CM0081 Formal Languages and Automata
§ 3.2 Finite Automata and Regular Expressions

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2024-1

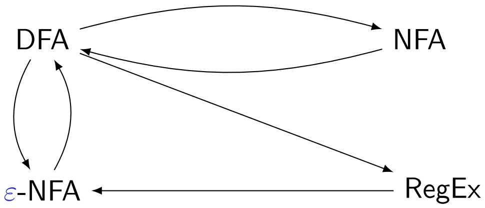
Preliminaries

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 $\mathcal{P}A$.

Introduction

Equivalences



From Finite Automata to Regular Expressions

Theorem 3.4

If $L = \mathrm{L}(D)$ for some DFA D, then there is a regular expression R such that $L = \mathrm{L}(R)$.

From Finite Automata to Regular Expressions

Proof (by induction on the number of states of the automaton)

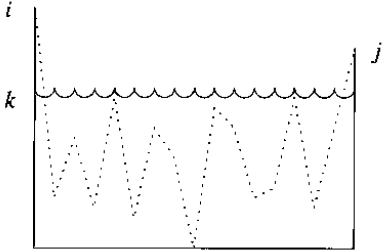
- Let the states of D be $\{1, 2, \dots, n\}$ with 1 the start state.

†Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.2].

From Finite Automata to Regular Expressions

Proof (by induction on the number of states of the automaton)

- Let the states of D be $\{1, 2, \dots, n\}$ with 1 the start state.
- R_{ij}^k: Regular expression describing the set of labels of all paths in D from state i to state j such that the path has no intermediate node whose number is greater than k.†



†Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.2].

From Finite Automata to Regular Expressions

Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i \neq j$

From Finite Automata to Regular Expressions

Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i \neq j$

- No transition from state i to state j

$$R^0_{ij} = \emptyset$$

From Finite Automata to Regular Expressions

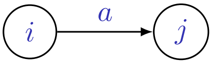
Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i \neq j$

- No transition from state i to state j

$$R_{ij}^{0} = \emptyset$$

- One transition from state i to state j



$$R_{ij}^{0} = \boldsymbol{a}$$

From Finite Automata to Regular Expressions

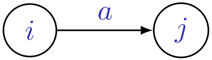
Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i \neq j$

- No transition from state i to state j

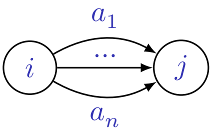
$$R_{ij}^0 = \emptyset$$

- One transition from state i to state j



$$R_{ij}^0 = \boldsymbol{a}$$

- Various transitions from state i to state j



$$R_{ij}^0 = \boldsymbol{a}_1 + \cdots + \boldsymbol{a_n}$$

From Finite Automata to Regular Expressions

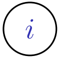
Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i = j$

Proof (continuation)

Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i = j$

- No loops

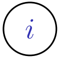


$$R_{ii}^{0} = \varepsilon$$

Proof (continuation)

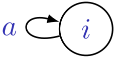
Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i = j$

- No loops



$$R_{ii}^{0} = \varepsilon$$

- One loop



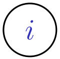
$$R_{ii}^{0} = \varepsilon + \boldsymbol{a}$$

From Finite Automata to Regular Expressions

Proof (continuation)

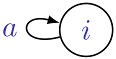
Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i = j$

- No loops



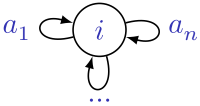
$$R_{ii}^{0} = \varepsilon$$

- One loop



$$R_{ii}^{0} = \varepsilon + \boldsymbol{a}$$

- Various loops



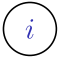
$$R_{ii}^{0} = \varepsilon + \boldsymbol{a}_1 + \cdots + \boldsymbol{a}_n$$

From Finite Automata to Regular Expressions

Proof (continuation)

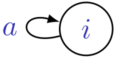
Basis step: Proof for $k = 0$ (i.e. no intermediate states) and $i = j$

- No loops



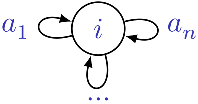
$$R^0_{ii} = \varepsilon$$

- One loop



$$R^0_{ii} = \varepsilon + \boldsymbol{a}$$

- Various loops



$$R^0_{ii} = \varepsilon + \boldsymbol{a}_1 + \cdots + \boldsymbol{a}_n$$

Question

From Finite Automata to Regular Expressions

Proof (continuation)

Inductive step: Proof for k

Inductive hypothesis R_{ij}^{k-1}: Path from state i to state j that goes through no state higher than $k-1$.

