Group isomorphism is nearly-linear time for most orders

IEEE Foundations On Computer Science FOCS 2021

Heiko Dietrich
Monash University, Australia

James B. Wilson (presenting)
Colorado State University, USA

February 8, 2022

Motivation

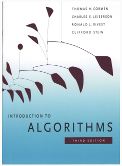
Outward Facing Motive: honest data types

Where in this...

.

Outward Facing Motive: honest data types

Where in this...

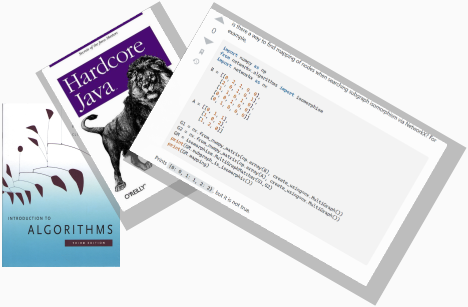


Outward Facing Motive: honest data types

Where in this...

Outward Facing Motive: honest data types

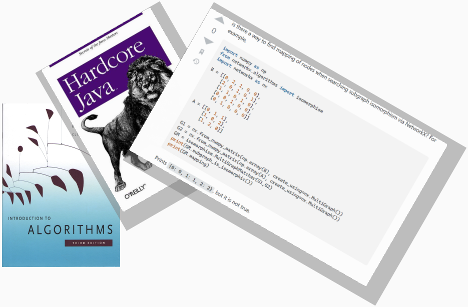
Where in this...



.

Outward Facing Motive: honest data types

Where in this...



...do we send people to get help making this...

```
boolean equals(Object that) {
    // <this> can transform into <that>?
}
```

Why groups(oids)?

- **Transitive→ Partial Multiplication**

$$trans_{xyz} : (x \equiv y) \wedge (y \equiv z) \Rightarrow (x \equiv z)$$
$$* : Eq \times Eq \dashrightarrow Eq$$

- **Reflexive→ Identity**

$$\frac{\begin{aligned} refl_x &: x \Rightarrow (x \equiv x) \\ trans_{xxy} &: (x \equiv x) \wedge (x \equiv y) \Rightarrow (x \equiv y) \end{aligned}}{Identity : refl * evidence = evidence}$$

- **Symmetric→ Inverse**

$$\frac{\begin{aligned} sym_{xy} &: (x \equiv y) \Rightarrow (y \equiv x) \\ trans_{xyx} &: (x \equiv y) \wedge (x \equiv y) \Rightarrow (x \equiv x) \end{aligned}}{Inverses : evidence * (evidence)^{-1} = refl}$$

Why groups(oids)?

- **Transitive→ Partial Multiplication**

$$trans_{xyz} :(x \equiv y) \wedge (y \equiv z) \Rightarrow (x \equiv z)$$
$$* :Eq \times Eq \dashrightarrow Eq$$

- **Reflexive→ Identity**

$$\frac{\begin{array}{r} refl_x : x \Rightarrow (x \equiv x) \\ trans_{xxy} : (x \equiv x) \wedge (x \equiv y) \Rightarrow (x \equiv y) \end{array}}{Identity : refl * evidence = evidence}$$

- **Symmetric→ Inverse**

$$\frac{\begin{array}{r} sym_{xy} : (x \equiv y) \Rightarrow (y \equiv x) \\ trans_{xyx} : (x \equiv y) \wedge (x \equiv y) \Rightarrow (x \equiv x) \end{array}}{Inverses : evidence * (evidence)^{-1} = refl}$$

Why groups(oids)?

