

Boolean Logic

Boolean values

We need to define:

- True
- False

Intuition:

Boolean values are 2-arity functions (two parameters).

True returns the first parameter.

False returns the second parameter.

$$\text{True} = \lambda x.\lambda y. x$$
$$\text{False} = \lambda x.\lambda y. y$$

*These are not declarations of names.
We simply denote True and False as
the short-hands of the LC expressions
on the right-hand side.*

Boolean Operators: AND

AND = $x \cdot y$. _____

where x, y are boolean values.

Such that

- (**AND** True True) \rightarrow True
- (**AND** True False) \rightarrow False
- (**AND** False True) \rightarrow False
- (**AND** False False) \rightarrow False

AND = $x \cdot y$. ($x \cdot y$ **False**)

Can you verify that this works as required?

Other operators

We need to also define:

OR = $\backslash x.\backslash y.$ _____

and

NOT = $\backslash x.$ _____

Here, the parameters x, y are boolean values.

OR = $\backslash x.\backslash y. (x \text{ True } y)$

NOT = $\backslash x. (x \text{ False True})$

Can you verify that this works as required?

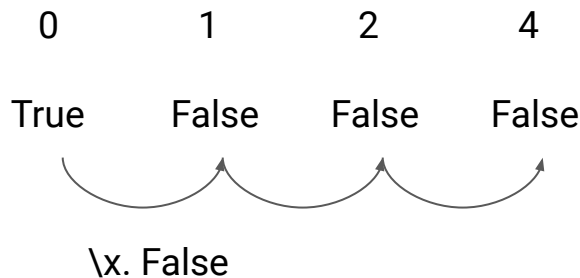
IsZero

Consider a function that takes a Church number as argument, and returns a boolean value.

$\text{IsZero } 0 = \text{True}$

$\text{IsZero } x = \text{False}$ *for all other* $x \neq 0$

Design an iterative definition of IsZero



$\text{IsZero}(n) = n (\x. \text{False}) \text{True}$

$\text{IsZero} = \lambda n. n (\lambda x. \text{False}) \text{True}$

If-Else

Consider the if-then-else construct:

```
If (u: Boolean) Then  
    m: Number  
Else  
    n: Number
```

Can we define a 3-arity function:

IfElse = \u.\m.\n. _____

Challenge:

Can you complete the LC expression for

IfElse?