CMPE 322/327 - Theory of Computation

Week 4: Pattern Matching & Regular Expressions

Burak Ekici

March 14-18, 2022

Outline

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet
  3. $\Delta : Q \times \Sigma \to 2^Q$: transition function

Definitions

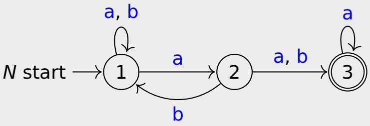
- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet
  3. $\Delta : Q \times \Sigma \to 2^Q$: transition function
  4. $S \subseteq Q$: set of start states

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet
  3. $\Delta : Q \times \Sigma \to 2^Q$: transition function
  4. $S \subseteq Q$: set of start states
  5. $F \subseteq Q$: final (accept) states

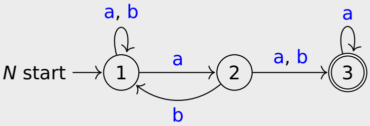
Example

$N = (Q, \Sigma, \Delta, S, F)$



Example

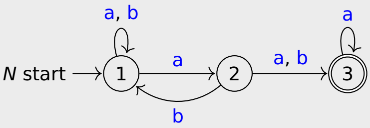
$N = (Q, \Sigma, \Delta, S, F)$



1. $Q = \{1, 2, 3\}$

Example

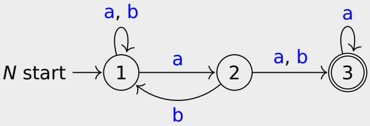
$N = (Q, \Sigma, \Delta, S, F)$



1. $Q = \{1, 2, 3\}$
2. $\Sigma = \{a, b\}$

Example

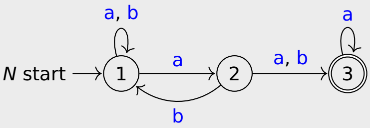
$N = (Q, \Sigma, \Delta, S, F)$



1. $Q = \{1, 2, 3\}$
2. $\Sigma = \{a, b\}$
3. $\Delta : Q \times \Sigma \to 2^Q$

Example

$N = (Q, \Sigma, \Delta, S, F)$

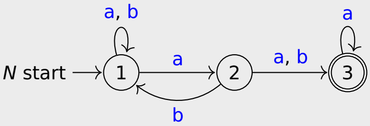


1. $Q = \{1, 2, 3\}$
2. $\Sigma = \{a, b\}$
3. $\Delta : Q \times \Sigma \to 2^Q$

Δ	a	b
1	$\{1, 2\}$	$\{1\}$
2	$\{3\}$	$\{1, 3\}$
3	$\{3\}$	$\emptyset$

Example

$N = (Q, \Sigma, \Delta, S, F)$

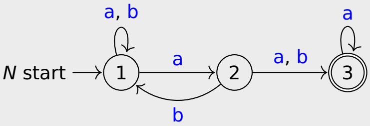


1. $Q = \{1, 2, 3\}$
2. $\Sigma = \{a, b\}$
3. $\Delta : Q \times \Sigma \to 2^Q$
4. $S = \{1\}$

Δ	a	b
1	$\{1, 2\}$	$\{1\}$
2	$\{3\}$	$\{1, 3\}$
3	$\{3\}$	$\varnothing$

Example

$N = (Q, \Sigma, \Delta, S, F)$



1. $Q = \{1, 2, 3\}$
2. $\Sigma = \{a, b\}$
3. $\Delta : Q \times \Sigma \to 2^Q$
4. $S = \{1\}$
5. $F = \{3\}$

Δ	a	b
1	$\{1, 2\}$	$\{1\}$
2	$\{3\}$	$\{1, 3\}$
3	$\{3\}$	$\varnothing$

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet
  3. $\Delta : Q \times \Sigma \to 2^Q$: transition function
  4. $S \subseteq Q$: set of start states
  5. $F \subseteq Q$: final (accept) states
- $\widehat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}(A, \varepsilon) := A \qquad \widehat{\Delta}(A, xa) := \bigcup_{q \in \widehat{\Delta}(A,x)} \Delta(q, a)$$

Definitions

- nondeterministic finite automaton (NFA) is quintuple $N = (Q, \Sigma, \Delta, s, F)$ with
  1. Q : finite set of states
  2. Σ : input alphabet
  3. $\Delta : Q \times \Sigma \to 2^Q$: transition function
  4. $S \subseteq Q$: set of start states
  5. $F \subseteq Q$: final (accept) states
- $\widehat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}(A, \varepsilon) := A \qquad \widehat{\Delta}(A, xa) := \bigcup_{q \in \widehat{\Delta}(A,x)} \Delta(q, a)$$

- string $x \in \Sigma^*$ is accepted by N if $\widehat{\Delta}(S, x) \cap F \neq \varnothing$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that

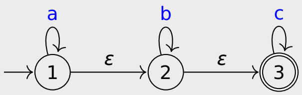
Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$