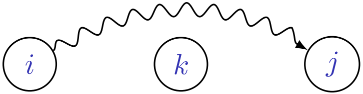
From Finite Automata to Regular Expressions

Proof (continuation)

Inductive step: Proof for k

Inductive hypothesis R_{ij}^{k-1}: Path from state i to state j that goes through no state higher than $k-1$.

- The path does not go through state k at all



$$R_{ij}^{k} = R_{ij}^{k-1}$$

From Finite Automata to Regular Expressions

Proof (continuation)

Inductive step: Proof for k

Inductive hypothesis R_{ij}^{k-1}: Path from state i to state j that goes through no state higher than $k-1$.

From Finite Automata to Regular Expressions

Proof (continuation)

Inductive step: Proof for k

Inductive hypothesis R_{ij}^{k-1}: Path from state i to state j that goes through no state higher than $k-1$.

- The path goes through state k at least once

$$i \rightsquigarrow k \rightsquigarrow k \rightsquigarrow k \rightsquigarrow j$$

$$R_{ij}^{k} = R_{ik}^{k-1}(R_{kk}^{k-1})^{*}R_{kj}^{k-1}$$

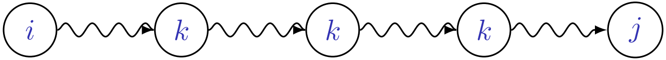
From Finite Automata to Regular Expressions

Proof (continuation)

Inductive step: Proof for k

Inductive hypothesis R_{ij}^{k-1}: Path from state i to state j that goes through no state higher than $k-1$.

- The path goes through state k at least once



$$R_{ij}^{k} = R_{ik}^{k-1}(R_{kk}^{k-1})^{*}R_{kj}^{k-1}$$

From the previous cases:

$$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^{*}R_{kj}^{k-1}.$$

From Finite Automata to Regular Expressions

Proof (continuation)

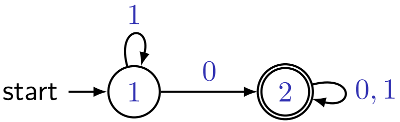
Given that the states of D are $\{1, 2, \dots, n\}$ with 1 the start state, then

$$\mathrm{L}(D) = \mathrm{L}(R^n_{1f_1} + \cdots + R^n_{1f_m}) \text{ with } f_i \in F.$$

■

Example

To convert the DFA D to a regular expression.

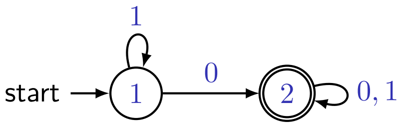


$\mathrm{L}(D) = \{\, w \in \{0,1\}^* \mid w$ has at least one $0 \,\}$.

(continued on next slide)

Example (continuation)

- R_{ij}^{0}



R_{ij}^{0}	Regexp
R_{11}^{0}	$\varepsilon + \mathbf{1}$
R_{12}^{0}	$\mathbf{0}$
R_{21}^{0}	$\emptyset$
R_{22}^{0}	$\varepsilon + \mathbf{0} + \mathbf{1}$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

- $R^1_{ij} = R^0_{ij} + R^0_{i1}(R^0_{11})^* R^0_{1j}$

R^0_{ij}	Regexp
R^0_{11}	$\varepsilon + \mathbf{1}$
R^0_{12}	$\mathbf{0}$
R^0_{21}	$\emptyset$
R^0_{22}	$\varepsilon + \mathbf{0} + \mathbf{1}$

R^1_{ij}	Regexp
R^1_{11}	$(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$
R^1_{12}	$\mathbf{0} + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*\mathbf{0}$
R^1_{21}	$\emptyset + \emptyset(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$
R^1_{22}	$\varepsilon + \mathbf{0} + \mathbf{1} + \emptyset(\varepsilon + \mathbf{1})^*\mathbf{0}$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

- Some simplifications for regular expressions

  Let M and N be regular expression variables.

$$
\begin{aligned}
(\varepsilon + M)^* &= M^* \\
(\varepsilon + M)M^* &= M^* \\
M + N^*M &= N^*M \\
M\emptyset = \emptyset M &= \emptyset && \text{($\emptyset$ is the annihilator for concatenation)} \\
M + \emptyset = \emptyset + M &= M && \text{($\emptyset$ is the identity for union)}
\end{aligned}
$$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

- $R^1_{ij} = R^0_{ij} + R^0_{i1}(R^0_{11})^* R^0_{1j}$

R^1_{ij}	Regexp	Simplified
R^1_{11}	$(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$	$\mathbf{1}^*$
R^1_{12}	$\mathbf{0} + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*\mathbf{0}$	$\mathbf{1}^*\mathbf{0}$
R^1_{21}	$\emptyset + \emptyset(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$	$\emptyset$
R^1_{22}	$\varepsilon + \mathbf{0} + \mathbf{1} + \emptyset(\varepsilon + \mathbf{1})^*\mathbf{0}$	$\varepsilon + \mathbf{0} + \mathbf{1}$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

