The Electronic Journal of Mathematics and Technology, Volume 18, Number 1, ISSN 1933-2823

Four Polytope Products: Join, Fusil, Prism, and Meet

Thomas L Ruen
tomruen@gmail.com

Abstract: There are four special operators on polytopes – join, fusil, prism, and meet. Joins create pyramid forms, joining all elements. Fusils creates cross polytopes, "fusing" all elements excluding bodies. Prisms create hypercubes as Cartesian products. Meets use Cartesian products of polytope nets to generate skew polytopes, also excluding bodies. This paper explores lower dimensional examples of these operators and how to work with them. The f-vectors for all 4 operators can be generated by products like coefficients of polynomial products. Detailed k-face elements are computed via product tables and can be shown in Hasse diagrams.

1 Introduction: Four Operators

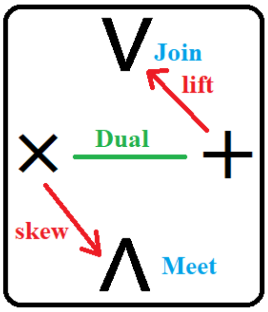


Figure 1

In this paper we use Norman Johnson names [4] with join, (**∨**), fusil (a rhombic "sum"), (**+**), prism (a cartesian product), **×**. Johnson named fusil from a rhombic shape, and it also contains the verb fuse as the element polytopes can share the same center without degeneration. Johnson didn't mention the final meet (**∧**), but join-meet are a natural extension, reminding us of union and intersection.

A 2016 paper [3], described all 4 products and names our meet operator as a *topological product*, symbol (□).

Richard Klitzing [5] name the four operators. The join operator, ($\times_{1,1}$) is called a pyramid product. The fusil operator ($\times_{1,0}$), called a tegum product. The prism operator, ($\times_{0,1}$). Finally, the meet operator, ($\times_{0,0}$) is called a honeycomb product.

Figure 1 shows the join can be seen as a "dimensional lift" of the fusil (+), while prism and fusil are duals, and meet is a skew down-rank construction from the prism.

The operators can be expressed purely abstractly, as orthogonal products of *k*-faces of each polytope. An *i*-element polytope A product with a *j*-element polytope B defines an *i*×*j* matrix of product elements. The join and fusil have joined elements, and the prism and meet have prism elements. These will be described in detailed examples with product tables and Hasse diagrams.

Figure 2 shows example figures for all 4 operators, joining a pentagon and a point, fusing a pentagon and a segment, prism of a pentagon and segment, and a meet of two orthogonal pentagons, projected down from 4-dimensions.

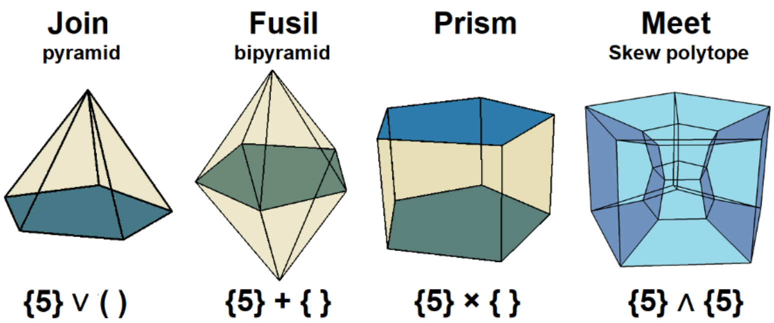


Figure 2

1.1 Polytope Rank

An *n*-polytope is said to have rank *n*, bounded by (*n*-1)-faces**.** A polygon is rank 2, bounded by 1-faces or edges, polyhedron rank 3, bounded by 2-faces.

The join operator adds one dimension or rank, while meet subtracts one rank.

- **Join:** Rank(A ∨ B) = Rank(A) + Rank(B) + 1
- **Fusil:** Rank(A + B) = Rank(A) + Rank(B)
- **Prism:** Rank(A × B) = Rank(A) + Rank(B)
- **Meet:** Rank(A ∧ B) = Rank(A) + Rank(B) – 1

1.2 Polytope f-vectors and products

A product polytope's f-vector can be computed like polynomial products from its elements. The polynomial $\mathbf{x}^k$ powers are mapped onto *k*-faces, with term coefficients as each f-vector count element.

The products can be seen in by extended f-vectors. An f-vector lists counts of *k*-faces, *k*=0…*n*-1. A f-vector is extended by a -1-face element (empty set, or nullitope) as count 1 for the join and fusil operations. The join and prism operators also include the body as an n-face (body) as 1.

A *regular polytope* is uniquely defined by its f-vector counts, while other polytopes must include a list of *k*-face polytope types for completeness.

These extended f-vectors are written with a leading 1 (nullitope), or trailing 1 (body). We can use the operators as (**∨, +, ×, ∧**), or subscript versions used by Klitizing ($\times_{1,1}$, $\times_{1,0}$, $\times_{0,1}$, $\times_{,0,0}$).

- Join ∨ or $\times_{1,1}$: (**1**,f,**1**)
- Fusil + or $\times_{1,0}$: (**1**,f)
- Prism × or $\times_{0,1}$: (f,**1**)
- Meet ∧ or $\times_{,0,0}$: (f)

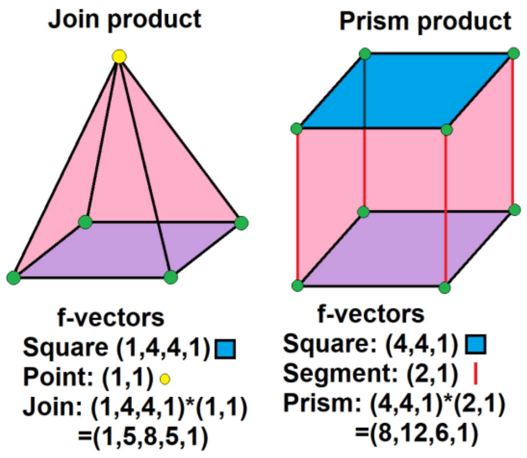


Figure 3

Figure 3 (left) shows a *square pyramid* as a join of a square base and an offset point.

The square base has extended f-vector **(1,4,4,1)**, and a point can be represented as **(1,1)**. Their product can be computed as $\mathbf{(1+4x+4x^2+x^3)(1+x)=1+5x+8x^2+5x^3+x^4}$. Then coefficients can be extracted as **(1,5,8,5,1)**. A square pyramid has 5 vertices, 8 edges, and 5 faces.

Figure 3 (right) shows a square prism product. It doesn't include the leading nullitope 1.

A square is **(4,4,1)**, and a segment **(2,1)**. Their product as a polynomial is $\mathbf{(4+4x+x^2)*(2+x)=8+12x+6x^2+x^3}$, coefficients extracted into extended f-vector **(8,12,6,1)**. A cube has 8 vertices, 12 edges, and 6 faces.

1.3 Characteristic polynomial and Euler characteristic

If we consider the extended f-vector a polynomial, f(x), with f(0)=1, and f(-1)=0 for convex polytopes (or topological spheres). This is related to the *Euler characteristic*, which is 0 for even dimensions, and 2 for odd dimensions. However the inclusion of the nullitope and body elements allow both even and odd dimensional convex polytopes to have f(-1)=0.

2 Polygon-point Joins (pyramids)

Figure 4 shows a series of *pyramids* (polygon-point joins). The first three can be equilateral. The rest will only have isosceles triangle lateral faces. A *p*-gonal pyramid has (*p*+1,2*p*,1+*p*) elements.

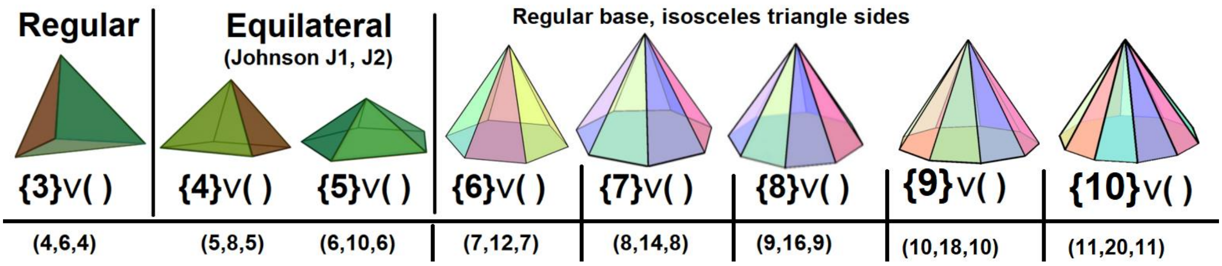


Figure 4

Higher dimensional polytopes can also be joined with a point to make higher polytopes. A dual join for a pyramid ***(AV( )) =(*A)V( ). {*p*}** is Schläfli symbol for regular p-gon, and **( )** for a point.

2.1 Hasse Diagrams and Product Tables

A Hasse diagram [6] can be used to represent the-face elements of a polytope. Symmetric polytopes can be represented as a reduced Hasse diagram with counts associated with each *k*-face type. For a polytope product **AVB**, each *i*-face of A is joined to each *j*-face of B, creating a (*i*+*j*+1)-face. This is represented in the product table, and has a direct correspondence to a Hasse diagram.

