The Mathematics of Four-Note Chords and Beyond

By James Hober

Before beginning this chapter, make sure you've read and understood *The 43 Four-Note Qualities*. Here I present my experiments and research into the mathematics of four-note chords and beyond. It's quite technical so it's fine if you skip this chapter. Really! Please only read this if you like math, perhaps know a little algebra, and are curious about where the number 43 comes from. You'll also learn how many three-note, five-note, six-note, etc. qualities there are.

The field of mathematics that is relevant to our investigation is Enumerative Combinatorics. It's a vast area of study and I have only learned some basics. Part of what it deals with is counting combinations and permutations. Both combinations and permutations are ways of listing a few things out of a larger group of things. The items listed are called elements. In our case, the elements listed will be numbers from 1 to 9. With combinations, the order of the elements doesn't matter. With permutations, the order of the elements matters. Since order does matter when we represent four-note chords, we'll be using permutations.

There are two kinds of permutations, those without repetition and those with repetition. If an element can repeat, you have a permutation with repetition. 1 - 3 - 4 - 2 is a permutation without repetition. It has four elements and no element is a repetition of a previous element. 2 - 2 - 1 - 3 is a permutation with repetition because the second element repeats the first element. Permutations with repetition aren't **required** to repeat. "With repetition" means that it is allowed to happen, not that it must. We'll be looking at permutations with repetition for our investigation of V-System chords because our elements represent the number of half steps between chord tones, and they can repeat.

The values that each element of our permutations can take will range from 1 to 9 inclusive. One is the smallest number of half steps possible between chord tones of a close spaced (V-1) four-note chord. It can't be zero because that would result in doubling. Nine is the largest value an element can take because our four elements must sum to 12, the number of half steps in an octave. 1 - 1 - 1 - 9 is the only permutation of the 43 that contains a 9.

When the number of elements is less than the number of possible values, the permutation is more correctly called a k-permutation, where k is the number of elements. Since our permutations will have four elements with nine possible values that can repeat, we'll be looking at 4-permutations of nine objects with repetition. But I'll just call them "permutations" for short and less strictly. Therefore we'll use permutations of the form: a - b - c - d, where a, b, c, and d can be any integer value between 1 and 9 inclusive. The dashes are just separators in our notation.

Without adding any further constraints, how many possible permutations of this form are there? 6561. That's nine to the fourth power. The number of 4-permutations of nine objects with repetition is 6561.

We'll add one constraint to reduce the 6561 permutations to 165. Later, we'll add another constraint to further reduce the 165 to 43.

Investigating the 165 Permutations

The first constraint is that the elements of each permutation must sum to 12, the number of half steps in an octave. This eliminates permutations like 9 - 9 - 9 - 9 and 9 - 8 - 7 - 6 and gets us down to 165 permutations. I wrote a simple computer program to count and list the 165. The relevant computer code is shown at the end of this chapter. I won't list all 165 permutations here because it would take up too much space. Here's the beginning and end of the list:

The list of 165 permutations begins:

1 - 1 - 1 - 9
1 - 1 - 2 - 8
1 - 1 - 3 - 7
1 - 1 - 4 - 6
1 - 1 - 5 - 5
1 - 1 - 6 - 4
1 - 1 - 7 - 3
1 - 1 - 8 - 2
1 - 1 - 9 - 1
1 - 2 - 1 - 8
1 - 2 - 2 - 7
...

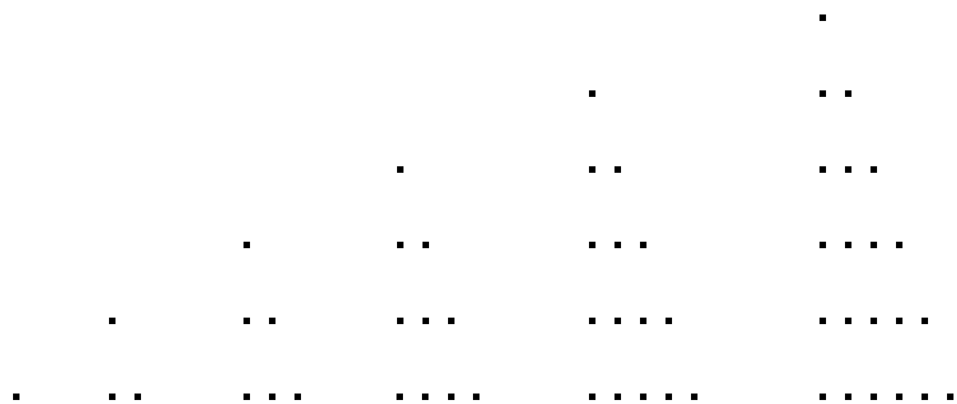
And the list of 165 permutations ends:

...
7 - 1 - 1 - 3
7 - 1 - 2 - 2
7 - 1 - 3 - 1
7 - 2 - 1 - 2
7 - 2 - 2 - 1
7 - 3 - 1 - 1
8 - 1 - 1 - 2
8 - 1 - 2 - 1
8 - 2 - 1 - 1
9 - 1 - 1 - 1

Even with this partial list, you can notice an interesting pattern. There's only one permutation that begins with a nine. That's the last one, 9 - 1 - 1 - 1. There are three that begin with an eight. They're near the end of the list, too. Here's a table I made from the complete list:

First Element	Number of Permutations
9	1
8	3
7	6
6	10
5	15
4	21
3	28
2	36
1	45

If you add up the second column, you get 165 total permutations. The sequence 1, 3, 6, 10, 15, 21, 28, 36, 45... is known as the triangular numbers. It's an important sequence in Combinatorics. They're called triangular numbers because if you build equilateral triangles with n dots on a side, the number of dots in the triangle will be the nth triangular number:



Another important sequence in Combinatorics is the tetrahedral numbers: 1, 4, 10, 20, 35, 56, 84, 120, 165, 220…. The tetrahedral numbers can be derived from the triangular numbers. Both series begin with 1. Add the first two triangular numbers to get the second tetrahedral number. Add the first three triangular numbers to get the third tetrahedral number. Add the first four triangular numbers to get the fourth tetrahedral number. And so on.

165 is the ninth tetrahedral number. The elements of our permutation can take on nine values. This is not a coincidence. Tetrahedral numbers are related to tetrahedrons that have four triangular sides. Our permutations have four elements. Again, this is not a coincidence.

We have found clues to the significance of the number 165 in the triangular numbers and the tetrahedral numbers. We can't discuss triangular numbers and tetrahedral numbers for long without mentioning Pascal's Triangle. What is Pascal's Triangle? Glad you asked.

Pascal's Triangle

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

Pascal's Triangle turns up in various Combinatorics problems. Its construction is simple: Each number in the triangle is the sum of the two numbers immediately above it. (If there's only one number above, it's a 1 and you use that number.)

Of interest to us are the diagonals. You can go diagonally down either to the left or to the right. It doesn't matter. The first diagonal is just an endless list of ones. The second diagonal is the counting numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9…. The third diagonal is the triangular numbers: 1, 3, 6, 10, 15, 21, 28, 36, 45, 55…. The fourth diagonal is the tetrahedral numbers: 1, 4, 10, 20, 35, 56, 84, 120, 165, 220…. Amazing! Here in Pascal's Triangle are both the triangular numbers and the tetrahedral numbers. Pascal's Triangle contains our number 165! It's in the fourth diagonal and our permutations have four elements. It's the ninth member of that diagonal and our permutations can take on nine values.

I could only fit the first eight rows of Pascal's Triangle above. The number 165 shows up in the twelfth row. Actually, mathematicians count the rows starting at 0. So 165 is in row 11 according to mathematicians. This row contains twelve numbers and there are twelve half steps in an octave. This row is important to us! The numbers in this row of Pascal's Triangle are:

1, 11, 55, 165, 330, 462, 462, 330, 165, 55, 11, 1

There's a formula to calculate any number in Pascal's Triangle: the binomial coefficient formula. It uses the factorial sign, which is an exclamation point. The formula is:

$n!/k!\,(n-k)!$ where n = the row counting from 0 and k = the member counting from 0.

We're looking at the 11th row counting from 0. 165 is the third member of the row counting from 0. So if we set $n = 11$ and $k = 3$ in the formula, we get 165.

Remember that we arrived at the number 165 because I programmed the computer to list every 4-permutation with repetition where the four elements summed to 12. What if I modified that program to list instead 3-permutations with repetition (for example, 1 - 1 - 10)? The 3-permutations would represent three-note chords and the three elements would still sum to 12, the number of half steps in an octave. I did that experiment and got a list of 55 three-note chords. Then I did it for two-note chords (intervals) and got 11. (Since we disallow doubling, the unison interval is not counted.) Then I did it for five-note chords and got 330. For six-note chords I got 462. In other words, the number of chords I found exactly matched the corresponding number in this row of Pascal's Triangle! There's a beautiful one-to-one correspondence between the number of x-note chords and the xth member of this row of Pascal's Triangle (counting from 1).

