

Course notes for csc 165 H:
Mathematical Expression and Reasoning
for Computer Science

Gary Baumgartner and Danny Heap and Richard Krueger and François Pitt

Fall 2011

CONTENTS

1	INTRODUCTION	5
1.1	What's CSC 165H about?	5
1.2	Human versus technical communication	7
1.3	Problem-solving	8
1.4	Inspirational puzzles	9
1.5	Some mathematical prerequisites	10
2	QUANTIFICATION, IMPLICATION, AND SYMBOLS	15
2.1	Universal quantification	15
2.2	Existential quantification	16
2.3	Properties, sets, and quantification	17
2.4	Sentences, statements, and predicates	18
2.5	Implications	19
2.6	More symbols	20
2.7	Implication in everyday English	21
3	LOGICAL CONNECTIVES	23
3.1	Universal quantification and implication again	23
3.2	Vacuous truth	24
3.3	Equivalence	24
3.4	Restricting domains	25
3.5	Conjunction (And)	25
3.6	Disjunction (Or)	25
3.7	Negation	26
3.8	Symbolic grammar	27
3.9	Truth tables	27
3.10	Tautology, satisfiability, unsatisfiability	28
3.11	Logical "arithmetic"	28
3.12	DeMorgan's Laws	29
3.13	Implication, bi-implication, with \neg , \vee , and \wedge	29
3.14	Transitivity of universally-quantified implication	29
3.15	Summary of manipulation rules	30
3.16	Multiple quantifiers	30
3.17	Mixed quantifiers	31
4	PROOFS	33
4.1	What is a proof?	33
4.2	Direct proof of universally-quantified implication	34
4.3	Hunting the elusive direct proof	34

4.4	An odd example of direct proof	35
4.5	Another example of direct proof	36
4.6	Direct proof of universally-quantified predicate	37
4.7	Indirect proof of universally-quantified implication	37
4.8	Proof by contradiction	37
4.9	Direct proof structure of the existential	38
4.10	Multiple quantifiers, implications, and conjunctions	38
4.11	Example of proving a statement about a sequence	39
4.12	Example of disproving a statement about a sequence	40
4.13	Non-boolean function example	41
4.14	Substituting known results	41
4.15	Proof by cases	42
4.16	Proving \vee using cases	43
4.17	Building formulae and taking formulae apart	44
4.18	Summary of inference rules	46
5	ALGORITHM ANALYSIS AND ASYMPTOTIC NOTATION	49
5.1	Correctness, running time of programs	49
5.2	Binary (base 2) notation	49
5.3	\log_2	50
5.4	Loop invariant for base 2 multiplication	50
5.5	Running time of programs	52
5.6	Linear search	52
5.7	Run time and constant factors	53
5.8	Asymptotic notation: Making Big-O precise	54
5.9	Insertion sort example	61
6	A TASTE OF COMPUTABILITY THEORY	65
6.1	The problem	65
6.2	An impossible proof	65
6.3	Countability	66
6.4	Diagonalization	69
6.5	Reductions	72