CSC 320 - Lecture 02

#languages #countability #automaton #DFA

Alphabets, Languages, Strings, Symbols

How Large is Σ^* ?

Is it finite or infinite. And if it is infinite is it countably infinite or uncountable.

We say that it is countably infinite (and therefore countable)!

Proof. $\Sigma = \{0, 1\}$

$\mathbb{N}$	Σ^*
0	ε
1	0
2	1
3	00
4	01
...	...

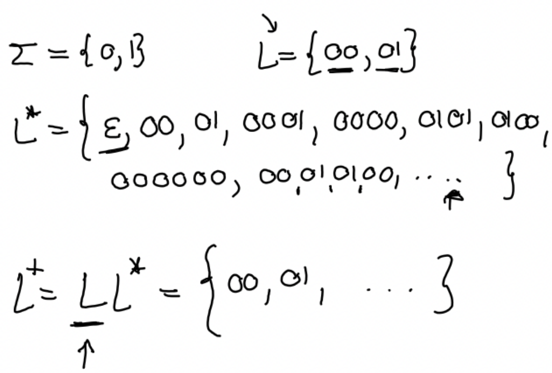
Operations and Relations on Strings

- **Concatenation** for strings x and y yields string xy.
  - Concatenation is an associative operation: $xyz := (xy)z = x(yz)$.
  - Eg. If $x = ab$, $y = bac$, and $z = bba$, then $xyz = abbacbba$.
- String v is a **substring** of string w if and only if there are strings x and y such that $w = xvy$.
  - We note that x and y can be the empty string.
  - If $x = \varepsilon$, then $w = xv$ and v is called **suffix** of w.
  - If $y = \varepsilon$, then $w = vy$ and v is called **prefix** of w.
  - Eg. If $w = abbacbba$, then $cbba$ is a suffix of w and abb is a prefix of w.
- A string w written backwards is denoted w^R and called the **reversal** of the string w.
  - Eg. If $w = abbacbba$, then $w^R = abbcabba$.

Languages

- A **language** is a set of strings over an alphabet Σ.
  - Set operations apply to languages (eg., union, intersection, set difference).
- For a language L over alphabet Σ, its **complement** is $\Sigma^* - L$, denoted $\bar{L}$.
- Given languages L_1 and L_2 over alphabet Σ, their **concatenation**, denoted L_1L_2 is defined by...
  - $L_1L_2 = \{w \in \Sigma^* | w = xy \text{ for some } x \in L_1 \text{ and } y \in L_2\}$
- The **Kleene star** L^* of a language L is the set of all strings obtained by **concatenating zero or more strings** from L:
  - $L^* = \{w \in \Sigma^* | w = w_1w_2 ... w_k, k \geq 0 \text{ and } w_i \in L \text{ for } 1 \leq i \leq k\}$
- Given a language L over alphabet Σ, the closure L^+ of L is $L^+ = LL^*$
  - L^+ is smallest language that includes L and all strings that are concatenations of strings in L.

Example



Note. They are all even. They can also never start with 1. ε is not in L, so it can be in L^*.

Observations

Let Σ be an alphabet. Then for any language L over Σ: $L \subseteq \Sigma^*$ and $\bar{L} \subseteq \Sigma^*$

How Large is the Set of All Languages over Σ ?

Recall. Σ^* is countably infinite (and therefore countable) and that languages are subsets of Σ^*.

Then the **set of all languages** is the set of all subsets of Σ^*: $\mathcal{P}(\Sigma^*)$.

Any language L, L is countably infinite or finite.

How large is the set of all languages over an alphabet Σ ?

Cardinality of the set of all languages equals cardinality of $\mathcal{P}(\Sigma^*)$: $|\mathcal{P}(\Sigma^*)|$

What does $|\mathcal{P}(\Sigma^*)| =?$

- **Idea**. Show that the powerset of any countably infinite set is uncountable. This would imply that the powerset $\mathcal{P}(\Sigma^*)$ is uncountable.
- **Recall**. Any countably infinite set has a bijection with $\mathbb{N}$, that is Σ^* has a bijection with $\mathbb{N}$.
- Therefore, if we show that $\mathcal{P}(\mathbb{N})$ is uncountable then we know that $\mathcal{P}(\Sigma^*)$ is uncountably infinite.

$|\mathcal{P}(\Sigma^*)|$ is uncountable

Proof by Contradiction

Goal. Define a subset that should be on the list but is not.

Assume that $\mathcal{P}(\mathbb{N})$ is **countably infinite**. We list every subset of $\mathbb{N}$ as $S_0, S_1, S_2, \ldots$ such that every subset of $\mathbb{N}$ is equal to a subset S_i for some i.

Next we define the subset $D \subseteq \mathbb{N} : D = \{i \in \mathbb{N} | i \notin S_i\}$. For each $j \in \mathbb{N}$, $j \in D$ if and only if $j \notin S_j$. Since $D \subseteq \mathbb{N} : D$ is on the list and there exists j with $S_j = D$.

- If $j \in D$ then $j \notin S_j = D$.
- If $j \notin D$ then $j \in S_j = D$.

That is, $j \in D$ if and only if $j \notin D$. Thus D cannot be a subset of $\mathbb{N}$.

Thus we obtain a contradiction (we used diagonalisation to achieve contradiction)!

Recap. Since $\mathcal{P}(\mathbb{N})$ is uncountably infinite, the powerset of any countably infinite set is uncountable. Since the set of all languages is the powerset of Σ^*, the set of all languages is uncountable.

Finite Automata and Regular Languages

Automata Theory: Finite Automata & Regular Languages. Pushdown Automata & Context-Free Languages.

- Understanding **computability** requires: models of a computer that capture its computational power.
- Most simple model: **finite state machine** or **finite automaton**. Model of computation with **finite amount of memory**, independent of problem size.

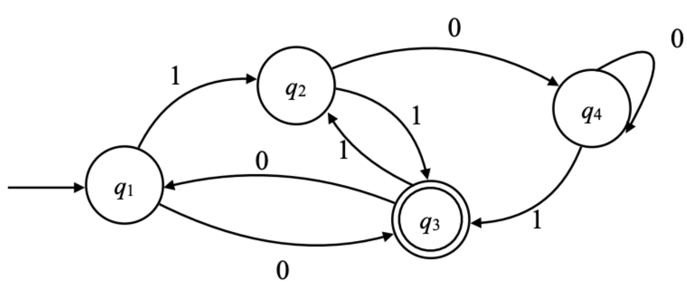
Finite Automata in Practice

- Automatic door controller: a finite automaton/computer with just a single bit of memory that records which of the two states the controller is in (closed or open).
- Many other common devices have controllers with somewhat larger memories.
- **Elevator Controller**: state represents floor elevator is on and inputs signals received from the buttons.
- **Controllers for Various Household Appliances**: dishwasher, electronic thermostats, parts of simple digital watches, and simple calculators.
- Other Applications: pattern recognition, speech recognition, optical character recognition, compilers, a (probabilistic) relative of the finite automaton (**Markov Chain**).

Abstract Description of the Finite Automaton

- **State Diagrams**: used to describe finite automata.
- Formal Definition: **Deterministic Finite Automaton**.

State Diagram



The states are $\{q_1, q_2, q_3, q_4\}$. The start state is q_1. The accept state is q_3.

- Transitions: arrows from one state to another (according to received inputs).
- Inputs (labels on transition): symbols from alphabet.

The above DFA accepts 11011 since it is in an accept state at the end of reading/processing of input string. The string processes until the end of the string. It doesn't stop at the first encounter of accept state. 0 is, but 00 is not.

Practice: How to explain what can and can't be included.

Formal Definition Deterministic Finite Automaton

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

1. Q is a finite set called the **states**.
2. Σ is a finite set called the **alphabet** (alphabet must always be finite).
3. Function $\delta : Q \times \Sigma \to Q$ is the **transition function**.
4. $q_0 \in Q$ is the **start state**. (Unique).
5. $F \in Q$ is the **set of accept** (or **final**) **states**. (Multiple).

Note. The **Transition Function** must be well defined: for each state there are as many outgoing transitions as symbols in the alphabet.

Eg. $(\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_1, \{q_3\})$ with $\delta : Q \times \Sigma \to Q$ defined by...

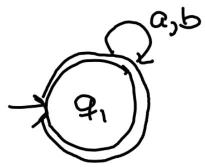
Transition Table:

δ	0	1
q_1	q_3	q_2
q_2	q_4	q_3
q_3	q_1	q_2
q_4	q_4	q_3

OR defined as... $\delta(q_2, 0) = q_4$.

Examples

A



$A = (Q, \Sigma, \delta, q_0, F)$

Valid DFA?

$Q = \{q_1\}$

$w \in \Sigma^*$

$\Sigma = \{a, b\}$

$\varepsilon \in L(A)$

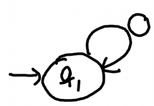
$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_1$

~~Start State~~: q_1

$F = \{q_1\}$

B



$B = (Q, \Sigma, \delta, q_0, F)$

Valid DFA?

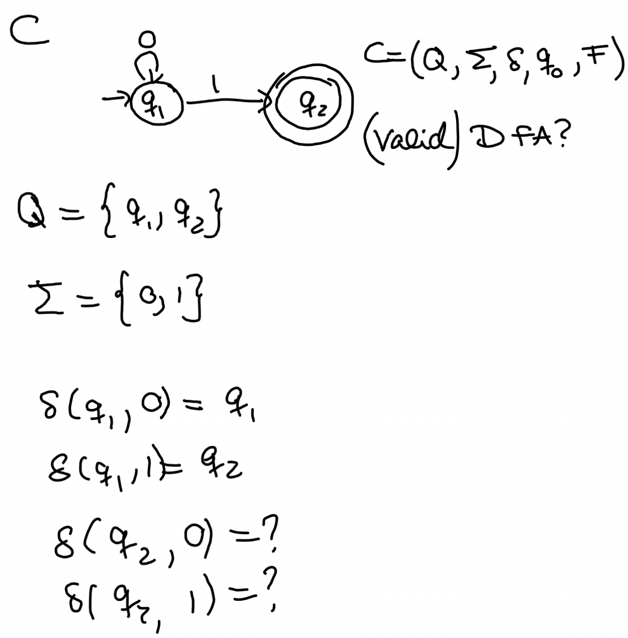
$F = \emptyset$ $L(B) = \emptyset$

$Q = \{q_1\}$

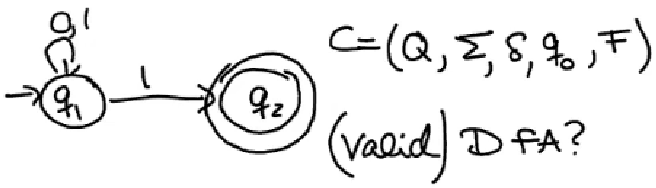
$\Sigma = \{0\}$

$\delta(q_1, 0) = q_1$

~~Start State~~: q_1



Note. Example C Part 1 one isn't valid because q_2 doesn't have any outgoing arrows for all elements Σ.



Note. This one isn't valid because there are two options for 1 going out of q_1. And we can't have two possibilities in a DFA.

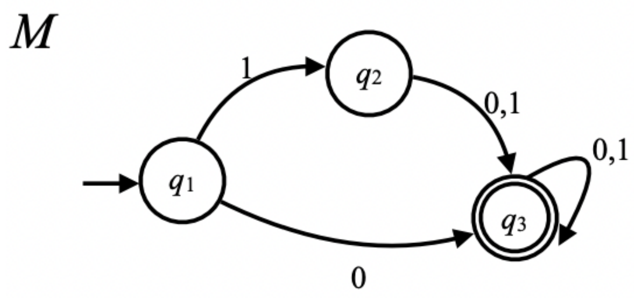
Language of a Deterministic Finite Automaton

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and A be the set of all strings that M accepts:

- A is called the **language of machine** M.
- $L(M) = A$.
- M **recognizes** language A.

Note. A machine that accepts no string still recognizes a language: **empty language** $\emptyset$. (This is very important)!

Example



Let $L(M) = \{w | w \in \Sigma^*$ of length at least 1 where: if w starts with symbol 1 then w is of length at least 2$\}$.

$(\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_1, \{q_3\})$ with $\delta : Q \times \Sigma \to Q$ defined by...

Transition Table:

δ	0	1
q_1	q_3	q_2
q_2	q_3	q_3
q_3	q_3	q_3

Previous Lecture

Lecture01

Next Lecture

Lecture03