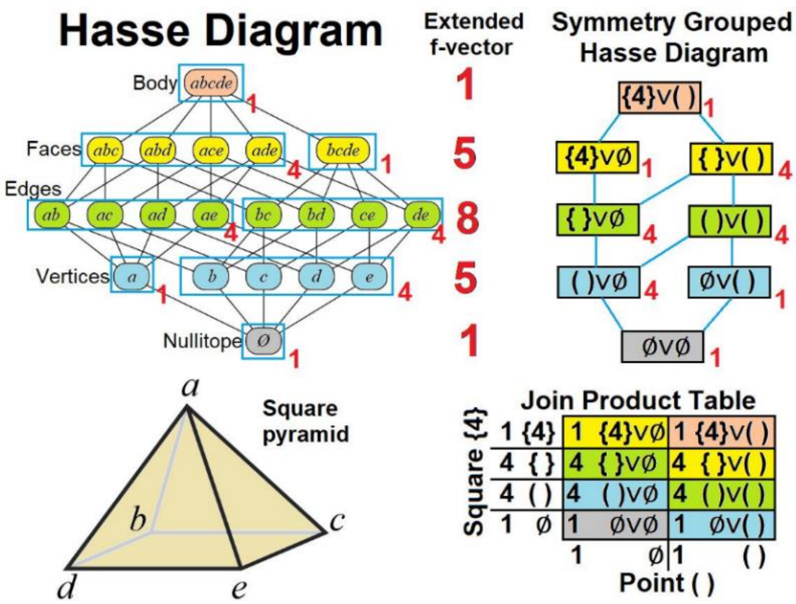


Figure 5

Figure 5 shows a Hasse Diagram for a Square pyramid left. On the right is the same Hasse diagram simplified by grouping elements in the same symmetry positions and including a node count. The lower right shows the equivalent product table with 4 square elements in row headers **(1,4,4,1)**, and two vertex elements in columns headers **(1,1)**.

2.2 Joins and Fusils of segments

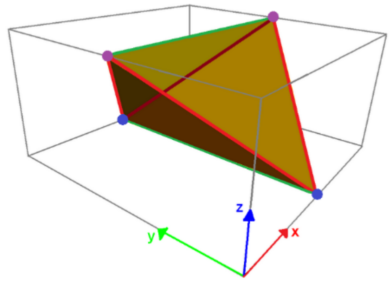
The fusil product represents a subset of a join product where body elements are excluded. A join and fusil contain the same number of vertices. Geometrically a fusil can have an orthogonal offset and this will create a "skew polytope" product.

For example, **figure 6** shows the join of 2 segments (1-polytopes) will create a tetragonal disphenoid, a lower symmetry of the regular tetrahedron.

Join product table

1 { }	1 { }vØ	2 { }v( )	1 { }v{ }
2 ( )	2 ( )vØ	4 ( )v( )	2 ( )v{ }
1 Ø	1 ØvØ	2 Øv( )	1 Øv{ }
	1 Ø	2 ( )	1 { }

Tetragonal Disphenoid



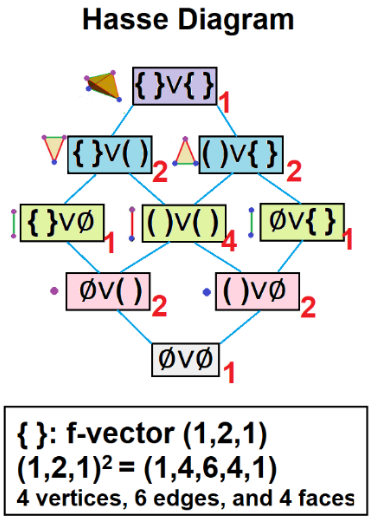


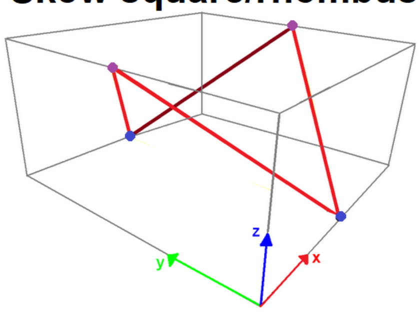
Figure 6

Figure 7 shows the fusil product of two segments is similar but drops the segment (body) elements, reducing the disphenoid into a skew square or rhombus. If the offset length of the join/fusil is reduced to zero it becomes a planar square or rhombus.

Fusil product table

2 ( )	2 ( )vØ	4 ( )v( )
1 Ø	1 ØvØ	2 Øv( )
	1 Ø	2 ( )

Skew square/rhombus



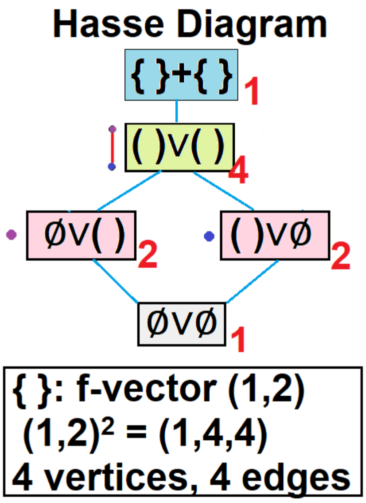


Figure 7

2.3 Joins, semi-joins, and fusils

We can see the join products are expressed as **(1,f_A,1)*(1,f_B,1)**, and a fusil is expressed as **(1,f_A)*(1,f_B)**.

This suggests a possibility of a hybrid **(1,f_A)*(1,f_B,1)** and **(1,f_A,1)*(1,f_B)**, and topologically this is valid, although it creates "open" polytopes, ridges with only 1 facet attached. We call these semi-joins, and use ⊢ and ⊣ as partial + symbols, reminding us the side that has the horizontal line dropped the body element.

1. **Join A ∨ B (1,f_A,1)*(1,f_B,1)**
2. **Semi-joins:**

   A ⊢ B **(1,f_A,1)*(1,f_B)**

   A **⊣ B (1,f_A)*(1,f_B,1)**
3. **Fusil** A + B **(1,f_A)*(1,f_B)**

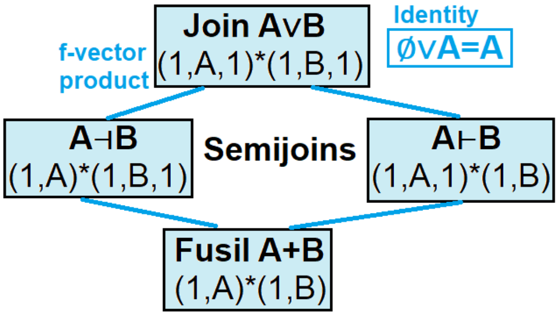


Figure 8

Figure 8 shows the join, semijoin,and fusil relations.

This allows us to see the join as the union of 2 semi-joins, the fusil the skew-intersection of semi-joins:

Union: **(A ⊢ B) ⋃ (A ⊣ B) = A ∨ B**

Intersection: **(A ⊢ B) ⋂ (A ⊣ B) = A + B**

Figure 9 shows the join of 2 segments again. The join makes a disphenoid tetrahedron. The semi-joins make half-polyhedra, and the fusil is the intersection or open boundary of each.

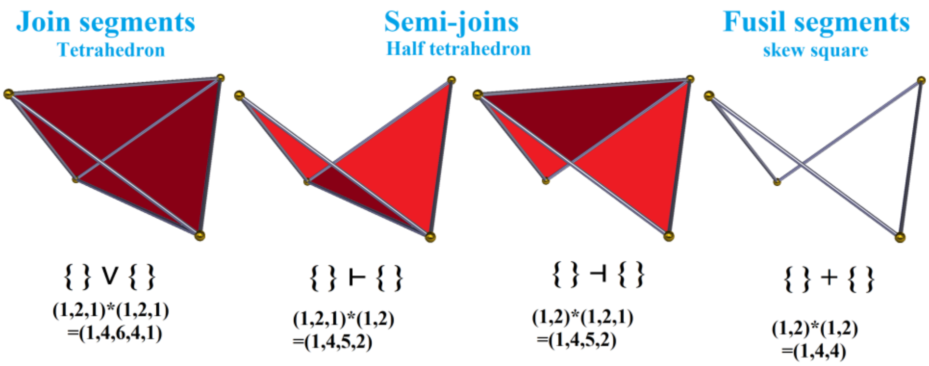


Figure 9

Again, we can see, geometrically, the join requires an orthogonal offset to avoid degeneracy, while the fusil does not, so it can be flattened into a planar square or rhombus.

A skew polytope is valid, while it just doesn't have a well-defined interior since it can't bound a volume in the higher dimensional space.

Triple fusils would have 4 semi-joins, and we'd need to union all of them to make a full join: **(A ⊢ B ⊢ C), (A ⊣ B ⊢ C), (A ⊢ B ⊣ C)**, and **(A ⊣ B ⊣ C)**.

2.4 Polygon-segment joins and fusils (wedges and dipyramids)

Joining a polygon and a segment requires 4-dimensions to avoid degeneracy, but we can draw them as projective 3D diagrams, as if the offset is zero, and elements overlap in space.

A polygonal join **{p} ∨ { }**. *{p}* is the Schläfli symbol for a regular *p*-gon, and **{ }** is the symbol for a line segment (1-polytope). This paper proposes calling a segment join as a **wedge**, while if both joined polytopes are higher than a segment, a double join, duo-join or **duo-wedge**.

The polygonal fusils can exist in 3-dimensions, known as bipyramids or dipyramids, seen as the union of an up and down polygonal pyramid sharing the same base polygon.

Figure 10 shows joins (wedges) above, and fusils (dipyramids). In general, they have isosceles triangle faces, the join of a segment and point. The wedges are drawn as wireframes, except for the original segment in red, and polygon in blue. The full joins include 2 pyramid cells (up and down), and *n* disphenoid cells sharing the red segment and each edge of the blue polygon.

