Theory of Computation

Fall 2023

The Basic Questions

- What can a computer do at all?

(Computability-Decidability)

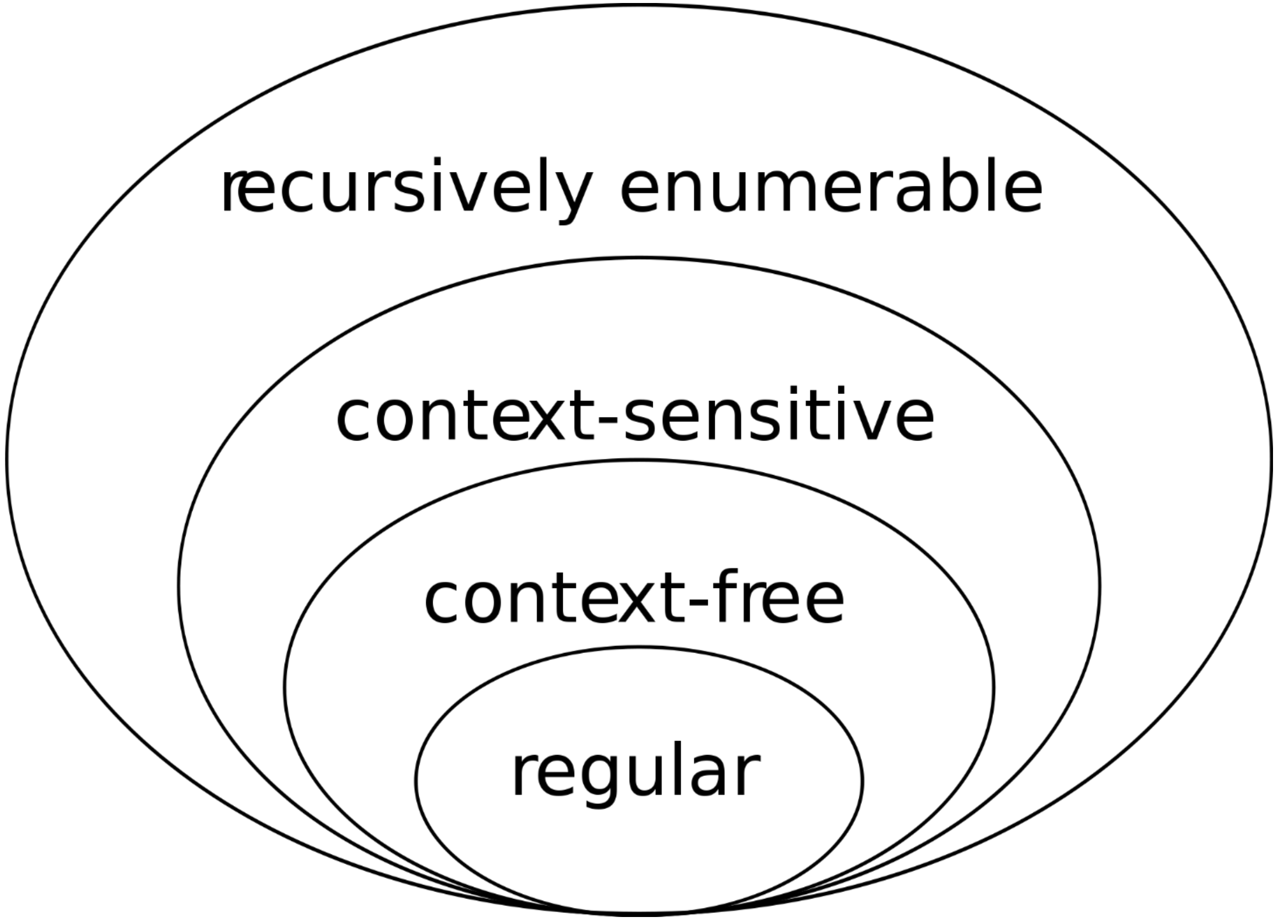
- What can a computer do efficiently?

(Complexity-Intractability)

Methodology

- Interplay between languages and machines
- Mathematical model = Machine = Automata

The Chomsky Hierarchy (Language Hierarchy)



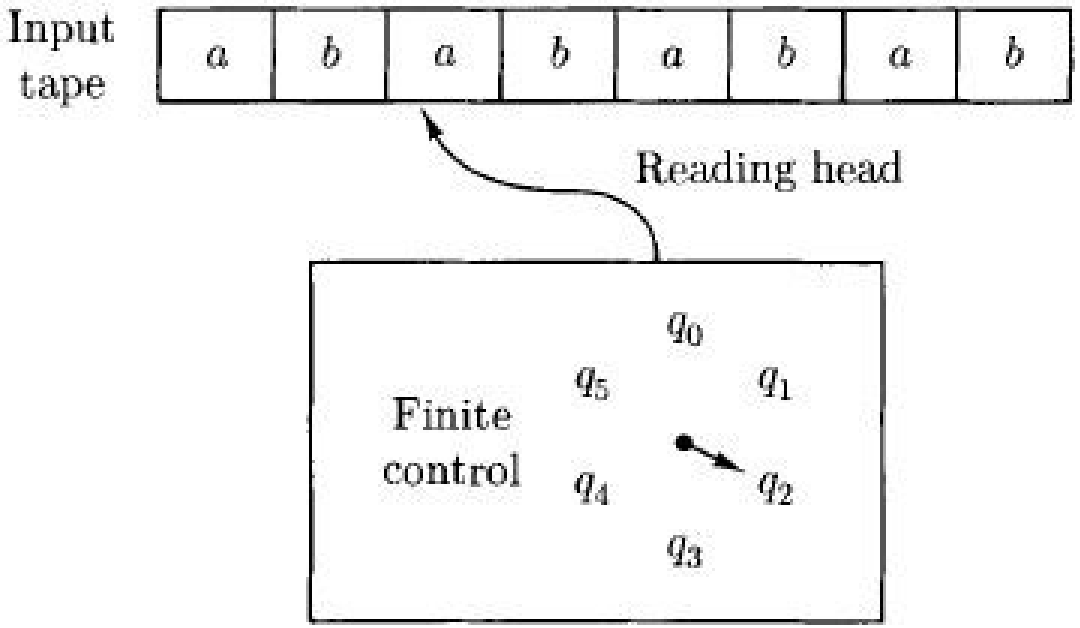
Definitions

- Alphabet: ❮: A finite set of symbols.
- Empty string, e.
- ❮*:

  The set of all strings, including the empty string, over an alphabet ❮.
- Language:

  Any set of strings over an alphabet ❮-that is any subset of ❮*.

Finite Automata



Machine Execution

- The movement of the machine is determined by the current state and input symbol.

Finite Automata

A deterministic finite automaton is a quintuple $M = (K, \Sigma, \delta, s, F)$ where

