CS130 - Propositional Logic Summary

- ↳ Propositions express deliberative statements that assert something
    - ↳ Can be true or false, but not both.
    - ↳ Can abstract their content away by assigning them to variables
    - ↳ Can use boolean operators on them.
    - ↳ Defining them doesn't tell you anything about their truth value.

- ↳ Functions
    - ↳ unambiguously map values from a domain to a codomain ($f: X \rightarrow Y$) - discussed in foundations summary
- ↳ Boolean functions
    - ↳ Take a tuple of boolean values and map them to a single one:

$$B = \{T, F\} \quad , \quad f: B^n \rightarrow B$$

- ↳ We can think of operations like AND or XOR, etc. as boolean functions.

- ↳ Material implication
    - ↳ Encodes an 'if... then...' statement.
    - ↳ Denoted by $x \rightarrow y$ (read as if x, then y)
    - ↳ Tells you nothing about the truthiness of either variable, just the relationship between them.

x	y	$x \rightarrow y$
F	F	T
F	T	T
T	F	F
T	T	T

↳ Always true, unless $x = T$ & $y = F$ (if $x = F$, tells you nothing about y)

$$x \rightarrow y = \cancel{\neg P \vee Q}$$
$$\neg x \vee y$$

↳ Logical equivalence / tautology

↳ ~~Denotes log $x \leftrightarrow y$~~ Propositions x & y are logically equivalent if $x \leftrightarrow y$ is a tautology

x	y	$x \leftrightarrow y$
F	F	T
F	T	F
T	F	F
T	T	T

A tautology is when a proposition is true regardless of its atomic propositions.

↳ $x \equiv y$ denotes logical equivalence

↳ '$\equiv$' is not an operator, but '$\leftrightarrow$' is.

↳ There are many laws for 'algebra' with propositional logic, see the cheat sheet.

CS130 - Logic of Predicates Summary ①.

- ↳ Predicates are statements about a variable.
  - ↳ can be true or false, but not both - dependent on the value of its variable(s).
  - ↳ the variables have to be drawn from a domain, e.g. $\mathbb{Z}$
  - ↳ the variables can be 'bound' with quantifiers
    - $\forall u$, read for all (universal)
    - $\exists u$, read there exists (existential)
    - (- $\exists! u$, read there exist only one (unique))

Non-bound variables called 'free'

Allow expression in a concise way

- ↳ often of the form:

$$\underbrace{\exists x}_{\text{Quantifier}} \in \underbrace{S}_{\text{Domain}} \; \underbrace{.}_{\text{'in'}} \; \underbrace{P(x)}_{\text{Predicate}}$$

CS130 - Logic Summary ①.

- ↳ General / abstract topic which can be built upon to solve many problems
- ↳ can be used to manipulate / encode statements, e.g. boolean functions, predicates, or even time dependent events