Logic

February 28, 2021

0.1 Notes

1. The notebooks are largely self-contained, i.e, if you see a symbol there will be an explanation about it at some point in the notebook.
   - Most often there will be links to the cell where the symbols are explained
   - If the symbols are not explained in this notebook, a reference to the appropriate notebook will be provided
2. **Github does a poor job of rendering this notebook**. The online render of this notebook is missing links, symbols, and notations are badly formatted. It is advised that you clone a local copy (or download the notebook) and open it locally.

1 Contents

1.1 Importing Libraries

[1]:
```
import random
import math
```

Boolean Operations

And

The Boolean And operation is denoted using $\wedge$

$$p \wedge q$$

[2]:
```python
p = bool()
q = bool()

p and q
```

[2]: False

Or

The Boolean Or operation is denoted using $\vee$

$$p \vee q$$

[3]:
```python
p = bool()
q = bool()

p or q
```

[3]: False

Not

The Boolean Not operation is denoted using $\sim$ or $\neg$ and sometimes just the text 'not' is used:

$$\sim p$$

or

$$\neg p$$

or

$$\text{not } p$$

[4]:
```python
p = bool()

not p
```

[4]: True

Exclusive Or

The Exclusive Or operation is denoted using $\veebar$ or $\oplus$ or just XOR

$$p \veebar q$$

or

$$p \oplus q$$

or

$$p \text{ XOR q}$$

[5]:
```
p = bool()
q = bool()

#Shorter code since bool Xor equivalent to bitwise Xor
XOR_short = p ^ q

#Longer code by definition
XOR_long = (not q and p) or (not p and q)

XOR_short, XOR_long
```

[5]:
```
(False, False)
```

Nand

The Nand operation is denoted using $\barwedge$

$$p \barwedge q$$

The Nand operator can be expanded to: $p \barwedge q = \neg(p \vee q)$

[6]:
```
p = bool()
q = bool()

not (p and q)
```

[6]:
```
True
```

Proof Symbols

Implies

In mathematical proofs, the term 'implies' means 'if a then b' or 'a implies b' and this is denoted using the symbol: $\Rightarrow$

$$a \Rightarrow b$$

Here a and b are any mathematical concepts (or *logical predicates*)

[7]:
```python
boo = [True, False]

for a in boo:
    for b in boo:
        print('a: ', a, 'b: ',b,', a implies b :', b or (not a))
```

```
a:  True b:  True , a implies b : True
a:  True b:  False , a implies b : False
a:  False b:  True , a implies b : True
a:  False b:  False , a implies b : True
```

To prove a theorem of this form, you must prove that b true whenever a is true. Example: x is greater than or equal to 4) then $2^x \geq x^2$

$$x \geq 4 \Rightarrow 2^x \geq x^2, \ \forall x \in \mathbb{R}$$

i.e: if ($a = x$ is greater than or equal to 4), then ($b = 2^x \geq x^2$)

For more information on the real set and the belongs to notation see the Collections notebook

[8]:
```python
print('if a then b')

X   = [-10.00,-2.20,0.00,2.00,3.10,4.00,5.5,6.00,7.8]

for x in X:
    condition_a = x >= 4
    condition_b = 2**x >= x**2
    print('x :', x, ', x>=4, a: ',condition_a,', 2^x>=x^2, b: ', condition_b)
```

```
if a then b
x : -10.0 , x>=4, a:  False , 2^x>=x^2, b:  False
x : -2.2 , x>=4, a:  False , 2^x>=x^2, b:  False
x : 0.0 , x>=4, a:  False , 2^x>=x^2, b:  True
x : 2.0 , x>=4, a:  False , 2^x>=x^2, b:  True
x : 3.1 , x>=4, a:  False , 2^x>=x^2, b:  False
x : 4.0 , x>=4, a:  True , 2^x>=x^2, b:  True
x : 5.5 , x>=4, a:  True , 2^x>=x^2, b:  True
x : 6.0 , x>=4, a:  True , 2^x>=x^2, b:  True
x : 7.8 , x>=4, a:  True , 2^x>=x^2, b:  True
```

