A polynomial-time theory of matrix groups

Ákos Seress
Joint with László Babai, Robert Beals

September 2010

Basics of Computational Group Theory

Two basic ways to input a group:

1) generators–relators

$G = \langle a, b \mid a^2 = b^5 = 1, b^a = b^{-1} \rangle$

Is $G = 1$? Is G finite?

Computational Group Theory 101

Two basic ways to input a group:

1) generators–relators

$G = \langle a, b \mid a^2 = b^5 = 1, b^a = b^{-1} \rangle$

Is $G = 1$? Is G finite?

2) "concrete" representation, with generating permutations or matrices (over finite fields)

$G = \langle X \rangle$

$|G| =$? Given $g \in S_n$ (or $g \in \mathrm{GL}(d, q)$), is $g \in G$?

G is too big to list

Key task: Constructive membership problem
1) Find “nice” generators Y for G
2) Procedure to write any $g \in G$ in terms of Y
(straight-line program from Y to g)

Key task: Constructive membership problem
1) Find "nice" generators Y for G
2) Procedure to write any $g \in G$ in terms of Y
(straight-line program from Y to g)

SLP: sequence of expressions $w_1, \ldots, w_m$
w_i: symbol for some $y \in Y$ or
$w_i = (w_j, w_k)$ for some $j, k < i$ or
$w_i = (w_j, -1)$ for some $j < i$

Evaluation:
$\mathrm{eval}(w_j, w_k) = \mathrm{eval}(w_j)\mathrm{eval}(w_k)$
$\mathrm{eval}(w_j, -1) = \mathrm{eval}(w_j)^{-1}$
$\mathrm{eval}(w_m) = g$

Key task: Constructive membership problem
1) Find "nice" generators Y for G
2) Procedure to write any $g \in G$ in terms of Y
(straight-line program from Y to g)

SLP: sequence of expressions $w_1, \dots, w_m$
w_i: symbol for some $y \in Y$ or
$w_i = (w_j, w_k)$ for some $j, k < i$ or
$w_i = (w_j, -1)$ for some $j < i$

Evaluation:
$\mathrm{eval}(w_j, w_k) = \mathrm{eval}(w_j)\mathrm{eval}(w_k)$
$\mathrm{eval}(w_j, -1) = \mathrm{eval}(w_j)^{-1}$
$\mathrm{eval}(w_m) = g$

$Y = \{y\}, \quad g = y^{1024}$
$w_1 = y, \; w_2 = (w_1, w_1), \; w_3 = (w_2, w_2), \dots, w_{11} = (w_{10}, w_{10})$

Permutation groups: under control

Parker–Nikolai (1958)
$G = \langle X \rangle \leq S_n$; is $G \geq A_n$?

$G \geq A_n \iff G$ transitive, contains p-cycle with $n/2 < p < n-2$
Random sample of elements has good chance to have the order of one of them divisible by such p

Constructive membership (Sims, late 1960's)

$G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$
$B = (\beta_1, \ldots, \beta_m)$ is base for G:
pointwise stabilizer $G_B = 1$
$G = G^{[1]} \geq G^{[2]} \geq \cdots \geq G^{[m+1]} = 1$
$G^{[i]} = G_{(\beta_1, \ldots, \beta_{i-1})}$

Constructive membership (Sims, late 1960's)

$G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$
$B = (\beta_1, \ldots, \beta_m)$ is base for G:
pointwise stabilizer $G_B = 1$
$G = G^{[1]} \geq G^{[2]} \geq \cdots \geq G^{[m+1]} = 1$
$G^{[i]} = G_{(\beta_1, \ldots, \beta_{i-1})}$

$G = \langle (1,5,2,6), (1,2)(3,4)(5,6) \rangle$
$B = (1,3)$
$G > G_1 > G_{13} = 1$

Constructive membership (Sims, late 1960's)

$G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$
$B = (\beta_1, \ldots, \beta_m)$ is base for G:
pointwise stabilizer $G_B = 1$
$G = G^{[1]} \geq G^{[2]} \geq \cdots \geq G^{[m+1]} = 1$
$G^{[i]} = G_{(\beta_1, \ldots, \beta_{i-1})}$

Y is strong generating set (SGS):
$G^{[i]} = \langle G^{[i]} \cap Y \rangle$

Constructive membership (Sims, late 1960's)

