

CS 65 Midterm Review, Fall 2008

These concepts were either discussed in lecture or represented in the assignments.

Functional programming concepts

- Functions
- Lists, association lists, functions dealing with these
- Possible data representations for directed graphs, matrices
- Anonymous functions, lambda expressions, Currying
- Recursion, Mutual recursion, Tail recursion
- Accumulator arguments

KWIC Index

Directed graph cycle test

Dijkstra's algorithm

Interpreter for logic functions

- Using environment to represent variable binding

Logic functions, identities, simplification

- Minterm expansion
- Boole-Shannon expansion
- Simplification using hypercubes and Karnaugh maps
- Combinational logic implementation using gates

Typical combinational elements, such as adder, multiplexor, encoder, decoder

Sequential logic, finite-state-machines, Mealy/Moore, acceptors

- Tabular representation
- Labeled graph representation
- Gate implementation

Typical sequential elements, such as register, counter

Regular expressions

- Uses for string matching
- Derivatives
- Conversion to finite-state acceptor
 - Using derivatives
 - Using NFA to DFA conversion (non-deterministic machines)

Cross product machines

- Implementing Boolean operators, such as intersection, equality check

Non-regular languages

- How to show a language is non-regular