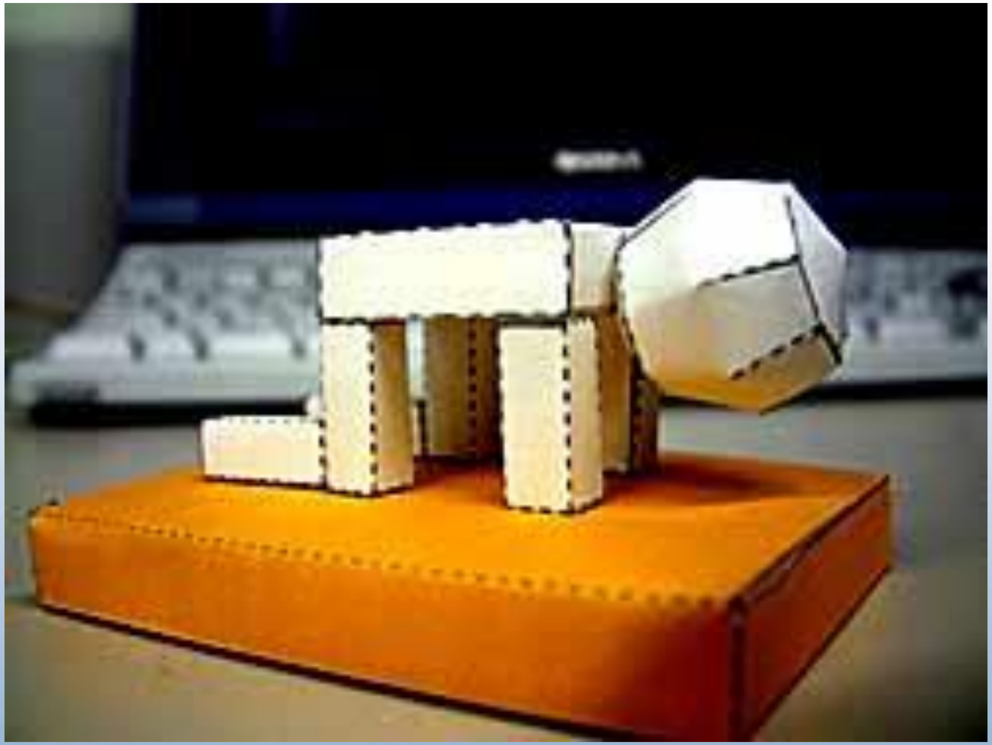
Solid Modelling

Boundary Representation BRep

Alexander Pasko, Evgenii Maltsev, Dmitry Popov

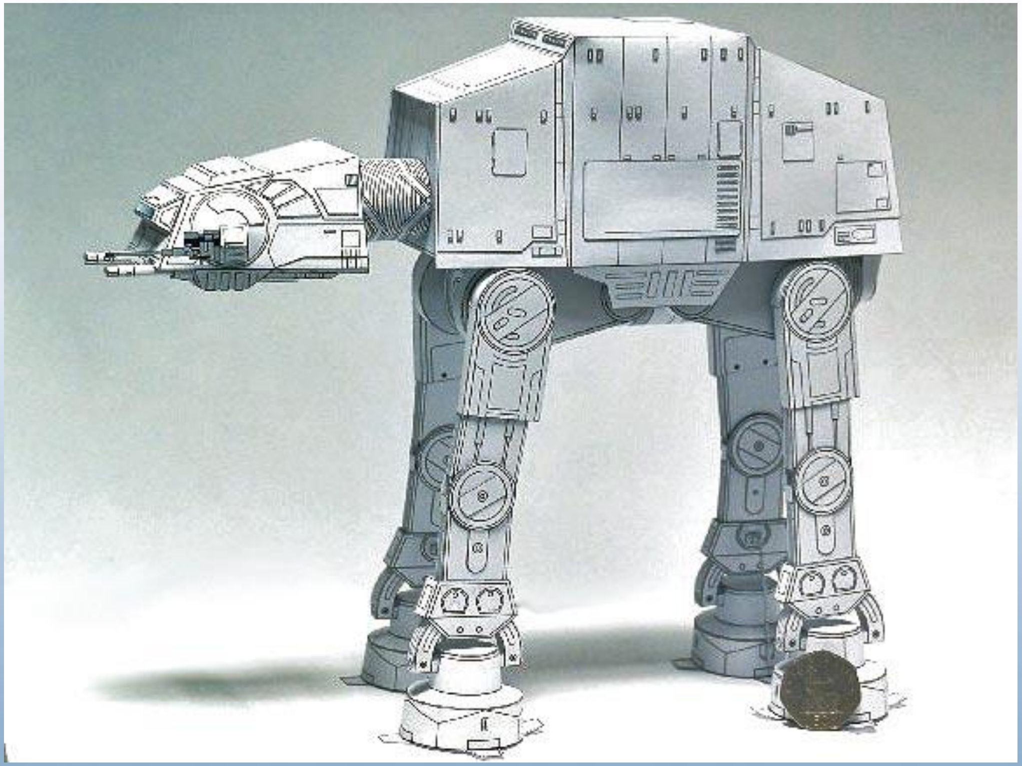
Contents

1. B-rep notions
2. Manifold or nonmanifold?
3. Conditions for B-rep faces
4. Topological conditions:
   Euler formula
5. Winged-edge structure



Contents

6. Euler operators
7. Set operations on B-rep
8. CSG to B-rep conversion
9. B-rep to CSG conversion



B-rep notions

In the boundary representation, a solid is represented by segmenting its boundary into a finite number of bounded subsets usually called “*faces*” or “patches”, and representing each face by its bounding edges and vertices.

B-rep notions

- This description has two parts, a topological description of the connectivity and orientation of vertices, edges, and faces, and a geometric description for embedding these surface elements in space.
- The topological description specifies vertices, edges, and faces abstractly, and indicates their incidences and adjacencies.
- The geometric description specifies, for example, the coordinates of vertices or the equations of the surfaces containing the faces.

B-rep notions

Historically, B-rep evolved from a description of polyhedra in computer graphics:

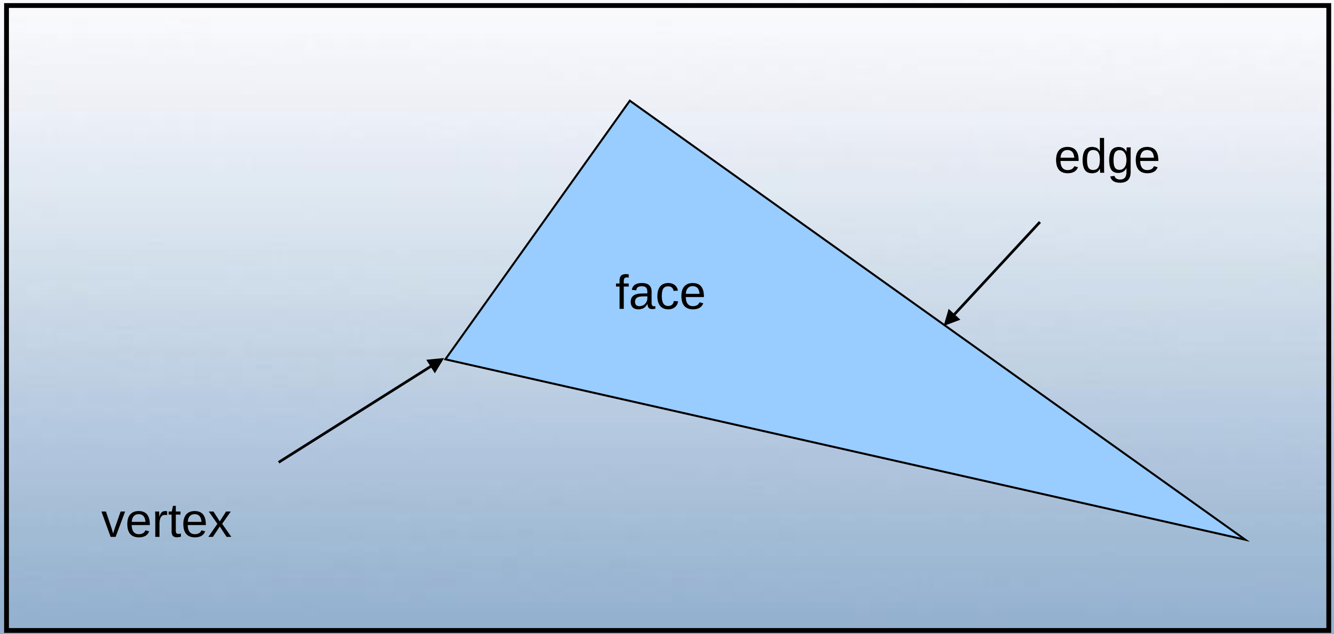
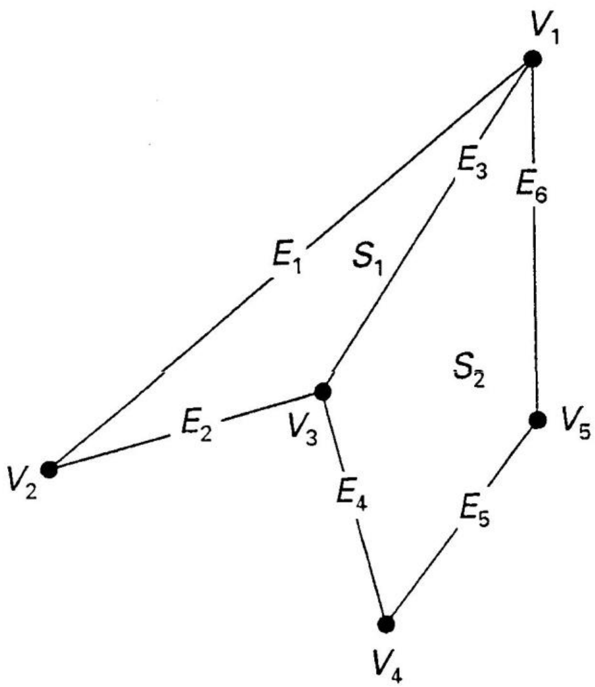


Fig. vertex-edge-face from CG course

B-rep notions

Historically, B-rep evolved from a description of polyhedra in computer graphics:



EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

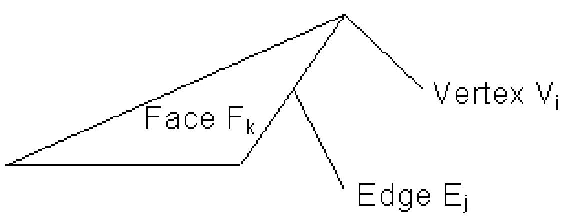
POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

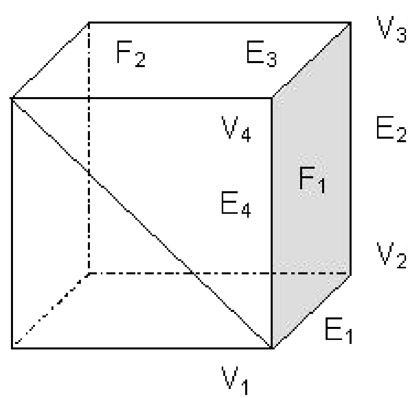
VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

