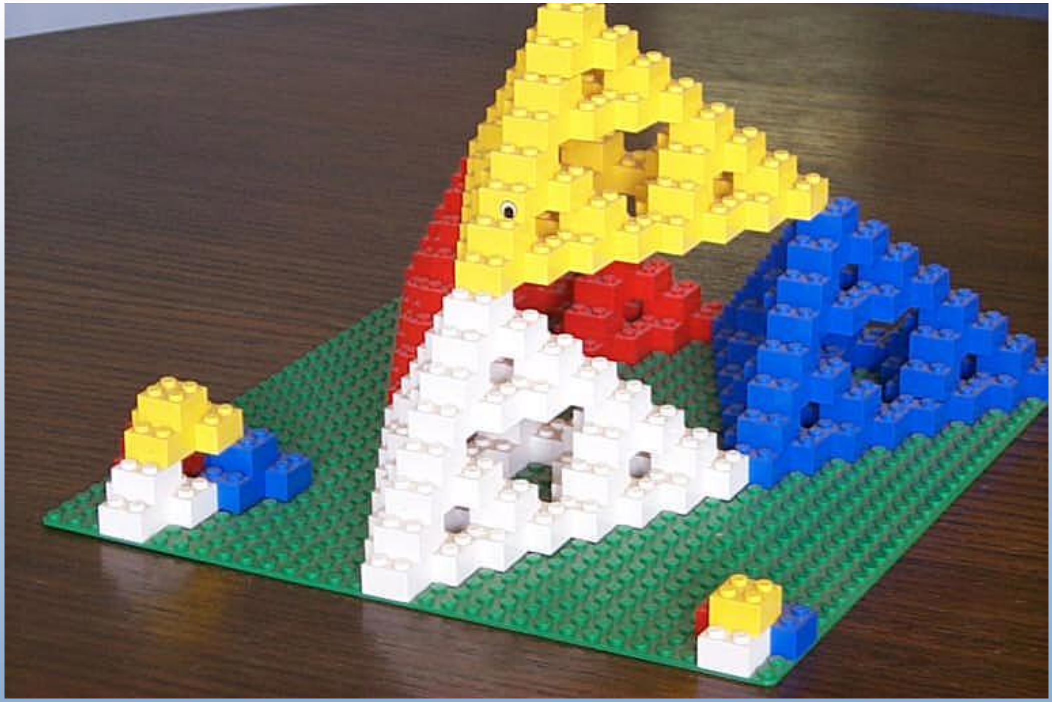
Geometric Modeling

Alexander Pasko, Evgenii Maltsev, Dmitry Popov

Solid Modelling

Constructive
Solid
Geometry

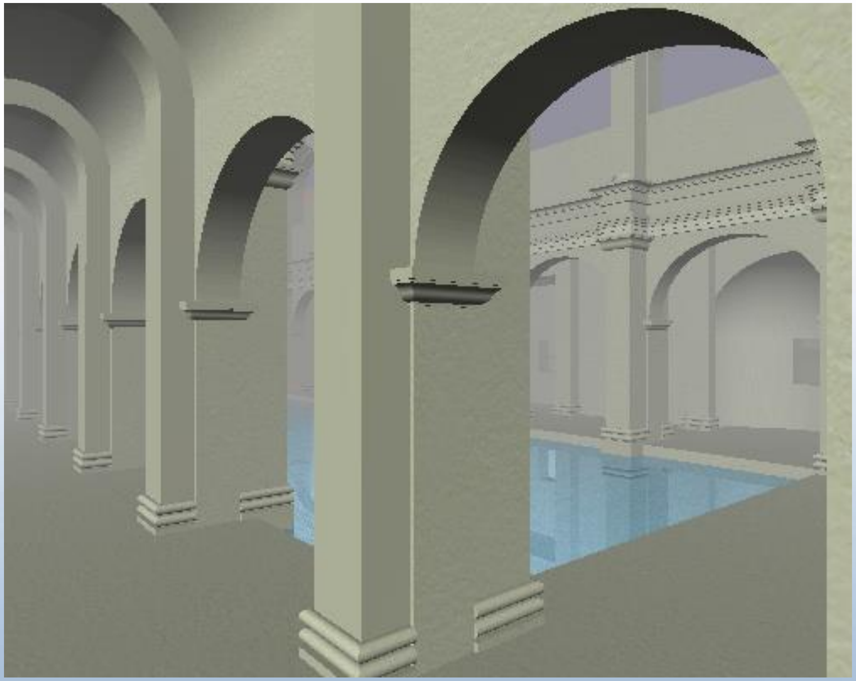
CSG



www.povcomp.com/hof/Villarceau_Circles-CSG.html

Contents

- Set theory basics
- Regularization
- CSG tree
- Point membership classification (PMC)
- Special cases and null objects
- PADL-2 system



The svLis model of the Great Bath in in Aquae Sulis as it was in 200 AD

Notions of set theory

- *Set*

  denotes any well-defined collection of objects

Universal set E

is any set that contains all the elements of all the sets under consideration

Null set

$\varnothing$ is a set which has no elements at all

$A \subset B$

Set A is a subset of B if every element in A is contained in B

Notions of set theory

- Set operations (set-theoretic, Boolean):

  - ***union*** $\boldsymbol{A \cup B}$

  - ***intersection*** $\boldsymbol{A \cap B}$

  - ***complement*** $\boldsymbol{cA}$
    or $\boldsymbol{\neg A}$ contains all elements in E that are not elements in A.

  - ***difference*** $\boldsymbol{A \backslash B = A - B = A \cap \neg B}$

Notions of set theory

Geometric interpretations:

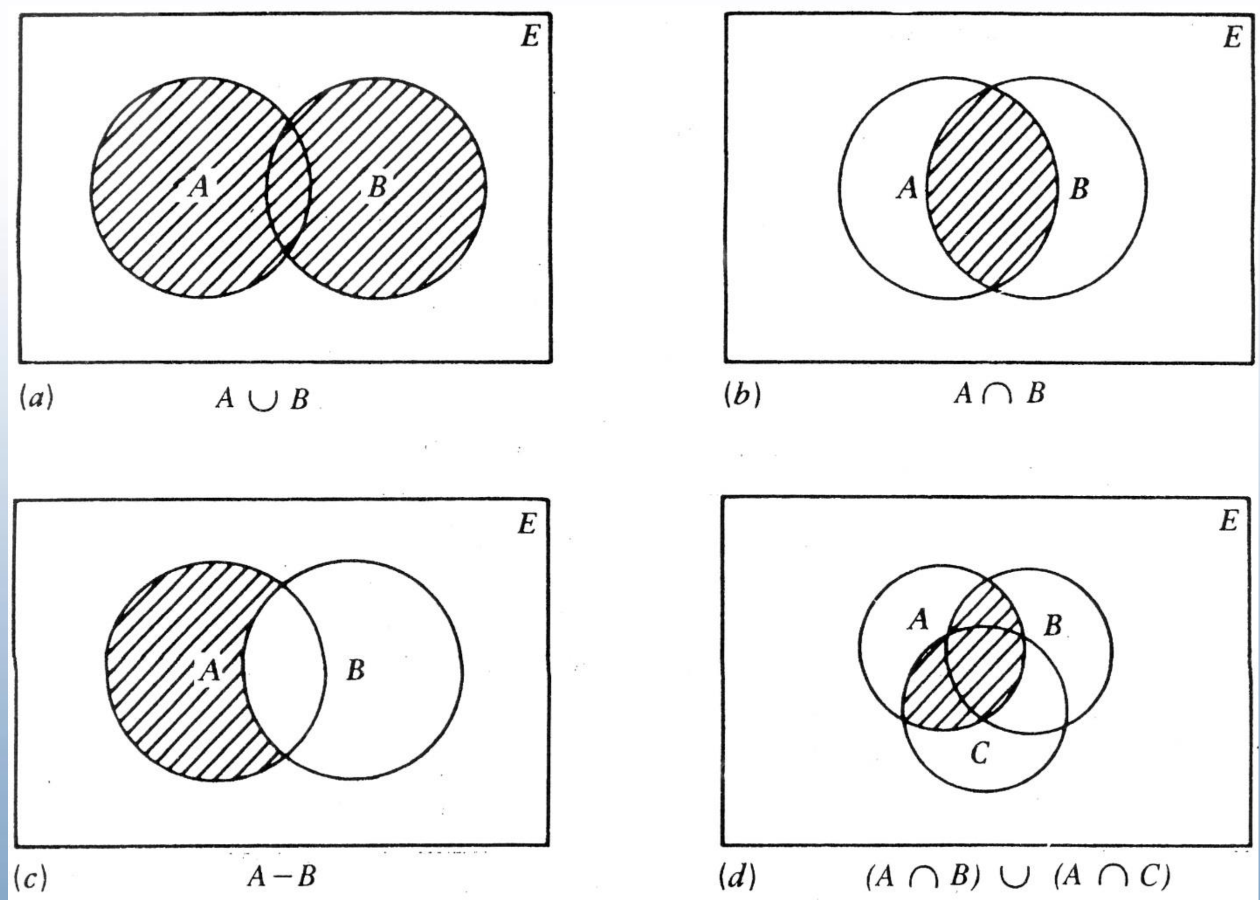
- Sets consist of points, and the universal set E is the set of points defining a Euclidean space.

- $\boldsymbol{P} \in \boldsymbol{A}$ point *P* is included (contained) in *A*.

Notions of set theory

Union

Inter-
section



Difference

Mortenson, p.401

Notions of set theory

- ***Boundary (limit) point*** has $P \in A$ and $P' \notin A$ in any its neighborhood.

Boundary bA is the set of all boundary points of A.

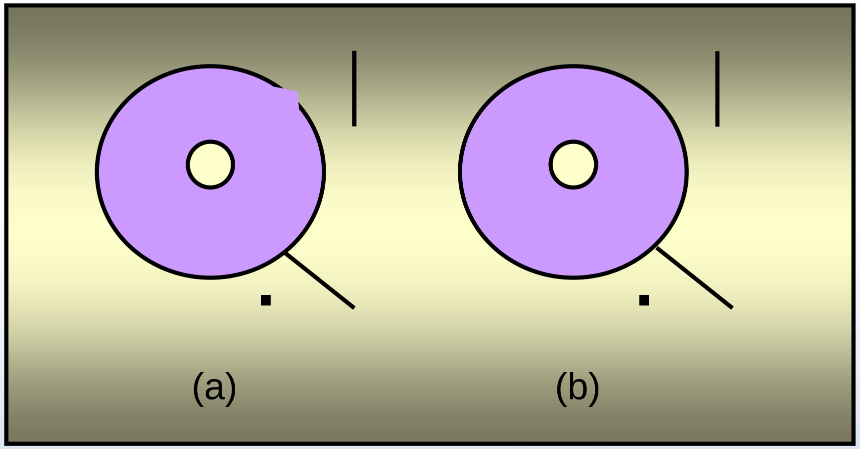
Interior iA is the set of all points $P \in A$ and $P \notin bA$.

$$\boldsymbol{A = bA \cup iA}$$

- ***Open set*** does not contain its boundary points.

