Solids

Solid modeling techniques are based on informationally complete, valid & unambiguous representation of objects. Simply stated, a complete geometric data representation of an object, if it is inside, outside or on the object. This clarification is sometimes called spatial addressability. A solid model of an object is a more complete representation than its surface model. It stores more information (geometry & topology) than wireframe or surface modelers (geometry only). It is unique in the topological information it stores which potentially permits functional automation & integration. For example, the mass property calculations or finite element mesh generation of an object can be performed fully automatically. In many cases solid models are easier to build than wireframe or surface models.

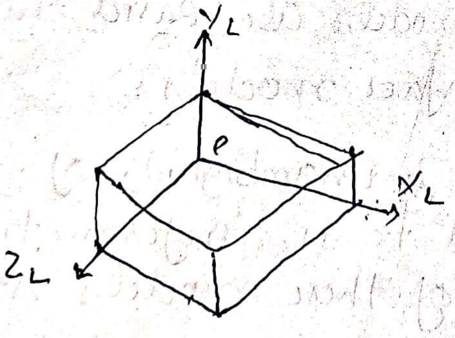
The completeness & unambiguity of solid models are attributed to the information that the related databases of these models store.

Solid Entities:-

Most commercially available solid modeling packages have a CSG (constructive Solid geometry) user input & therefore provide users with a certain set of building blocks, often called primitives. Primitives are simple basic shapes & are considered the solid modeling entities which can be combined by a mathematical set of boolean operations to create the solid. Primitives themselves are considered valid "off-the-shelf" solids.

There is a wide variety of primitives available commercially to users. However, the four most commonly used are the block, cylinder, cone & sphere. These are based on the four natural quadrics; planes, cylinders, cones & spheres. For example, the block is formed by intersecting six planes. Following are descriptions of the most commonly used primitives.

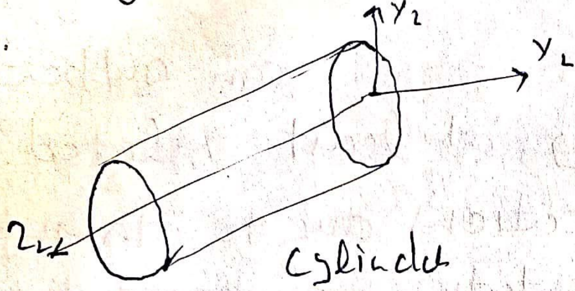
1. **Block**:- This is a box whose geometrical data is its width, height & depth. Its local coordinate system X_L, Y_L, Z_L is shown below. Point P defines the origin of $X_L Y_L Z_L$ system.



Block

2. **Cylinder**:- This primitive is right circular cylinder whose geometry is defined by its radius (or diameter) R & length H. The length H is usually taken along the

direction of the Z_L axis. H can be +ve or -ve.



Cylinder

3. Cone :-

This is right circular cone or frustum of a right circular cone whose base radius R, top radius (for truncated cone) & height H are userdefined.

4. Sphere :-

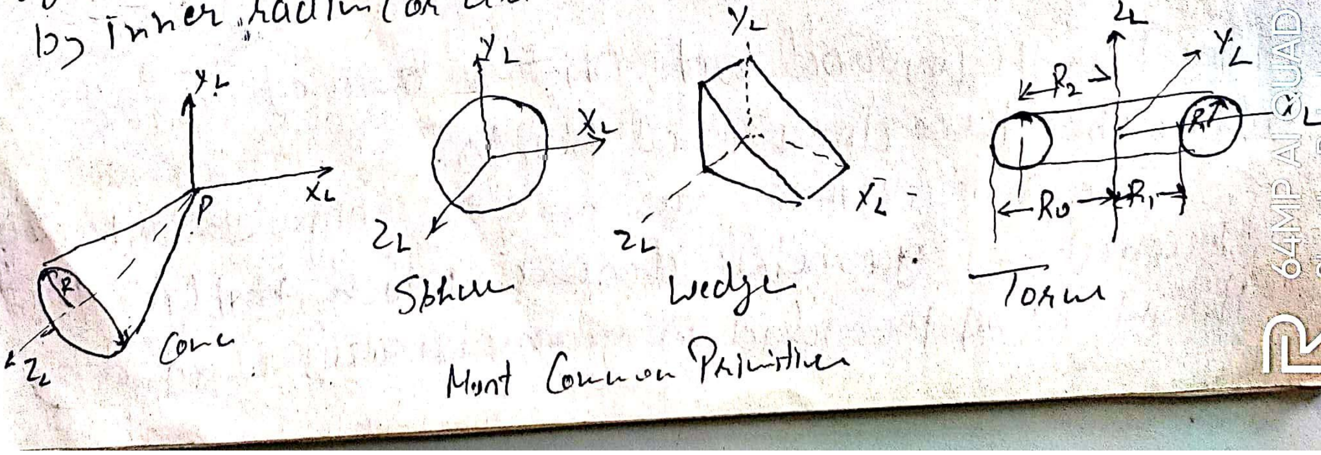
This is defined by its radius or diameter & is centered about the origin of its local coordinate system.