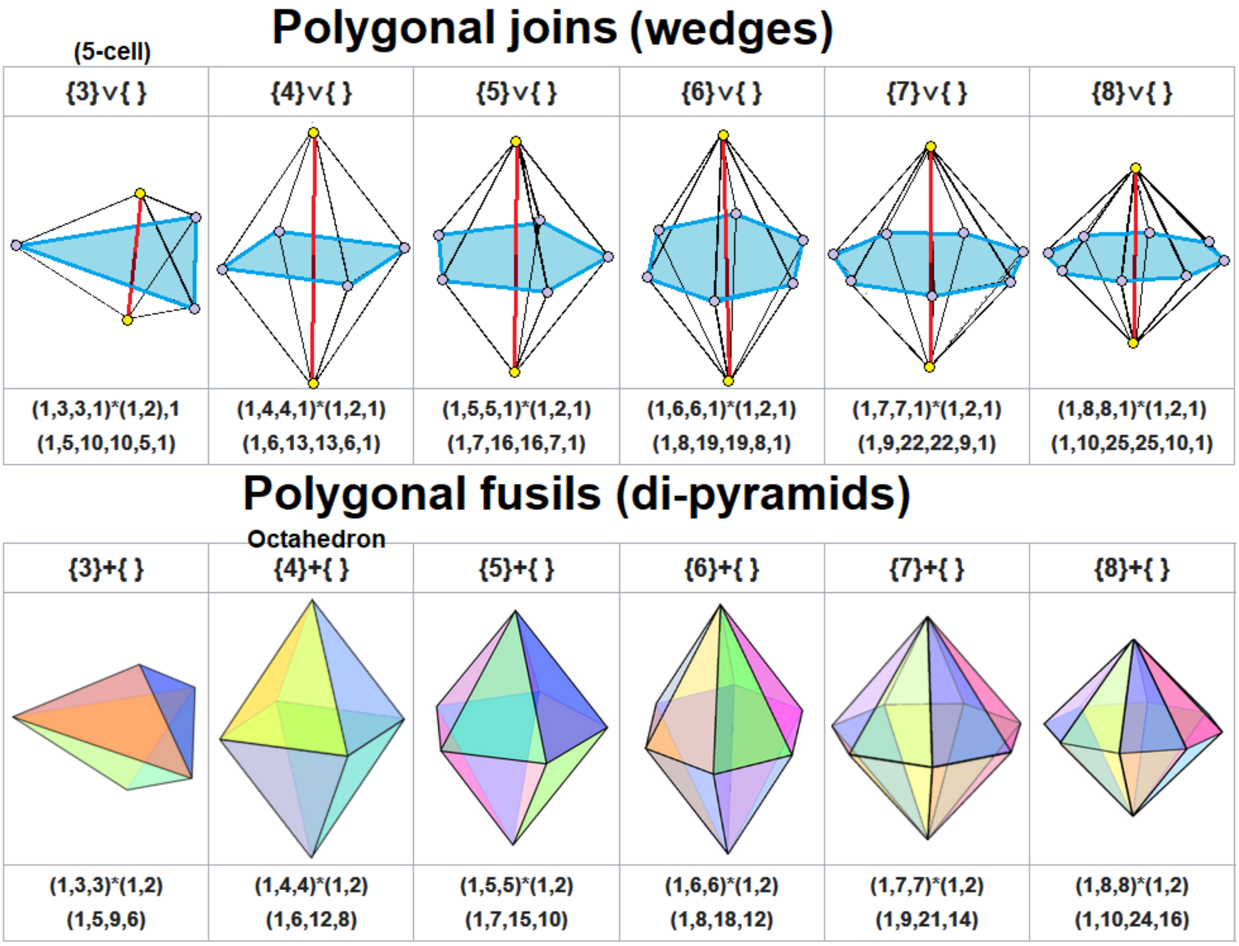


Figure 10

3 Prism and meet products

Abstractly, prism products compute similarly to join product tables, except there are no nullitope elements. A prism product of a polytope and a point creates the same polytope, so the point can be considered an identity element.

Figure 11 shows a prism product of a pentagon and a segment, producing a pentagonal prism in 3-dimensions. We can see the prism product table defines all of the elements, where a point is an identity element, **A × ( ) = A**. The symmetry-grouped Hasse diagram shows the same product table as a graph, red numbers as counts, and rows representing elements of the same rank, and blue lines show sub-element relations.

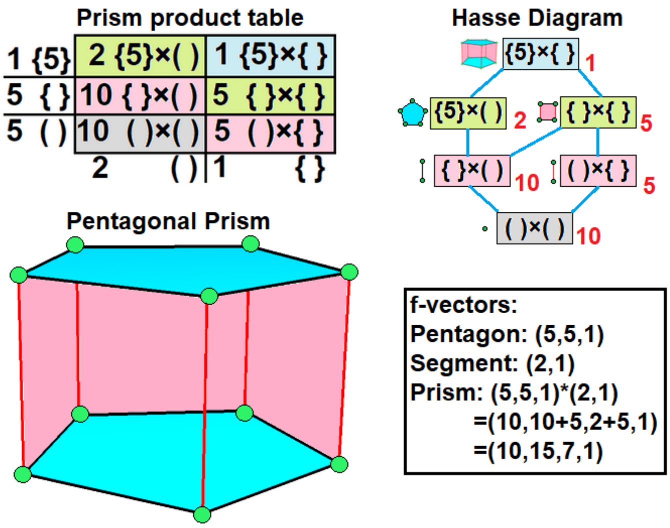


Figure 11

The f-vector calculation takes a pentagon **(5,5,1)** with segment **(2,1)**, producing an f-vector **(10,10+5,2+5,1)**, 10 vertices, 15 edge, and 7 faces.

3.1 Polygonal prisms and meets

Figure 12 (top) shows a series of polygonal prisms, **{*p*}×( ),** is a product of a polygon **{*p*}** and point **( )**. The f-vector product will be **(*p*,*p*,1)*(2,1) = (2*p*,3*p*,2+*p*,1)**, having 2*p* vertices, 3*p* edges, and 2+*p* faces (*p* squares, and 2 *p*-gons).

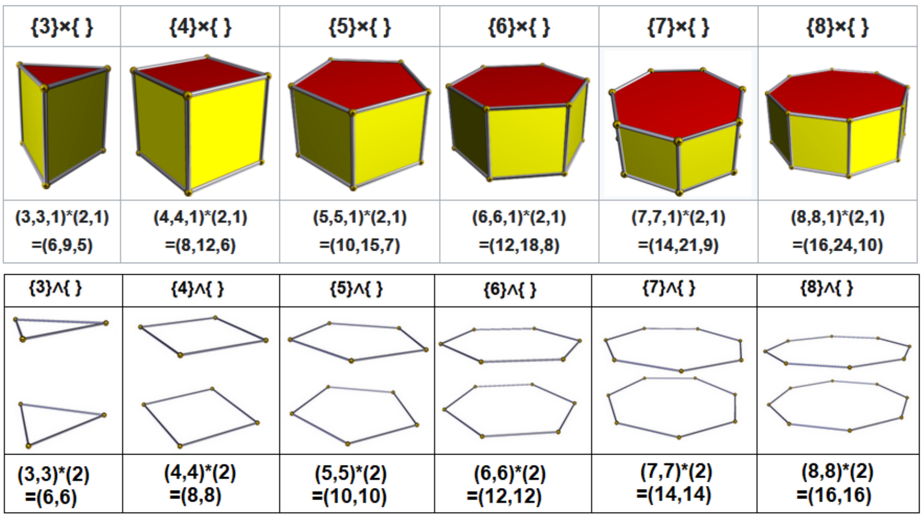


Figure 12

Figure 12 (bottom) shows a polygon-segment meet, computed as **(*p*,*p*)*(2)=(2*p*,2*p*)**, containing 2*p* vertices and edges, but it comes out as 2 parallel *p*-gons, the top and bottom edges of the *p*-gonal prisms.

We can call these polygon-segment meets “skew polygons”, but being disconnected is problematic. For meets of polygons or higher, they are connected.

3.2 Prisms, semi-prisms, and meets

We can also look at a hybrid semi-prisms that mix a prism and meet term, where the meet polytope doesn't have a body element. We propose symbols (⋋, ⋌) to represent the semi-prism, with the double-angles on the side including the body element.

1. **Prism A × B $(f_A,1)*(f_B,1)$**
2. **Semi-prisms:**
   A ⋋ B **$(f_A,1)*(f_B)$**
   A **⋌ B $(f_A)*(f_B,1)$**
3. **Meet** A ∧ B **$(f_A)*(f_B)$**

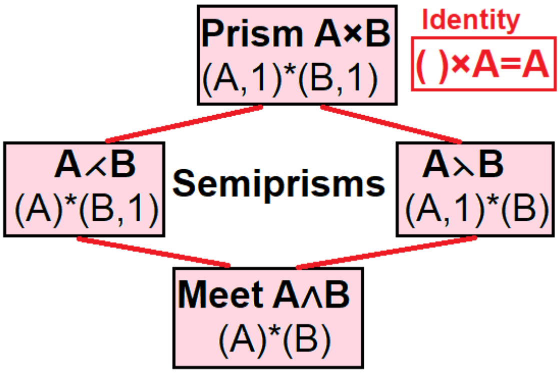


Figure 13

Figure 13 shows the prism, semiprism,and meet relations.

This allows us to see the join as the union of 2 semi-joins, and the fusil the skew-intersection of the semi-joins:

- Union: **(A ⋋ B) ⋃ (A ⋌ B) = A × B**
- Intersection: **(A ⋋ B) ⋂ (A ⋌ B) = A ∧ B**

Figure 14 shows the prism of a triangle and a segment, making a triangular prism. The semi-prisms make half-polyhedra, and the meet is the intersection or open boundary of each, making our "skew hexagon" of 6 vertices, 6 edges, but connected as 2 cycles.

