Nondeterministic Finite Auomtata

CENG 280

Course outline

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

Nondeterministic Finite Automata

- NFA
- NFA semantics
- Subset construction algorithm

Nondeterministic Finite Automaton

- the successor state "partially" depends on the current state and the input symbol
- several possible combinations of "next states"

Nondeterministic Finite Automaton

- the successor state "partially" depends on the current state and the input symbol
- several possible combinations of "next states"

Example

Consider $(ab \cup aba)^{\star}$

Nondeterministic Finite Automaton

Definition (Nondeterministic finite state automaton)

Nondeterministic finite state 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
- $\Delta \subseteq K \times (\Sigma \cup \{e\}) \times K$ is the transition relation.

$(q, a, p) \in \Delta$ is called a transition of M. (q, e, p) indicates that the machine can pass to state p from state q without reading an input symbol.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**. Note that $\vdash_M$ might not be a function, i.e., there might be several pairs (q', w') (or none at all) such that $(q, w) \vdash_M (q', w')$.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**. Note that $\vdash_M$ might not be a function, i.e., there might be several pairs (q', w') (or none at all) such that $(q, w) \vdash_M (q', w')$.
- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$. $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**. Note that $\vdash_M$ might not be a function, i.e., there might be several pairs (q', w') (or none at all) such that $(q, w) \vdash_M (q', w')$.
- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$. $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if there is a state $f \in F$ such that $(s, w) \vdash_M^\star (f, e)$.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**. Note that $\vdash_M$ might not be a function, i.e., there might be several pairs (q', w') (or none at all) such that $(q, w) \vdash_M (q', w')$.
- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$. $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if there is a state $f \in F$ such that $(s, w) \vdash_M^\star (f, e)$.
- The **language** of M, $L(M)$, is the set of strings accepted by M.

Nondeterministic Finite Automaton

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma \cup \{e\}$ and $(q, a, q') \in \Delta$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**. Note that $\vdash_M$ might not be a function, i.e., there might be several pairs (q', w') (or none at all) such that $(q, w) \vdash_M (q', w')$.
- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$. $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if there is a state $f \in F$ such that $(s, w) \vdash_M^\star (f, e)$.
- The **language** of M, $L(M)$, is the set of strings accepted by M.

Example

Construct an NFA M that recognize
$L = \{w \in \{a, b\}^\star \mid w \text{ contains } ba \text{ or } bba\}$. Show all executions over $babba$. Decide whether $babba \in L(M)$.

Nondeterministic Finite Automaton

Example

Let $\Sigma = \{a_1, \ldots, a_n\}$. Consider
$L = \{w \in \Sigma^\star \mid \text{there is a symbol } a_i \in \Sigma \text{ that does not appear in } w\}$. DFA for n=1,2, NFA for n=3.

Nondeterministic Finite Automaton

- A deterministic finite state automaton is just a special type of nondeterministic finite state automaton.
- We obtain a DFA when Δ defines a function from $K \times \Sigma$ to K.
- In other words, an NFA $M = (K, \Sigma, \Delta, s, F)$is deterministic if there are no transitions of the form (q, e, p) and for each $q \in K$ and $a \in \Sigma$, there exists exactly one $p \in K$ such that $(q, a, p) \in \Delta$.

Nondeterministic Finite Automaton

- A deterministic finite state automaton is just a special type of nondeterministic finite state automaton.
- We obtain a DFA when Δ defines a function from $K \times \Sigma$ to K.
- In other words, an NFA $M = (K, \Sigma, \Delta, s, F)$is deterministic if there are no transitions of the form (q, e, p) and for each $q \in K$ and $a \in \Sigma$, there exists exactly one $p \in K$ such that $(q, a, p) \in \Delta$.
- The class of languages recognized by deterministic finite state automaton is a subset of the class of languages recognized by nondeterministic finite state automaton.

Nondeterministic Finite Automaton

- A deterministic finite state automaton is just a special type of nondeterministic finite state automaton.
- We obtain a DFA when Δ defines a function from $K \times \Sigma$ to K.
- In other words, an NFA $M = (K, \Sigma, \Delta, s, F)$is deterministic if there are no transitions of the form (q, e, p) and for each $q \in K$ and $a \in \Sigma$, there exists exactly one $p \in K$ such that $(q, a, p) \in \Delta$.
- The class of languages recognized by deterministic finite state automaton is a subset of the class of languages recognized by nondeterministic finite state automaton.
- **A nondeterministic finite automaton can always be converted to an equivalent deterministic finite state automaton.**

Definition

Two automaton M_1 and M_2 are said to be **equivalent** when $L(M_1) = L(M_2)$.

Nondeterministic Finite Automaton - Subset construction

Theorem

For each nondeterministic finite automaton, there exists an equivalent deterministic finite automaton.

Nondeterministic Finite Automaton - Subset construction

Theorem

For each nondeterministic finite automaton, there exists an equivalent deterministic finite automaton.

Constructive proof: **subset construction algorithm**

Nondeterministic Finite Automaton - Subset construction

Theorem

For each nondeterministic finite automaton, there exists an equivalent deterministic finite automaton.

Constructive proof: **subset construction algorithm**

Given an NFA $M = (K, \Sigma, s, \Delta, F)$, construct an equivalent DFA $M' = (K', \Sigma, s', \delta, F')$ as follows.

$$E(q) = \{p \in K \mid (q, e) \vdash_M^{\star} (p, e)\}$$

The reflexive transitive closure of $\{q\}$ under the relation $\{(p, r) \mid (p, e, r) \in \Delta\}$

Nondeterministic Finite Automaton - Subset construction

Theorem

For each nondeterministic finite automaton, there exists an equivalent deterministic finite automaton.

Constructive proof: **subset construction algorithm**
Given an NFA $M = (K, \Sigma, s, \Delta, F)$, construct an equivalent DFA $M' = (K', \Sigma, s', \delta, F')$ as follows.

$$E(q) = \{p \in K \mid (q, e) \vdash_M^\star (p, e)\}$$

The reflexive transitive closure of $\{q\}$ under the relation $\{(p, r) \mid (p, e, r) \in \Delta\}$
The DFA is defined as:

$$K' = 2^K, s = E(s)$$

$$F' = \{Q \subseteq K \mid Q \cap F \neq \emptyset\}$$

for each $Q \in K'$ and $a \in \Sigma$

$$\delta(Q, a) = \{E(p) : p \in K, (q, a, p) \in \Delta \text{ for some } q \in Q\}$$

Prove that M and M' are equivalent, show that for any string $w \in \Sigma^\star$

Nondeterministic Finite Automaton - Subset construction

Prove that M and M' are equivalent, show that for any string $w \in \Sigma^\star$

$$(s, w) \vdash_M^\star (f, e) \text{ for some } f \in F \text{ iff}$$
$$(E(s), w) \vdash_{M'}^\star (P, e) \text{ for some } P \text{ containing } f$$

Thus they recognize the same language.

Nondeterministic Finite Automaton - Subset construction

Prove that M and M' are equivalent, show that for any string $w \in \Sigma^\star$

$$(s, w) \vdash_M^\star (f, e) \text{ for some } f \in F \text{ iff}$$
$$(E(s), w) \vdash_{M'}^\star (P, e) \text{ for some } P \text{ containing } f$$

Thus they recognize the same language.
Proof by induction on $|w|$ for

$$(q, w) \vdash_M^\star (p, e) \text{ iff } (E(q), w) \vdash_{M'}^\star (P, e) \text{ for some } P \text{ with } p \in P$$

Nondeterministic Finite Automaton - Examples

Example

Construct a DFA that is equivalent to the given NFA .