Informatics 1A

Computation and Logic 1

a small universe

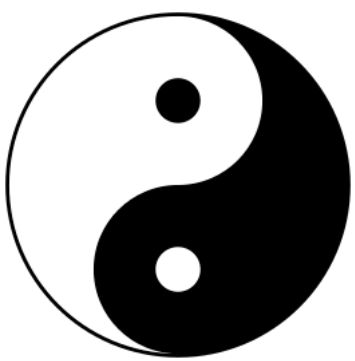
INF1A-CL

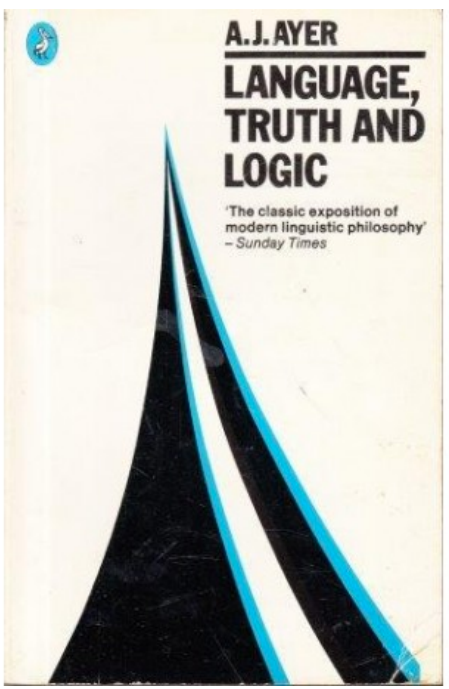
a small universe

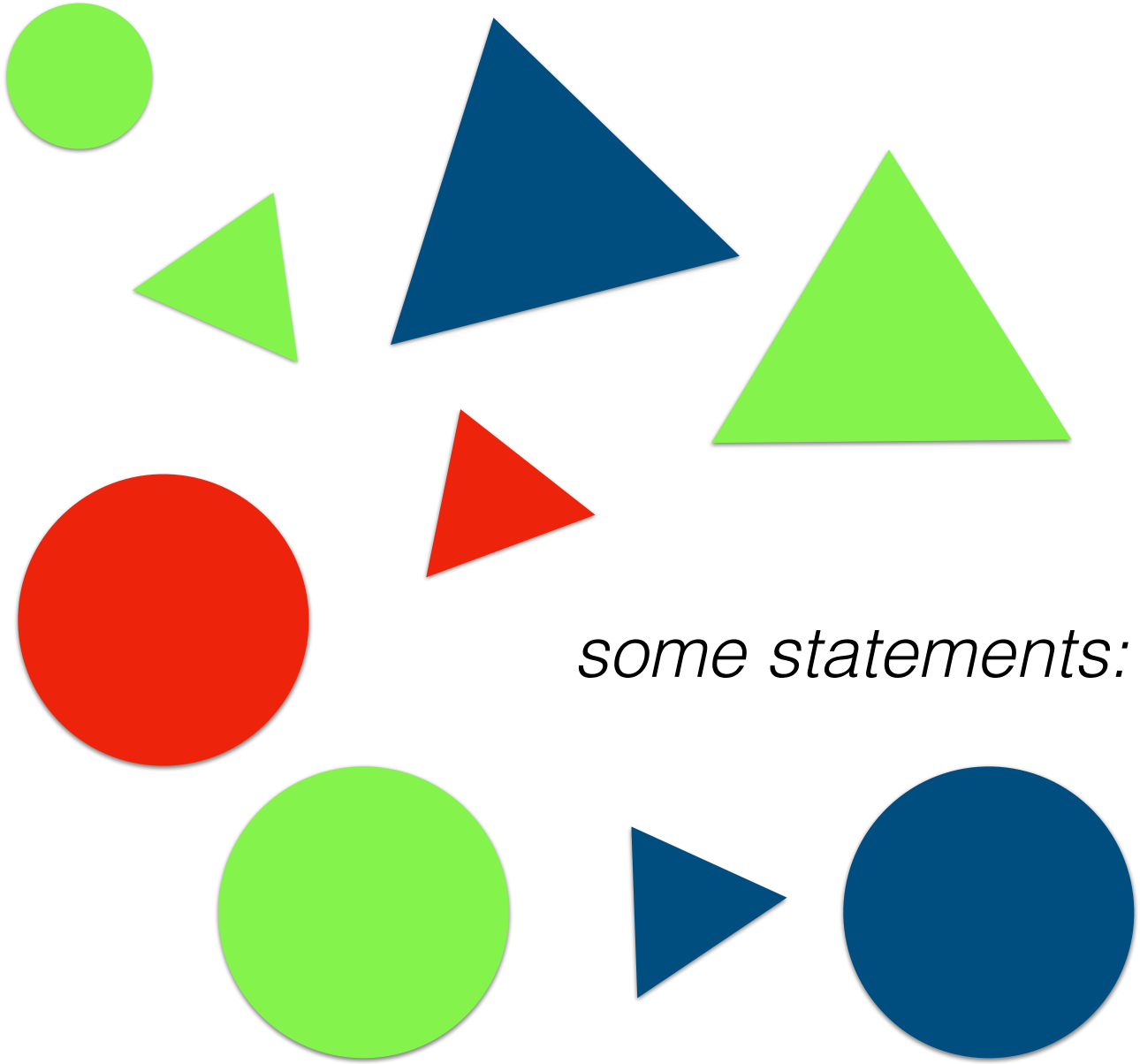
truth
logic
language

Michael P. Fourman

@mp4man







Blue	Amber
Green	Red

some statements:

every red triangle is small ✔

every small triangle is red ✘

some big triangle is green ?

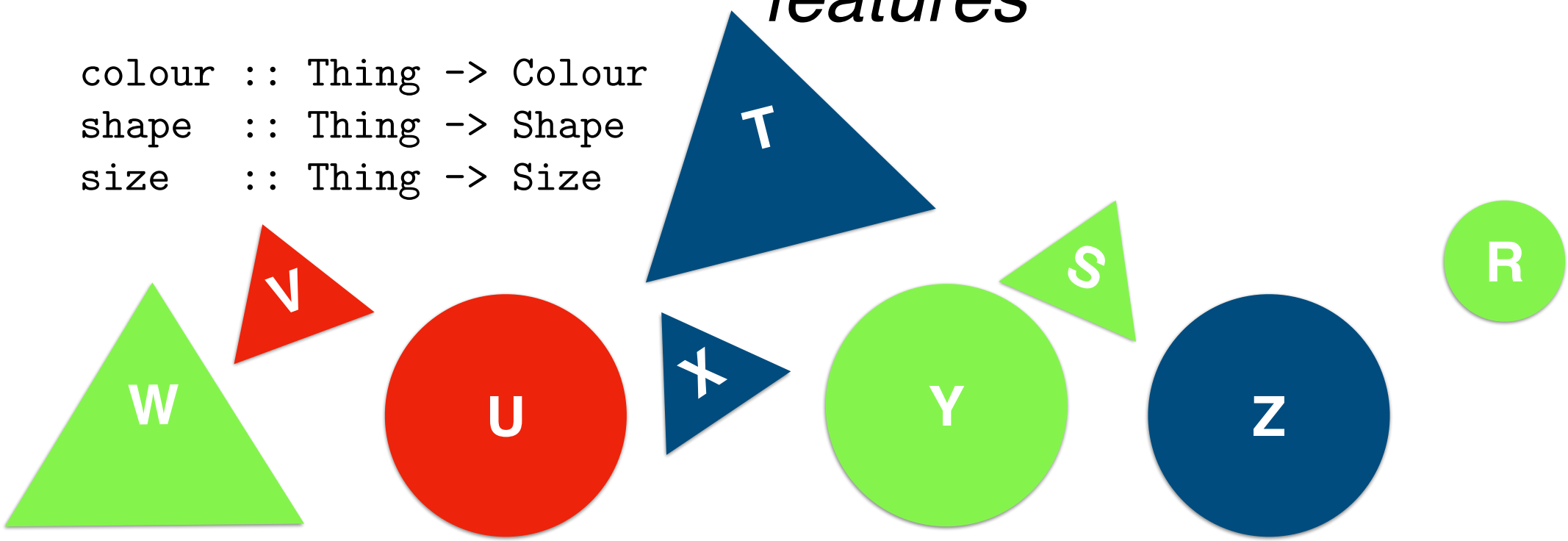
some small disc is red ?

no red thing is blue ?

```
data Thing  = R | S | T | U | V | W | X | Y | Z
things = [ R, S, T, U, V, W, X, Y, Z ]

data Colour = Red | Blue | Green
data Shape  = Disc | Triangle
data Size   = Big | Small

colour :: Thing -> Colour
shape  :: Thing -> Shape
size   :: Thing -> Size
```

