**Triangles, Squares, Generating Functions**

To figure out the general rules for constructing a simplex it is instructive to look at the lower dimensional cases. For example, a triangle is constructed by taking a line segment, finding the middle, moving away from it in the second dimension and making a point. Then, one connects the other points to the new point with a line segment. For a tetrahedron one starts with a triangle, finds its center, and projects from that point orthogonally into the 3rd dimension. As before this new point is connected to the other points by means of line segments. In general the rule is to add a point in a new direction. It must be orthogonal to the other directions, and directly above the center of the previous figure. Once such a point is made, connect all the old points to the new point with line segments. This process is shown for the first few iterations in the following Figure 1. The points are numbered so that the reader can track the changes that occur from one figure to the next.

|  |  |
| --- | --- |
|  | This is a point |
|  | Two points  One line segment |
|  | Three points  Three line segments  One triangle |
|  | Four points  Six line segments  Four triangles  One tetrahedron |

Figure 1: *0, 1, 2, and 3 simplices (Wolfram)*

The rules for generating hypercubes can also be inferred from the lower dimensional versions. When we want to create a square from a line segment we can simply translate a copy of the line segment across in the second dimension and connect the new points to the ones that they were copied from to make a square. To make a cube from a square the procedure is similar: we translate a copy of the entire square in the 3rd dimension, then connect the points to the original points. The general rule then is to copy the figure into the next orthogonal dimension and connect the points across in that dimension. This is shown in the Figure 2. As before the points are numbered so inferences can be drawn across dimensions.

|  |  |
| --- | --- |
|  | This is a point |
|  | Two points  One line segment |
|  | Four points  Four lines  One square |
|  | Eight points  Twelve lines  Six squares  One cube |

Figure 2: *0, 1, 2, and 3 dimensional hypercubes*

We run out of figures that are easily rendered however. This is shown in Figure 3 and Figure 4 as the first four dimensional simplex and hypercube. This leads to the use of generating functions to give structure to the higher dimensional versions of the square and triangle.

|  |  |
| --- | --- |
|  | Five points  Ten line segments  Ten triangles  Five tetrahedrons  One 4 simplex |

Figure 3: *The 4 simplex (Wolfram)*

|  |  |
| --- | --- |
|  | Sixteen points  Thirty-two lines  Twenty-four squares  Eight cubes  One 4-dimensional hypercube |

Figure 4: *The 4 dimensional hypercube*

Generating functions are a way of using polynomials and their properties to help collect and organize objects. They are a formal power series with one or more variables and are often used in combinatorics. In this chapter generating functions are used to count the numbers of different dimensional objects in higher dimensional objects.

A formal power series is represented as

The number multiplier, or coefficient, in the polynomial tells how many of the objects there are, the power of the variable or indeterminate (x or y) tells the dimension of the object being counted. The coefficient here is a*­­n* and refers to the number of the objects there are, the x*n* or y*n* correspond to an object with n dimensions. In this paper we assign to powers of the variable x figures associated with simplices, and to powers of the variable y figures associated with hypercubes.

For a few examples consider a triangle, a square and a cube. The triangle is a figure that has one 2 dimensional object, the triangle itself, three 1 dimensional objects which are line segments, and three 0 dimensional objects that are points. This in itself could be used to describe it but we can write it in a more efficient manner as (using the variable x for simplicial figures)

We can remove the parenthetical descriptions to leave the bare algebraic structure:

For squares, a very similar notation exists. However squares have four points and four lines and so will look like this:

We can remove the redundant description here as well:

Cubes are different from squares and triangles in that they are 3 dimensional. This will correspond to having more objects to count. We can elucidate a cube as:

orThe n dimensional simplex will be represented as S*n*, the variable will be in x, and it will have *n+1* points. A triangle is a 2 simplex

S*n*+1=S*n*

because the previous figure is part of the new figure,

+x\*S*n*

all points make lines when connected to the new point, all lines form triangles when connected to the new point and all n-1 dimensional figures will likewise form n dimensional figures,

+1

for the new point since it is not covered in the previous steps.

[This is explained graphically in Figures 1 and 3 and logically in the second paragraph.]

Rewritten and without the explanations it looks like

S*n*+1=(x+1)S*n* + 1  
S0= 1

So then we can see that S1 is a line by

and that S2 is a triangle

and that Sn can be written as a sum like so:

We can note that we can also write it via the geometric series as

and since we can reduce this with the binomial theorem

we can write it in a final form as

A similar analysis will now be applied to hypercubes. The hypercube in n dimensions will be represented as H*n* and the variable will be in y. [Reference Figures 2 and 4 for visualization, and paragraph 3 for the logic.] Thus we have

H*n*+1= 2\*H*n*

because of the doubling in a direction orthogonal will keep the original and make a copy,

+y\*H*n*

connecting the points results in new lines, all original lines lead to squares and in general all *n-1* dimensional figures will create *n* dimensional figures.