- $R^2_{ij} = R^1_{ij} + R^1_{i2}(R^1_{22})^* R^1_{2j}$

R^1_{ij}	Regexp
R^1_{11}	$\mathbf{1}^*$
R^1_{12}	$\mathbf{1}^*\mathbf{0}$
R^1_{21}	$\emptyset$
R^1_{22}	$\varepsilon + \mathbf{0} + \mathbf{1}$

R^2_{ij}	Regexp
R^2_{11}	$\mathbf{1}^* + \mathbf{1}^*\mathbf{0}(\varepsilon + \mathbf{0} + \mathbf{1})^*\emptyset$
R^2_{12}	$\mathbf{1}^*\mathbf{0} + \mathbf{1}^*\mathbf{0}(\varepsilon + \mathbf{0} + \mathbf{1})^*(\varepsilon + \mathbf{0} + \mathbf{1})$
R^2_{21}	$\emptyset + (\varepsilon + \mathbf{0} + \mathbf{1})(\varepsilon + \mathbf{0} + \mathbf{1})^*\emptyset$
R^2_{22}	$\varepsilon + \mathbf{0} + \mathbf{1} + (\varepsilon + \mathbf{0} + \mathbf{1})(\varepsilon + \mathbf{0} + \mathbf{1})^*(\varepsilon + \mathbf{0} + \mathbf{1})$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

- $R^2_{ij} = R^1_{ij} + R^1_{i2}(R^1_{22})^* R^1_{2j}$

R^2_{ij}	Regexp	Simplified
R^2_{11}	$\mathbf{1}^* + \mathbf{1}^*\mathbf{0}(\varepsilon + \mathbf{0} + \mathbf{1})^*\emptyset$	$\mathbf{1}^*$
R^2_{12}	$\mathbf{1}^*\mathbf{0} + \mathbf{1}^*\mathbf{0}(\varepsilon + \mathbf{0} + \mathbf{1})^*(\varepsilon + \mathbf{0} + \mathbf{1})$	$\mathbf{1}^*\mathbf{0}(\mathbf{0} + \mathbf{1})^*$
R^2_{21}	$\emptyset + (\varepsilon + \mathbf{0} + \mathbf{1})(\varepsilon + \mathbf{0} + \mathbf{1})^*\emptyset$	$\emptyset$
R^2_{22}	$\varepsilon + \mathbf{0} + \mathbf{1} + (\varepsilon + \mathbf{0} + \mathbf{1})(\varepsilon + \mathbf{0} + \mathbf{1})^*(\varepsilon + \mathbf{0} + \mathbf{1})$	$(\mathbf{0} + \mathbf{1})^*$

(continued on next slide)

From Finite Automata to Regular Expressions

Example (continuation)

The regular expression equivalent to the automaton D is

$$R_{12}^{2} = \mathbf{1}^{*}\mathbf{0}(\mathbf{0} + \mathbf{1})^{*}.$$

From Regular Expressions to Finite Automata

Theorem 3.7

Every language defined by a regular expression is also defined by a finite automaton.

From Regular Expressions to Finite Automata

Theorem 3.7

Every language defined by a regular expression is also defined by a finite automaton.

Proof (by induction)

For each regular expression E we construct an ε-NFA A such that $\mathrm{L}(E) = \mathrm{L}(A)$ with:

(i) exactly one accepting state,

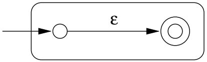
(ii) no arcs into the initial state and

(iii) no arcs out of the accepting state.

From Regular Expressions to Finite Automata

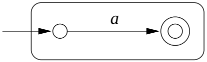
Proof (continuation)

Basis step: Automata for (a) $E = \varepsilon$, (b) $E = \emptyset$ and (c) $E = \boldsymbol{a}$.[†]



(a)

(b)



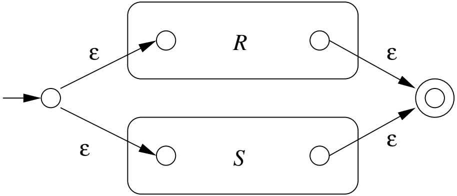
(c)

[†] Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.16].

From Regular Expressions to Finite Automata

Proof (continuation)

Inductive step: Automaton for $E = R + S$.[†]

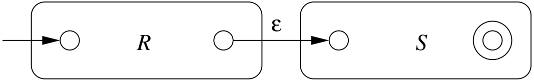


[†]Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.17].

