuasiIdVal($\mathbf{A}$) is the problem:

**Given:** A quasi-identity $\Phi := \forall \boldsymbol{x} : \left(\bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x})\right) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$.
Here, $s_i, t_i$, $u, v$ are terms in the language of $\mathbf{A}$ over the variables $\boldsymbol{x}$.

**Asked:** Does $\Phi$ hold in $\mathbf{A}$?

**Computational Complexity:** For finite $\mathbf{A}$ of finite type, QuasiIdVal($\mathbf{A}$) is in co-**NP**:
$\boldsymbol{a} \in A^n$ witnesses failure of $\Phi$ if $\left(\bigwedge_{i \in \underline{k}} s_i^{\mathbf{A}}(\boldsymbol{a}) = t_i^{\mathbf{A}}(\boldsymbol{a})\right) \wedge u^{\mathbf{A}}(\boldsymbol{a}) \neq v^{\mathbf{A}}(\boldsymbol{a})$.

# Quasi-identity validity

Let $\mathbf{A}$ be an algebra. QuasiIdVal($\mathbf{A}$) is the problem:

**Given:** A quasi-identity $\Phi := \forall \boldsymbol{x} : \left(\bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x})\right) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$.
Here, $s_i, t_i$, $u, v$ are terms in the language of $\mathbf{A}$ over the variables $\boldsymbol{x}$.

**Asked:** Does $\Phi$ hold in $\mathbf{A}$?

**Computational Complexity:** For finite $\mathbf{A}$ of finite type, QuasiIdVal($\mathbf{A}$) is in co-**NP**:
$\boldsymbol{a} \in A^n$ witnesses failure of $\Phi$ if $\left(\bigwedge_{i \in \underline{k}} s_i^{\mathbf{A}}(\boldsymbol{a}) = t_i^{\mathbf{A}}(\boldsymbol{a})\right) \wedge u^{\mathbf{A}}(\boldsymbol{a}) \neq v^{\mathbf{A}}(\boldsymbol{a})$.

**Exponential time method:** A quasi-identity of length $\ell$ contains at most $\ell$ different variables that can take at most $|A|^\ell$ values.

**Question:** For which algebras do we have faster methods (e.g. polynomial time)?

# Quasi-identity validity and polynomial systems

**Relations to other problems:**

- If we can decide solvability of polynomial systems, then we can check the validity of quasi-identities.

# Quasi-identity validity and polynomial systems

**Relations to other problems:**

- If we can decide solvability of polynomial systems, then we can check the validity of quasi-identities.
- We search for a counter-example: $\forall \boldsymbol{x} : \left( \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}) \right) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$ holds iff for all $a, b \in A$ with $a \neq b$,

$$\bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}),\ u(\boldsymbol{x}) = a,\ v(\boldsymbol{x}) = b$$

  has no solution.
- These systems use constants: $a$ and $b$.
  Therefore they are **polynomial systems** and not just **term systems**.

# Quasi-identity validity and polynomial systems

**Relations to other problems:**

- If we can decide solvability of polynomial systems, then we can check the validity of quasi-identities.
- We search for a counter-example: $\forall \boldsymbol{x} : \left( \bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}) \right) \Longrightarrow u(\boldsymbol{x}) = v(\boldsymbol{x})$ holds iff for all $a, b \in A$ with $a \neq b$,

$$\bigwedge_{i \in \underline{k}} s_i(\boldsymbol{x}) = t_i(\boldsymbol{x}),\ u(\boldsymbol{x}) = a,\ v(\boldsymbol{x}) = b$$

  has no solution.
- These systems use constants: $a$ and $b$.
  Therefore they are **polynomial systems** and not just **term systems**.
- **Conclusion:** $\textsc{QuasiIdVal}(\mathbf{A}) \leq_{\text{truth table}} \textsc{PolSysSat}(\mathbf{A})$.

