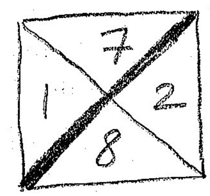
DESCRIPTION OF SPATIAL DECOMPOSITION OF FACIAL PLANES OF AN OCTAHEDRON BY A BINARY SYSTEM.

In an effort to systematically describe the cells in which space is decomposed by the facial planes of a polyhedron, we offer the following.

We number the facial planes of an octahedron as follows, so that planes 1 & 5, 2 & 6, 3 & 7, and 4 & 8 form opposite (parallel) pairs. This is done so that we may view the octahedron as being inscribed by the tetrahedron formed by planes 1, 2, 3, & 4, or the tetrahedron formed by planes 5, 6, 7, & 8.



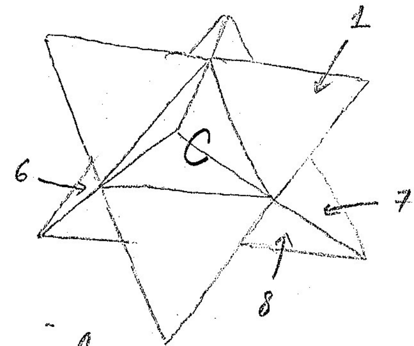
Let a cell C be considered. We describe C by a binary string of length 8, where the i^{th} digit is

0 — if the cell lies on the same side of plane i as the original octahedron

1 — if the cell lies on the opposite side of plane i as the original octahedron.

Thus, the original octahedron is represented by 00000000.

Now let us consider the tetrahedral cells forming the stella octangula. The numbers in the figure represent facial planes of the octahedron. The cell C would be represented by 10000000. This is because plane 1 is the only facial plane which "separates" the octahedron and cell C. The other such 7 cells are clearly 01000000, ..., 00000001.



Consider now the cells O_2 (see accompanying notes of 24 Sept 93; these were sent in previous correspondence). One sees that the facial planes which separate such cells from the octahedron comprise the facial planes of two <u>adjacent</u> faces of the octahedron. Thus these twelve are represented by

10000100	0100 1000	00101000	0001 1000
10000010	0100 0010	00100100	0001 0100
10000001	0 1000001	0010 0001	0001 0010

Now consider the cells O_3. One may show that these 24 cells are represented by

1100 0010	0110 1000	0010 1100	1000 0110
1100 0001	0110 0001	0001 1100	0001 0110
1010 0100	0101 1000	0100 1010	1000 0101
1010 0001	0101 0010	0001 1010	0010 0101
1001 0100	0011 1000	0100 1001	1000 0011
1001 0010	0011 0100	0010 1001	0100 0011

In the cells O_4 we have

1100 0011	1001 0110	0101 1010
1010 0101	0110 1001	0011 1100

Finally, for O_5, we have

1110 0001,	1101 0010	1011 0100,	0111 1000
0001 1110	0010 1101	0100 1011	1000 0111

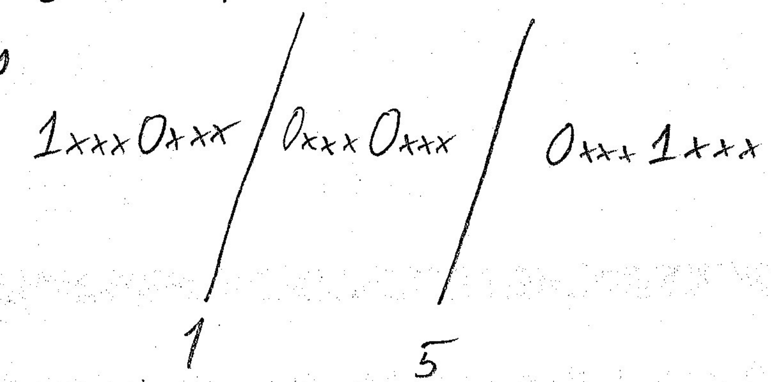
This yields a total of 59 cells. But there are 2^8 binary strings of length 8. What of the other $256 - 59 = 197$?

One easily sees that any string of one of the forms

1xxx 1xxx, x1xx x1xx, xx1x xx1x, xxx1 xxx1

is impossible, for planes 1 & 5 are parallel — no cell can be on opposite sides of planes 1 & 5 as the original octahedron.