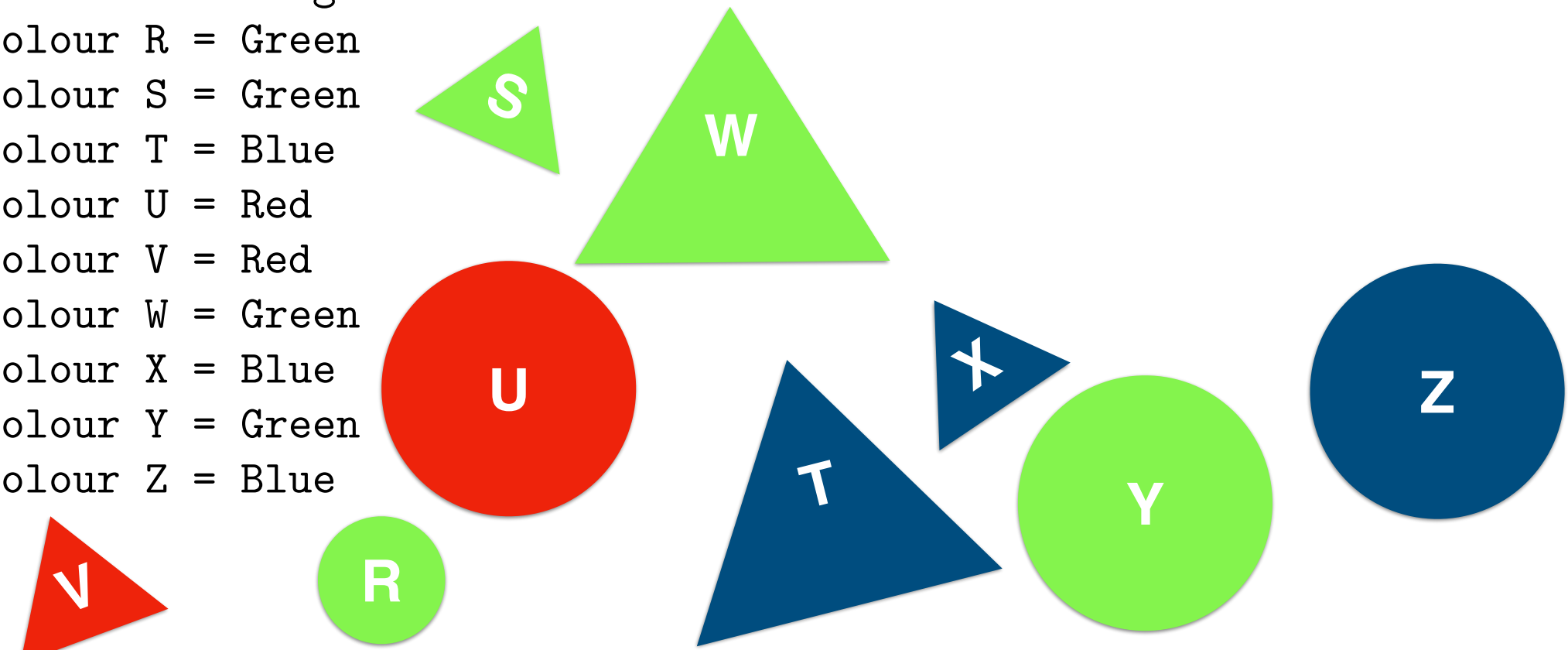
Blue	Amber
Green	Red

features



```
data Thing  = R | S | T | U | V | W | X | Y | Z
things = [ R, S, T, U, V, W, X, Y, Z ]

data Colour = Red | Blue | Green
colour :: Thing -> Colour
colour R = Green
colour S = Green
colour T = Blue
colour U = Red
colour V = Red
colour W = Green
colour X = Blue
colour Y = Green
colour Z = Blue
```

Blue	Amber
Green	Red



```
data Thing  = R | S | T | U | V | W | X | Y | Z
things = [ R, S, T, U, V, W, X, Y, Z ]
```