# Quasi-identity validity and systems of term equations

- If we can check the validity of quasi-identities, then we can decide solvability of term equations.

# Quasi-identity validity and systems of term equations

- If we can check the validity of quasi-identities, then we can decide solvability of term equations.
- The system $s_1 = t_1, \ldots, s_k = t_k$ has no solution iff

$$s_1 = t_1 \wedge \ldots \wedge s_k = t_k \Longrightarrow y = z$$

is valid in $\mathbf{A}$. ($y, z$ ... new variables, $|A| > 1$).

# Quasi-identity validity and systems of term equations

- If we can check the validity of quasi-identities, then we can decide solvability of term equations.
- The system $s_1 = t_1, \ldots, s_k = t_k$ has no solution iff

$$s_1 = t_1 \wedge \ldots \wedge s_k = t_k \Longrightarrow y = z$$

  is valid in $\mathbf{A}$. ($y, z$ ... new variables, $|A| > 1$).
- **Conclusion:** co-TermSysSat($\mathbf{A}$) $\leq_P$ QuasiIdVal($\mathbf{A}$).

# Quasi-identity validity and checking term equivalence

- If we can check the validity of quasi-identities, we can check whether two terms induce the same function.

# Quasi-identity validity and checking term equivalence

- If we can check the validity of quasi-identities, we can check whether two terms induce the same function.
- $\forall \boldsymbol{x} : s(\boldsymbol{x}) = t(\boldsymbol{x})$ is valid iff

$$y = y \Longrightarrow s(\boldsymbol{x}) = t(\boldsymbol{x})$$

is valid in $\mathbf{A}$.

# Quasi-identity validity and checking term equivalence

- If we can check the validity of quasi-identities, we can check whether two terms induce the same function.
- $\forall \boldsymbol{x} : s(\boldsymbol{x}) = t(\boldsymbol{x})$ is valid iff

$$y = y \Longrightarrow s(\boldsymbol{x}) = t(\boldsymbol{x})$$

  is valid in $\mathbf{A}$.
- **Conclusion:** $\textsc{TermEqv}(\mathbf{A}) \leq_P \textsc{QuasiIdVal}(\mathbf{A})$.

# Quasi-identity validity: connections with well-studied problems.

**Connections:**

- $\textsc{QuasiIdVal}(\mathbf{A}) \leq_{\text{truth table}} \textsc{PolSysSat}(\mathbf{A})$.
- co-$\textsc{TermSysSat}(\mathbf{A}) \leq_P \textsc{QuasiIdVal}(\mathbf{A})$.
- $\textsc{TermEqv}(\mathbf{A}) \leq_P \textsc{QuasiIdVal}(\mathbf{A})$.

# Quasi-identity validity: connections with well-studied problems.

**Connections:**

- $\textsc{QuasiIdVal}(\mathbf{A}) \leq_{\text{truth table}} \textsc{PolSysSat}(\mathbf{A})$.
- co-$\textsc{TermSysSat}(\mathbf{A}) \leq_P \textsc{QuasiIdVal}(\mathbf{A})$.
- $\textsc{TermEqv}(\mathbf{A}) \leq_P \textsc{QuasiIdVal}(\mathbf{A})$.
- In 2004, M. Volkov constructed a 10-element semigroup $\mathbf{Q}$ with $\textsc{TermEqv}(\mathbf{Q}) \in \mathbf{P}$, and $\textsc{QuasiIdVal}(\mathbf{Q})$ co-$\mathbf{NP}$-complete because it solves 3-$\textsc{Colorability}$ for graphs.

# Quasi-identity validity: connections with well-studied problems.

Let $\mathbf{A}$ be an algebra with a Mal'cev term.

**Consequences:**

- $\mathbf{A}$ is abelian $\Longrightarrow$ $\textsc{QuasiIdVal}(\mathbf{A}) \in \mathbf{P}$.
  (Reason: $\textsc{PolSysSat}$, which is analyzed in [Larose, Zádori 2006])

# Quasi-identity validity: connections with well-studied problems.

Let $\mathbf{A}$ be an algebra with a Mal'cev term.