Implied by

In mathematical proofs, the term 'implied by' means 'if b then a' or 'a is implied by b' and this is denoted using the symbol: $\Leftarrow$

$$a \Leftarrow b$$

Here a and b are any mathematical concepts (or *logical predicates*)

[9]:
```
boo = [True, False]

for a in boo:
    for b in boo:
        print('a: ', a, 'b: ',b,', a implied by b :', a or (not b))
```

```
a:  True b:  True , a implied by b : True
a:  True b:  False , a implied by b : True
a:  False b:  True , a implied by b : False
a:  False b:  False , a implied by b : True
```

To prove a theorem of this form, you must prove that a true whenever b is true.

If and only if

In mathematical proofs, the term 'if and only if' means 'if a then b' as well as 'if b then a' and this is denoted using the symbol: $\iff$ but it is also sometimes abbreviated as: iff

$$a \iff b$$

or

$$a \text{ iff } b$$

The concept of iff is also logically equivalent to $(a \Rightarrow b) \land (b \Rightarrow a)$

Here a and b are any mathematical concepts (or *logical predicates*).

[10]:
```
boo = [True, False]

for a in boo:
    for b in boo:
        print('a: ', a, 'b: ',b,', a iff b :', (b or (not a)) and (a or (not␣
 ↪b)))
```

```
a:  True b:  True , a iff b : True
a:  True b:  False , a iff b : False
a:  False b:  True , a iff b : False
a:  False b:  False , a iff b : True
```

To prove a theorem of this form, you must prove that a and b are equivalent. Not only is b true whenever a is true, but a is true whenever b is true. Example: The integer n is odd if and only if n^2 is odd.

$$n \text{ is odd } \iff n^2 \text{ is odd } \forall n \in \mathbb{Z}$$

i.e: $(a = n \text{ is odd})$ iff $(b = n^2 \text{ is odd})$

For more information on the integer set and the belongs to notation see the Collections notebook

[11]:
```python
def is_odd(x):
    return (x%2 == 0)

print('if a then b')

#Check if a then b
N = [random.randint(-100,100) for i in range(5)]
for n in N:
    print('n: ', n, ', odd(n), a:',is_odd(n),
          '\nn2: ',n**2,', odd(n2), b:', is_odd(n**2),'\n')

print('if b then a')
#Check if b then a:
N_new = [random.randint(-100,100)**2 for i in range(5)]
for n_squared in N_new:
    n = int(math.sqrt(n_squared))
    print('n2: ',n_squared,', odd(n2), a:', is_odd(n_squared),
          '\nn: ', n, ', odd(n): ',all([is_odd(n),is_odd(-n)]), '\n')
```

```
if a then b
n:  69 , odd(n), a: False
n2:  4761 , odd(n2), b: False

n:  -80 , odd(n), a: True
n2:  6400 , odd(n2), b: True

n:  33 , odd(n), a: False
n2:  1089 , odd(n2), b: False

n:  26 , odd(n), a: True
n2:  676 , odd(n2), b: True

n:  -68 , odd(n), a: True
n2:  4624 , odd(n2), b: True

if b then a
n2:  2809 , odd(n2), a: False
n:  53 , odd(n):  False

n2:  4761 , odd(n2), a: False
n:  69 , odd(n):  False

n2:  1936 , odd(n2), a: True
n:  44 , odd(n):  True

n2:  576 , odd(n2), a: True
n:  24 , odd(n):  True
```

```
n2:  1521 , odd(n2), a: False
n:  39 , odd(n):  False
```

Quantifiers

For all

Also called a universal quantifier. The 'for all' symbol is used simply to denote that a concept or relation is applied to every member of the domain. Denoted by $\forall$

Squares of all real numbers are positive or zero can be expressed through:

$$\forall x \in \mathbb{R}, x^2 \geq 0$$

Which can be read as, for all x belonging to the set of real numbers, the square of x is always greater or equal to zero.

[12]:
```python
trials = 5

for i in range(trials):
    x = random.uniform(-100000, 100000)**2
    print(x >= 0)
```

```
True
True
True
True
True
```