CM0081 Formal Languages and Automata
§ 2.3 Non-Deterministic Finite Automata

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2024-1

Preliminaries

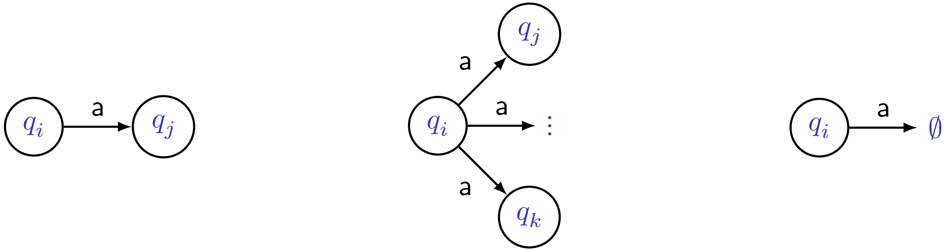
Conventions

- The number and page numbers assigned to chapters, examples, exercises, figures, quotes, sections and theorems on these slides correspond to the numbers assigned in the textbook [Hopcroft, Motwani and Ullman (1979) 2007].
- The natural numbers include the zero, that is, $\mathbb{N} = \{0, 1, 2, \dots\}$.
- The power set of a set A, that is, the set of its subsets, is denoted by $\mathcal{P}A$.

Non-Deterministic Finite Automata

Introduction

- The transition from a state and a symbol can be to: one state, various states or no state.



(continued on next slide)

Non-Deterministic Finite Automata

Introduction (continuation)

- Nondeterminism does not increase the computational power (or expressive power) of finite automata.

Non-Deterministic Finite Automata

Introduction (continuation)

- Nondeterminism does not increase the computational power (or expressive power) of finite automata.
- The processing of an input by a non-deterministic finite automaton can be thought of in terms of guess and verify [Kozen (1997) 2012].

Non-Deterministic Finite Automata

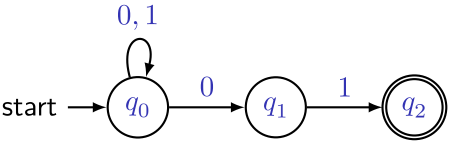
Introduction (continuation)

- Nondeterminism does not increase the computational power (or expressive power) of finite automata.
- The processing of an input by a non-deterministic finite automaton can be thought of in terms of guess and verify [Kozen (1997) 2012].
- Nondeterminism facilitates the design of the automata.

Non-Deterministic Finite Automata

Example

A non-deterministic finite automaton accepting all the binary strings that end in 01.



- q_0: The automaton 'guess' that the final 01 has not begun.
- q_1: The automaton 'guess' that the final 01 has begun.
- q_2: The word ends in 01.

Non-Deterministic Finite Automata

Definition

A **non-deterministic finite automaton** (NFA) is a 5-tuple

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

where

(i) Q is the finite set of states,

(ii) Σ is the alphabet of input symbols,

(iii) $\delta : Q \times \Sigma \to \mathcal{P}Q$ is the transition function,

(iv) $q_0 \in Q$ is the start state,

(v) $F \subseteq Q$ is the set of accepting (or final) states.

Extension of the Transition Function for NFAs

Definition

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA. The **extension of the transition function**, denoted by $\hat{\delta}$, is recursively defined by

$$\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}Q$$

$$\hat{\delta}(q, \varepsilon) = \{q\},$$

$$\hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a).$$

Languages Accepted by NFAs

Recall

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Recall that the language accepted by D was defined by

$$\mathrm{L}(D) := \left\{ w \in \Sigma^* \;\middle|\; \hat{\delta}(q_0, w) \in F \right\}.$$

Languages Accepted by NFAs

Definitions

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA and let $w \in \Sigma^*$ be a string.

(i) The string w is **accepted** by N iff $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

Languages Accepted by NFAs

Definitions

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA and let $w \in \Sigma^*$ be a string.

(i) The string w is **accepted** by N iff $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

(ii) The string w is **rejected** by N iff $\hat{\delta}(q_0, w) \cap F = \emptyset$.

Languages Accepted by NFAs

Definitions

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA and let $w \in \Sigma^*$ be a string.

(i) The string w is **accepted** by N iff $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

(ii) The string w is **rejected** by N iff $\hat{\delta}(q_0, w) \cap F = \emptyset$.

(iii) The **language accepted** by N, denoted $\mathrm{L}(N)$, is the set of strings accepted by N, that is,

$$\mathrm{L}(N) := \left\{ w \in \Sigma^* \;\middle|\; \hat{\delta}(q_0, w) \cap F \neq \emptyset \right\}.$$

Languages Accepted by NFAs

Definitions

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA and let $w \in \Sigma^*$ be a string.

