A fast coset-translation algorithm for computing the cycle structure of Comer relation algebras over \( \mathbb{Z}/p\mathbb{Z} \)

Jeremy F. Alm Andrew Ylvisaker

August 17, 2017

Abstract

Proper relation algebras can be constructed using \( \mathbb{Z}/p\mathbb{Z} \) as a base set using a method due to Comer. The cycle structure of such an algebra must, in general, be determined \textit{a posteriori}, normally with the aid of a computer. In this paper, we give an improved algorithm for checking the cycle structure that reduces the time complexity from \( O(p^2) \) to \( O(p) \).

1 Introduction

Comer [4] introduced a technique for constructing finite integral proper relation algebras using \( \mathbb{Z}/p\mathbb{Z} \) as a base set for \( p \) prime. Set \( p = nk + 1 \). Then there is a multiplicative subgroup \( H < (\mathbb{Z}/p\mathbb{Z})^\times \) of size \( k \) and index \( n \), and the subgroup \( H \) can be used to construct a proper relation algebra of order \( 2^n+1 \). Specifically, fix \( m \in \mathbb{Z}^+ \), and let \( X_0 = H \) be the unique multiplicative subgroup of \((\mathbb{Z}/p\mathbb{Z})^\times \) of order \( k \). Let \( X_1, \ldots, X_{n-1} \) be its cosets; in particular, let \( X_i = g^i \cdot X_0 = \{g^{an+i} : a \in \mathbb{Z}^+\} \), where \( g \) is a primitive root modulo \( p \), i.e., \( g \) is a generator of \((\mathbb{Z}/p\mathbb{Z})^\times \). Then define relations

\[ R_i = \{(x, y) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} : x - y \in X_i\} \]

The \( R_i \)'s, along with \( \text{Id} = \{(x, x) : x \in \mathbb{Z}/p\mathbb{Z}\} \), partition the set \( \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} \). The \( R_i \)'s will be symmetric if \( k \) is even and asymmetric otherwise.

Given a prime \( p \) and integer \( n \) a divisor of \( p-1 \), it is difficult in general to say much about the cycle structure of the algebra generated by the \( R_i \)'s (although it is shown in [1] that if \( p > m^4 + 5 \) then the cycles \((i, i, i)\) are mandatory); one must use a computational approach. The cycle \((R_i, R_j, R_k)\) is forbidden just in case \( X_i + X_j \cap X_k = \emptyset \). Naively, one computes all the sumsets \( X_i + X_j \), though because of rotational symmetry, one may assume \( i = 0 \):

**Lemma 1.** Let \( n \in \mathbb{Z}^+ \) and let \( p = nk + 1 \) be a prime number and \( g \) a primitive root modulo \( p \).
For \( i \in \{0, 1, \ldots, n-1\} \), define
\[
X_i = \{ g^i, g^{n+i}, g^{2n+i}, \ldots, g^{(k-1)n+i} \}.
\]

Then \((X_0 + X_j) \cap X_k = \emptyset\) if and only if \((X_i + X_{i+j}) \cap X_{i+k} = \emptyset\).

The lemma is trivial to prove: just multiply through by \( g^i \).

Comer used his technique to construct (representations of) symmetric algebras with \( n \) diversity atoms forbidding exactly the 1-cycles (i.e., the “monochrome triangles”) for \( 1 \leq n \leq 5 \). Comer did the computations by hand. Comer’s work was extended in \([5]\) \((n \leq 7)\), \([3]\) \((n \leq 400, n \neq 8, 13)\), and \([1]\) \((401 \leq n \leq 2000)\).

This last significant advance was made possible by a much less general version of the improved algorithm presented here. Another variation was used in \([2]\) to construct representations of algebras in which all the diversity atoms are flexible.

## 2 Symmetric Comer algebras

The following naive algorithm, used in \([3]\) to find Ramsey algebras for \( n \leq 400, n \neq 8, 13 \), computes all the sumsets \(X_0 + X_i\).

**Data:** A prime \( p \), a divisor \( n \) of \((p-1)/2\) such that \((p-1)/n\) is even, a primitive root \( g \) modulo \( p \).