$G = \langle X \rangle \le \mathrm{Sym}(\Omega)$
$B = (\beta_1, \ldots, \beta_m)$ is base for G:
pointwise stabilizer $G_B = 1$
$G = G^{[1]} \ge G^{[2]} \ge \cdots \ge G^{[m+1]} = 1$
$G^{[i]} = G_{(\beta_1,\ldots,\beta_{i-1})}$

Y is strong generating set (SGS):
$G^{[i]} = \langle G^{[i]} \cap Y \rangle$

$G = \langle (1,5,2,6), (1,2)(3,4)(5,6) \rangle$
$B = (1,3)$
$G > G_1 > G_{13} = 1$
$Y = \{(1,5,2,6), (1,2)(3,4)(5,6), (3,4)\}$

Constructive membership (Sims, late 1960's)

$G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$
$B = (\beta_1, \ldots, \beta_m)$ is base for G:
pointwise stabilizer $G_B = 1$
$G = G^{[1]} \geq G^{[2]} \geq \cdots \geq G^{[m+1]} = 1$
$G^{[i]} = G_{(\beta_1, \ldots, \beta_{i-1})}$

Y is strong generating set (SGS):
$G^{[i]} = \langle G^{[i]} \cap Y \rangle$
T_i: (right) cosetreps $G^{[i]}$ mod $G^{[i+1]}$ (easy from SGS)
sifting: write any $g \in G$ as $g = r_m \cdots r_1$, $r_i \in T_i$

Parallel (NC) algorithms: polynomially many processors, only polylogarithmic time

sifting is inherently sequential

Luks: divide-and-conquer approach, using normal subgroups instead of point stabilizer chain

Parallel (NC) algorithms: polynomially many processors, only polylogarithmic time

sifting is inherently sequential

Luks: divide-and-conquer approach, using normal subgroups instead of point stabilizer chain

Babai, Luks, Seress (1987): permutation groups in NC

By the time $|G|$ is computed: also composition series of G

Sequential consequence (Babai, Luks, Seress 1988)

$G \leq \mathrm{Sym}(\Omega)$ arbitrary

1) detect large alternating composition factors

2) constructive membership by special methods

3) rest of group: Sims's algorithm

4) put it together

Current status: randomized speedup of Sims, BLS
fast (both in practical and theoretical sense)
Babai, Beals, Cooperman, Finkelstein, Kantor, Law, Leedham-Green, Luks, Niemeyer, Praeger, Seress, Sims

Implementation: Neunhöffer, Seress

Current status: randomized speedup of Sims, BLS
fast (both in practical and theoretical sense)
Babai, Beals, Cooperman, Finkelstein, Kantor, Law, Leedham-Green, Luks, Niemeyer, Praeger, Seress, Sims

Implementation: Neunhöffer, Seress

Moral: in permutation groups, structural exploration and randomization not unavoidable, but helps

Current status: randomized speedup of Sims, BLS
fast (both in practical and theoretical sense)
Babai, Beals, Cooperman, Finkelstein, Kantor, Law, Leedham-Green, Luks, Niemeyer, Praeger, Seress, Sims

Implementation: Neunhöffer, Seress

Moral: in permutation groups, structural exploration and randomization not unavoidable, but helps

Á. Seress: Permutation Group Algorithms. Cambridge Univ. Press 2003.

Matrix groups: much harder

1) no subgroup chain with small indices (no Sims-type approach)

Matrix groups: much harder

1) no subgroup chain with small indices (no Sims-type approach)

2) even for 1x1 matrices:
$G = \langle 25, 142, 261 \rangle \leq \mathrm{GF}(683)^*$, $|G| =?$
Discrete log problem: $a, b \in \mathrm{GF}(q)^*$
Is $a \in \langle b \rangle$? If yes, $?x \, (a = b^x)$

Matrix groups: much harder

1) no subgroup chain with small indices (no Sims-type approach)

2) even for 1x1 matrices:
$G = \langle 25, 142, 261 \rangle \leq \mathrm{GF}(683)^*$, $|G| =?$
Discrete log problem: $a, b \in \mathrm{GF}(q)^*$
Is $a \in \langle b \rangle$? If yes, $?x \, (a = b^x)$

3) factorization of large integers

4) great variety of large primitive matrix groups

Matrix groups: much harder

1) no subgroup chain with small indices (no Sims-type approach)

2) even for 1x1 matrices:
$G = \langle 25, 142, 261 \rangle \leq \mathrm{GF}(683)^*$, $|G| =?$
Discrete log problem: $a, b \in \mathrm{GF}(q)^*$
Is $a \in \langle b \rangle$? If yes, $?x\,(a = b^x)$