Hearn/ Baker, p.307

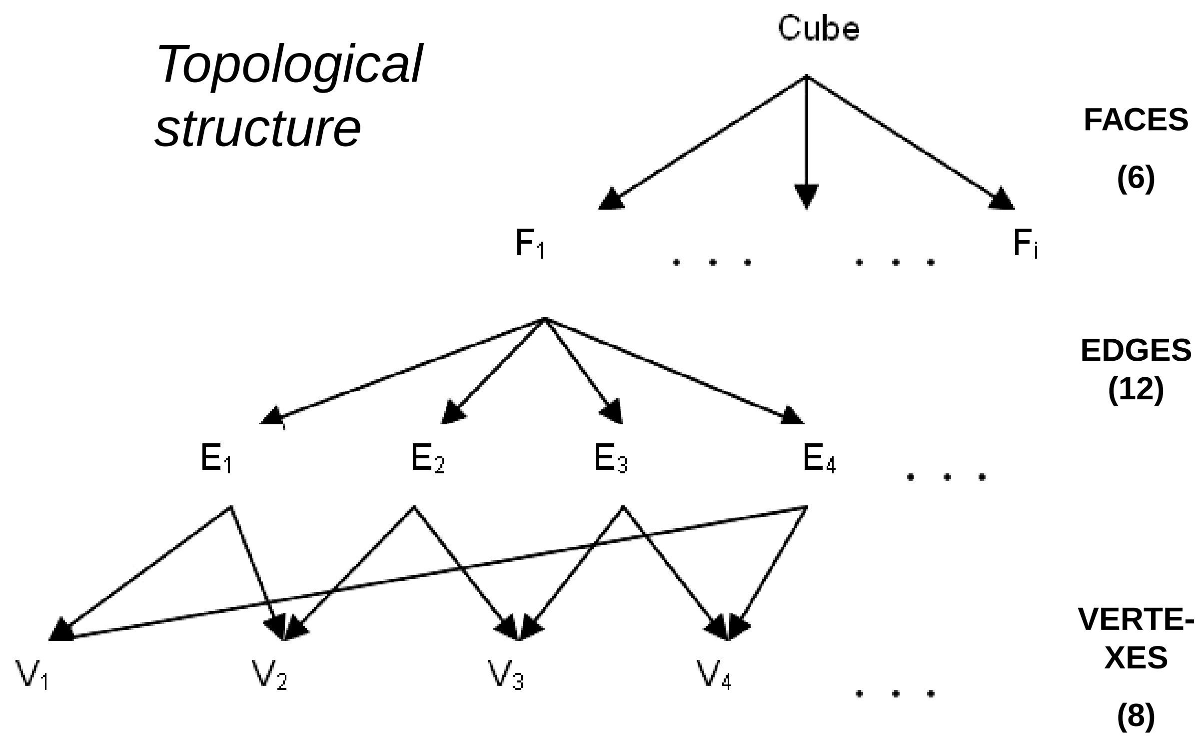
B-rep notions

Example: A boundary representation for a cube





Topological structure



BRep Properties

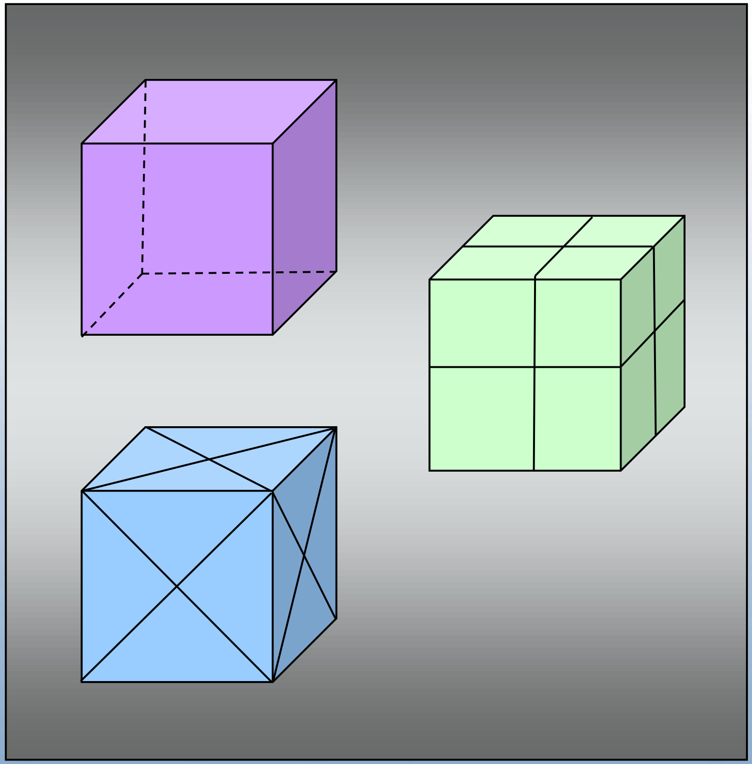
- Domains

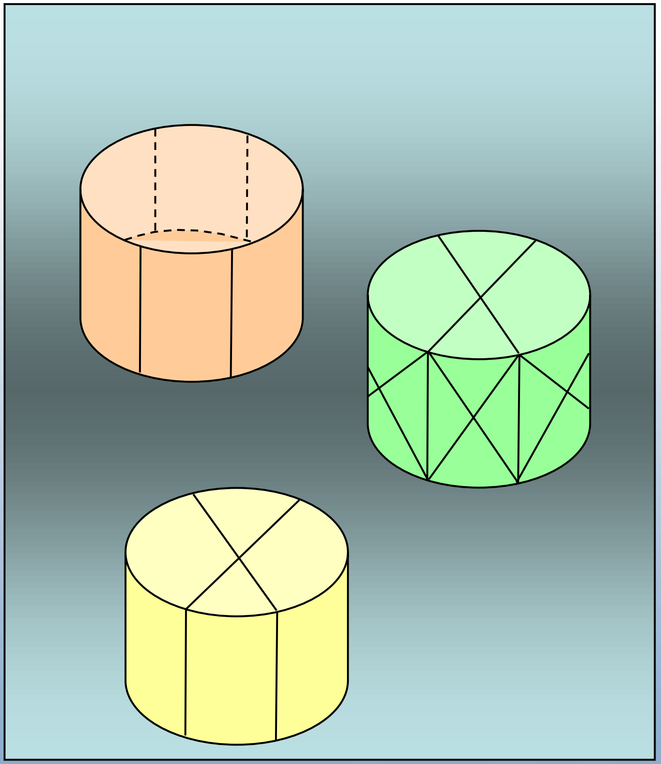
  are as reach as those of other representations.

- Unambiguous

  if faces are represented unambiguously.

- Not unique





Properties

- Validity

  control requires expensive calculations.
- Not concise (verbose)

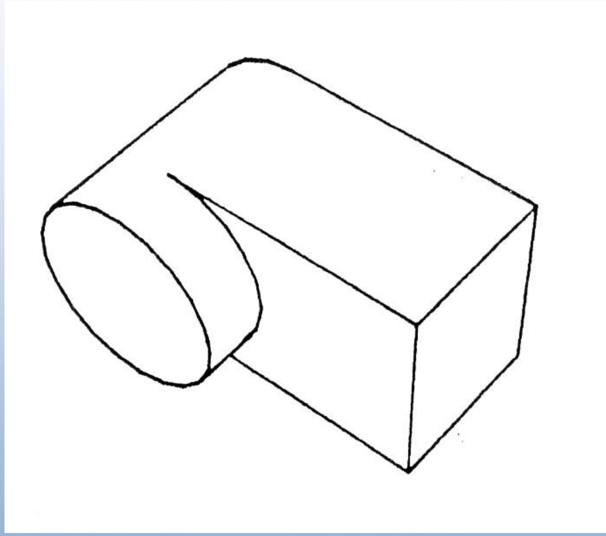
  More than 10 times longer than corresponding CSG.
- Difficult

  for humans to construct.
- Efficient

  in line and shaded drawings, graphic interaction and topological applications.

Vertex - edge - face

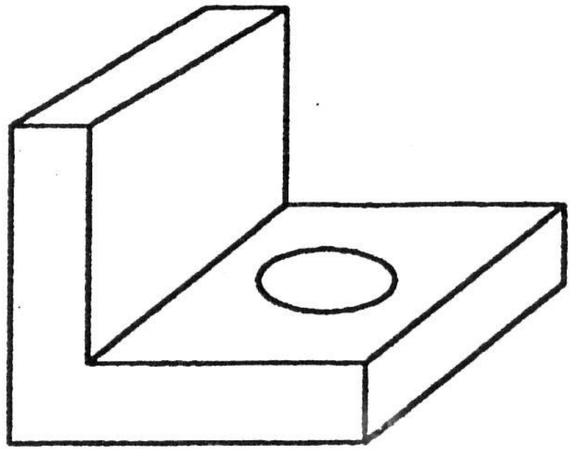
- Types: polygonal and curved faces
- Curved faces can be approximated by polygons or represented by parametric (implicit) surfaces



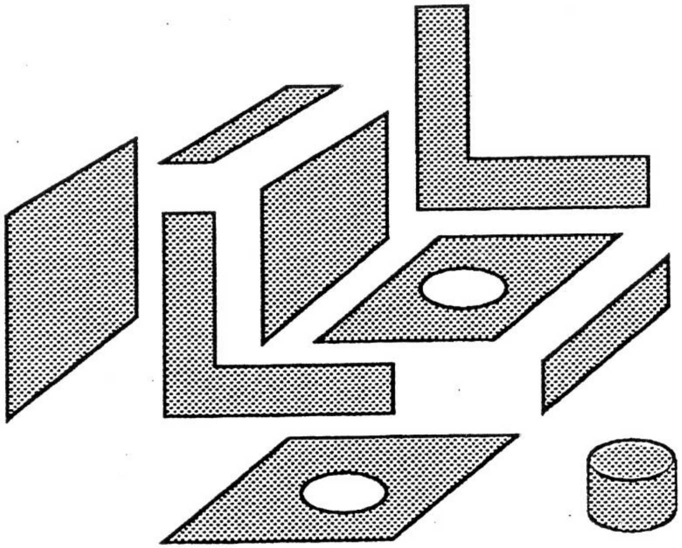
What are the faces?

Requicha, Comp. surveys, 1980, p.453

B-rep example



Solid



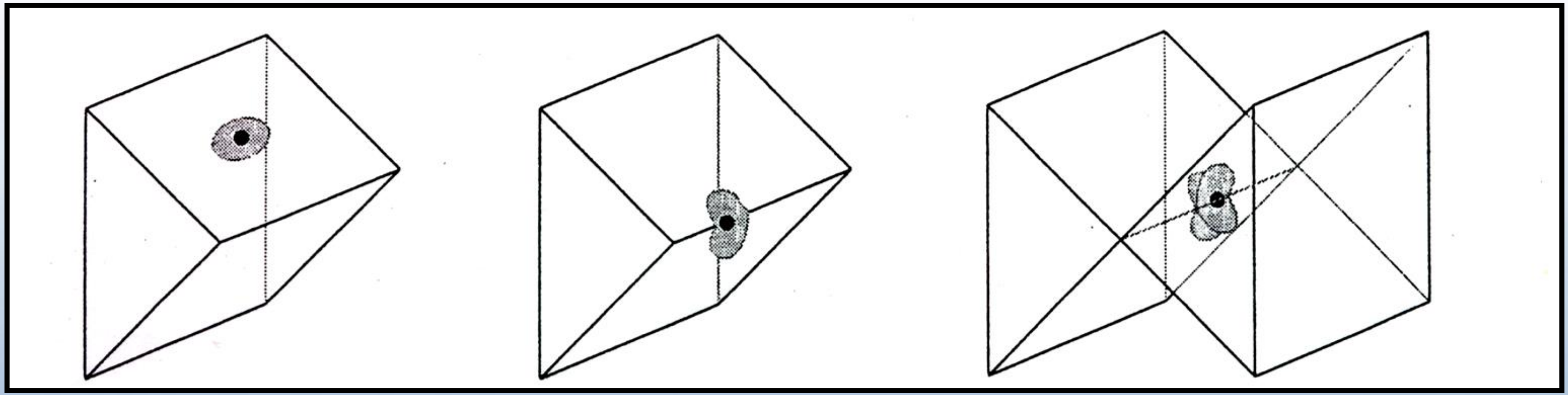
BRep

Requicha, IEEE CG&A, 1983, p.26

2-manifold

- Every point on a 2-manifold has a neighborhood of points around it that is topologically the same as a disk in the plane.
- There is a continuous one-to –one correspondence between the neighborhood and the disk.