**Result:** a list of mandatory and forbidden cycles of the form \((0, x, y)\).

```
Algorithm 1: Naive algorithm for symmetric Comer algebras
```

```
Data: A prime \( p \), a divisor \( n \) of \((p-1)/2\) such that \((p-1)/n\) is even, a primitive root \( g \) modulo \( p \).
Result: a list of mandatory and forbidden cycles of the form \((0, x, y)\).

Compute \(X_0 = \{g^{an} \pmod{p} : 0 \leq a < (p-1)/n\}\);
for \( i \leftarrow 0 \) to \( n-1 \) do
    \(X_i = \{g^{an+i} \pmod{p} : 0 \leq a < (p-1)/n\}\);
    Compute \(X_0 + X_i \pmod{p}\);
    for \( j \leftarrow i \) to \( n-1 \) do
        Compute \(X_j = \{g^{an+j} \pmod{p} : 0 \leq a < (p-1)/n\}\);
        if \((X_0 + X_i) \supseteq X_j\) then
            add \((0, i, j)\) to list of mandatory cycles
        else
            add \((0, i, j)\) to list of forbidden cycles
    end
end
```

The following lemma is very easy to prove and was apparently known to Comer. The algorithmic speed-up it provides, however, was not previously noticed.

**Lemma 2.** Let \( n \in \mathbb{Z}^+ \) and let \( p = nk+1 \) be a prime number, \( k \) even, and \( g \) a primitive root modulo \( p \). For \( i \in \{0, 1, \ldots, n-1\} \), define
\[
X_i = \{ g^i, g^{n+i}, g^{2n+i}, \ldots, g^{(k-1)n+i} \}.
\]

Then if \((X_0 + X_i) \cap X_j \neq \emptyset\), then \((X_0 + X_i) \supseteq X_j\).
Corollary 3. A diversity cycle \((X_0, X_i, X_j)\) is forbidden if and only if \((g^j - X_0) \cap X_i = \emptyset\).

Corollary 3 affords us the following faster algorithm for computing the cycle structure of Comer relation algebras.

Data: A prime \(p\), a divisor \(n\) of \((p - 1)/2\) such that \((p - 1)/n\) is even, a primitive root \(g\) modulo \(p\).

Result: a list of mandatory and forbidden cycles of the form \((0, x, y)\)

Compute \(X_0 = \{g^{an}\pmod{p} : 0 \leq a < (p - 1)/n\}\);

Compute \(g^j - X_0 \pmod{p}\) for each \(0 \leq j < n\);

for \(i \leftarrow 0\) to \(n - 1\) do
  \(X_i = \{g^{an+i}\pmod{p} : 0 \leq a < (p - 1)/n\}\)
  for \(j \leftarrow i\) to \(n - 1\) do
    if \((g^j - X_0) \cap X_i \neq \emptyset\) then
      add \((0, i, j)\) to list of mandatory cycles
    else
      add \((0, i, j)\) to list of forbidden cycles
  end
end

Algorithm 2: Fast algorithm for symmetric Comer algebras

Theorem 4. Algorithm 1 runs in \(O(p^2)\) time while Algorithm 2 runs in \(O(p)\) time, for \(n\) fixed.

Proof. In Algorithm 1, each pass through the inner loop requires \(O(k)\) comparisons, and each pass through the outer loop requires \(O(k^2)\) additions. So overall, \(O(nk^2)\) additions and \(O(n^2k)\) comparisons are required, for an overall runtime of \(O(p^2)\) for \(n\) fixed.

In Algorithm 2, each pass through the inner loop requires \(O(k)\) additions and \(O(k)\) comparisons. So overall, \(O(n^2k)\) additions and \(O(n^2k)\) comparisons are required, for an overall runtime of \(O(p)\) for \(n\) fixed.