(i) The string w is **accepted** by N iff $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

(ii) The string w is **rejected** by N iff $\hat{\delta}(q_0, w) \cap F = \emptyset$.

(iii) The **language accepted** by N, denoted $\mathrm{L}(N)$, is the set of strings accepted by N, that is,

$$\mathrm{L}(N) := \left\{ w \in \Sigma^* \;\middle|\; \hat{\delta}(q_0, w) \cap F \neq \emptyset \right\}.$$

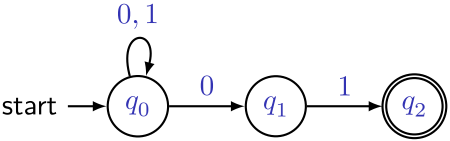
Reading

§ 2.4. An application: Text search.

Languages Accepted by NFAs

Example 2.9

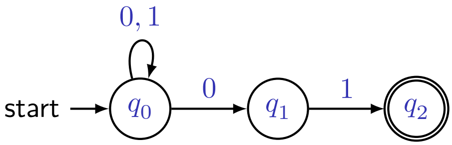
For the NFA of the figure, $\mathrm{L}(N) = \{\, w \in \{0, 1\}^* \mid w$ ends in $01 \,\}$.



Languages Accepted by NFAs

Example 2.9

For the NFA of the figure, $\mathrm{L}(N) = \{\, w \in \{0,1\}^* \mid w$ ends in $01\,\}$.



Sketch of proof

Mutual induction on the following propositions:

$S_0(w)$: $q_0 \in \hat{\delta}(q_0, w)$ for all $w \in \Sigma^*$
$S_1(w)$: $q_1 \in \hat{\delta}(q_0, w) \Leftrightarrow w$ ends in 0
$S_2(w)$: $q_2 \in \hat{\delta}(q_0, w) \Leftrightarrow w$ ends in 01

From $S_2(w)$ and $F = \{q_2\}$ the theorem follows.

Languages Accepted by NFAs

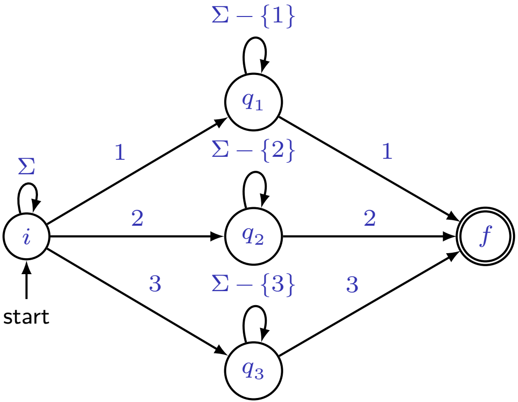
Example (Exercise 2.3.4.a)

NFA accepting the set of strings over $\Sigma = \{1, 2, 3\}$ such that the final digit has appeared before.

Languages Accepted by NFAs

Example (Exercise 2.3.4.a)

NFA accepting the set of strings over $\Sigma = \{1, 2, 3\}$ such that the final digit has appeared before.



- q_i: The automaton 'guess' that the repeated digit is i.

Languages Accepted by NFAs

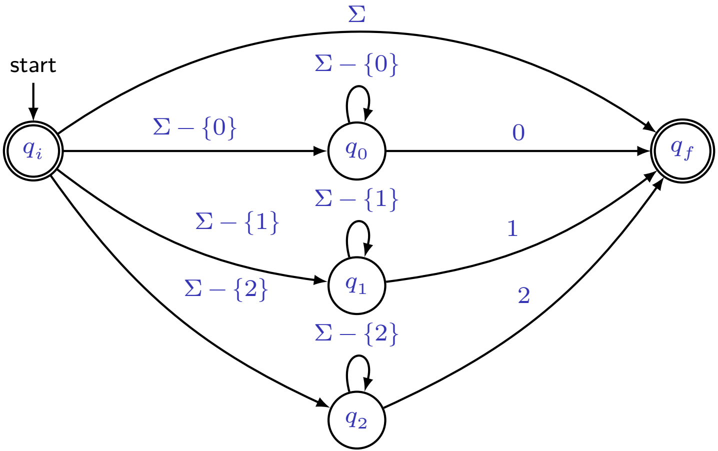
Example (Exercise 2.3.4.b)

NFA accepting the set of strings over $\Sigma = \{0, 1, 2\}$ such that the final digit has not appeared before.

Languages Accepted by NFAs

Example (Exercise 2.3.4.b)

NFA accepting the set of strings over $\Sigma = \{0, 1, 2\}$ such that the final digit has not appeared before.