Example: if more than two faces share an edge (Figure c), any neighborhood contains points from each of those faces. Thus, the surface is not a 2-manifold.



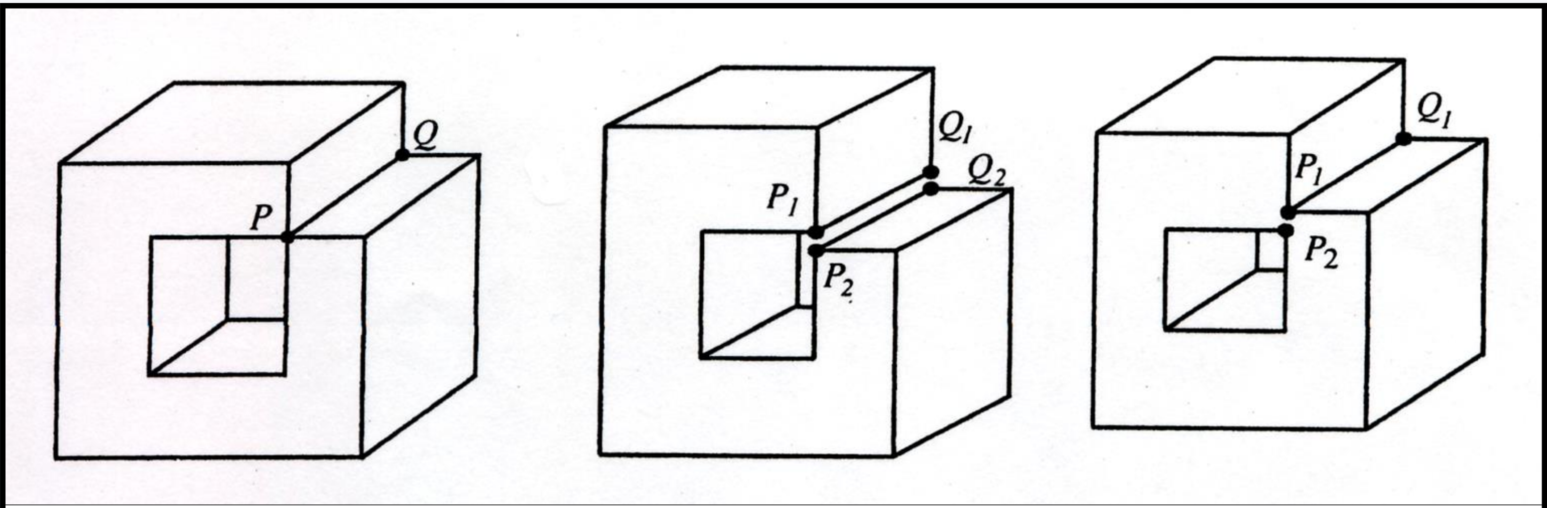
On a 2-manifold, each point, shown as a black dot, has a neighborhood of surrounding points that is a topological disk, shown in gray in (a) and (b). (c) If an object is not a 2-manifold, then it has points that do not have a neighborhood that is a topological disk.

Manifold or nonmanifold?

- Many BRep systems support only solids whose boundaries are closed, oriented 2-manifolds in 3D space. Thus surfaces that intersect or touch themselves are excluded.
- A manifold surface is orientable if we can distinguish two different sides (sphere, torus, etc.). Mobius strip and Klein bottle are nonorientable surfaces.

Manifold or nonmanifold?

- Regularized set-theoretic operations on two manifold objects may result in a nonmanifold object. For example, the union of two L-brackets:



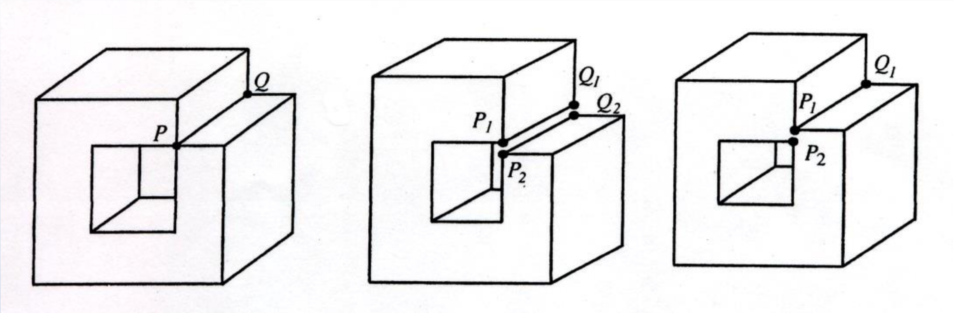
(a) (b) (c)

Hoffmann, p.38

Manifold or nonmanifold?

Three approaches to treating non-manifolds:

- 1. Objects must be manifolds. Operations on solids with nonmanifold results are considered an error.
- 2. Objects are topological manifolds, but geometric description permits coincidence of topologically separate structures (b,c).
- 3. Nonmanifold objects are permitted.



(a) (b) (c)

Conditions for B-rep faces

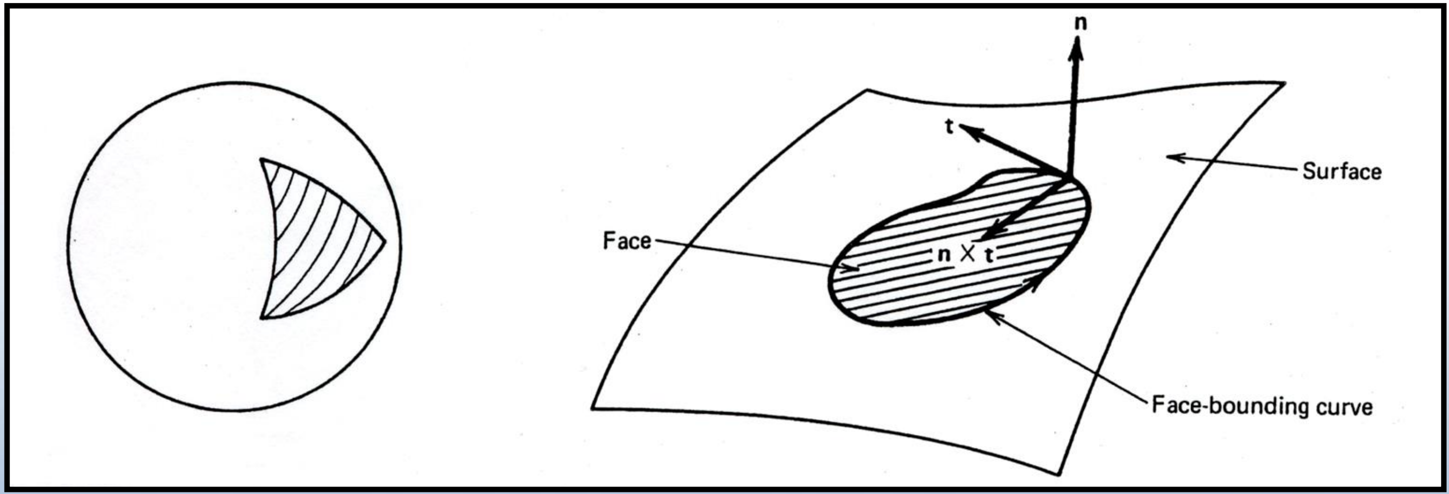
Faces should satisfy the following conditions:

- 1. A finite number of faces defines the boundary of a solid.
- 2. A face of an object is a subset of the object’s boundary.
- 3. The union of all faces of an object defines its boundary.
- 4. A face is itself a subset or limited region of some primitive surface.
- 5. A face must have a finite area and must be dimensionally homogeneous (must not have dangling edges or isolated points).

Conditions for B-rep faces

Faces must be represented unambiguously.

What is the face below?



Bounding edges of the face are oriented according to some convention. For example, a face-bonding curve is parameterized in a consistent direction so that the vector $\boldsymbol{n} \times \boldsymbol{t}$ points to the face side of the curve.

Requicha, comp. surveys, 1980, p.453
Mortenson, p.471

Polyhedra and Euler's Formula

- A 3D polyhedron is a solid that is bounded by a set of polygons:

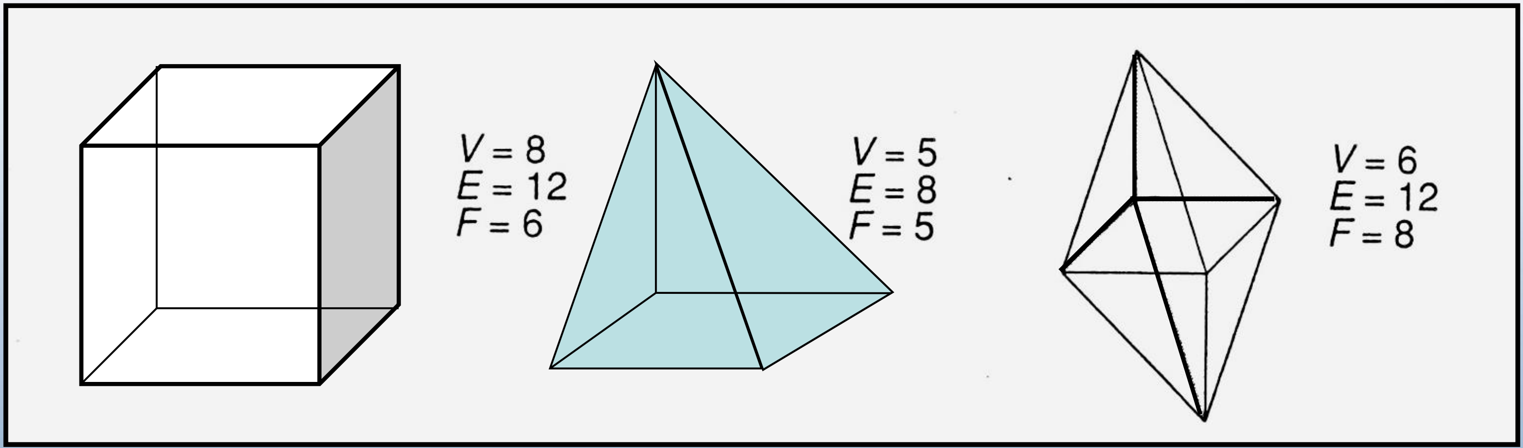
- each edge connects two vertices and is shared by exactly two faces

- at least three edges meet at each vertex

- faces do not interpenetrate.

Polyhedra and Euler's Formula

- A simple polyhedron can be deformed into a sphere (no holes).



Some simple polyhedron with their V, E and F values.

Foley, p.545

Polyhedra and Euler's Formula

- The BRep of simple polyhedron satisfies Euler's formula:

$$\mathbf{V - E + F = 2}$$

where

V is the number of vertices

E is the number of edges

F is the number of faces.

