CSE-217: Theory of Computation

REGULAR Expression

Md Jakaria

Lecturer
Department of Computer Science and Engineering
Military Institute of Science and Technology

August 15, 2019

Regular Expression

REGULAR EXPRESSIONS

- In arithmetic, we can use the operations + and $\times$ to build up expressions such as $(5+3)\times 4$.
- Similarly, we can use the regular operations to build up expressions describing languages.
- These are called regular expressions.

REGULAR EXPRESSIONS

- In arithmetic, we can use the operations + and $\times$ to build up expressions such as $(5+3) \times 4$.
- Similarly, we can use the regular operations to build up expressions describing languages.
- These are called regular expressions.
- An example is:

$$(0 \cup 1)0^*$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

$$(\{0\} \cup \{1\})\{0\}^*$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

$$(\{0\} \cup \{1\})\{0\}^*$$

$$\{0, 1\}\{0\}^*$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

$$(\{0\} \cup \{1\})\{0\}^*$$

$$\{0, 1\}\{0\}^*$$

$$\{0, 1\}\circ\{0\}^*$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

$$(\{0\} \cup \{1\})\{0\}^*$$

$$\{0, 1\}\{0\}^*$$

$$\{0, 1\} \circ \{0\}^*$$

$$\{0, 1\} \circ \{\epsilon, 0, 00, 000, \ldots\}$$

REGULAR EXPRESSIONS

$$(0 \cup 1)0^*$$

$$(\{0\} \cup \{1\})\{0\}^*$$

$$\{0, 1\}\{0\}^*$$

$$\{0, 1\} \circ \{0\}^*$$

$$\{0, 1\} \circ \{\epsilon, 0, 00, 000, \ldots\}$$

$$\{0, 00, 000, \ldots, 1, 10, 100, \ldots\}$$

REGULAR EXPRESSIONS example

- $(0 \cup 1)^*$

REGULAR EXPRESSIONS example

- $(0 \cup 1)^*$
- Σ^* where $\Sigma = \{0, 1\}$

REGULAR EXPRESSIONS example

- $(0 \cup 1)^*$
- Σ^* where $\Sigma = \{0, 1\}$
- Σ^*1 where $\Sigma = \{0, 1\}$

REGULAR EXPRESSIONS example

- $(0 \cup 1)^*$
- Σ^* where $\Sigma = \{0, 1\}$
- Σ^*1 where $\Sigma = \{0, 1\}$
- $(0^*\Sigma) \cup (\Sigma^*1)$ where $\Sigma = \{0, 1\}$

REGULAR EXPRESSIONS

DEFINITION 1.52

Say that R is a ***regular expression*** if R is

1. a for some a in the alphabet Σ,
2. ε,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$, where R_1 and R_2 are regular expressions, or
6. (R_1^*), where R_1 is a regular expression.

In items 1 and 2, the regular expressions a and ε represent the languages $\{a\}$ and $\{\varepsilon\}$, respectively. In item 3, the regular expression $\emptyset$ represents the empty language. In items 4, 5, and 6, the expressions represent the languages obtained by taking the union or concatenation of the languages R_1 and R_2, or the star of the language R_1, respectively.

Regular Expression Example

REGULAR EXPRESSIONS shorthand

- $R^+ \equiv RR^*$
- $R^+ \cup \{\epsilon\} \equiv R^*$
- R^k be the concatenation of k R's
- $L(R)$ to be the language of R.

