INTRODUCTION TO
DISCRETE STRUCTURES

11.1 Finite State Machines

1.1–1.2

11.1 Finite State Machines

- Theory of computation
- Finite State Machine
- Examples of FSM

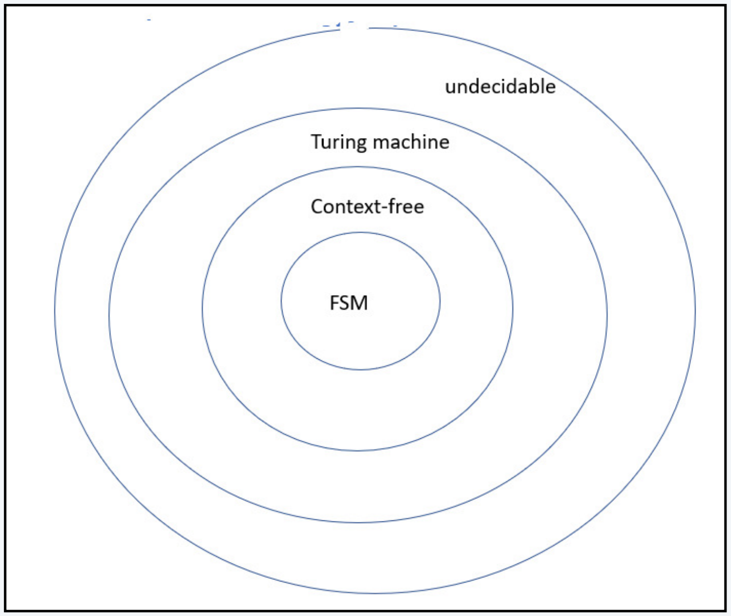
Theory of Computation

The theory of computation is the study of general properties of computation

It studies the questions of how efficiently problems can be solved on a model of computation using an algorithm.

Basic Question. What are the fundamental capabilities and limitations of computers?

Computing Categories



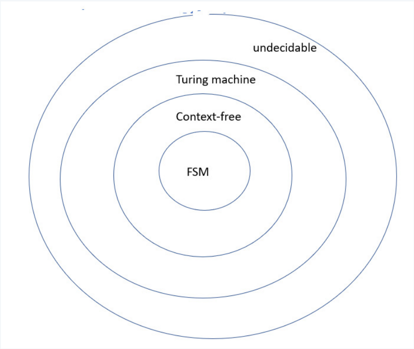
Classes of problems

Undecidable. Requires a Yes, No answer. But there cannot possibly be any computer program that always gives the correct answer. Sometimes gives wrong answer or run forever. e.g. can a given a set of matrices be multiplied in some order to get zero matrix?

Turing machine. A mathematical model of computation that defines an abstract machine. There exists a Turing algorithm for any problem that can be solved by a modern computer. TM has memory

Context-Free Grammar. A set of production rules that describes all possible strings in A formal language. Languages generated from context-free grammars are called context-free languages (Java). Rule based.

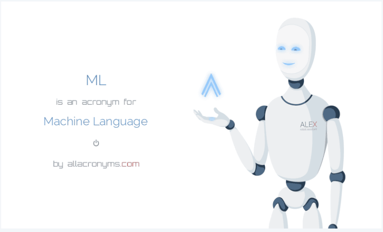
Finite State Machine. A mathematical model of computation with no memory. A transition machine. FSM equivalent to regular grammar.



Languages are machines

L(G) = Language derived by the grammar

L(G) = { w | w in T*, S ➔* w}, where T is a subset of V (vocabulary) consisting of terminal symbols and S is a start symbol. L(G) is the set of all strings derivable from language rules.



Example. V={0,1,A,B,S}, T={0,1}

Production rules. S ➔ 0A, S➔1A, A➔0B, B➔1A, B➔1

Can $0^3 1^3$ be derived from this grammar?

If not, what sentences can be derived?

A language generates a set of strings accepted by the language.

