Discrete Exam 1 Cheat Sheet

by TheEmu001 via cheatography.com/30732/cs/9211/

Useful Symbols

∀	for all (universal operator)
∃	exists (existential operator)
∊	in the set
∧	and
∨	or
~	not
≡	equivalent
⊂	subset
⊃	superset
{}, ∅	empty set
↔	biconditional (both are true)

Statements

Universal	For all, for each
Existential	At least, there exists
Conditional	If → then
Universal Conditional	For all & if-then
Universal Existential	For all & there exists
Existential Universal	There exists & for all

List of Equivalences

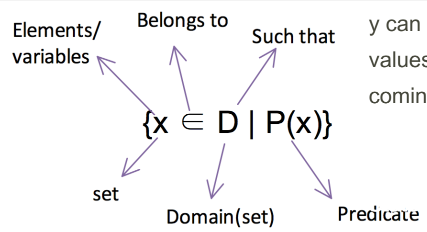
Conditional Statemnts	p→q≡(p∧q)	p→q≡~p∨q
contra-positive	p→q≡q→p	
Converse	p→q (cond)	q→p (converse)

List of Equivalences (cont)

inverse	p→q (cond)	p→p (inverse)

vacuously true = true by absence
converse and inverse are the **SAME**

Set-Builder Notation



Set-Roster Notation

A = {1, 2, 3 ... 100}

use ellipses for larger sets

Subsets

B⊆A	B=subset, A=superset

Proper Subsets: elements that belong to superset but NOT subset

Relations

Relations=	subsets of cartesian product
R ⊆ A x B	Relation ⊆ Domain x Codomain
Domain	**SET** that includes every element from source

don't always have to include ordered pairs

Functions

Requirements:
- Arrow coming out of **every element** in domain
- Every element can only have **one** element of *domain* connected to one element of *codomain*

unsatisfied requirement = relation
y can be used repeatedly but x values only have one arrow coming out

Predicates and Quantified Statements

Statement type	original	negated
Universal	∀x∊D, P(x)	∃x∊D, ~P(x)
Existential		
Universal Conditional		

DeMorgan's Law

• Tells us how to handle conjunction and disjunction negations
(p∧q) ≡ p∨~q
(p∨q) ≡ p∧~q
"The connector is loose(l) or the machine is unplugged(u)"
l ∨ u -- negation --> (l ∨ u) ≡ l ∧~u
"The connector is **not** loose *and* the machine is **not** unplugged"
p∨q is the opposite of p∧q

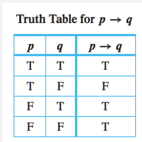
When using DeMorgan's law, no need for truth table

Tautologies and Contradictions

Tautologies	Always true statements	t
Contradictions	Always false statements	c
p∧~p≡c	T∧F≡c	F∧T≡F
p∨t≡t	p∧c≡c	

Absorption law: variable absorbing operator
⇒use truth table to prove law
⇒other variables don't play a role in statement validity
p∨(p∧q)≡ p; p∧(p∨q)≡ p

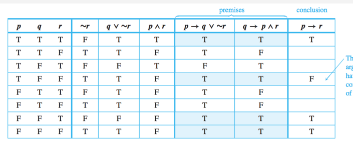
p→q truth table



Truth Table for p → q

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Argument Truth Table



p	q	r	~r	q ∨ ~r	p ∧ r	p → q ∨ ~r (premises)	q → p ∧ r (premises)	p → r (conclusion)
T	T	T	F	T	T	T	T	T
T	T	F	T	T	F	T	F	
T	F	T	F	F	T	F	T	
T	F	F	T	T	F	T	T	F
F	T	T	F	T	F	T	F	
F	T	F	T	T	F	T	F	
F	F	T	F	F	F	T	T	T
F	F	F	T	T	F	T	T	T

Critical row = row where both premises are true
premises and conclusion = TRUE is a **valid** argument

Arguments

p→q	major premise
p	minor premise
∴q	therefore, conclusion

premises aka assumptions or hypotheses
verified using truth table

Argument Forms (VALID)

Modus Pones	p→q
	p
	∴q
Modus Tollens	p→q
	~q
	∴~p
Gneralization	p
	∴pvq
Specialization	p∧q
	∴q
Elimination	pvq
	~q
	∴p
Transitivity	p→q
	q→r
	∴p→r
Proof by div. into cases	pvq
	p→r
	q→r
	∴r

Fallacy (INVALID ARGUMENTS)

Converse Error	p→q
	q
⇒	∴p
Inverse Error	q→p
	~p
	∴~q