Constructive Solid Geometry
You can do a lot of things with simple primitives (e.g., cube, sphere, cylinder, etc.)

The power is in the operations that can be performed on them
Wireframe Model

- The simplest way of representing solids
- Data structures:
  - A vertex table
  - An edge table
  - A face table (optional): faces are *not* displayed
- Fast
- Ambiguity (is a face there or not)
Example

1, 2, 3, 4, 5, 6, 7, 8

a, b, c, d, e, f, g, h, i, j, k, l

A, B, C, D, E, F

Computer Graphics
True Solids

- A few primitives
  - Cube, sphere, cylinder

- A few operations
  - Singleton
    - Translate, rotate, scale
  - Binary
    - Union (merge), intersection, difference, invert
Examples

Computer Graphics
Example

$A \cup B \quad A \cap B \quad A - B \quad B - A$
Example

Computer Graphics
CSG Expression and CSG Tree

Difference(
  Union(
    Trans(cylinder),
    Trans(cube)
  ),
  Trans(sphere)
)

- Each CSG expression has a tree representation
- The construction is not unique (there can be two trees that represent the same results)
How to Realize Those Operations?

- CSG operations are difficult to represent mathematically (no closed-form solutions in general)
- However, for rendering (ray tracing), closed-form solutions are not needed

Union A,D
Intersection C,B
Difference (obj2-obj1) B,D
Difference (obj1-obj2) A,C
Procedures

- For each ray
  - For each object
    - Use bounding box to avoid intersection test
    - If needed, use inside-outside function to determine whether ray-object intersection occurs
    - Traverse the CSG tree bottom-up and apply operations to find identity of extremal objects
Bounding Volume

\[ \text{Slab: } aX + bY + cZ + d = 0 \]

\[ \begin{align*}
X &= X_o + t\Delta X \\
Y &= Y_o + t\Delta Y \\
Z &= Z_o + t\Delta Z \\
t &= -\frac{aX_o + bY_o + cZ_o + d}{a\Delta X + b\Delta Y + c\Delta Z} = \frac{A + D}{B}
\end{align*} \]

- \( A \): per ray per slab set
- \( B \): per ray per slab set
- \( D \): per slab
Bounding Volume (cont.)

- All the maximum (circle) intersections must be after all the minimum (square) intersections
Spatial Partitioning

- Ray can be advanced from cell to cell
- Only those objects in the cells lying on the path of the ray need be considered
- First intersection terminates the search
Packages

- A number of them built on top of povray
  - http://www.povray.org
- With simple VRML like languages to specify operations and appearance (texture mapping)