Languages Accepted by NFAs

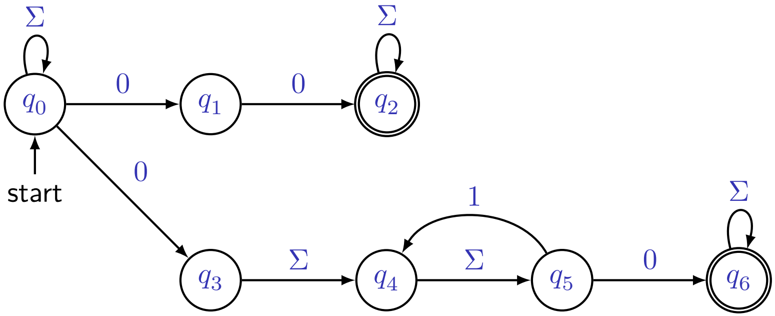
Example (Exercise 2.3.4.c)

NFA accepting the set of strings over $\Sigma = \{0, 1\}$ such that there are two 0's separated by a number of positions that is multiple of 2. Note that 0 is an allowable multiple of 2.

Languages Accepted by NFAs

Example (Exercise 2.3.4.c)

NFA accepting the set of strings over $\Sigma = \{0, 1\}$ such that there are two 0's separated by a number of positions that is multiple of 2. Note that 0 is an allowable multiple of 2.



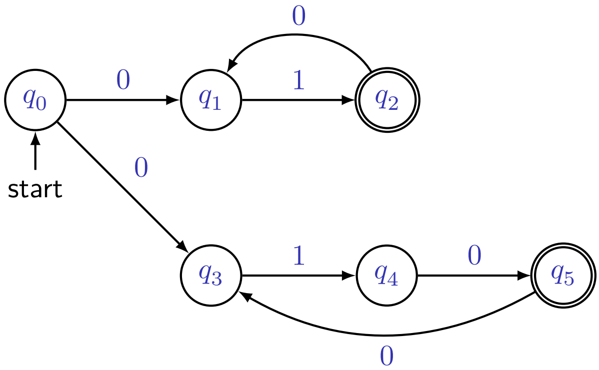
Languages Accepted by NFAs

Example (Exercise 2.5.3.b)

NFA accepting the set of strings that consist of either 01 repeated one or more times or 010 repeated one or more times.

Example (Exercise 2.5.3.b)

NFA accepting the set of strings that consist of either 01 repeated one or more times or 010 repeated one or more times.



The Subset Construction

Construction

Input: A NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$
Output: A DFA $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ where

The Subset Construction

Construction

Input: A NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$
Output: A DFA $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ where

$$Q_D = \mathcal{P}Q_N,$$

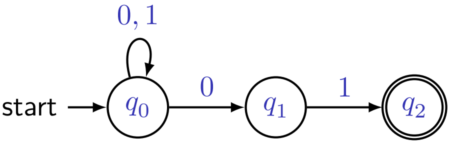
$$F_D = \{\, S \in \mathcal{P}Q_N \mid S \cap F_N \neq \emptyset \,\},$$

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a), \text{ for each } S \in \mathcal{P}Q_N \text{ and } a \in \Sigma.$$

The Subset Construction

Example

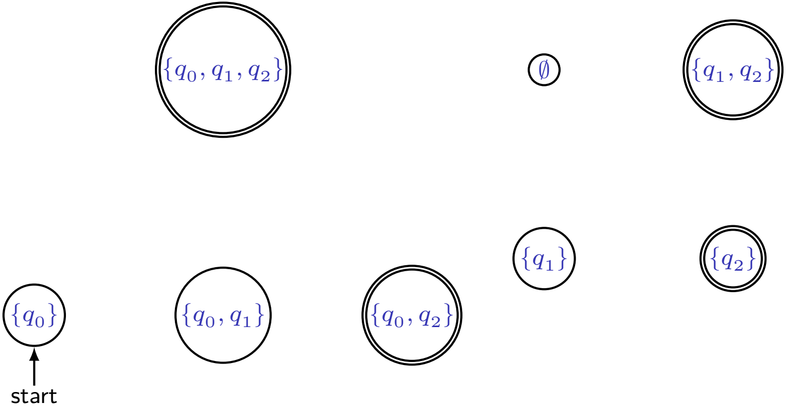
Given the following NFA to build the DFA given by the subset construction.