Blue	Amber
Green	Red

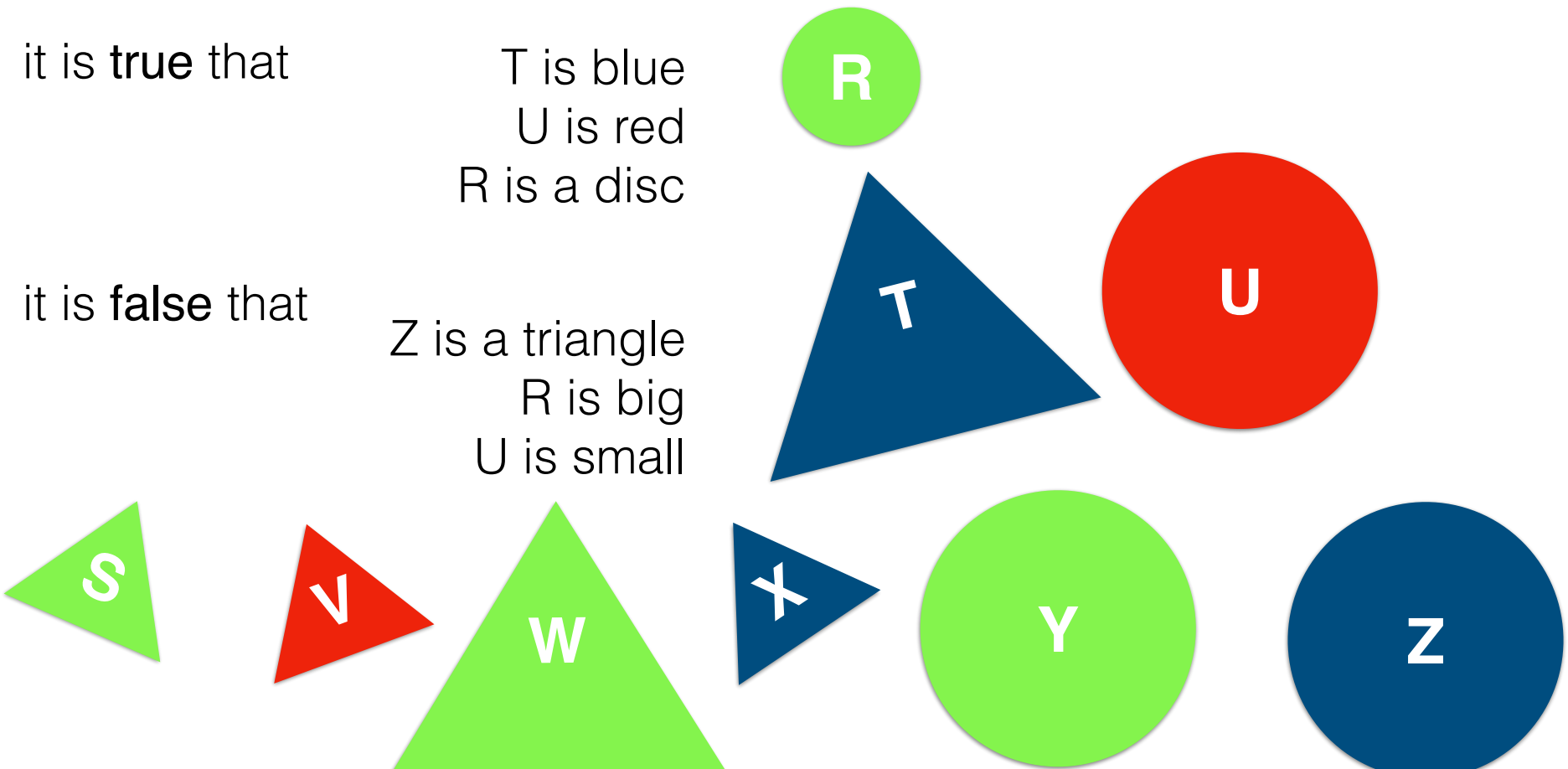
Idea: Use predicates to describe the universe!

it is **true** that

T is blue
U is red
R is a disc

it is **false** that

Z is a triangle
R is big
U is small

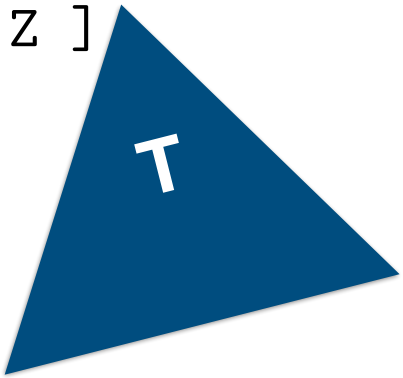


```
data Thing  = R | S | T | U | V | W | X | Y | Z
things = [ R, S, T, U, V, W, X, Y, Z ]
```

```
data Bool
```

