Algorithmische Gruppentheorie und das GAP System

DMV Tagung Bonn,
September 2006

Bettina Eick
TU Braunschweig

GAP
Groups, Algorithms and Programming

- Software System zur Gruppentheorie
- Für Forschung und Lehre
- Erhältlich: www.gap-system.org (frei)

In diesem Vortrag:

- Überblick zum GAP System
- Aktuelle Forschungsthemen
- Neue Software in GAP

Geschichte des GAP Systems

1986 Projekt beginnt an der RWTH Aachen

1988 Version 2.4 – erstes Release von GAP

1991 Version 3.1 – wesentliche Überarbeitung

1997 GAP zieht von Aachen nach St. Andrews

1999 Version 4.1 – weitere neue Überarbeitung

2004 Gründung von ‘GAP Centers’ (Aachen, Braunschweig, Fort Collins, St. Andrews)

heute Aktuelle Version ist GAP 4.4

Einige weitere Features von GAP

- GAP Entwicklung:
  internationales Team von Leuten
- GAP Packages:
  extern entwickelte Packages, werden referiert
- GAP Webseiten:
  www.gap-system.org
- GAP Forum:
  e-mail Forum zur Gruppentheorie und GAP
- GAP Support:
  e-mail Hilfe zu technischen Fragen

Was kann GAP

- Programmiersprache
- Viele Funktionen zur Untersuchung und Konstruktion von Gruppen
- Viele Datenbanken von Gruppen, z.B.:
  - Gruppen der Ordnung ≤ 2000 (Besche, Eick, O'Brien)
  - Transitive Gruppen vom Grad ≤ 30 (Hulpke)
  - Perfekte Gruppen der Ordnung $\leq 10^6$ (Holt, Plesken), (Smith)
  - Maximal endliche $G \leq GL(n, Z)$, $n \leq 31$ (Plesken, Souvignier, Nebe, Felsch)
  - ....

Untersuchung von Gruppen

Algorithmen hängen von der Darstellung der gegebenen Gruppen ab.

'Klassische' Darstellungen:

- Matrixgruppen
- Permutationsgruppen
- Endlich präsentierte Gruppen
- Polyzyklisch präsentierte Gruppen

Neue Ideen:

- Automaten definieren Gruppen
- Unendliche Präsentationen
- Untergruppen vom $Sym(\infty)$

Matrixgruppen

$G \leq GL(d, K)$ durch Erzeuger gegeben:

K endlicher Körper:

- 'Matrix Group Recognition Project'
- Grosses Forschungsthema in CGT
- Ziel: Berechne

$$G = G_1 \rhd G_2 \rhd \ldots \rhd G_n \rhd G_{n+1} = \{1\}$$

  mit G_i/G_{i+1} einfach und identifiziere G_i/G_{i+1}
- Methode: Satz von Aschbacher und Rechnen mit endlichen einfachen Gruppen
- Rechnen mit endlichen einfachen Gruppen ist derzeit Hauptproblem

Matrixgruppen II

$G \leq GL(d, K)$ durch Erzeuger gegeben:

K unendlicher Körper, z.B. $K = \mathbb{Q}$:

- Tits Alternative: G ist virtuell auflösbar oder enthält eine nicht-abelsche freie Untergruppe
- Tits Alternative kann entschieden werden
- Mit virtuell auflösbaren Gruppen kann man durch ihre Struktur etwas anfangen
- Mit nicht-abelschen freien Matrixgruppen kann man sehr wenig anfangen

Viele elementare Fragen sind hier noch offen:
z.B. Bestimme einen irreduziblen Teilmodul

$$W \leq V = K^d$$

Permutationsgruppen

$G \leq Sym(n)$:

n endlich:

- Viele effiziente Algorithmen vorhanden
- Kaum neuere Entwicklungen

$n = \infty$:

- 'Residue Class-Wise Affine Groups'
  RCWA Package (Kohl)
- Hat zur Entdeckung neuer unendlicher einfacher Gruppen geführt
- Mit diesen einfachen Gruppen kann man in GAP rechnen

Permutationsgruppen II

Anwendungen von Permutationsgruppen:

- Design Package (Soicher):
  Konstruktion und Klassifikation von Blockdesigns
- Grape Package (Soicher):
  Konstruktion und Untersuchung von Graphen

Endlich präsentierte Gruppen

- Viele Probleme sind allgemein nicht entscheidbar
- Trotzdem sind viele Algorithmen erhältlich
- Neue Entwicklung von Quotientenalgorithmen (Eick, Linton, Niemeyer)
- Neue Entwicklung von Matrixdarstellungen (Plesken, Robertz)

Unendlich präsentierte Gruppen

Beispiel: Gupta-Sidki-Gruppe

- Operiert auf einem unendlichen Graphen
- War eines der ersten Gegenbeispiele zum Burnsideproblem
- Endlich erzeugt, aber nicht endlich präsentiert

Rekursiv präsentierte Gruppen:

- haben eine Präsentation auf endlich vielen Erzeugern mit Relationen der Form

  $$R_1, \ldots, R_l, \sigma^i(R_{l+1}), \ldots, \sigma^i(R_{l+m}) \quad (i \in \mathbb{N}_0)$$

  σ Endomorphismus der freien Gruppe
- Algorithmen zum Rechnen mit solchen rekursiven Präsentation werden entwickelt (Bartholdi, Eick)
- Anwendungen: z.B. die Gupta-Sidki-Gruppe und viele Verwandte

Spezielle Klassen endlich präsentierter Gruppen

Polyzyklischen Gruppen:

- Viele effektive Algorithmen vorhanden
- Offene Fragen für unendliche polyzyklische G:
- Berechne $Aut(G)$
  (Grunewald, Segal: entscheidbar, aber kein praktikabler Algorithmus)
- Berechne $d(G)$ die minimale Erzeugeranzahl

Hybridgruppen

G ist Hybridgruppe, falls

- $N \trianglelefteq G$ mit N auflösbar
- $G/N \to Sym(n)$ oder $G/N \to GL(d, \mathbb{F}_p)$

Ziel: Algorithmen für Hybridgruppen,
die die gegebene Struktur ausnutzen

Beispiele:

- Automorphismengruppen von p-Gruppen
- Untergruppen von $GL(d, \mathbb{Z}/p^n\mathbb{Z})$.

Solche Methoden werden derzeit entwickelt
(Hulpke, Eick)

Kohomologie

Zwei neue GAP Packages:

- HAP (Ellis)
- crime (Bishop)

HAP:

- Kann $H^n(G, M)$ berechnen für gegebenes n
- G kann endlich sein, oder abelsch-mal-endlich (Raumgruppen) oder endlich erzeugt nilpotent
- M ist G-Modul endlicher Dimension

crime:

- Kann Kohomologieringe $H^*(G, \mathbb{F}_p)$ berechnen
- G ist endliche p-Gruppe

Automaten und Gruppen

Zwei GAP Packages:

- Automata (Delgado, Linton, Morais)
- FGA (Sievers)

Anwendungen von FGA:

- Rechnen mit Untergruppen von freien Gruppen
- Bestimme Rang von $G \leq F_n$
- Bestimme Normalisator von $G \leq F_n$

Gruppen und Algebren

Drei GAP Packages:

- Laguna (Bovdi, Konovalov, Rossmanith, Schneider)
- Sophus (Schneider)
- UnitLib (Konovalov, Yakimenko)

Laguna und UnitLib:

- Rechnen mit modularen Gruppenalgebren
- Berechnung der Einheitengruppe
- Datenbank solcher Einheitengruppen
- Anwendung: das modulare Isomorphieproblem

Gruppen und Algebren II

Sophus:

- Konstruktion und Klassifikation nilpotenter Lie Algebren bis zur Dimension n über $\mathbb{F}_q$
- Anwendung: Beitrag zur Klassifikation von Lie Algebren