I Logic: Propositions and Operations

Def: A proposition is a statement that can be unambiguously evaluated as true or false.

Ex: p = "There is life on Mars" ← unambiguously false

q = "My name is Annalise" ← unambiguously true

Non-Ex: r = "What time is it now?"

s = "More people have been in Moscow than I have"

· grammatically correct but non-sensical

t = "Ttsssppp"

$(a+b)^2 = a^2 + 2ab + b^2$ is a proposition in algebra

p = "Don't do it" → not a proposition bc it can't be assigned a truth value

p_2 = "$x+1=2$" → not a proposition bc we don't know what x is

p_3 = "A square has 4 sides" → true proposition

p_4 = "$x^2 \geq 0$" → not a proposition bc x is not necessarily a real number

1.1 Negation:

Def let p be a proposition. Then $-p$ or $\bar{p}$ is the proposition, called negation of p,

$-p$ = "It is not the case that p"

Ex let p = "Jane is healthy"

$-p$ = "It is not the case that Jane is healthy" or "Jane is sick"

Ex let p = "I got the highest score on the exam"

$-p$ = "I didn't the highest score on the exam"

Truth Tables

P	-P
F	T
T	F

1.2 Logical AND, or Conjunction

Def Let p and q be two propositions. Then the conjunction of p and q is the proposition $p \wedge q$, given by

P	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

Ex Let p = "I am vegetarian" (true)

Let q = "Sarah likes apples" (false)

(false) $p \wedge q$ = "I am vegetarian and Sarah likes apple"

I Logic: Propositions and Operations

1.3 Logical OR, or disjunction

Def let p and q be two propositions. Then the disjunction of p and q is the proposition $p \vee q$ is given by

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

Ex p = "I want to visit France" (t)

q = "I want to visit Spain" (t)

$p \vee q$ = "I want to visit France or Spain" (t)

1.4 Exclusive OR, or XOR

Truth table $p \oplus q$

p	q	$p \oplus q$
F	F	F
T	F	T
F	T	T
T	T	F

Ex p = "You can take a cake"

q = "You can take a muffin"

$p \oplus q$ = "You can take a cake or a muffin, but not both."

✱ specify by using "or both" or "but not both" ✱

1.5 Conditional Statement or implication

Def Let p, q be two propositions. Then the conditional statement $p \to q$ is the proposition given by

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

Ex Let p = "I am the president"

Let q = "I will lower taxes"

$p \to q$ = "If I am the president, I will lower taxes."

p	q	$p \to q$	
T	T	T	president fulfilled the promise
T	F	F	president lied
F	T	T	president did not break promise
F	F	T	president did not break promise

Ex Let p = "We live on mars" (f)

let q = "$2+2=10119$" (f)

$p \to q$ = "If we live on Mars, then "$2+2=10119$" = true

✱ if you assume nonsense, any statement can be true ✱

$p \to q$: • If p then q • p is sufficient for q • q provided that p

(I) Logic: Propositions and Operations

In $p \to q$

- p is called <u>condition</u> or <u>premise</u>
- q is called <u>conclusion</u> or <u>consequence</u>

<u>Def</u> Let $p \to q$ be a conditional statement

Then
- $q \to p$ is called <u>converse</u>
- $\neg q \to \neg p$ is called <u>contrapositive</u>
- $\neg p \to \neg q$ is called <u>inverse</u>

1.6 Biconditional Statements

<u>Def</u> let p an q be two propositions. Then the compound proposition $p \leftrightarrow q$ is called <u>biconditional</u> or <u>equivalence</u> and given by the table

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

<u>Ex</u> p = "You can get a flight"

q = "You have a ticket"

$p \leftrightarrow q$ = "You can fly if and only if you have a ticket"

p is equivalent to q <u>or</u> p is exactly when q

"Sarah will go for a run unless weather is bad" (T)
- if weather is good, Sarah will run and if it is bad, she will not.

p = "Sarah runs" q = "weather is good" $p \leftrightarrow q$

"You can't win the lottery unless you buy a ticket."
- if you don't buy you can't win and if you do you can

p = "you can't win a lottery" q = "you don't buy a ticket" $q \to p$ $\neg L \to H$

"A lunka is healthy unless it has a lympcka" (T) H L
- if lunka is healthy it doesn't have lympcka
- if lunka is sick it has a lympcka
- if lunka has lympcka it is sick
- if it doesn't have lympcka it is healthy

hardest question on midterm 1

Translation between logic and language
- p unless q means $\neg q \to p$
- p only if q means $p \to q$
- p is sufficient for q means $p \to q$
- p is necessary for q means $q \to p$
- p is necessary and sufficient for q means $p \leftrightarrow q$

I Logic: Predicates and Quantifiers

Ex Domain: all people, $P(x)$ = "x is a chess player."