- **Transitive→ Partial Multiplication**

$$trans_{xyz} : (x \equiv y) \wedge (y \equiv z) \Rightarrow (x \equiv z)$$
$$* : Eq \times Eq \dashrightarrow Eq$$

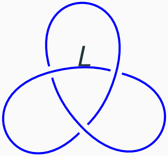
- **Reflexive→ Identity**

$$refl_x : x \Rightarrow (x \equiv x)$$
$$\frac{trans_{xxy} : (x \equiv x) \wedge (x \equiv y) \Rightarrow (x \equiv y)}{Identity : refl * evidence = evidence}$$

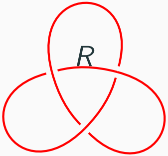
- **Symmetric→ Inverse**

$$sym_{xy} : (x \equiv y) \Rightarrow (y \equiv x)$$
$$\frac{trans_{xyx} : (x \equiv y) \wedge (x \equiv y) \Rightarrow (x \equiv x)}{Inverses : evidence * (evidence)^{-1} = refl}$$

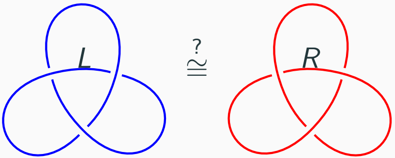
Anatomy of hard equality



$\stackrel{?}{\cong}$



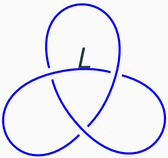
Anatomy of hard equality



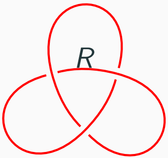
Relax category until automorphisms computable.

$\pi_1(L) = \langle x, y \mid x^2 = y^3 \rangle$ $\qquad$ $\pi_1(R) = \langle x, y \mid x^2 = y^{-3} \rangle$

Anatomy of hard equality



$\overset{?}{\cong}$



Relax category until automorphisms computable.

$$\pi_1(L) = \langle x, y \mid x^2 = y^3 \rangle \quad \overset{?}{\cong} \quad \pi_1(R) = \langle x, y \mid x^2 = y^{-3} \rangle$$

$$\mathrm{Aut}_\chi \pi_1(L) \quad \not\cong_\chi \quad \mathrm{Aut}_\chi \pi_1(R)$$

Recursively refine comparing automorphisms with incrementally stricter properties. E.g. respect crossing number χ?

Inward facing Motive: equalivance surveys complexity

FPGroupIso Adjan, Rabin ‘50’s		Undecideable
PlaneGroupIso Dietrich et.al. STACS‘21		Σ_3^P
BlackBoxGroupIso Babai-Szemerédi FOCS‘84	**MatroidIso**	Σ_2^P
PermGroupIso Luks DIMACS		$\Sigma_1^P = NP$
CayleyGroupIso Tarjan	**GraphIso** Babai	$DTIME(2^{\log^c n})$
TableGroupIsoAbel Kivitha (nearly-linear in RAM model)		$DTIME(n^2 \log^c n)$
TableGroupIsoMostOrders, IsGroup This Talk		$DTIME(n \log^c n)$

Schreier-Sims

Problem: Transport

Given: A set Ω, allowed permutations X, $\omega, \omega' \in \Omega$

Return: decide if a string g over X maps ω to ω', written $\omega^g = \omega'$, and give all such g.[1]

[1] Give words W over X so that $\omega^h = \omega'$ implies $h = wg$ for a string w over W.

String Isomorphism

"Eighth" $==$ "HeigHt"

String Isomorphism

- **Given** strings $s, t : I \to \Sigma$ allowed permutations $G = \langle g_k \rangle \leq \mathrm{Sym}_I$, $H = \langle h_k \rangle \leq \mathrm{Sym}_\Sigma$
- **Return** strings $g = g_{a_1} \cdots g_{a_u}$ and $h = h_{b_1} \cdots h_{b_v}$ where $h(s_i) = t_{g(i)}$; or prove impossible.

$$\begin{array}{ccc} I & \xrightarrow{s} & \Sigma \\ \downarrow g & & \downarrow h \\ I & \xrightarrow{t} & \Sigma \end{array}$$

Theorem. (Babai 2016+) If Σ fixed, $\textsc{StringIso}$ is in Quasipolynomial $n^{O((\log n)^c)}$-time.