5. Wedge :-

This is a right-angled wedge whose height H, width W & base depth (D) form its geometric data.

6. Torus :-

This primitive is generated by the revolution of circle about an axis lying in its plane. The torus geometry can be defined by ~~its~~ the radius (or diameter) of the centerline of its body R_1 & radius (or diameter) of the torus body R_2 or the geometry can be defined by inner radius (or diameter) R_1 & outer radius (or diameter) R_0.



Cone Sphere Wedge Torus

Most Common Primitives

Solid Representation :-

Solid representation of an object can support reliably & automatically, at least related de & manufacturing applications due to its informati completeness. Such representation is based fundamenta on the notion that a physical object devides an n dimensional space, E^n, in to two regions : interior & exterior separated by the object boundaries. A region is denned as portion of space E^n & the boundary of a region is closed surface, as in case of a sphere, or a collection of open surfaces connected at proper edges, as in the case of a box.

In terms of the above notion, a solid model of an object is defined mathematically as a point set S in three dimensional Euclidean space (E^3). If we denote the interior & the boundary of the set by iS & bS respectively, we can write

$$S = iS \cup bS \qquad ①$$

& if we let the exterior be defined by cS [complement of S], then

$$W = iS \cup bS \cup cS \qquad ②$$

where W is the universal set, which in case of E^3 is all possible three dimensional points.

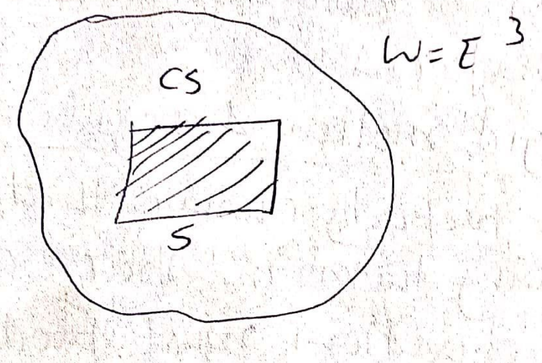
The solid definition eqn given by eqn ① introduces the concept of geometric closure which implies that the interior of the solid is geometrically closed by

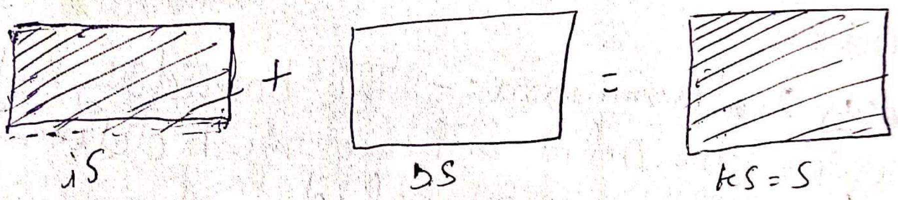
Its boundaries. Thus eqn ① can be written as

$$S = kS$$

where kS is the closure of the solid or point set S & is given by the right-hand side of eqn ① i.e.

$$kS = iS \cup bS$$





Solid & Geometric Closure Definitions.

The properties that a solid model or an abstract solid should capture mathematically can be stated as follows.

