CSG:
Constructive Solid Geometry

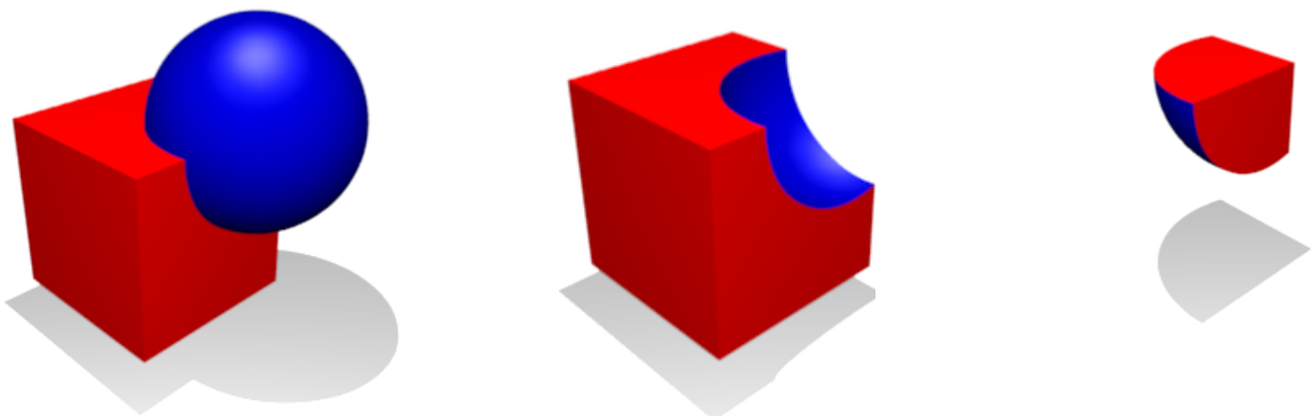
Solid Geometry

- So far, we have represented 3D objects by specifying their boundary: this is called Boundary Representation, or B-Rep.
- Sometimes we need to represent objects explicitly as solids.

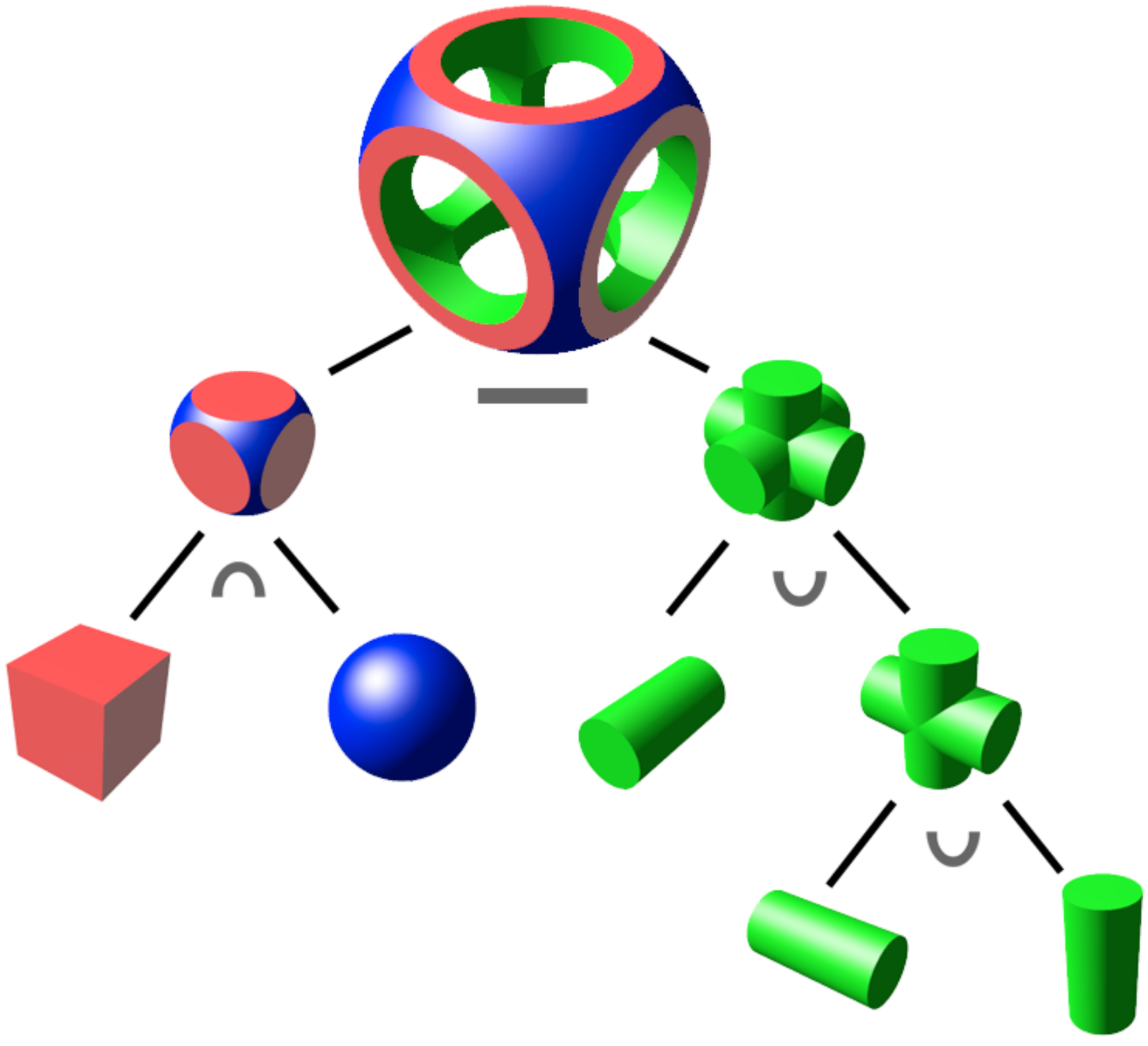
- CSG represents solid objects by constructing them from solid primitives using boolean operations.