Kleene's Theorem. A set is called regular if and only if it is recognized by a Finite State Automaton

INTRODUCTION TO
DISCRETE STRUCTURES

11.1 Finite State Machines

- Theory of Computation
- Finite State Machines
- Examples of FSM

Notations used

FSM – Finite State Machine

FSA – Finite State Automata

DFA – Deterministic Finite Automata

NFA – non-deterministic finite automata

Vending machine example

What is a vending machine?

A machine that transition from state to state based on inputs

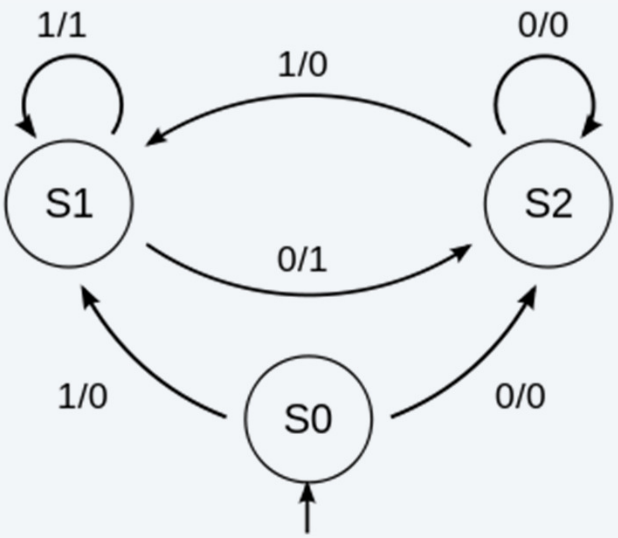
Possible states = {0-cent, 25-cent, 50-cent}

Possible inputs = {25-cent, cancel, dispense}

Possible outputs = {coke, 25-cent, 50-cent}

Formalizing a state machine

A finite state machine M = (S, I, O, f, g, s_0) consists of a finite set of states (S), finite input alphabet (I), finite output alphabet(O), a transition function f (assigns each state and input pair to a new state) and output function g(assigns to each state and input pair an output) and an initial state s_0



Transition Table

A FSM can be defined by its transition function f

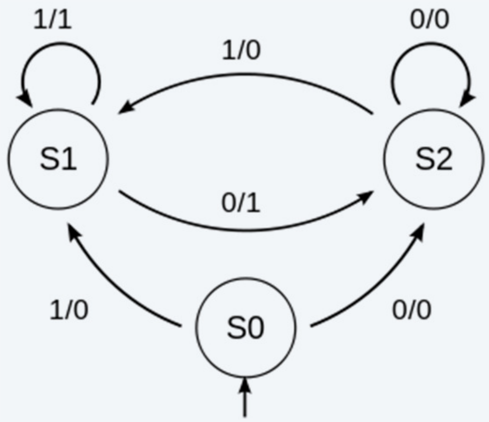
f(s0, 0) = s2 (output 0)

f(s0, 1) = s1 (output 0)

f(s1, 1) = s1 (output 1)

f(s1, 0) = s2 (output 1)

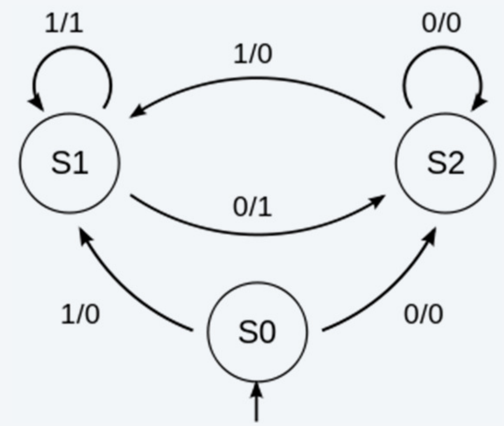
….