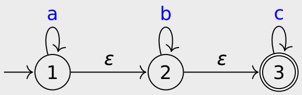
Example



Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^* \}$

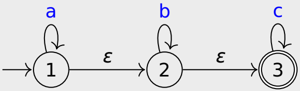
Example



Definitions

- NFA with ϵ-transitions (NFA$_\epsilon$) is sextuple $N = (Q, \Sigma, \epsilon, \Delta, S, F)$ such that
  1. $\epsilon \notin \Sigma$
  2. $N_\epsilon = (Q, \Sigma \cup \{\epsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\epsilon\}$
- ϵ-closure of set $A \subseteq Q$ is defined as $C_\epsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\epsilon}(A, x) \mid x \in \{\epsilon\}^* \}$

Example



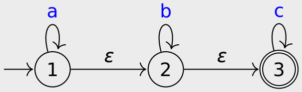
$$C_\epsilon(\{1\}) = \{1, 2, 3\}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^* \}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{ C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x) \}$$

Example



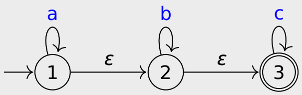
$$C_\varepsilon(\{1\}) \quad = \quad \{1, 2, 3\}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^* \}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{ C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x) \}$$

Example



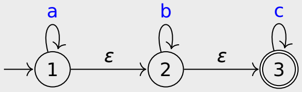
$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= \end{aligned}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^* \}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad\qquad \widehat{\Delta}_N(A, xa) = \bigcup \{ C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x) \}$$

Example



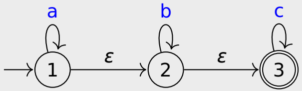
$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= C_\varepsilon(\{1\}) \end{aligned}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{\widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^*\}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x)\}$$

Example



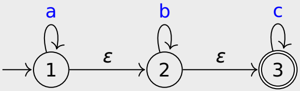
$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= C_\varepsilon(\Delta(1, b)) \cup C_\varepsilon(\Delta(2, b)) \cup C_\varepsilon(\Delta(3, b)) \end{aligned}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{\widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^*\}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x)\}$$

Example



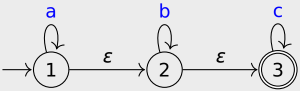
$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= C_\varepsilon(\varnothing) \cup C_\varepsilon(\{2\}) \cup C_\varepsilon(\varnothing) \end{aligned}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{ \widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^* \}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{ C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x) \}$$

Example



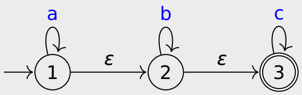
$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= \varnothing \cup \{2, 3\} \cup \varnothing \end{aligned}$$

Definitions

- NFA with ε-transitions (NFA$_\varepsilon$) is sextuple $N = (Q, \Sigma, \varepsilon, \Delta, S, F)$ such that
  1. $\varepsilon \notin \Sigma$
  2. $N_\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, \Delta, S, F)$ is NFA over alphabet $\Sigma \cup \{\varepsilon\}$
- ε-closure of set $A \subseteq Q$ is defined as $C_\varepsilon(A) = \bigcup \{\widehat{\Delta}_{N_\varepsilon}(A, x) \mid x \in \{\varepsilon\}^*\}$
- $\widehat{\Delta}_N : 2^Q \times \Sigma^* \to 2^Q$ is inductively defined by

$$\widehat{\Delta}_N(A, \varepsilon) = C_\varepsilon(A) \qquad \widehat{\Delta}_N(A, xa) = \bigcup \{C_\varepsilon(\Delta(q, a)) \mid q \in \widehat{\Delta}_N(A, x)\}$$

Example



$$\begin{aligned} C_\varepsilon(\{1\}) &= \{1, 2, 3\} \\ \widehat{\Delta}(\{1\}, b) &= \{2, 3\} \end{aligned}$$

Theorem

every set accepted by NFA is regular

Theorem

every set accepted by NFA is regular

Theorem

every set accepted by NFA_ε is regular

Theorem

every set accepted by NFA is regular

Theorem

every set accepted by NFA_ϵ is regular

Theorem

regular sets are effectively closed under concatenation

Theorem

every set accepted by NFA is regular

Theorem

every set accepted by NFA_ε is regular

Theorem

regular sets are effectively closed under concatenation

Theorem

regular sets are effectively closed under asterate

Outline

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line
$	matches end of line

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line
$	matches end of line
c	matches character c

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line	.	matches any character
$	matches end of line		
c	matches character c		

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line	.	matches any character
$	matches end of line	[abc]	matches a or b or c
c	matches character c		

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

^	matches beginning of line	`.`	matches any character
$	matches end of line	`[abc]`	matches a or b or c
c	matches character c	`[a-zA-Z]`	matches any letter

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

`^`	matches beginning of line	`.`	matches any character
`$`	matches end of line	`[abc]`	matches a or b or c
`c`	matches character c	`[a-zA-Z]`	matches any letter