Generalized Euler's Formula

- The BRep of 2-manifolds that have faces with holes satisfies the generalized Euler's formula:

$$V - E + F - H = 2 (C - G)$$

where

V is the number of vertices

E is the number of edges

F is the number of faces

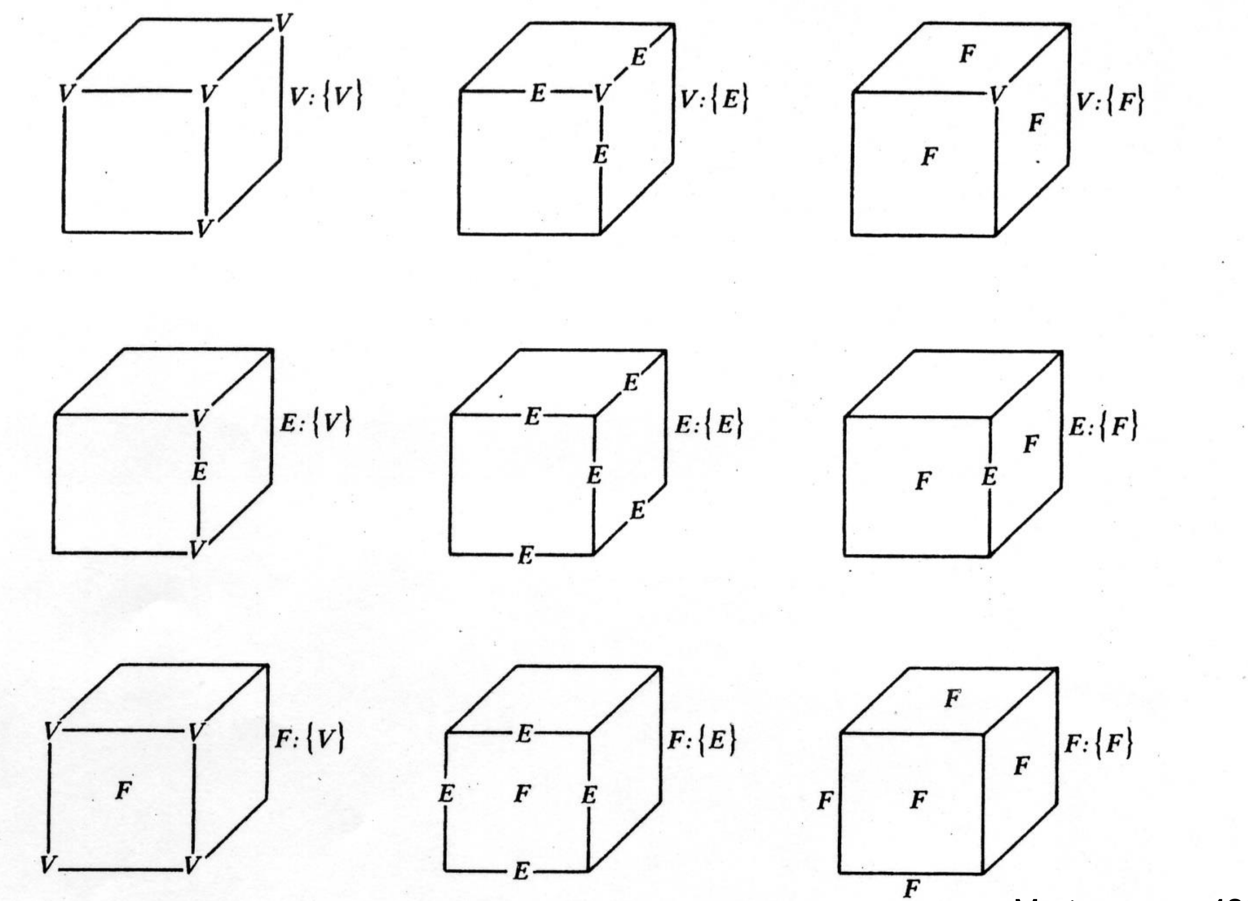
H is the number of holes in the faces

C is the number of separate components (parts)

G is the genus (for a torus G = 1)

Topological relationships in BRep

A polyhedron has nine classes of topological relationships between pairs of elements: vertices, edges, and faces.



Mortenson, p.426

Topological relationships in B-rep

Different applications need different adjacency information:

- *V:* {*V*}, *E:* {*V*}, *F:* {*V*} in wireframe (vector) graphics to know how vertices are joined

- *V:* {*F*} in set operations to know the ring of faces around the vertex

- *F:* {*F*} adjacency among faces is needed in Euler operators.

Winged-edge structure

This data structure represents the boundary of a manifold polyhedral object. The topological information is as follows:

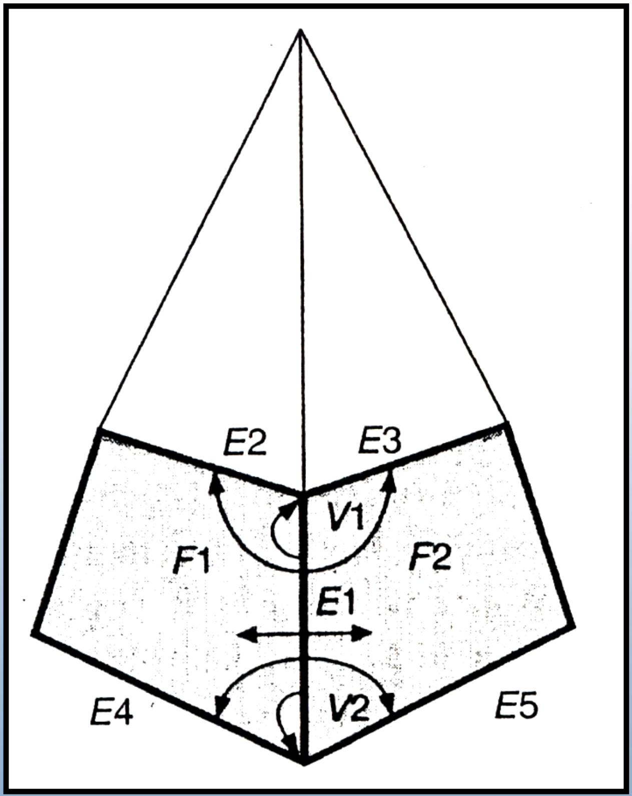
- Each face is bounded by a set of disjoint edge cycles. One cycle is the outside boundary of the face, the others bounding holes.
- Each vertex is adjacent to a circularly ordered set of edges, so the vertex table specifies one of these edges for each vertex.

Winged-edge structure

- For each edge the following information is given (see figure):

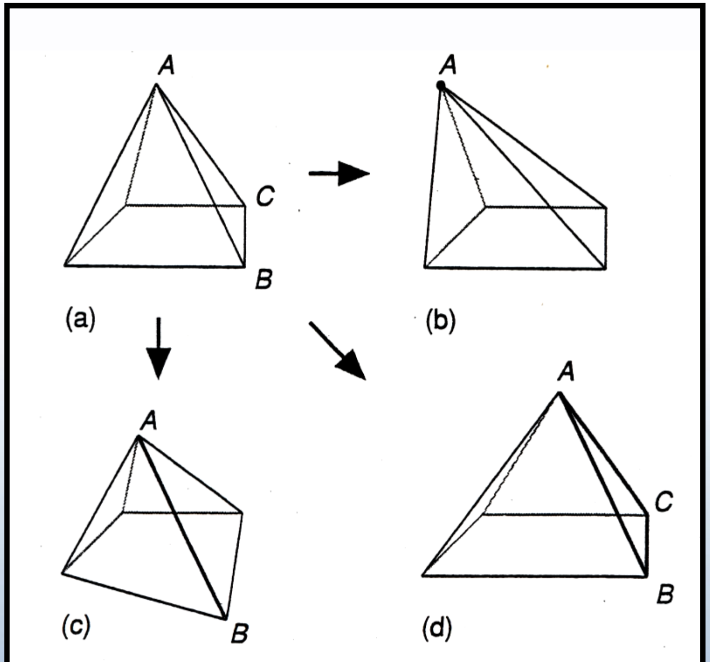
1) Incident vertices (V1, V2)
2) Left and right adjacent face (F2, F1)
3) Two edges that share V1 (E2, E3)
4) Two edges that share V2 (E4, E5)

This structure makes it possible to determine in constant time which vertices or faces are associated with an edge.

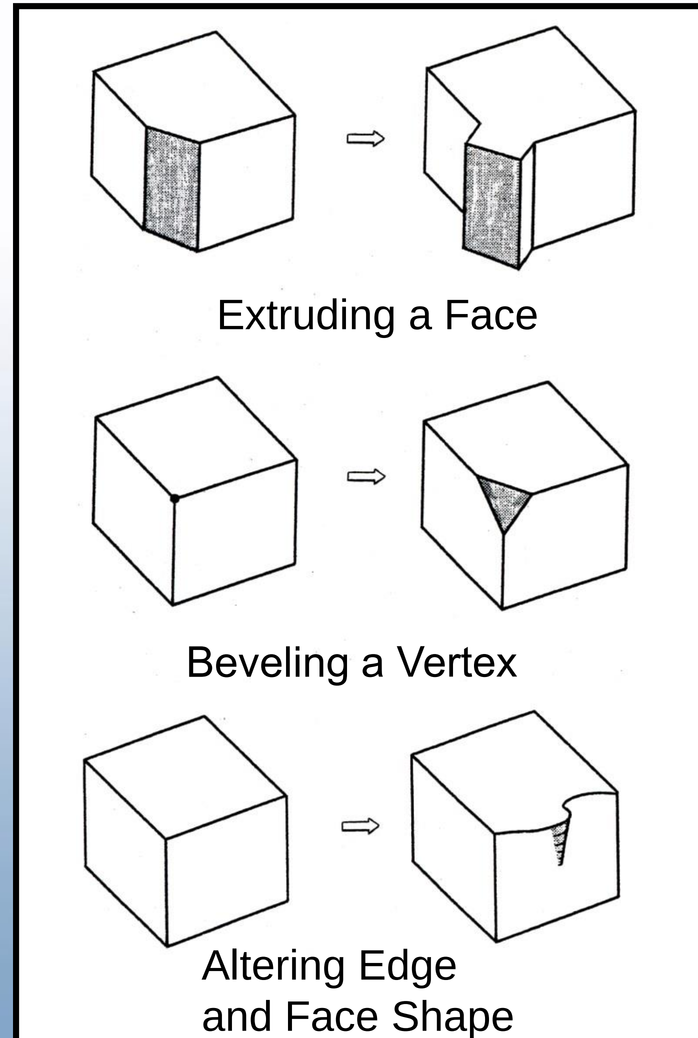


Foley, p.545

Local modifications



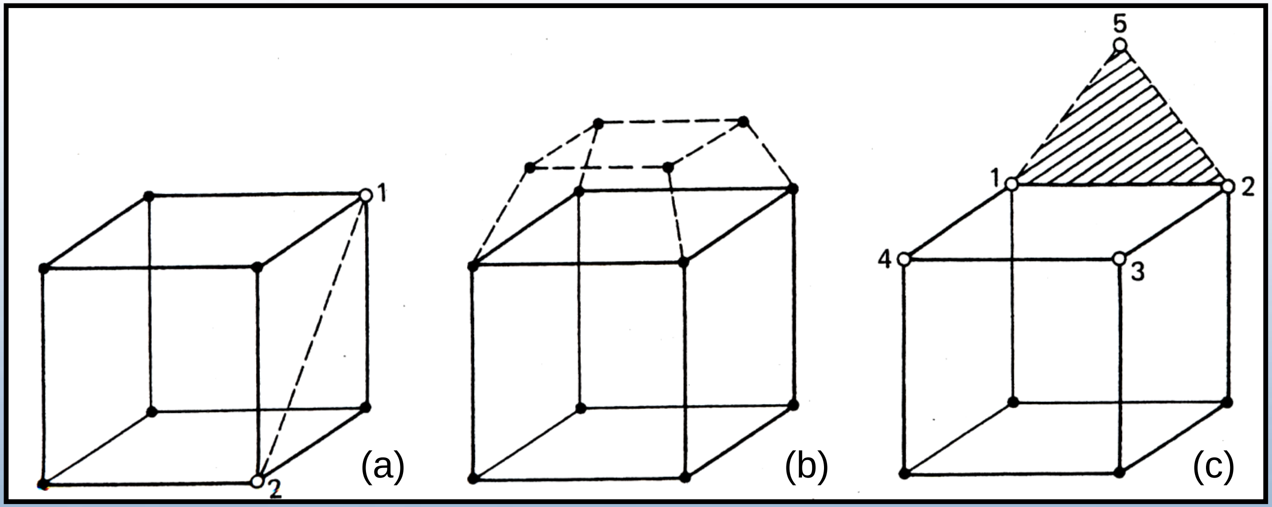
(a) An object on which tweaking operations are performed to move, (b) vertex A, (c) edge AB, (d) face ABC



Foley, p.544
Hoffmann, p.18

Euler operators

Euler operators transform the objects satisfying Euler’s formula by adding and removing vertices, edges and faces.