workshop

Find the output for the FSM for the given input

Input	1	0	1	0	1	1
State						
output						



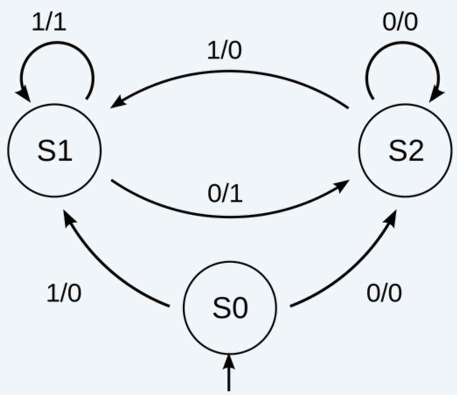
INTRODUCTION TO
DISCRETE STRUCTURES

11.1 Finite State Machines

- Theory of Computation
- Finite State Machine
- Examples of FSM

workshop

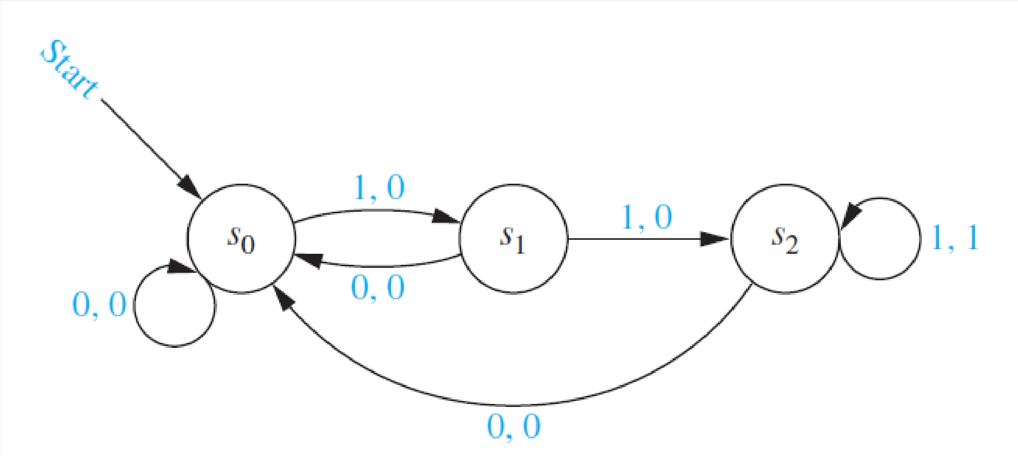
A finite state machine with output.



Find the output given the input: 0 0 1 0 1

Example 2

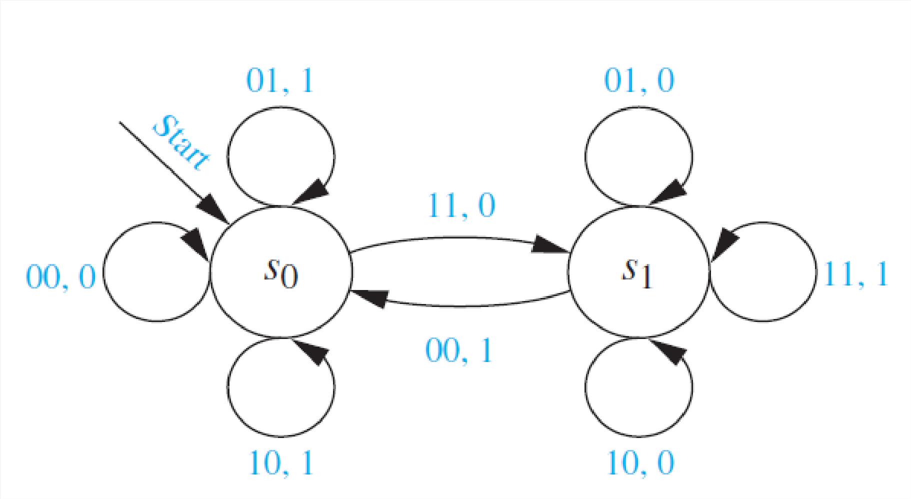
Consider the finite state machine below.



What is the output given the input : 0 0 1 0 1 1 1

A Mystery FSM

What does this FSM do, given the following possible inputs and outputs?



Possible Inputs: {00, 01, 10, 11}

Possible Outputs: {0, 1}

What is the outcome of this input sequence? 01 00 11 01 11

INTRODUCTION TO DISCRETE STRUCTURES

11.1 Finite State Machines

- Theory of computation
- Finite State Machine
- Examples of FSM

INTRODUCTION TO
DISCRETE STRUCTURES

11.1 Finite State Machines

1.1–1.2