1. **Rigidity**:- This implies that the shape of a solid model is invariant & does not depend on the model location or orientation in space.
2. **Homogeneous three-dimensionality**:- Solid boundaries must be in contact with the interior. No isolated or ~~damagin~~ dangling boundaries should be permitted.
3. **finiteness & finite describability**:- The former property means that the size of the solid is not

limited amount of information can describe the solid. The latter property is needed in order to able to store solid models in to computers whose storage space is always limited. For example, a cylinder which may have a finite radius & length may be described by an infinite number of planar surfaces.

4. Closure under rigid motion & regularized boolean operations :- This property ensure that manipulation of solids by moving them in space or changing them via boolean operations must produce other valid solids.

5. Boundary determinism :- The boundary of a solid must contain the solid & hence must determine distinctively the interior of solid.

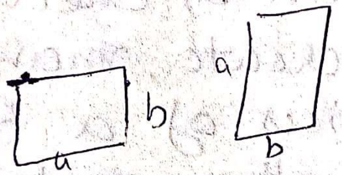
The formal properties of representation schemes which determine their usefulness in geometric modeling can be stated as follows.

1. Domain :- The domain of a representation scheme is class of objects that the scheme can represent or it is the geometric coverage of the scheme.

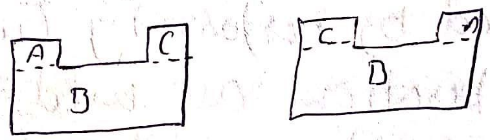
2. Validity :- The validity of a representation scheme is determined by its range, that is, the set of valid representations or models it can produce. Validity checks can be achieved in three ways: tests the resulting database via a given algorithm, build checks into the scheme generator itself or design scheme elements (such as primitives) that can be manipulated via a given syntax

Completeness or Unambiguousness :- This property determines the ability of the scheme to support analysis & other engineering applications. A complete scheme must provide models with sufficient data for any geometric calculation to be performed on them.

4. Uniqueness :- This property is useful to determine object equality. It is a custom in algebra to check for uniqueness but it is rare to do so in geometry. This is because it is difficult to develop algorithm to detect the equivalence of two objects & it is computationally expensive to implement these algorithms if they exist. Positional & permutational nonuniqueness are two simple cases.



Positional nonuniqueness



$$S = \begin{cases} A \cup B \cup C \\ B \cup C \cup A \\ C \cup A \cup B \end{cases} \qquad S = \begin{cases} A \cup C \cup B \\ B \cup A \cup C \\ C \cup B \cup A \end{cases}$$

Permutational Nonuniqueness

Boundary Representation (B-rep)

B-rep is one of the most popular & widely used schemes to create the solid models of physical objects. A B-rep model or boundary model is based on the topological notion that a physical object is bounded by a set of faces. These faces are regions or subsets of closed & orientable surfaces. A closed surface is one that is continuous without breaks. An orientable surface is one in which it is possible to distinguish two sides by using the direction of surface normal to point to the inside or outside of the solid model under construction. Each face is bounded by edges & each edge is bounded by vertices. Thus, topologically, a boundary model of an object is comprised of faces, edges & vertices of the object linked together in such a way as to ensure the topological consistency of the model.

The database of a boundary model contains both its topology & geometry. Topology is created by performing Euler operations & geometry is created by performing Euclidean ~~of~~ calculation. Euler operations are used to create, manipulate & edit the faces, edges & vertices of a boundary model as the set (boolean) operations create, manipulate & edit primitives of CSG model. Eulers operators as boolean operators, ensures the integrity (closeness, no dangling faces or edges, etc) of boundary models. They offer a mechanism to check the validity of these models. Geometry includes coordinates of vertices, rigid motion & transformation (translation, rotation etc.) & metric information such as distances, angles, areas, volumes & inertia tensors. It should be noted that topology & geometry are interrelated & cannot be separated entirely.

while
analysis
computa
man
h

while B-rep systems store only the bounding surfaces of the solid, it is still possible to compute volumetric properties such as mass properties (assuming uniform density) by virtue of ~~g~~ Gauss divergence theorem which relates volume integrals to surface ones. The speed & accuracy of these calculations depend on the types of surfaces used by the models.