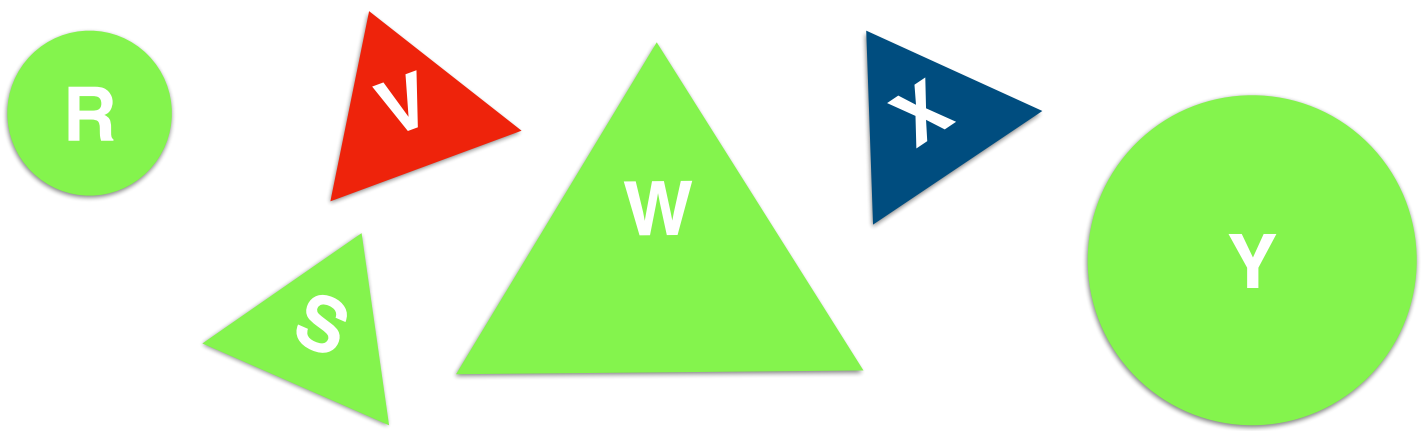
Constructors

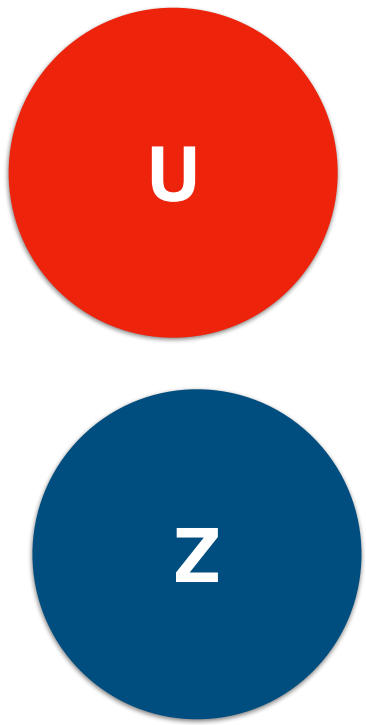
```
False
True
```



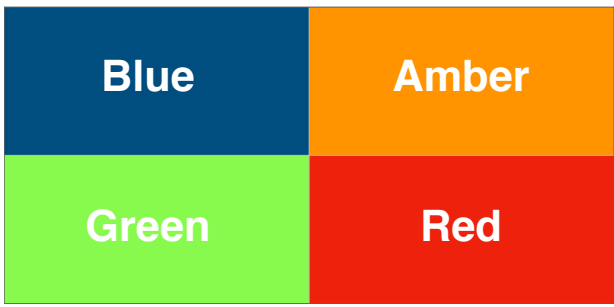
We will use predicates to describe the universe

