Theory of Computer Science

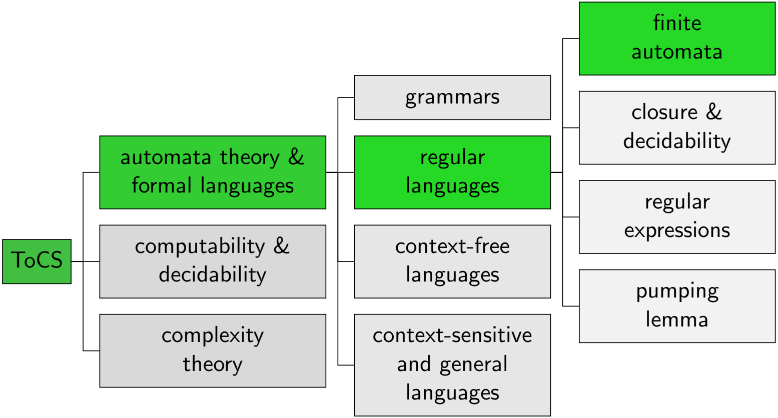
B4. Finite Automata: Characterization

Gabriele Röger

University of Basel

March 13, 2024

Content of the Course



B4.1 Introduction

Finite Automata

Last chapter:

- Two kinds of finite automata: DFAs and NFAs.
- DFAs can be seen as a special case of NFAs.

Questions for today:

- Are there languages that can only be recognized by one kind of finite automaton (but not the other)?
- Can we characterize the languages that DFAs/NFAs can recognize, e.g. within the Chomsky hierarchy?

B4.2 DFAs vs. NFAs

DFAs are No More Powerful than NFAs

Observation

Every language recognized by a DFA is also recognized by an NFA.

We can transform a DFA into an NFA by replacing every transition $\delta(q, a) = q'$ with $\delta(q, a) = \{q'\}$.

Question

DFAs are
no more powerful than NFAs.
But are there languages
that can be recognized
by an NFA but not by a DFA?

Picture courtesy of imagerymajestic / FreeDigitalPhotos.net

NFAs are No More Powerful than DFAs

Theorem (Rabin, Scott)

Every language recognized by an NFA is also recognized by a DFA.

The proof of the theorem is constructive and shows how we can convert an NFA to an equivalent DFA. Let's first have a look at the idea by means of an example (on the blackboard).

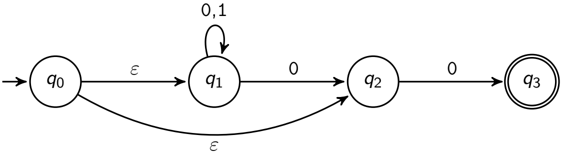
NFAs are No More Powerful than DFAs

Theorem (Rabin, Scott)

Every language recognized by an NFA is also recognized by a DFA.

The proof of the theorem is constructive and shows how we can convert an NFA to an equivalent DFA. Let's first have a look at the idea by means of an example (on the blackboard).

Conversion of an NFA to an Equivalent DFA: Example



NFAs are No More Powerful than DFAs

Theorem (Rabin, Scott)

Every language recognized by an NFA is also recognized by a DFA.

Proof.

For every NFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ we can construct a DFA $M' = \langle Q', \Sigma, \delta', q'_0, F' \rangle$ with $\mathcal{L}(M) = \mathcal{L}(M')$.
Here M' is defined as follows:

- $Q' := \mathcal{P}(Q)$ (the power set of Q)
- $q'_0 := E(q_0)$
- $F' := \{\mathcal{Q} \subseteq Q \mid \mathcal{Q} \cap F \neq \emptyset\}$
- For all $\mathcal{Q} \in Q'$: $\delta'(\mathcal{Q}, a) := \bigcup_{q \in \mathcal{Q}} \bigcup_{q' \in \delta(q,a)} E(q')$

...

NFAs are No More Powerful than DFAs

Theorem (Rabin, Scott)

Every language recognized by an NFA is also recognized by a DFA.

Proof (continued).

For every $w = a_1 a_2 \dots a_n \in \Sigma^*$:

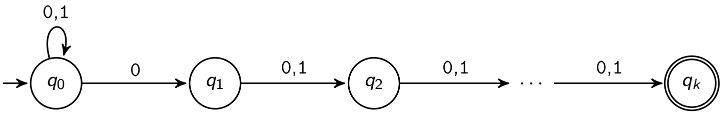
$w \in \mathcal{L}(M)$
iff there is a sequence of states $p_0, p_1, \dots, p_n$ with
$p_0 \in E(q_0)$, $p_n \in F$ and
$p_i \in \bigcup_{q \in \delta(p_{i-1}, a_i)} E(q)$ for all $i \in \{1, \dots, n\}$
iff there is a sequence of subsets $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_n$ with
$\mathcal{Q}_0 = q'_0$, $\mathcal{Q}_n \in F'$ and $\delta'(\mathcal{Q}_{i-1}, a_i) = \mathcal{Q}_i$ for all $i \in \{1, \dots, n\}$
iff $w \in \mathcal{L}(M')$ □

NFAs are More Compact than DFAs

Example

For $k \geq 1$ consider the language
$L_k = \{w \in \{0, 1\}^* \mid |w| \geq k$ and the k-th last symbol of w is $0\}$.