(a, b) a cubical polyhedron is correctly modified;
(c) the result is not a polyhedron because edges (1,5) and (2,5) are not shared by two faces each.

Mortenson, p.421

Euler operators

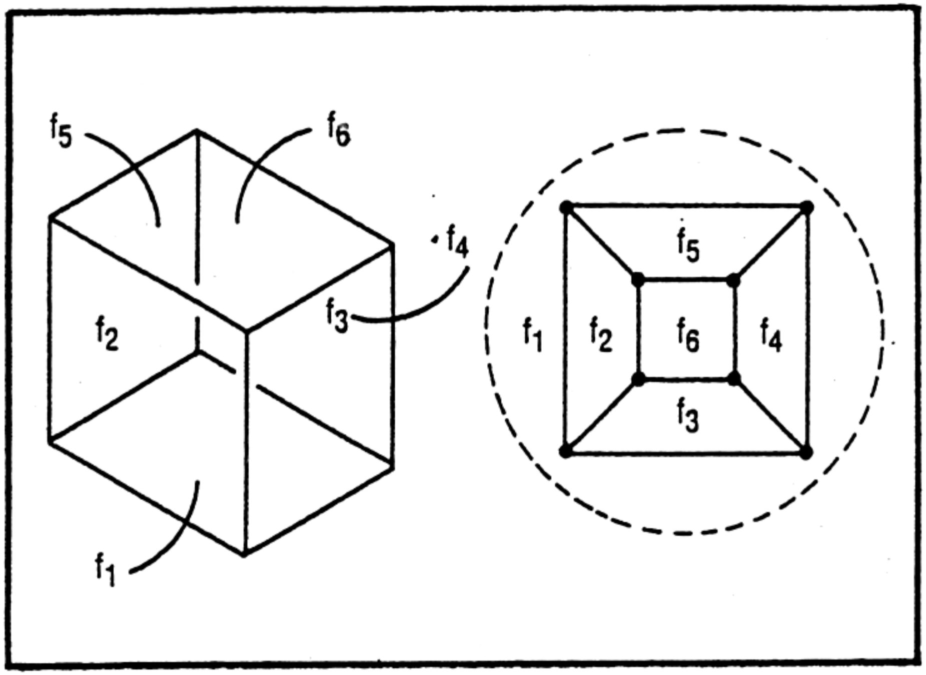
A linear combination of five primitive operators (with their inverses) can represent all objects satisfying Euler formula:

1. Make (kill) an edge and a vertex (***mev*** / ***kev***)
2. Make a face and an edge (***mfe / kfe***)
3. Make a body, a face, and a vertex (***mbfv / kbfv***);
4. Make a cavity, or passage, and a body (***mrb / krb***);
5. Make an edge and kill a hole (***me-kh***).

Advantages of Euler operators:

- Ensured topological validity of the resulting solids
- Intermediate language isolating high-level operations from the underlying data structures

Euler operators in the GWB system

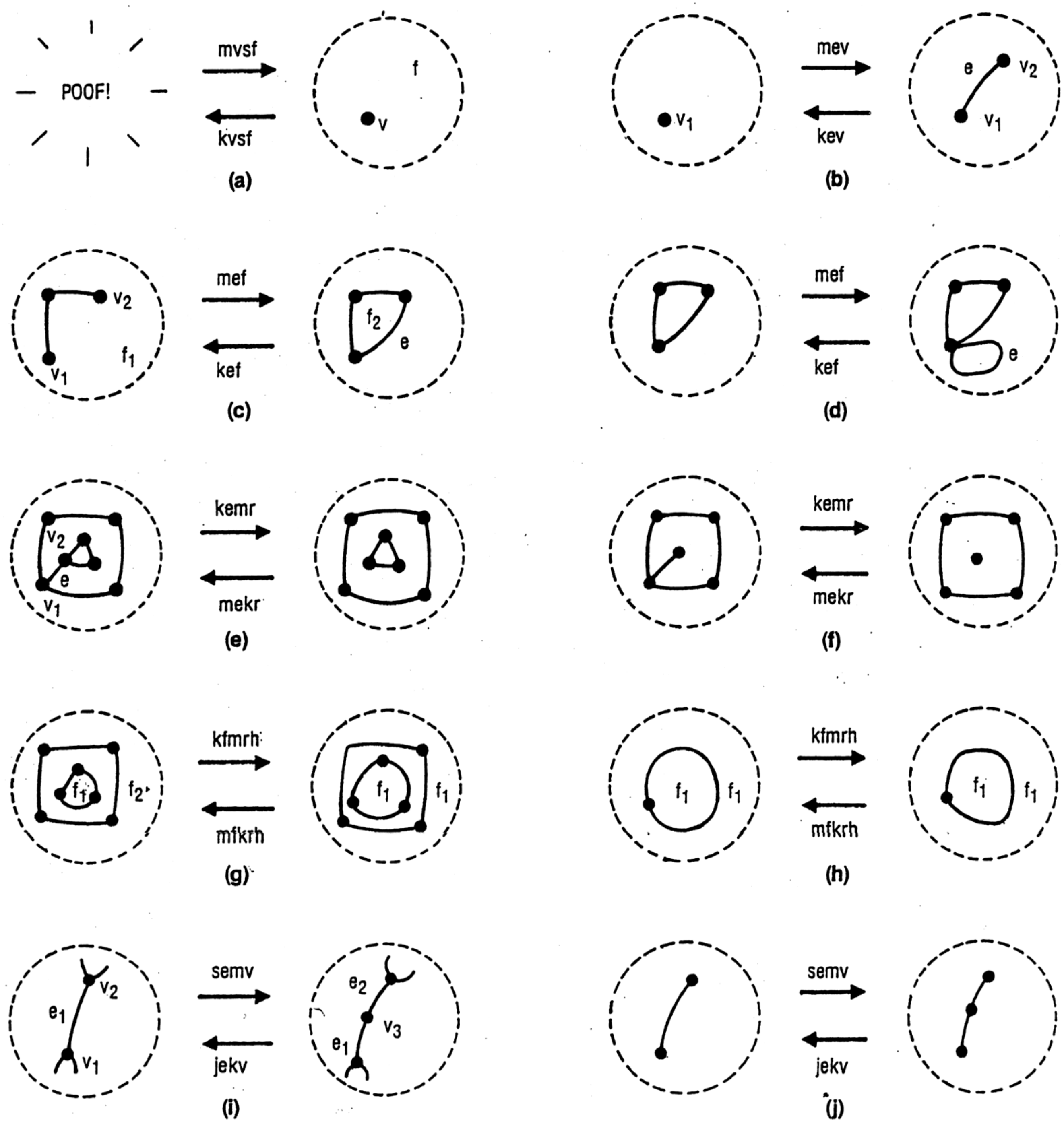


Cube topology: a plane model

Euler operators

OPERATOR	EXPLANATION
mvsf(f,v)	MAKE VERTEX, SOLID, FACE
kvsf()	KILL VERTEX, SOLID, FACE
mev(v_1,v_2,e)	MAKE EDGE, VERTEX
kev(e,v)	KILL EDGE, VERTEX
mef(v_1,v_2,f_1,f_2,e)	MAKE EDGE, FACE
kef(e)	KILL EDGE, FACE
kemr(e)	KILL EDGE, MAKE RING
mekr(v_1,v_2,e)	MAKE EDGE, KILL RING
kfmrh(f_1,f_2)	KILL FACE, MAKE RING, HOLE
mfkrh(f_1,f_2)	MAKE FACE, KILL RING, HOLE
semv(e_1,v,e_2)	SPLIT EDGE, MAKE VERTEX
jekv(e_1,e_2)	JOIN EDGES, KILL VERTEX

Euler operators in the GWB system



Mantila, CG&A, 1982, p.20

Set operations on BRep

- General approach: generate and test algorithm
- Solid *S* that results from a Boolean operation can be computed as follows:

- first generating a superset of its boundary as the union of the boundary faces of the solids being combined

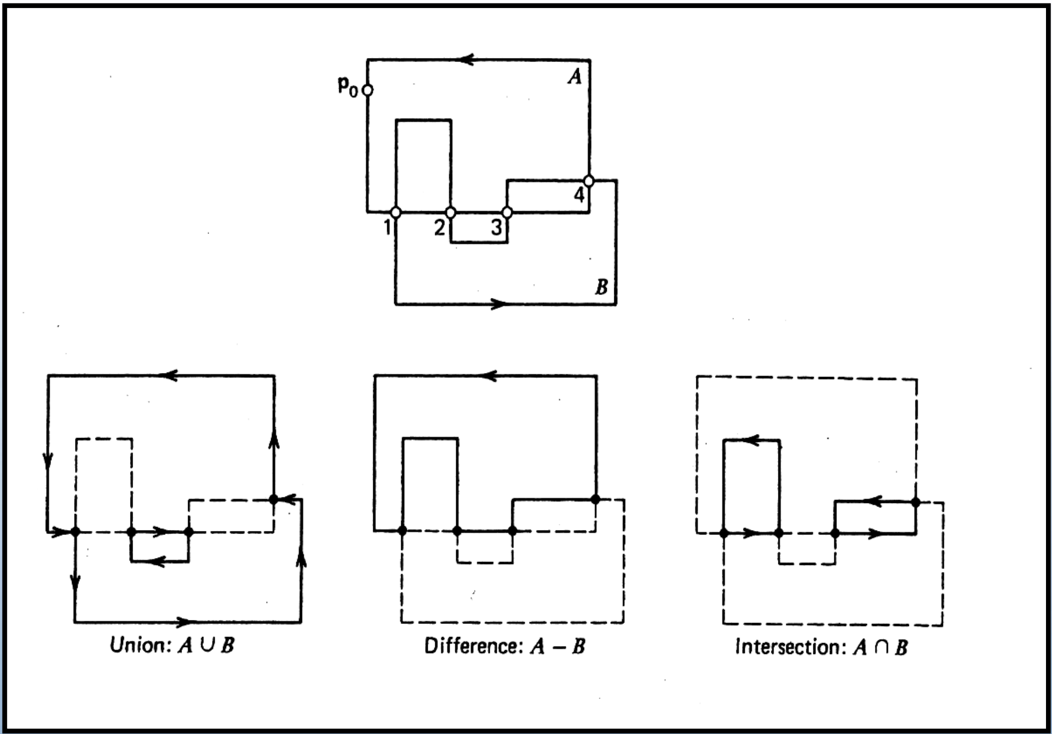
- discarding those faces that are not *on S*.

