Contents

1 Lecture 1

1.1 What is Computation?

"Computation is the evolution process of some environment by a sequence of simple, local steps."

- Avi Wigderson, *Math & Computation*

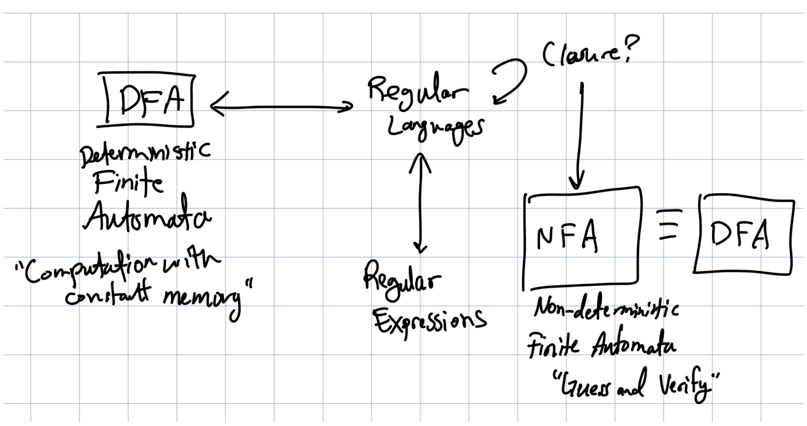
Here are some examples of what computation may be:

- How bits evolve in a computer.
- Computers in a network.
- Atoms in matter.
- Neurons in the brain.
- Prices in a market.

Note 1.1 (Models of Computation)
Mathematical modeling of computation gives us systematic ways to think and argue about computers.

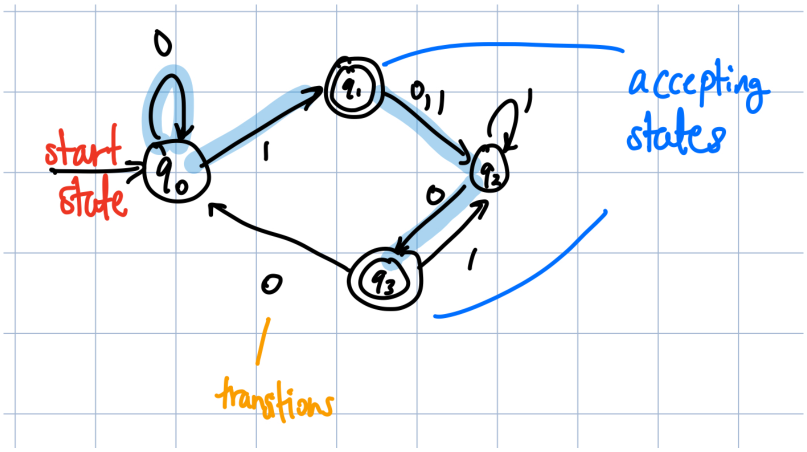
- Independent of architecture.
- Idealized model: accurate in some aspects while not in others.
- Much simpler than an actual computer but captures key properties of it.

Here is a roadmap.



1.2 Deterministic Finite Automata

Here is a Deterministic Finite Automaton or Finite State Machine.



A computation on 0110 is highlighted on the diagram. Since we ended at q_3, which is an accepting state. This means the string 0110 is accepted by this machine. A string that terminates a computation but it not accepted is rejected.

Why are DFAs useful?

- They are a simple model of computation
- Useful for verification, compilers
- Close to Turing machines but we have a much better understanding of them
- Taste of "non-determinism", limitations
- Similar to streaming algorithms

Definition 1.1 (Alphabet, String, Language)
An alphabet Σ is a finite set of characters, e.g. $\Sigma = \{0, 1\}$ or $\Sigma = \{A, \ldots, Z\}$.
A string over Σ is a finite length sequence of symbols from Σ.
For a string x, $|x|$ denotes the length of the string.
In addition:

$$\Sigma^* = \{\text{string over } \Sigma$$

And the empty string is ε, where $|\varepsilon| = 0$.
A language over Σ is a set of strings over Σ.

Now we can formally define a DFA.

Definition 1.2 (Deterministic Finite Automaton)
A DFA (Deterministic Finite Automaton) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$:

- Q is the set of all states
- Σ is the alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of all accept/final states

Then we can define our computation more formally.

Definition 1.3 (Computation Path, Acceptance)
Let $w = w_1 w_2 \ldots w_n$ be a string in Σ^* of length n. The computation path of an automata M on w is the sequence of states $r_0, r_1, \ldots, r_n \in Q$ defined by:

- $r_0 = q_0$
- $r_i = \delta(r_{i-1}, w_i)$

M accepts w if and only if $r_n \in F$.

And finally we get to the idea of a language corresponding to a DFA.

Definition 1.4
If M is a DFA, then $L(M)$, the language recognized by M is the set of all strings that M is accepted.