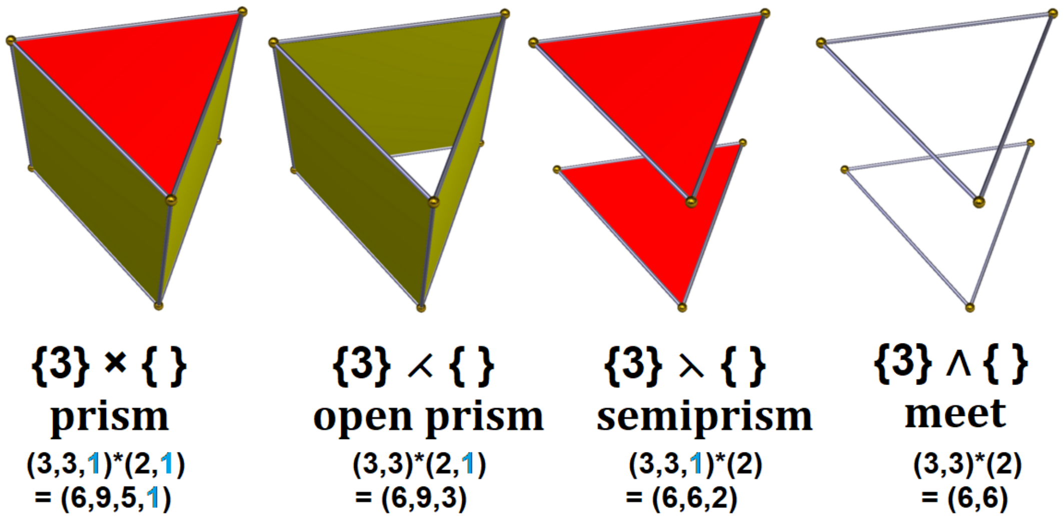


Figure 14

3.3 Polygonal double prisms and double meets

The prism product of two polygons, **{*p*}×{*q*}** is called a *double prism* or *duoprism*. These prisms exist in four dimensions, so we can't see them except in projection.

Figure 15 shows the duo-prism product of a triangle and square, **{3}×{4}**. It has an f-vector that can be computed as **(3,3,1)*(4,4,1)= (12,24,19,7,1)**, having 12 vertices, 24 edges, 19 faces (4 triangles, 3 squares, and 12 square/rectangles), and 7 cells (3 square prisms, and 4 triangular prisms).

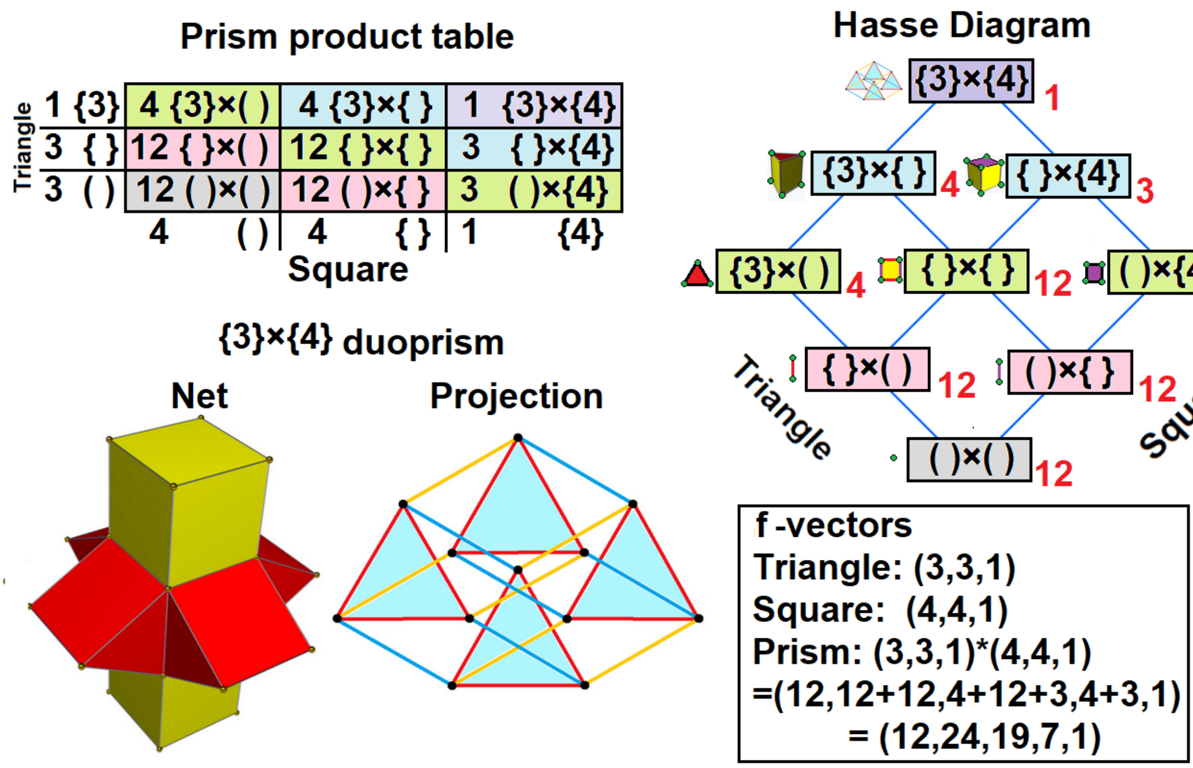


Figure 15

Figure 16 shows a *double meet* or *duomeet* of a triangle and square, **{3}∧{4}**, seen as a subset of the duoprism table, removing full polygon faces, considered "holes". The net becomes a 3×4 square grid which can be wrapped into a triangular prism adding a third dimension, and a square prism in a 4th dimension.

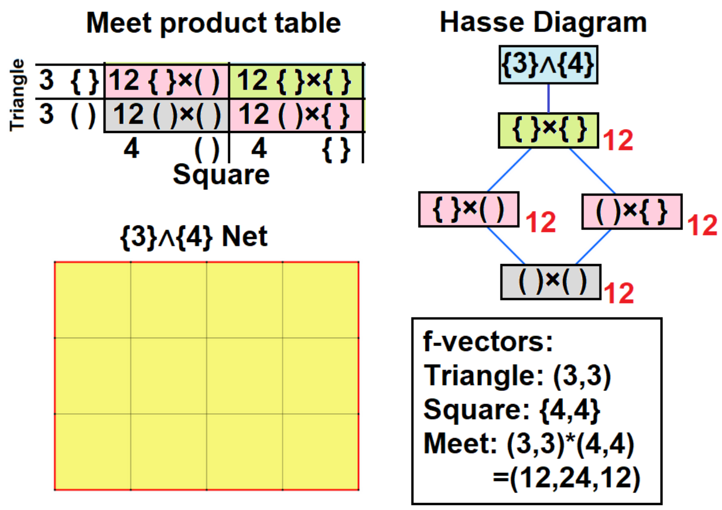


Figure 16

Figure 17 shows **{p}×{p}** or **{*p*}²** duo-prisms for p=3,4,5,6,7,8. Each is shown in 2 symmetry orthogonal projections, with central overlapping vertices in yellow on odd p's. Their meets have the same vertices and edges, and have nets as $p \times p$ grids that can be wrapped in both directions.

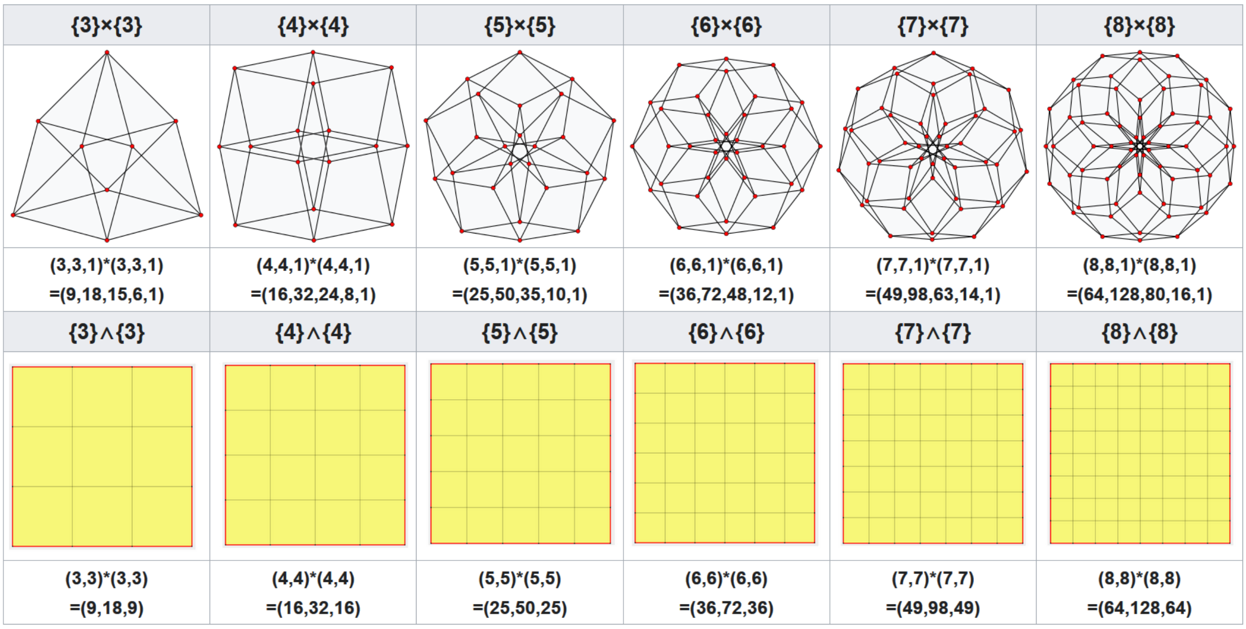


Figure 17

Figure 18 shows *duo-joins* or *duo-wedges*, **{*p*}∨{*p*}** or **2 · {p}** in 5-dimension, with **{3}∨{3}** as the regular 5-simplex. And duo-fusils, **{*p*}+{*p*}** or **2 {*p*}**, for p=3,4,5,6,7,8, (also called [duo-pyramids]) in 4-dimensions, with **{4}+{4}** making the regular 16-cell. These *duo-joins* are self-dual, while the *duo-fusils* are the duals of the *duo-prism*.

Both are shown as vertex-edge graphs, projected into a regular $2p$-gon, with red and blue edges outlining the two p-gons elements.

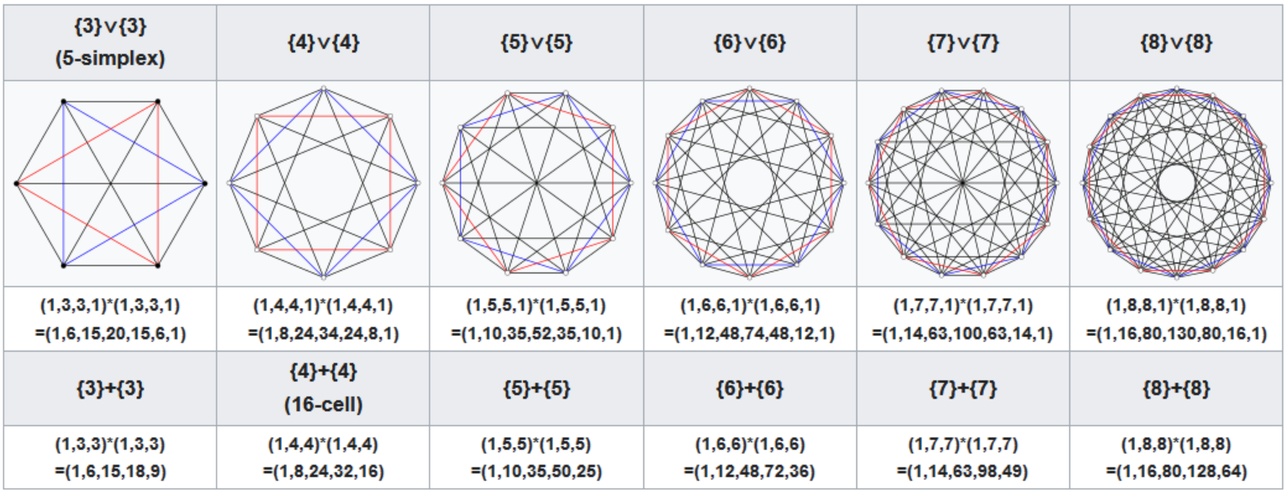


Figure 18

3.4 Triple prisms and meets

A *triple prism* product of polygons, or *tri-prism*, makes polytopes in 6-dimensions.

For example, a 6-cube can be decomposed as a product of three squares, **{4}**^{3}. The extended f-vector product is computed as **(4,4,1)**^{3} = **(64,192,240,160,60,12,1)**. It requires a 3×3×3 product table, and 27 elements in a Hasse diagram.

The product table is split into three 3×3 tables in **figure 19**, while the Hasse diagram is left undrawn.

{4}×{4}×{4} tri-prism product table

1 {4}	16 {4}×( )×{4}	16 {4}×{ }×{4}	4 {4}×{4}×{4}
4 { }	64 { }×( )×{4}	64 { }×{ }×{4}	16 { }×{4}×{4}
4 ( )	64 ( )×( )×{4}	64 ( )×{ }×{4}	16 ( )×{4}×{4}
	4×1 ( )×{4}	4×1 { }×{4}	1×1 {4}×{4}

1 {4}	16 {4}×( )×{ }	16 {4}×{ }×{ }	4 {4}×{4}×{ }
4 { }	64 { }×( )×{ }	64 { }×{ }×{ }	16 { }×{4}×{ }
4 ( )	64 ( )×( )×{ }	64 ( )×{ }×{ }	16 ( )×{4}×{ }
	4×4 ( )×{ }	4×4 { }×{ }	1×4 {4}×{ }

1 {4}	16 {4}×( )×( )	16 {4}×{ }×( )	4 {4}×{4}×( )
4 { }	64 { }×( )×( )	64 { }×{ }×( )	16 { }×{4}×( )
4 ( )	64 ( )×( )×( )	64 ( )×{ }×( )	16 ( )×{4}×( )
	4×4 ( )×( )	4×4 { }×( )	1×4 {4}×( )

f-vector: (4,4,1)^3 = (64,192,240,160,60,12,1)

Figure 19

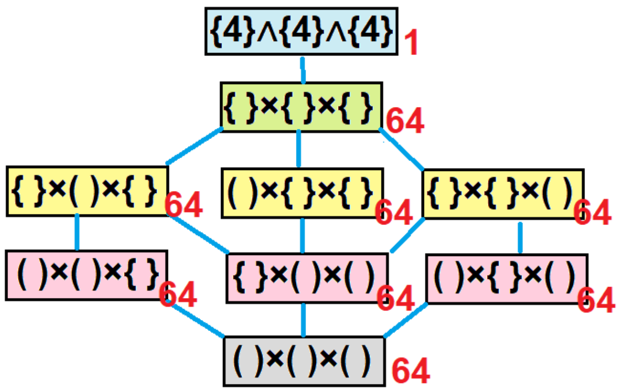
In contrast, the triple meet **{4}∧{4}∧{4}** or **{4}**$^{(3)}$ in **figure 20** is much simpler, just 8 elements. It can be seen as a net in 3-dimensions as array of 4×4×4 cubes, where opposite faces can be "folded" into 4-cycles by each added dimension, needing 6 dimension. This represents a "flat 3D surface" we could live in, repeating in 3 dimensions without intersection, and also called a 3-torus. Coxeter would name it **{4,3,4 | 4}**, a *cubic honeycomb* **{4,3,4}** wrapped with square "holes".

{4}∧{4}∧{4} tri-meet product table

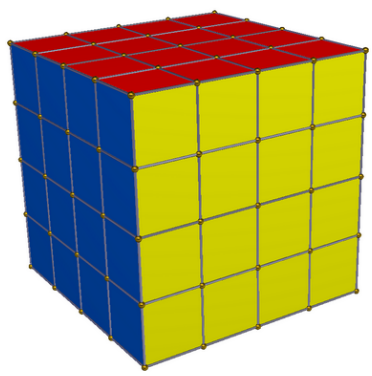
4 { }	64 { }×( )×{ }	64 { }×{ }×{ }
4 ( )	64 ( )×( )×{ }	64 ( )×{ }×{ }
	4×4 ( )×{ }	4×4 { }×{ }

4 { }	64 { }×( )×( )	64 { }×{ }×( )
4 ( )	64 ( )×( )×( )	64 ( )×{ }×( )
	4×4 ( )×( )	4×4 { }×( )

Hasse Diagram



4x4x4 cubic net

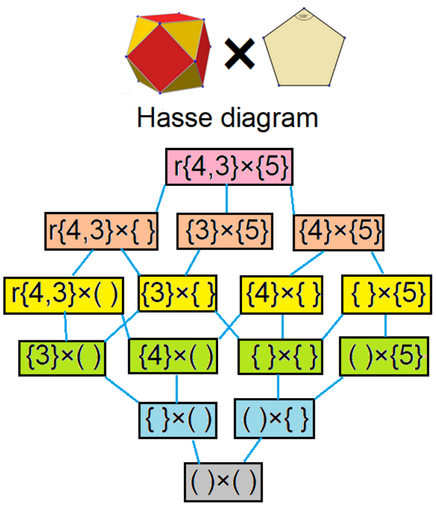


f-vectors:
Square: (4,4)
Tri-meet: (4,4)^3 = 4^3(1,1)^3
= 4^3(1,3,3,1)
=(64,192,192,64)

Figure 20

3.5 Prism and meet product with a nonregular polytope

Figure 21 shows a *cuboctahedron*-pentagon double prism can be computed, with tracking of 2 types of cuboctahedron faces, squares and triangles. This just expands the product table by 1 row. A cuboctahedron, by Coxeter, is represented as **r{4,3}**.



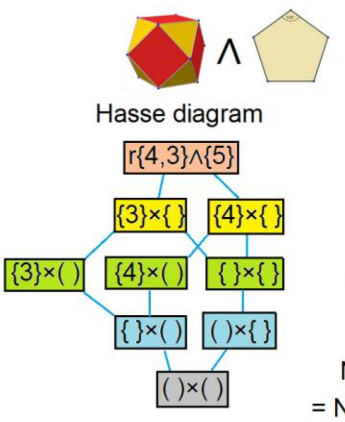
Prism product table

1 r{4,3}	5 r{4,3}×( )	5 r{4,3}×{ }	1 r{4,3}×{5}
8 {3}	40 {3}×( )	40 {3}×{ }	8 {3}×{5}
6 {4}	30 {4}×( )	30 {4}×{ }	6 {4}×{5}
24 { }	120 { }×( )	120 { }×{ }	24 { }×{5}
12 ( )	60 ( )×( )	60 ( )×{ }	12 ( )×{5}
	5 ( )	5 { }	1 {5}

F-vector product
(12,24,6+8,1)*(5,5,1)
=(60, 60+120, 40+30+120+12, 5+40+30+24, 5+8+6,1)
=(60,180,202,99,19,1)

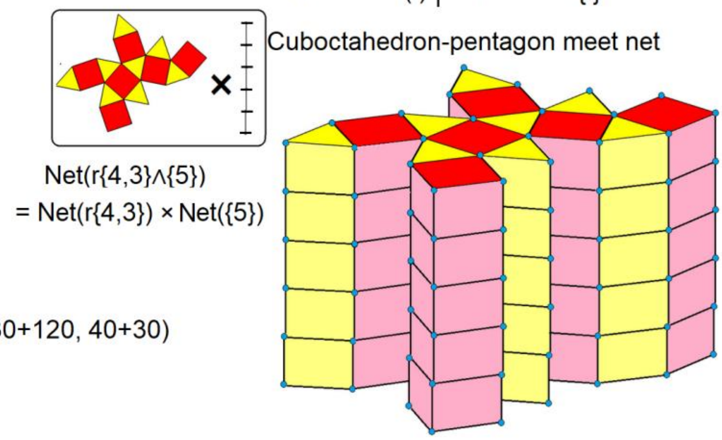
Figure 21

Figure 22 shows meet product of a cuboctahedron-pentagon, removes the body elements, **r{4,3}**, and **{5}**. The resulting skew polytope exists in 5-dimensional space, but its net can be drawn in 3D, by the prism product of a cuboctahedron net and a pentagonal net (6 linear points with 5 edges). Folding each component raises the dimension by 1.