**Consequences:**

- $\mathbf{A}$ is abelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) $\in \mathbf{P}$.
  (Reason: PolSysSat, which is analyzed in [Larose, Zádori 2006])
- Core($\mathbf{A}$) is nonabelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.
  (Reason: TermSysSat, which is analyzed in [Mayr 2023])

# Quasi-identity validity: connections with well-studied problems.

Let $\mathbf{A}$ be an algebra with a Mal'cev term.

**Consequences:**

- $\mathbf{A}$ is abelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) $\in \mathbf{P}$.
  (Reason: PolSysSat, which is analyzed in [Larose, Zádori 2006])
- Core($\mathbf{A}$) is nonabelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.
  (Reason: TermSysSat, which is analyzed in [Mayr 2023])
- $\mathbf{A}$ non-solvable group $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.
  (Reason: TermEqv, which is analyzed in [Horváth, Lawrence, Mérai, Szabó 2007])

# Quasi-identity validity: connections with well-studied problems.

Let $\mathbf{A}$ be an algebra with a Mal'cev term.

**Consequences:**

- $\mathbf{A}$ is abelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) $\in \mathbf{P}$.
  (Reason: PolSysSat, which is analyzed in [Larose, Zádori 2006])
- Core($\mathbf{A}$) is nonabelian $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.
  (Reason: TermSysSat, which is analyzed in [Mayr 2023])
- $\mathbf{A}$ non-solvable group $\Longrightarrow$ QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.
  (Reason: TermEqv, which is analyzed in [Horváth, Lawrence, Mérai, Szabó 2007])

**Open:** nonabelian nilpotent groups, nonzero nilpotent rings.

# A reduction of graph coloring to quasi-identities

# Quasi-identity validity

**Theorem** Aichinger, Grünbacher, STACS 2023

**A** finite algebra of finite type with a Mal'cev term. Then

1. QuasiIdVal($\mathbf{A}$) $\in \mathbf{P}$ if $\mathbf{A}$ is abelian.
2. QuasiIdVal($\mathbf{A}$) is co-**NP**-complete if $\mathbf{A}$ is nonabelian.

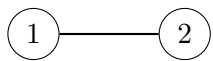
New content: item (2).

**Proof idea:** we reduce the $H$-coloring problem to QuasiIdVal($\mathbf{A}$).

# $H$-coloring of graphs

$H$-COLORING:
**Given:** a graph $G$.
**Asked:** Is there a graph homomorphism $h$ from $G$ to $H$ ($G \to H$)?

- $H = K_2$:

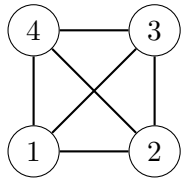



$G \to H$ iff $G$ is bipartite: edges in $G$ only go from $h^{-1}(\{1\})$ to $h^{-1}(\{2\})$.

# $H$-coloring of graphs

$H$-COLORING:
**Given:** a graph $G$.
**Asked:** Is there a graph homomorphism $h$ from $G$ to $H$ ($G \to H$)?

- $H = K_4$:

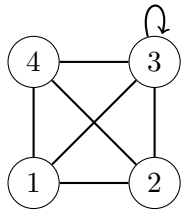



$G \to H$ if the vertices of $G$ can be coloured with 4 colors such that no adjacent vertices have the same colour.

# $H$-coloring of graphs

$H$-COLORING:
**Given:** a graph $G$.
**Asked:** Is there a graph homomorphism $h$ from $G$ to $H$ ($G \to H$)?

- $H$ a graph with loops:




$G \to H$ holds for every graph $G$: use $h(v) = 3$ for each vertex $v$ of $G$.

**Theorem** Hell, Nešetřil 1990.

Let $H$ be a finite loopless graph that contains a triangle. Then $H$-COLORING is **NP**-complete.