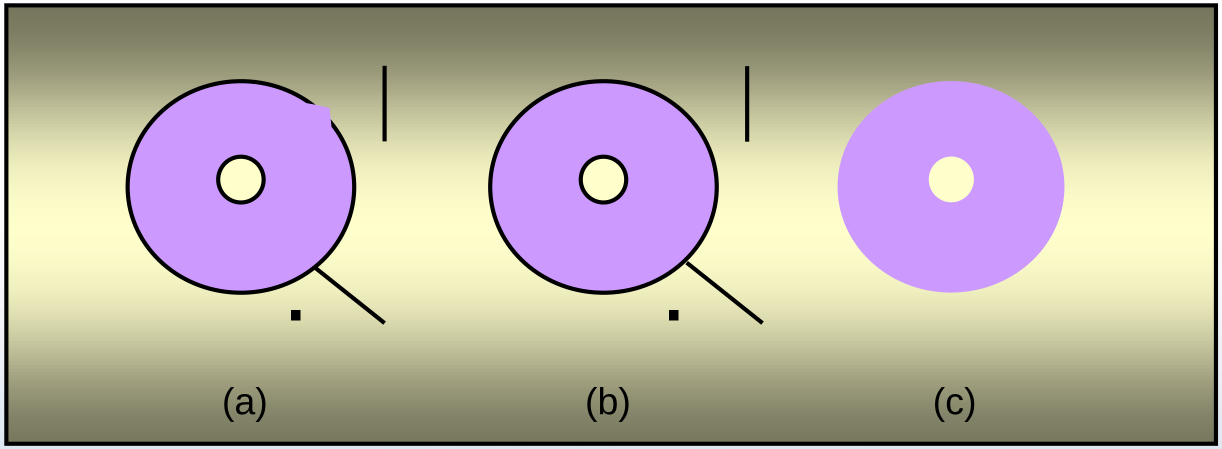
Closure kA of an open set is the union of the set with all its boundary points.

Regularizing a solid

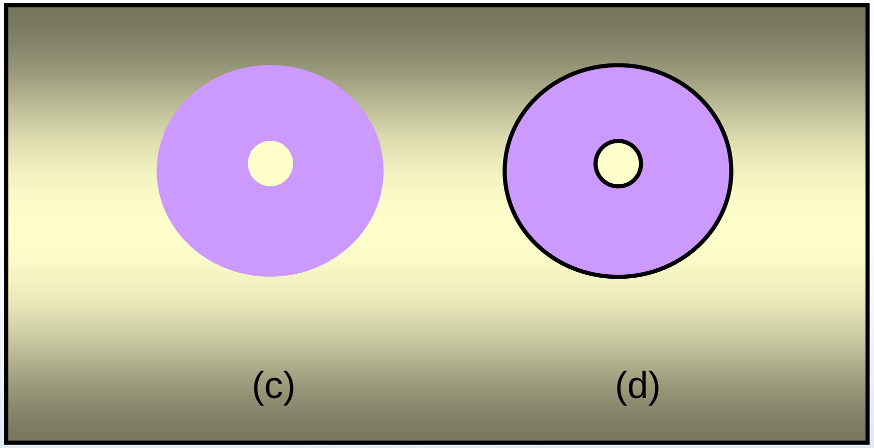


- (a) The object is defined by interior points and boundary points. The object has dangling and unattached points and lines. Not all boundary points are included.

It is a ***nonregular*** solid.



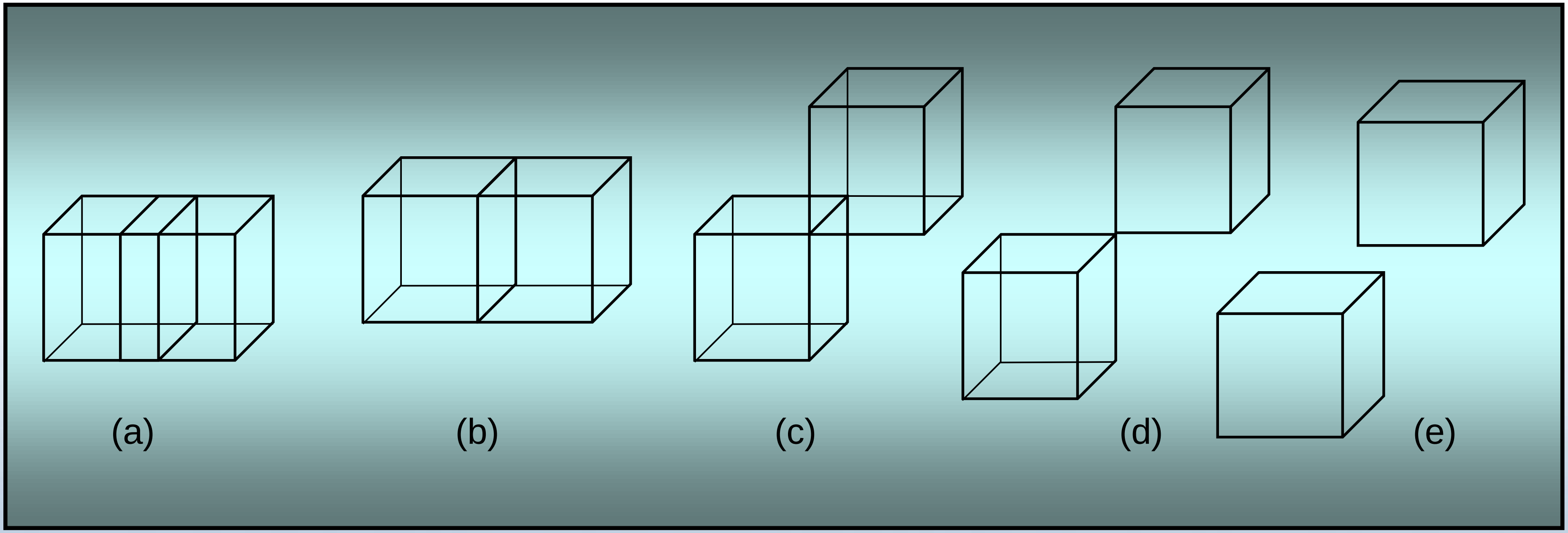
- (b) Closure of the object. All boundary points are part of the object.
- (c) Interior of the object. Dangling and unattached points have been eliminated.



- (d) ***Regularization*** of the object is **the closure of its interior.**
- A ***regular set*** equals the closure of its interior:

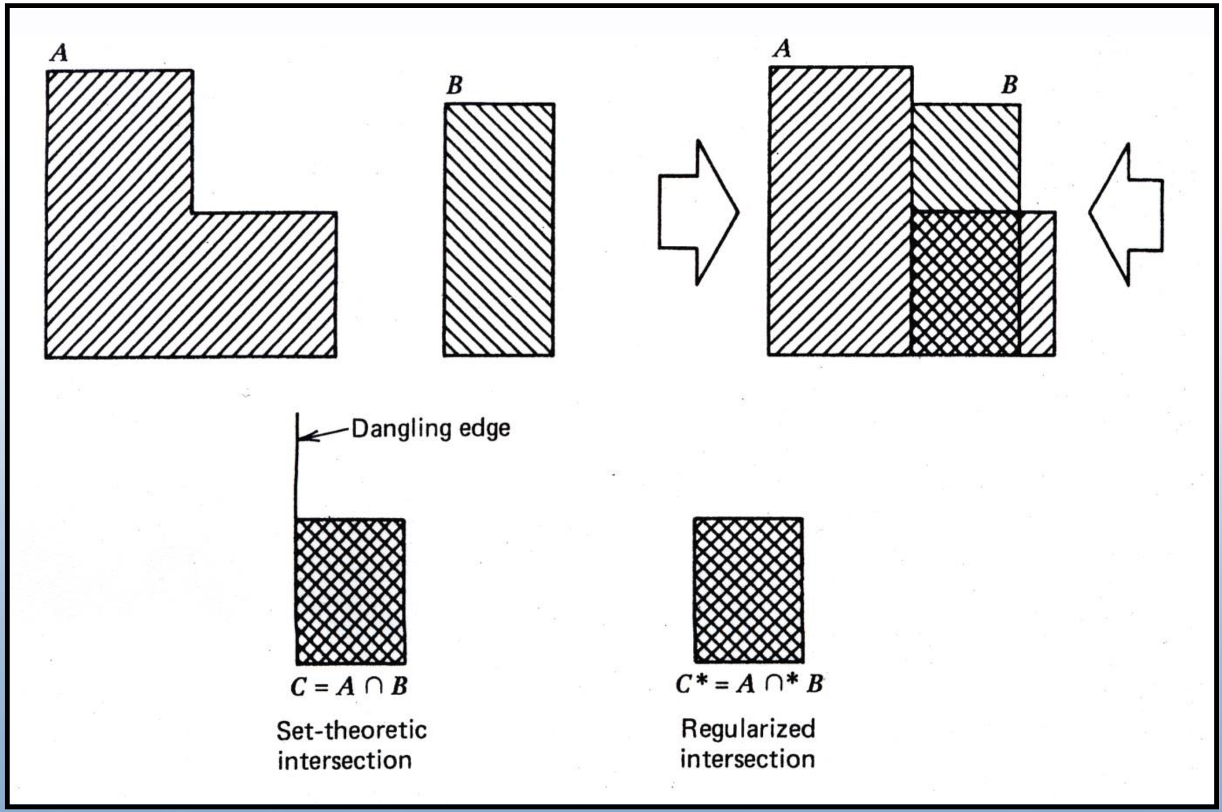
$$\boldsymbol{A = kiA}$$

Regularized set-theoretic operations



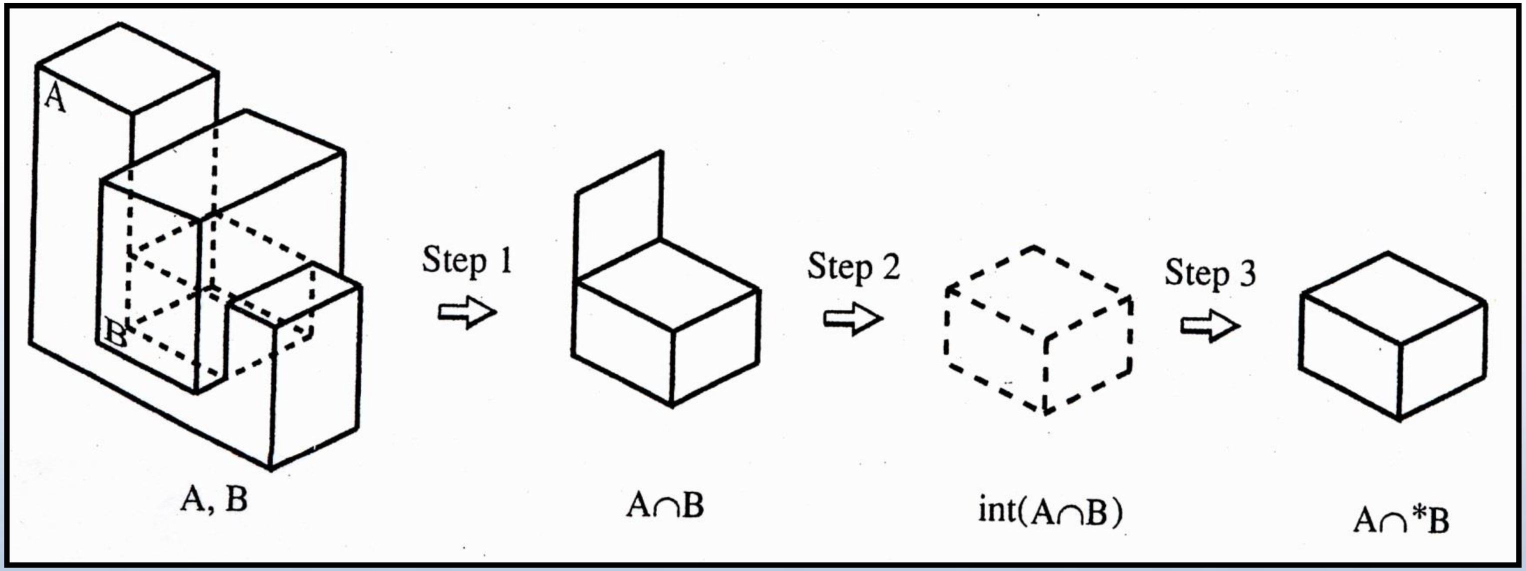
- Difference between standard and regularized set intersection operations

Set-theoretic and Regularized intersection



<Fig. 9.22, Mortenson, p.406>

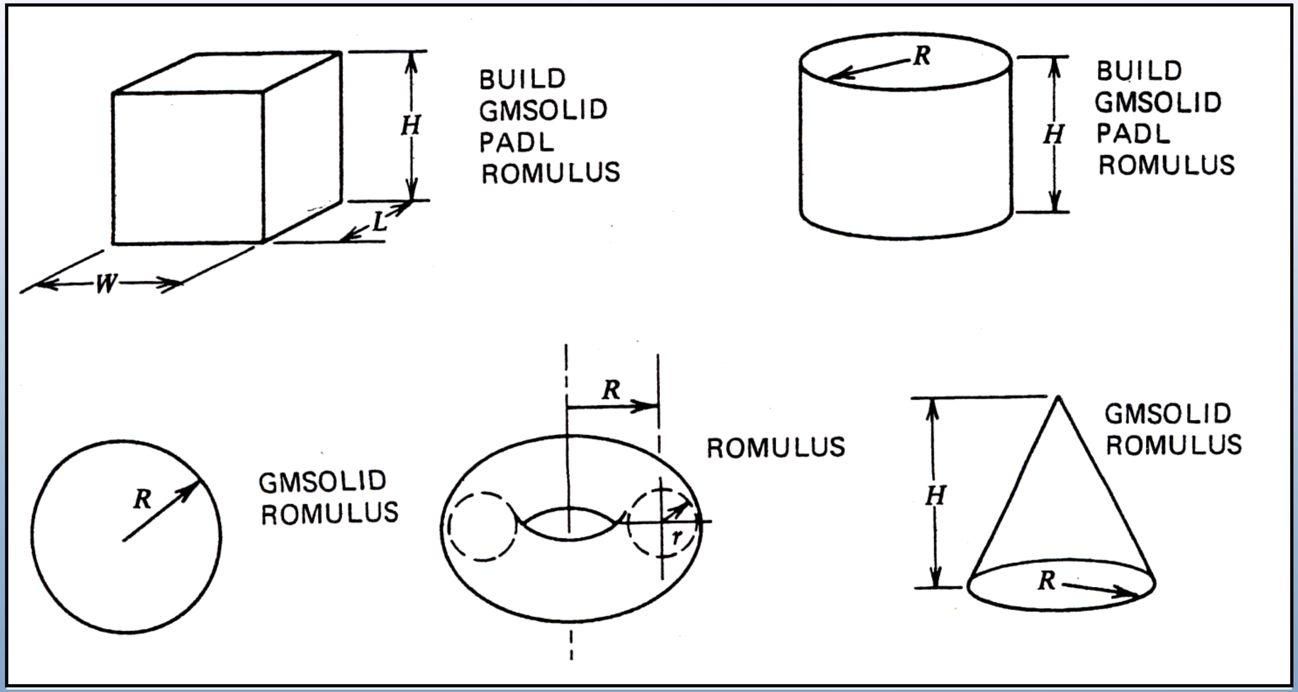
Set-theoretic and Regularized intersection



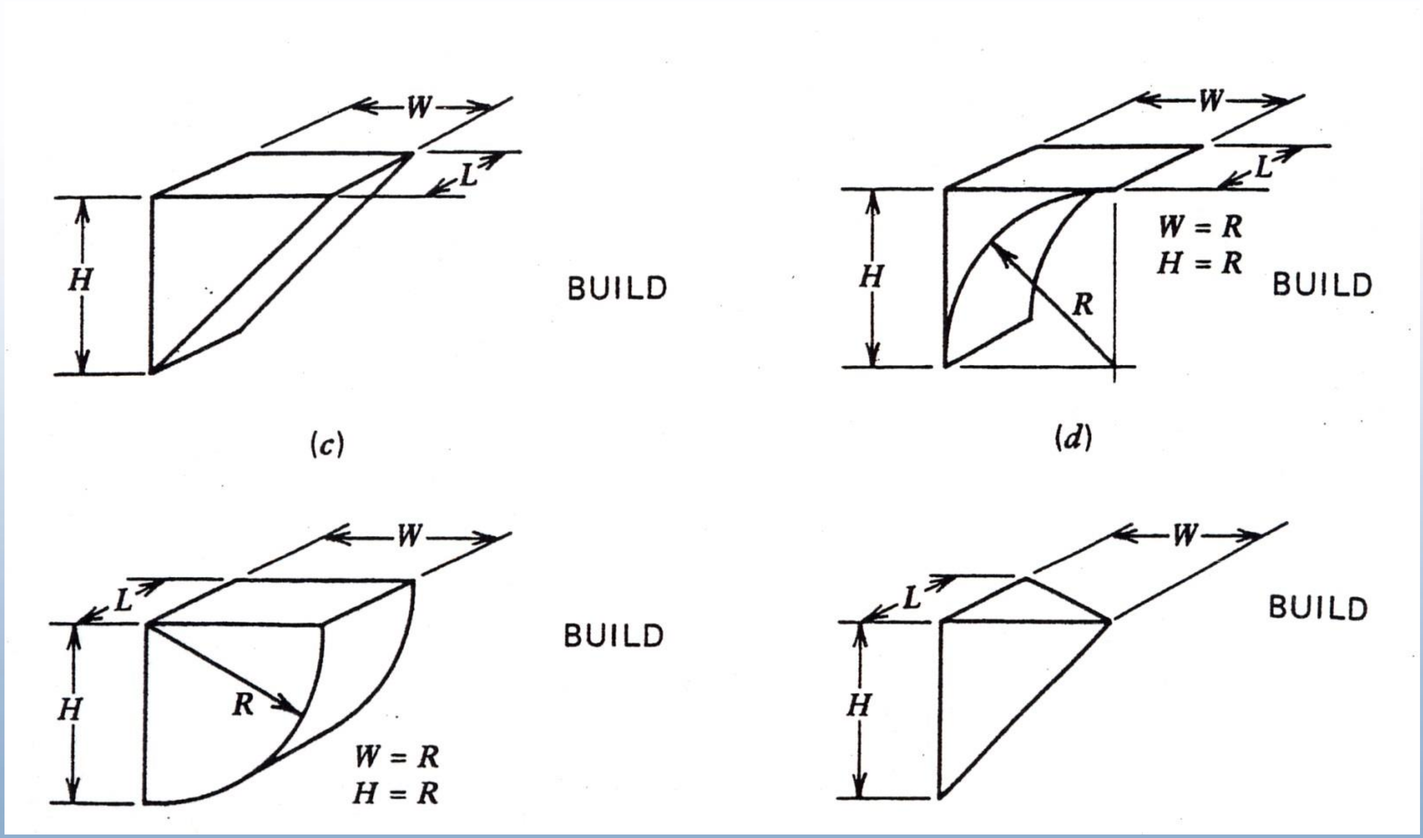
CSG representation: primitives

In CSG, simple primitives are combined by means of regularized set operations and rigid motions.
Standard CSG primitives:

block, cylinder, sphere, cone, and torus

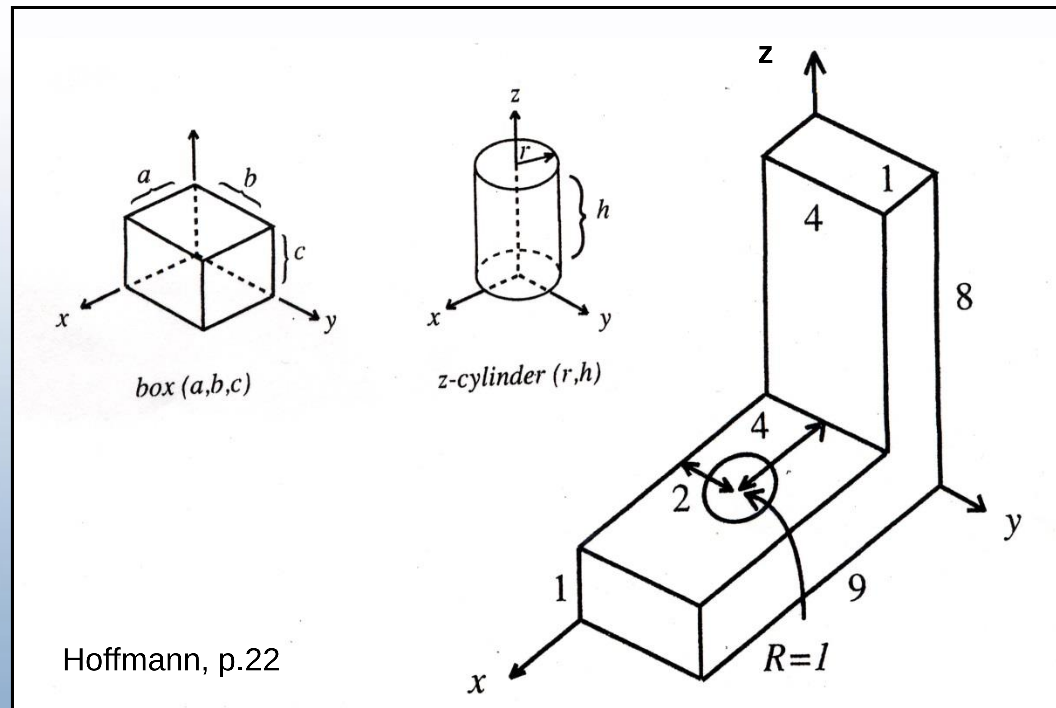


CSG representation: primitives



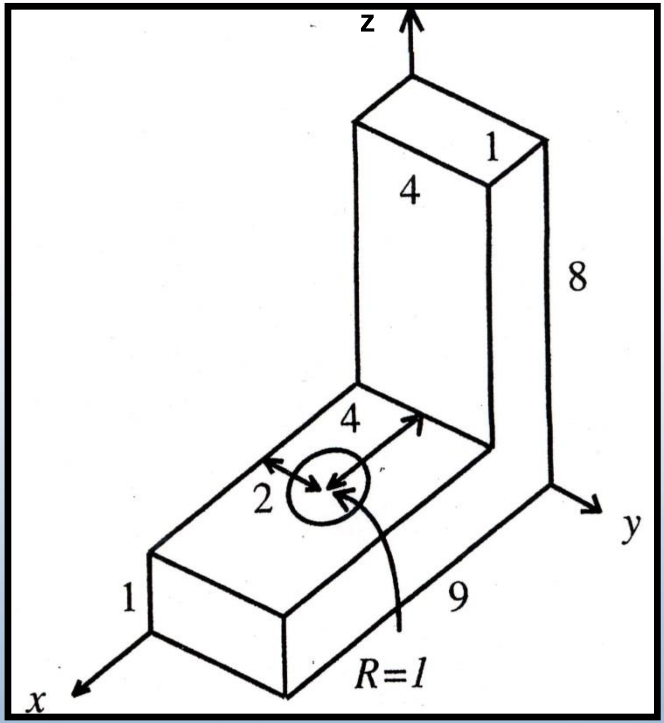
CSG Tree

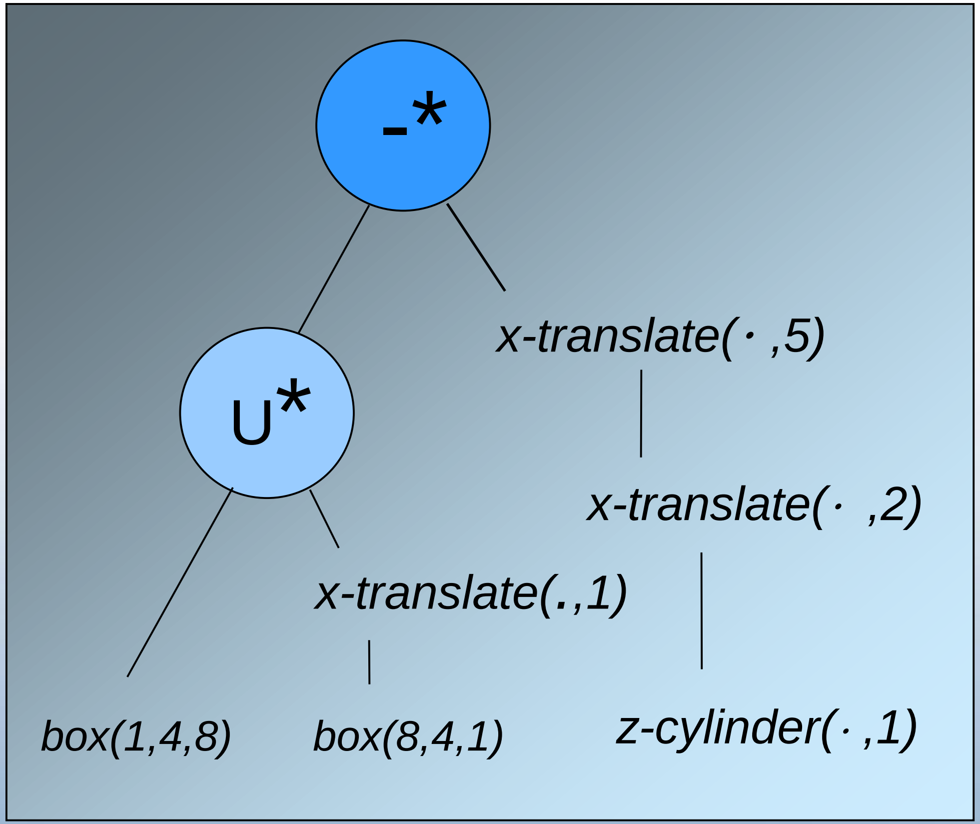
- An object is represented as a ***binary tree*** with operations at the internal nodes and primitives at the leaves
- Nodes: regularized set operations or rigid motions (translation, rotation)



Hoffmann, p.22

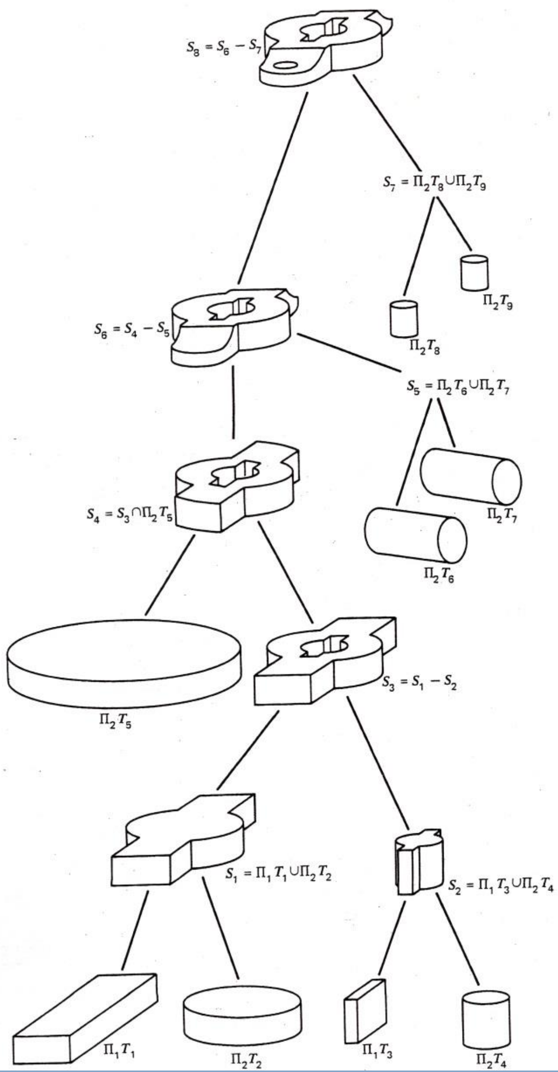
CSG Tree

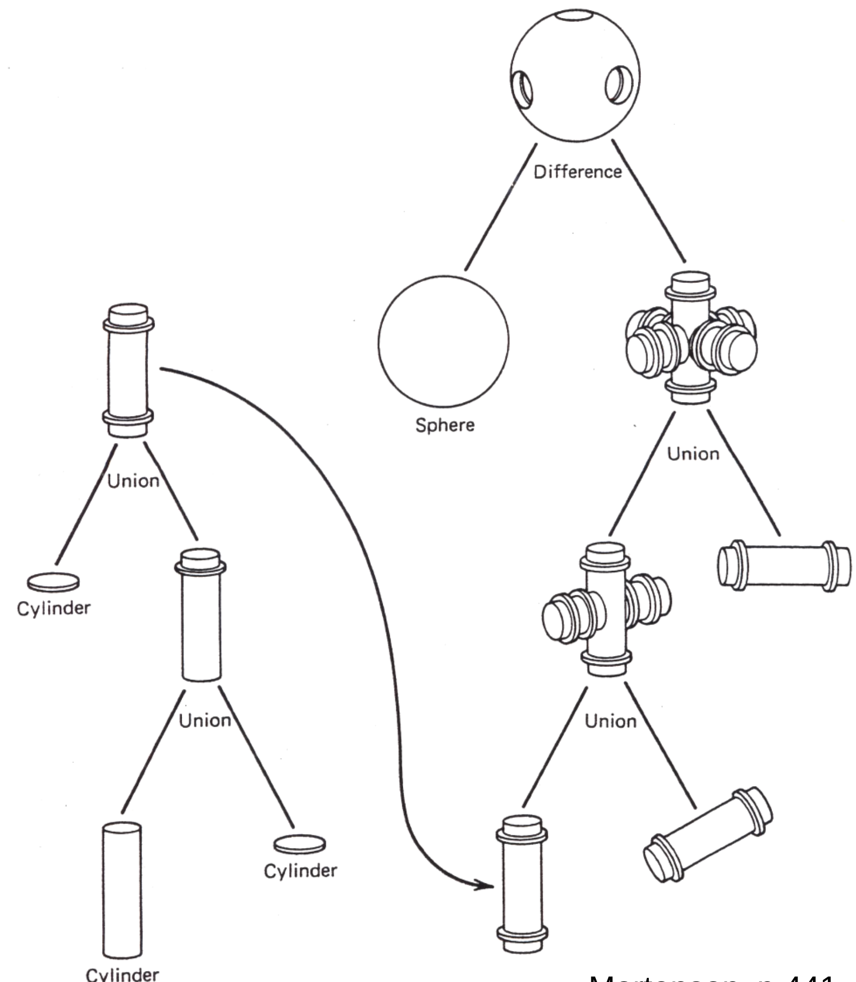




(block(1,4,8) U* x-translate(block(8,4,1),1) -* x-translate(y-translate(z-cylinder(1,1),2),5)

CSG Tree: Examples





Mortenson, p.441

Point membership classification

A solid *S* and a point *X* are given.