Prism product table

8 {3}	40 {3}×( )	40 {3}×{ }
6 {4}	30 {4}×( )	30 {4}×{ }
24 { }	120 { }×( )	120 { }×{ }
12 ( )	60 ( )×( )	60 ( )×{ }
	5 ( )	5 { }



F-vector product
(12,24,6+8)*(5,5)
=(60, 120+60, 40+30+120, 40+30)
=(60,180,190,70)

Figure 22

4 Summary

We have introduced 4 primary polytope operators and 4 semi-operators as intermediate forms. We advance simple operator symbols and names that are easy to remember.

These operators can be expressed with f-vector products and Hasses diagrams to help us explore interesting higher dimensional polytopes to be confidently computed abstractly, while drawing higher dimensional polytopes is often challenging, with nets, orthogonal and perspective projections sometimes helping for high symmetry forms.

Showing lower examples in this paper helps us see the operators in action, while everything applies to higher dimensions that become harder to visualize.

Appendix I and II summarizes the operators and names on various polytopes. Appendix III shows regular constructions as powers of the four operators. Appendix IV shows the family of Hanner polytopes as an example, and Appendix V shows variations of skew polytopes on *n*-cubes.

Figure 23 shows the relations of all 8 operators.

We can see join-fusil make one set, and prism-meet make another set. The join includes the nullitope **(∅)** as an identity element, while the prism set has a point, **( )**, as the identity element.

The fusil and prism are related by duality (*****). The join and meet operators are self-dual. All dualities are re-applied to each element, like De Morgan's Law, although polygons and lower are topologically self-dual.

The union of the *semi-joins* make a *join*, while the union of *semi-prisms* make the *prism*.

The intersection of two *semi-joins* makes *fusil*, and the intersection of two *semi-prisms* makes the *meet*.

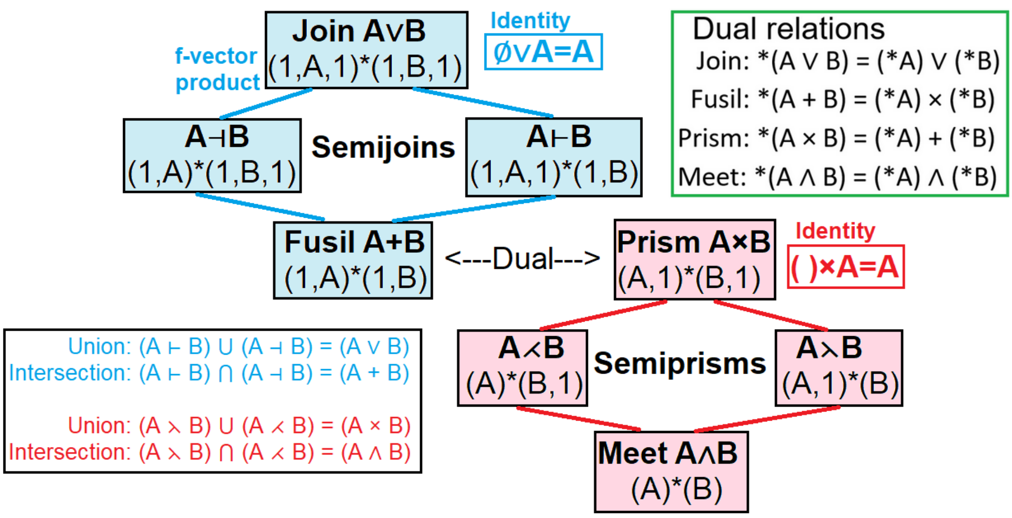


Figure 23

Appendix I: Names and symbols

In this paper we advance a set of names for the operators and polytopes:

Four operators: *names, symbols, and extended f-vectors:*

- ***Join*** $\vee$ $\times_{1,1}$ $(\mathbf{1}, f, \mathbf{1})$
- ***Fusil*** $+$ $\times_{1,0}$ $(\mathbf{1}, f)$
- ***Prism*** $\times$ $\times_{0,1}$ $(f, \mathbf{1})$
- ***Meet*** $\wedge$ $\times_{0,0}$ (f)

Polytope names:

- ***Join***
  - Point **pyramid** **A ∨ ( )**
  - Segment **wedge** **A ∨ { }**
  - Polygon+ **duo-wedge** **A ∨ B**
- ***Fusil***
  - Segment **fusil** **A + { }**
  - Polygon+ **duo-fusil** **A + B**
- **Prism**
  - Segment **prism** **A × { }**
  - Polygon+ **duo-prism** **A × B**
- **Meet**
  - Segment **meet** **A ∧ { }**
  - Polygon+ **duo-meet** **A ∧ B**

Higher product tuples: (We use Latin prefixes as *n*-tuples)

- Double **duo-** {wedge, fusil, prism, meet}
- Triple **tri-** {wedge, fusil, prism, meet}
- Quadruple **quadri-**{wedge, fusil, prism, meet}
- Quintuple **quinti-**{wedge, fusil, prism, meet}
- Sextuple **sexti-** {wedge, fusil, prism, meet}
- Septuple **septi-** {wedge, fusil, prism, meet}
- Octuple **octi-** {wedge, fusil, prism, meet}
- *n*-tuple ***n*-** {wedge, fusil, prism, meet}

Recursive products:

- **n-join** $n \cdot \mathbf{A}$
- **n-fusil** $n\,\mathbf{A}$
- **n-prism** $\mathbf{A}^{n}$
- **n-meet** $\mathbf{A}^{(n)}$

Appendix I: Names and symbols (continued)

Index and Glossary

- *k*-torus – a skew polytope, topological cartesian product of *k* polygons, in 2*k*-space.
- Body – A whole polytope, usually represented as interior.
- Cell – A 3-face (polyhedron) of a higher polytope
- Complex polytope – A polytope defined in a complex vector spaces C^n.
- *n*-cross / n-orthoplex – Regular n-polytopes with 2*n* vertices (±1, 0, …, 0)
- *n*-cube / hypercube / n-orthotope – Regular n-polytopes with 2^n vertices (±1, ±1, ±1…, ±1)
- *n*-cubic honeycomb – Infinite n-polytopes, Schläfli symbols: {4,4}, {4,3,4}, {4,3,3,4}, …
- Dipyramid/bi-pyramid – A fusil product of a polygon (or polytope) and a segment.
- Duality – Polytopes with swapped *k*,(*n*-*k*) elements, vertices/faces, edges/ridges, etc.
- Duopyramid – A join of two polytopes, usually 2 polygons.
- Edges – 1-polytope, line segments
- *k*-faces – *k*-polytope elements
- Facets – (*n*-1)-faces of an *n*-polytope
- **Fusil –** a product polytope direct sum operator, connecting all elements without body
- Ridges – (*n*-2)-faces in an *n*-polytope
- f-vector / extended f-vector – A list of polytope *k*-face counts, *k*=0..*n*-1. Extended includes: -1-face (1 nullitope), *n*-face (1 body).
- Hasse diagram – a mathematical diagram used to represent a finite partially ordered set, representing a hierarchy of elements.
- **Join** – a product polytope direct sum operator, connecting all elements, including body.
- **Meet** – a polytope product cartesian product operator, excluding body elements.
- Nullitope – a -1-rank polytope (no elements)
- Point – a 0-rank polytope
- Polygon / Polyhedron – Specific names for 2-polytope, 3-polytopes
- **Prism** – a polytope product cartesian product operator, including body elements.
- Pyramid – join product of a polytope and a point
- Rank – An *n*-polytope has rank *n*, elements, from 0-faces (vertices) to (*n*-1)-faces (facets).
- Regular polytope – a polytope where all k-face elements are identical by symmetry.
- Schläfli symbol – a description of a regular polytope, {*a*,*b*,*c*,..,*y*,*z*} with {*a*,*b*,*c*,…,*y*} facets.
- Segment – a 1-polytope bounded by 2 vertices.
- **Semi-join** – intermediate operator between join and fusil
- **Semi-prism** – intermediate operator between prism and meet
- *n*-simplex – a polytope constructed by joining (*n*+1) vertices.
- *k*-skeleton – a substructure of an n-polytope, excluding elements above *k*-faces.
- Skew polytope – a polytope spanning a dimension higher than its rank
- Vertices – point elements of a polytope
- **Wedge** – join of a polytope and segment or higher

Appendix II: Summary table of operators and polytopes

This table shows common operator names, symbols, and recursive power names. The last two columns show f-vector products and names for specific cases.