From Regular Expressions to Finite Automata

Proof (continuation)

Inductive step: Automaton for $E = RS$.[†]

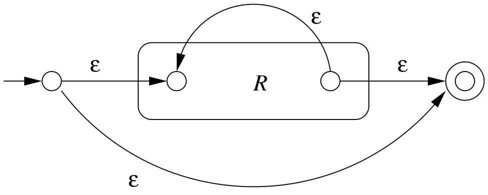


[†]Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.17].

From Regular Expressions to Finite Automata

Proof (continuation)

Inductive step: Automaton for $E = R^*$.[†]



■

[†] Figure from Hopcroft, Motwani and Ullman [(1979) 2007, Fig. 3.17].

From Regular Expressions to Finite Automata

Example

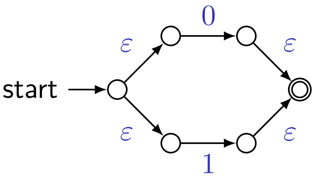
Convert the regular expression $E = (\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1})$ to an ε-NFA A such that $\mathrm{L}(E) = \mathrm{L}(A)$.

From Regular Expressions to Finite Automata

Example

Convert the regular expression $E = (\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1})$ to an ε-NFA A such that $\mathrm{L}(E) = \mathrm{L}(A)$.

1. ε-NFA for $\mathbf{0} + \mathbf{1}$:

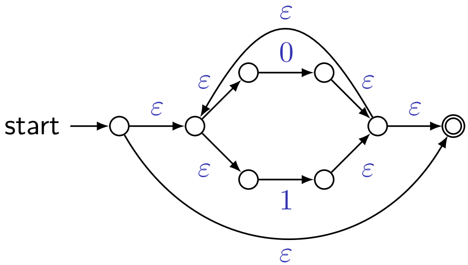


(continued on next slide)

From Regular Expressions to Finite Automata

Example (continuation)

2. ε-NFA for $(\mathbf{0}+\mathbf{1})^*$:

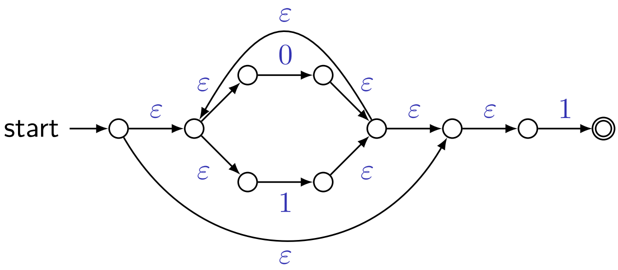


(continued on next slide)

From Regular Expressions to Finite Automata

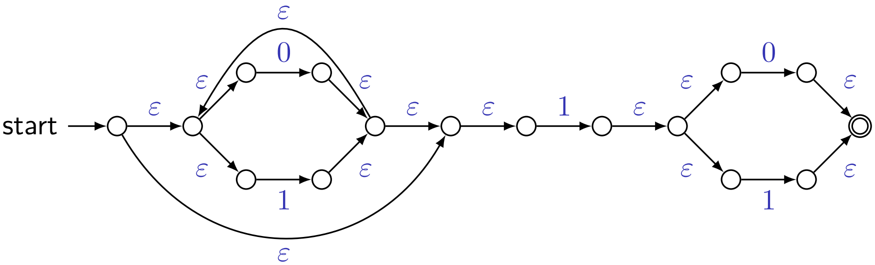
Example (continuation)

3. ε-NFA for $(\mathbf{0}+\mathbf{1})^*1$:



Example (continuation)

4. ε-NFA for $(\mathbf{0}+\mathbf{1})^*1(\mathbf{0}+\mathbf{1})$:



From Regular Expressions to Finite Automata

Exercise 3.2.7

There are some simplifications to the constructions of Theorem 3.7, where we converted a regular expression to an ε-NFA. Here are three:

1) For the union operator, instead of creating new start and accepting states, merge the two start states into one state with all the transitions of both start states. Likewise, merge the two accepting states, having all transitions to either go to the merged state instead.

2) For the concatenation operator, merge the accepting state of the first automaton with the start state of the second.

3) For the closure operator, simply add ε-transitions from the accepting state to the start state and vice-versa.

(continued on next slide)

From Regular Expressions to Finite Automata

Exercise 3.2.7 (continuation)

Each of these simplifications, by themselves, still yield a correct construction; that is, the resulting ε-NFA for any regular expression accepts the language of the expression. Which subsets of changes 1), 2) and 3) may be made to the construction together, while still yielding a correct automaton for every regular expression?

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 pp. 2, 5, 6, 32–35).