**COT3210–Computability and Automata**

**Final Exam Study Guide**

Expected Outcomes

1. Construct the basic kinds of finite automata and describe their formal representations (Sec. 1.1–1.2);
2. Construct finite automate corresponding to given regular expressions and vice versa (Sec. 1.3);
3. Construct context-free grammars for given context-free languages and vice versa (Sec. 2.1);
4. Construct pushdown automata to recognize context-free languages (Sec. 2.2);
5. Describe Turing Machines and explain their relation to language recognition and simple computations (Sec. 3.1); and
6. Discuss concepts of computability and complexity (Sec. 7.1–7.4).

Chapter 1 – Regular Languages

Section 1.1 – Finite Automata (FA) (All examples are DFAs)

* Finite State Machines (FSM) – Useful tools for modeling computers with a limited amount of resources.
  + Conveyed using the following methods:
    - Visually
      * Model
      * Transition Table
    - Formal Definition
  + Simple FSM Example – Automatic Door (Figure 1.2)
* Characteristics of FSM:
  + Nodes
  + Arcs (Edges)
  + Start State – Can be only one (Indicated with 🡪)
  + Final State – Can be more than one (Indicated with Double Circles)
* Purpose of a FSM is to accept or reject strings from the input:
  + The input string consists of symbols in an alphabet (Σ):
    - All the alphabet must be able to be processed by the FSM.
  + How to tell if string is accepted:
    - At end of string input if current state in accepted the string is accepted.
    - At end of string input if current state is not accepted the string is rejected.
* Languages:
  + Example of a Language:
    - L(M) = {w | where w contains at least one 1 and an even number of 0s follow the last 1}.
  + Regular Languages:
    - We say that a language is regular if some finite automaton recognizes it.
* Operations:
  + Union: A U B = {x | x ∈ A or x ∈ B}
  + Concatenation: A ◦ B = {xy | x ∈ A and y ∈ B}
  + Star: A\* = {x1x2 · · · xk | k ≥ 0 and each xi ∈ A}
* DFA Characteristics:
  + Unique Next State
  + One edge for one Label
  + No ε transitions
  + No choices for next state
  + Only one state to consider for processing strings

Section 1.2 – Nondeterministic Finite Automata (NFA)

* NFA Characteristics:
  + May be multiple next states
  + May be multiple edges for the same label
  + ε transitions allowed
  + Next state chosen at random
  + All possible next states chosen in parallel
* Regular operations:
  + The operations Union, Concatenation, and Star are now much easier to show in NFAs compared t before with DFAs.
* DFA to NFA Conversion: