

CS A257: BOOLEAN ALGEBRA AND LOGIC

Item	Value
Curriculum Committee Approval Date	11/13/2024
Top Code	070600 - Computer Science (Transfer)
Units	3 Total Units
Hours	54 Total Hours (Lecture Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Open Entry/Open Exit	No
Grading Policy	Standard Letter (S)

Course Description

An introduction to the discrete structures used in Computer Science. Topics covered include basic logic, proof techniques, relations, Boolean algebra, logic gates, languages and grammars, finite-state machines, and Turing machines. PREREQUISITE: CS A131, CS A150, or CS A170. Transfer Credit: CSU; UC.

Course Level Student Learning Outcome(s)

1. Prove logical equivalency of propositions.
2. Derive Boolean expressions from truth tables and logic circuits.
3. Construct finite-state machines given specific parameters.

Course Objectives

- 1. Use standard notation of propositional logic.
- 2. Apply proof techniques.
- 3. Prove that logical expressions are or are not equivalent.
- 4. Use set notation for set operations.
- 5. Determine the characteristics associated with a given function, including injections, surjections, and bijections.
- 6. Define different types of relations.
- 7. Know how to scale a matrix, take the transpose of a matrix, and add and multiply matrices.
- 8. Construct truth tables.
- 9. Apply Boolean algebra to electrical switches.
- 10. Design simple circuits.
- 11. Construct finite-state machines, with or without output, given specific parameters.
- 12. Produce regular grammars for finite-state machines.
- 13. Use a Turing machine to recognize sets.

Lecture Content

Logic Propositional logic Applications of propositional logic Propositional equivalences Predicates and quantifiers Nested quantifiers Rules of inference Proofs Introduction to proofs Proof methods and strategy Basic Structures Sets Set operations Functions Relations Relations and their

properties n-ary relations and their applications Representing relations Closures of relations Equivalence relations Partial orderings Matrices Matrix arithmetic Transposes and power of matrices Zero-one matrices Boolean Algebra Boolean functions Representing Boolean functions Identities of Boolean algebra Duality Sum-of-products expansions Functional completeness Logic Gates Combinations of gates Circuits Adders Minimization of circuits Languages and Grammars Phase-structure grammars Types of phase-structure grammars Derivation trees Backus-Naur form Finite-state Automata with Output Finite-state machines with output Mealy machines Moore machines Finite-state Automata with no Output Finite-state machines with no output Deterministic finite-state machine Nondeterministic finite-state machine Language Recognition Regular sets Kleene's Theorem Pushdown automaton Turing Machines Definition of Turing machines Using Turing machines to recognize sets Computing functions with Turing machines

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)

Instructional Techniques

Lecture, demonstrations, and exercises.

Reading Assignments

Students will spend a minimum of 3 hours per week reading assignments from the class textbook and/or supplemental material illustrating programming syntax and techniques.

Writing Assignments

Students will spend a minimum of 6 hours per week writing logical, step-by-step solutions to selected out-of-class assignments.

Out-of-class Assignments

A minimum of 6 hours per week will be spent on homework exercises that will include: Finding solutions to regularly-assigned homework problems for each topic. Designing algorithms and/or creating applications to solve selected problems.

Demonstration of Critical Thinking

Critical thinking will be evaluated in both the midterm and final exams through a problem-solving approach.

Required Writing, Problem Solving, Skills Demonstration

Writing is encouraged throughout the course, but is not necessarily a part of the grading or the exams.

Eligible Disciplines

Computer science: Master's degree in computer science or computer engineering OR bachelor's degree in either of the above AND master's degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelor's degree in engineering AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in mathematics AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in any of the above AND a master's degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum

qualification purposes, under the discipline of the application. Master's degree required.

Textbooks Resources

1. Required Rosen, K.H.. Discrete Mathematics and Its Applications, 7th ed. McGraw-Hill, 2011
2. Required Vahid, F., Lysecky, R., Irani, S., Edgcomb, A., Majidi, A.J.. CS A257: Boolean Algebra Logic, ed. Wiley, 2019