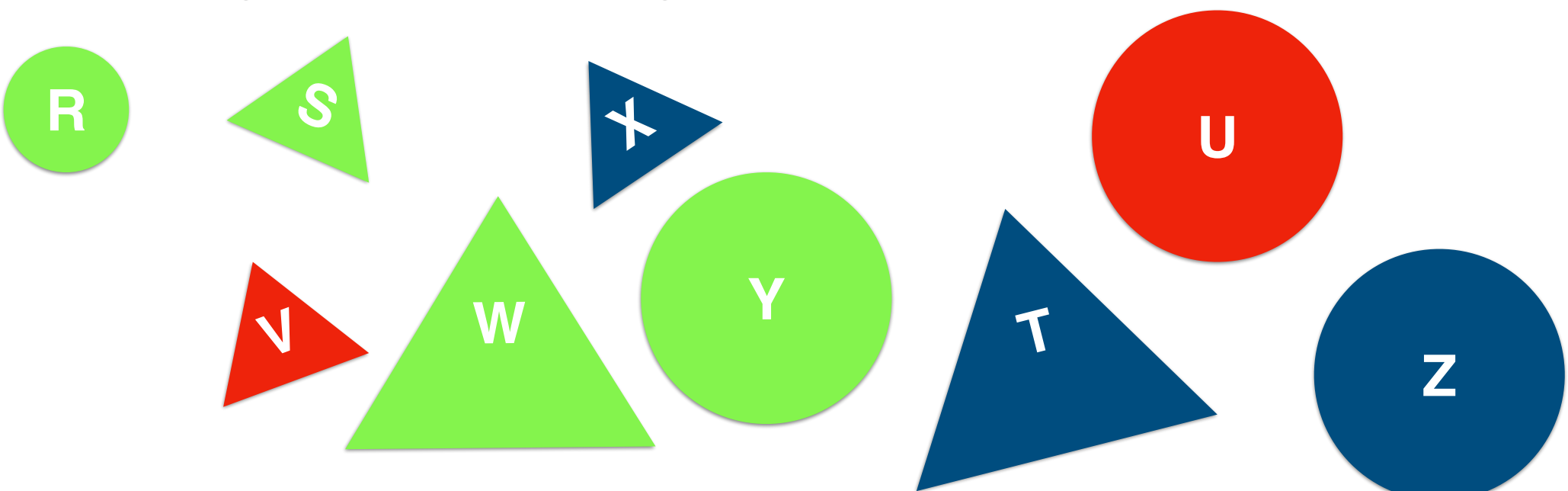
type Predicate = Thing -> Bool





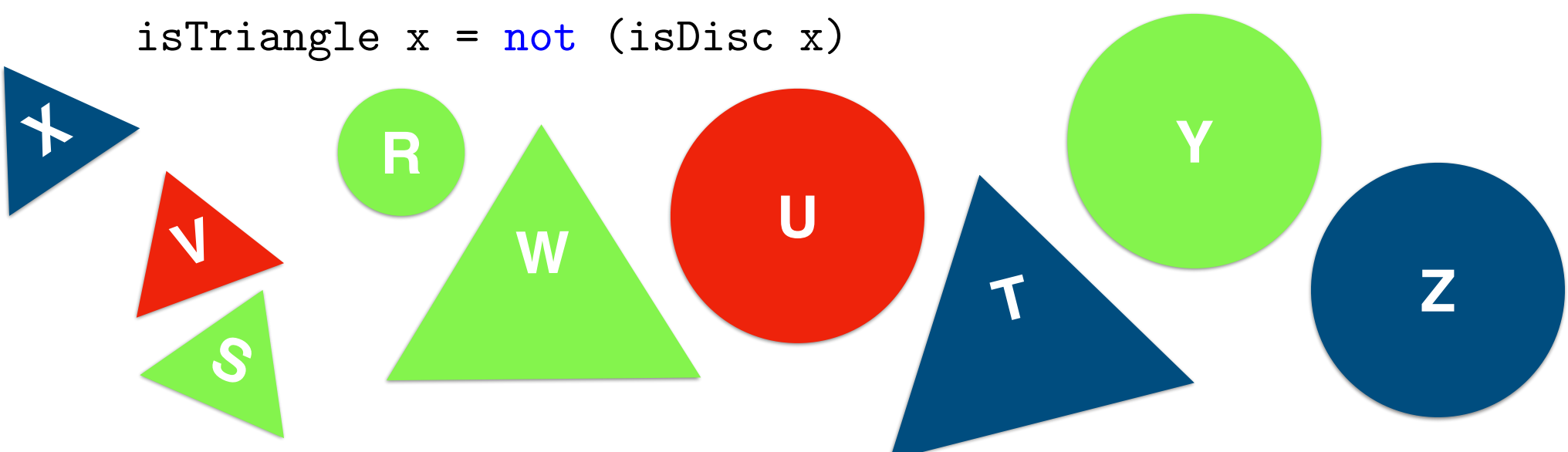
```
data Thing  = R | S | T | U | V | W | X | Y | Z

things :: [ Thing ]
things =  [ R, S, T, U, V, W, X, Y, Z ]

type Predicate = Thing -> Bool
isRed, isBlue, isGreen, isDisc,
  isTriangle, isSmall, isBig :: Predicate
```





```
data Thing = R | S | T | U | V | W | X | Y | Z

isDisc R = True
isDisc U = True
isDisc Y = True
isDisc Z = True
isDisc _ = False

isTriangle x = not (isDisc x)
```

Blue	Amber
Green	Red



Operations

```
data Bool = False | True
```

```
(&&) :: Bool -> Bool -> Bool    | infixr 3
```

Boolean "and"

```
(||) :: Bool -> Bool -> Bool    | infixr 2
```

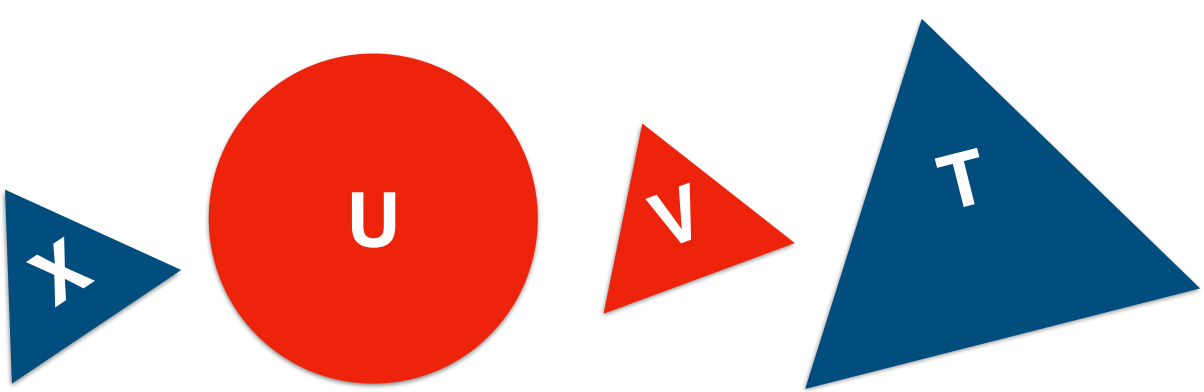
Boolean "or"

```
not :: Bool -> Bool
```

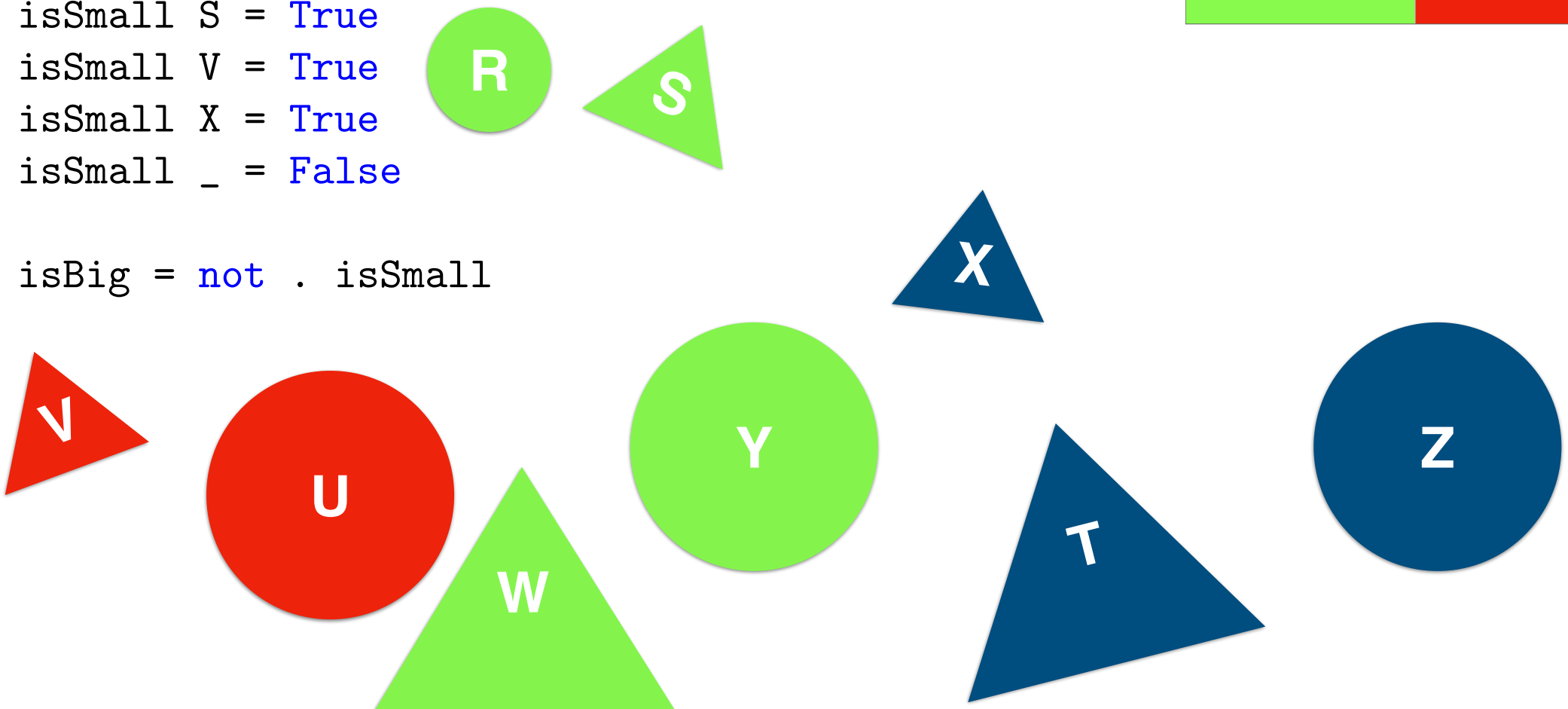
Boolean "not"