3) factorization of large integers

4) great variety of large primitive matrix groups

Randomization, structural exploration (chopping into manageable pieces) are necessary

Luks (1992): In solvable matrix groups, order, constructive membership, composition series in deterministic polynomial time, using discrete log and factorization oracles

Luks (1992): In solvable matrix groups, order, constructive membership, composition series in deterministic polynomial time, using discrete log and factorization oracles

randomized algorithm is Monte Carlo: output may be wrong, with probability controlled by user

randomized algorithm is Las Vegas: output is always correct, may report failure with probability controlled by user

Neubüser's question

Given $G = \langle X \rangle \leq \mathrm{GL}(d, q)$; is $G \geq \mathrm{SL}(d, q)$?

Neumann, Praeger (1990): nonconstructive recognition of $\mathrm{SL}(d, q)$ (by Monte Carlo algorithm)

Geometric approach

(suggested by Neumann, Praeger in 1990, led by Leedham-Green and O'Brien)

Aschbacher's classification of matrix groups (1984): nine categories
C1–C7: reductive classes
natural normal subgroup N associated with the action of G
handle N, G/N recursively

C1: G acts reducibly

G/N, N: action and kernel of action on invariant subspace $W < V$

C2: imprimitive action

$V = V_1 \oplus \cdots \oplus V_k$, G permutes the V_i

G/N, N: permutation action on blocks and its kernel

etc.

Reduction bottoms out:

C8: giants classical groups in natural representation

C9: almost simple modulo scalars, absolutely irreducible action
no geometry to exploit

Black-box group approach of matrix groups

Babai, Beals (1993, 1999): find abstract structure

$1 \leq \mathrm{Rad}(G) \leq \mathrm{Soc}^*(G) \leq \mathrm{PKer}(G) \leq G$

$\mathrm{Rad}(G)$: largest solvable normal subgroup

$\mathrm{Soc}^*(G)/\mathrm{Rad}(G)$: socle of $G/\mathrm{Rad}(G)$
$\mathrm{Soc}^*(G)/\mathrm{Rad}(G) \cong T_1 \times \cdots \times T_k$
T_i nonabelian simple, G permutes them by conjugation

$\mathrm{PKer}(G)$: kernel of this permutation action
$\mathrm{PKer}(G)/\mathrm{Soc}^*(G) \leq \mathrm{Out}(T_1) \times \cdots \times \mathrm{Out}(T_k)$ solvable (CFSG)

Black-box group:

group elements are represented by 0-1 strings of uniform length n
given $g, h \in G$, oracle to compute (strings representing) gh, g^{-1}, and decide $g = 1$?

Black-box group:

group elements are represented by 0-1 strings of uniform length n
given $g, h \in G$, oracle to compute (strings representing) gh, g^{-1}, and decide $g = 1$?

Why lose information?
(i) sometimes we cannot use more info (random element generation, C9 groups)
(ii) permutation group elements as words in strong generators: faster group operation than perm multiplication
(iii) working in factor groups of matrix groups, we lose geometry

G black-box group

Construct $H \leq G$: compute generators for H

Recognize $H \leq G$: able to test membership in H

G black-box group

Construct $H \le G$: compute generators for H

Recognize $H \le G$: able to test membership in H

$H = Z(G), \mathrm{Rad}(G)$ are recognizable, but hard to construct

$N \lhd G$, N recognizable: we can consider G/N as black-box group

Black-box group of characteristic p: G is a section (factor group of a subgroup) of $\mathrm{GL}(n, p)$

Babai, Beals work with bb groups of characteristic p

1) Landazuri–Seitz–Zaleskii, Feit–Tits: Lie-type composition factors of characteristic $\neq p$ have small permutation representation

2) some idea about primes occurring in $|G|$

$$\begin{aligned} L = \{\text{primes} < n^9\} \\ \cup \ \{p^i - 1 \mid 1 \leq i \leq n\} \cup \{2^{2t+1} \pm 2^{t+1} + 1 \mid 1 \leq t \leq n\} \\ \cup \ \{2^{4t+2} + 2^{2t+1} + 1 \pm 2^{t+1}(2^{2t+1} + 1) \mid 1 \leq t \leq n\} \\ \cup \ \{3^{2t+1} \pm 3^{t+1} + 1 \mid 1 \leq t \leq n\} \end{aligned}$$

pseudo-primes: refinement of L into pairwise relative prime numbers

New results

Given $G = \langle X \rangle \leq \mathrm{GL}(d, p^e)$,
(1) find $|G/\mathrm{Rad}(G)|$ in Monte Carlo pol. time
(2) if p is odd then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time
(3) if $p = 2$ and no exceptional Lie-type groups among the T_i then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time using discrete log oracles