It can be rewritten as

H*n*+1 = (y+2)\*H*n*

and since

H0=1

it is easily seen that

Hn=(y+2)n ﻿

which is the same as the sum given by the binomial theorem:

These previous geometric forms had a great deal of symmetry due to being created by only one geometric operation. We can however mix these two types of geometric generating actions. Quickly we come to the conclusion that they do not commute. As an example, start with a line segment. It is denoted as before equally well by or by. We will choose to represent it as . As the next step we take two distinct copies of the line, one to operate on with a simplex operation, and the other with a hypercube operation. We get both

and

The interesting part is when we do one of the hypercube operations on the triangle, and one of the simplex operations on the square. The first of these proceeds as

which results in

Intuitively when we move a triangle in a space orthogonal to it and connect it across we should have a triangular prism. We now can render our new shape into English as

There is a corresponding illustration in Figure 5

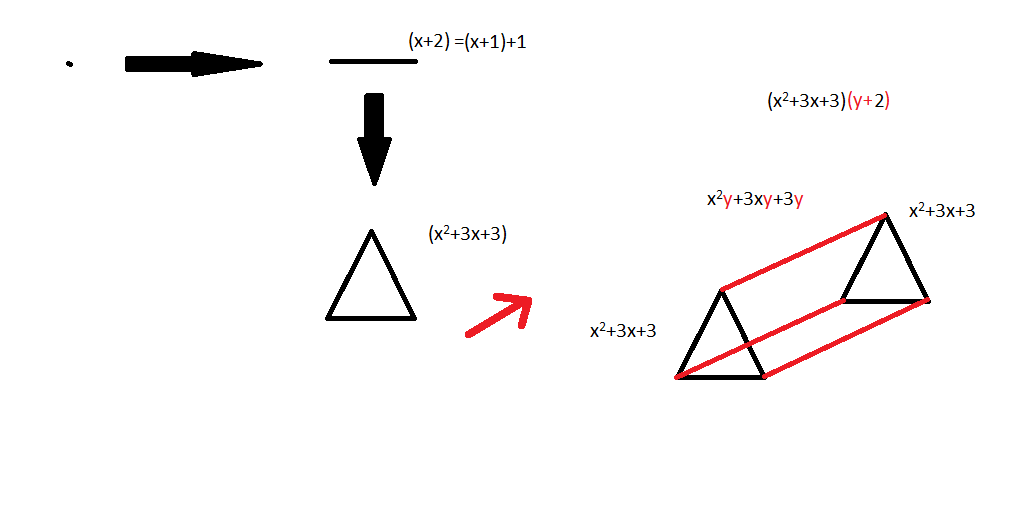
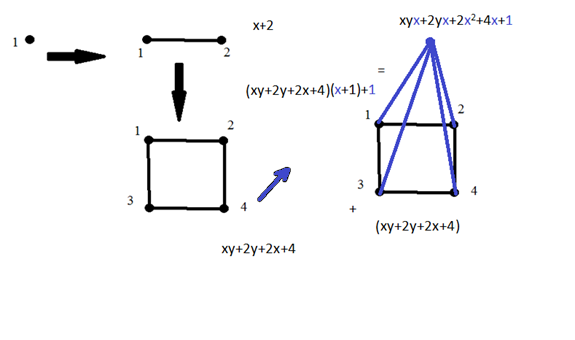


Figure 5: *Triangular prism as an archetype of a mixed figure*

But we still have the square to do a simplex operation on to compare it with:

And if we think about it intuitively, if we have a square and then add one more point above it and connect all the other points we get a square pyramid figure

and we can compare differences between these two figures. One has an xy contribution, the other has a yx contribution, and we can see that one results in squares, the other triangles. Neither the operations nor the variables will commute.Figure 6: *Square pyramid as an archetype of a mixed figure*

*Figures 5 and 6 have arrows with two different colors to better display the changes a figure undergoes when being translated orthogonally or by being connected to an point lying orthogonal to it*

But why stop here? After all there are higher dimensional versions of these 3 dimensional mixed figures. We can label each figure by a vector in a higher dimensional discrete noncommutative function space. This helps organize things by telling what operations have occurred to create the figure.

The method I use utilizes a vector in the aforementioned function space whose odd indices indicate steps of the hypercube type, and whose even terms indicate steps of the simplex type. For an example, (1,4,7,2) would correspond to starting with a point, doing one hypercube operation, then 4 simplex operations back to back, then 7 hypercube operations, then 2 simplex operations to create a 14 dimensional figure.

For a specific example if we have a vector of this sort

it corresponds to

which looks daunting. If we take it term by term though it is much simpler. Also a simplification by using different letters to show what is at the core of the calculation is quite helpful. If we substitute Ai for and with Bi and lastly with Ci­ it will look much nicer. Explicitly,

This can be written as

or, if we allow there to exist an initial C0 that equals 1 we can write it as a sum of products, useful for disentangling nested sequences,

We come to realize that our large sum that corresponds to our vector can be rewritten in a simpler and more elegant form, namely that the vector

will correspond to a figure described by

We can prove this result inductively. Using a base case of n=1,

as intuitively expected. We next assume it holds for *n* and prove it will hold for *n+1* from that:

Then if we apply hypercubing a*n*+1 many times followed by b*n*+1 many simplexing steps:

as expected.

If we substitute back what our capital letters stood in for we get (remembering that b0=1)

which from our previous analysis can be further decomposed into