COS210 - Theoretical Computer Science
Finite Automata and Regular Languages (Part 6)

Regular Expressions: Motivation

Regular languages:

$$L = \{w : \text{start with 0 or 1, second symbol is 0, end with zero or more 1's}\}$$

Operations on languages:

- union: $\cup$
- concatenation: ww'
- star: $*$

Regular Expressions: Motivation

Regular languages:

$$L = \{w : \underbrace{\underbrace{\text{start with 0 or 1}}_{\text{union}}, \text{second symbol is 0}, \underbrace{\text{end with zero or more 1's}}_{\text{star}}}_{\text{concatenation}}\}$$

Operations on languages:

- union: $\cup$
- concatenation: ww'
- star: $*$

Regular Expressions: Motivation

Regular languages:

$$L = \{w : \underbrace{\underbrace{\text{start with 0 or 1}}_{0 \cup 1}, \underbrace{\text{second symbol is 0}}_{0}, \underbrace{\text{end with zero or more 1's}}_{1^*}}_{(0 \cup 1)01^*}\}$$

Operations on languages:

- union: $\cup$
- concatenation: ww'
- star: *

Regular expressions:

$$(0 \cup 1)01^*$$

allow to describe regular languages formally

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

0 0

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w$ contains exactly two 0s$\}$

$$1^*01^*01^*$$

- $L_2 = \{w \in \{0,1\}^* : w$ the first and last symbols of w are equal$\}$

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

$$1^*01^*01^*$$

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

0 0

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

$$1^*01^*01^*$$

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

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

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: * (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

$$1^*01^*01^*$$

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

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

Regular Expressions

Operations on languages:

- union: $\cup$ (used as an OR)
- concatenation: ww'
- star: $*$ (zero or more occurrences of a pattern)

Which regular expressions that accept the following languages?

- $L_1 = \{w \in \{0,1\}^* : w \text{ contains exactly two 0s}\}$

$$1^*01^*01^*$$

- $L_2 = \{w \in \{0,1\}^* : w \text{ the first and last symbols of } w \text{ are equal}\}$

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

Regular Expressions: Formal Definition

Definition (Regular Expression)

Let Σ be an alphabet, then

1. ϵ is a regular expression,
2. $\emptyset$ is a regular expression,
3. each $a \in \Sigma$ is a regular expression,
4. if R_1 and R_2 are regular expressions, then $R_1 \cup R_2$ is a regular expression,
5. if R_1 and R_2 are regular expressions, then $R_1 R_2$ is a regular expression,
6. if R is a regular expression, then R^* is a regular expression.

brackets () can be used to give explicit precedence of construction

implicit order of precedence is: *brackets, star, concatenation, union*

Regular Expressions: Formal Definition

Definition (Regular Expression)

Let Σ be an alphabet, then

1. ϵ is a regular expression,
2. $\emptyset$ is a regular expression,
3. each $a \in \Sigma$ is a regular expression,
4. if R_1 and R_2 are regular expressions, then $R_1 \cup R_2$ is a regular expression,
5. if R_1 and R_2 are regular expressions, then $R_1 R_2$ is a regular expression,
6. if R is a regular expression, then R^* is a regular expression.

brackets () can be used to give explicit precedence of construction

implicit order of precedence is: *brackets, star, concatenation, union*

Regular Expressions: Construction using Formal Definition

Consider:

$$0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1 \quad \text{over} \quad \Sigma = \{0, 1\}$$

- 0 is regular
- 1 is regular
- $0 \cup 1$ is regular
- $(0 \cup 1)^*$ is regular
- $0(0 \cup 1)^*$ is regular
- $\ldots$
- $0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1$ is regular

Language of a Regular Expression

Definition (Language described by a Regular expression)

let Σ be an alphabet, then

1. the regular expression ϵ describes the language $\{\epsilon\}$,
2. the regular expression $\emptyset$ describes the language $\emptyset$,
3. for each $a \in \Sigma$, the regular expression a describes the language $\{a\}$,
4. let R_1 and R_2 be the regular expressions that describe the languages L_1 and L_2 respectively, then $R_1 \cup R_2$ describes the language $L_1 \cup L_2$,
5. let R_1 and R_2 be the regular expressions that describe the languages L_1 and L_2 respectively, then $R_1 R_2$ describes the language $L_1 L_2$,
6. let R be the regular expression that describes the language A, then the regular expression R^* describes the language A^*.