($\textsc{GraphIso} \leq_P \textsc{StringIso}$)

String Isomorphism

"Eighth" == "HeigHt"

String Isomorphism

- **Given** strings $s, t : I \to \Sigma$ allowed permutations $G = \langle g_k \rangle \leq \mathrm{Sym}_I$, $H = \langle h_k \rangle \leq \mathrm{Sym}_\Sigma$
- **Return** strings $g = g_{a_1} \cdots g_{a_u}$ and $h = h_{b_1} \cdots h_{b_v}$ where $h(s_i) = t_{g(i)}$; or prove impossible.

$$\begin{tikzcd} I \arrow[r, "s"] \arrow[d, "g"] & \Sigma \arrow[d, "h"] \\ I \arrow[r, "t"] & \Sigma \end{tikzcd}$$

Theorem. (Babai 2016+) If Σ fixed, StringIso is in Quasipolynomial $n^{O((\log n)^c)}$-time.

(GraphIso$\leq_P$ StringIso)

String Isomorphism

$$\text{“Eighth”} == \text{“HeigHt”}$$

String Isomorphism

- **Given** strings $s, t : I \to \Sigma$ allowed permutations $G = \langle g_k \rangle \leq \mathrm{Sym}_I$, $H = \langle h_k \rangle \leq \mathrm{Sym}_\Sigma$
- **Return** strings $g = g_{a_1} \cdots g_{a_u}$ and $h = h_{b_1} \cdots h_{b_v}$ where $h(s_i) = t_{g(i)}$; or prove impossible.

$$\begin{tikzcd} I \arrow[r, "s"] \arrow[d, "g"] & \Sigma \arrow[d, "h"] \\ I \arrow[r, "t"] & \Sigma \end{tikzcd}$$

Theorem. (Babai 2016+) If Σ fixed, $\textsc{StringIso}$ is in Quasipolynomial $n^{O((\log n)^c)}$-time.

($\textsc{GraphIso} \leq_P \textsc{StringIso}$)

