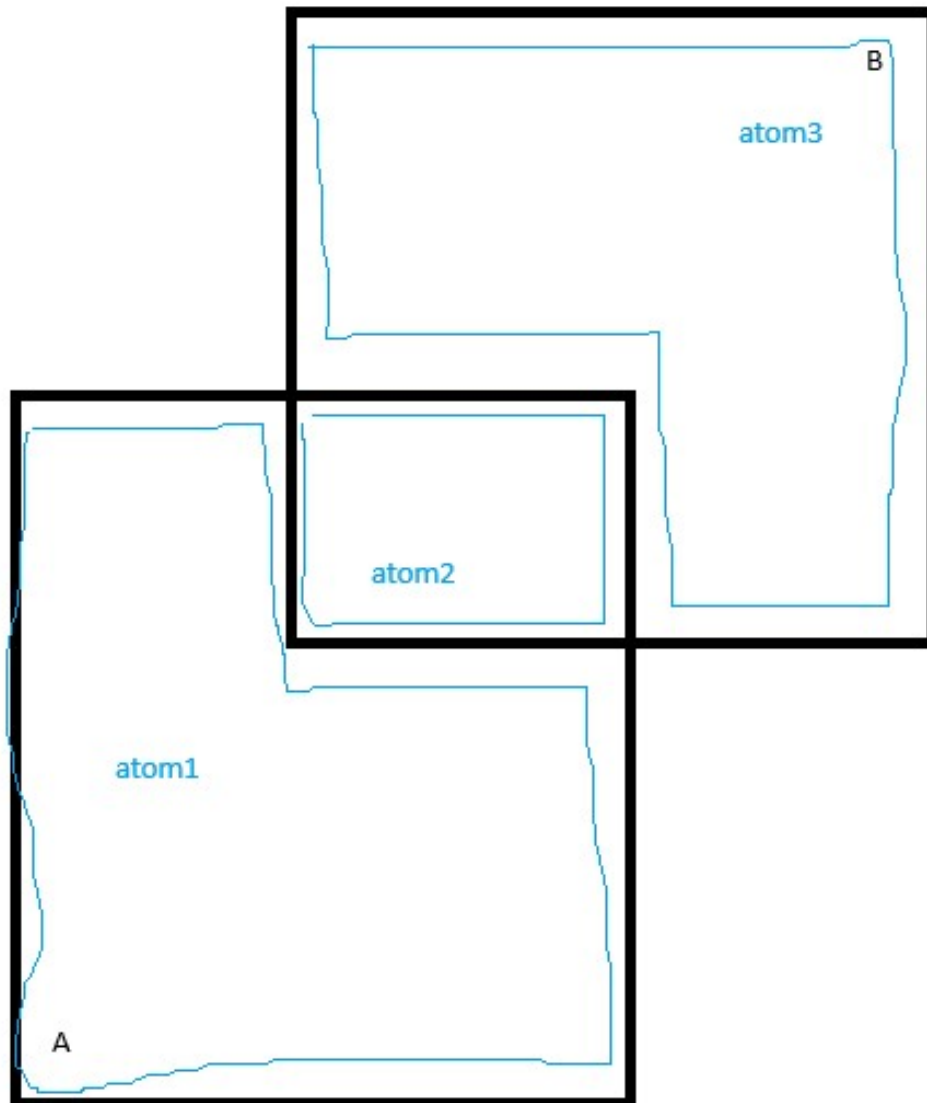


CASE WITH TWO LAR INPUT ARGUMENTS (A,B)



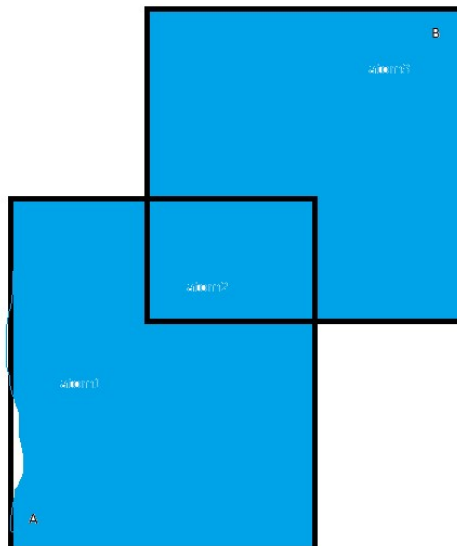
UNION(A,B)

