# Finite Automata and Regular Languages

CENG 280

# Course outline

- Preliminaries: Alphabets and languages
- Regular languages
  - Regular expressions
  - Finite automata: DFA and NFA
  - Finite automata - regular languages
  - Pumping lemma
  - State minimization for DFA
- Context-free languages
- Turing-machines

# Finite Automata and Regular Languages

- NFA closure properties
- RE to NFA
- NFA to RE

# Finite Automata and Regular Languages - Closure properties

## Theorem

*The class of languages accepted by finite automata is closed under*

a. *union,*
b. *concatenation,*
c. *Kleene star,*
d. *complementation,*
e. *intersection.*

# Finite Automata and Regular Languages - Closure properties

## Theorem

*The class of languages accepted by finite automata is closed under*

- a *union,*
- b *concatenation,*
- c *Kleene star,*
- d *complementation,*
- e *intersection.*

Constructive proof: Given $M_1$ and $M_2$, construct $M$ that accepts the corresponding language for each case: a) $L(M_1) \cup L(M_2)$, b) $L(M_1)L(M_2)$, c) $L(M_1)^{\star}$, d) $\overline{L(M_1)}$, e) $L(M_1) \cap L(M_2)$

# Finite Automata and Regular Languages - RE to NFA

**Theorem**

*A language is regular if and only if it is accepted by a finite automaton.*

**Theorem**

*A language is regular if and only if it is accepted by a finite automaton.*

**Proof idea:** *(only if, i.e., RE to NFA)* The class of regular languages is the closure of $\{\{a\} \mid a \in \Sigma\} \cup \{\emptyset\}$ under union, concatenation, and Kleene star. There exist finite automata to accept singletons and empty set. By Theorem 1, every regular language is accepted by some finite automaton.

**Theorem**

*A language is regular if and only if it is accepted by a finite automaton.*

*If, (NFA to RE)* Given a nondeterministic finite automaton $M = (K, \Sigma, \Delta, s, F)$, construct regular expression $R$ such that $L(R) = L(M)$.

# Finite Automata and Regular Languages - NFA to RE

**Theorem**

*A language is regular if and only if it is accepted by a finite automaton.*

*If, (NFA to RE)* Given a nondeterministic finite automaton $M = (K, \Sigma, \Delta, s, F)$, construct regular expression $R$ such that $L(R) = L(M)$.
Let $K = \{q_1, \ldots, q_n\}$ and $s = q_1$. For $i, j = 1, \ldots, n$ and $k = 0, \ldots, n$:
$R(i, j, k)$: the set of strings $w \in \Sigma^\star$ that drives $M$ from $q_i$ to $q_j$ without passing through an intermediate state from $\{q_{k+1}, \ldots, q_n\}$.

# Finite Automata and Regular Languages - NFA to RE

**Theorem**

*A language is regular if and only if it is accepted by a finite automaton.*

*If, (NFA to RE)* Given a nondeterministic finite automaton $M = (K, \Sigma, \Delta, s, F)$, construct regular expression $R$ such that $L(R) = L(M)$.

Let $K = \{q_1, \ldots, q_n\}$ and $s = q_1$. For $i, j = 1, \ldots, n$ and $k = 0, \ldots, n$:
$R(i, j, k)$: the set of strings $w \in \Sigma^\star$ that drives $M$ from $q_i$ to $q_j$ without passing through an intermediate state from $\{q_{k+1}, \ldots, q_n\}$.

$$R(i, j, n) = \{w \in \Sigma^\star \mid (q_i, w) \vdash_M^\star (q_j, e)\}$$

## Theorem

*A language is regular if and only if it is accepted by a finite automaton.*

*If, (NFA to RE)* Given a nondeterministic finite automaton $M = (K, \Sigma, \Delta, s, F)$, construct regular expression $R$ such that $L(R) = L(M)$.

Let $K = \{q_1, \ldots, q_n\}$ and $s = q_1$. For $i, j = 1, \ldots, n$ and $k = 0, \ldots, n$:
$R(i, j, k)$: the set of strings $w \in \Sigma^\star$ that drives $M$ from $q_i$ to $q_j$ without passing through an intermediate state from $\{q_{k+1}, \ldots, q_n\}$.

$$R(i, j, n) = \{w \in \Sigma^\star \mid (q_i, w) \vdash_M^\star (q_j, e)\}$$

$$L(M) = \bigcup_{q_j \in F} R(1, j, n)$$

## Theorem

*A language is regular if and only if it is accepted by a finite automaton.*

*If, (NFA to RE)* Given a nondeterministic finite automaton $M = (K, \Sigma, \Delta, s, F)$, construct regular expression $R$ such that $L(R) = L(M)$.

Let $K = \{q_1, \ldots, q_n\}$ and $s = q_1$. For $i, j = 1, \ldots, n$ and $k = 0, \ldots, n$:
$R(i, j, k)$: the set of strings $w \in \Sigma^\star$ that drives $M$ from $q_i$ to $q_j$ without passing through an intermediate state from $\{q_{k+1}, \ldots, q_n\}$.

$$R(i, j, n) = \{w \in \Sigma^\star \mid (q_i, w) \vdash_M^\star (q_j, e)\}$$

$$L(M) = \bigcup_{q_j \in F} R(1, j, n)$$

Proof by induction that $R(1, j, n)$, thus $L(M)$ is regular

$$R(i, j, k) = R(i, j, k-1) \cup R(i, k, k-1) R(k, k, k-1)^\star R(k, j, k-1)$$

# Finite Automata and Regular Languages - NFA to RE

Special form to simplify RE writing process from NFA

- the automaton has a single final state, $F = \{f\}$,
- the initial state does not have an incoming transition, and
- the final state does not have an outgoing transition.

Every automaton can be converted to an equivalent automaton in this form.

Special form to simplify RE writing process from NFA

- the automaton has a single final state, $F = \{f\}$,
- the initial state does not have an incoming transition, and
- the final state does not have an outgoing transition.

Every automaton can be converted to an equivalent automaton in this form.

RE construction from NFA:

1. Convert FA to an NFA in special form.

Special form to simplify RE writing process from NFA

- the automaton has a single final state, $F = \{f\}$,
- the initial state does not have an incoming transition, and
- the final state does not have an outgoing transition.

Every automaton can be converted to an equivalent automaton in this form.

RE construction from NFA:

1. Convert FA to an NFA in special form.
2. Starting from $k = 0$, iteratively compute $R(i, j, k)$ until $k = n$ (i.e. eliminate $q_k$)

# Finite Automata and Regular Languages - NFA to RE

Special form to simplify RE writing process from NFA

- the automaton has a single final state, $F = \{f\}$,
- the initial state does not have an incoming transition, and
- the final state does not have an outgoing transition.

Every automaton can be converted to an equivalent automaton in this form.

RE construction from NFA:

1. Convert FA to an NFA in special form.
2. Starting from $k = 0$, iteratively compute $R(i, j, k)$ until $k = n$ (i.e. eliminate $q_k$)
3. Return $R(s, f, n)$.

### Example

$L = \{w \in \{0,1\}^\star \mid (\sum_{i=1}^{|w|} w_i) = 3k + 1 \text{ for some } k \in \mathbb{N}\}$