Snapshot of solving a hard isomorphism problem

Math Facts

Schreier-Sims

‘65

Snapshot of solving a hard isomorphism problem

Math Facts

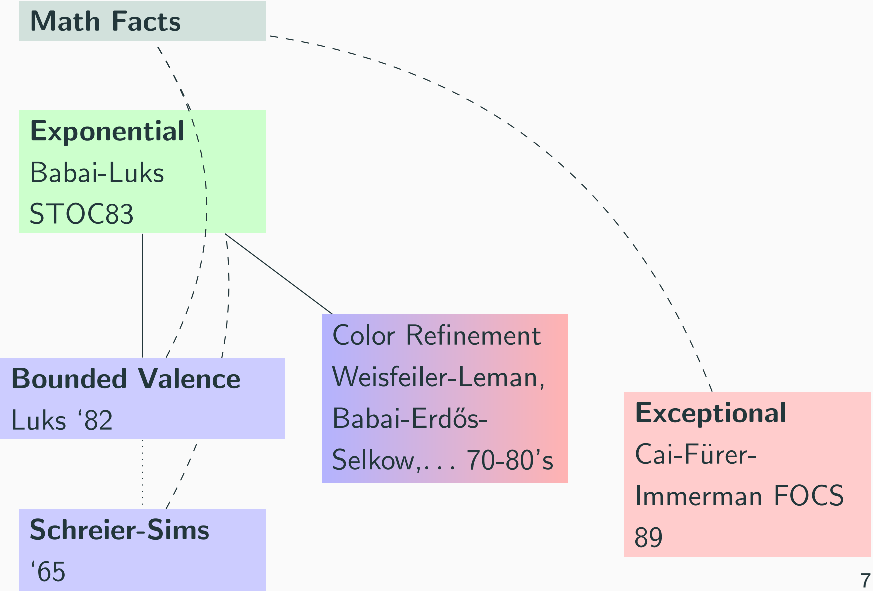
Exponential
Babai-Luks
STOC83

Bounded Valence
Luks ‘82

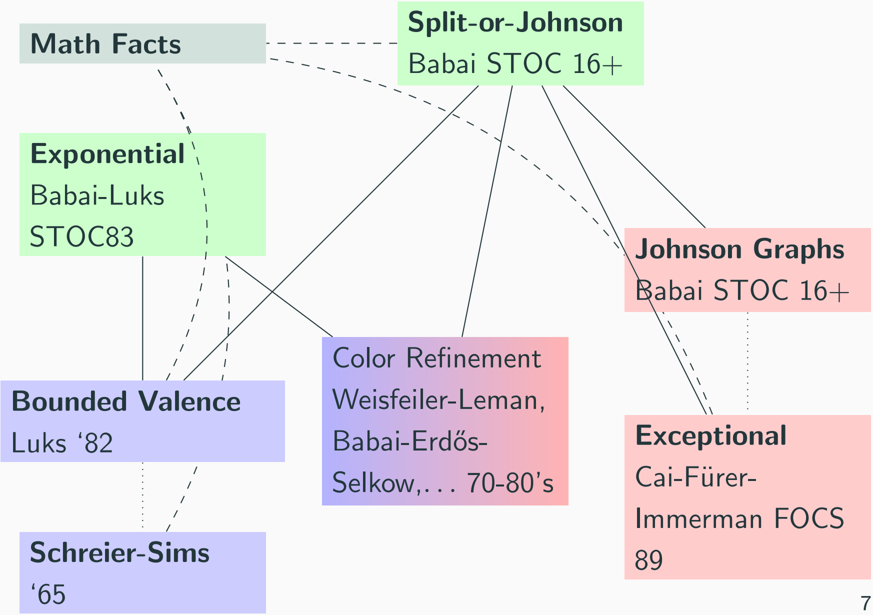
Color Refinement
Weisfeiler-Leman,
Babai-Erdős-
Selkow,... 70-80’s

Schreier-Sims
‘65

Snapshot of solving a hard isomorphism problem



Snapshot of solving a hard isomorphism problem



Isomorphism of Tables

Code Equivalence

$$\boxed{\begin{array}{cc} L & i \\ v & e \end{array}} == \boxed{\begin{array}{cc} E & v \\ i & l \end{array}}$$

Code Equivalence[2]

- **Given** $s, t : I \times J \to \Sigma$, (generators for) permutations $R \leq \mathrm{Sym}_I$, $C \leq \mathrm{Sym}_J$, & $V \leq \mathrm{Sym}_\Sigma$
- **Return** $\sigma \in R$, $\tau \in C$, $\mu \in V$,

$$\begin{array}{ccccc} I & \times & J & \xrightarrow{s} & \Sigma \\ \downarrow \sigma & & \downarrow \tau & & \downarrow \mu \\ I & \times & J & \xrightarrow{t} & \Sigma \end{array}$$

$$\mu(s_{ij}) = t_{\sigma(i)\tau(j)}$$

Babai-Codenotti-Grochow-Qiao $2^{O(n)}$-time bound for constant alphabet Σ (SODA '11)

Builds on Luks $2^{O(n)}$-hypergraph isomorphism, FOCS '99.

[2] Non-linear twisted, with variable alphabet.

