Automata, Logic and Computability

Introduction to Automata and Computability / Regular Languages and Finite Automata - Lecture 1

James Marshall

Automata and computability underlies computer science and the study, design and programming of computational devices.

This course will introduce you to:

1. Models of computation of varying computational power

2. The fundamental limits of computation

3. The distinction between computationally tractable and intractable problems

The abstract models of computation you will be introduced to have direct application in computer science and computer systems engineering:

Computational Model	*Application*
Finite automata / regular languages	
Pushdown automata / context-free languages	
Turing machines / Turing-recognisable languages	

A Simple Example Controller (2 Light Switches for Stairwell Light)

Circuit diagram

State transition diagram (finite automaton)

Definition - Deterministic Finite Automaton

A ***deterministic finite automaton*** *is a 5-tuple* ($Q, \Sigma, \delta, q_0, F$) where

1. *Q is a finite set of* ***states***

2. *Σ is a finite set called the* ***alphabet***

3. *δ: $Q \times \Sigma \rightarrow Q$ defines the* ***transition function***

4. *$q_0 \in Q$ is the* ***start state***

5. *$F \subseteq Q$ is the set of* ***accept states***

Definition - Regular Language

A ***language*** *is called a* ***regular language*** *if some finite automaton* ***recognises*** *it.*

An automaton ***recognises*** *a language if it* ***accepts*** *all strings that are members of the language.*

Examples

- Binary numbers divisible by 2: {0, 10, 110, 100, ...}
- Regular expressions over some alphabet: *e.g.* , , ...
- *etc.*

Q: Can you devise a language that is not regular?