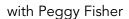
Foundations of Programming: Discrete Mathematics





Set Theory

a∈A	a is an element of set A
a∉A	a is not an element of set A
{a ₁ ,a ₂ ,,a _n }	Set with elements a ₁ a _n
$\{x \in D \mid P(x)\}$	The set of all x in D for which $P(x)$ is true
R	All real numbers
Z	All integer numbers
Q	All rational numbers
W	Whole numbers (counting numbers plus zero)
N	Natural numbers (counting numbers 1,2,3,4,5)
A⊆B	A is a subset of B
A⊈B	A is NOT a subset of B
A = B	A equals B
AUB	A union B
AΛB	A intersects B
В-А	Elements in B but not in A
A ^c or \overline{A}	Complement of A
(x,y)	Ordered pair
AxB	Cartesian product
Ø, { }	Empty set
℘ (A)	Power set of A

Logical Equivalences

t = tautology \sim = NOT Λ = AND

c = contradiction V = OR $\equiv = logically equivalent$

Name	Form 1	Form 2
Commutative Laws	p~ ≡ qVp	pvq ≡ qvp
Associative Laws	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	$(pVq)Vr \equiv pV(qVr)$
Distributive Laws	$p\Lambda(qVr) \equiv (p\Lambda q)V(p\Lambda r)$	$pV(q\Lambda r) \equiv (pVq)\Lambda(pVr)$
Identity Laws	p∧t ≡ p	pvc ≡ p
Negation Laws	pV~p ≡ t	p∧~p ≡ c
Double Negation Laws	~~p ≡ p	_
Idempotent	p∧p ≡ p	pvp ≡ p
Universal Bounds	pVt ≡ t	p∧c ≡ c
De Morgan's Laws	~(p∧q) ≡ ~p∨~q	~(pVq) ≡ ~p∧~q
Absorption Laws	$pV(p\Lambda q) \equiv p$	$p\Lambda(pVq) \equiv p$
Negations of t and c	∿t = c	~c = t

Rules of Inference with Propositions

p p → q	Modus ponens	р ф ∴ р∧ф	Conjunction
∴ q ————		p → q q → r	Hypothetical syllogism/transitivity
¬ q p → q	Modus tollens	• p→r	
		a) pVq	Disjunctive syllogism/elimination
a) p pVq b) q ∴ pVq	Addition/generalization	∴ p b) pvq ¬ p ∴ q	
a) pVq ∴ p b) p∧q ∴ q	Simplification/specialization	pVq ¬pVr ∴qVr	Resolution
		pVq p→r q→ p ∴ r	Proof by division into cases