# Rechnen in endlichen Matrixgruppen

Max Neunhöffer

Lehrstuhl D für Mathematik
RWTH Aachen

26.6.2007

# Matrixgruppen . . .

Es sei $\mathbb{F}_q$ der Körper mit $q$ Elementen und

$$\mathrm{GL}_n(\mathbb{F}_q) := \{M \in \mathbb{F}_q^{n\times n} \mid M \text{ invertierbar}\}$$

Gegeben: $M_1, \ldots, M_k \in \mathrm{GL}_n(\mathbb{F}_q)$

Dann erzeugen die $M_i$ eine Gruppe $G \leq \mathrm{GL}_n(\mathbb{F}_q)$.

Diese ist endlich, es ist $|\mathrm{GL}_n(\mathbb{F}_q)| = q^{n(n-1)/2} \prod_{i=1}^{n}(q^i - 1)$

## Was will man algorithmisch über $G$ herausfinden?

- Gruppenordnung $|G|$?
- Mitgliedschaftstest: Liegt $M \in \mathrm{GL}_n(\mathbb{F}_q)$ in $G$?
- Homomorphismen $\varphi : G \to H$?
- Kerne von Homomorphismen? Ist $G$ einfach?
- Vergleich mit bekannten Gruppen.
- (Maximale) Untergruppen?
- . . .

# Permutationsgruppen und Matrixgruppen

Es sei $n \in \mathbb{N}$ und $S_n$ die symmetrische Gruppe.

$$S_n = \{\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \mid \pi \text{ bijektiv}\}.$$

Gegeben: $\pi_1, \ldots, \pi_k \in S_n$

Dann erzeugen die $\pi_i$ eine Gruppe $G \leq S_n$.

Diese ist endlich, es ist $|S_n| = n!$

---

Es sei $\mathbb{F}_q$ der Körper mit $q$ Elementen und

$$\mathrm{GL}_n(\mathbb{F}_q) := \{M \in \mathbb{F}_q^{n \times n} \mid M \text{ invertierbar}\}$$

Gegeben: $M_1, \ldots, M_k \in \mathrm{GL}_n(\mathbb{F}_q)$

Dann erzeugen die $M_i$ eine Gruppe $G \leq \mathrm{GL}_n(\mathbb{F}_q)$.

Diese ist endlich, es ist $|\mathrm{GL}_n(\mathbb{F}_q)| = q^{n(n-1)/2} \prod_{i=1}^{n} (q^i - 1)$

# Permutationsgruppen

Es sei $n \in \mathbb{N}$ und $S_n$ die symmetrische Gruppe.

$$S_n = \{\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \mid \pi \text{ bijektiv}\}.$$

Gegeben: $\pi_1, \ldots, \pi_k \in S_n$

Dann erzeugen die $\pi_i$ eine Gruppe $G \leq S_n$.

Diese ist endlich, es ist $|S_n| = n!$

**Man kann z.B. algorithmisch über $G$ herausfinden:**

- Gruppenordnung $|G|$?
- Mitgliedschaftstest: Liegt $M \in S_n$ in $G$?
- Homomorphismen $\varphi : G \to H$?
- Kerne von Homomorphismen? Ist $G$ einfach?
- Vergleich mit bekannten Gruppen.
- (Maximale) Untergruppen?
- ...

# Konstruktive Erkennung — Vorformulierung

## Problem

Es seien $\mathbb{F}_q$ der Körper mit $q$ Elementen und

$$M_1, \ldots, M_k \in \mathrm{GL}_n(\mathbb{F}_q)$$

gegeben. Finde für $G := \langle M_1, \ldots, M_k \rangle$:

- Die Gruppenordnung $|G|$ und
- einen Algorithmus, der zu gegebenem $M \in \mathrm{GL}_n(\mathbb{F}_q)$
  - entscheidet, ob $M \in G$ ist und
  - wenn ja, $M$ als Produkt der $M_i$ schreibt.

Ist dieses Problem gelöst, dann nennen wir

$\langle M_1, \ldots, M_k \rangle$ konstruktiv erkannt.

# Komplexität von Algorithmen

Um die Effizienz eines Algorithmus zu messen, betrachten wir eine Klasse $\mathcal{P}$ von Problemen, die der Algorithmus lösen kann.

Wir ordnen jedem $P \in \mathcal{P}$ seine Größe $g(P)$ zu,

und beweisen eine Abschätzung für die Laufzeit $L(P)$ des Algorithmus für $P$:

$$L(P) \leq f(g(P))$$

für eine Funktion $f$.

Das Wachstumsverhalten von $f$ misst die Komplexität.

## Beispiel (Konstruktive Matrixgruppen-Erkennung)

- Problem gegeben durch $M_1, \ldots, M_k \in \mathrm{GL}_n(\mathbb{F}_q)$.
- Größe bestimmt durch $n$, $k$ und $\log q$.
- Laufzeit soll $\leq$ Polynom in $n$, $k$ und $\log q$ sein.

# Randomisierte Algorithmen

## Definition (Monte Carlo Algorithmus)

Ein Monte Carlo Algorithmus mit Fehlerwahrscheinlichkeit $\epsilon$ ist ein Algorithmus, der garantiert nach einer endlichen Zeit terminiert, so dass die Wahrscheinlichkeit, dass er ein falsches Ergebnis erzielt, höchstens $\epsilon$ ist.

## Definition (Las Vegas Algorithmus)

Ein Las Vegas Algorithmus mit Fehlerwahrscheinlichkeit $\epsilon$ ist ein Algorithmus, der garantiert nach einer endlichen Zeit terminiert, so dass die Wahrscheinlichkeit, dass er kein Ergebnis erzielt, höchstens $\epsilon$ ist.

Beispiel: Berechnung $|G| = 4\,089\,470\,473\,293\,004\,800$ für Permutationsgruppe $G = \langle \pi_1, \pi_2 \rangle$ ($n = 137\,632$):

deterministisch: 112s  Monte Carlo $\epsilon = 1\%$: 6s

Ersparnis: 95% der Rechenzeit

# Konstruktive Erkennung

## Problem

Es seien $\mathbb{F}_q$ der Körper mit $q$ Elementen und

$$M_1, \ldots, M_k \in \mathrm{GL}_n(\mathbb{F}_q)$$

gegeben. Finde für $G := \langle M_1, \ldots, M_k \rangle$:

- Die Gruppenordnung $|G|$ und
- einen Algorithmus, der zu gegebenem $M \in \mathrm{GL}_n(\mathbb{F}_q)$
  - entscheidet, ob $M \in G$ ist, und,
  - wenn ja, $M$ als Produkt der $M_i$ schreibt.
- Dabei soll die Laufzeit beschränkt sein durch ein Polynom in $n$, $k$ und $\log q$.
- Monte Carlo Algorithmus reicht aus. (Verifikation!)

Ist dieses Problem gelöst, dann nennen wir

$\langle M_1, \ldots, M_k \rangle$ konstruktiv erkannt.

# Schwierigkeiten

## Das diskrete Logarithmus-Problem

Ist $M_1 = [z] \in \mathbb{F}_q^{1\times 1}$ mit $z$ Primitivwurzel von $\mathbb{F}_q$. Dann:

Gegeben $0 \neq [x] \in \mathbb{F}_q^{1\times 1}$, finde $i \in \mathbb{N}$ mit $[x] = [z]^i$.

Es ist keine Lösung in polynomieller Zeit in $\log q$ bekannt!

## Ganzzahlige Faktorisierung

Manche Methoden brauchen eine Faktorisierung von $q^i - 1$ für ein $i \leq n$.

Es ist keine Lösung in polynomieller Zeit in $\log q$ bekannt!

In der Praxis ist $q$ klein $\Rightarrow$ kein Problem.

Wir ignorieren beides!

# Was ist eine Reduktion?

Es sei $G := \langle M_1, \ldots, M_k \rangle \leq \mathrm{GL}_n(\mathbb{F}_q)$.

Eine Reduktion ist ein Gruppenhomomorphismus

$$\begin{array}{rccll} \varphi & : & G & \to & H \\ & & M_i & \mapsto & P_i \qquad \text{für alle } i \end{array}$$

mit folgenden Eigenschaften:

- $\varphi(M)$ ist explizit berechenbar für alle $M \in G$
- $\varphi$ ist surjektiv: $H = \langle P_1, \ldots, P_k \rangle$
- $H$ ist in gewissem Sinne „kleiner“
- oder zumindest „leichter konstruktiv zu erkennen“
- z.B. $H \leq S_m$ oder $H \leq \mathrm{GL}_{n'}(\mathbb{F}_{q'})$ mit $n' \log q' < n \log q$

# Bestimmung des Kerns

Es sei $\varphi : G \to H$ Reduktion und $H$ konstruktiv erkannt.