Algebra Isomorphism

$$\begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \\ \hline\end{array} == \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 1 & 5 & 2 & 4 \\ 4 & 5 & 2 & 3 & 1 \\ 5 & 3 & 4 & 1 & 2 \\ \hline\end{array}$$

Algebra Isomorphism

- **Given** $s, t : I \times I \to I$, (generators for) permutations $G \leq \mathrm{Sym}_I$,
- **Return** $\sigma \in G$,

$$\begin{array}{ccccc} I & \times & I & \xrightarrow{s} & I \\ \downarrow\sigma & & \downarrow\sigma & & \downarrow\sigma \\ I & \times & I & \xrightarrow{t} & I \end{array}$$

$$\sigma(s_{ij}) = t_{\sigma(i)\sigma(j)}$$

Group Isomorphism Strategy

Math Facts

Generator Test
Tarjan, Miller
STOC78

Group Isomorphism Strategy

Data Types

Math Facts

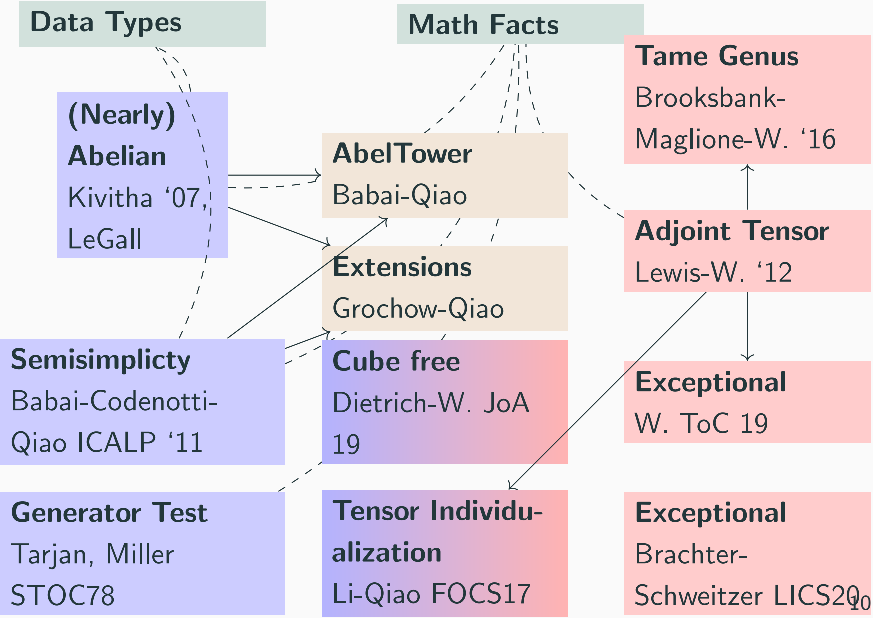
(Nearly) Abelian
Kivitha '07, LeGall

Adjoint Tensor
Lewis-W. '12

Semisimplicty
Babai-Codenotti-Qiao ICALP '11

Generator Test
Tarjan, Miller STOC78

Group Isomorphism Strategy



Group Isomorphism of most orders

Divide and conquer

Isomorphism of $G = A \times B$, i.e.

$$(a, b)(\tilde{a}, \tilde{b}) = (a\tilde{a}, b\tilde{b})$$

reduces to isomorphism of A and B in parallel.[3]

Isomorphism of $G = A \ltimes_\theta B$, i.e.

$$(a, b)(\tilde{a}, \tilde{b}) = (a\tilde{a}, \theta(\tilde{a})(b)\tilde{b})$$

reduces to isomorphism of A and B sequentially, plus adjusting θ.[4]

[3] Technical detail: decompose maximally and adjust non-unique decompositions by Krull-Schmidt; see W. 2008.

[4] Lemma 2.1

Divide and conquer

Isomorphism of $G = A \times B$, i.e.

$$(a, b)(\tilde{a}, \tilde{b}) = (a\tilde{a}, b\tilde{b})$$

reduces to isomorphism of A and B in parallel.[3]

Isomorphism of $G = A \ltimes_\theta B$, i.e.

$$(a, b)(\tilde{a}, \tilde{b}) = (a\tilde{a}, \theta(\tilde{a})(b)\tilde{b})$$

reduces to isomorphism of A and B sequentially, plus adjusting θ.[4]

[3] Technical detail: decompose maximally and adjust non-unique decompositions by Krull-Schmidt; see W. 2008.
[4] Lemma 2.1

Division Graph: Erdős-Pálfy

Factor n into a *graph* $\Gamma(n)$.