Said another way the planes 1 & 5 divide space into 3 regions



Clearly, 1xxx 1xxx is impossible, as are the other possibilities. A careful count reveals that this eliminates 175 strings.

And aside from the original octahedron and the 8 finite tetrahedral cells, there can be no cell of one of the forms

0000xxxx   xxxx0000

This accounts for the remaining 22 cells.

What are the advantages of such an approach? Let me offer a few observations.

1. This system gives a natural "layering" of cells. A "layer" may be determined by the number of "1"'s in the string.

2. I believe we have the following result (which appears to be true for the octahedron and which I conjecture is true in general):

If the string for cell A is formed by changing one "0" in the string for cell B to a "1", then cell A lies directly on top of cell B.

For example, changing any "0" in 00000000 to a "1" yields a tetrahedral cell on top of the octahedron.

One finds that the three infinite cells on top of 10000000 are indeed 10000100, 10000010, 10000001.

Thus one can immediately form a cell diagram by inspecting the strings representing cells.

3. The result in (2) offers, I believe, a straightforward method of calculating a cell diagram for any polyhedron. I give a brief description:

A. Randomly select a point in space, and see on which side of each facial plane the point lies. Create the corresponding binary string. This string will represent the cell in which the point lies. Do this with "enough" points so that one is relatively sure that a complete list of cells is generated.

B. Group "similar" cells together. This is done as follows. Generate a set of permutations which represent the symmetry of the solid. For example, permuting the binary string as determined by $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 8 & 6 & 5 & 3 & 4 & 2 & 1 \end{pmatrix}$ represents a rotation along a 4-fold axis of symmetry (see the diagram on page 1). Thus, for the octahedron, we would need a set of 24 permutations.

Now given a cell A find the other 23 cells which are obtained from the string representing

A by permuting this string in ways which represent the symmetries of the octahedron.

For example, the string 11000010 would yield the 24 cells O_3. The string 10000000 would result in the other 7 cells representing tetrahedral cells. Each group represents one type of cell.

C. Separate these groups in "layers" according to the number of "1"'s in the string.

D. Clearly, layer 1 (1 "1") is on top of layer "0" (0 "1"'s, the original polyhedron).

E. Now choose a string S from layer 1. For each "0" in S, change it to a "1", and see if it belongs to a group determined in (B) above. If so, this kind of cell is on top of S.

F. Repeat with all groups in layer 2. For each group, select a string and change each "0" to a "1", and check if it belongs to a group in layer 3. If so the group in layer 3 is on top of the group in layer 2.

G. Do this for all groups in all layers.

The result is a cell diagram for all finite and infinite cells. By somehow (?) deciding which cells are finite, one may then determine all fully supported stellations.

Note: The stellation will be chiral ~ i.e., right and left-handed cells will necessarily belong to <u>different</u> groups. One may determine which cells have "handedness" by considering permutations of digits which correspond to "reflections" of the solid. For example, the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$

corresponds to the reflection in the plane represented by the heavy line in the diagram on page <u>1</u>.

H. The only drawback seems to be that if a cell is small, one must be "lucky" in one's random selection of points in order to obtain a point in such a cell. However, if one is lucky enough to obtain just one such point in just one cell, the other cells may be determined by the considerations of symmetry described above.

- - - - - - - - - - - - - - - - - - - -

Of course, the entire algorithm depends on the conjecture in (2) above, which I believe is correct. If so, an effective algorithm will determine once and for all an exhaustive list of the fully-supported stellations for <u>any</u> polyhedron.

I say <u>effective</u>, since the more faces on the polyhedron, the greater the magnitude of the calculations.

Note regarding G: As an afterthought, I have a method for deciding which cells are infinite: Suppose the polyhedron may be inscribed in a sphere of radius 1. Randomly select a large number of points on a sphere of, say, radius 10,000. Determine the cells to which these points belong as in 3A. These will, in all likelihood, determine the cells which are infinite. Construct a modified cell diagram, and proceed from there.

Further note: Should this algorithm prove effective, we may use it to find all fully supported stellations of polyhedra in dimensions higher than 3.