K is a finite set of states,

Σ is an alphabet,

$s \in K$ is the initial state,

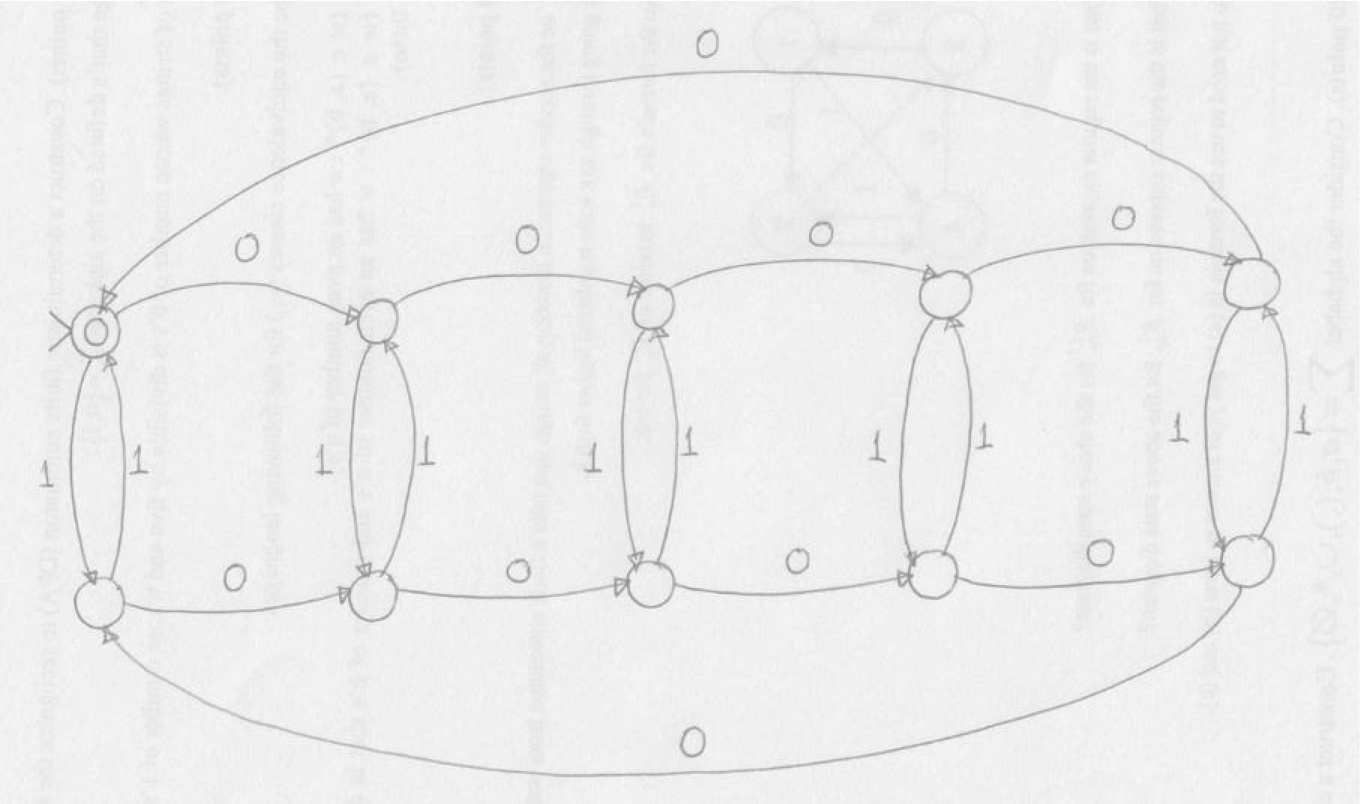
$F \subseteq K$ is the set of final states, and

δ, the transition function, is a function from $\mathrm{K} \times \Sigma$ to K.