Example

`grep "0" file` returns lines containing 0

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, `sed`, `perl`

Some Patterns

`^`	matches beginning of line	`.`	matches any character
`$`	matches end of line	`[abc]`	matches a or b or c
`c`	matches character c	`[a-zA-Z]`	matches any letter

Example

`grep "0" file`	returns lines containing 0
`grep "0$" file`	returns lines ending with 0

Applications of Regular expressions: `grep`

- `grep foo file` returns lines in file containing pattern foo
- basis for more powerful tools like awk, sed, `perl`

Some Patterns

^	matches beginning of line	`.`	matches any character
$	matches end of line	`[abc]`	matches a or b or c
c	matches character c	`[a-zA-Z]`	matches any letter

Example

`grep "0" file`	returns lines containing 0
`grep "0$" file`	returns lines ending with 0
`grep "b.g" file`	returns lines containing e.g. bag, big, bug, buggy

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$

-

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$

-

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$

-

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ

-

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
$@$	Σ^*

-

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
@	Σ^*

-

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
$@$	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$
-

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
@	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$
-

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
$@$	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$
β^*	$L(\beta)^*$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$
-

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
@	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$
-

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\epsilon}$	$\{\epsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
$@$	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$
$\sim \beta$	$\sim L(\beta) = \Sigma^* - L(\beta)$

Pattern matching is important for

- lexical analysis of programs
- scripting languages (Perl, Ruby)
- search engines (Google Code Search)
- DNA analysis

Definitions

- pattern is string α that represents set of strings $L(\alpha) \subseteq \Sigma^*$

atomic pattern α	$L(\alpha)$
$\mathbf{a} \in \Sigma$	$\{a\}$
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$
$\boldsymbol{\varnothing}$	$\varnothing$
$\#$	Σ
$@$	Σ^*

compound pattern α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$
$\sim \beta$	$\sim L(\beta) = \Sigma^* - L(\beta)$

- string $x \in \Sigma^*$ matches pattern α if $x \in L(\alpha)$

Example

pattern	matched string
$@a@a@a@$	strings containing at least 3 occurrences of a

Example

pattern	matched string
$@a@a@a@$	strings containing at least 3 occurrences of a
$@a@b@$	strings containing a followed later by b

Example

pattern	matched string
$@a@a@a@$	strings containing at least 3 occurrences of a
$@a@b@$	strings containing a followed later by b
$\# \cap \sim a$	single letters except a

Example

pattern	matched string
$@a@a@a@$	strings containing at least 3 occurrences of a
$@a@b@$	strings containing a followed later by b
$\# \cap \sim a$	single letters except a
$(\# \cap \sim a)^*$	strings without a

Questions

- how difficult is pattern matching?

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$\varepsilon \equiv \sim(\#@) \equiv \emptyset^*$$

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$\begin{aligned} \boldsymbol{\varepsilon} &\equiv \sim(\#@) \equiv \boldsymbol{\emptyset}^* \\ @ &\equiv \#^* \end{aligned}$$

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$\begin{aligned} \varepsilon &\equiv \sim(\#@) \equiv \emptyset^* \\ @ &\equiv \#^* \\ \alpha^+ &\equiv \alpha\alpha^* \end{aligned}$$

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$
\begin{array}{lll}
\varepsilon & \equiv & \sim(\#@) \equiv \emptyset^* \\
@ & \equiv & \#^* \\
\alpha^+ & \equiv & \alpha\alpha^* \\
\# & \equiv & a_1 \ldots a_n \qquad \text{if } \Sigma = \{a_1 \ldots a_n\}
\end{array}
$$

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$\begin{array}{lll} \boldsymbol{\varepsilon} & \equiv & \sim(\#@) \equiv \boldsymbol{\emptyset}^* \\ @ & \equiv & \#^* \\ \alpha^+ & \equiv & \alpha\alpha^* \\ \# & \equiv & a_1 \ldots a_n \qquad \text{if } \Sigma = \{a_1 \ldots a_n\} \\ \alpha \cap \beta & \equiv & \sim(\sim\alpha + \sim\beta) \end{array}$$

Questions

- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

$$
\begin{array}{lcll}
\boldsymbol{\varepsilon} & \equiv & \sim(\#@) \equiv \boldsymbol{\emptyset}^* & \\
@ & \equiv & \#^* & \\
\alpha^+ & \equiv & \alpha\alpha^* & \\
\# & \equiv & a_1 \ldots a_n & \text{if } \Sigma = \{a_1 \ldots a_n\} \\
\alpha \cap \beta & \equiv & \sim(\sim\alpha + \sim\beta) & \\
\sim\alpha & \equiv & ? &
\end{array}
$$

Questions

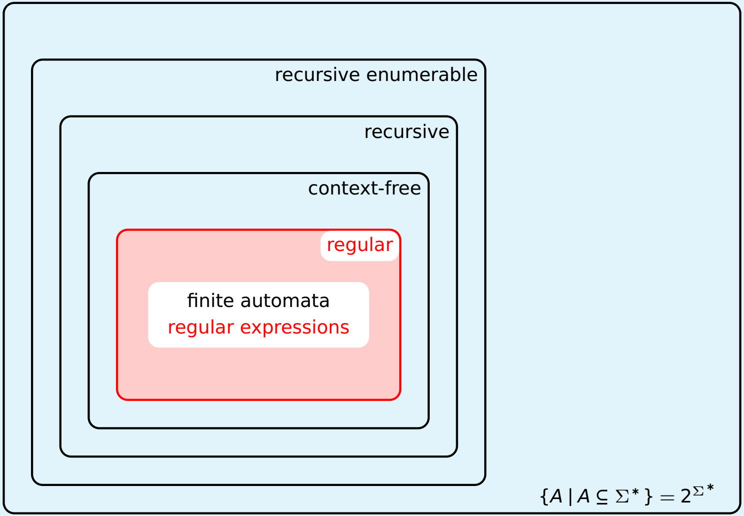
- how difficult is pattern matching?
- is pattern equivalence $(L(\alpha) = L(\beta))$ decidable?
- which operators are redundant?

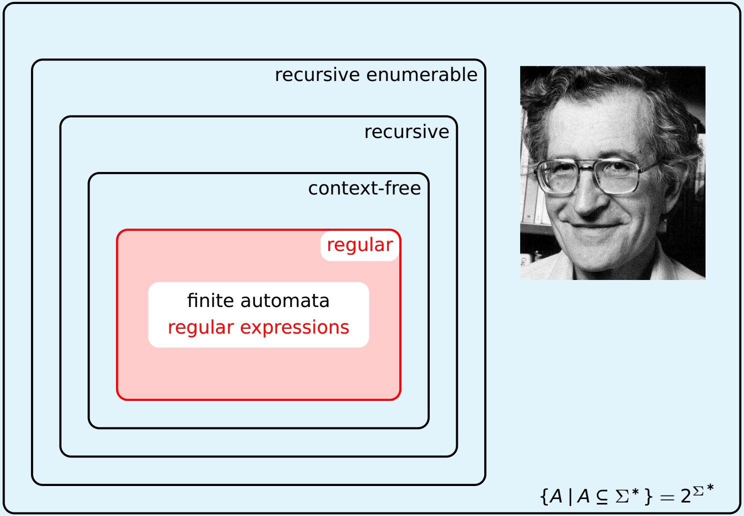
$$
\begin{array}{lll}
\boldsymbol{\epsilon} & \equiv & \sim(\#@) \equiv \boldsymbol{\varnothing}^* \\
@ & \equiv & \#^* \\
\alpha^+ & \equiv & \alpha\alpha^* \\
\# & \equiv & a_1 \ldots a_n \qquad \text{if } \Sigma = \{a_1 \ldots a_n\} \\
\alpha \cap \beta & \equiv & \sim(\sim\alpha + \sim\beta) \\
\sim\alpha & \equiv & ?
\end{array}
$$

Notation

$\alpha \equiv \beta \quad$ if $L(\alpha) = L(\beta)$

Outline


recursive enumerable
recursive
context-free
regular
finite automata
regular expressions
$\{A \mid A \subseteq \Sigma^*\} = 2^{\Sigma^*}$


recursive enumerable
recursive
context-free
regular
finite automata
regular expressions
$\{A \mid A \subseteq \Sigma^*\} = 2^{\Sigma^*}$

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha + \beta \quad \alpha^* \quad \alpha\beta$$

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha + \beta \quad \alpha^* \quad \alpha\beta$$

Theorem

finite automata, patterns, and regular expressions are equivalent:

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha + \beta \quad \alpha^* \quad \alpha\beta$$

Theorem

finite automata, patterns, and regular expressions are equivalent:

for all $A \subseteq \Sigma^*$ ➊ A is regular

$\iff$ ➋ $A = L(\alpha)$ for some pattern α

$\iff$ ➌ $A = L(\alpha)$ for some regular expression α

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha + \beta \quad \alpha^* \quad \alpha\beta$$

Theorem

finite automata, patterns, and regular expressions are equivalent:

for all $A \subseteq \Sigma^*$ ➀ A is regular

$\iff$ ➁ $A = L(\alpha)$ for some pattern α

$\iff$ ➂ $A = L(\alpha)$ for some regular expression α

Proof.

➂ $\Longrightarrow$ ➁ trivial (every regular expression is a pattern)

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha + \beta \quad \alpha^* \quad \alpha\beta$$

Theorem

finite automata, patterns, and regular expressions are equivalent:

for all $A \subseteq \Sigma^*$ ➀ A is regular

$\iff$ ➁ $A = L(\alpha)$ for some pattern α

$\iff$ ➂ $A = L(\alpha)$ for some regular expression α

Proof.

➂ $\implies$ ➁ trivial (every regular expression is a pattern)

➁ $\implies$ ➀ induction on α (see slides $\#16-17$)

Definition

regular expressions are restricted patterns which use only

$$\boldsymbol{a} \in \Sigma \quad \boldsymbol{\varepsilon} \quad \boldsymbol{\varnothing} \quad \alpha+\beta \quad \alpha^* \quad \alpha\beta$$

Theorem

finite automata, patterns, and regular expressions are equivalent:

for all $A \subseteq \Sigma^*$ ➊ A is regular

$\iff$ ➋ $A = L(\alpha)$ for some pattern α

$\iff$ ➌ $A = L(\alpha)$ for some regular expression α

Proof.

➌	$\Longrightarrow$	➋	trivial	(every regular expression is a pattern)
➋	$\Longrightarrow$	➊	induction on α	(see slides $\#16-17$)
➊	$\Longrightarrow$	➌		(see slide $\#18$)

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

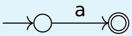
1. atomic pattern $\Sigma = \{a_1, \ldots, a_n\}$

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

1 atomic pattern $\Sigma = \{a_1, \ldots, a_n\}$

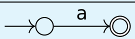
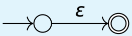
α	$L(\alpha)$	finite automaton
$\boldsymbol{a} \in \Sigma$	$\{a\}$	

Proof. (2 $\Longrightarrow$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

1. atomic pattern $\Sigma = \{a_1, \ldots, a_n\}$

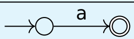
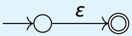
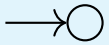
α	$L(\alpha)$	finite automaton
$\boldsymbol{a} \in \Sigma$	$\{a\}$	
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$	

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

1 atomic pattern $\Sigma = \{a_1, \ldots, a_n\}$

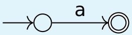
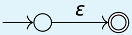
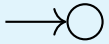
α	$L(\alpha)$	finite automaton
$\boldsymbol{a} \in \Sigma$	$\{a\}$	
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$	
$\boldsymbol{\emptyset}$	$\emptyset$	

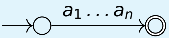
Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

1. atomic pattern $\Sigma = \{a_1, \ldots, a_n\}$

α	$L(\alpha)$	finite automaton
$\boldsymbol{a} \in \Sigma$	$\{a\}$	
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$	
$\boldsymbol{\varnothing}$	$\varnothing$	

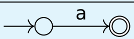
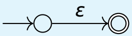
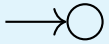
α	$L(\alpha)$	finite automaton
$\#$	Σ	

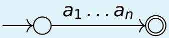
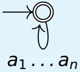
Proof. (2 $\Longrightarrow$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

1. atomic pattern $\quad \Sigma = \{a_1, \ldots, a_n\}$

α	$L(\alpha)$	finite automaton
$\boldsymbol{a} \in \Sigma$	$\{a\}$	
$\boldsymbol{\varepsilon}$	$\{\varepsilon\}$	
$\boldsymbol{\varnothing}$	$\varnothing$	

α	$L(\alpha)$	finite automaton
$\#$	Σ	
$@$	Σ^*	

Proof. (➋ $\Longrightarrow$ ➊)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

➋ compound patterns

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2 compound patterns

α	$L(\alpha)$
$\beta+\gamma$	$L(\beta)\cup L(\gamma)$

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2 compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$

Proof. (2 $\Longrightarrow$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2 compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2. compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

α	$L(\alpha)$
β^*	$L(\beta)^*$

Proof. (2 $\Longrightarrow$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2 compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

α	$L(\alpha)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$

Proof. (2 $\implies$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2 compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

α	$L(\alpha)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$
$\sim \beta$	$\sim L(\beta)$

Proof. (② $\Longrightarrow$ ①)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

② compound patterns

α	$L(\alpha)$
$\beta+\gamma$	$L(\beta)\cup L(\gamma)$
$\beta\cap\gamma$	$L(\beta)\cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

α	$L(\alpha)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$
$\sim\beta$	$\sim L(\beta)$

$L(\beta)$ and $L(\gamma)$ are regular according to induction hypothesis

Proof. (2 $\Longrightarrow$ 1)

statement: for any pattern α, $L(\alpha)$ is regular

induction on pattern α

2. compound patterns

α	$L(\alpha)$
$\beta + \gamma$	$L(\beta) \cup L(\gamma)$
$\beta \cap \gamma$	$L(\beta) \cap L(\gamma)$
$\beta\gamma$	$L(\beta)L(\gamma)$

α	$L(\alpha)$
β^*	$L(\beta)^*$
β^+	$L(\beta)^+$
$\sim\beta$	$\sim L(\beta)$

$L(\beta)$ and $L(\gamma)$ are regular according to induction hypothesis

hence $L(\alpha)$ is regular according to closure properties of regular sets

Proof. (1 $\implies$ 3 – An idea)

given NFA$_\varepsilon$ $N_\varepsilon = (Q, \Sigma, \varepsilon, \Delta, S, F)$

$\forall Y \subseteq Q \quad \forall u, v \in Q$ construct regular expression α_{uv}^{Y} such that