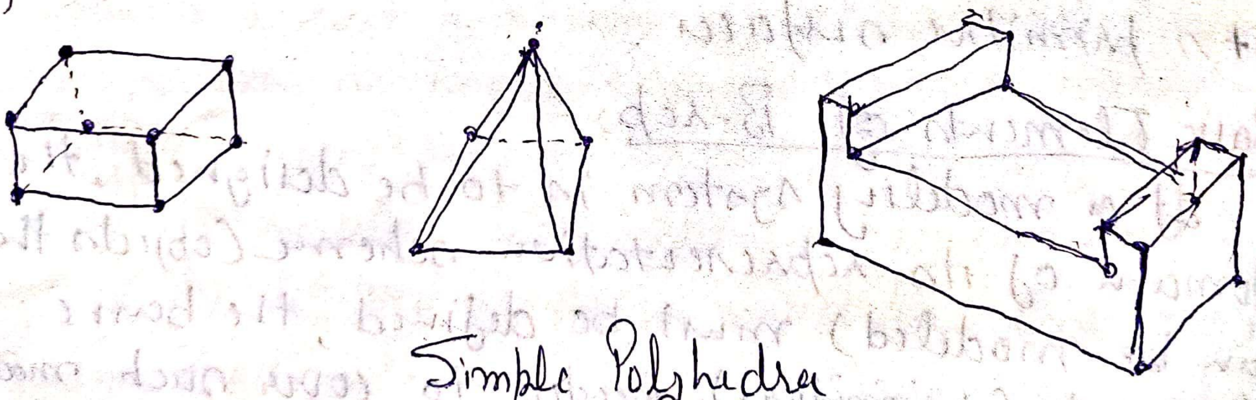
The modeling domain (or the range of objects that can be modeled) of a B-rep scheme is potentially large & depends mainly on the primitive surfaces (planar, curved or sculptured) that are admissible by the scheme based on half-spaces, a B-rep scheme with the same domain can be designed by using the boundary surfaces of the half-spaces as its primitive surfaces.

Basic Elements of B-rep:-

If a modeling system is to be designed, the domain of its representation scheme (objects that can be modeled) must be defined, the basic elements (primitives) needed to cover such modeling domain must be identified, the proper operators that enable the system users to build complex objects by combining the primitives must be developed & finally a suitable data structure must be designed to store all relevant data & information of the solid model. Other system & geometric utilities (such as intersection algorithms) may also need to be designed.

A vertex is a unique point (an ordered triplet) in sp
An edge is a finite, non self-intersecting, directed space curv bounded by two vertices that are not necessarily distinct. A face is defined as a finite connected, non-self-intersecting, region of a closed oriented surface bounded by one or more loops. A loop is an ordered alternating sequence of vertices & edges. A loops defines a non-self-intersecting, piecewise closed space curve which in turn may be a boundary of a face.

Different types of Polyhedra objects are the basic primitives of B-rep system. A polyhedral object (planar-face polyhedron) consists of planar faces (or sides) connected at the straight edges which, in turn, are connected at vertices. A curved object (curved polyhedron) is similar to a polyhedral object but with curved faces & edges instead.



Simple Polyhedra

Euler in ~~1759~~ (1752) proved that polyhedra that are homomorphic to a sphere (i.e. their faces are non-self-intersecting & belong to closed orientable surfaces) are topologically valid if they satisfy the following equation

$F - E + V -$

where F, E, V, L, B & G are the number of faces, edges, vertices, ~~faces~~, inner loop, bodies & genus (handles or through holes) respectively. Equation ① is known as the Euler or Euler-Poincare Law. The simplest version of this equation is

$$F - E + V = 2 \quad ②$$

which applies to the simple Polyhedra.

Eulers law given by equation ① applies to closed polyhedral objects only. These are the valid solid models we like to deal with. However open polyhedral objects donot satisfy eqn ①. This class of objects includes open polyhedral that may result during constructing boundary models of closed objects as well as all two dimensional polygonal objects. Open objects satisfy the following Euler's law.

$$F - E + V - L = B - G \quad ③$$

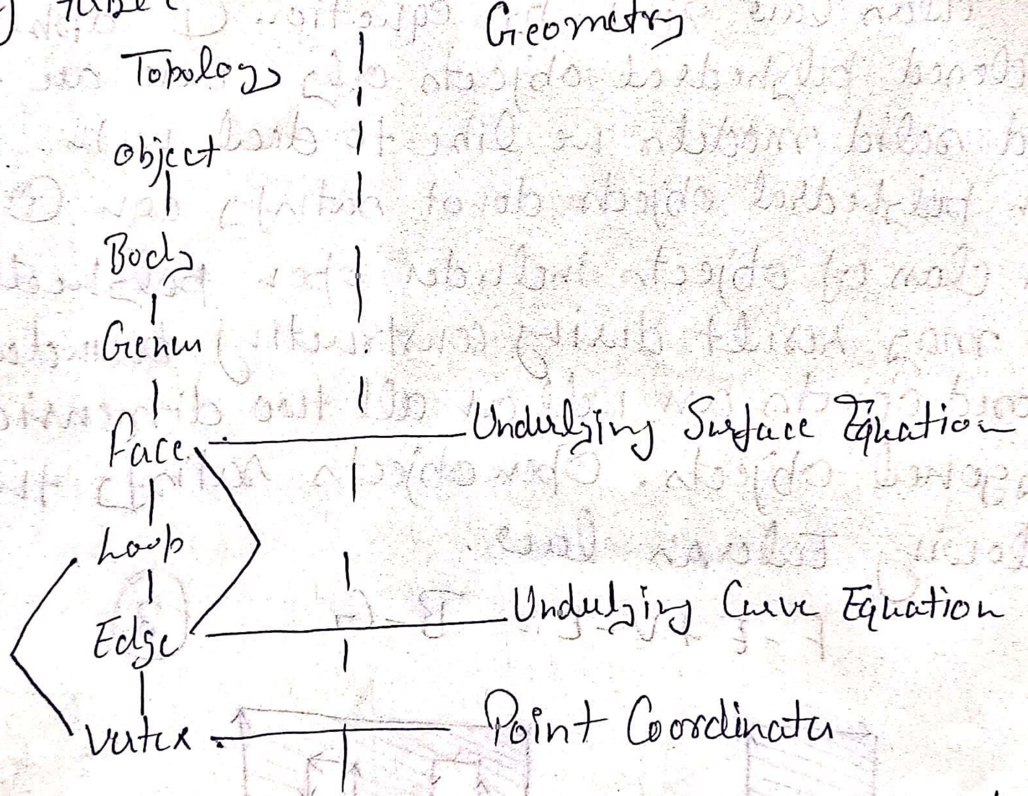
(i) (ii)

$1 - 4 + 4 - 0 = 1 - 0$

$1 = 1$

These are the some examples of open objects. The same guidline & rules are applied to the curved objects or polythe curved polyhedral like cylinder & sphere.

A general data structure for a boundary mod should have both topological & geometrical in The structure as shown here is based on eqn ①. A rational database model is very effective to implement such a data structure. List of bodies, faces, loops, edges & vertices, we can store in form of table.



General Data Structure for Boundary Modeling

Building operations :-

Equation ① forms the basis to develop operations to create boundary models of complex objects. Euler operators (or Euler primitives) are based on the equation. There are many variations on how these operators can be implemented. Sample operators are MBFV, KEV, MEF & GLUE. In these operators, M & K stand for Make & Kill ectively & other letters means the same as in eqn ①. Other operators are available to add convenience & flexibility to the construction process. Each operator usually has a complement that has the exact opposite effect on the construction process.

Some Eulers operations

Operation	Operator	Complement	Description of operator
Initialize database & begin creation	MBFV	KBFV	Make Body, Face, Vertex
Create edges & vertices	MEV	KEV	Make Edge, Vertex
Create edges & faces	MEKL	KEML	Make Edge, Kill Loop
	MEKBFL	KEMBFL	Make Edge, Kill body, face, Loop
	MFKLG	KFMLG	Make face, Kill Loop, Genus
Glue	KFEVMG	MFEVKG	Kill face, Edge, Vertex, Make Genus
	KFEVB	MFEVB	Kill face, Edge, Vertex, Body
Composite operations	MME	KME	Make Multiple Edges
	ESPLIT	ESQUEEZE	Edge-Split
	KVE		Kill Vertex, Edge

The table shows that the user is not free to construct faces, edges or vertices as with wireframe & surface modeling. There is no such operator as MF, MV or MF only because they all violate Euler's law.