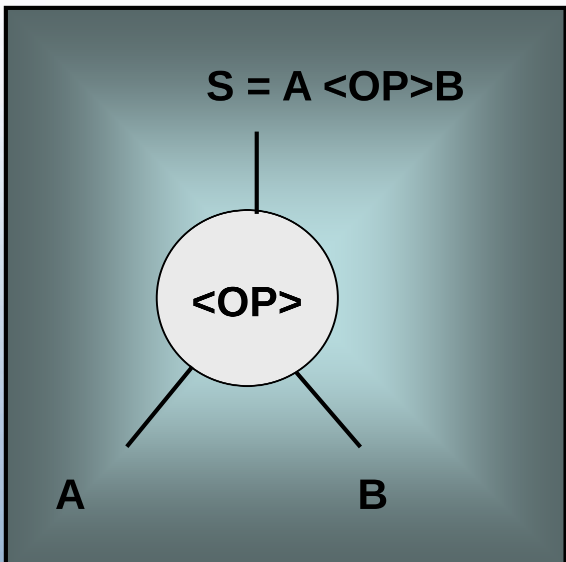
Query:

Is *X* inside, outside or on the boundary of *S* ?

Point membership classification (PMC) function

M[X,S] = (XinS, XonS, XoutS)

Divide-and-conquer paradigm



Tilove IEEE Tr. on Comp. 1980, p.877

F(S) $\leftarrow$ IF (S is a primitive) THEN prim - f(S) ELSE combine (f(A), f(B), <OP>)

The divide and conquer paradigm. <OP> is a regularized operator ($\cup^*$, $\cap^*$, or $-^*$)

Point membership classification

M [X, S] F(S) ← IF (S is a primitive)

THEN prim – M (X,S)

ELSE combine (M [X, left - subtree(S),

M [X, right – subtree (S)], root <S>))

- *prim-M* is a primitive classification procedure and must produce “in”, “on” or “out” answers for a given point
- *combine* applies set operations (three-valued logic!) to its arguments.

Recursive PMC algorithm structure

1. Point coordinates *(x,y,z)* are sent to the root of the CSG tree.

2. Downward propagation

- Coordinates are propagated into the tree down to the leaves, possibly altered.

- At each leaf, the final point coordinates describe the same point, but in the local coordinate frame of the primitive solid.

Downward propagation cases:

1) If *(x,y,z)* arrives at a set-theoretic operation node, it is passed unchanged to the two subtrees

2) If *(x,y,z)* arrives at a motion node, the inverse transformation is applied to *(x,y,z)*, resulting in new local coordinates *(x',y',z')*, which are sent to the two subtrees

3) If *(x,y,z)* arrives at a leaf, the point is classified against the primitive, and the classification is returned to the parent of the leaf

3. Upward propagation

- Classifications from the subtrees are combined in the set-theoretic operation node

- No work is done at motion nodes.

"On/on" ambiguity for PMC

- Combination of point classifications for union and intersection:

$\cup^*$	*in*	*on*	*out*
in	*in*	*in*	*in*
on	*in*	*on?*	*on*
out	*in*	*on*	*out*

$\cap^*$	*in*	*on*	*out*
in	*in*	*on*	*out*
on	*on*	*on?*	*out*
out	*out*	*out*	*out*

Hoffman, p.26

“On/on” ambiguity for PMC

Regularized intersection of A and B with point P “on”:

$$S = A \cap^* B$$

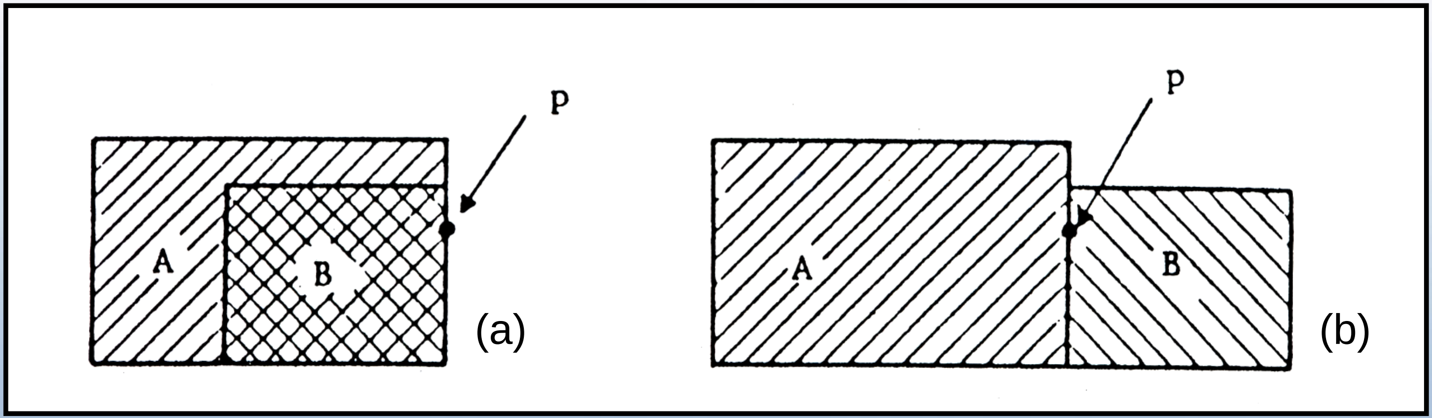


Figure (a): *P* is “on” *S*
Figure (b): *P* is “out” of *S* (because *S* is empty)

Image by Tilove, IEEE Tr. on Comp. 1980, p.877

“On/on” ambiguity for PMC

The classification of a point in a regularized set operation node cannot be computed only from the subtree classifications.

The additional information is given by a neighbourhood of the point

Neighborhood model

To resolve the “on/on” ambiguity, we need to know which points “near” point *P* are elements of solid *S*.

A **neighborhood** of radius *R>0* of *P* with respect to *S*, is the intersection with *S* of an open ball of radius *R* centered at *P*:

$$N(P, S; R) = B(P;R) \cap^{*} S,$$

where *N* is the neighborhood and *B* is the open ball.

PMC using the neighborhood:

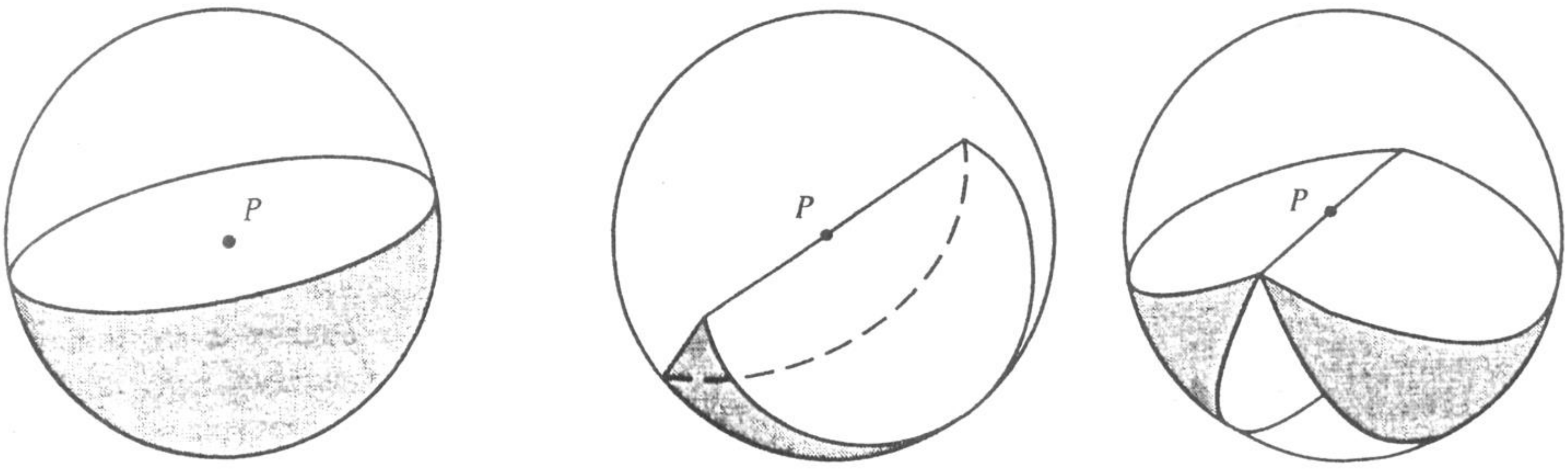
- *P* is "inside" of *S* if and only if it has a "full" neighborhood, i.e. $N = kB$
- *P* is "outside of *S* if it has an "empty" neighborhood,

$$N = \varnothing$$

- $P \in bS$ if every neighborhood is neither full nor empty

Neighborhood model

Examples of point neighborhoods:



Face point *Edge points*

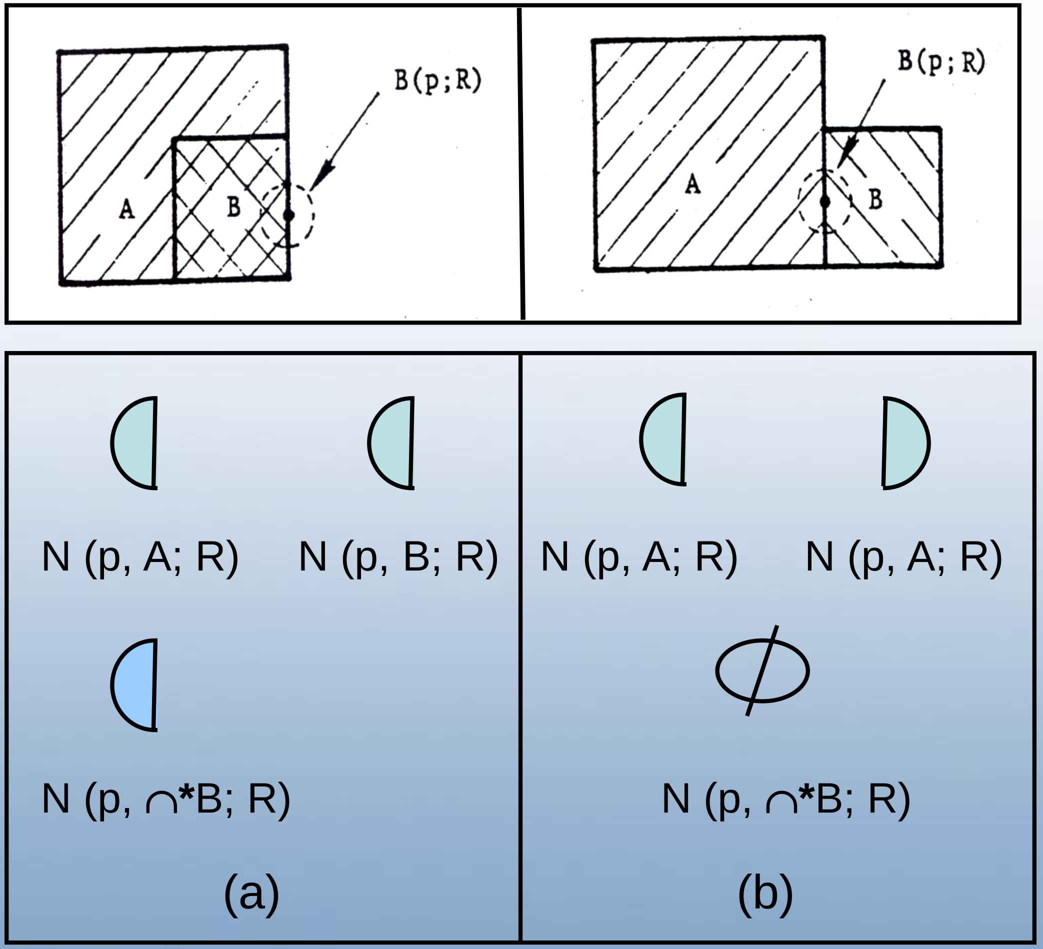
Hoffmann, p.27

Resolving the “on/ on” ambiguity

To get the correct answer, we must perform the respective set operation on two neighborhoods.

(a) A neighborhood of the point with respect to A ∩* B is “partially full. The point is “on” the solid boundary.

(b) A neighborhood is empty. The point is “out” of the solid.



Tilove, IEEE Tr. on Comp., p.878

Curve/ Solid classification

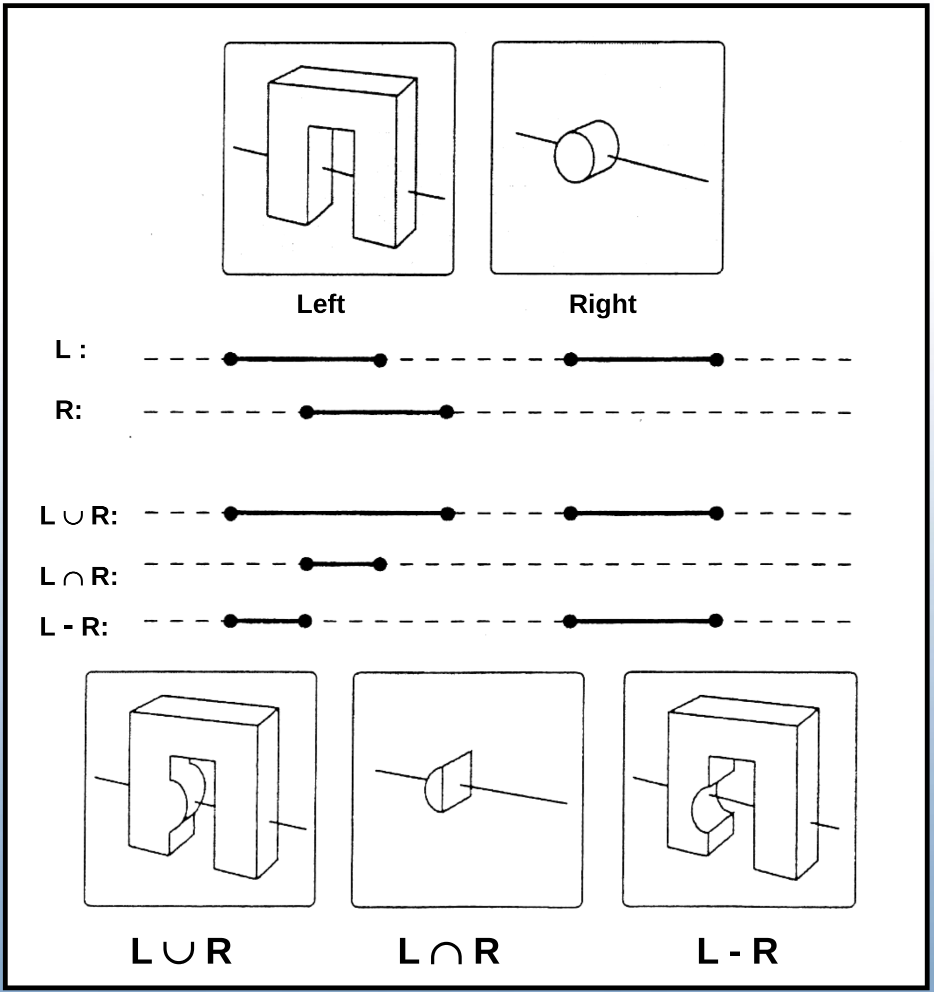
The algorithm structure:

1. Downward propagation
   - Send the line or curve description to the leaves of the CSG tree.
   - In a leaf, find the intersection points between the curve and the surface of the primitive.
   - Partition the curve into segments labeled ”in”, “out” or “on” the surface of the primitive

2. Upward Propagation

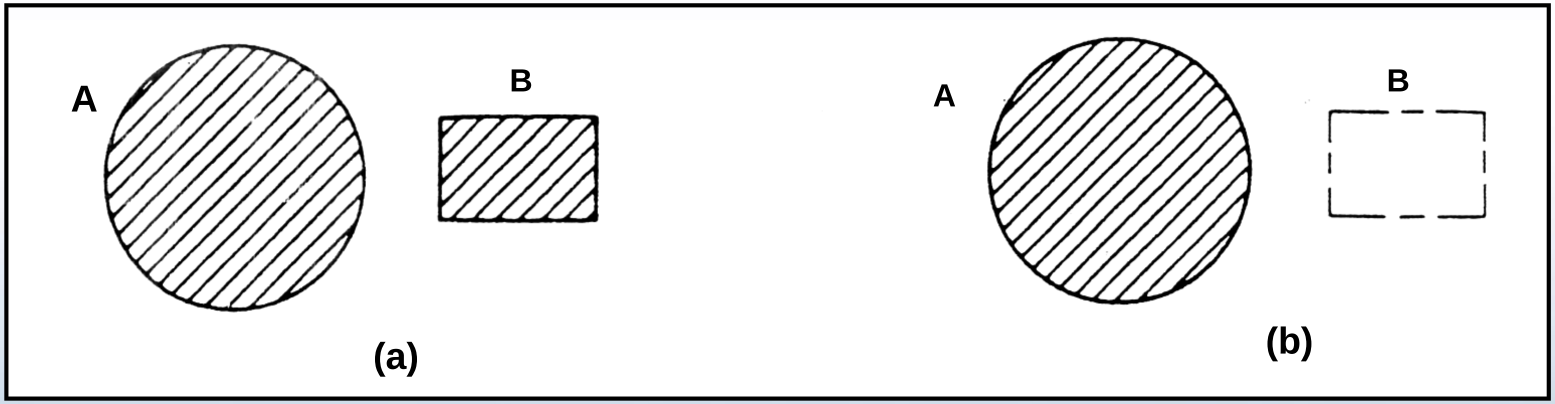
- Propagate the segments upward to the set operation nodes
- Merge the labeled segments in accordance with the operation.

Curve/Solid classification



Foley, p.712

Special cases in CSG trees

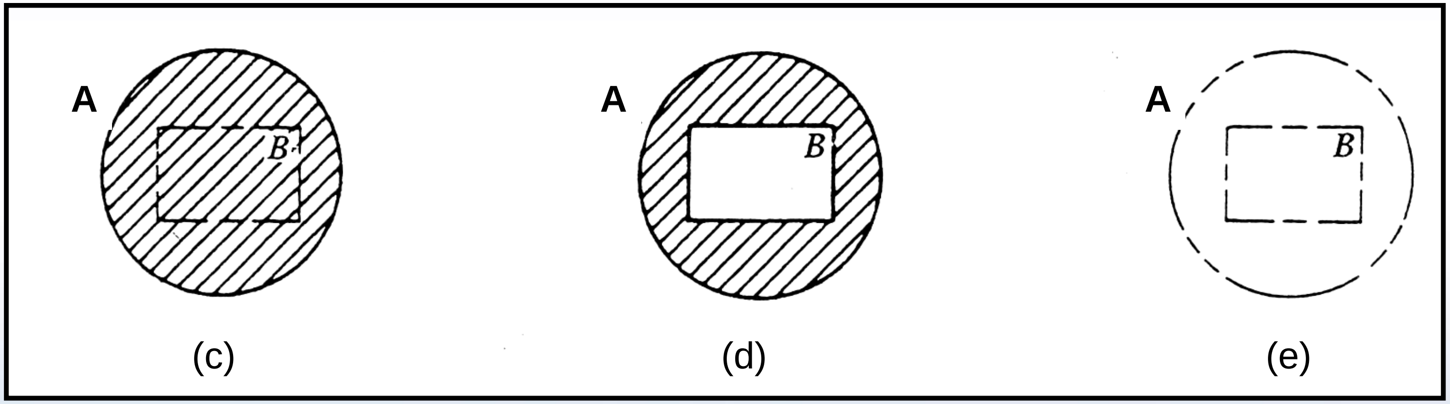


(a) $A \cup B$ *The union of two disjoint primitives.*

(b) $A \setminus B$ *The difference of two disjoint primitives.*

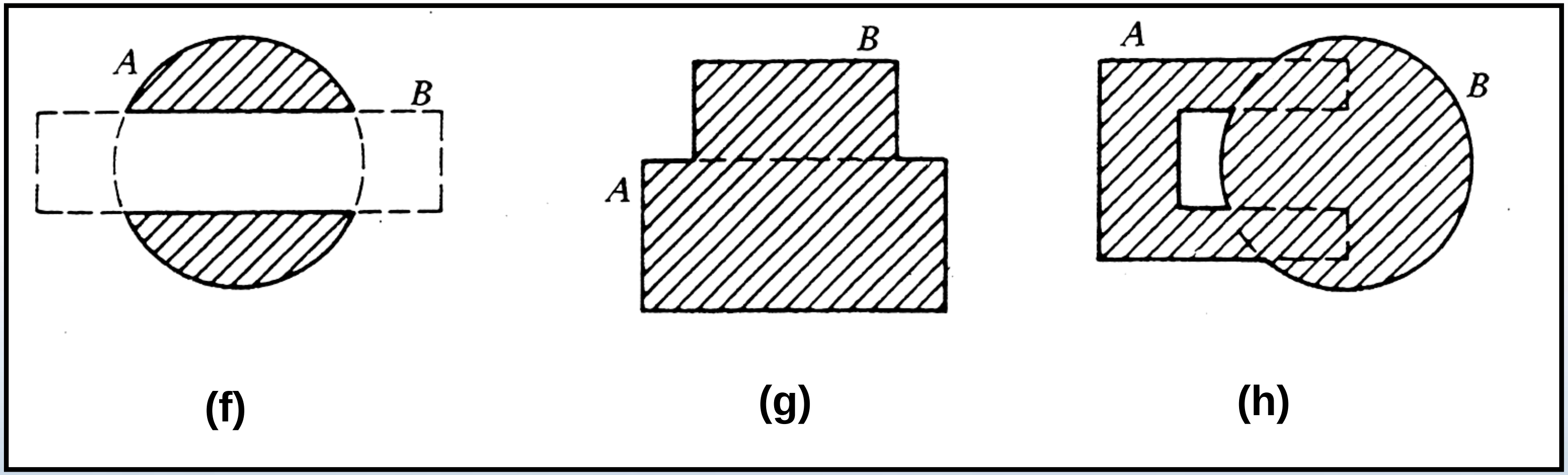
Mortenson, p.444

Special cases in CSG trees



- (c) $A \cup B$ The union of two primitives where one wholly contains the other
- (d) $A \setminus B$ The difference of two primitives where A wholly contains B
- (e) $\neg A \setminus B$