Language of a Regular Expression

Definition (Language described by a Regular expression)

let Σ be an alphabet, then

1. the regular expression ϵ describes the language $\{\epsilon\}$,
2. the regular expression $\emptyset$ describes the language $\emptyset$,
3. for each $a \in \Sigma$, the regular expression a describes the language $\{a\}$,
4. let R_1 and R_2 be the regular expressions that describe the languages L_1 and L_2 respectively, then $R_1 \cup R_2$ describes the language $L_1 \cup L_2$,
5. let R_1 and R_2 be the regular expressions that describe the languages L_1 and L_2 respectively, then $R_1 R_2$ describes the language $L_1 L_2$,
6. let R be the regular expression that describes the language A, then the regular expression R^* describes the language A^*.

Language of a Regular Expression

Examples:

- $(0 \cup \epsilon)(1 \cup \epsilon)$ describes the language

$$\{01, 0, 1, \epsilon\}$$

Language of a Regular Expression

Examples:

- $(0 \cup \epsilon)(1 \cup \epsilon)$ describes the language

$$\{01, 0, 1, \epsilon\}$$

- $(0 \cup \epsilon)1^*$ describes the language

$$\{0, 01, 011, 0111, \ldots, \epsilon, 1, 11, 111, \ldots\}$$

Language of a Regular Expression

Examples:

- $(0 \cup \epsilon)(1 \cup \epsilon)$ describes the language

$$\{01, 0, 1, \epsilon\}$$

- $(0 \cup \epsilon)1^*$ describes the language

$$\{0, 01, 011, 0111, \ldots, \epsilon, 1, 11, 111, \ldots\}$$

- $1^*\emptyset$ describes the language

$$\emptyset$$

A = {ε, 1, 11, …} (language of 1*)
B = ∅ (language of ∅)

AB = {ww' : w ∈ A and w' ∈ B} = ∅
because there is no w' to build any ww'

Language of a Regular Expression

Examples:

- $(0 \cup \epsilon)(1 \cup \epsilon)$ describes the language

$$\{01, 0, 1, \epsilon\}$$

- $(0 \cup \epsilon)1^*$ describes the language

$$\{0, 01, 011, 0111, \ldots, \epsilon, 1, 11, 111, \ldots\}$$

- $1^*\emptyset$ describes the language

$$\emptyset$$

- $\emptyset^*$ describes the language

$$\{\epsilon\}$$

For any regular language A,
The empty symbol ε is contained in A*

Regular Expressions: Equivalence

Definition (Equivalence of Regular Expressions)

Let R_1 and R_2 be regular expressions and let L_1 and L_2 be the languages described by them, respectively.

$$\text{If } L_1 = L_2, \text{ then } R_1 = R_2.$$

Example:

- $(0 \cup \epsilon)1^*$ and $01^* \cup 1^*$ are equivalent expressions
- as both describe the language $L = \{0, 01, 011, 0111, \ldots, \epsilon, 1, 11, 111, \ldots\}$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $\boxed{R_1\emptyset = \emptyset R_1 = \emptyset}$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $\boxed{R_1\epsilon = \epsilon R_1 = R_1}$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $\boxed{R_1 \cup \emptyset = \emptyset \cup R_1 = R_1}$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $\boxed{R_1 \cup R_1 = R_1}$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $\boxed{R_1 \cup R_2 = R_2 \cup R_1}$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $\boxed{R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3}$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $\boxed{(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3}$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $\boxed{R_1(R_2R_3) = (R_1R_2)R_3}$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Regular Expressions: Standard Equivalences

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $\boxed{R_1\epsilon = \epsilon R_1 = R_1}$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $\boxed{(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3}$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

$(0 \cup \epsilon)1^*$ and $01^* \cup 1^*$ are equivalent expressions

Equivalence of Regular Expressions and Regular Languages

The following theorem holds:

Theorem (1)

Let L be a language, then:

$$L \text{ is regular} \quad \Leftrightarrow \quad \text{there exists a regular expression } R \text{ that describes } L$$

$\Longleftarrow$:

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

$\Longrightarrow$:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

Equivalence of Regular Expressions and Regular Languages

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

Proof by induction and construction:
Base cases:

- $R = \epsilon$
- $R = \emptyset$
- $R = a$ where $a \in \Sigma$

for each R, construct M that accepts language described by R

Inductive steps:
Assume R_1, R_2 describe regular languages L_1, L_2

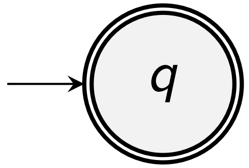
- $R = R_1 \cup R_2$
- $R = R_1 R_2$
- $R = R_1^*$

for each R, construct M that accepts language described by R

Equivalence of Regular Expressions and Regular Languages

First base case:

- $R = \epsilon$
- language described by R is $L = \{\epsilon\}$
- construction of M that accepts L:

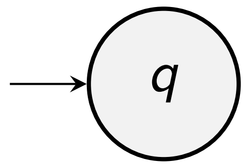


$\Rightarrow$ the language $L = \{\epsilon\}$ is regular

Equivalence of Regular Expressions and Regular Languages

Second base case:

- $R = \emptyset$
- language described by R is $L = \emptyset$
- construction of M that accepts L:

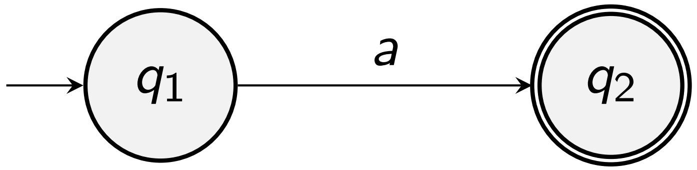


$\Rightarrow$ the language $L = \emptyset$ is regular

Equivalence of Regular Expressions and Regular Languages

Third base case:

- $R = a$ where $a \in \Sigma$
- language described by R is $L = \{a\}$
- construction of M that accepts L:



$\Rightarrow$ the language $L = \{a\}$ is regular

Equivalence of Regular Expressions and Regular Languages

First inductive step (union):

- Assume R_1, R_2 describe regular languages L_1, L_2 (hypothesis)
- Let $R = R_1 \cup R_2$
- L_1, L_2 regular $\Rightarrow L_1 \cup L_2$ regular (closure of union)

$\Rightarrow$ R describes a regular language

Equivalence of Regular Expressions and Regular Languages

Second inductive step (concatenation):

- Assume R_1, R_2 describe regular languages L_1, L_2 (hypothesis)
- Let $R = R_1 R_2$
- L_1, L_2 regular $\Rightarrow L_1 L_2$ regular (closure of concatenation)

$\Rightarrow$ R describes a regular language

Equivalence of Regular Expressions and Regular Languages

Third inductive step (star):

- Assume R_1 describes regular language A (hypothesis)
- Let $R = R_1^*$
- A regular $\Rightarrow A^*$ regular (closure of star)

$\Rightarrow$ R describes a regular language

We can conclude that every regular expression constructed by arbitrary combinations of union, concatenation and star describes a regular language. $\square$

(proof of Theorem 1B in the next lecture)

Equivalence of Regular Expressions and Regular Languages

The following theorem holds:

Theorem (1)

Let L be a language, then:

$$L \text{ is regular} \\ \Leftrightarrow \\ \text{there exists a regular expression } R \text{ that describes } L$$

$\Longleftarrow$:

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

$\Longrightarrow$:

Theorem (1B)

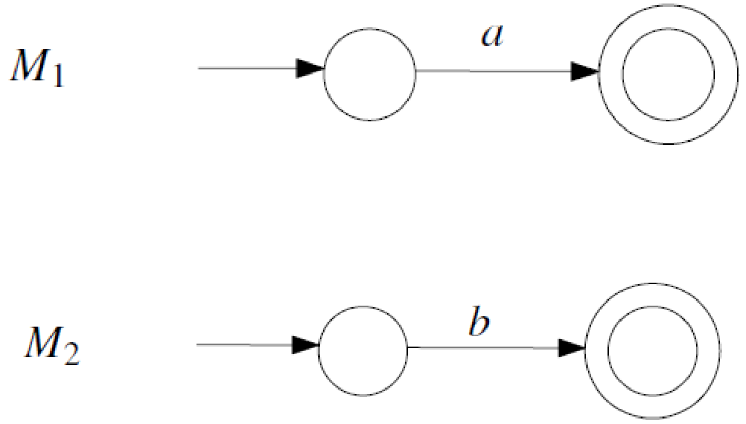
For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