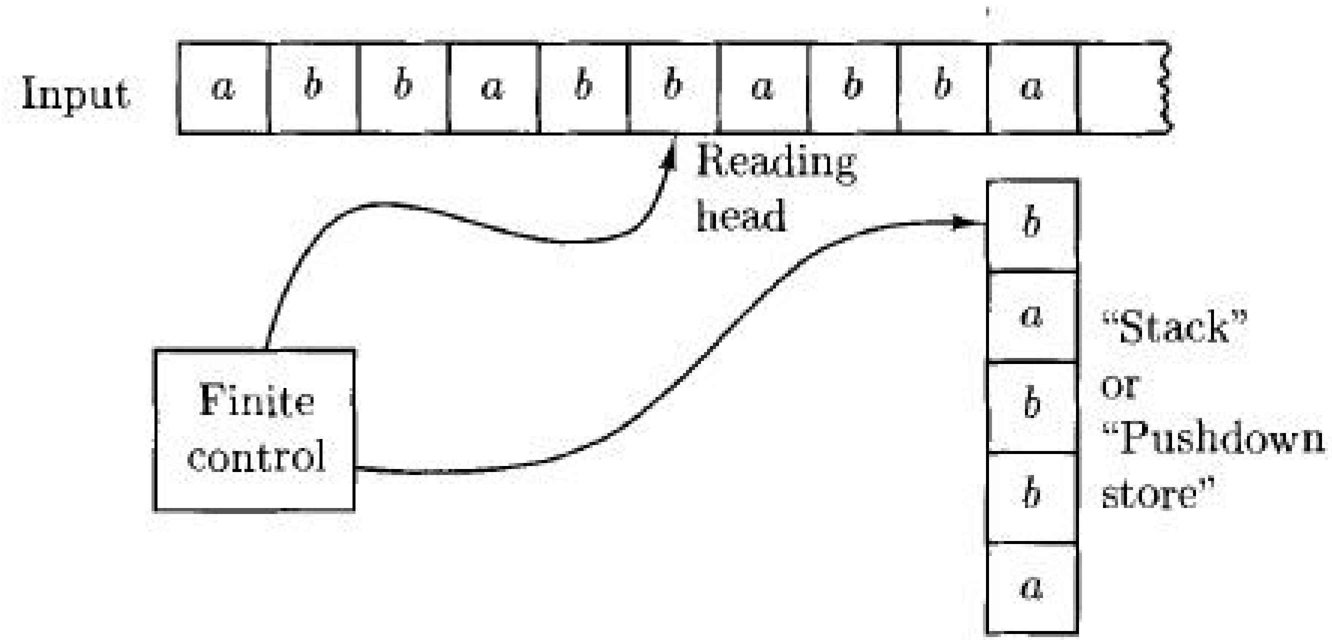
Finite Automata

Construct a deterministic finite automaton (DFA) to recognize the following language that's defined on the alphabet $\Sigma = \{0,1\}$:

The set of strings whose number of 0's is divisible by five and whose number of 1's is even.



Pushdown Automata



Pushdown Automata

A pushdown automaton is a sextuple $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

K is a finite set of states,

Σ is an alphabet (the input symbols),

Γ is an alphabet (the stack symbols),

Δ, is the transition relation, is a finite subset of $(\mathrm{K} \times (\Sigma \cup \{\mathrm{e}\}) \times \Gamma^*) \times (K \times \Gamma^*)$,

$s \in K$ is the initial state,

$F \subseteq K$ is the set of final states.

Pushdown Automata

Construct a pushdown automaton M to accept the language

$L = \{wcw^R : w \in \{a, b\}^*\}$.