	A	B	Result
Atom1	1	0	1 (==1 0)
Atom2	1	1	1 (==1 1)
Atom3	0	1	1 (==0 1)

```
function Union(v::AbstractArray)
    return any(v)
end

@assert(Union([false, false])==false)
@assert(Union([true, false])==true)
@assert(Union([true, true])==true)
@assert(Union([false, true])==true)
```

NOTE: Result is needed to know if an atom will be included in the boolean operation, or must be excluded

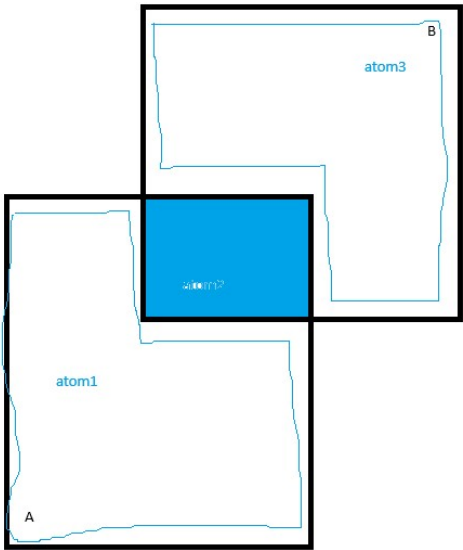


INTERSECTION(A,B)

	A	B	Result
Atom1	1	0	0 (==1 & 0)
Atom2	1	1	1 (==1 & 1)
Atom3	0	1	0 (== 0 & 1)

```
function Intersection(v::AbstractArray)
    return all(v)
end

@assert(Intersection ([false, false])==false)
@assert(Intersection ([true, false])==false)
@assert(Intersection ([true, true])==true)
@assert(Intersection ([false, true])==false)
```

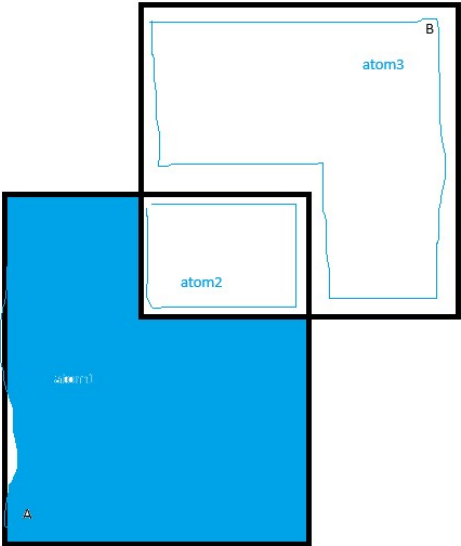


DIFFERENCE(A,B) == A-B

	A	B	Result
Atom1	1	0	1 (==1-0)
Atom2	1	1	0 (==1-1)
Atom3	0	1	0 (==0-1)

```
function Difference(v::AbstractArray)
    return v[1] && !any(v[2:end])
end

@assert(Difference ([false, false])==false)
@assert(Difference ([true, false])==true)
@assert(Difference ([true, true])==false)
@assert(Difference ([false, true])==false)
```

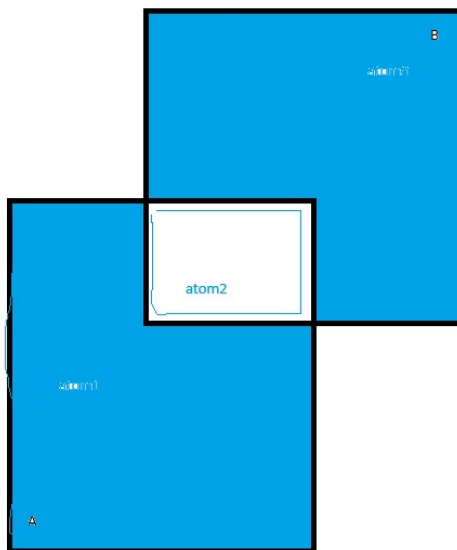


XOR(A,B)