`True`	$\top$
`False`	$\bot$
`not`	$\neg$
`&&`	$\wedge$
`||`	$\vee$

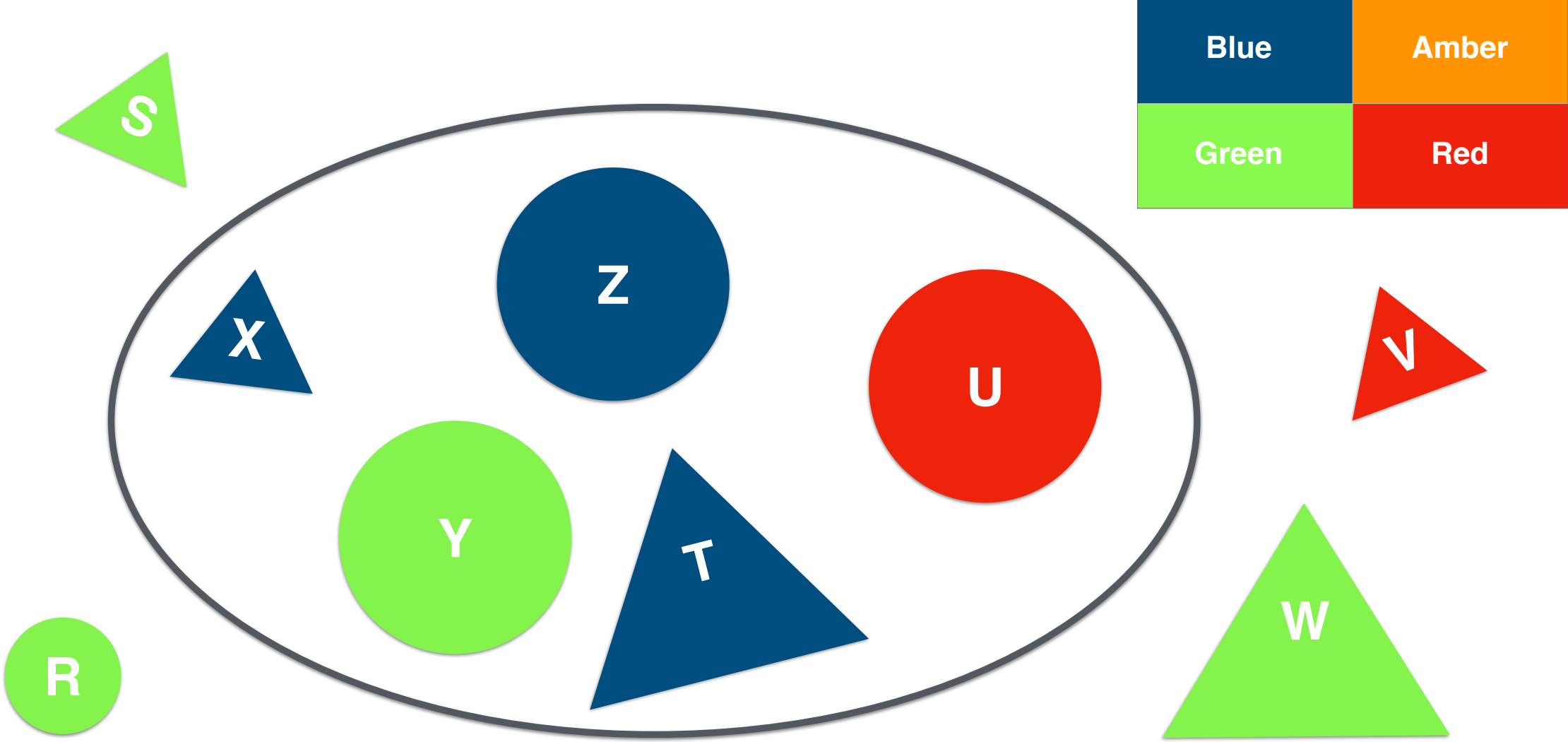
```
data Thing = R | S | T | U | V | W | X | Y | Z

isRed U = True
isRed V = True
isRed _ = False

isBlue T = True
isBlue X = True
isBlue Z = True
isBlue _ = False

isGreen x = not (isRed x || isBlue x)
```



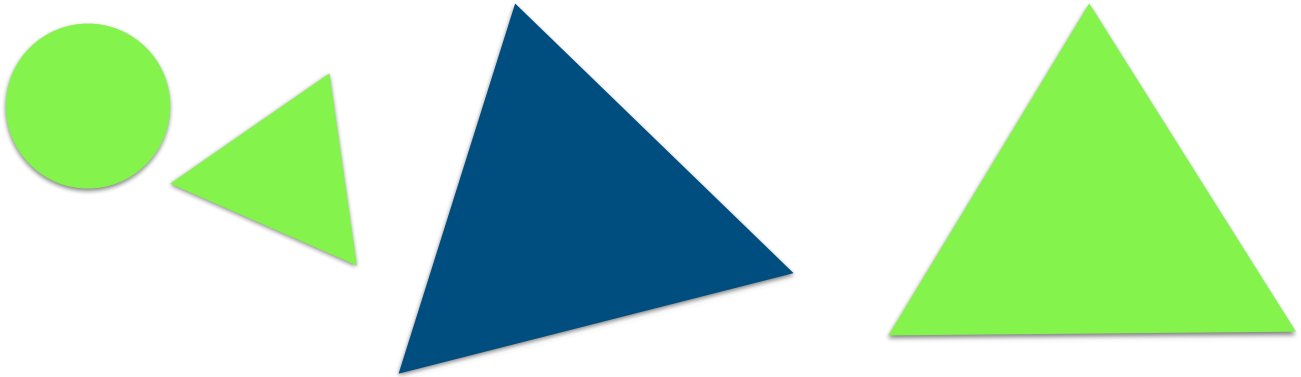
```
data Thing = R | S | T | U | V | W | X | Y | Z

isSmall R = True
isSmall S = True
isSmall V = True
isSmall X = True
isSmall _ = False

isBig = not . isSmall
```



Blue	Amber
Green	Red




```
[ x | x <- things, isBlue x || (isBig x && isDisc x) ]
```



Blue	Amber
Green	Red

```
and [ isSmall x | x <- things, isRed x, isTriangle x ]
```

some statements:

every red triangle is small ✔

every small triangle is red ✘

some big triangle is green ?

some small disc is red ?

no red thing is blue ?