A consequence stated in CSP-language:

**Theorem**

Let $\mathbb{H} = (H, \rho)$ be a relational structure with an antireflexive and symmetric binary relation $\rho$.
If $\mathbb{H}$ has $\mathbb{K}_3 = (\{1, 2, 3\}; \neq)$ as a substructure, then CSP($\mathbb{H}$) is **NP**-complete.

# Proof of the Theorem

**Plan:**

- We want to prove that checking the validity of quasi-identities of $\mathbf{R} := (3\mathbb{Z}_{27}, +, -, \cdot, 0)$ is co-**NP**-complete.
- We will show: there is a graph $H$ such that

  for every graph $G$ : $\quad G \to H \iff$ the quasi-identity $\Phi(G)$ is not valid.

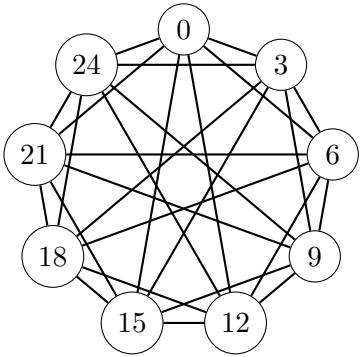
- This will imply that QuasiIdVal($\mathbf{R}$) is co-**NP**-complete.

**Details:**

- $R = \{[0]_{27}, [3]_{27}, \ldots, [24]_{27}\}$.
- $H$ is the "difference graph" or "apartness graph" on $R$ : $(r, s)$ is an edge if $r - s \notin \{[0]_{27}, [9]_{27}, [18]_{27}\}$.

# Proof of the Theorem

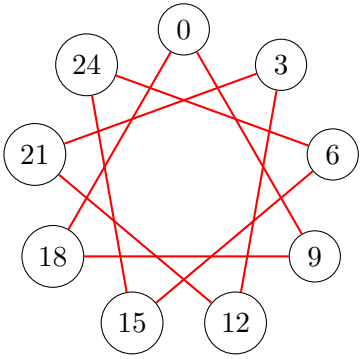
The graph $H$ for $3\mathbb{Z}_{27}$




$$E(H) = \{(x, y) \mid x - y \notin \{0, 9, 18\}\}.$$

# Proof of the Theorem

The graph $H$ for $3\mathbb{Z}_{27}$

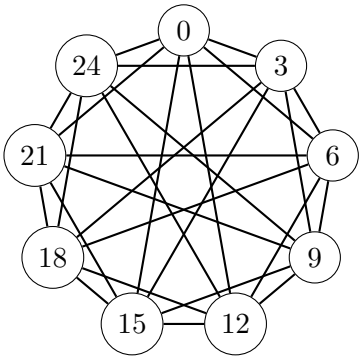



$E(H) = \{(x, y) \mid x - y \notin \{0, 9, 18\}\}$.

- non-edges of $H$

# Proof of the Theorem

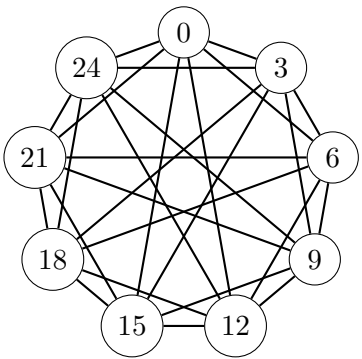
The graph $H$ for $3\mathbb{Z}_{27}$




- $G$ graph. We want to find out whether $G \to H$ **using a quasi-identity on R**.
- $\Phi = \Big( \bigwedge_{(u,v)\in E(G)} a = z_{u,v} \cdot (x_u - x_v) \Big) \Rightarrow a = 0.$