$x \in L(\alpha_{uv}^{Y}) \iff$ $\boxed{\exists$ a path from u to v labeled x $(v \in \widehat{\Delta}(\{u\}, x))$ such that all intermediate states belong to $Y}$

Proof. (1 $\implies$ 3 – An idea)

given NFA$_\varepsilon$ $N_\varepsilon = (Q, \Sigma, \varepsilon, \Delta, S, F)$

$\forall Y \subseteq Q \quad \forall u, v \in Q$ construct regular expression α_{uv}^{Y} such that

$x \in L(\alpha_{uv}^{Y}) \iff$ $\exists$ a path from u to v labeled x $(v \in \widehat{\Delta}(\{u\}, x))$ such that all intermediate states belong to Y

Definitions

- $$\alpha_{uv}^{\emptyset} := \begin{cases} \mathbf{a_1} + \ldots + \mathbf{a_k} & \text{if } u \neq v \text{ and } k > 0 \\ \boldsymbol{\emptyset} & \text{if } u \neq v \text{ and } k = 0 \\ \mathbf{a_1} + \ldots + \mathbf{a_k} + \boldsymbol{\varepsilon} & \text{if } u = v \text{ and } k > 0 \\ \boldsymbol{\varepsilon} & \text{if } u = v \text{ and } k = 0 \end{cases}$$

$$\{a_1, \ldots, a_k\} := \{a \in \Sigma \cup \{\varepsilon\} \mid v \in \Delta(u, a)\}$$

Proof. (1 $\Longrightarrow$ 3 – An idea)

given NFA$_\varepsilon$ $N_\varepsilon = (Q, \Sigma, \varepsilon, \Delta, S, F)$

$\forall Y \subseteq Q \quad \forall u, v \in Q$ construct regular expression α_{uv}^{Y} such that

$x \in L(\alpha_{uv}^{Y}) \iff$ $\boxed{\text{$\exists$ a path from u to v labeled x ($v \in \widehat{\Delta}(\{u\}, x)$) such that all intermediate states belong to Y}}$

Definitions

- $$\alpha_{uv}^{\emptyset} := \begin{cases} \mathbf{a_1} + \ldots + \mathbf{a_k} & \text{if } u \neq v \text{ and } k > 0 \\ \boldsymbol{\emptyset} & \text{if } u \neq v \text{ and } k = 0 \\ \mathbf{a_1} + \ldots + \mathbf{a_k} + \boldsymbol{\varepsilon} & \text{if } u = v \text{ and } k > 0 \\ \boldsymbol{\varepsilon} & \text{if } u = v \text{ and } k = 0 \end{cases} \qquad \{a_1, \ldots, a_k\} := \{a \in \Sigma \cup \{\varepsilon\} \mid v \in \Delta(u, a)\}$$
- $\alpha_{uv}^{Y} := \alpha_{uv}^{Y-\{q\}} + \alpha_{uq}^{Y-\{q\}} (\alpha_{qq}^{Y-\{q\}})^{*} \alpha_{qv}^{Y-\{q\}}$ for some fixed $q \in Y$

Proof. (① $\implies$ ③ – An idea)

given NFA$_\varepsilon$ $N_\varepsilon = (Q, \Sigma, \varepsilon, \Delta, S, F)$

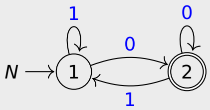
$\forall Y \subseteq Q \quad \forall u, v \in Q \quad$ construct regular expression α^Y_{uv} such that

$x \in L(\alpha^Y_{uv}) \iff$ $\exists$ a path from u to v labeled x ($v \in \widehat{\Delta}(\{u\}, x)$) such that all intermediate states belong to Y

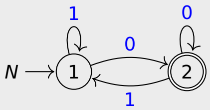
Theorem

$$L(N_\varepsilon) = L\left(\sum_{s\in S, t\in F} \alpha^Q_{st}\right)$$

Example

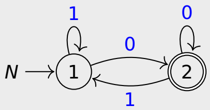


Example



$L(N) = L(\alpha)$ with

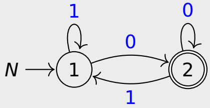
Example



$L(N) = L(\alpha)$ with

$$\alpha \quad = \quad \alpha_{12}^{\{1,2\}}$$

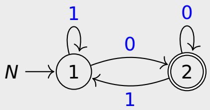
Example



$L(N) = L(\alpha)$ with

$$\alpha \quad = \quad \alpha_{12}^{\{1,2\}} \quad = \quad \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^{*}\alpha_{22}^{\{1\}} \qquad (q = 2)$$

