Math 310
Final Exam Review Sheet

Part I: Logic

• Understand propositions and logical operators, and be able to build their truth tables.
• Be able to translate back and forth between English statements and symbolic logical propositions.
• Be able to prove the logical equivalence of a pair of propositions by building truth tables or by using a 2-column proof.
• Be able to negate propositions using De Morgan's Laws and other equivalences.
• Understand predicates and quantifiers and be able to determine the truth value of statements involving predicates and/or quantifiers. Also be able to translate English statements into statements involving one or more predicates and quantifiers.
• Be able to negate statements involving predicates and quantifiers and be able to recognize free and bound variables.
• Know the definition of an argument and how to use either a truth table or a 2-column proof to determine whether a given argument form is valid.
• Know the standard rules of inference for arguments with statements involving propositions and for arguments with statements involving predicates and quantifiers.
• Be able to recognize from the form of a short argument whether it is a known valid argument form or a known fallacy.

Part II: Proof Methods

• Understand the basic terminology for paragraph proofs.
• Understand how to use a *counterexample* to show that a statement is false.
• Be able to write short proofs making use of: Direct proof, Contraposition, and Proof by Contradiction.
• Understand the definition of odd numbers, even numbers, and rational numbers.
• Understand and be able to utilize Proof by Cases and be able to rule out unnecessary cases.
• Understand and be able to carry out an existence proofs and uniqueness proofs.
• Understand the role of counterexamples and conjectures in proving and disproving statements.
• Understand how to prove results using "backwards reasoning"
• Understand the principle of mathematical induction and be able to prove theorems using mathematical induction and using strong induction.

Part III: Sets

• Understand the definitions of sets, elements, subsets, and proper subsets, as well as symbolic notation for these terms.
• Understand both roster and set builder notation, and be able to determine whether a given set is well defined.
• Understand the definition of set equality, the cardinality of a set, the universal set, the empty set, singleton sets, the power set of a set, and know how to form the Cartesian Product of two sets.
• Understand the definition of the basic set operations.
• Understand what it means for two sets to be disjoint and know how to compute the cardinality of the union of two sets using the principle of inclusion-exclusion.
• Be able to determine which elements are in a set resulting from multiple set operations. Also be able to draw a Venn diagram representing such a set.
• Know the basic set identities (set equalities) and be able to prove them using membership tables, two-column proofs, or paragraph (double containment) proofs.

Part IV: Binary Relations

• Know the definition of an binary relation and how it is represented as a set of ordered pairs.
• Be able to determine whether a given relation on a set is reflexive, irreflexive, symmetric, antisymmetric, or transitive.
• Be able to combine relations to form new ones using union, intersection, complementation, difference, symmetric difference, and composition.
• Understand how to represent a relations using matrices and graphs.
• Be able to recognize whether a relation on a set is reflexive, irreflexive, symmetric, antisymmetric, or transitive by looking at its matrix or its directed graph.
• Be able to apply the operations union, intersection, complementation, difference, symmetric difference, and composition to matrices of relations and to directed graphs of relations.
• Know the definition of an equivalence relation and be able to prove whether a given relation is an equivalence relation. Also be able to find the equivalence class of a given element.
• Understand the relationship between equivalence classes and partitions of the underlying set. Be also to tell whether or not a given family of subsets is a partition, and if so, be able find the equivalence relation described by the partition.
• Know the definition of a partial order and be able to prove whether a given relation is a partial order.
• Understand definition of a total ordered set and a well ordered set. Also know the lexicographic order, and be able to order a given pair of elements with respect to this ordering.

• Be able to draw the Hasse diagram for a poset. Also understand the definition of maximal elements, minimal elements, greatest elements, least elements, upper bounds, lower bounds, least upper bounds, greatest lower bounds, and the definition of a lattice. Also be able to find these (when they exist) in a given poset.

Part V: Graphs and Trees

• Understand the definitions of the different types of graph models and know how various types of graphs can be used to model real world applications.
• Understand the basic graph terminology and main results like the Handshaking Theorem.
• Know the main classes of simple graphs like K_n, C_n, W_n, Q_n, and $K_{m,n}$.
• Understand the definition of a bipartite graph and be able to prove whether or not a given graph is bipartite. Also be able to use bipartite graphs to solve matching problems.
• Understand how to form subgraphs of a graph, the union of two graphs, and how to find the complement of a simple graph.
• Understand how to represent both directed and undirected graphs using Adjacency Lists, Adjacency Matrices, and Incidence Matrices. Also know the definitions of a graph isomorphism and a graph invariant.
• Given a pair of graphs, be able to prove whether or not the given pair are isomorphic.
• Know the definitions of paths and circuits.
• Understand what it means for a graph to be connected, strongly connected, and weakly connected. Also be able to find the connected components (or strongly connected components) of a given graph.
• Understand the path length is a graph invariant, and be able to use this fact in graph isomorphism proofs.
• Understand the definition of Euler Paths, Euler Circuits, Hamilton Paths, and Hamilton Circuits. Also understand how these are used in graph models.
• Given a specific (di)graph, be able to determine whether or not the graph has an Euler Path, an Euler Circuit, a Hamilton Path or a Hamilton Circuit by using key theorems. Also be able to find and label them in a graph when they exist.
• Understand the definition of and be able to identify cut vertices and bridge edges in a given graph.
• Understand Dijkstra's algorithm and be able to use it to find the shortest path between a pair of vertices in a weighted graph.
• Understand the definition of a planar representation of a graph and be able to find a planar representation for a graph (when one exists).
• Memorize Euler's Formula and the corollaries to Euler's Formula and be able to use them to show that a graph is or is not planar. Also Know and be able to apply Kuratowski's Theorem.
• Be able to find the vertex or edge chromatic number of a given graph.
• Be able to use graph vertex colorings to solve applications problems (like scheduling and radio frequency assignments).
• Understand the definition of a tree, a forest, a simple circuit. Also know the main theorems about both trees and rooted trees.
• Understand the definition of a binary search tree and be able to construct a binary search tree for a given list of elements from a totally ordered set.
• Understand how to build the decision tree for a specific application and be able to use such a tree to find an algorithm to solve the given application.
• Understand the definition of a prefix code, and be able to find the Huffman binary coding given an alphabet and a set of letter frequencies. Also be able to decode binary encoded strings for a given prefix code.
• Understand how to find the universal address system for a given rooted m-ary tree.
• Be able to find the preorder, inorder, and postorder traversal for a given m-ary tree.
• Be able to draw the tree of a computation given in prenix, infix, or postfix form, and be able to compute the value of the computation.
• Know the definition of a spanning tree for a connected graph and be able to find a spanning tree for a graph using either a depth first search or a breadth first search.
• Understand how to use Prim's Algorithm and Kruskal's Algorithm to find a minimum spanning tree for a given connected weighted graph.

Part VI: Boolean Algebra

• Understand Boolean Operations, Boolean Expressions, and be able to compute the value of an expression.
• Understand Boolean functions and know the Boolean identities and be able to verify identities using value tables and 2-column proofs.
• Be able to find the sum-of-products representation of a function. Also understand how to verify that a set of Boolean operations is functionally complete and be able to rewrite expressions in terms of a given set of operations.
• Be able to find a circuit for a given Boolean function and be able to find the output of a given circuit. Also know how to find a circuit to perform a given task (such as binary addition, etc.)
• Be able to use K-maps to find a minimal expression for a Boolean function.