The **bold** names show the products that generate regular polytopes. (See also Appendix III)

Operator names	Symbols	Powers	Extended f-vectors	Polytope names
Join [3][4] Pyramid product [5]	A ∨ B A ⋈ B A $\times_{1,1}$ B	$n \cdot$ A $n \cdot ( )$	$(\mathbf{1},f_A,\mathbf{1})*(\mathbf{1},\mathbf{1})$ $(\mathbf{1},f_A,\mathbf{1})*(\mathbf{1},2,\mathbf{1})$ $(\mathbf{1},f_A,\mathbf{1})*(\mathbf{1},f_B,\mathbf{1})$ $(\mathbf{1},f_A,\mathbf{1})*(\mathbf{1},f_B,\mathbf{1})*(\mathbf{1},f_C,\mathbf{1})$ $(\mathbf{1},f_A,\mathbf{1})^n$ $(\mathbf{1},\mathbf{1})^n$ $(\mathbf{1},2,\mathbf{1})^n$ $(\mathbf{1},3,3,\mathbf{1})^n$ $(\mathbf{1},p,\mathbf{1})^n$ $(\mathbf{1},p,p,\mathbf{1})^n$	A ∨ ( ) = pyramid A ∨ { } = wedge A ∨ B = double wedge, duowedge A ∨ B ∨ C = triple wedge, triwedge $n \cdot$ A = A-topal n-wedge $n \cdot$ ( ) = **(n-1)-simplex, α_{n-1}** $n \cdot$ { } = **(2n-1)-simplex, α_{2n-1}** $n \cdot$ {3} = **(3n-1)-simplex, α_{3n-1}** $n \cdot {}_p$\{ \} = complex n-wedge $n \cdot \{p\}$ = p-gonal n-wedge
Fusil Rhombic sum [4] Direct sum [3] Tegum product [5]	A + B A $\times_{0,1}$ B	n A n { }	$(\mathbf{1},f_A)*(\mathbf{1},2)$ $(\mathbf{1},f_A)*(\mathbf{1},f_B)$ $(\mathbf{1},f_A)*(\mathbf{1},f_B)*(\mathbf{1},f_C)$ $(\mathbf{1},f_A)^n$ $(\mathbf{1},2)^n$ $(\mathbf{1},p)^n$ $(\mathbf{1},p,p)^n$	A + { } = fusil, dipyramid A + B = double fusil, duofusil A + B + C = Triple fusil, trifusil n A=A-topal n-fusil n{ }=n-fusil, **n-orthoplex, β_n** n ${}_p${}= **generalized n-orthoplex, β_n^p** $n\{p\}$=p-gonal n-fusil
Prism [5] Rectangular product [4] Cartesian product [3]	A × B A $\times_{0,1}$ B	A^n $\{\ \}^n$	$(f_A,\mathbf{1})*(2,\mathbf{1})$ $(f_A,\mathbf{1})*(f_B,\mathbf{1})$ $(f_A,\mathbf{1})*(f_B,\mathbf{1})*(f_C,\mathbf{1})$ $(f_A,\mathbf{1})^n$ $(2,\mathbf{1})^n$ $(p,\mathbf{1})^n$ $(p,p,\mathbf{1})^n$	A × { } = prism A × B = double prism, duoprism A × B × C = triple prism, tri-prism A^n = A-topal n-prism $\{\ \}^n$ = n-prism, **n-cube, γ_n** ${}_p\{\ \}^n$ = **generalized n-cube, γ_n^p** $\{p\}^n$ = p-gonal n-prism
Meet Topological product [3] Honeycomb [5]	A ∧ B A □ B A $\times_{0,0}$ B	$A^{(n)}$ $\{\ \}^{(n)}$ $\{p\}^{(n)}$	$(f_A)*(2)$ $(f_A)*(f_B)$ $(f_A)*(f_B)*(f_C)$ $(f_A)^n$ $(2)^n$ $(p)^n$ $(p,p)^n = p^n(1,1)^n$ $(\infty,\infty)^n$	A ∧ { } = skew meet A ∧ B = skew double meet A ∧ B ∧ C = skew triple meet $A^{(n)}$ = skew A-topal n-meet $\{\ \}^{(n)}$ = skew n-meet ${}_p\{\ \}^{(n)}$ =**complex skew n-meet** $\{p\}^{(n)}$ = **skew p-gonal n-meet** $\{\infty\}^{(n)}$ = **cubic n-comb, δ_{n+1}**

Appendix III: Infinite families of regular polytopes

Product Polytopes as polynomials exist as infinite series, with extended f-vector elements computed by the *binomial theorem*:

$$(x+y)^n = \binom{n}{0}x^n y^0 + \binom{n}{1}x^{n-1}y^1 + \binom{n}{2}x^{n-2}y^2 + \cdots + \binom{n}{n-1}x^1 y^{n-1} + \binom{n}{n}x^0 y^n$$

These solutions produce infinite families of regular polytopes. Recursively joining points produces an n-simplex. Recursively fusing segments produces the cross-polytopes. Recursive Cartesian products of segments produces the measure polytope, hypercubes, or n-cube. Recursively meeting polygons makes Coxeter's regular skew polygons $\{p\}^{(n)} = \{4,3^{n-2},4 \mid p\}$.

Table 2

Family	Power Symbol	Schläfli Symbol	Extended f-vector	Vertices	Facets
n-simplex	$(n+1)\cdot(\ )$	$\{3^{n-1}\}$	$(1, 1)^{n+1}$	$n+1$	$n+1$
n-fusil/n-cross Generalized [2]	$n\,\{\ \}$ $n\ {}_p\{\ \}$	$\{3^{n-2}, 4\}$ ${}_2\{3^{n-2}\}_2\{4\}_p$	$(1, 2)^n$ $(1, p)^n$	$2n$ pn	2^n p^n
n-prism/n-cube Generalized [2]	$\{\ \}^n$ ${}_p\{\ \}^n$	$\{4, 3^{n-2}\}$ ${}_p\{4\}_2\{3^{n-2}\}_2$	$(2, 1)^n$ $(p, 1)^n$	2^n p^n	$2n$ pn
p-gonal n-meet Aperiotopal	$\{p\}^{(n)}$ $\{\infty\}^{(n)}$	$\{4, 3^{n-2}, 4 \mid p\}$ $\{4, 3^{n-2}, 4\}$	$(p,p)^n$ $\infty(1,1)^n$	p^n ∞	p^n ∞

As well, Coxeter explored generalized cross polytopes and hypercubes. These exist in Complex space C^n where segments of 2 points are replaced by a rotational set of p-points in a complex plane, labeled ${}_p\{\ \}$.

Figure 24 shows polytopes as vertex-edge skeletons projected into the plane. Joins and fusil are combined by a shared center plus optional offset. It shows prism and meet products are projected as pairwise vector sums of vertices of element polytope vertices.

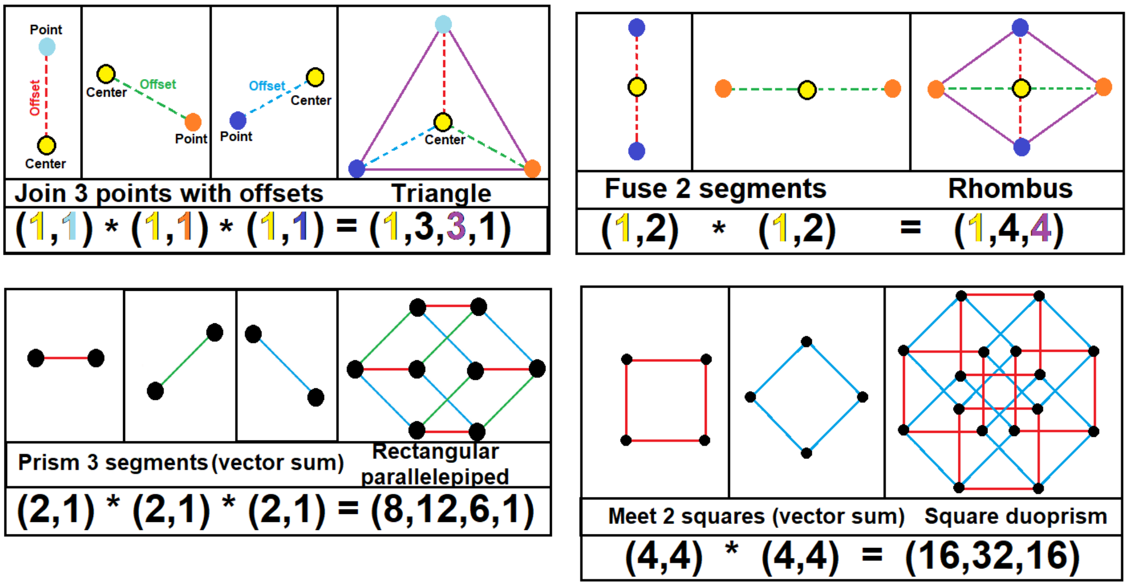


Figure 24

Appendix III: Infinite families of regular polytopes (continued)

Example graphs of the regular polytopes are drawn for n=2,3,4,5,6, (n+1)-joins (simplices) on top row, $(1,1)^{n+1}$, n-fusils (cross-polytopes) second row, $(1,2)^n$, n-prisms (hypercubes) third, $(2,1)^n$, and n-meets (skew polytopes last, triangle case), $(3,3)^n$. Graphs are orthogonal projections in Petrie polygon planes. Drawn as 1-skeletons, the n-prisms, n-meets look the same for polygons and higher, although the n-prisms are only uniform, not regular.