Mortenson, p.404

Set operations on BRep: 2D polygons

The steps of the algorithm for finding $A \cup B$:

1. Find all intersection points of the edges of *A* and *B* (points 1, 2, 3, 4)
2. Segment the edges of *A* and *B*. If the boundary of *A* is parametrized from *u=0* to *u=1*, then it has four segments: [u1, u2], [u2, u3], [u3,u4], [u4,u1]

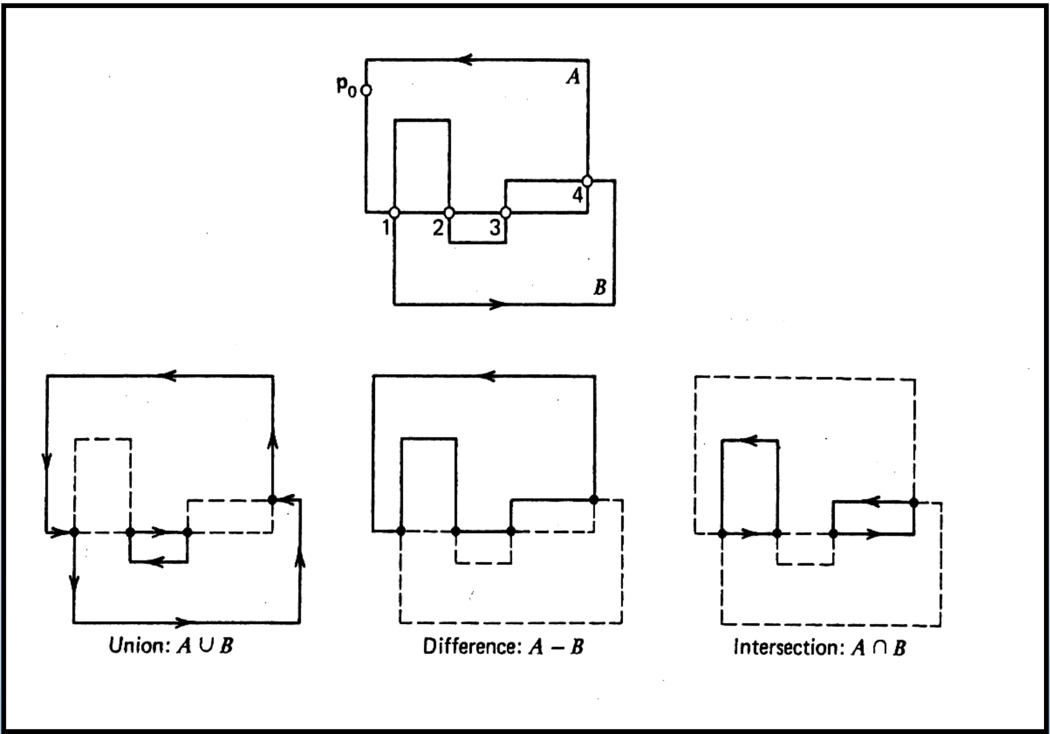


Mortenson, p.404

Set operations on BRep: 2D polygons

3. Find a point ***p0*** on *A* that is outside of *B*. Then that segment (here [*u4*,*u1*]) is also outside *B*.

4. Start at ***p0*** and trace *A* to the next intersection with *B* (point 1).



5. Trace this segment of B to its intersection with A (point 4). We have found one loop, but have not checked all segments.

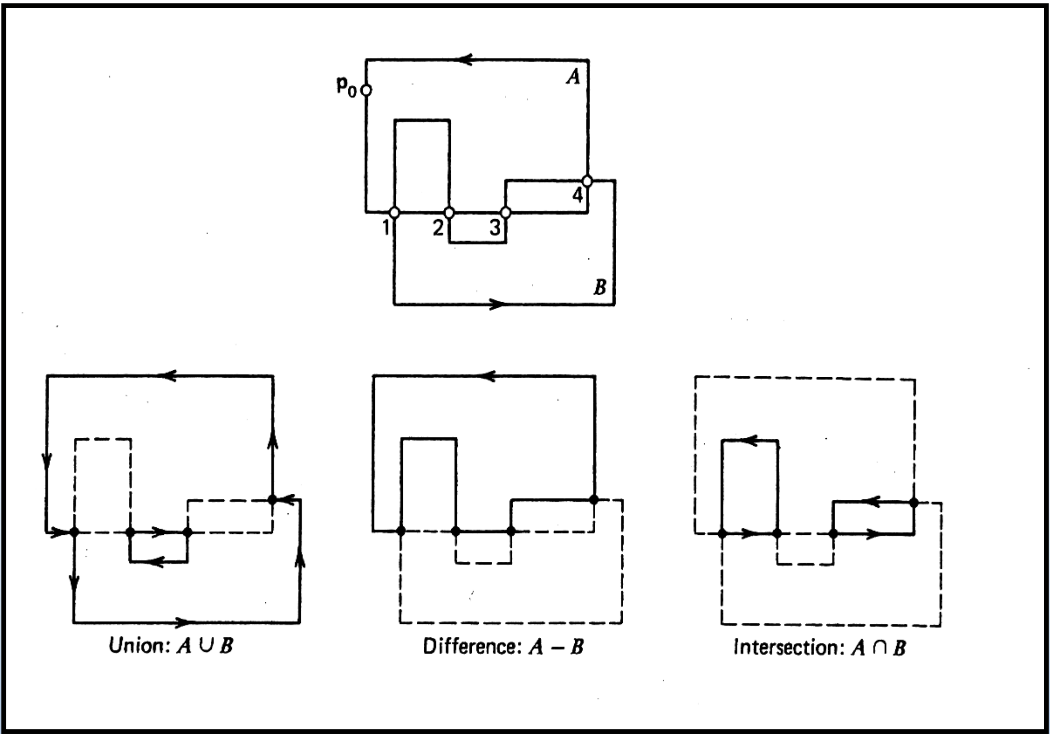
Mortenson, p.404

Set operations on BRep: 2D polygons

6. Repeat 3 and find the segment [*u2, u3*]

7. Repeat 4 and find point 3

8. Repeat 5 and find point 2. We have found another loop.



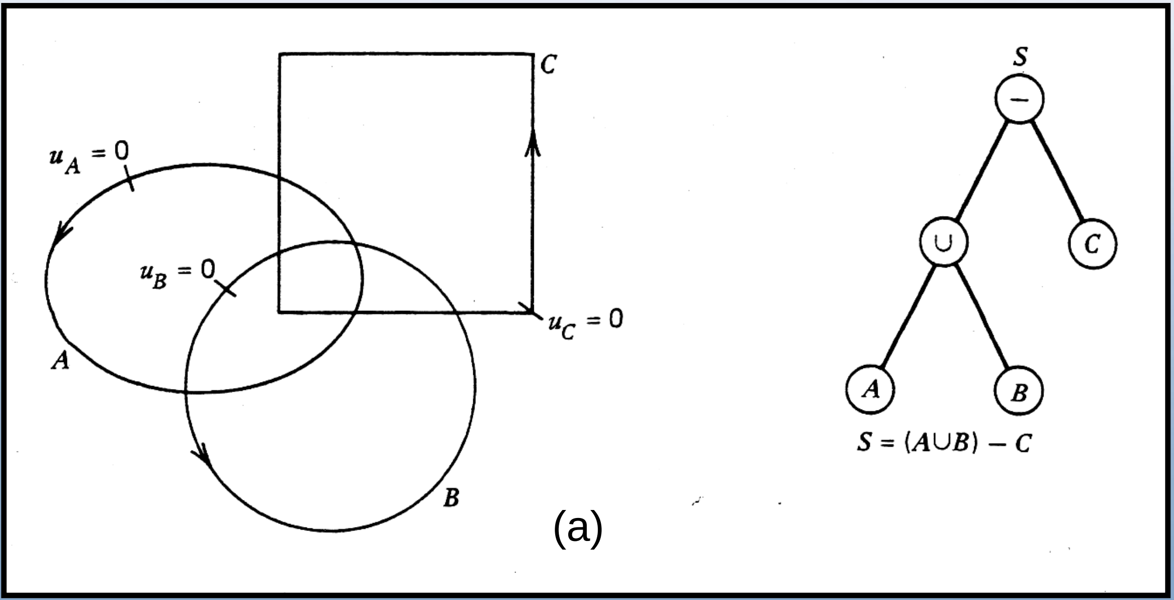
Active segments of *A*: [*u4,u1*], [*u2, u3*]
of *B*: [*v1,v4*], [*v3,v2*]

Mortenson, p.404

Set operations: 2D parametric curves

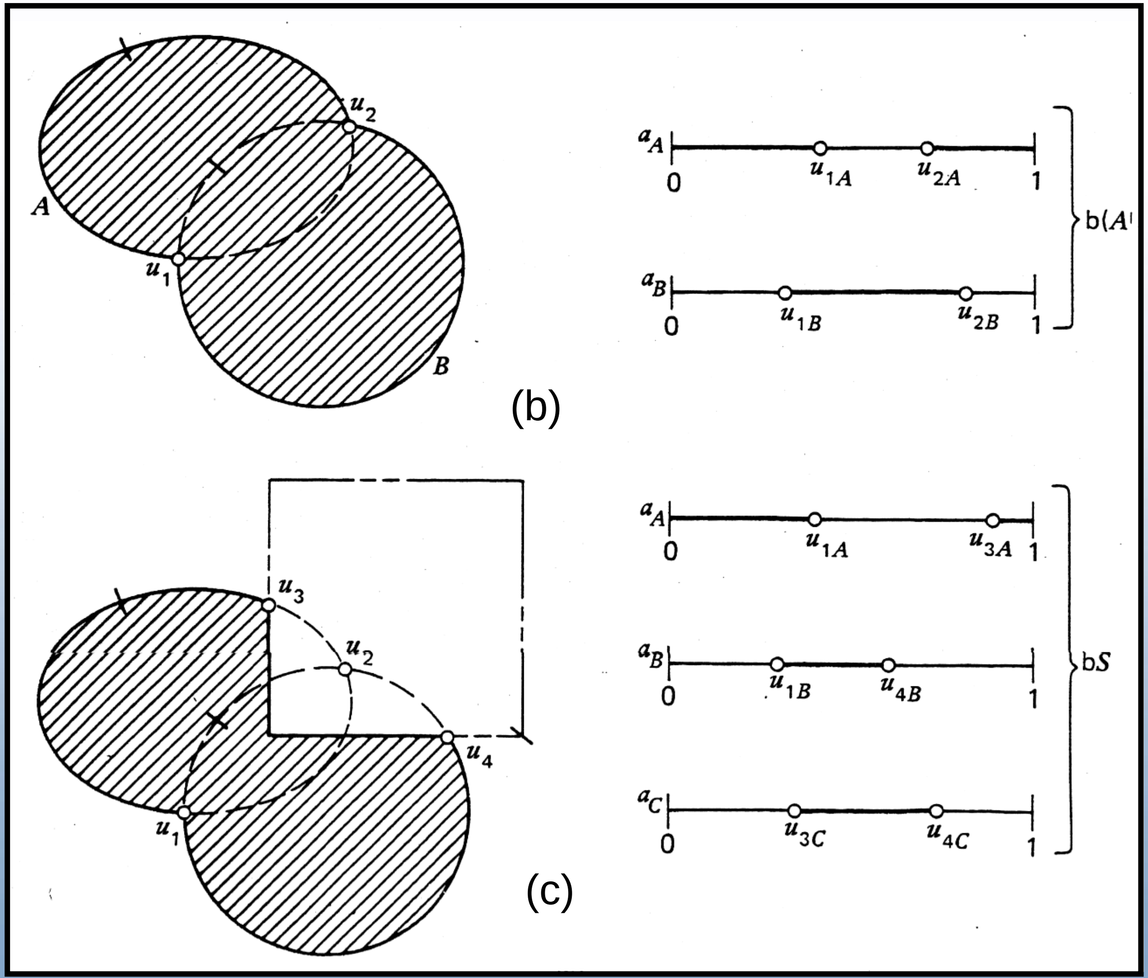
The idea is to find active regions of the bounding curves. These regions are defined by intersection points of primitives.

