

# New Ideas In The Enumeration Of Coherent Subalgebras Of A Prescribed Coherent Algebra

Sven Reichard

School of Mathematics and Statistics  
University of Western Australia

May 6, 2006  
Special Semester on Gröbner Bases  
Linz

# Outline

- 1 The Problem
- 2 The Traditional Algorithm
- 3 New Ideas

# Outline

- 1 The Problem
- 2 The Traditional Algorithm
- 3 New Ideas

# Coherent Algebras

A *coherent algebra*  $W$  is a matrix algebra over  $\mathbb{C}$  such that

- $W$  has a basis  $\mathcal{A} = \{A_1, \dots, A_r\}$  of  $(0, 1)$ -matrices (standard basis);
- $W$  contains  $I$ , the identity matrix;
- $\sum_{i=1}^r A_i = J$ , the all-one matrix;
- $W$  is closed under transposition.

# Sources of coherent algebras

- Centralizer rings of permutation groups
- Strongly regular graphs
- Distance regular graphs
- Partial Geometries
- Maximal arcs in projective planes
- ...

# Coherent subalgebras

- A **merging** (or fusion) of  $W$  is a coherent subalgebra of  $W$ .
- Thus, its standard basis consists of sums of distinct basis elements of  $W$ .
- A merging is uniquely determined by a partition of the basis  $\mathcal{A}$ .

# The problem

## Problem

*Given a coherent algebra  $W$ , determine all mergings*

The solution helps, e.g., to find all distance regular graphs invariant under a given permutation group

# A running example

## Example

Consider the Schur ring  $W$  over the cyclic group  $C_8$

We have elements  $\underline{0}, \underline{1}, \dots, \underline{7}$  in the group ring

Multiplying two elements is straightforward, e.g.,

$$\underline{2} \cdot \underline{3} = \underline{5}$$

More complicated:

$$\begin{aligned} \underline{2,4} \cdot \underline{4,6} &= \underline{6} + \underline{0} + \underline{0} + \underline{2} \\ &= 2 \cdot \underline{0} + \underline{2,6} \end{aligned}$$

# Outline

- 1 The Problem
- 2 The Traditional Algorithm
- 3 New Ideas

# COCO

- Computer package COCO
- Available for DOS and Unix
- Very efficient for certain tasks

# History of COCO

- 1979-80 (I. Faradžev, M. Klin, V.A. Zaichenko)
- 1981-84 (Faradžev, A.A.Ivanov, Klin.)  
Primitive representations of simple groups  
of order up to  $10^6$
- 1984-89 (A.V.Ivanov + above) Sporadic computations
- 1990-92 (Faradžev, Klin, D. Pasechnik) Implementation for  
PC
- 1992 (Brouwer) Unix port
- 1996-98 (Faradžev, F. Fiedler, Ch. Pech, Klin)  
Slight improvements

# Outline Of The Algorithm

Mergings (partitions) are constructed as follows:

- Find sets that can form classes in the mergings (“good sets”).
- Find partitions of the basis which consist of good sets.
- Check each to see whether it gives a merging.

## Finding good sets

- Subsets of the basis are considered, of size up to half the rank.
- Only consider symmetrical or antisymmetrical sets
- Check that the set isn't subdivided by squaring

## Example

In  $W$ , take the set  $\{1, 2\}$ . We compute the square:

$$\underline{1, 2} \cdot \underline{1, 2} = \underline{2, 4} + 2 \cdot \underline{3}$$

2 occurs once, 1 doesn't occur  $\longrightarrow$  not good! Other set:

$$\underline{2, 4, 6} \cdot \underline{2, 4, 6} = 3 \cdot \underline{0} + 2 \cdot \underline{2, 4, 6}$$

Here, each of 2, 4, 6 occurs exactly twice  $\longrightarrow$  good!

# Constructing mergings from good sets

- Construct partitions from good sets
- Check which are closed under multiplication and transposition

## Example

Let  $S_1 = \underline{0}$ ,  $S_2 = \underline{1, 3, 5, 7}$ ,  $S_3 = \underline{2, 4, 6}$ . Then, e.g.,

$$S_2 \cdot S_2 = 4 \cdot S_1 + 4 \cdot S_3$$

$$S_2 \cdot S_3 = 3 \cdot S_2$$

In fact, this is a merging

# Limitations of COCO

- Only constructs homogeneous mergings
- Does not exploit symmetries
- Restriction to 100 good subsets

## Example

- We have  $(\underline{3}, \underline{5})^2 = 2 \cdot \underline{0} + \underline{2} + \underline{6}$
- Moreover,  $(\underline{1})^2 = \underline{2}$ , so both sets are good sets.
- So, we would consider all partitions containing both sets...

# Outline

- 1 The Problem
- 2 The Traditional Algorithm
- 3 New Ideas**

# Bannai-Muzychuk method

- Instead of tensor, consider character table
- Work with quadratic instead of cubic information
- Partition of rows induces partition of colors; sizes have to coincide
- Only applicable in the commutative case!
- How do we find the eigenvalues?

## Finding eigenvalues

- In the commutative case the algebra is diagonalizable
- Irreducible representations are one-dimensional
- Recall that

$$A_i \cdot A_j = \sum p_{ij}^k A_k$$

- Get quadratic equation for eigenvalues: For all  $i \leq j$ ,

$$x_i x_j - \sum p_{ij}^k x_k = 0$$

- Solve using Gröbner bases

## Example

Take the two sets 1 and 3, 5. We have

$$\begin{aligned}\underline{1}^2 &= \underline{2} \\ \underline{1} \cdot \underline{2} &= \underline{3}\end{aligned}$$

Thus, 3, 5 is split  $\longrightarrow$  no merging containing both sets!  
In fact, 1 generates all of  $W$

# Stabilization

- Problem: Find the coarsest merging containing a given partition
- Goes back to Weisfeiler-Leman
- Idea: Multiply any two classes; refine as necessary
- If no further refinement occurs, we are done
- Complexity?

## Example

Taking the set 3, 5, we get the partition

0; 3, 5; 1, 2, 4, 6, 7

Stabilization gives

0; 3, 5; 1, 7; 2, 6; 4

## Using groups

- Interested in mergings up to isomorphism
- Sufficient to determine up to  $CAut(W)$
- In fact, mergings depend only on tensor, hence use  $AAut(W)$
- Compute groups using partition backtracking (Leon)

## Example

In  $W$ ,  $(1, 7)(2, 6)(3, 5)$  is an algebraic automorphism

# Set orbit enumeration

- For good sets, want to check only one representative from each orbit
- Reduced to a canonicity check
- Can check (rather) efficiently if a set is smallest in its orbit
- Enumerate orbits depth-first

# Independent sets

- Idea: Remove redundancy in the classical partition enumeration
- A generating set is independent if no element can be omitted
- Lemma: Any generating set contains an independent set
- Enumerate independent sets instead of partitions

# An algorithm

- Find “real good” sets up to isomorphism under  $\text{AAut}(W)$
- Incrementally construct partial partitions
- Stabilize, check independence
- Check that there is no “smaller” generating set
- In each step, get a new merging

## Example ( $E_{32}$ )

- Consider regular action of elementary abelian group  $E_{32}$
- Take its centralizer ring (“Schur ring”)
- $AAut = CAut = SL(5, 2)$ , size  $10^7$
- 614,482 good sets in 31 orbits
- Found 402 mergings (up to alg. isomorphisms)  
in three hours

# Summary

- We are considering the **enumeration of mergings**
- One important ingredient: **Stabilization**
- Other ingredient: **Symmetry**
  
- Implementation in COCO-II,  
a GAP package due in Spring '07