(1) $((s, a, e), (s, a))$

(2) $((s, b, e), (s, b))$

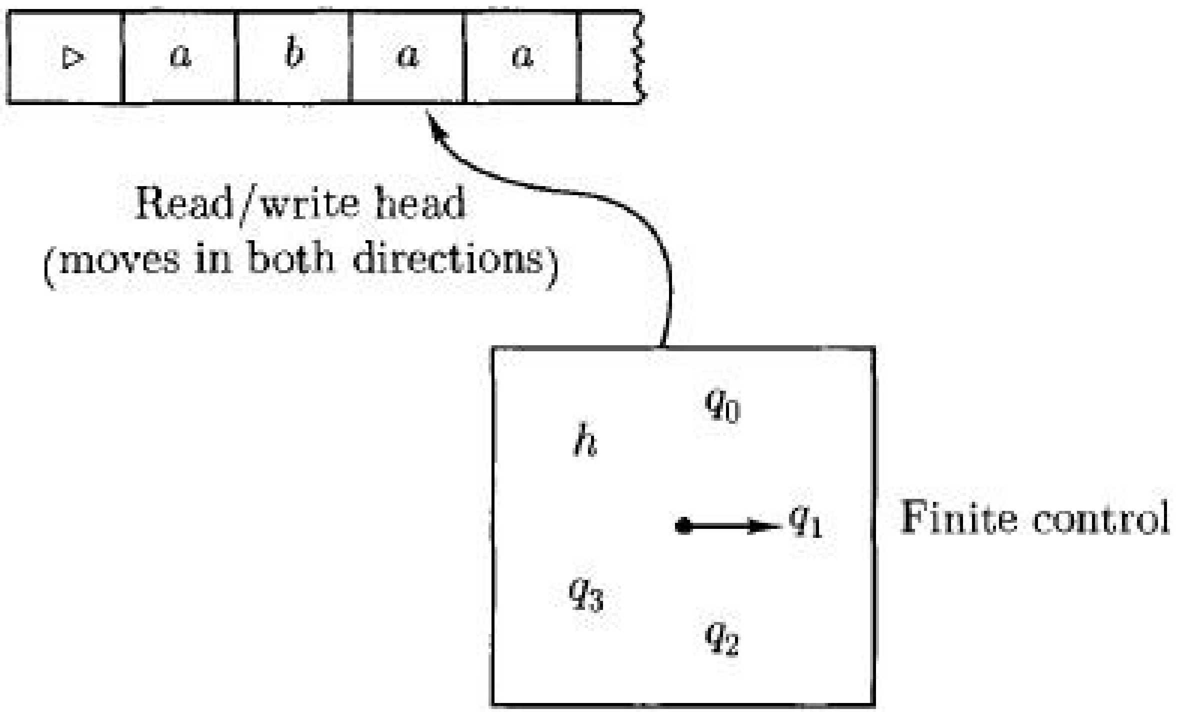
(3) $((s, c, e), (f, e))$

(4) $((f, a, a), (f, e))$

(5) $((f, b, b), (f, e))$

State	Unread Input	Stack	Transition Used
s	$abbcbba$	e	-
s	$bbcbba$	a	1
s	$bcbba$	ba	2
s	$cbba$	bba	2
f	bba	bba	3
f	ba	ba	5
f	a	a	5
f	e	e	4

Turing Machines



Turing Machines

A Turing machine is a quintuple $M = (K, \Sigma, \delta, s, H)$, where

K is a finite set of states,

Σ is an alphabet, containing the blank symbol $\sqcup$ and the left end symbol $\triangleright$,

but not containing the symbols $\leftarrow$ and $\rightarrow$,

$s \in K$ is the initial state,

$H \subseteq K$ is the set of halting states,

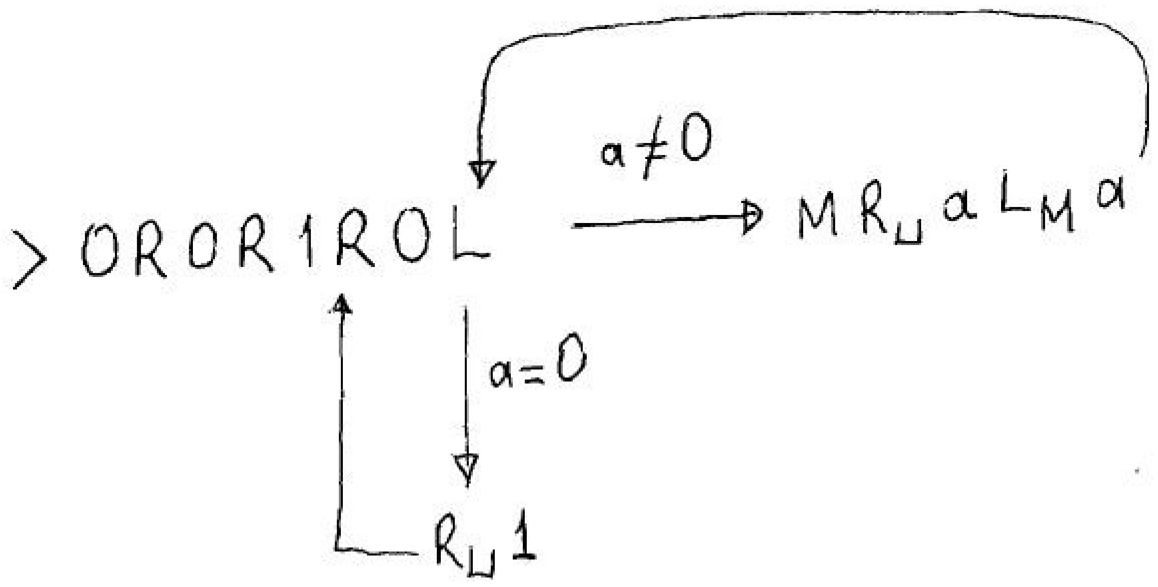
δ, the transition function, is a function from $(K - H) \times \Sigma$ to $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ such that

(a) for all $q \in K - H$, if $\delta(q, \triangleright)$, then $b = \rightarrow$

(b) for all $q \in K - H$ and $a \in \Sigma$, if $\delta(q, a) = (p, b)$ then b $\neq \triangleright$.

Turing Machines

Construct a Turing machine to compute the sequence 001011011101111011111… on the alphabet $\Sigma = \{0,1\}$:



Interplay between Languages and Machines (Automata)

- Regular Languages-Finite Automata
- Context-Free Languages-Pushdown Automata
- Recursive Languages-Deterministic Turing Machines
- Recursively Enumerable Languages-Nondeterministic Turing Machines

Determinism vs. Nondeterminism

- The movement of the machine is only partially determined by the current state and input symbol.
    - It can enter into a new state by consuming nothing.
    - It has potential to move into more than one state.