# Proof of the Theorem

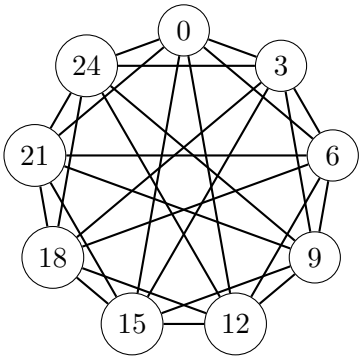
The graph $H$ for $3\mathbb{Z}_{27}$




- $G$ graph. We want to find out whether $G \to H$ **using a quasi-identity on R**.
- $\Phi = \Big( \bigwedge_{(u,v)\in E(G)} a = z_{u,v} \cdot (x_u - x_v) \Big) \Rightarrow a = 0.$
- Suppose $\Phi$ is invalid. Then $a \neq 0$.
- Let $(u, v) \in E(G)$. Then $x_u - x_v \notin \{0, 9, 18\}$.
- Thus $(x_u, x_v)$ is an edge of $H$.
- $u \mapsto x_u$ is a homomorphism from $G$ to $H$.
- Hence if $\Phi$ is invalid, $G \to H$.

# Proof of the Theorem

The graph $H$ for $3\mathbb{Z}_{27}$




- $G$ graph. We want to find out whether $G \to H$ **using a quasi-identity on R**.
- $\Phi = \Big( \bigwedge_{(u,v)\in E(G)} a = z_{u,v} \cdot (x_u - x_v) \Big) \Rightarrow a = 0.$
- Suppose $G \to H$.
- ...
- This is a counterexample to $\Phi$.
- Hence $\Phi$ is invalid.

# Proof of the Theorem

- Hence $\Phi$ is not valid iff $G \to H$.
- $H$-coloring is **NP**-complete [Hell, Nešetřil 1990].
- Thus QuasiIdVal($\mathbf{R}$) is co-**NP**-complete.

# Proof for Mal'cev algebras

**Theorem**

Let $\mathbf{A}$ be a finite nonabelian algebra of finite type with a Mal'cev term. Then QuasiIdVal($\mathbf{A}$) is co-**NP**-complete.

- Instead of the ring multiplication, use commutators [Smith 1976, Hagemann, Herrmann 1979].
- This works for subdirectly irreducible $\mathbf{A}$.
- For arbitrary $\mathbf{A}$, use "difference graphs" for several congruences of $\mathbf{A}$.
- Order these graphs and pick a maximal one.
- EA and Simon Grünbacher. The Complexity of Checking Quasi-Identities over Finite Algebras with a Mal'cev Term, STACS 2023.

# IV. Open questions

# Polynomial equations on $S_4$

**Problem:** Solve $\textsc{PolSat}(S_4)$ or $\textsc{PolEqv}(S_4)$ in time $m^{c\log^d(m)}$.

Possible approaches:

- Show: every nonconstant absorbing polynomial in $k$ arguments has length at least $2^{ck^{1/d}}$.
- The following problems are – in some sense – equivalent:
  - Does a fully expanded polynomial $p \in \mathrm{Mat}_2(\mathbb{F}_2)\langle X_1, \ldots, X_n\rangle$ vanish on all $6^n$ invertible inputs?
  - (Grünbacher) Let $\omega$ be primitive in $\mathbb{F}_4$, $p_1, \ldots, p_n$ expanded polynomials in $\mathbb{Z}_3[X_1, \ldots, X_n]$. Does

$$\sum_{i=1}^{n} \omega^{p_i(X_1,\ldots,X_n)}$$

    vanish on $\{-1, 1\}^n$?

# Circuit equivalence

- The polynomials $p, q$ may be given as circuits.
- We want to test whether they compute the same function.

## Problem 1 from [Kawałek, Kompatscher, Krzaczkowski, STACS 2024]

Let $\mathbf{A}$ be a finite algebra from a congruence modular variety with supernilpotent rank 2. Is there a deterministic polynomial time algorithm solving circuit equivalence on $\mathbf{A}$?

Děkuji za pozvání a pozornost!