Example

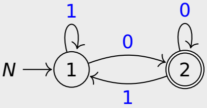


$L(N) = L(\alpha)$ with

$$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \quad (q = 2)$$

$$\alpha_{12}^{\{1\}} = \alpha_{12}^{\emptyset} + \alpha_{11}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset}$$

Example



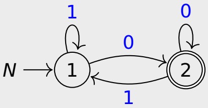
$L(N) = L(\alpha)$ with

$$\alpha \quad = \quad \alpha_{12}^{\{1,2\}} \quad = \quad \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \qquad (q = 2)$$

$$\alpha_{12}^{\{1\}} \quad = \quad \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing}$$

$$\alpha_{22}^{\{1\}} \quad = \quad \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing}$$

Example



$L(N) = L(\alpha)$ with

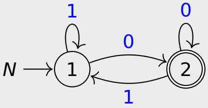
$$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \quad \text{(q = 2)}$$

$$\alpha_{12}^{\{1\}} = \alpha_{12}^{\emptyset} + \alpha_{11}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset}$$

$$\alpha_{22}^{\{1\}} = \alpha_{22}^{\emptyset} + \alpha_{21}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset}$$

$$\alpha_{12}^{\emptyset} = 0$$

Example



$L(N) = L(\alpha)$ with

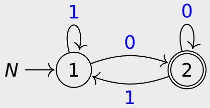
$$\alpha \quad = \quad \alpha_{12}^{\{1,2\}} \quad = \quad \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \qquad (q = 2)$$

$$\alpha_{12}^{\{1\}} \quad = \quad \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing}$$

$$\alpha_{22}^{\{1\}} \quad = \quad \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing}$$

$$\alpha_{12}^{\varnothing} = 0 \qquad \alpha_{11}^{\varnothing} = 1 + \boldsymbol{\varepsilon}$$

Example



$L(N) = L(\alpha)$ with

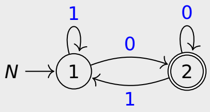
$$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^*\alpha_{22}^{\{1\}} \quad (q = 2)$$

$$\alpha_{12}^{\{1\}} = \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing}$$

$$\alpha_{22}^{\{1\}} = \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing}$$

$$\alpha_{12}^{\varnothing} = 0 \quad \alpha_{11}^{\varnothing} = 1 + \boldsymbol{\varepsilon} \quad \alpha_{22}^{\varnothing} = 0 + \boldsymbol{\varepsilon}$$

Example



$L(N) = L(\alpha)$ with

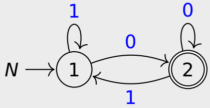
$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^*\alpha_{22}^{\{1\}}$ (q = 2)

$\alpha_{12}^{\{1\}} = \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing}$

$\alpha_{22}^{\{1\}} = \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing}$

$\alpha_{12}^{\varnothing} = 0 \quad \alpha_{11}^{\varnothing} = 1 + \boldsymbol{\varepsilon} \quad \alpha_{22}^{\varnothing} = 0 + \boldsymbol{\varepsilon} \quad \alpha_{21}^{\varnothing} = 1$

Example

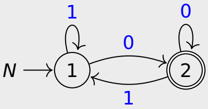


$L(N) = L(\alpha)$ with

$$
\begin{aligned}
\alpha &= \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \quad (\text{q} = 2) \\
\alpha_{12}^{\{1\}} &= \alpha_{12}^{\emptyset} + \alpha_{11}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset} = 0 + (1 + \boldsymbol{\varepsilon})(1 + \boldsymbol{\varepsilon})^* 0 \\
\alpha_{22}^{\{1\}} &= \alpha_{22}^{\emptyset} + \alpha_{21}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset}
\end{aligned}
$$

$\alpha_{12}^{\emptyset} = 0 \quad \alpha_{11}^{\emptyset} = 1 + \boldsymbol{\varepsilon} \quad \alpha_{22}^{\emptyset} = 0 + \boldsymbol{\varepsilon} \quad \alpha_{21}^{\emptyset} = 1$

Example



$L(N) = L(\alpha)$ with

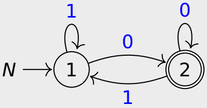
$$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \quad \text{(q = 2)}$$

$$\alpha_{12}^{\{1\}} = \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing} = 0 + (1+\boldsymbol{\varepsilon})(1+\boldsymbol{\varepsilon})^* 0$$

$$\alpha_{22}^{\{1\}} = \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^* \alpha_{12}^{\varnothing} = (0+\boldsymbol{\varepsilon}) + 1(1+\boldsymbol{\varepsilon})^* 0$$

$$\alpha_{12}^{\varnothing} = 0 \quad \alpha_{11}^{\varnothing} = 1 + \boldsymbol{\varepsilon} \quad \alpha_{22}^{\varnothing} = 0 + \boldsymbol{\varepsilon} \quad \alpha_{21}^{\varnothing} = 1$$

