

# CSG Logical Operators

This example creates a simple geometry and shows how the various logical operators work.

First we will create two circles and halfspace regions that we will use in our examples.

```
In [1]: # Load the Library
using ConstructiveSolidGeometry

left_circle = InfCylinder(Coord(-0.25, 0.0, 0.0), unitize(Coord(0.0,
0.0, 1.0)), 0.5)
right_circle = InfCylinder(Coord(0.25, 0.0, 0.0), unitize(Coord(0.0, 0
.0, 1.0)), 0.5)

regions = Array{Region}(0)
push!(regions, Region(left_circle, -1))
push!(regions, Region(right_circle, -1));
```

With these surfaces to use, we can experiment with combining them using different operators to form cells

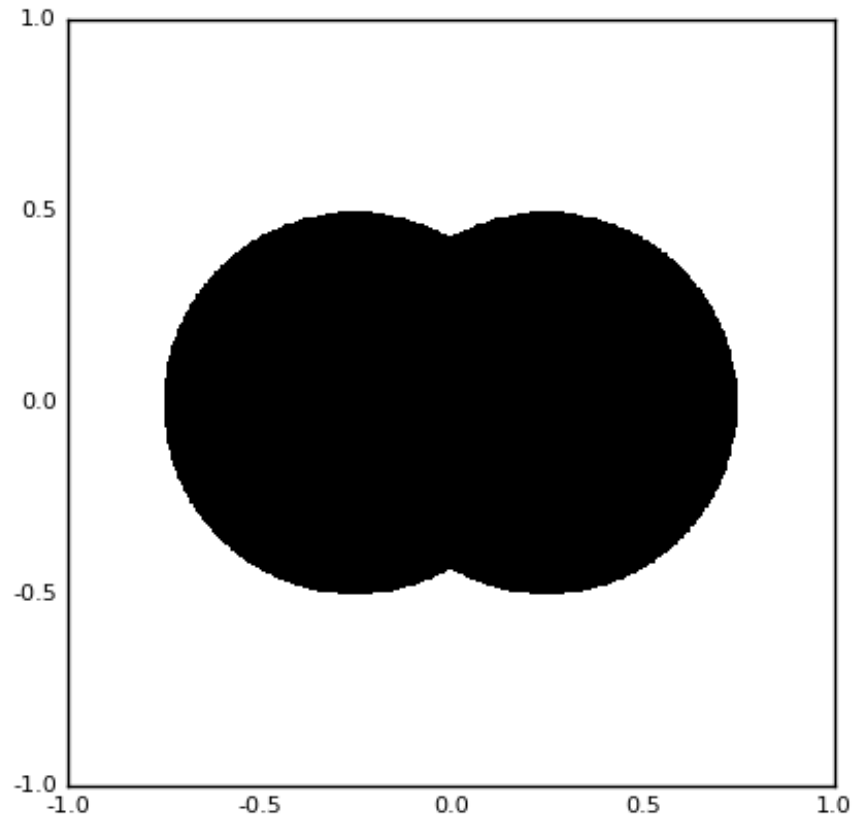
## Union of two circles

```
In [2]: definition = :(1 | 2)

        cells = Array{Cell}(0)
        push!(cells, Cell(regions, definition));
        bounding_box = Box(Coord(-1.0, -1.0, -1.0), Coord(1.0, 1.0, 1.0))
        geometry = Geometry(cells, bounding_box)

        view = Box(Coord(-1.0, -1.0, 0), Coord(1.0, 1.0, 0))
        plot_cell_2D(geometry, view , 500, 1)
```

Out[2]:



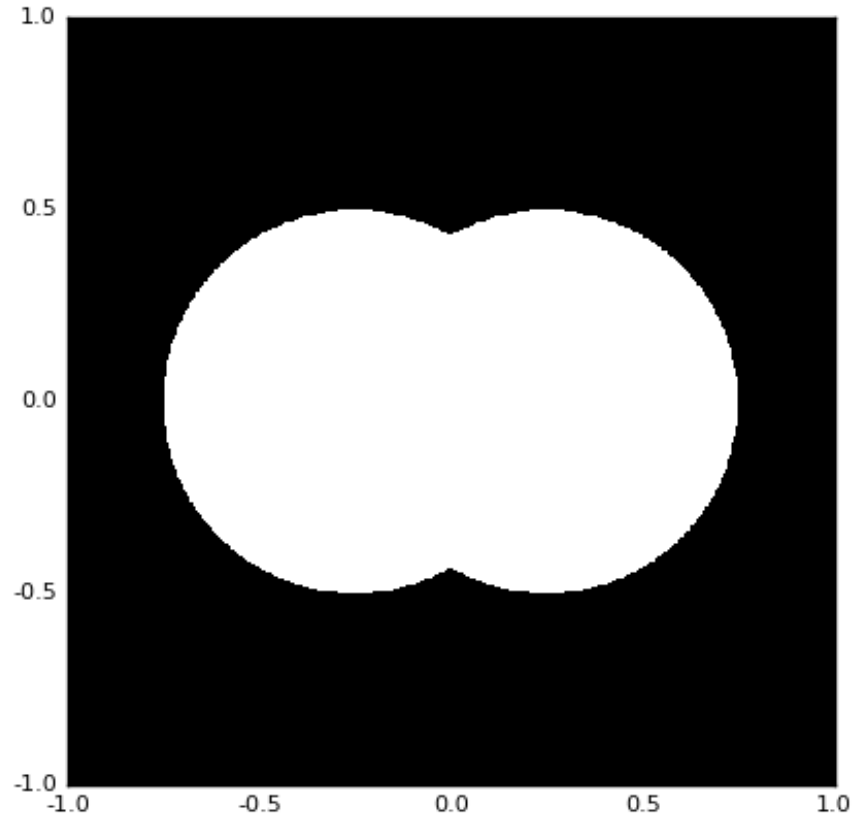
## Complement of the union of two circles

```
In [3]: definition = ~(1 | 2)

        cells = Array{Cell}(0)
        push!(cells, Cell(regions, definition))
        geometry = Geometry(cells, bounding_box)

        plot_cell_2D(geometry, view, 500, 1)
```

Out[3]:



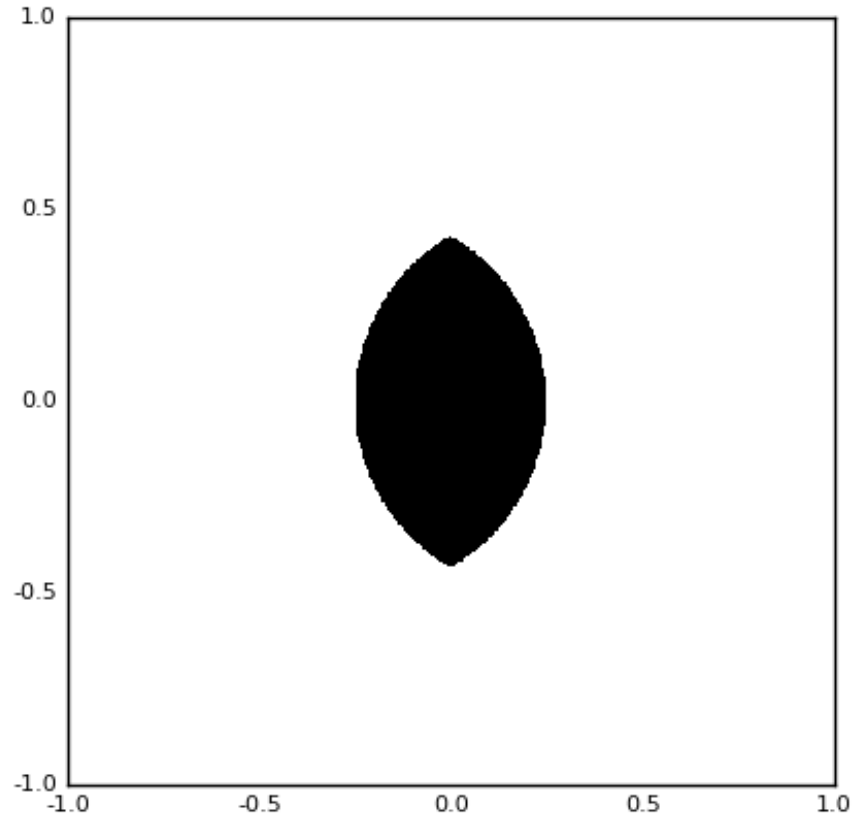
## Intersection of two circles

```
In [4]: definition = :(1 ^ 2)

        cells = Array{Cell}(0)
        push!(cells, Cell(regions, definition))
        geometry = Geometry(cells, bounding_box)

        plot_cell_2D(geometry, view, 500, 1)
```