Example: $A \cup B - C$



(a)

Set operations:2D parametric curves

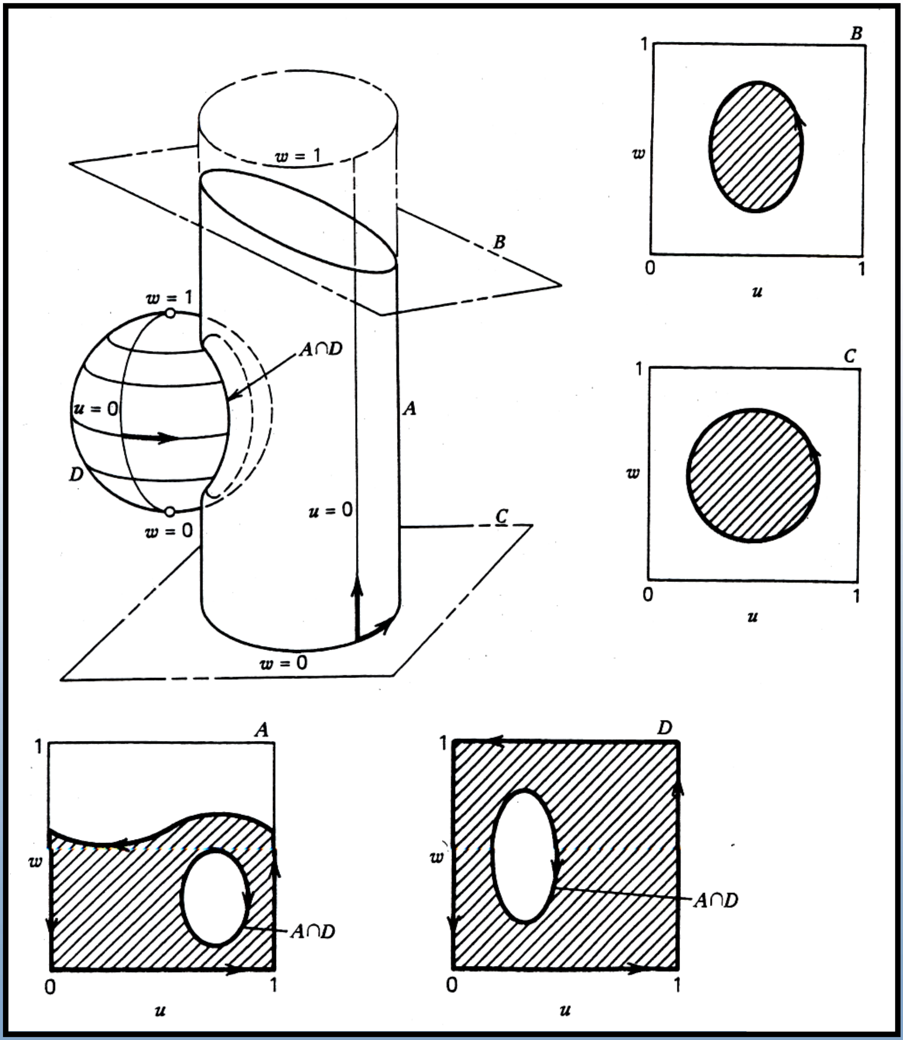


Mortenson, p.474

Set operations: 3D parametric surfaces

Points bound active regions on curves, and curves bound active regions on surfaces. The active surface regions (faces) on all the primitives define a closed surface of a solid.

Example: *(A ∩ B ∩ C) ∪ D*
The active regions of the surfaces are shaded in the *uw*-plane.



Mortenson, p.476

Point membership classification

Problem statement:

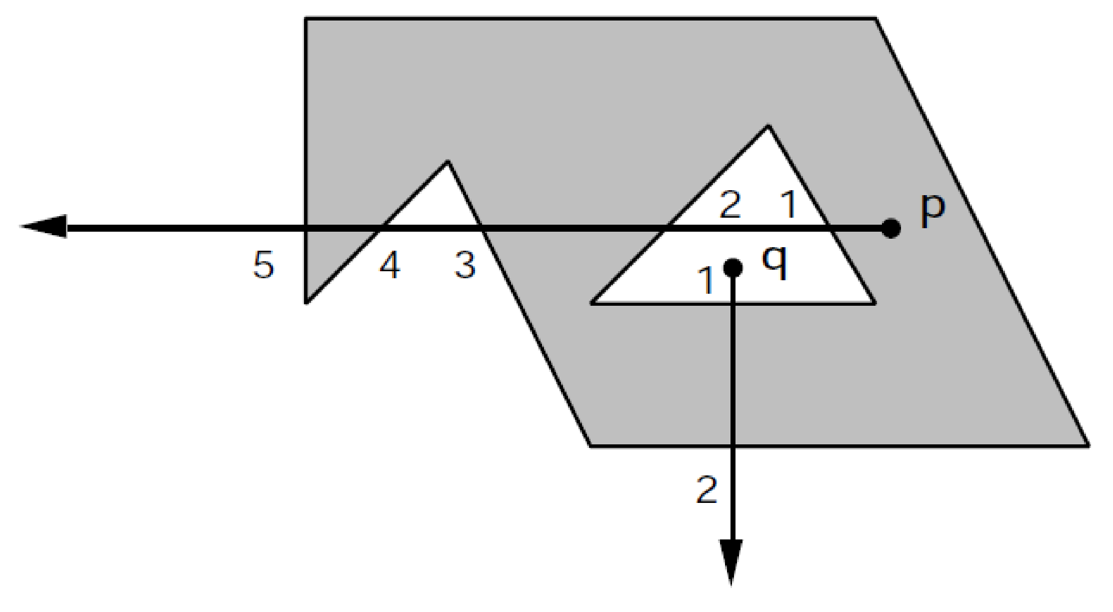
For the given point in space and a BRep solid model detect whether the point is inside, outside or on the boundary of the solid.

Standard algorithm:

Casting a ray from the given point in arbitrary direction and counting how many times it intersects the solid's boundary.

Point membership classification

Casting a ray: if the number of ray-surface intersection points is odd, the given point is *in*; if it is even, the point is *out*.



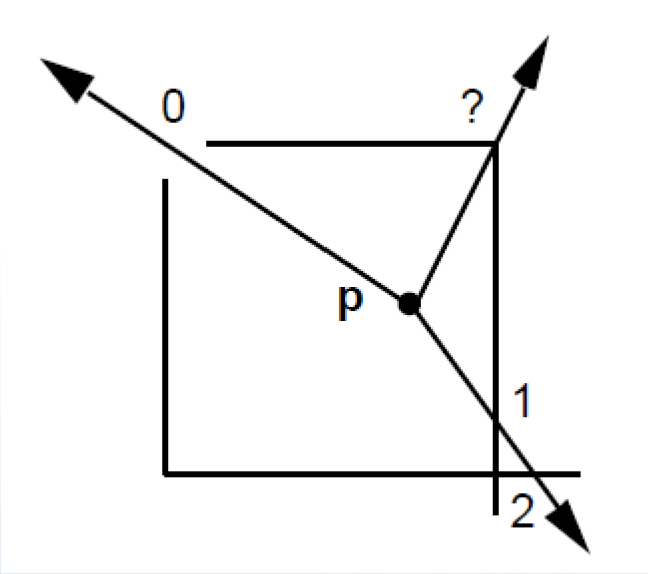
The ray cast from *in* point **p** has 5 intersections with the polygon's boundary, whereas the ray from *out* point **q** has 2.

Point membership classification

Problems with casting a ray:

- How do we count intersections when the given point is *on*?
- Numerical errors associated with the intersection calculations may produce wrong counts.
- Ray may intersect an edge or a vertex, or partially lie in a face or an edge. How do we count intersections in such singular cases?

Point membership classification



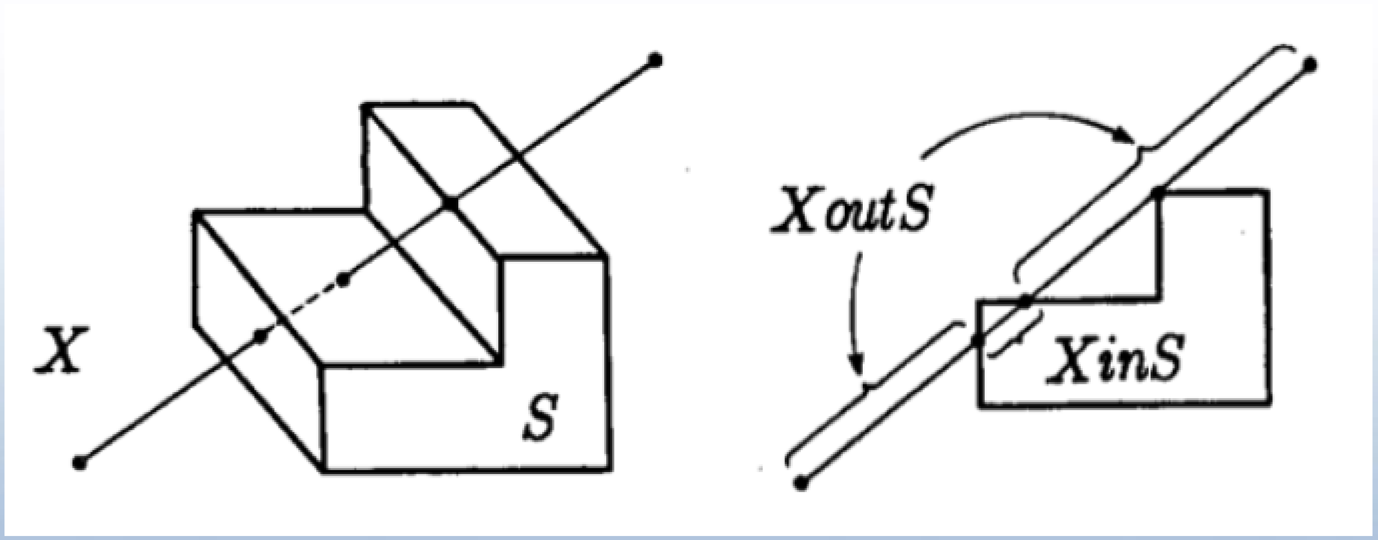
Ray 0: no intersection found

Ray 1: two intersection points

Ray ?: intersection with a vertex - should we count one or two intersections?

Point membership classification

Casting a ray in 3D: case of the ray intersecting an edge.



Point membership classification

Solutions:

- Choose a ray that does not intersect any

vertices or edges and does not lie in any faces or edges.

- Cast several rays in different directions and use the value that occurs most often.
- Use more complex algorithms:
  First intersection point analysis [Requicha 1996]
  Pseudo normals [Baerentzen 2005]
  Generalized winding number [Jacobson 2013]

CSG to B-rep conversion

Exact conversion from CSG to boundary representations is called boundary evaluation and consists of the following steps:

1. Consider all pairs of intersecting primitives in the CSG tree

2. For each pair, obtain a set of space curves in which they intersect

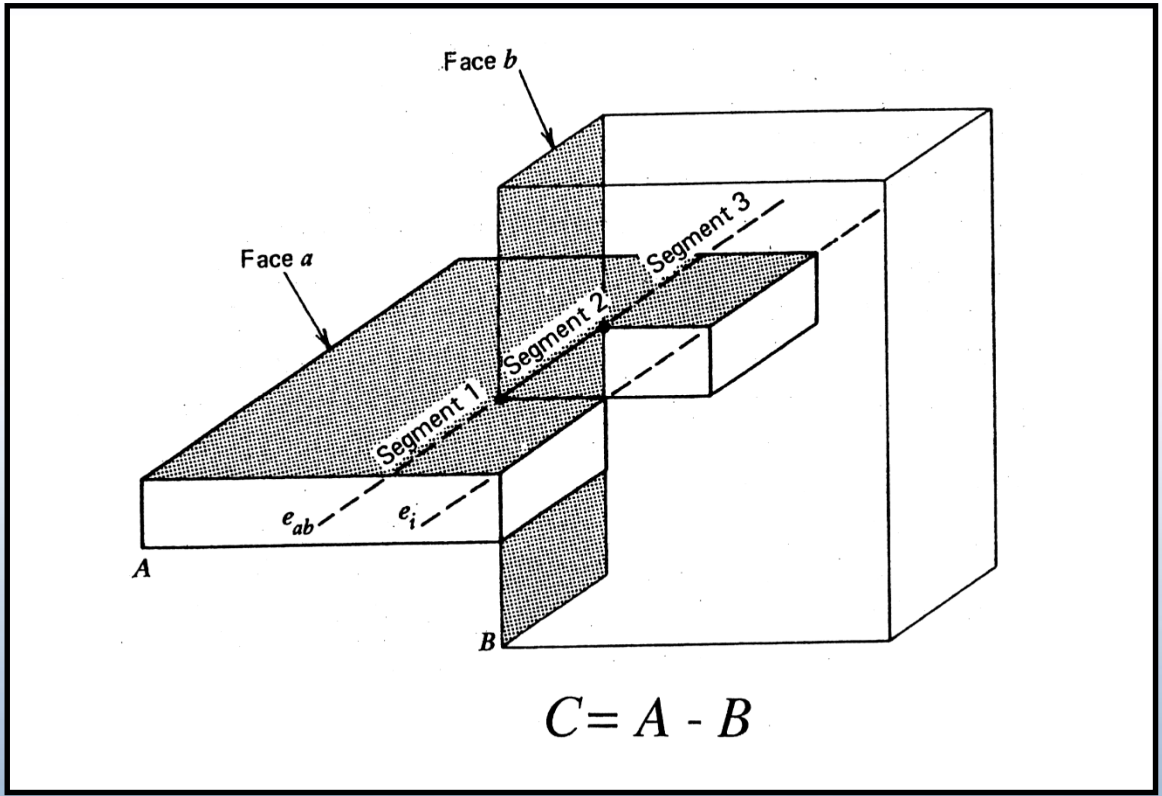
CSG to B-rep conversion

3. Classify each curve against the solid, we can determine those segments that are on the boundary of the solid. Each segment will be an edge of the boundary representation.