Edge $(p_i^{e_i}, p_j^{e_j})$ where $p_i | p_j^k - 1$ for some $k \leq e_j$, & symmetrically.

Erdős-Pálfy, 1999
A group G of order n factors as

$$N_1 \times \cdots \times N_\ell,$$

$|N_i| = n_i$, order of connected components of $\Gamma(n)$.

Example $n = 1785$

$|G| = n \Rightarrow N_{3\cdot 7} \times N_5 \times N_{17}$

Division Graph: Erdős-Pálfy

Factor n into a *graph* $\Gamma(n)$.

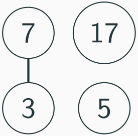
Edge $(p_i^{e_i}, p_j^{e_j})$ where $p_i | p_j^k - 1$ for some $k \leq e_j$, & symmetrically.

Erdős-Pálfy, 1999
A group G of order n factors as

$$N_1 \times \cdots \times N_\ell,$$

$|N_i| = n_i$, order of connected components of $\Gamma(n)$.

Example $n = 1785$



$$|G| = n \Rightarrow N_{3\cdot 7} \times N_5 \times N_{17}$$

Extending implications

Factor n into a *direct hypergraph* $\mathcal{H}(n)$. (i) *Oriented* Erdős-Pálfy *hyper*-edges, (ii) exceptions for finite nonabelian simple groups.

Proposition
A group G of order n factors as

$$N_0 \ltimes (N_1 \ltimes \cdots \ltimes N_\ell),$$

$|N_i| = n_i$, where n_i is order of interconnected components of $\mathcal{H}(n)$.

Example $n = 5,810,340$

$|G| = n \Rightarrow N_{60} \ltimes (N_{179} \times N_{547})$

Extending implications

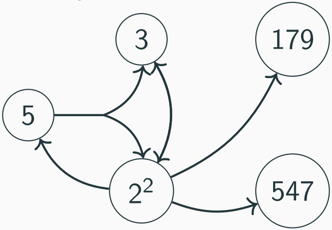
Factor n into a *direct hypergraph* $\mathcal{H}(n)$. (i) *Oriented* Erdős-Pálfy *hyper*-edges, (ii) exceptions for finite nonabelian simple groups.

Proposition
A group G of order n factors as

$$N_0 \ltimes (N_1 \ltimes \cdots \ltimes N_\ell),$$

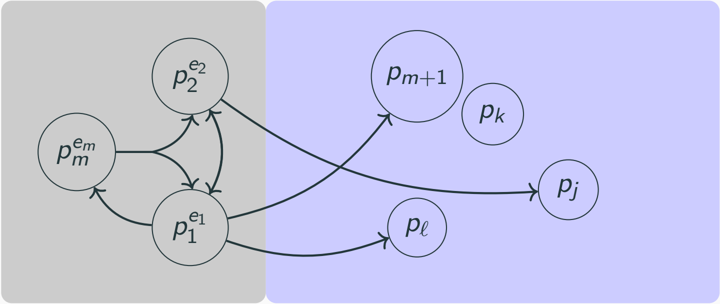
$|N_i| = n_i$, where n_i is order of interconnected components of $\mathcal{H}(n)$.