Special cases in CSG trees

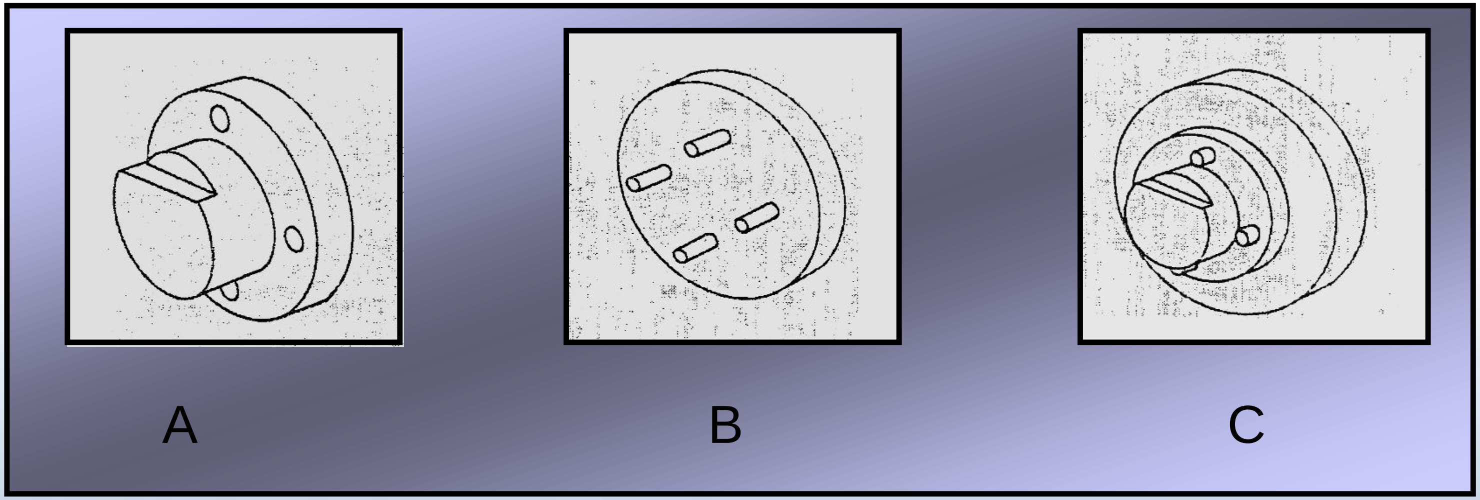


(f) $A \setminus B$ Two or more new objects.

(g) $A \cup B$ The union of two primitives that are tangent.

(h) $A \cup B$ The union creates inner loops or cavities.

CSG representation of a null set



$$\mathbf{A} \cup^{*} \mathbf{B} = \mathbf{C}$$

$$\mathbf{A} \cap^{*} \mathbf{B} = \varnothing$$

Tilove CACM 1984, p.686

Redundancies in CSG trees

- A redundant subtree is one that can be eliminated without altering the object defined by the CSG tree.

  If a tree represents empty space, it is said to define the null object

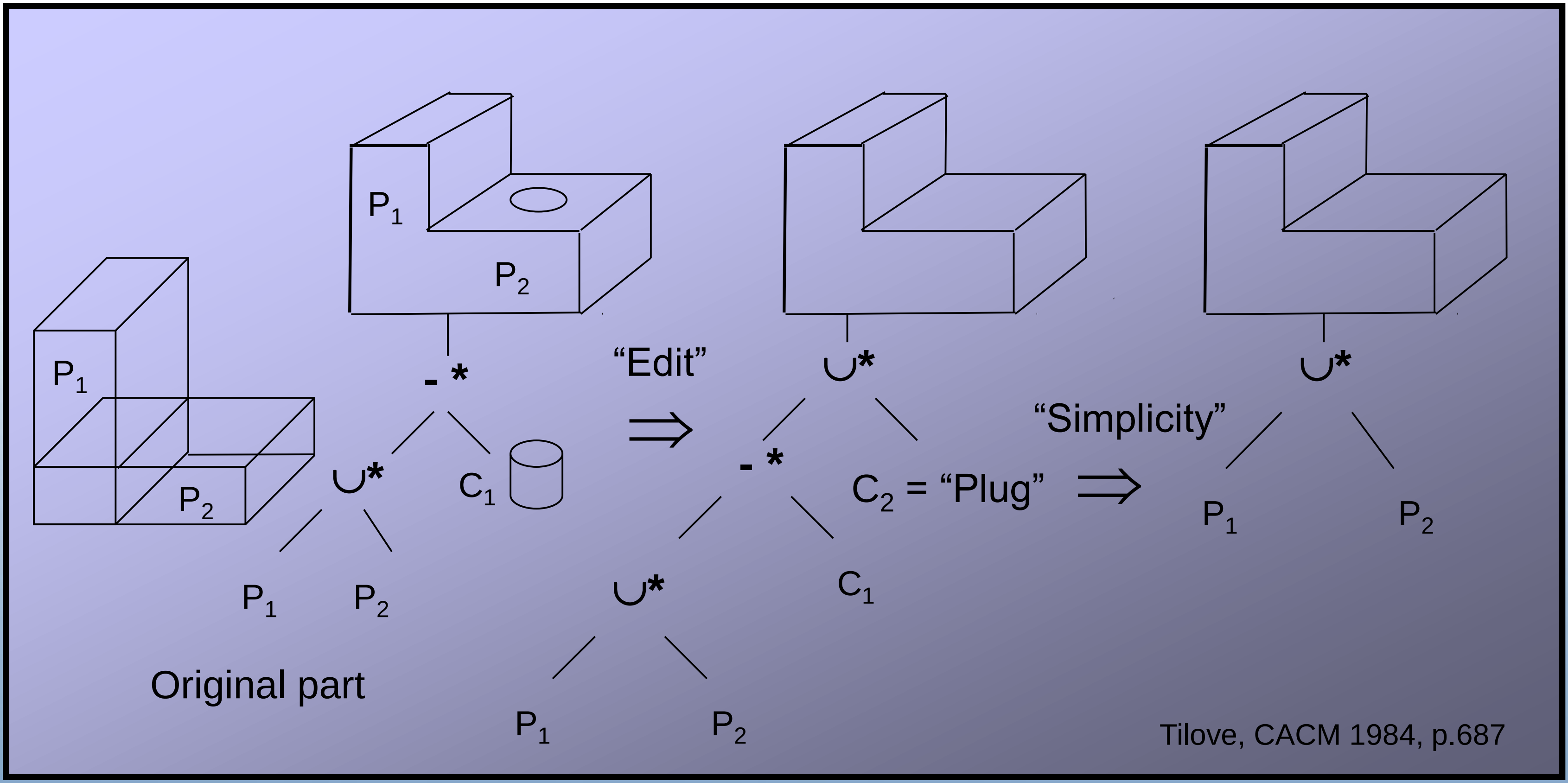
- The problem: for two given solids *A* and *B*, determine whether or not

$$\boldsymbol{A \cap^* B = \varnothing \quad ?}$$

- Applications: collision detection, packaging studies, design verification.

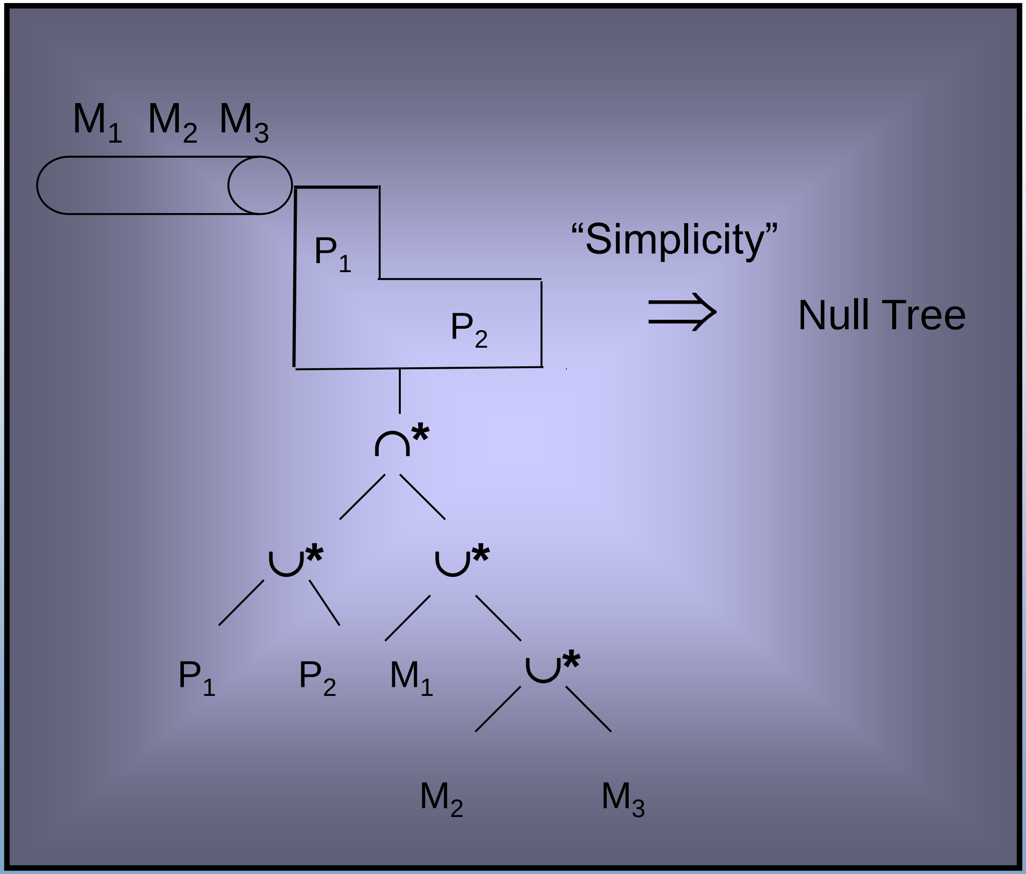
Redundancies in CSG trees

Simplification (reduction) of CSG trees:



Redundancies in CSG trees

Simplification (reduction) of CSG trees:



Tilove, CACM 1984, p.687

PADL-2 system characteristics

PADL-2 is a hybrid CSG/ B-rep modeler with CSG as a primary representation.