Construction of NFA from Regular Expression: Example

$$R = (ab \cup a)^* \quad \text{over} \quad \Sigma = \{a, b\}$$

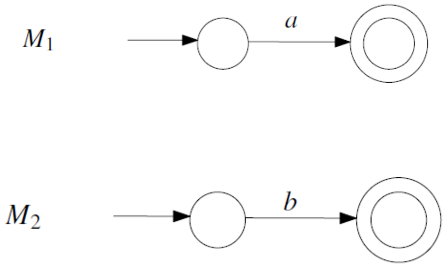
Step 1:

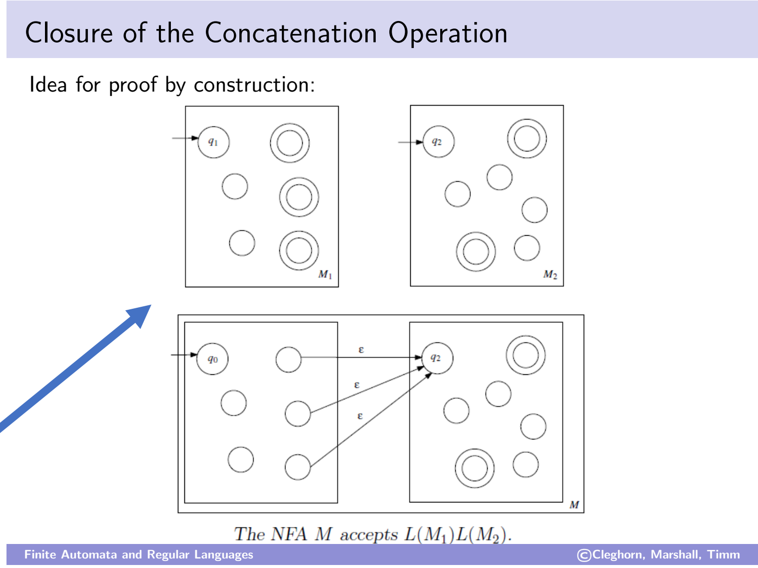
- consider the *atomic sub-expressions* a, b
- build an NFA for each sub-expressions:



Construction of NFA from Regular Expression: Example

$$R = (ab \cup a)^* \quad \text{over} \quad \Sigma = \{a, b\}$$




Closure of the Concatenation Operation

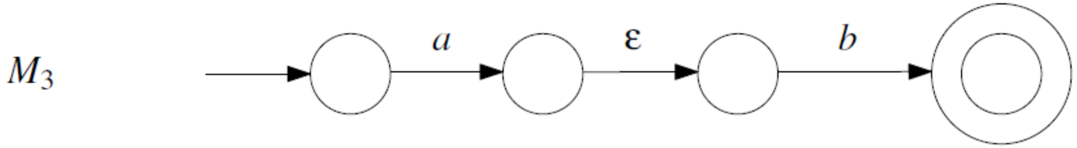
Idea for proof by construction:

The NFA M accepts $L(M_1)L(M_2)$.

Finite Automata and Regular Languages ©Cleghorn, Marshall, Timm

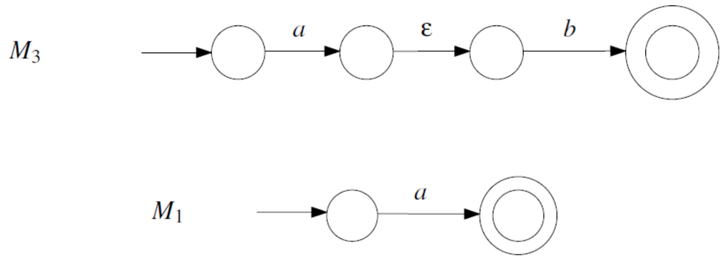
Step 2:

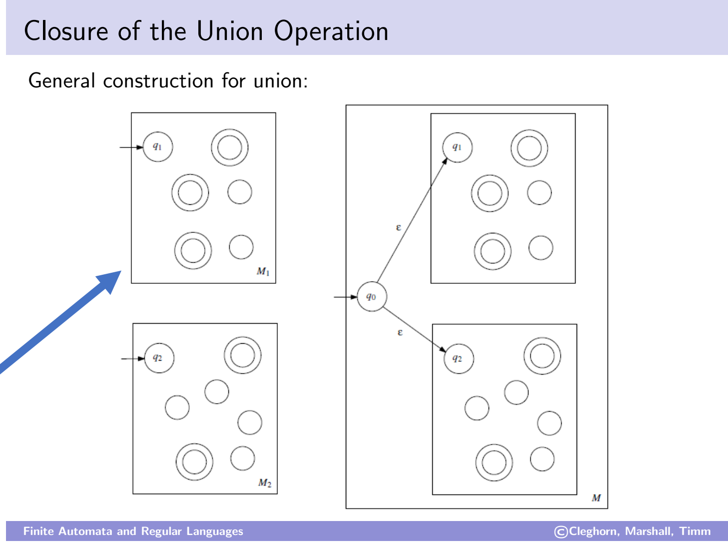
- consider the concatenation *ab*
- use the construction from proof of *closure of concatenation* to build NFA for *ab*:



Construction of NFA from Regular Expression: Example

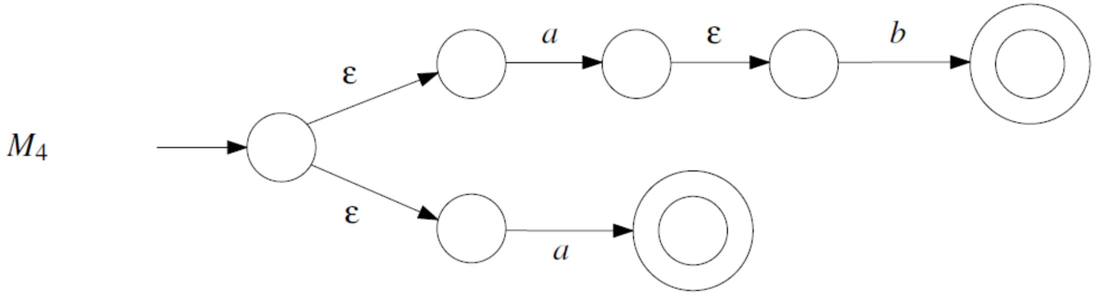
$$R = (ab \cup a)^* \quad \text{over} \quad \Sigma = \{a, b\}$$





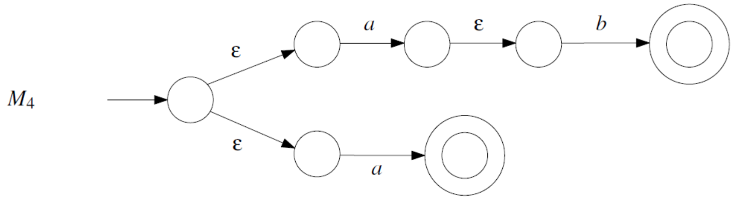
Step 3:

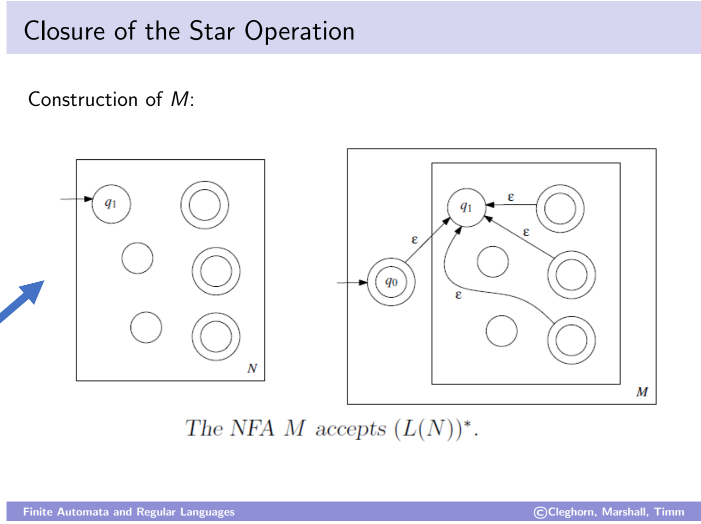
- consider the union $ab \cup a$
- use the construction from proof of *closure of union* to build NFA for $ab \cup a$:



Construction of NFA from Regular Expression: Example

$$R = (ab \cup a)^* \quad \text{over} \quad \Sigma = \{a, b\}$$





Step 4:

- consider the star $(ab \cup a)^*$
- use the construction from proof of *closure of star* to build NFA for $(ab \cup a)^*$.