New results

Given $G = \langle X \rangle \leq \mathrm{GL}(d, p^e)$,
(1) find $|G/\mathrm{Rad}(G)|$ in Monte Carlo pol. time
(2) if p is odd then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time
(3) if $p = 2$ and no exceptional Lie-type groups among the T_i then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time using discrete log oracles

(4) if p is odd then constructive membership in G in Monte Carlo pol. time using discrete log oracles
(5) if $p = 2$ and no exceptional Lie-type groups among the T_i then constructive membership in G in Monte Carlo pol. time using discrete log oracles

New results

Given $G = \langle X \rangle \leq \mathrm{GL}(d, p^e)$,
(1) find $|G/\mathrm{Rad}(G)|$ in Monte Carlo pol. time
(2) if p is odd then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time
(3) if $p = 2$ and no exceptional Lie-type groups among the T_i then construct $\mathrm{Rad}(G)$ in Monte Carlo pol. time using discrete log oracles

(4) if p is odd then constructive membership in G in Monte Carlo pol. time using discrete log oracles
(5) if $p = 2$ and no exceptional Lie-type groups among the T_i then constructive membership in G in Monte Carlo pol. time using discrete log oracles

(6) for any p: if no exceptional Lie-type groups among the T_i then upgrade the algorithms to Las Vegas

$1 \leq \mathrm{Rad}(G) \leq \mathrm{Soc}^*(G) \leq \mathrm{PKer}(G) \leq G$
$\mathrm{Soc}^*(G)/\mathrm{Rad}(G) = T_1 \times \cdots \times T_k$

Babai, Beals (1999): Polynomial-time Monte Carlo algorithm to construct perfect $H_i \leq G$, $H_i/\mathrm{Rad}(H_i) \cong T_i$

$1 \leq \mathrm{Rad}(G) \leq \mathrm{Soc}^*(G) \leq \mathrm{PKer}(G) \leq G$
$\mathrm{Soc}^*(G)/\mathrm{Rad}(G) = T_1 \times \cdots \times T_k$

Babai, Beals (1999): Polynomial-time Monte Carlo algorithm to construct perfect $H_i \leq G$, $H_i/\mathrm{Rad}(H_i) \cong T_i$

consequences:
1) Construct $G/\mathrm{PKer}(G) \leq S_k$ and $\mathrm{Soc}^*(G)/\mathrm{Rad}(G)$
2) name the T_i (Babai–Kantor–Pálfy–Seress, Altseimer–Borovik)

$1 \le \mathrm{Rad}(G) \le \mathrm{Soc}^*(G) \le \mathrm{PKer}(G) \le G$
$\mathrm{Soc}^*(G)/\mathrm{Rad}(G) = T_1 \times \cdots \times T_k$

Babai, Beals (1999): Polynomial-time Monte Carlo algorithm to construct perfect $H_i \le G$, $H_i/\mathrm{Rad}(H_i) \cong T_i$

consequences:
1) Construct $G/\mathrm{PKer}(G) \le S_k$ and $\mathrm{Soc}^*(G)/\mathrm{Rad}(G)$
2) name the T_i (Babai–Kantor–Pálfy–Seress, Altseimer–Borovik)

works with black-box groups of characteristic p

New additions:

(1) construct $\mathrm{PKer}(G)/\mathrm{Soc}^*(G)$
(2) construct $\mathrm{Rad}(H_i)$ for $1 \leq i \leq k$
(3) construct the part of $\mathrm{Rad}(G)$ not generated by $\langle \mathrm{Rad}(H_i) \rangle$

New additions:

(1) construct $\mathrm{PKer}(G)/\mathrm{Soc}^*(G)$
(2) construct $\mathrm{Rad}(H_i)$ for $1 \leq i \leq k$
(3) construct the part of $\mathrm{Rad}(G)$ not generated by $\langle \mathrm{Rad}(H_i) \rangle$

(1), (3): in Monte Carlo polynomial time
for odd p, (2) in Monte Carlo polynomial time
works with black-box groups of characteristic p

New additions:

(1) construct $\mathrm{PKer}(G)/\mathrm{Soc}^*(G)$
(2) construct $\mathrm{Rad}(H_i)$ for $1 \leq i \leq k$
(3) construct the part of $\mathrm{Rad}(G)$ not generated by $\langle \mathrm{Rad}(H_i) \rangle$

(1), (3): in Monte Carlo polynomial time
for odd p, (2) in Monte Carlo polynomial time
works with black-box groups of characteristic p

for $p = 2$, (2) in Monte Carlo polynomial time using discrete log oracles if no T_i is exceptional
works only for matrix groups

(1) construct $\mathrm{PKer}(G)/\mathrm{Soc}^*(G) \leq \mathrm{Out}(T_1) \times \cdots \times \mathrm{Out}(T_k)$

Leedham-Green trick: T simple, $g \in \mathrm{Aut}(T)$; is $g \in T$?

multiply g by random $h \in T$
gcd of $\{|gh|\}$ is 1: $g \in T$
gcd > 1: with high probability, $g \notin T$

Justification by Babai–Pálfy–Saxl:
there is absolute constant c such that for all prime r, all simple T, proportion of elements of order not divisible by r is at least $c/\mathrm{rank}(T)$

(1) construct $\mathrm{PKer}(G)/\mathrm{Soc}^*(G) \leq \mathrm{Out}(T_1) \times \cdots \times \mathrm{Out}(T_k)$

Leedham-Green trick: T simple, $g \in \mathrm{Aut}(T)$; is $g \in T$?

multiply g by random $h \in T$
gcd of $\{|gh|\}$ is 1: $g \in T$
gcd > 1: with high probability, $g \notin T$

Justification by Babai–Pálfy–Saxl:
there is absolute constant c such that for all prime r, all simple T, proportion of elements of order not divisible by r is at least $c/\mathrm{rank}(T)$

Solution of (1): generalization of LG-trick to product of simples

for each $i \leq k$, construct regular permutation representation of projection of $\mathrm{PKer}(G)/\mathrm{Soc}^*(G)$ into $\mathrm{Out}(T_i)$

(2) $H = \mathrm{Rad}(H).T$ perfect, construct $\mathrm{Rad}(H)$

T sporadic: brute force
T alternating: constructive recognition in Monte Carlo pol. time (Beals–Leedham-Green–Niemeyer–Praeger–Seress)

(2) $H = \mathrm{Rad}(H).T$ perfect, construct $\mathrm{Rad}(H)$

T sporadic: brute force
T alternating: constructive recognition in Monte Carlo pol. time (Beals–Leedham-Green–Niemeyer–Praeger–Seress)
T Lie-type of characteristic $\neq p$: construct permutation representation (Babai–Beals)
T Lie-type of characteristic p, acting nontrivially on a non-p chief layer of $\mathrm{Rad}(H)$: construct permutation representation

(2) $H = \mathrm{Rad}(H).T$ perfect, construct $\mathrm{Rad}(H)$

T sporadic: brute force

T alternating: constructive recognition in Monte Carlo pol. time (Beals–Leedham-Green–Niemeyer–Praeger–Seress)

T Lie-type of characteristic $\neq p$: construct permutation representation (Babai–Beals)

T Lie-type of characteristic p, acting nontrivially on a non-p chief layer of $\mathrm{Rad}(H)$: construct permutation representation

Remains: T Lie-type of characteristic p, acting trivially on non-p chief layers of $\mathrm{Rad}(H)$

Two base cases

$H = Z_p^d.T$
Parker–Wilson, Yalcinkaya: in Monte Carlo pol. time, construct $h \in Z_p^d$
works only for odd p (uses centralizer of involution computations)

Two base cases

$H = Z_p^d.T$
Parker–Wilson, Yalcinkaya: in Monte Carlo pol. time, construct $h \in Z_p^d$
works only for odd p (uses centralizer of involution computations)

$H = Z_r^d.T$, $r \neq p$, $Z(H) = Z_r^d$
Babai–Shalev: in Monte Carlo pol. time, construct $h \in Z_r^d$
easy, based on Babai–Pálfy–Saxl

both algorithms work for black-box groups

$H = \mathrm{Rad}(H).T$, $S \lhd \mathrm{Rad}(H)$ is the already constructed part of $\mathrm{Rad}(H)$

if $S \neq \mathrm{Rad}(H)$: there is $S \leq N \text{ char } \mathrm{Rad}(H)$, $\mathrm{Rad}(H)/N$ elementary abelian, H/N one of the base cases
S, N not recognizable
How to apply the base case algorithms?