But we have not yet applied our second constraint.

The Second Constraint

To reduce our 165 down to 43, we must apply a second constraint: removing inversions. A maj7 is a single quality whether it has the root, third, fifth, or seventh in the bass. If we write the number of half steps between each note of a Cmaj7, we get 4 - 3 - 4 - 1 for the root position chord. The first, second, and third inversions, respectively, would be:

3 - 4 - 1 - 4,
4 - 1 - 4 - 3,
1 - 4 - 3 – 4.

These are rotations of the root position permutation. We only need to count one of the four rotations to specify a maj7 chord. It could be any of the four.

(Note that I'm using the word "inversion" in the musical sense of harmonic inversion. A V-1 spaced chord can have any one of its four tones in the bass and each constitutes a different inversion. When we specify a permutation by the number of half steps between chord tones, rotations represent the four inversions. Sometimes mathematicians use the word "inversion" to indicate a permutation in reverse order. I'm not using the word in that mathematical sense. Also, sometimes guitarists loosely refer to different voicings of chords as inversions. I'm not using the word in that sense either.)

In counting the 165, all four representations of the maj7 quality were included:

4 - 3 - 4 - 1,
3 - 4 - 1 - 4,
4 - 1 - 4 - 3,
1 - 4 - 3 - 4.

But we only need to count one of these inversions/rotations. It would be nice if we could just divide 165 by 4 inversions and get 43. But things aren't quite that simple. (165 ÷ 4 = 41.25)

It's almost that simple. For 160 of the 165 permutations, we can eliminate three inversions and reduce the 160 down 40. All we had to do was divide by four.

Let's look at the remaining special cases after we have removed 160 permutations from the 165:

1 - 5 - 1 - 5
5 - 1 - 5 - 1
2 - 4 - 2 - 4
4 - 2 - 4 - 2
3 - 3 - 3 - 3

These five special cases reduce to:

1 - 5 - 1 - 5
2 - 4 - 2 - 4
3 - 3 - 3 – 3

You can see that a certain kind of symmetry in these permutations is what threw a wrench into our being able to divide the entire 165 by four. 1 - 5 - 1 - 5 only produces the single rotation, 5 - 1 - 5 - 1, not three other rotations. Similarly, 2 - 4 - 2 - 4 only produces a single rotation. And 3 - 3 - 3 - 3 doesn't produce any rotations.

So that's how 165 permutations reduce to 43. 160 of the 165 reduce to 40. The remaining 5 of the 165 reduce to 3. What seemed like a strange number, 43, comes about due to a few symmetrical chords that don't produce three additional inversions.

Musically, 1 - 5 - 1 - 5 is a 13#9 no R, no 5 chord. C13#9 no R, 5 = F#13#9 no R, 5. The exact same quality appears on two different roots a tritone apart. Similarly, 2 - 4 - 2 - 4 is a 7b5 chord. C7b5 = F#7b5. Again, the same quality appears on two different roots. Finally, 3 - 3 - 3 - 3 is a dim7 chord, also known as a 7b9 no R chord. Cdim7 = Ebdim7 = F#dim7 = Adim7. The dim7 quality appears on four different roots. Inverting these special case chords does not produce three other inversions due to their internal symmetries.

Mathematicians

Have mathematicians studied these kinds of things? Definitely. There's an important mathematician named George Pólya. His theorem, the Redfield- Pólya theorem, is used to calculate numbers like our 43, where symmetries can complicate calculations. I have used a computer to list and count permutations. But mathematicians use Pólya theory to elegantly determine the number of permutations in situations like ours.

A mathematician named Harald Fripertinger applied Pólya theory to counting musical chords. He calculated not only the number of four-note chords but also the number of three-note, five-note, six-note, seven-note chords, and so on. I found his work on the internet after I had done the same counts using the computer. My results and his agree. Fripertinger found the following series:

1, 1, 6, 19, 43, 66, 80, 66, 43, 19, 6, 1, 1

where 1 is the number of zero-note chords (to give the series symmetry?), 1 is the number of one-note chords, 6 is the number of two-note chords, 19 is the number of three-note chords, 43 is the number of four-note chords, etc.