(continued on next slide)

The Subset Construction

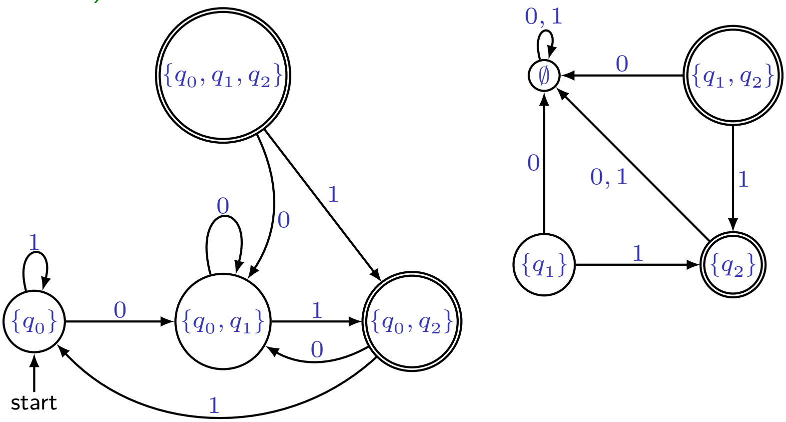
Example (continuation)



(continued on next slide)

The Subset Construction

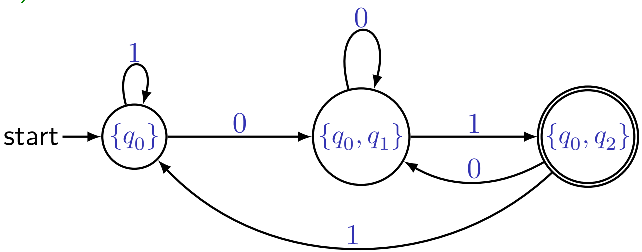
Example (continuation)



(continued on next slide)

The Subset Construction

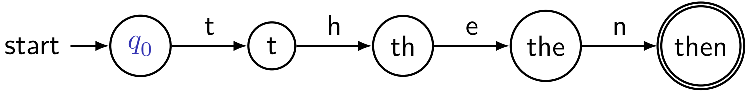
Example (continuation)



The Subset Construction

Example (special case)

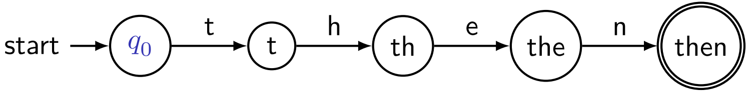
A NFA recognising the word ‘then’:



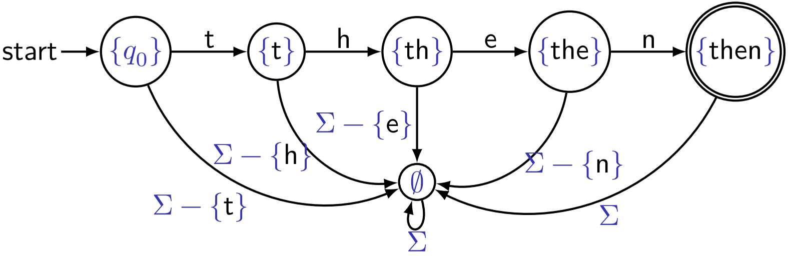
The Subset Construction

Example (special case)

A NFA recognising the word 'then':



The DFA given by the subset construction:



Equivalence of DFAs and NFAs

Theorem 2.11

Let $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ be the DFA constructed from a NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction. Then

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

Equivalence of DFAs and NFAs

Theorem 2.11

Let $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ be the DFA constructed from a NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction. Then

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

and therefore

$$\mathrm{L}(D) = \mathrm{L}(N).$$

Equivalence of DFAs and NFAs

Theorem 2.11

Let $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ be the DFA constructed from a NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction. Then

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

and therefore

$$\mathrm{L}(D) = \mathrm{L}(N).$$

Proof (by structural induction on w)

In the blackboard.

Equivalence of DFAs and NFAs

Theorem 2.12

A language L is accepted by some DFA iff L is accepted by some NFA.

Equivalence of DFAs and NFAs

Theorem 2.12

A language L is accepted by some DFA iff L is accepted by some NFA.

Definition

A language L is **regular** iff exists a finite automaton A (DFA or NFA) such that $L = \mathrm{L}(A)$.

References

- Hopcroft, J. E., Motwani, R. and Ullman, J. D. [1979] (2007). Introduction to Automata Theory, Languages, and Computation. 3rd ed. Pearson Education (cit. on p. 2).
- Kozen, D. C. [1997] (2012). Automata and Computability. Third printing. Undergraduate Texts in Computer Science. Springer. DOI: 10.1007/978-1-4612-1844-9 (cit. on pp. 4–6).