Geometric_coverage

The core system should cover 90-95% of typical unsculptured industrial parts.

The domain is the class of objects representable by bounded compositions of planar, cylindrical, conical, spherical, and toroidal halfspaces using the (general) regularized set operators (union, intersection, difference) plus an aggregation operator for "**assemble**" and (general) rigid motions (translations and rotations).

PADL-2 system characteristics

- **Informational completeness**

  The core system should contain at least one formally complete representation scheme

- **Validity and consistency**

  Nonsense objects should be excluded automatically

- **Extensibility**

  The design should permit geometric coverage to be extended and new applications to be supported without redesign of software

PADL-2 system characteristics

- Efficient support of diverse applications
  Alternative representation schemes should be supported

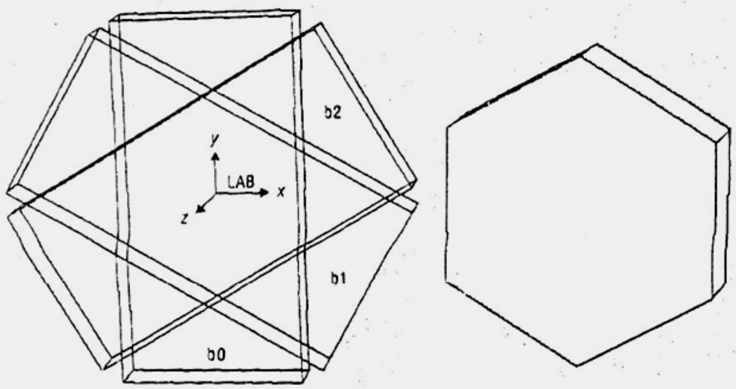
- Environment

  32-bit, medium-sized, virtual memory computers. The PADL-2 development system was a VAX-11/780 running the VMS operating system. Software is implemented in Fortran 77 for portability.

PADL-2 system user interface

- Three main input interfaces are:
- (1) interactive curve editor + curve-to-CSG converter;
- (2) general set of callable routines;
- (3) invertible text representation.
- CSG graphs can be inverse translated into text. Solid definitions arising from any input modality can be archived, edited, and used by other definitions.

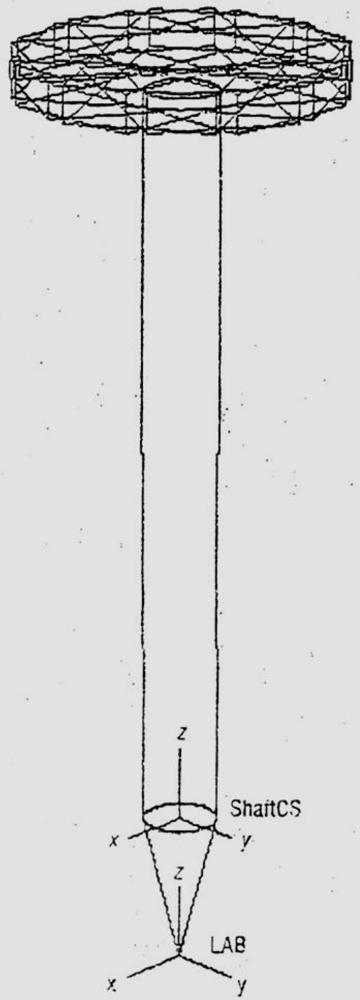
PADL-2 system user interface




```
GENERIC   Hex (HexPrism);
   b0        = BLO(z = Size/4, x = Size, y = 2*Size);
   m1        = (MOVX = -Size/2, MOVY = -Size);
   b1        = MOVE b0 BY (m1) WRT LAB;
   m2        = (ROTZ = (2*3.1416)/3);
   b2        = MOVE b1 BY (m2);
   b3        = MOVE b2 BY (m2);
   HexPrism  = b1 INT b2 INT b3;
   Size      = 10;
   END;
```

Brown, IEEE CG&A 1982, pp.73,80

PADL-2 system user interface

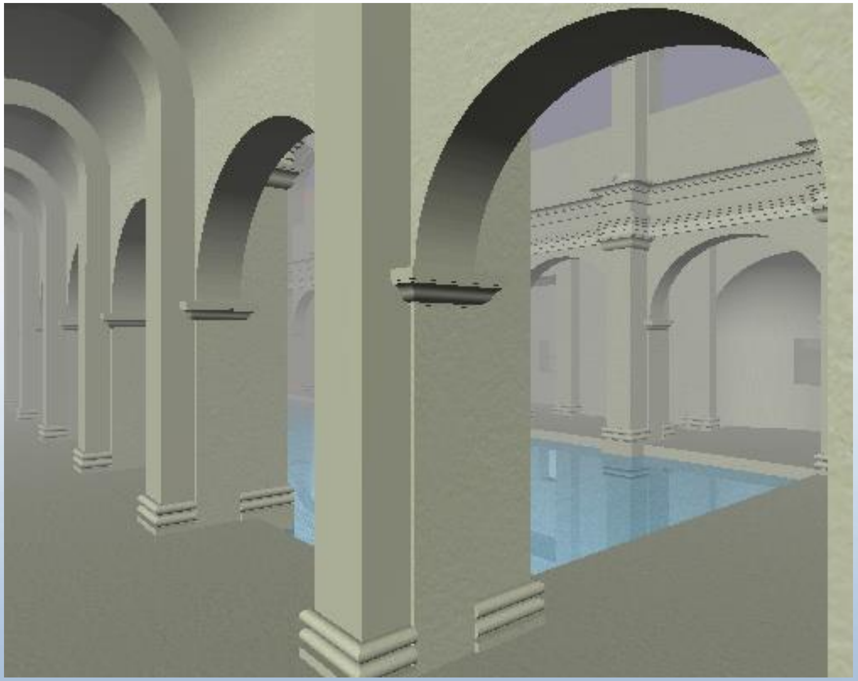



```
GENERIC  Nail  (NailRoot);
  Point     = CON(r = NailRad, h = PointHt);
  ShaftCs   = XFORM LAB BY (MOVZ = PointHt) WRT LAB;
  Shaft     = CYL(r = NailRad, h = NailHt) AT ShaftCS;
  HeadCS    = XFORM ShaftCS BY (MOVZ = NailHt) WRT ShaftCS;
  Head1     = Hex(Size = 4 * NailRad) AT HeadCS;
  Head2     = MOVE Head1 BY (ROTZ = 3.1416/6);
  NailRoot  = (Head1 INT Head2) UN Shaft UN Point;
  NailRad   = .5;
  NailHt    = 12 * NailRad;
  PointHt   = 3 * NailRad;
  END;
```

Brown, IEEE CG&A 1982, pp.73,80

svLis

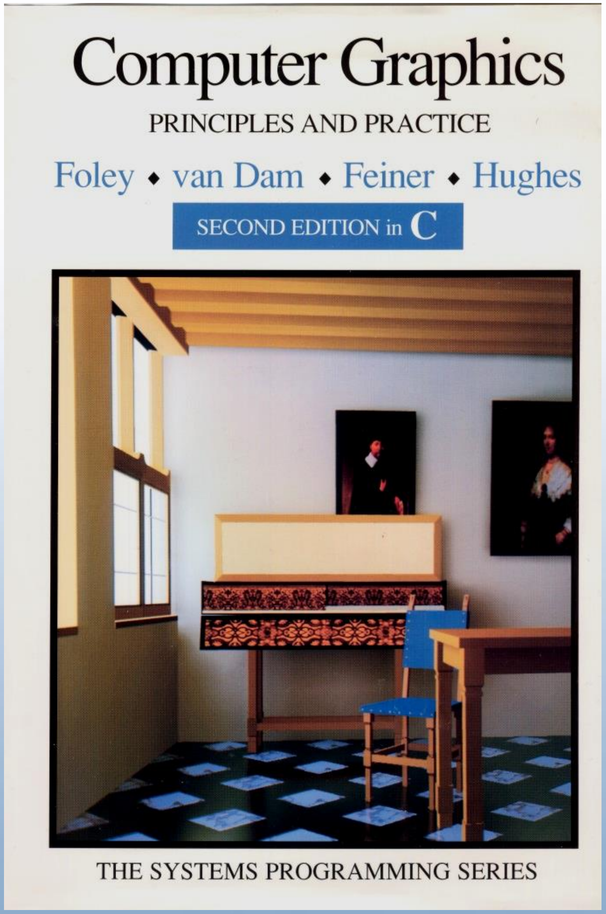
- svLis set-theoretic solid modeller
- Extensibility of primitives
- Algebraic operations on primitives
- Pure CSG tree
- GPL license



The svLis model of the Great Bath in in Aquae Sulis as it was in 200 AD

References

- James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics: Principles and Practice (2nd Edition in C), Addison-Wesley, Reading, MA, 1997.



References

- Michael E. Mortenson, Geometric Modeling, John Wiley and Sons, 1985.
  Second edition, 1997, Third edition, 2006.

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

- Robert B. Tilove
  “Set membership classification: a unified approach to geometric intersection problems”, IEEE Transactions on Computers, vol.C-29, N10, 1980, pp.874-883.

References

- R. B. Tilove and A.A.G. Requicha
  "Closure of Boolean operations on geometric entities", Computer-Aided Design, vol.12, N5, 1980, pp.219-220.

- Robert B. Tilove
  "A null-object detection algorithm for constructive solid geometry", Communications of the ACM, vol.27, N7, 1984, pp.684-694.

- Christopher M. Brown
  "PADL-2: a technical summary", IEEE Computer Graphics and Applications, vol.2, N2, 1982, pp.69-84.

References

- Brown C., PADL2: A Technical Summary. *IEEE Computer Graphics and Applications, 2* (March 1982), pp. 69-84.