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

1. 0^*10^*

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

1. $0^*10^* = \{w \mid w \text{ contains a single } 1\}$
2. $\Sigma^*1\Sigma^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

1. $0^*10^* = \{w \mid w \text{ contains a single } 1\}$
2. $\Sigma^*1\Sigma^* = \{w \mid w \text{ has at least one } 1\}$
3. $\Sigma^*001\Sigma^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

1. $0^*10^* =$ {w | w contains a single 1}
2. $\Sigma^*1\Sigma^*$ = {w | w has at least one 1}
3. $\Sigma^*001\Sigma^*$ = {w | w contains the string 001 as a substring}.
4. $1^*(01^+)^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

1. $0^*10^* =$ {w | w contains a single 1}
2. $\Sigma^*1\Sigma^*$ = {w | w has at least one 1}
3. $\Sigma^*001\Sigma^*$ = {w | w contains the string 001 as a substring}.
4. $1^*(01^+)^*$ = {w | every 0 in w is followed by at least one 1}

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

5 $(\Sigma\Sigma)^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

5 $(\Sigma\Sigma)^* = \{w \mid w \text{ is a string of even length}\}$

6 $(\Sigma\Sigma\Sigma)^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

5 $(\Sigma\Sigma)^* =$ {w | w is a string of even length}

6 $(\Sigma\Sigma\Sigma)^*$ = {w | the length of w is a multiple of 3}

7 $01 \cup 10$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

5 $(\Sigma\Sigma)^* = \{w \mid w$ is a string of even length$\}$

6 $(\Sigma\Sigma\Sigma)^* = \{w \mid$ the length of w is a multiple of 3$\}$

7 $01 \cup 10 = \{01, 10\}$.

8 $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

5 $(\Sigma\Sigma)^* = $ {w | w is a string of even length}

6 $(\Sigma\Sigma\Sigma)^*$ = {w | the length of w is a multiple of 3}

7 $01 \cup 10$ = {01, 10}.

8 $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$ = {w | w starts and ends with the same symbol}

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

9 $(0 \cup \epsilon)1^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

9 $(0 \cup \epsilon)1^* = 01^* \cup 1^*$.

10 $(0 \cup \epsilon)(1 \cup \epsilon)$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

9 $(0 \cup \epsilon)1^* = 01^* \cup 1^*$.

10 $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$

11 $1^*\varnothing$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

9 $(0 \cup \epsilon)1^* = 01^* \cup 1^*$.

10 $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$

11 $1^*\varnothing = \varnothing$.

12 $\varnothing^*$

REGULAR EXPRESSIONS Example

In the following instances, we assume that the alphabet Σ is $\{0, 1\}$.

9 $(0 \cup \epsilon)1^* = 01^* \cup 1^*$.

10 $(0 \cup \epsilon)(1 \cup \epsilon)$ = $\{\epsilon$, 0, 1, 01$\}$

11 $1^*\varnothing = \varnothing$.

12 $\varnothing^* = \{\epsilon\}$

EQUIVALENCE WITH FINITE AUTOMATA

EQUIVALENCE WITH FINITE AUTOMATA

- Regular expressions and finite automata are equivalent in their descriptive power.
- Any regular expression can be converted into a finite automaton

EQUIVALENCE WITH FINITE AUTOMATA

Theorem

A language is regular if and only if some regular expression describes it.

EQUIVALENCE WITH FINITE AUTOMATA

Lemma

If a language is described by a regular expression, then it is regular.

EQUIVALENCE WITH FINITE AUTOMATA

PROOF IDEA

1. Say that we have a regular expression R describing some language A.
2. We show how to convert R into an NFA recognizing A.
3. If an NFA recognizes A then A is regular.

EQUIVALENCE WITH FINITE AUTOMATA

PROOF

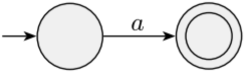
Let's convert R into an NFA N. We consider the six cases in the formal definition of regular expressions.

EQUIVALENCE WITH FINITE AUTOMATA

1. $R = a$ for some $a \in \Sigma$ Then $L(R) = \{a\}$, and the following NFA recognizes $L(R)$.

EQUIVALENCE WITH FINITE AUTOMATA

1 $R = a$ for some $a \in \Sigma$ Then $L(R) = \{a\}$, and the following NFA recognizes $L(R)$.