Example $n = 5, 810, 340$

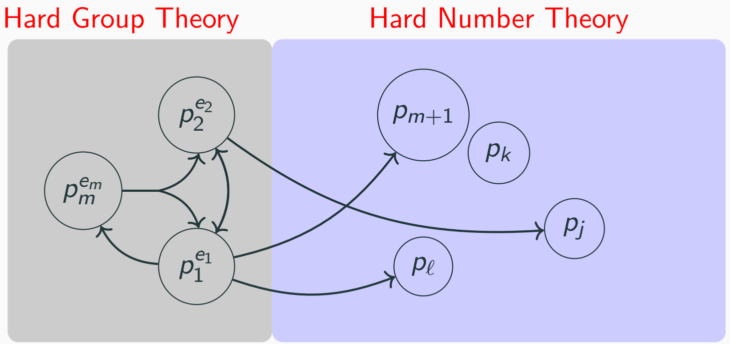


$$|G| = n \Rightarrow N_{60} \ltimes (N_{179} \times N_{547})$$

Most Orders

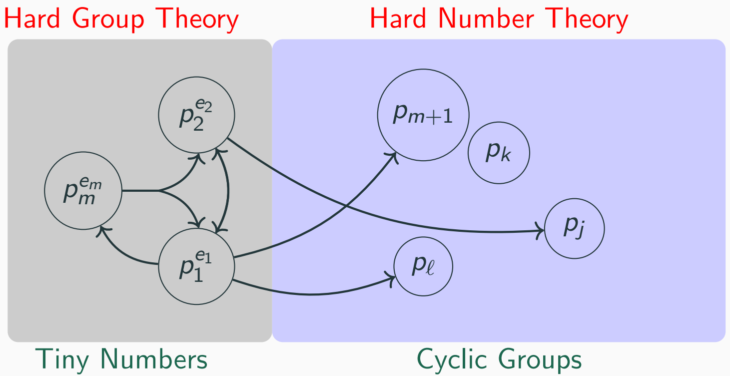


Most Orders



Hard Group Theory
Hard Number Theory
$p_2^{e_2}$
$p_m^{e_m}$
$p_1^{e_1}$
p_{m+1}
p_k
p_j
p_ℓ

Most Orders



$$G = H \ltimes (\mathbb{Z}_{p_{m+1}} \times \cdots \times \mathbb{Z}_{p_\ell})$$

Is it a Group Table?

Promise-to-decision

A great many computational algebra are analyzed as **promise problems** not **decision problems**.

Identity testing is needed to remove the promise; unsolvable in general (word problem), but on tables at least brute-force.

Theorem Rajagopalan-Schulman, 2000
Given $* : [n] \times [n] \to [n]$, test associativity (and other identities) in nearly-linear time $\tilde{O}(n^2)$ in RAM model (constant time ops and memory access). Also can test if a group.

Promise-to-decision

A great many computational algebra are analyzed as **promise problems** not **decision problems**.

Identity testing is needed to remove the promise; unsolvable in general (word problem), but on tables at least brute-force.

Theorem Rajagopalan-Schulman, 2000
Given $*: [n] \times [n] \to [n]$, test associativity (and other identities) in nearly-linear time $\tilde{O}(n^2)$ in RAM model (constant time ops and memory access). Also can test if a group.

RAM-to-TM

At larger scales Turing Machine (TM) model better match to computations that are communication bounded (typical in practice).

RAM -¿ TM at most a quadratic blow-up.

Corollary
Nearly Quadratic-time $\tilde{O}(n^4)$ on multi-tape Turing Machine (TM).

Theorem Dietrich-W.
Given $* : [n] \times [n] \to [n]$, test if a group in time nearly-linear time $\tilde{O}(n^2)$ on deterministic multi-tape TM.

RAM-to-TM

At larger scales Turing Machine (TM) model better match to computations that are communication bounded (typical in practice).

RAM -¿ TM at most a quadratic blow-up.

Corollary
Nearly Quadratic-time $\tilde{O}(n^4)$ on multi-tape Turing Machine (TM).

Theorem Dietrich-W.
Given $* : [n] \times [n] \to [n]$, test if a group in time nearly-linear time $\tilde{O}(n^2)$ on deterministic multi-tape TM.

