Chapter 2: Finite Automata

A. Deterministic finite automata

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

   K is a finite set of **states**,
   Σ is an alphabet,
   $\delta : K \times \Sigma \to K$ is a
   $s \in K$ is the
   $F \subseteq K$ is the set of

2. Examples and graphical representation.

3. Terminology and notation:
   - Another name:
   - A **configuration** of a DFA is an element of
   - One configuration **yields** another ($(q, w) \vdash_M (q', w')$) iff
   - The reflexive, transitive closure of $\vdash_M$ is denoted $\vdash_M^*$.
   - A string $w \in \Sigma^*$ is **accepted** by M iff $\exists q \in F$ s.t.
   - The set of all strings (i.e., language) accepted by M is denoted $L(M)$.
   - Two DFA M_1 and M_2 are said to be **equivalent** iff

4. Extensions, etc.
   - (Problem 2.1.4) A **finite-state transducer** is a FA that may output a symbol on each step.
   - (Problem 2.1.5) A **2-tape finite automaton** is a FA that accepts pairs of strings.
   - (Problem 2.1.8) For every FA there is an equivalent FA with a minimal number of states.

B. Nondeterminism

1. Idea: allow the transition "function" to be nondeterministic.
2. Motivation:

3. **Definition**. A **nondeterministic finite automaton** (NDFA) is a quintuple $M = (K, \Sigma, \Delta, s, F)$ where

   K is a finite set of **states**,
   Σ is an alphabet,
   $\Delta \subseteq K \times \Sigma^* \times K$ is a
   $s \in K$ is the
   $F \subseteq K$ is the set of

4. Examples.

5. Equivalence of NDFA and DFA.
   - **Theorem.** For each NDFA M there exists an equivalent DFA M''.
   - Example.

C. Languages and finite automata

1. **Theorem 2.4.1.** The class of languages accepted by finite automata is closed under union, concatenation, Kleene star, complementation, and intersection.
2. **Theorem 2.4.2.** There exist "algorithms" for answering the following questions:

3. **Theorem 2.5.1.** A language is regular iff it is accepted by a FA.
4. **Theorem 2.6.1** (Pumping Theorem). If L is an infinite regular language, then there are strings x, $y \neq \varepsilon$, and z s.t.

5. **Theorem** (Problem 2.6.4; strong version of Pumping Theorem). Let $M = (K, \Sigma, \delta, s, F)$ be a DFA. Then for any $w \in L(M)$ where $|w| \geq |K|$, there exist strings x, $y \neq \varepsilon$, z s.t.

6. Proving languages regular or nonregular.
   - To show a language regular, we often "build" a FA for it by using regular expressions and Theorem 2.4.1.
   - Pumping theorems are useful for showing languages to be nonregular.
   - Examples.