Fripertinger's series is called OEIS #A035495. On the internet, you can go to oeis.org/A035495 and see his series. OEIS stands for Online Encyclopedia of Integer Sequences. It's an enormous online listing of mathematical sequences maintained by a Combinatorics mathematician named N.J.A. Sloane.

Another mathematician who studied Combinatorics problems was Theodor Molien. He came up with a lot of sequences of numbers that are called Molien series. There's one that applies to our situation called OEIS #A008610. The ninth member of this sequence is 43. The fancy name of this particular sequence is, "Molien series of 4-dimensional representation of cyclic group of order 4 over GF(2)."

In the comments, it says the series can be used to solve a necklace problem. Suppose you want to make a necklace with four black beads and eight white beads. How many different ways can you make the necklace? 43! The analogy to four-note chords is perfect. There are 12 pitches. Four of them are taken up by the four notes in a chord. The remaining eight will be the half steps between the chord tones. The four chord tones are the black beads. The eight unused pitches are the white beads. Considering inversions of a quality to be equivalent is analogous to the fact that rotated necklaces are equivalent. Mathematically, a four-note chord is just like a necklace with four black beads and eight white beads.

To generate the 43 qualities with a computer, I programmed it to recognize an octave as having 12 half steps. But if I change the number 12 in my program to 11, I get 30 qualities, the next lower member of the Molien series. No matter how many half steps I designate for an octave, I get one of the members of the Molien series. That's because I'm doing what's analogous to changing the total number of beads in the necklace while keeping four black beads.

How Many Chords Are There?

Ted listed four-note chords with paper and pencil and counted them by hand. I wrote computer programs to count the number of two-note, three-note, four-note, etc. chords. And mathematicians have counted them using Pólya theory. Here is a table showing how many chords there are of a certain number of notes with no doublings and excluding transpositions.

The row that includes chord inversions is the 11th row of Pascal's Triangle. The row that excludes inversions is Fripertinger's OEIS #A035495.

Notes Per Chord	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Including Inversions	1	11	55	165	330	462	462	330	165	55	11	1
Excluding Inversions	1	6	19	43	66	80	66	43	19	6	1	1

For computer programmers, here are the simple algorithms, in Objective-C, for enumerating the 165 and the 43 permutations:

The 165 Permutations Enumeration Algorithm

```
int p[400][4] = {0}; //the permutations, i.e. 4-distinct-note chord qualities

int count = 0; //total number of permutations

int i, j, k, m, n; //counters

for (i = 1; i < 10; ++i)
{
    for (j = 1; j < 10; ++j)
    {
        for (k = 1; k < 10; ++k)
        {
            for (m = 1; m < 10; ++m)
            {
               if (12 == i + j + k + m) //do the intervals add up to an octave?
               {
                   p[count][0] = i;
                   p[count][1] = j;
                   p[count][2] = k;
                   p[count][3] = m;
                   ++count;
               }
            }
        }
    }
}
```

The 43 Permutations Enumeration Algorithm

```
int p[400][4] = {0}; //the permutations, i.e. 4-distinct-note chord qualities

int count = 0; //total number of permutations

int i, j, k, m, n; //counters
for (i = 1; i < 10; ++i)
{
    for (j = 1; j < 10; ++j)
    {
        for (k = 1; k < 10; ++k)
        {
            for (m = 1; m < 10; ++m)
            {
                if (12 == i + j + k + m) //do the intervals add up to an octave?
                {
                    //check through the saved permutations in p[][] to see if we have found a duplicate
                    //we rotate the saved permutations to check against all four musical inversions
                    BOOL duplicate = NO;
                    for (n = 0; n < count; ++n)
                    {
                        if ((p[n][0] == i && p[n][1] == j && p[n][2] == k && p[n][3] == m) ||
                            (p[n][1] == i && p[n][2] == j && p[n][3] == k && p[n][0] == m) ||
                            (p[n][2] == i && p[n][3] == j && p[n][0] == k && p[n][1] == m) ||
                            (p[n][3] == i && p[n][0] == j && p[n][1] == k && p[n][2] == m))
                        {
                            duplicate = YES;
                            break;
                        }
                    }
                    if (!duplicate) //if not a duplicate, save the permutation in p[][] and increment
count
                    {
                        p[count][0] = i;
                        p[count][1] = j;
                        p[count][2] = k;
                        p[count][3] = m;
                        ++count;
                    }
                }
            }
        }
    }
}
```

In addition, I modified the above algorithms to count two-note, three-note, five-note, six-note, etc. chords.

—James