IsGroup

•	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

IsGroup

$\bullet$	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

$$\rho(2) = \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 1 & 4 & 5 & 3 \\ \hline\end{array} = (1,2)(3,5,4)$$

IsGroup

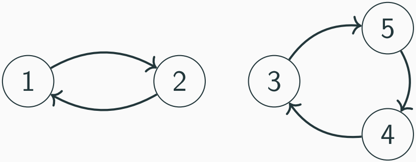
$\bullet$	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

$$\rho(2) = \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 1 & 4 & 5 & 3 \\ \hline \end{array} = (1,2)(3,5,4)$$

$$G_1 = \langle (354) \rangle$$

$\bullet$	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

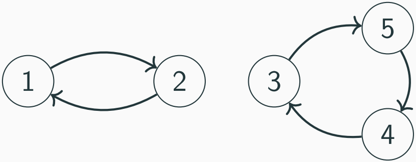
$$\rho(2) = \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 1 & 4 & 5 & 3 \\ \hline\end{array} = (1,2)(3,5,4)$$



$G_1 = \langle (354) \rangle$ $\quad$ $G_{13} = \{1\}$

•	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

$$\rho(2) = \begin{array}{|ccccc|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 1 & 4 & 5 & 3 \\ \hline \end{array} = (1,2)(3,5,4)$$



$G_1 = \langle (354) \rangle$ $\quad$ $G_{13} = \{1\}$

$|G| = [G : G_1][G_1 : G_{13}]$
$= 2 \cdot 3 = 6.$

Should be 5,
not a group.

IsGroup

$*$	1	2	3	4	5
1	1	2	3	4	5
2	2	4	1	5	3
3	3	5	4	2	1
4	4	1	5	3	2
5	5	3	2	1	4

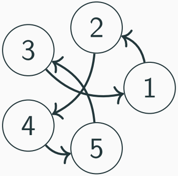
IsGroup

$*$	1	2	3	4	5
1	1	2	3	4	5
2	2	4	1	5	3
3	3	5	4	2	1
4	4	1	5	3	2
5	5	3	2	1	4

$$\rho(2) = \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 4 & 1 & 5 & 3 \\ \hline\end{array} = (1, 2, 4, 5, 3)$$

IsGroup

*	1	2	3	4	5
1	1	2	3	4	5
2	2	4	1	5	3
3	3	5	4	2	1
4	4	1	5	3	2
5	5	3	2	1	4

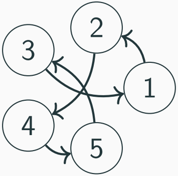


$$|G| = [G : G_1| = 5$$

$$\rho(2) = \begin{array}{|ccccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 4 & 1 & 5 & 3 \\ \hline\end{array} = (1, 2, 4, 5, 3)$$

IsGroup

$*$	1	2	3	4	5
1	1	2	3	4	5
2	2	4	1	5	3
3	3	5	4	2	1
4	4	1	5	3	2
5	5	3	2	1	4



$|G| = [G : G_1| = 5$

$$\rho(2) = \begin{array}{|ccccc|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline 2 & 4 & 1 & 5 & 3 \\ \hline \end{array} = (1, 2, 4, 5, 3)$$

$*$	1	2	3	4	5
$\rho(2)^0$	1	2	3	4	5
$\rho(2)^1$	2	4	1	5	3
$\rho(2)^2$	4	5	2	3	1

$\rho(2)^2 = 45231 \neq T_4 = 41532$

Not a group.

Summary

Summary

IsGroup nearly linear time
From $\tilde{O}(n^4)$ to $\tilde{O}(n^2)$: Promise-to-decision by transferring group to permutation model.

GroupIso most orders nearly linear time
From $n^{O(\log n)}$ to $\tilde{O}(n^2)$: Split group into

$$\text{hard group} \ltimes \text{hard numbers} = \text{tiny numbers} \ltimes \text{cyclic groups}.$$

Then standard divide-and-conquer.

Thanks to: Newton Institute (Cambridge, UK) EPSRC Grant Number EP/R014604/1, Australian Research Council grant DP190100317, and Simons Foundation Grant 636189.