Remarks of B-rep.

The B-rep scheme is very popular & has a strong history in computer graphics because it is closely related to traditional drafting. Its main advantage is that it is very appropriate to construct solid models of unusual shapes that are difficult to build using primitives. Examples are aircraft fuselage & automobile body styling. Another major advantage is that it is relatively simple to convert a B-rep model in to wireframe model because the model's boundary definition is similar to the wireframe definition. for engineering applications studied to date, alogorithms based on B-rep are reliable & competitive with those based on CSG.

One of the major disadvantages of boundary model is that it requires large amount of storage because it stores the explicit definition of model boundaries. It is also verbose scheme - more verbose than CSG. The model is defined by its faces, edges & vertices which tend to grow fairly fast for complex models. If B-rep systems do not have a CSG-compatible user interface, then it becomes inconvenient & slow to use Euler operators in a design & production environment. In addition, faceted B-rep is not suitable for many applications such as tool path generation.

Constructive Solid Geometry :- CSG

A CSG model is based on the topological notion that a physical object can be divided in to a set of primitives (basic elements or shapes) that can be combined in certain order following a set of rules (boolean operations) to form the object. Primitives themselves are considered valid CSG models. Each primitive is bounded by set of surfaces; usually closed & orientable. The primitives surfaces are combined via a boundary evaluation process to form the boundary of object, that is, to find its faces, edges & vertices. In addition to degenerating an object to a collection of primitives, a CSG model is fundamentally & topologically different from a B-rep model in that case the former does not store explicitly the faces, edges & vertices. Instead, it evaluates them whenever they are needed by the applications' algorithm' eg:- generation of line drawings.

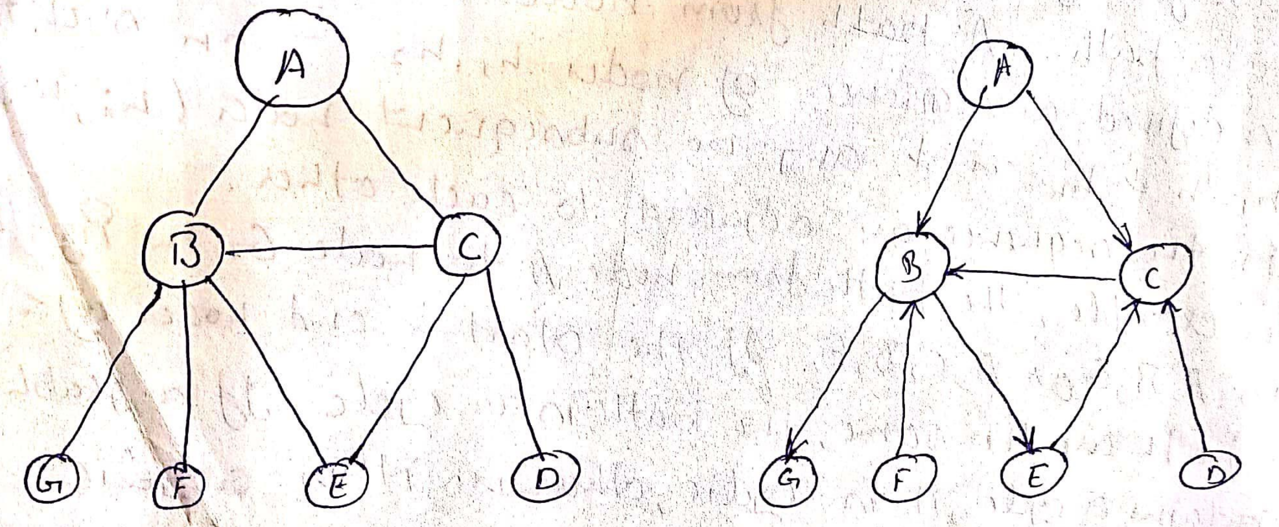
There are two types of CSG schemes. The most popular one & the one is based on bounded solid primitives that is r-sets. The other one, less popular, is based on generally unbounded half spaces. The bounded solid primitives are considered composite half-spaces & the boundaries of their primitives are the surfaces of the corresponding half-spaces. CSG systems based on bounded primitives (eg:- PADL-2 & GMSOLID) allow their sophisticated users to use both their bounded primitives &/or half-spaces to create new primitives, typically called metaprimitives.