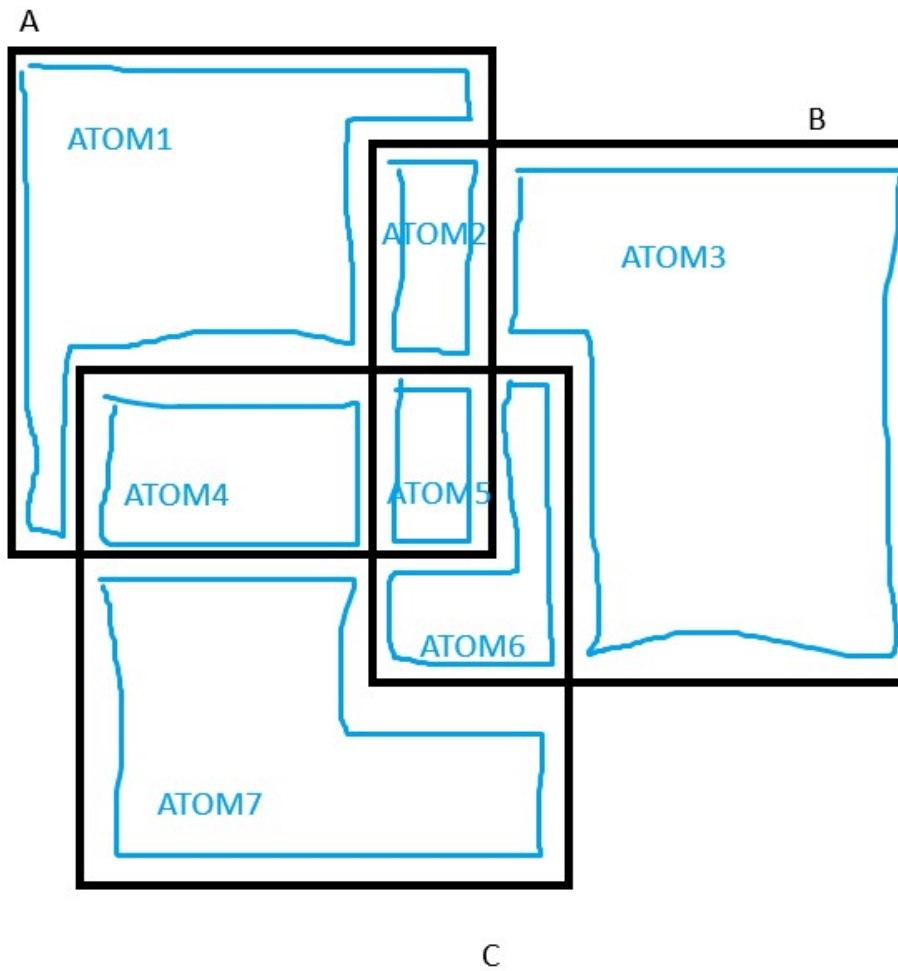
	A	B	Result
Atom1	1	0	1 ($==1 \vee 0$)
Atom2	1	1	0 ($==1 \vee 1$)
Atom3	0	1	1 ($==0 \vee 1$)

```
function Xor(v::AbstractArray)
    return (length([it for it in v if it]) % 2) == 1
end
```

```
@assert(Xor([false, false]) == false)
@assert(Xor([true, false]) == true)
@assert(Xor([true, true]) == false)
@assert(Xor([false, false]) == true)
```

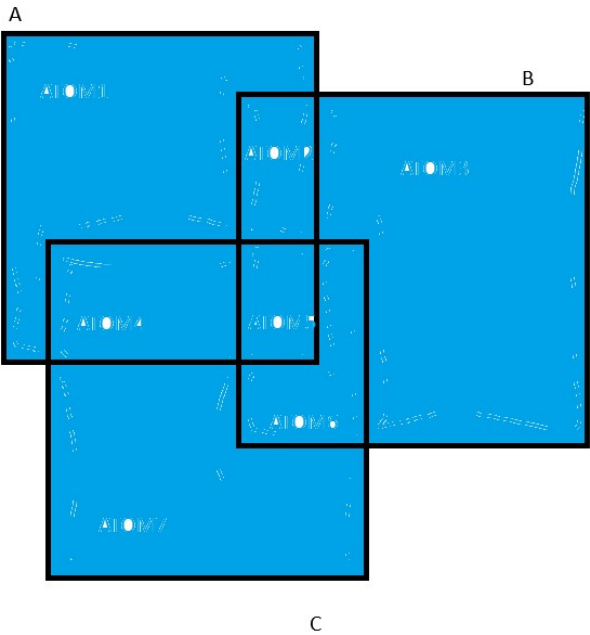


CASE WITH 3 LAR INPUT ARGUMENTS (A,B,C)



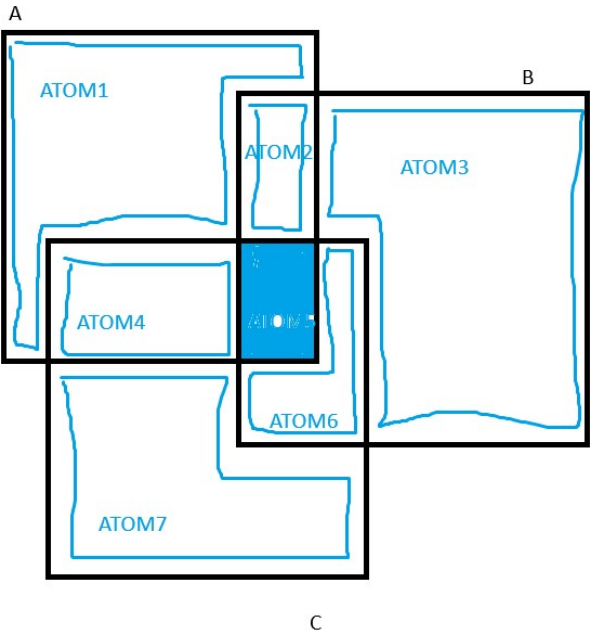
UNION(A,B,C)

	A	B	C	Result
Atom1	1	0	0	1 (==1 0 0)
Atom2	1	1	0	1
Atom3	0	1	0	1
Atom4	1	0	1	1
Atom5	1	1	1	1
Atom6	0	1	1	1
Atom7	0	0	1	1



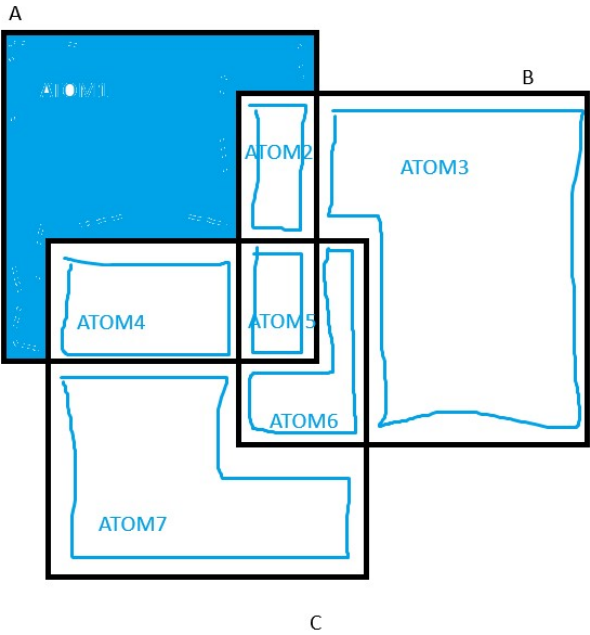
INTERSECTION(A,B,C)

	A	B	C	Result
Atom1	1	0	0	0
Atom2	1	1	0	0
Atom3	0	1	0	0
Atom4	1	0	1	0
Atom5	1	1	1	1
Atom6	0	1	1	0
Atom7	0	0	1	0



DIFFERENCE(A,B,C)==A-(B UNION C)

	A	B	C	Result
Atom1	1	0	0	1
Atom2	1	1	0	0
Atom3	0	1	0	0
Atom4	1	0	1	0
Atom5	1	1	1	0
Atom6	0	1	1	0
Atom7	0	0	1	0



XOR(A,B,C)

	A	B	C	Result
Atom1	1	0	0	1
Atom2	1	1	0	0
Atom3	0	1	0	1
Atom4	1	0	1	0
Atom5	1	1	1	1
Atom6	0	1	1	0
Atom7	0	0	1	1