Out[4]:

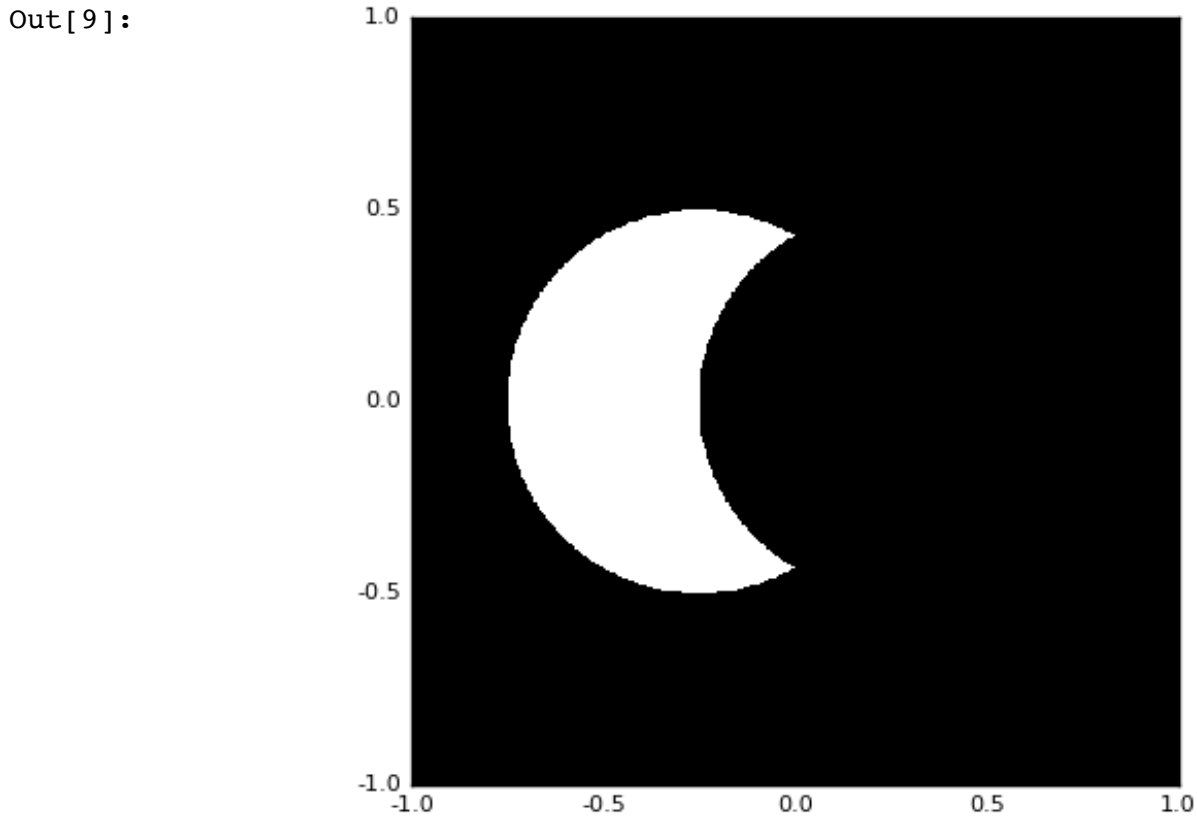


## Crescent Moon

```
In [9]: definition = ~(1 ^ ~2)

        cells = Array{Cell}(0)
        push!(cells, Cell(regions, definition))
        geometry = Geometry(cells, bounding_box)

        plot_cell_2D(geometry, view, 500, 1)
```



## Circles Cancelling

```
In [16]: definition = :(~(1 ^ 2) ^ (1 | 2))

cells = Array{Cell}(0)
push!(cells, Cell(regions, definition))
geometry = Geometry(cells, bounding_box)

plot_cell_2D(geometry, view, 500, 1)
```

Out[16]:

