Introduction to Computation Theory

Lecture Slides 1–2

Vinsong

NTU AMFL

October 18, 2025

Course Roadmap

Lecture 1: Basic Knowledge

Section Map

Lecture 1 Goals

- Review foundational mathematical objects needed for automata theory.
- Establish proof vocabulary and common proof strategies.
- Introduce deterministic finite automata and regular operations.

Set Basics

Definition (Set)

A set collects distinct elements without regard to order or multiplicity.

- $\{1, 2, 3\} = \{3, 2, 1\}$ because order is ignored.
- $\{1, 1, 2, 3\} = \{1, 2, 3\}$ because repetition does not change membership.

Sequences and Tuples

Definition (Sequence)

An ordered list of objects where order and repetition matter.

Definition (Tuple)

A finite sequence; $(1, 2, 3)$ is a 3-tuple.

- $(1, 2, 3) \neq (2, 1, 3)$ since order differs.
- $(1, 2, 2)$ and $(1, 2, 3)$ are distinct because repetition counts.

Sets vs. Sequences

- **Order**: ignored in sets, preserved in sequences.
- **Multiplicity**: collapsed in sets, tracked in sequences.
- Example:

$$\text{Sequence } (1,1,2,3) \neq (1,2,3), \quad \text{Set } \{1,1,2,3\} = \{1,2,3\}.$$

Cartesian Product

Definition (Cartesian Product)

$A \times B = \{(a, b) \mid a \in A, b \in B\}$.

Example

For $A = \{1, 2\}$ and $B = \{x, y\}$,

$$A \times B = \{(1, x), (1, y), (2, x), (2, y)\}.$$

Function as a Machine

Definition (Function)

Associates each input with exactly one output.

- Domain: permissible inputs.
- Codomain: possible outputs.
- Think of a machine with a single exit.

Equivalence Relation

Definition (Equivalence Relation)

A relation R is equivalent when

- Reflexive: $\forall x,\ xRx$.
- Symmetric: $\forall x, y,\ xRy \Rightarrow yRx$.
- Transitive: $\forall x, y, z,\ xRy \land yRz \Rightarrow xRz$.

Example: Modulo 7

Example

$i \equiv_7 j$ if $i - j$ is divisible by 7.

- Reflexive: $i - i = 0 \equiv 0 \pmod 7$.
- Symmetric: $i - j = 7a \Rightarrow j - i = -7a$.
- Transitive: $i - j = 7a$ and $j - k = 7b \Rightarrow i - k = 7(a + b)$.

Alphabet and Strings

- **Alphabet**: finite set of symbols, e.g., $\{0, 1\}$.
- **String**: finite sequence of symbols, e.g., 01000.

Languages

Definition (Language)

A set of strings over an alphabet. For machine A, write $L(A)$.

- Languages can be finite or infinite.
- Example: all binary strings ending with 01.

Terminology

- **Definition**: introduces a new concept.
- **Statement**: a sentence that is true or false.
- **Theorem**: a statement proven true.
- **Lemma**: a helping theorem.
- **Corollary**: follows quickly from another theorem.

Proof by Construction

Proposition

The sum of degrees of every graph is even.

Idea.

Each edge touches two vertices, so

$$\sum_{v \in V} \deg(v) = 2|E|,$$

which is even. $\square$

elies on the definition of how edges contribute to degree.

Proof by Contradiction

- Assume the statement is false.
- Deduce a contradiction.
- Conclude the original statement must be true.

Proof by Induction

- Basis: prove the statement for a simple starting case, e.g., $n = 0$ or $n = 1$.
- Inductive Step: assume truth for $n = k$ and prove for $n = k + 1$.

Automata Vocabulary

- **Automaton**: a single machine.
- **Automata**: plural form.

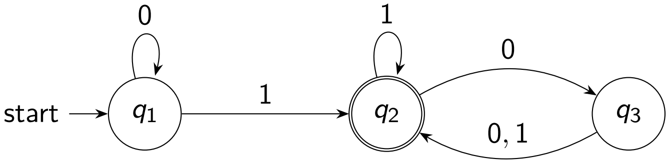
Definition of DFA

Definition

A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q: finite set of states,
- Σ: finite alphabet,
- $\delta : Q \times \Sigma \to Q$,
- $q_0 \in Q$: start state,
- $F \subseteq Q$: accept states.

Example DFA



Transition Function Table

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Language of a Machine

Definition

The language recognized by M is $L(M) = A$. We say A is accepted by M.

Computation of a DFA

Let $M = (Q, \Sigma, \delta, q_0, F)$ and $w = w_1 \cdots w_n$.

Theorem

M accepts w if there exist states $r_0, \ldots, r_n$ such that

1. $r_0 = q_0$,
2. $r_{i+1} = \delta(r_i, w_{i+1})$ *for* $0 \leq i < n$,
3. $r_n \in F$.

Regular Languages

Definition (Regular Language)

A language is regular if it is recognized by some finite automaton.

Regular Operations

Let A and B be languages.

- Union: $A \cup B = \{w \mid w \in A \vee w \in B\}$.
- Concatenation: $A \circ B = \{w_1 w_2 \mid w_1 \in A,\ w_2 \in B\}$.
- Kleene Star: $A^* = \{w_1 \cdots w_k \mid k \geq 0,\ w_i \in A\}$.

Kleene Star Detail

$$A^* = \{\epsilon\} \cup A \cup A^2 \cup A^3 \cup \cdots$$

with

- $A^0 = \{\epsilon\}$,
- $A^n = \{wv \mid w \in A^{n-1},\ v \in A\}$ for $n \geq 1$.

Closure Concepts

Definition (Closed Operation)

An operation R is closed on a set A if $x, y \in A$ implies $xRy \in A$.

Theorem

Regular languages are closed under union, concatenation, and Kleene star.

Closure Under Union

Given

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1), \quad M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2),$$

build product automaton

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

with

- $Q = Q_1 \times Q_2$,
- $q_0 = (q_1, q_2)$,
- $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$,
- $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}$.

Then $L(M) = L(M_1) \cup L(M_2)$.

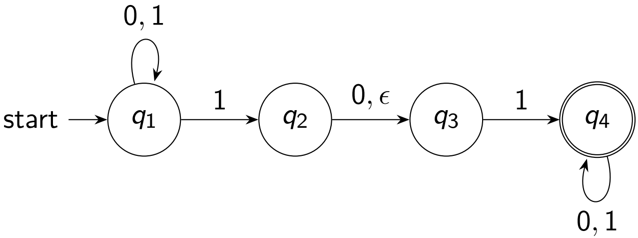
Lecture 2: Nondeterminism

Section Map

Lecture 2 Goals

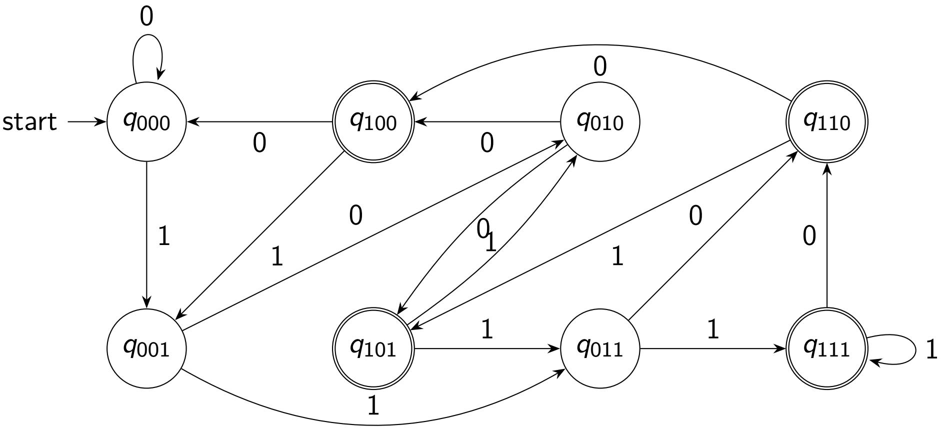
- Define nondeterministic finite automata (NFAs).
- Show equivalence between NFAs and DFAs.
- Construct NFAs for union, concatenation, and Kleene star.

Sample NFA



- Accepts strings with a 1 in the third position from the end.
- $\delta(q_1, 1)$ returns q_1 or q_2.
- ϵ between q_2 and q_3 consumes no input.

Determinizing the Example



- Subset construction encodes sets of NFA states as DFA states.
- May require up to $2^{|Q|}$ deterministic states.

Power Set Reminder

Definition (Power Set)

$P(Q) = \{X \mid X \subseteq Q\}$ contains all $2^{|Q|}$ subsets of Q.

Definition of NFA

Definition (NFA)

$M = (Q, \Sigma_\epsilon, \delta, q_0, F)$ with

- Q: finite set of states,
- $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$,
- $\delta : Q \times \Sigma_\epsilon \to P(Q)$,
- $q_0 \in Q$,
- $F \subseteq Q$.

Acceptance in an NFA

Take $w = y_1 \cdots y_m$ with $y_i \in \Sigma_\epsilon$.

Theorem

M accepts w if there exist states $r_0, \ldots, r_m$ such that

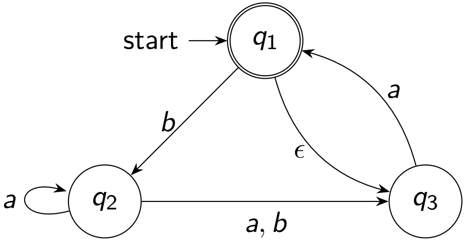
1. $r_0 = q_0$,
2. $r_{i+1} \in \delta(r_i, y_{i+1})$ *for* $0 \leq i < m$,
3. $r_m \in F$.

he sequence length m may differ from the input length because ϵ-moves consume no symbols.

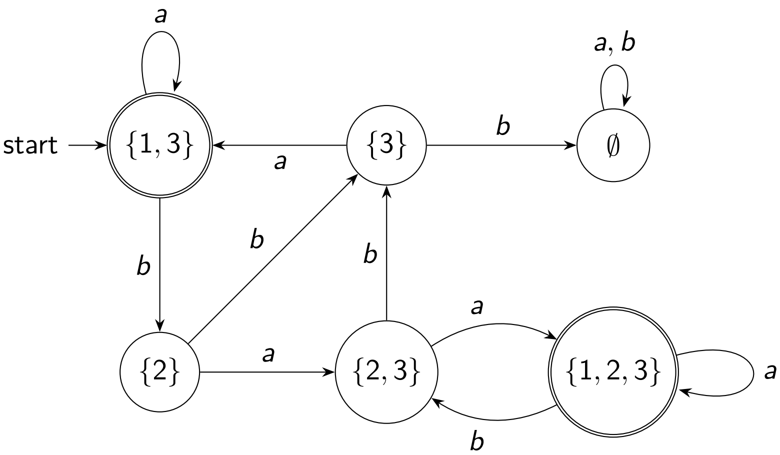
DFA as NFA

- A DFA is an NFA with no true nondeterminism.
- Add $\delta(q, \epsilon) = \emptyset$ to align alphabets: $\Sigma \subset \Sigma_\epsilon$.

Example NFA



Subset Construction Result



Refining the Converted DFA

- Remove unreachable subsets.
- Include all ϵ-reachable states in the start subset:

$$\{q_1\} \text{ (wrong)} \quad \rightarrow \quad \{q_1, q_3\} \text{ (correct).}$$

Epsilon-Closure

Definition

$E(\{q_0\}) = \{q_0\} \cup \{\text{states reachable by } \epsilon \text{ from } q_0\}$.

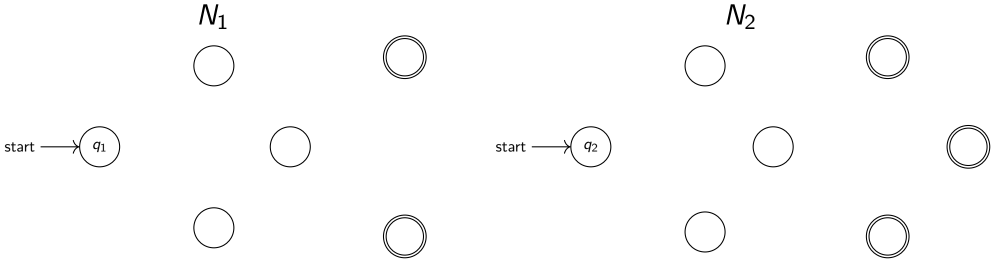
Subset Construction Theorem

Given NFA $M = (Q, \Sigma, \delta, q_0, F)$, define DFA $M' = (Q', \Sigma, \delta', q_0', F')$ with