CSG

- Primitives:
  - Plane (half-space)
  - Boxes, spheres, cones, cylinders, …
- Affine transformations:
  - To modify canonical primitives
  - To get the primitives where we want them
- Boolean operations:
  - Union (∪), Intersection (∩), Difference (\)

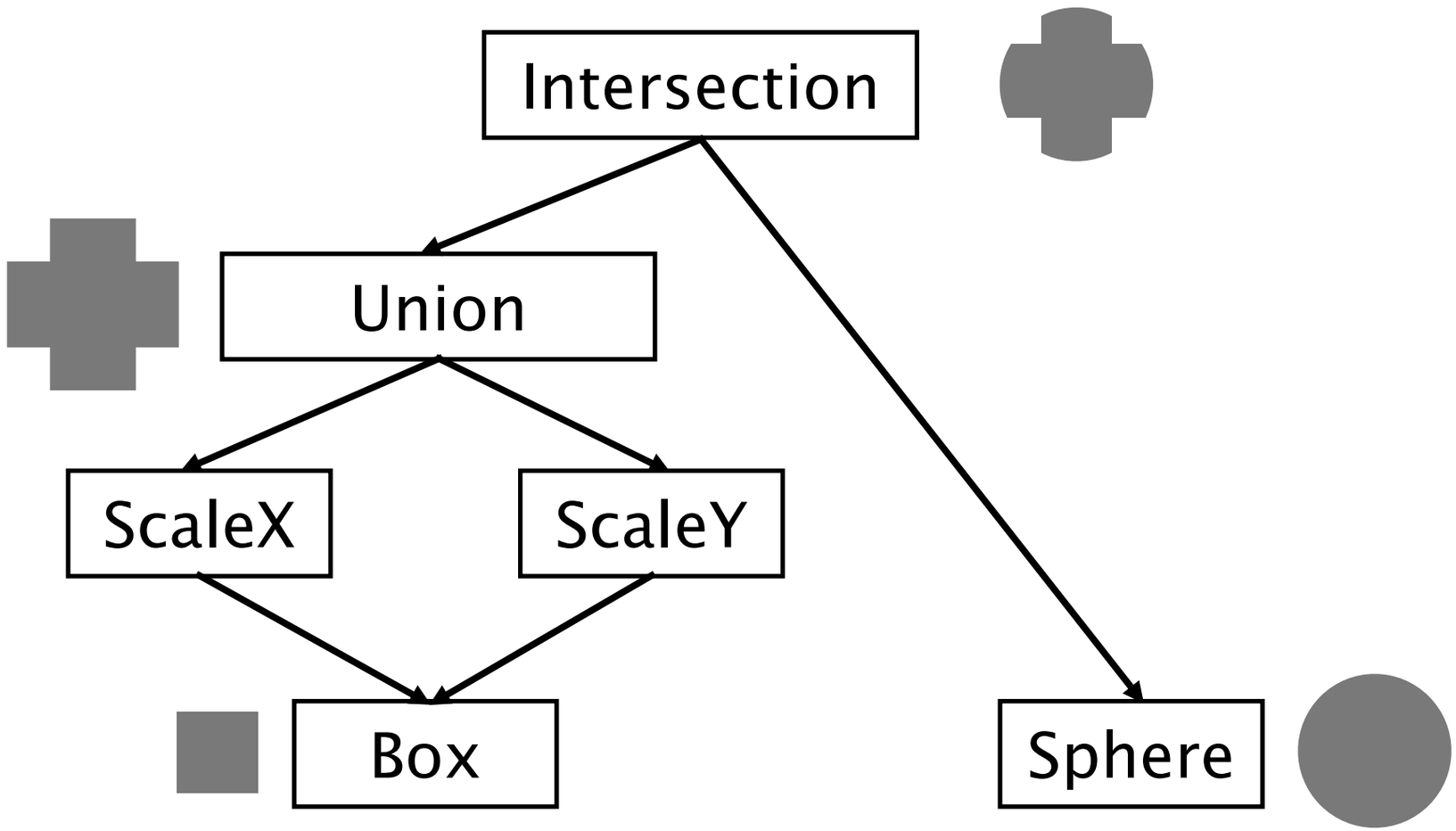


CSG



CSG Graph

- The primitives, affine transformations, and boolean operators are arranged as a DAG: Directed Acyclic Graph:



Point Membership Classification

- PMC query: given a solid S and a point p, does p belong to S?
- *PMC(CSGnode A, Point p)*

```
If PrimitiveNode(A) return A.DirectPMC(p)
If AffineNode(A) {
        T = A.GetMatrix();
        Return PMC(A.GetChild(), T-1(p))
}
Left = PMC(A.GetLeftChild(), p)
Right = PMC(A.GetRightChild(), p)
If Intersection(A) return IntersectionTable(Left, Right)
If Union(A) return UnionTable(Left, Right)
If Difference(A) return DifferenceTable(Left, Right)
```

Example: Intersection Table

	In	**Out**	**On**
In	In	Out	On
Out	Out	Out	Out
On	On	Out	?