Algorithm 2 is also much faster in practice. Both algorithms were implemented by the first author in Python 2.7. Timing data were collected for primes \(p \equiv 1 \pmod{23}\) under 20,000 on a 2012 HP Folio laptop. See Figure 1. The quadratic nature of Algorithm 1 is evident.
3 Asymmetric Corner algebras

For the case of asymmetric algebras, we need to take a little more care in our enumeration over indices \( i, j \) in checking whether \((X_0 + X_i) \supseteq X_j\). Let \( n \) be even, where \( n = 2m \). Since \(-X_i = X_{i+m}\), where all indices are computed mod \( n \), we have the following equivalence:

\[
(X_0 + X_i) \supseteq X_j \iff (X_0 + X_{j+m}) \supseteq X_{i+m}.
\]  

Thus for every triple \((0, i, j)\) of indices, there is an equivalent triple \((0, j + m, i + m)\) that would be redundant to check. So consider the involution \((0, i, j) \mapsto (0, j + m, i + m)\) on triples of indices. The fixed points of this involution are of the form \((0, i, i + m)\). (Of course, we continue to compute indices mod \( n \).) Consider an \( n \times n \) matrix \( A \) where the entry \( A_{ij} = (0, i, j + m) \). For example, see the matrix below, where \( n = 6 \):

\[
\begin{bmatrix}
(0, 0, 3) & (0, 0, 4) & (0, 0, 5) & (0, 0, 0) & (0, 0, 1) & (0, 0, 2) \\
(0, 1, 3) & (0, 1, 4) & (0, 1, 5) & (0, 1, 0) & (0, 1, 1) & (0, 1, 2) \\
(0, 2, 3) & (0, 2, 4) & (0, 2, 5) & (0, 2, 0) & (0, 2, 1) & (0, 2, 2) \\
(0, 3, 3) & (0, 3, 4) & (0, 3, 5) & (0, 3, 0) & (0, 3, 1) & (0, 3, 2) \\
(0, 4, 3) & (0, 4, 4) & (0, 4, 5) & (0, 4, 0) & (0, 4, 1) & (0, 4, 2) \\
(0, 5, 3) & (0, 5, 4) & (0, 5, 5) & (0, 5, 0) & (0, 5, 1) & (0, 5, 2)
\end{bmatrix}
\]

Then the fixed points of the involution are on the diagonal, and \( A_{ij} \) is equivalent to \( A_{ji} \) by the equivalence \((1)\). Thus it suffices to enumerate over the “upper triangle” of the matrix.
In Algorithm 3 below, the enumeration is done according to the discussion in the previous paragraph.

**Data:** A prime $p$, a divisor $n$ of $(p - 1)/2$ such that $(p - 1)/n$ is odd, a primitive root $g$ modulo $p$

**Result:** a list of mandatory and forbidden cycles of the form $(0, x, y)$

Let $m = n/2$. Compute $X_0 = \{g^{an} \pmod{p} : 0 \leq a < (p - 1)/n\}$; Compute $g^j - X_0 \pmod{p}$ for each $0 \leq j < n$;

for $i \leftarrow 0$ to $n - 1$ do

\[ X_i = \{g^{an+i} \pmod{p} : 0 \leq a < (p - 1)/n\} \]

for $j \leftarrow i + m \pmod{n}$ to $n + m - 1 \pmod{n}$ do

\[ \text{if } (g^j - X_0) \cap X_i \neq \emptyset \text{ then} \]

\[ \text{add } (0, i, j) \text{ to list of mandatory cycles} \]

\[ \text{else} \]

\[ \text{add } (0, i, j) \text{ to list of forbidden cycles} \]

end

end

**Algorithm 3:** Fast algorithm for asymmetric Comer algebras

**References**

[1] Jeremy F. Alm. 401 and beyond: improved bounds and algorithms for the Ramsey algebra search. *arXiv preprint arXiv:1609.01817*, 2016.

[2] Jeremy F. Alm, David Andrews, and Jacob Manske. Relation-algebraic sumset problems in abelian groups, part I: cyclic groups. *In Preparation*.

[3] Jeremy F. Alm and Jacob Manske. Sum-free cyclic multi-bases and constructions of Ramsey algebras. *Discrete Appl. Math.*, 180:204–212, 2015.

[4] S. D. Comer. Color schemes forbidding monochrome triangles. In *Proceedings of the fourteenth Southeastern conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1983)*, volume 39, pages 231–236, 1983.

[5] R. Maddux. Do all the Ramsey algebras exist? Presented at the AMS sectional meeting in Iowa City on March 18, 2011.