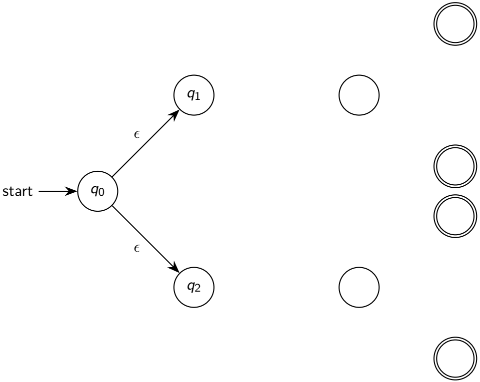
- $Q' = P(Q)$,
- $q_0' = E(\{q_0\})$,
- $F' = \{R \mid R \cap F \neq \emptyset\}$,
- $\delta'(R, a) = \bigcup_{r \in R} E(\delta(r, a))$.

NFA Building Blocks

Take $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ with $\epsilon \notin \Sigma$.



Union Construction (Diagram)



Union Construction (Formal)

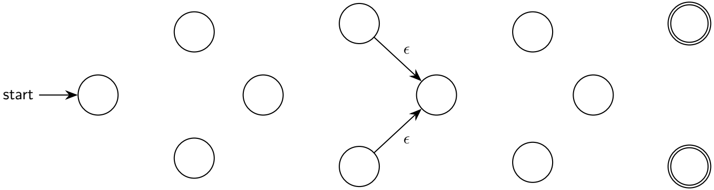
Proposition (Union)

$$N_1 \cup N_2 = (Q, \Sigma, \delta, q_0, F)$$

where

- $Q = Q_1 \cup Q_2 \cup \{q_0\}$,
- $\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1, \\ \delta_2(q, a) & q \in Q_2, \\ \{q_1, q_2\} & q = q_0,\ a = \epsilon, \\ \emptyset & q = q_0,\ a \neq \epsilon, \end{cases}$
- $F = F_1 \cup F_2$.

Concatenation Construction (Diagram)



Concatenation Construction (Formal)

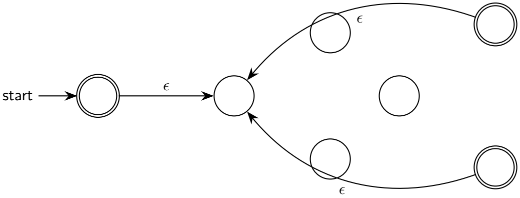
Proposition (Concatenation)

$$N_1 \circ N_2 = (Q, \Sigma, \delta, q_0, F)$$

where

- $Q = Q_1 \cup Q_2$,
- $\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \setminus F_1, \\ \delta_2(q, a) & q \in Q_2, \\ \delta_1(q, \epsilon) \cup \{q_2\} & q \in F_1,\ a = \epsilon, \\ \delta_1(q, \epsilon) & q \in F_1,\ a \neq \epsilon, \end{cases}$
- $q_0 = q_1$,
- $F = F_2$.

Kleene Star Construction (Diagram)



Kleene Star Construction (Formal)

Proposition (Kleene Star)

$$N_1^* = (Q, \Sigma, \delta, q_0, F)$$

where

- $Q = Q_1 \cup \{q_0\}$,
- $\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \setminus F_1, \\ \delta_1(q, a) \cup \{q_1\} & q \in F_1,\ a = \epsilon, \\ \delta_1(q, \epsilon) & q \in F_1,\ a \neq \epsilon, \\ \{q_1\} & q = q_0,\ a = \epsilon, \\ \emptyset & q = q_0,\ a \neq \epsilon, \end{cases}$
- $F = F_1 \cup \{q_0\}$.

Additional Closure Notes

Regular languages are also closed under

- **Intersection**: accept when both components of the product automaton accept.
- **Set Difference**: accept when in F_1 but not F_2.
- **Complement**: $A_1^c = \Sigma^* - A_1$; difference preserves regularity.

Wrap Up

Section Map

Key Takeaways

- Lecture 1 established sets, sequences, proof methods, and deterministic automata.
- Lecture 2 introduced nondeterminism, subset construction, and NFA-based closures.
- Regular languages remain closed under core operations in both DFA and NFA settings.