The language L_k can be recognized by an NFA with $k + 1$ states:



There is no DFA with less than 2^k states that recognizes L_k (without proof).

NFAs can often represent languages more compactly than DFAs.

B4.3 Finite Automata vs. Regular Languages

Languages Recognized by DFAs are Regular

Theorem

Every language recognized by a DFA is regular (type 3).

Proof.

Let $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ be a DFA.
We define a regular grammar G with $\mathcal{L}(G) = \mathcal{L}(M)$.

Define $G = \langle Q, \Sigma, R, q_0 \rangle$ where R contains

- a rule $q \to aq'$ for every $\delta(q, a) = q'$, and
- a rule $q \to \varepsilon$ for every $q \in F$.

(We can eliminate forbidden epsilon rules as described in Ch. B2.)

...

Languages Recognized by DFAs are Regular

Theorem

Every language recognized by a DFA is regular (type 3).

Proof (continued).

For every $w = a_1 a_2 \dots a_n \in \Sigma^*$:

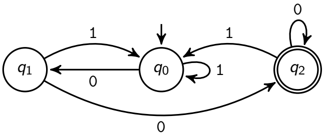
$w \in \mathcal{L}(M)$

iff there is a sequence of states $q'_0, q'_1, \dots, q'_n$ with
$q'_0 = q_0$, $q'_n \in F$ and $\delta(q'_{i-1}, a_i) = q'_i$ for all $i \in \{1, \dots, n\}$

iff there is a sequence of variables $q'_0, q'_1, \dots, q'_n$ with
q'_0 is start variable and we have $q'_0 \Rightarrow a_1 q'_1 \Rightarrow a_1 a_2 q'_2 \Rightarrow \dots \Rightarrow a_1 a_2 \dots a_n q'_n \Rightarrow a_1 a_2 \dots a_n$.

iff $w \in \mathcal{L}(G)$ □

Exercise



Specify a regular grammar that generates the language recognized by this DFA.

Question

Is the inverse true as well:
for every regular language, is there a DFA that recognizes it? That is, are the languages recognized by DFAs exactly the regular languages?

Yes!
We will prove this via a detour.

Picture courtesy of imagerymajestic / FreeDigitalPhotos.net

Regular Grammars are No More Powerful than NFAs

Theorem

For every regular grammar G there is an NFA M with $\mathcal{L}(G) = \mathcal{L}(M)$.

Proof.

Let $G = \langle V, \Sigma, R, S\rangle$ be a regular grammar.
Define NFA $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ with

$$Q = V \cup \{X\}, \quad X \notin V$$

$$q_0 = S$$

$$F = \begin{cases} \{S, X\} & \text{if } S \to \varepsilon \in R \\ \{X\} & \text{if } S \to \varepsilon \notin R \end{cases}$$

$B \in \delta(A, a)$ if $A \to aB \in R$

$X \in \delta(A, a)$ if $A \to a \in R$

...

Regular Grammars are No More Powerful than NFAs

Theorem

For every regular grammar G there is an NFA M with $\mathcal{L}(G) = \mathcal{L}(M)$.

Proof (continued).

For every $w = a_1 a_2 \dots a_n \in \Sigma^*$ with $n \geq 1$:

$w \in \mathcal{L}(G)$

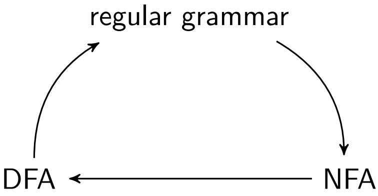
iff there is a sequence on variables $A_1, A_2, \dots, A_{n-1}$ with
$S \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \dots \Rightarrow a_1 a_2 \dots a_{n-1} A_{n-1} \Rightarrow a_1 a_2 \dots a_n$.

iff there is a sequence of variables $A_1, A_2, \dots, A_{n-1}$ with
$A_1 \in \delta(S, a_1), A_2 \in \delta(A_1, a_2), \dots, X \in \delta(A_{n-1}, a_n)$.

iff $w \in \mathcal{L}(M)$.

Case $w = \varepsilon$ is also covered because $S \in F$ iff $S \to \varepsilon \in R$. □

Finite Automata and Regular Languages



In particular, this implies:

Corollary

$\mathcal{L}$ regular $\iff$ $\mathcal{L}$ is recognized by a DFA.
$\mathcal{L}$ regular $\iff$ $\mathcal{L}$ is recognized by an NFA.

Summary

- DFAs and NFAs recognize the same languages.
- These are exactly the regular languages.