It is also possible to extend the modeling domain of a system by implementing new half-spaces & eventually new primitives, in to its software.

The database of a CSG model, similar to B-rep, store its topology & geometry. Topology is created via the regularized set (boolean) operations that combine primitives. Therefore, the validity of the resulting model is reduced to the validity checks of the used primitives. For bounded primitives, these checks are usually simple (in the form of greater than zero) & validity of the CSG model may be ensured essentially at the syntactical level. This mean that in a CSG Language a model is a valid if it can be described syntactically correct using this language (user interface). The geometry stored in the database of a CSG model include configuration parameters of its primitive & rigid motion & transformation. Geometry of faces, edges & vertices are not stored but can be calculated via the boundary evaluation process.

While data structures of most boundary representations are based on the winged-edge structure developed by Baumgart in 1972, data structure of most CSG representation are based on the concept of graphs & trees.

A graph is defined as set of nodes connected by set of branches or lines. Each branch in a graph is specified by a pair of nodes. fig given below illustrates a graph.



(a) Graph (b) Digraph

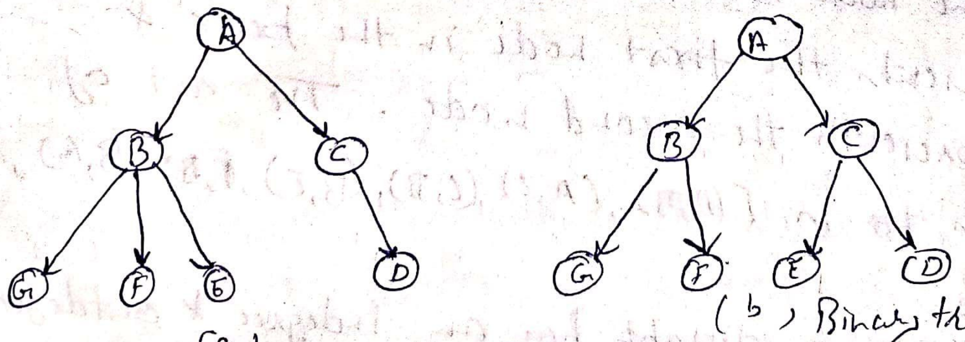
fig 1

The set of nodes is {A, B, C, D, E, F, G} & the set of branches, or the set of Pair is {{A,B}, {A,C}, {B,C}, {B,E}, {B,F}, {B,G}, {C,D}, {C,E}}. Branches have directions in a digraph & become in a sense arrows going from one node to another. The tail of each arrow represents the first node in the pair & its head represents the second node. The set of ordered pairs ~~for~~ is {(A,B), (A,C), (C,B), (B,E), (F,B), (B,G), (B,C), (E,C)}

Each node in a digraph has an indegree & outdegree & has a path it belongs to. The indegree of a node is the number of arrow heads entering the node

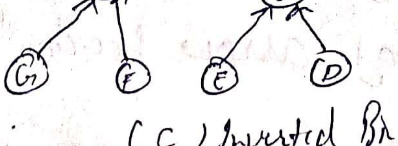
& its outdegree is the number of the arrow tails leaving the node. For example, node B in the fig has an indegree of 3 & an outdegree of 2 while node D has a zero indegree & an outdegree of 1. Each node in a digraph belongs to a path. A path from node n to node m is defined as sequence of nodes $n_1, n_2 -- n_k$ such that $n_1 = n$ & $n_k = m$ & any two subsequent nodes (n_i, n_{i+1}) in the sequence are adjacent to each other.
For example, the path from node A to node G in Fig 1(b) is A, B, G or A, C, B, G. If the start & end nodes of a path are the same, the path is a cycle. If a graph contains a cycle, it is cyclic, otherwise it is acyclic.

A tree is defined as an acyclic graph in which only a single node, called the root, has a zero indegree & every other node has an indegree of 1. This implies that any node in the tree except the root but a tree must be a graph. The digraph shown in fig 1(b) is not a tree. However, its modification shown in Fig 2(a) is a tree.