4. These segments now define, on the surface of the primitives, faces of the boundary representation.

5. By considering the neighborhoods, we derive the topological relationships between different faces.

CSG to B-rep conversion



Edge e_{ab} = *Face a* $\cap$ *Face b*. Segment 2 of *Edge* e_{ab} is on the boundary of *C* $\rightarrow$ it is an edge of a face of *C*.

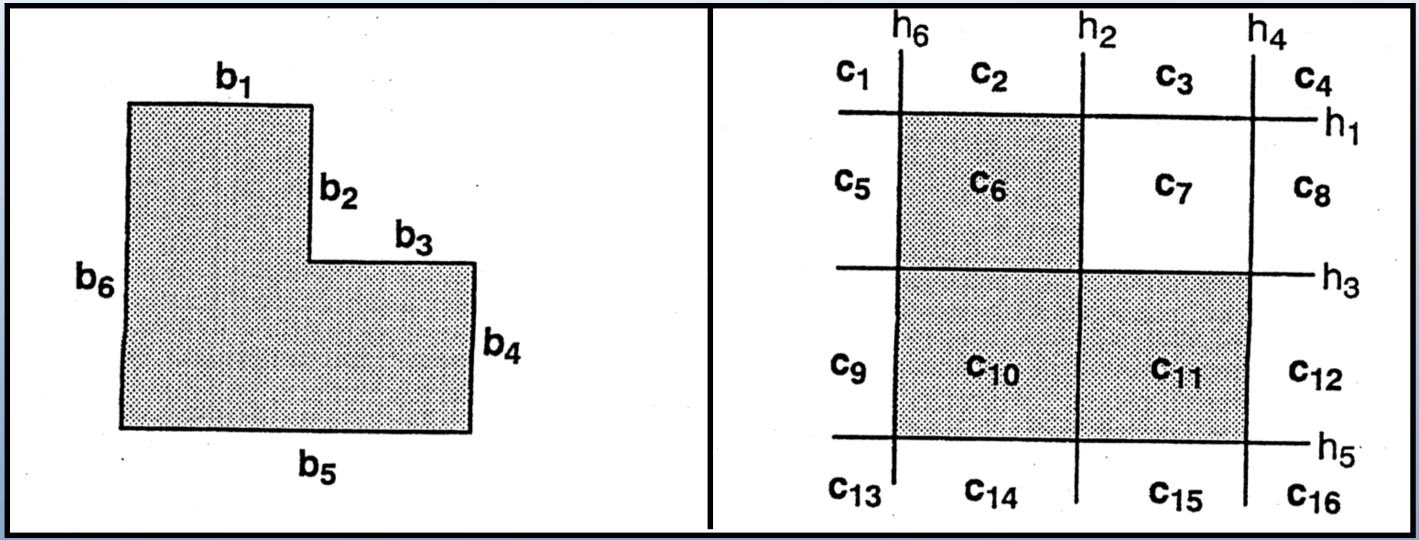
Mortenson, p.466

B-rep to CSG conversion

The first phase (geometric phase) has two steps:

1) Decompose the whole 3D model space using halfspaces.

The halfspaces are constructed from the face geometry of the B-rep model. The halfspaces divide the model space into a set of cells.

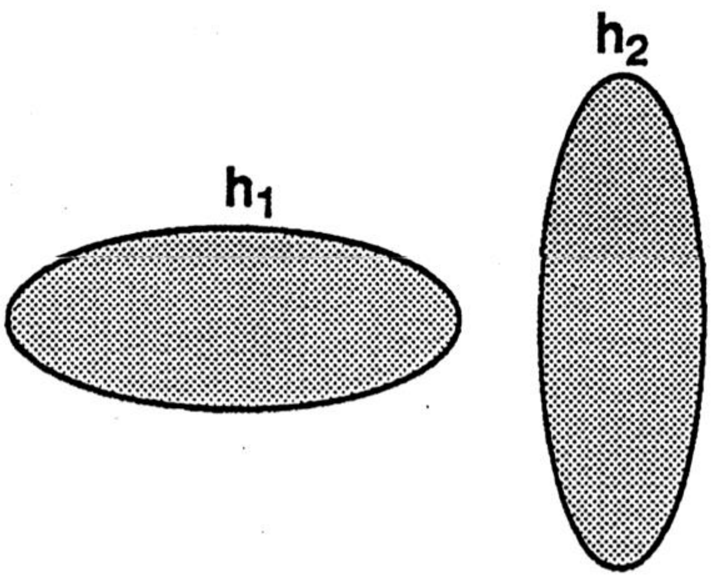


Paraphrase, No.10, Nov 1994, p.12

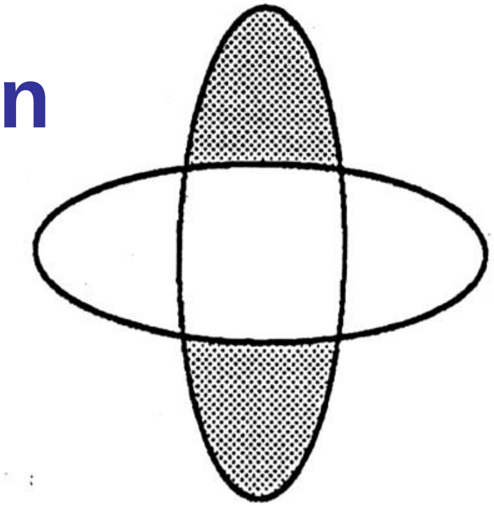
B-rep to CSG conversion

Paraphrase, No.10, Nov 1994, p.12

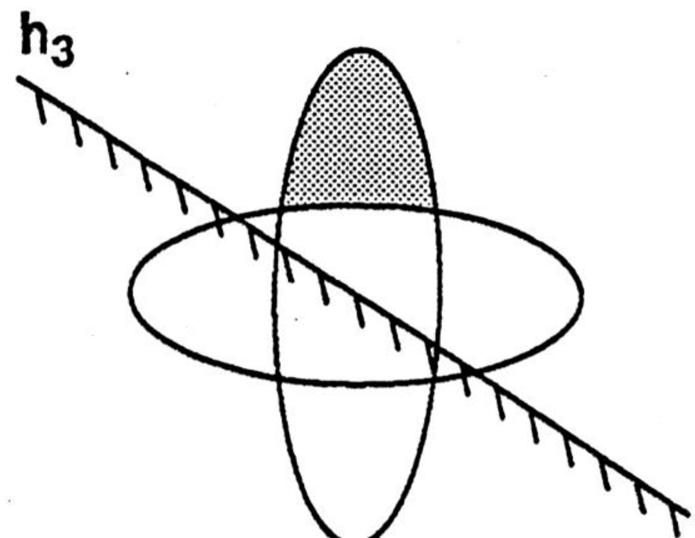
Complications are introduced by curvature. Suppose we want to represent the shape shown above, which is bounded by elliptical edges. (Again we are using a 2D analogy.)



To represent it in CSG we need two elliptical halfspaces h_1 and h_2.

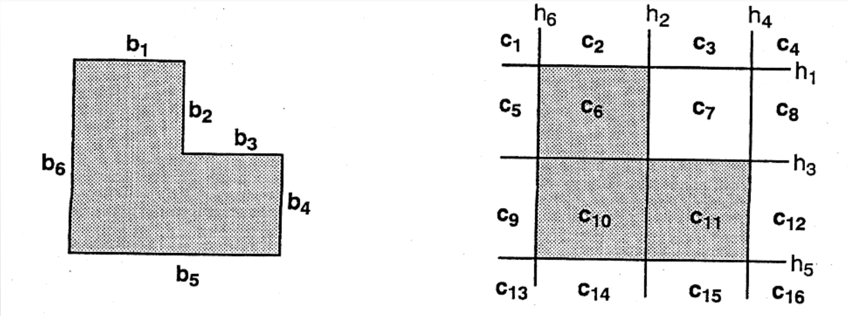


We can describe the above shaded areas as the intersection of the area *outside* h_1 and the area *inside* h_2.



B-rep to CSG conversion

2) Classify each of the cells as being occupied by the solid or being empty space. In the figure, C6, C10, and C11 are occupied by the solid. An initial CSG representation can be obtained by union of these cells.



- The second or combinatorial phase involves simplifying or minimizing the initial CSG tree. The algorithm applies well-known Boolean optimization techniques from digital logic design.

BCSG package

- BCSG is a system for converting boundary representations of solids ('B-reps') to efficient constructive ('CSG') representations.

  BCSG is a C language implementation of the Vossler (EDS Unigraphics) / Shapiro (Cornell) procedure.

  BCSG handles solids having up to two dozen natural-quadric (planar, cylindrical, spherical, conical) faces.

  It accepts B-reps represented as Parasolid 'bodies', and converts them to CSG text representations defined in the PADL-2 language.

References

- Michael E. Mortenson
  “Geometric Modeling”, John Wiley & Sons, 1985, 763 p.

- Christoph M. Hoffmann
  “Geometric and Solid Modeling. An Introduction”, Morgan Kaufmann Publishers, 1989, 338 p.

- Martti Mantila and Reijo Sulonen
  “GWB: a solid modeler with Euler operators”, IEEE Computer Graphics and Applications, vol.2, No.7, 1982, pp.17-31.

References

- Kevin Weiler

 “Edge-based data structures for solid modeling in curved-surface environments”, IEEE Computer Graphics and Applications, vol.5, No.1, 1985, pp.21-40.

- Kevin Weiler

 “The radial edge structure: a topological representation for non-manifold geometric boundary modeling”, in Geometric Modeling for CAD Applications, M.J.Wozny et al (Eds.), Elsevier Science Publishers B.V., 1988, pp.3-36.

References

- Shapiro, V., and Vossler, D. L.
  "Construction and optimization of CSG representations," Computer-Aided Design, vol. 23, no. 1, pp. 4-19, Jan/Feb, 1991.
- Requicha A., Geometric Modeling: A First Course, on-line book, 1996-2011 http://www-bcf.usc.edu/~requicha/book.html
- Baerentzen J. and Aanaes H., 2005. Signed distance computation using the angle weighted pseudonormal. IEEE Transactions on Visualization and Computer Graphics, 11(3), 243{253.

References

- Jacobson A., Kavan L. and Sorkine-Hornung O., 2013. Robust insideoutside segmentation using generalized winding numbers. ACM Transactions on Graphics, 32(4), 33:1{33:12.