Dann können wir den Kern $N$ von $\varphi$ bestimmen:

1. Erzeuge (pseudo-) zufälliges Element $M \in G$,
2. bilde mit $\varphi$ auf $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$ ab,
3. schreibe $\varphi(M)$ als Produkt der $P_i$,
4. evaluiere das gleiche Produkt in den $M_i$,
5. erhalte Element $M' \in G$ mit $M \cdot M'^{-1} \in N$.
6. $M$ gleichverteilt in $G \Rightarrow M \cdot M'^{-1}$ gleichverteilt in $N$

$\to$ Monte Carlo Algorithmus zur Bestimmung von $N$

# Erkennung von Bild und Kern löst Problem

Es sei $\varphi : G \to H$ Reduktion und sowohl $H$ als auch der Kern $N = \langle N_1, \ldots, N_m \rangle$ von $\varphi$ seien konstruktiv erkannt.

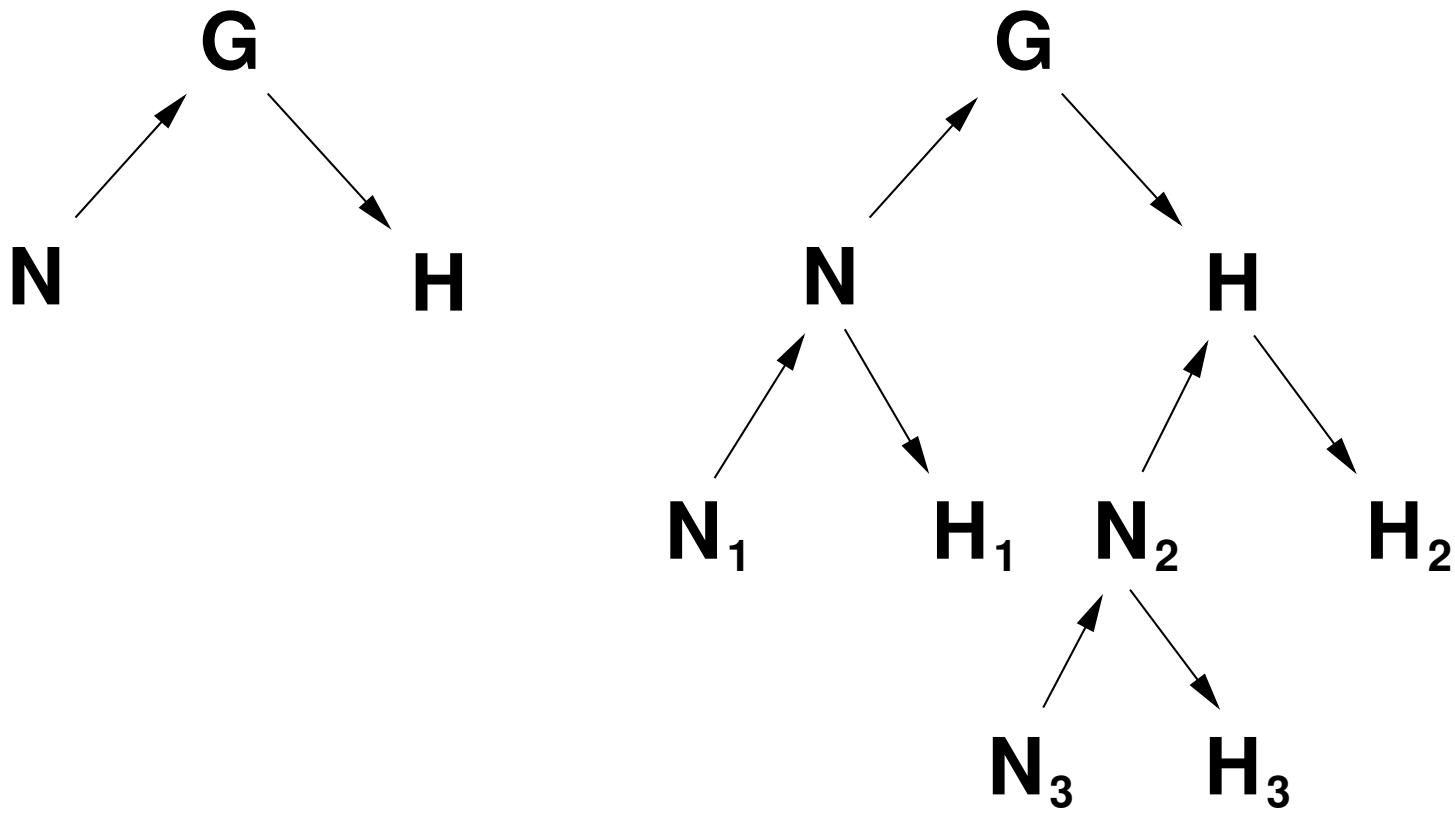
Dann haben wir $G$ konstruktiv erkannt:

$|G| = |H| \cdot |N|$. Und für $M \in \mathrm{GL}_n(\mathbb{F}_q)$:

1. bilde $M$ mit $\varphi$ auf $\varphi(M) \in H = \langle P_1, \ldots, P_k \rangle$ ab,
2. schreibe $\varphi(M)$ als Produkt der $P_i$,
3. evaluiere das gleiche Produkt in den $M_i$,
4. erhalte Element $M' \in G$ mit $M \cdot M'^{-1} \in N$,
5. schreibe $M \cdot M'^{-1}$ als Produkt der $N_j$,
6. erhalte $M$ als Produkt in den $M_i$ und den $N_j$:
   $M' = \prod$ in den $M_i$, $\quad M \cdot M'^{-1} = \prod$ in den $N_j$
   $\Rightarrow M = (\prod$ in den $N_j) \cdot (\prod$ in den $M_i)$.
7. Wenn $M \notin G$, dann funktioniert ein Schritt nicht.

# Rekursion: Kompositionsbäume

Wir erhalten einen Baum:




Aufwärts-Pfeile: Einbettungen

Abwärts-Pfeile: Homomorphismen

Alte Idee, wesentliche Verbesserungen: Seress & N. 2006

# Beispiel: Invarianter Teilraum

Es sei $V = \mathbb{F}_q^n$, dann operiert $G$ auf $V$.
Es sei $W \leq V$ ein invarianter Teilraum, das heißt:

$$MW = W \quad \text{für alle } M \in G$$

Wähle Basis $(w_1, \ldots, w_d)$ von $W$ und ergänze zu Basis

$$(w_1, \ldots, w_d, w_{d+1}, \ldots, w_n)$$

von $V$. Nach Basiswechsel sehen Matrizen in $G$ so aus:

$$\left[\begin{array}{c|c} A & B \\ \hline \mathbf{0} & D \end{array}\right] \quad \text{mit } A \in \mathbb{F}_q^{d\times d}, B \in \mathbb{F}_q^{d\times(n-d)}, D \in \mathbb{F}_q^{(n-d)\times(n-d)}$$

und

$$G \to \mathrm{GL}_{n-d}(\mathbb{F}_q), \left[\begin{array}{cc} A & B \\ \mathbf{0} & D \end{array}\right] \mapsto D$$

ist ein Homomorphismus von Gruppen.

# Beispiel: Invarianter Teilraum

$$G \to \mathrm{GL}_{n-d}(\mathbb{F}_q), \begin{bmatrix} A & B \\ \mathbf{0} & D \end{bmatrix} \mapsto D$$

ist ein Homomorphismus von Gruppen, sein Kern ist

$$N := \left\{ \begin{bmatrix} A & B \\ \mathbf{0} & D \end{bmatrix} \in G \mid D = \mathbf{1} \right\}.$$

Die Abbildung

$$N \to \mathrm{GL}_d(\mathbb{F}_q), \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mapsto A$$

ist auch ein Gruppenhomomorphismus und hat Kern

$$N_2 := \left\{ \begin{bmatrix} A & B \\ \mathbf{0} & D \end{bmatrix} \in G \mid A = D = \mathbf{1} \right\}.$$

Diese Gruppe ist eine $p$-Gruppe für $q = p^e$:

$$\begin{bmatrix} \mathbf{1} & B \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} & B' \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & B + B' \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Beim Reduzieren fallen Zusatzinformationen an!

# Wie findet man Reduktionen?

Aschbacher hat Klassen C1 bis C8 von Untergruppen von $\mathrm{GL}_n(\mathbb{F}_q)$ definiert.