$\exists x P(x)$ = "there exists person x such that x is a chess player"
= "There exists a chess player" (true)

Ex Domain: all real numbers, $P(x) = "x > 3"$

$\exists x P(x)$ = "there exists number x s.t. $x > 3$"

Ex Domain: all people on Mars, $P(x)$ = "x has a leg"

$\exists x P(x)$ = "There exists a person on Mars with a leg." (false) (no ex.)

$\forall x P(x)$ = "Every person on Mars has a leg." (true) (no counter-ex.)

- to show that $\exists x P(x)$ is true is usually easy b/c it is enough to provide an example.

Quantifiers over Finite Domains

Domain: Jessica, Mike, Bob, Alice $P(x)$ - predicate

- $\forall x P(x)$ is the same as $P(\text{Jessica}) \land P(\text{Mike}) \land P(\text{Bob}) \land P(\text{Alice})$
- $\exists x P(x)$ is the same as $P(\text{Jessica}) \lor P(\text{Mike}) \lor P(\text{Bob}) \lor P(\text{Alice})$

But if domain is infinite, you can't do this.

Common Mistake $P(x) = "x^3 > 0"$ *must always specify domain*

$\forall x P(x)$ Domain: all pos. real num $\forall x P(x)$ - true

$\exists x P(x)$ $\exists x P(x)$ - true

Domain: all reals $\forall x P(x)$ - false $\exists x P(x)$ - true

Domain: all neg reals $\forall x P(x)$ - false $\exists x P(x)$ - false

Domain: empty $\forall x P(x)$ - true $\exists x P(x)$ - false

Logical Equivalence

Def Two expressions with predicates and quantifiers are logically equivalent if they are equivalent as propositions for all choices of domains, predicates, and specified variables.

Ex $(\forall x P(x)) \land (\forall y Q(y)) \equiv \forall x \forall y (P(x) \land P(y))$

Notation is often simplified

Ex Domain: all reals
$\forall x (x^2 \geq 0)$

Ex $\forall x ((x < 0) \rightarrow (x^3 < 0))$, where x is a real number.

Ex $\exists x ((x^2 = 2) \land (x > 0))$, where x is a real number.

Ex $\forall x < 0 \, (x^3 < 0)$ (Domain: all reals) Ex $\exists x > 0 \, (x^2 = 2)$ (Domain: all reals)

Ⓘ Logic: Predicates and Quantifiers

Logical Equivalence

Ex $\forall x[P(x) \wedge Q(x)] \equiv [\forall x P(x)] \wedge [\forall x Q(x)]$

De Morgan Laws for Quantifiers

Ex Domain: all students $P(x) =$ "x took calculus"

$\forall x P(x) =$ "All students took calculus"

$\neg[\forall x P(x)] =$ "Not all students took calculus."

= "There is a student who didn't take calculus"

$= \exists x[\neg P(x)]$.

Ex Domain: all people $P(x) =$ "x is an honest politician"

$\exists x P(x)$: "There exists an honest politician."

$\neg[\exists x P(x)]$: "Every politician is dishonest." $= \forall x[\neg P(x)]$

Def
$$\boxed{\begin{aligned} \neg[\forall x P(x)] &\equiv \exists x[\neg P(x)] \\ \neg[\exists x P(x)] &\equiv \forall x[\neg P(x)] \end{aligned}}$$

Ex Domain: rational numbers $(5/7, 1/2, -3/6)$ $\exists x(x^2 = 2)$

$\neg[\exists x(x^2=2)] \equiv \forall x[x^2 \neq 2]$.

Ex $\forall x(x^2 \geq x)$, where x is a real number $\boxed{\neg(p \to q) \equiv p \wedge (\neg q)}$

$\neg[\forall x(x^2 \geq x)] \equiv \exists x[(x^2 < x)]$.

Ex $\neg(\forall x[P(x) \to Q(x)]) \equiv \exists x[\neg(P(x) \to Q(x))]$

$\equiv \exists x[P(x) \wedge (\neg Q(x))]$

Ex · Domain: all students $C(x) =$ "x took calculus" $\forall x S(x) \to \forall x C(x)$

$\forall x C(x) =$ "All students took calculus."

· Domain: all people $S(x) =$ "x is a student"

$\forall x[S(x) \to C(x)]$. correct. $\forall x[S(x) \wedge C(x)]$. wrong.

Ex · Domain: all students $R(x) =$ "x visited Russia"

$\exists x R(x) =$ "There is a student who visited Russia"

· Domain: all people $S(x) =$ "x is a student"

$\exists x[S(x) \wedge R(x)]$ correct. $\exists x[S(x) \to R(x)]$ wrong. *

Remark about variable names

Ex Domain: all cities

$P(x) =$ "x is in the U.S."

$\exists s P(s)$

Ex $P(x,y) =$ "x is a city in y." $\exists z \exists w P(z,w)$ (can call variables anything)