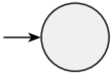
Formally, $N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q2\})$, where we describe δ by saying that $\delta(q_1, a) = \{q_2\}$ and that $\delta(r, b) = \varnothing$ for $r \neq q1$ or $b \neq a$.

EQUIVALENCE WITH FINITE AUTOMATA

3 $R = \emptyset$. Then $L(R) = \emptyset$, and the following NFA recognizes $L(R)$.

EQUIVALENCE WITH FINITE AUTOMATA

3 $R = \emptyset$. Then $L(R) = \emptyset$, and the following NFA recognizes $L(R)$.



Formally, $N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$, where $\delta(r, b) = \emptyset$ for any r and b.

EQUIVALENCE WITH FINITE AUTOMATA

4 $R = R_1 \cup R_2$.

5 $R = R_1 \circ R_2$.

6 $R = R_1^*$.

For the last three cases, we use the constructions given in the proofs that the class of regular languages is closed under the regular operations. In other words, we construct the NFA for R from the NFAs for R_1 and R_2 (or just R_1 in case 6) and the appropriate closure construction.

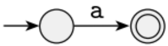
Example

EQUIVALENCE WITH FINITE AUTOMATA

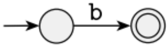
1. Build an NFA from the regular expression $(ab \cup b)^*$

EQUIVALENCE WITH FINITE AUTOMATA

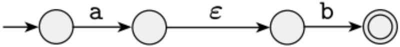
a



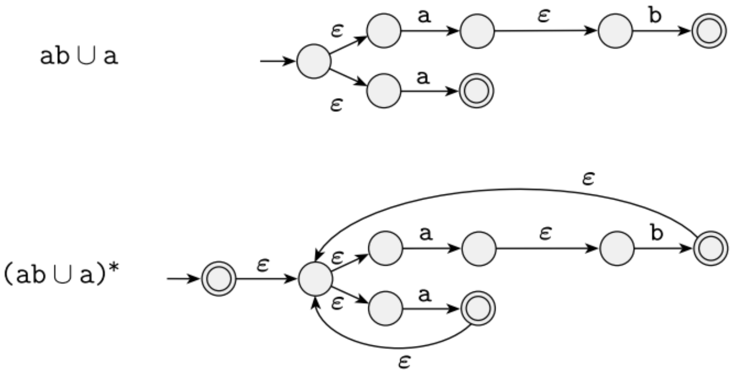
b



ab



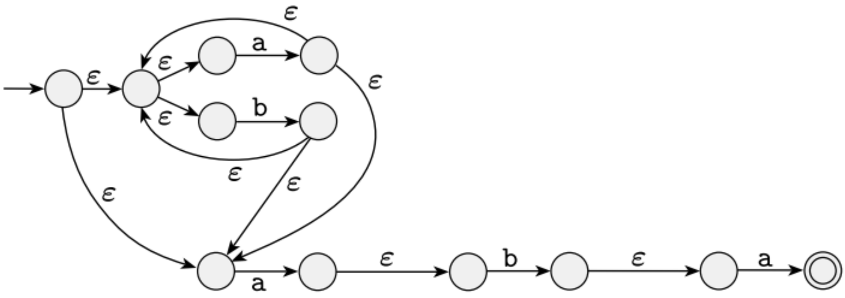
EQUIVALENCE WITH FINITE AUTOMATA



EQUIVALENCE WITH FINITE AUTOMATA

2 Build an NFA from the regular expression $(a \cup b)^{*}aba$

EQUIVALENCE WITH FINITE AUTOMATA



EQUIVALENCE WITH FINITE AUTOMATA

Theorem

A language is regular if and only if some regular expression describes it.

EQUIVALENCE WITH FINITE AUTOMATA

Lemma

If a language is regular, then it is described by a regular expression.

EQUIVALENCE WITH FINITE AUTOMATA

Proof

EQUIVALENCE WITH FINITE AUTOMATA

Proof

Homework!