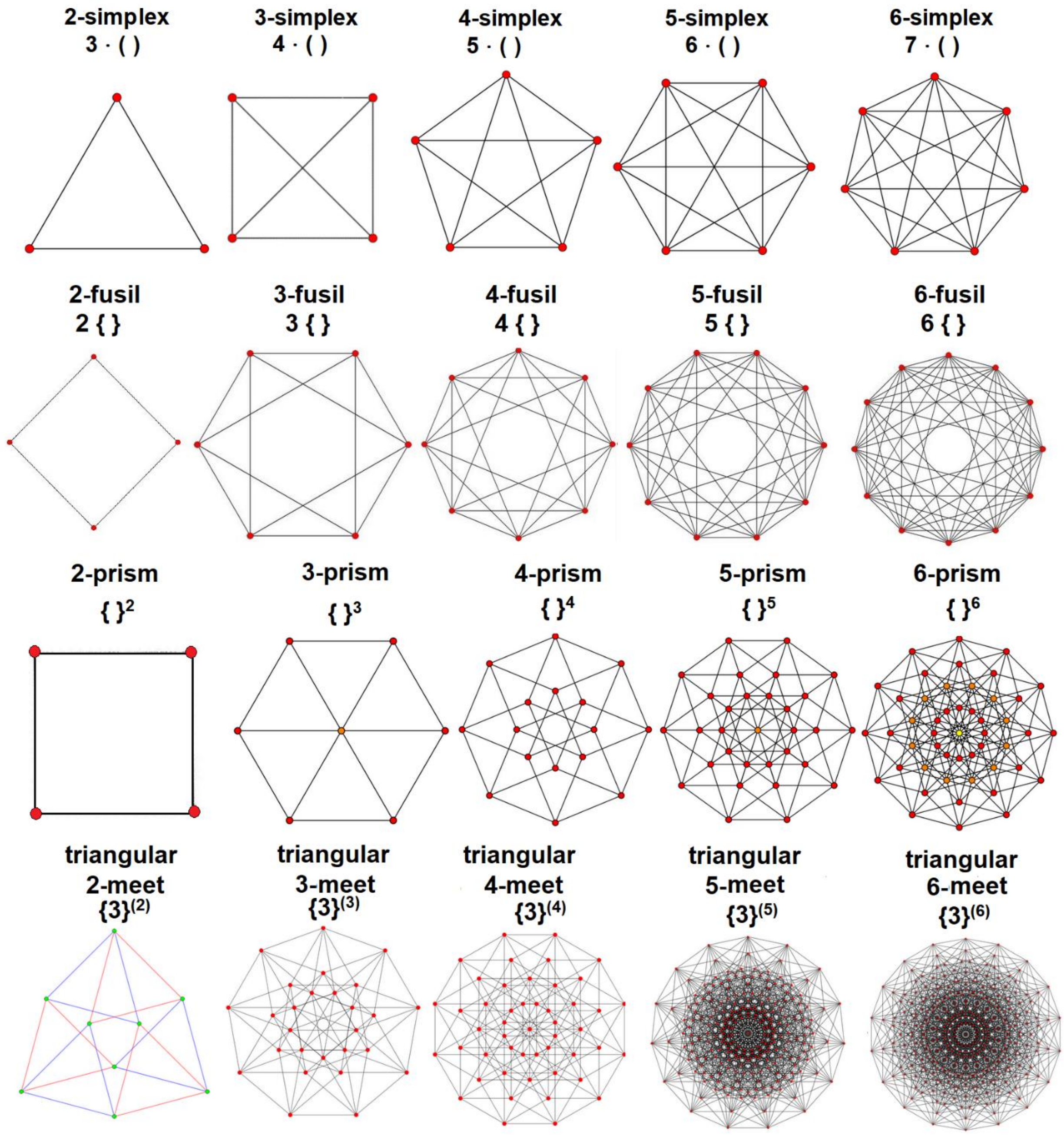


Figure 25

Appendix IV: Special examples

A Hanner polytope is computed as a recursive product of 1-polytopes, **{ }**, or higher Hanner polytopes. For 2D, there is just the square, and 3D just cube and octahedron. For 4D there are 4 cases, tesseract, **{4,3,3}** and 16-cell, **{3,3,4}**, but also **{4,3}+{ }**, **{3,4}×{ }**, a cubic dipyramid, and octahedral prism. The cases grow exponentially, while the sum of the f-vector values sum to 3^n for a rank *n* Hanner polytope, like the *n*-cubes.

Binary	Construction	Alternate	v	e	f	c	Euler	Sum
	{ }	segment	2				2	2
0	{ }+{ }	{4} = square	4	4			0	8
1	{ }×{ }	{4} = square	4	4			0	8
01	{ }+{ }×{ }	{4,3} = cube	8	12	6		2	26
11	{ }×{ }×{ }	{4,3} = cube	8	12	6		2	26
00	{ }+{ }+{ }	{3,4} = octahedron	6	12	8		2	26
10	{ }×{ }+{ }	{3,4} = octahedron	6	12	8		2	26
011	{ }+{ }×{ }×{ }	{4,3,3} = tesseract	16	32	24	8	0	80
111	{ }×{ }×{ }×{ }	{4,3,3} = tesseract	16	32	24	8	0	80
(0)1(0)	({ }+{ })×({ }+{ })	{4,3,3} = tesseract	16	32	24	8	0	80
001	{ }+{ }+{ }×{ }	{3,4} × { }	12	30	28	10	0	80
101	{ }×{ }+{ }×{ }	{3,4} × { }	12	30	28	10	0	80
010	{ }+{ }×{ }+{ }	{4,3} + { }	10	28	30	12	0	80
110	{ }×{ }×{ }+{ }	{4,3} + { }	10	28	30	12	0	80
000	{ }+{ }+{ }+{ }	{3,3,4} =16-cell	8	24	32	16	0	80
(1)0(1)	({ }×{ })+({ }×{ })	{3,3,4} =16-cell	8	24	32	16	0	80

One interesting fact, we see the sum of the k-faces in a Hanner polytope are constant, and this arises from the fact both prism and fusil operations on segments use **(1,2)**, and **(2,1)**, so elements are $\mathbf{3^n}$**-1** for *n*-dimension.

Many merry meets!

For multi-prisms the number of permutations increases exponentially when replacing a prism with a meet.

The table below shows regular skew polytopes from prisms of 2 to 4 squares, sharing the 1-skeletons of the 4-cube, 6-cube, and 8-cube.

Expressions are evaluated left to right, with a power of 2 increase in solutions, but at tetra-prisms, we need parenthesis to express 2 more recursive constructions for the complete set.

We can see the vertex and edge counts are fixed in each prism to meet substitution.

Binary	Construction	Alternate	Rank	Dim	f0	f1	f2	f3	f4	f5	f6	f7	Euler
0	{4}∧{4}	{4,4|4}	3	4	16	32	16						0
1	{4}×{4}	{4,3,3} = 4-cube	**4**	**4**	16	32	24	8					0
00	{4}∧{4}∧{4}	{4,3,4|4}	4	6	64	192	192	64					0
01	{4}∧{4}×{4}	{4,4|4} × {4}	5	6	64	192	208	100	20				0
10	{4}×{4}∧{4}	{4,3,3} ∧ {4}	5	6	64	192	224	128	32				0
11	{4}×{4}×{4}	{4,3,3,3,3} =6-cube	**6**	**6**	64	192	240	160	60	12			0
000	{4}∧{4}∧{4}∧{4}	{4,3,3,4|4}	5	8	256	1024	1536	1024	256				0
(0)1(0)	({4}∧{4})×({4}∧{4})	{4,4|4} × {4,4|4}	6	8	256	1024	1536	1056	320	32			0
001	{4}∧{4}∧{4}×{4}	{4,3,4|4} × {4}	6	8	256	1024	1600	1216	452	68			0
010	{4}∧{4}×{4}∧{4}	{4,3|4} × {4} ∧ {4}	6	8	256	1024	1600	1232	480	80			0
100	{4}×{4}∧{4}∧{4}	{4,3,3} ∧ {4,4|4}	6	8	256	1024	1664	1408	640	128			0
011	{4}∧{4}×{4}×{4}	{4,4|4} × {4,3,3}	7	8	256	1024	1664	1424	688	184	24		0
101	{4}×{4}∧{4}×{4}	{4,3,3} ∧ {4} × {4}	7	8	256	1024	1728	1600	864	260	36		0
110	{4}×{4}×{4}∧{4}	{4,3,3,3,3} ∧ {4}	7	8	256	1024	1728	1600	880	288	48		0
(1)0(1)	({4}×{4})∧({4}×{4})	{4,3,3} ∧ {4,3,3}	7	8	256	1024	1792	1792	1088	384	64		0
111	{4}×{4}×{4}×{4}	{4,3,3,3,3,3,3} = 8-cube	**8**	**8**	256	1024	1792	1792	1120	448	112	16	0

Dedication

I dedicate this paper to Norman Johnson (1930-2017) for his patient correspondences by email.

Resources

Many of the fancier 3D polytopes were rendered with Stella: Polyhedron Navigator [7]

The 2D point-edge projected images were rendered by myself in SVG graphics, some of which also exist on Wikipedia commons from my uploads.

References

[1] Coxeter, Regular Polytopes, Third edition, (1973), Dover edition, ISBN 0-486-61480-8 p.120-124 Pyramids, dipyramids and prisms, p.296 Table I-iii: Three regular polytopes in n-dimensions. Table II Regular Honeycombs

[2] Coxeter, Regular Complex Polytopes, Second edition, 1991, Cambridge University Press. ISBN 0-521-39490-2. Generalized cross-polytope, generalized cube p.118-119.

[3] Gleason, I.; Hubard, I., Products of abstract polytopes, 11 Mar 2016 https://arxiv.org/abs/1603.03585

[4] Johnson, N., 2018, Geometries and Transformations, 11.3 Pyramids, Prisms, and Antiprisms, p.163-165, p.176

[5] Klitzing, R., Different Products, occurring with Polytopes https://bendwavy.org/klitzing/explain/product.htm

[6] McMullen, P., Schulte, E. Abstract Regular Polytopes, Cambridge University Print, 2002. Hasse diagram, p.26.

[7] Stella Polyhedron Navigator, software by Robert Webb. https://www.software3d.com/Stella.php