(a) Tree

(b) Binary tree



(c) Inverted Binary tree

Fig 2

Types of Trees

Node A is the root of the tree & nodes E, F, & G, for example, have node B as their ancestor or node B has nodes E, F & G as its descendants. If the descendants of each node are in order, say, from left to right, then the tree is an ordered one. Moreover, when each node of an ordered tree has two descendants (left & right), the tree is called a binary tree. Finally, if the arrow directions in a binary tree are reversed such that every node, except the root, in the tree has an out-degree of 1 & the root has a zero outdegree, the tree is called an inverted binary tree. An inverted binary tree is very useful to understand the data structure of CSG models (sometime called boolean models)

Any node in a tree that does not have descendants, that is, with an outdegree equal to zero, is called a leaf node & any node that does not have descendants (outdegree greater than zero) is an interior node.

Building operations of CSG :-

The main building operations in CSG scheme are achieved via the set operators or more specifically the regularized operators: union ($\cup^*$), intersection ($\cap^*$) & difference ($-^*$). Set operators are known as boolean operators due to close correspondence between the two. Union, intersection & difference are equivalent to OR, AND & NOTAND respectively. Due to deep root of CSG schemes in set theory or boolean algebra, CSG models are usually referred to as set-theoretic, boolean or combinatorical models. Set-operations algorithms are amongst the most fundamental & delicate software component of the solid modelers:

Unlike Euler operators, regularized set operators are not based on a given law or equation, but they derive their properties from the set theory & & the concept of closure. They are considered higher level operators than Euler operators. The validity checks for set operators are usually simple in the case of bounded primitives.

Set operations are performed by so-called boundary merging in B-rep while in CSG they are performed by so-called boundary evaluation. In CSG, non-incremental & incremental boundary evaluations are available. In the non-incremental evaluator, only the boundary of the final solid S is evaluated & not for subsolids of S. In the incremental evaluator, the boundaries of the intermediate subsolids are evaluated as the CSG tree is traversed to produce the boundary for the final solid. This latter evaluator is actually a boundary-merging procedure similar to what is used by B-rep schemes.

Remarks:- The CSG scheme is a very powerful representation scheme. It is not closely related to conventional drafting language

& has many advantages. It is easy to construct out of primitives & boolean operations. It is concise & requires minimum storage to store solid definitions (the CSG graph). This is why it is slow to retrieve the model because it has to build a boundary from the CSG graph. It is also due to this fact that the CSG is slow in generating wireframes, that is, line drawings. CSG must be converted internally in to a B-rep (similar to the set-operation algorithm covered earlier) to display the model or generate its line drawings.

Application algorithm based on CSG schemes are very reliable & competitive with those based on B-rep schemes. However, major disadvantage of CSG is in its inability to represent sculptured surfaces & half surfaces.

Sweep Representation:-

Shemes based on sweep representation are useful in creating solid models of two & half-dimensional objects. The class of two & a half dimensional objects includes both solid of uniform thickness & axisymmetric solids. The former are known as extruded solid & are created via linear or translational sweep. ~~Sweeping is used~~ as the latter are solids of revolution which can be created via rotational sweep. Sweeping is used in general as a means of entering object descriptions in to B-rep or CSG modelers. There exists no sweeping-based modelers due to the limited modeling domain of sweep representations & the lack of formal underlying theory of sweeping. For example, general validity & regularization conditions for sweep representations are not known & are usually left to the user.

There are three types of sweep:

Linear
Non-Linear
Hybrid Sweeps

In Linear sweep, the path is linear or circular vector described by a linear, most often parametric equation while in non linear sweep, the path is a curve described by a higher order equation (quadratic, cubic or higher). Hybrid sweep combines linear and or nonlinear sweep via set operations & is, therefore, a means of increasing the modeling domain of sweep representations.

Linear sweep can be divided further in to translational & rotational sweep. In translational sweep, a planar two-dimensional point set described by its boundary (or contour) can be moved a given distance in space in a perpendicular direction (called the directrix) to the plane of the set. This is similar to entity projection & surface offsetting or translation in wireframe & surface representations respectively