$H = \mathrm{Rad}(H).T$, $S \lhd \mathrm{Rad}(H)$ is the already constructed part of $\mathrm{Rad}(H)$

if $S \neq \mathrm{Rad}(H)$: there is $S \leq N \text{ char } \mathrm{Rad}(H)$, $\mathrm{Rad}(H)/N$ elementary abelian, H/N one of the base cases
S, N not recognizable
How to apply the base case algorithms?

Adaptation principle
run the base case algorithms for H/N
at queries $h = 1$? (i.e., $h \in N$?)
if $h \in \mathrm{Rad}(H)$ then answer yes, store h

$H = \mathrm{Rad}(H).T$, $S \lhd \mathrm{Rad}(H)$ is the already constructed part of $\mathrm{Rad}(H)$

if $S \neq \mathrm{Rad}(H)$: there is $S \leq N$ char $\mathrm{Rad}(H)$, $\mathrm{Rad}(H)/N$ elementary abelian, H/N one of the base cases
S, N not recognizable
How to apply the base case algorithms?

Adaptation principle
run the base case algorithms for H/N
at queries $h = 1$? (i.e., $h \in N$?)
if $h \in \mathrm{Rad}(H)$ then answer yes, store h

if all answers are correct: with high probability, one of the algorithms constructs $h \in \mathrm{Rad}(H) \setminus N$

if one of the answers is incorrect: stored $h \in \mathrm{Rad}(H) \setminus N$

$p = 2$, $H = \mathrm{Rad}(H).T$

Kantor–Seress: construct quasisimple matrix representation $M = Z.T$, $Z = Z(M)$ for T
uses that H is a matrix group

Brooksbank, Kantor: for T classical, constructive recognition of M in Monte Carlo pol. time, using $\mathrm{PSL}(2, q)$ oracles

Conder–Leedham-Green–O'Brien: constructive recognition of any quasisimple matrix representation of $\mathrm{PSL}(2, q)$, using discrete logs

(3) construct the part of $\mathrm{Rad}(G)$ not generated by $\langle \mathrm{Rad}(H_i) \rangle$

easy, based on Babai–Pálfy–Saxl and adaptation principle

constructive membership (construct SLP to given $g \in G$)

Holmes–Linton–O'Brien–Ryba–Wilson:
given T simple Lie-type and $g \in T$ as bb group of odd char. p

Monte Carlo pol. time algorithm to construct involutions $x_1, x_2, x_3 \in T$ so that SLP to g is reduced to constructive membership in $C_T(x_i)$

if T is classical of large rank then the x_i can be chosen to be balanced (± 1-eigenspaces are roughly half dimensional)

constructive membership (construct SLP to given $g \in G$)

Holmes–Linton–O'Brien–Ryba–Wilson:
given T simple Lie-type and $g \in T$ as bb group of odd char. p

Monte Carlo pol. time algorithm to construct involutions $x_1, x_2, x_3 \in T$ so that SLP to g is reduced to constructive membership in $C_T(x_i)$

if T is classical of large rank then the x_i can be chosen to be balanced (± 1-eigenspaces are roughly half dimensional)

Theorem (*) Constructive membership in T in Monte Carlo polynomial time
$C_T(x_i)$ are not quasisimple; recursive steps use full machinery

constructive membership in arbitrary G of odd char.

$1 \leq \mathrm{Rad}(G) \leq \mathrm{Soc}^*(G) \leq \mathrm{PKer}(G) \leq G$

given $g \in G$, construct SLP to $h \in G$ with $\mathrm{Soc}^*(G)g = \mathrm{Soc}^*(G)h$ by permutation group methods

constructive membership in arbitrary G of odd char.

$1 \leq \mathrm{Rad}(G) \leq \mathrm{Soc}^*(G) \leq \mathrm{PKer}(G) \leq G$

given $g \in G$, construct SLP to $h \in G$ with $\mathrm{Soc}^*(G)g = \mathrm{Soc}^*(G)h$ by permutation group methods

use Theorem (*) to construct SLP to $k \in \mathrm{Soc}^*(G)$ with $\mathrm{Rad}(G)gh^{-1} = \mathrm{Rad}(G)k$

use Luks to construct SLP to $gh^{-1}k^{-1} \in \mathrm{Rad}(G)$

constructive membership in G of even char., no exceptional factors

as above, just use constructive recognition in $\mathrm{Soc}^*(G)/\mathrm{Rad}(G)$ level