## Theorem (Aschbacher, 1984)

*Es sei $G \leq \mathrm{GL}_n(\mathbb{F}_q)$ und $Z := G \cap Z(\mathrm{GL}_n(\mathbb{F}_q))$ die Untergruppe der Skalarmatrizen. Dann liegt $G$ in mindestens einer der Klassen C1 bis C8 oder es gilt:*

- *$T \subseteq G/Z \subseteq Aut(T)$ für eine nicht-abelsche, einfache Gruppe $T$, und*
- *$G$ operiert absolut irreduzibel auf $V = \mathbb{F}_q^n$.*

(Letzterer Fall wird C9 genannt.)

Also können wir schweres Geschütz auffahren:

- Die Klassifikation der endlichen einfachen Gruppen
- Die modulare Darstellungstheorie einfacher Gruppen

# Vorgehensweise für Blätter des Baums

Wenn keiner der Algorithmen für C1 bis C8 Erfolg hatte:

1. Bei „kleinen“ Gruppen direkten Isomorphismus auf Permutationsgruppe.
2. Erkenne, für welche (einfache) Gruppe $T$ gilt: $T \leq G/Z \leq \mathrm{Aut}(T)$.
3. Finde einen expliziten Isomorphismus auf eine „Standardform“ einer Zwischengruppe $S$.
4. Benutze schließlich Informationen über $S$, um $G$ konstruktiv zu erkennen.

Dies benutzt:

- Klassifikation der endlichen einfachen Gruppen
- Informationen über Automorphismengruppen
- Informationen über Elementordnungen
- Informationen über Konjugiertenklassen
- Klassifikation der irreduziblen Darstellungen
- Informationen über Untergruppenstruktur

# Nicht-konstruktive Erkennung

Methoden zur nicht-konstruktiven Erkennung:

- Wissen über Darstellungen schränkt Auswahl ein.
- Statistik von Ordnungen zufälliger Elemente (dazu Minimalpolynom wichtig, effizienterer Algorithmus Praeger & N. 2007)

In der Regel liefert das Monte Carlo Algorithmen.

# Standarderzeuger

In $G$ können wir nur multiplizieren, invertieren und die Ordnung berechnen.

Annahme: $G \cong S$ mit $T \leq S \leq \mathrm{Aut}(T)$ und $T$ einfach.

Finde ein Tupel $(s_1, \ldots, s_r) \in S^r$ zusammen mit gewissen Produkten $p_1, \ldots, p_m$ der $s_i$, so dass gilt:

- $S = \langle s_1, \ldots, s_r \rangle$,
- wenn $(s'_1, \ldots, s'_r) \in S^r$ ist mit
  - $|s_i| = |s'_i|$ für $1 \leq i \leq r$,
  - $|p_j| = |p'_j|$ für $1 \leq j \leq m$,
    ($p'_j$ die den $p_j$ entsprechenden Produkte der $s'_i$)

  dann definiert $s_i \mapsto s'_i$ für $1 \leq i \leq r$ einen Automorphismus von $S$.

Solche Elemente heißen „Standarderzeuger“ von $S$.

Man findet $G \cong S$ explizit, indem man in $G$ ein Tupel $(M_1, \ldots, M_r)$ von Standarderzeugern sucht.

Oft erhält man effiziente Las Vegas Algorithmen zum Auffinden expliziter Isomorphismen.

# Verifikation

Überall war Zufall im Spiel: Las Vegas und Monte Carlo.

$\Rightarrow$ Müssen testen, ob das Ergebnis auch stimmt!

Idee:

- Finde (kurze) Präsentationen für die Blatt-Gruppen,
- setze diese zu einer für die gesamte Gruppe zusammen.
- Rechne die Relationen nach und beweise so Ergebnis.

# Rück- und Überblick

- Das Matrixgruppen-Erkennungsproblem
- Möglichkeiten zur Reduktion $\longrightarrow$ Kompositionsbaum
- Lösungsansätze für die Blätter

Neuere Verbesserungen:

- Effiziente Implementation von Matrizen (N. (2007))
- Verbesserter Algorithmus zur Berechnung von Minimalpolynomen (N., Cheryl Praeger (2006)).
- Verbessertes Framework zur Implementation von konstruktiven Erkennungsalgorithmen für Gruppen (N., Ákos Seress (2006)).
- Neue Methoden für die Aschbacher-Klassen C3 und C5 (Jon F. Carlson, N., Colva Roney-Dougal (2007)).
- Neue Implementation der bekannten Matrixgruppen-Erkennungsalgorithmen in GAP (N., Ákos Seress et al. (2005–2007)).