Example



$L(N) = L(\alpha)$ with

$$\alpha \quad = \quad \alpha_{12}^{\{1,2\}} \quad = \quad \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^* \alpha_{22}^{\{1\}} \qquad (\text{q = 2})$$

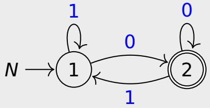
$$\alpha_{12}^{\{1\}} \quad = \quad \alpha_{12}^{\emptyset} + \alpha_{11}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset} \quad = \quad 0 + (1 + \boldsymbol{\varepsilon})(1 + \boldsymbol{\varepsilon})^* 0$$

$$\alpha_{22}^{\{1\}} \quad = \quad \alpha_{22}^{\emptyset} + \alpha_{21}^{\emptyset}(\alpha_{11}^{\emptyset})^* \alpha_{12}^{\emptyset} \quad = \quad (0 + \boldsymbol{\varepsilon}) + 1(1 + \boldsymbol{\varepsilon})^* 0$$

$$\alpha_{12}^{\emptyset} = 0 \quad \alpha_{11}^{\emptyset} = 1 + \boldsymbol{\varepsilon} \quad \alpha_{22}^{\emptyset} = 0 + \boldsymbol{\varepsilon} \quad \alpha_{21}^{\emptyset} = 1$$

$$\alpha \quad = \quad (0 + (1 + \boldsymbol{\varepsilon})(1 + \boldsymbol{\varepsilon})^* 0) + (0 + (1 + \boldsymbol{\varepsilon})(1 + \boldsymbol{\varepsilon})^* 0)((0 + \boldsymbol{\varepsilon}) + 1(1 + \boldsymbol{\varepsilon})^* 0)^*((0 + \boldsymbol{\varepsilon}) + 1(1 + \boldsymbol{\varepsilon})^* 0)$$

Example



$L(N) = L(\alpha)$ with

$$\alpha = \alpha_{12}^{\{1,2\}} = \alpha_{12}^{\{1\}} + \alpha_{12}^{\{1\}}(\alpha_{22}^{\{1\}})^*\alpha_{22}^{\{1\}} \quad (q = 2)$$

$$\alpha_{12}^{\{1\}} = \alpha_{12}^{\varnothing} + \alpha_{11}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing} = 0 + (1+\boldsymbol{\varepsilon})(1+\boldsymbol{\varepsilon})^*0$$

$$\alpha_{22}^{\{1\}} = \alpha_{22}^{\varnothing} + \alpha_{21}^{\varnothing}(\alpha_{11}^{\varnothing})^*\alpha_{12}^{\varnothing} = (0+\boldsymbol{\varepsilon}) + 1(1+\boldsymbol{\varepsilon})^*0$$

$$\alpha_{12}^{\varnothing} = 0 \quad \alpha_{11}^{\varnothing} = 1+\boldsymbol{\varepsilon} \quad \alpha_{22}^{\varnothing} = 0+\boldsymbol{\varepsilon} \quad \alpha_{21}^{\varnothing} = 1$$

$$\alpha = (0+(1+\boldsymbol{\varepsilon})(1+\boldsymbol{\varepsilon})^*0) + (0+(1+\boldsymbol{\varepsilon})(1+\boldsymbol{\varepsilon})^*0)((0+\boldsymbol{\varepsilon})+1(1+\boldsymbol{\varepsilon})^*0)^*((0+\boldsymbol{\varepsilon})+1(1+\boldsymbol{\varepsilon})^*0)$$

$$\equiv (0+1)^*0$$

Outline

Theorem

regular sets are effectively closed under homomorphic image and preimage

Theorem

regular sets are effectively closed under homomorphic image and preimage

Definitions

- homomorphism is mapping $h\colon \Sigma^* \to \Gamma^*$ such that

$$h(\varepsilon) = \varepsilon \qquad\qquad h(xy) = h(x)h(y)$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Definitions

- homomorphism is mapping $h\colon \Sigma^* \to \Gamma^*$ such that

$$h(\varepsilon) = \varepsilon \qquad\qquad h(xy) = h(x)h(y)$$

so homomorphism is completely determined by its effect on Σ

Theorem

regular sets are effectively closed under homomorphic image and preimage

Definitions

- homomorphism is mapping $h\colon \Sigma^* \to \Gamma^*$ such that

$$h(\varepsilon) = \varepsilon \qquad\qquad h(xy) = h(x)h(y)$$

so homomorphism is completely determined by its effect on Σ

- if $A \subseteq \Sigma^*$ then $\qquad h(A) \;=\; \{h(x) \mid x \in A\} \subseteq \Gamma^* \qquad$ "image of A under h"

Theorem

regular sets are effectively closed under homomorphic image and preimage

Definitions

- homomorphism is mapping $h\colon \Sigma^* \to \Gamma^*$ such that

$$h(\varepsilon) = \varepsilon \qquad\qquad h(xy) = h(x)h(y)$$

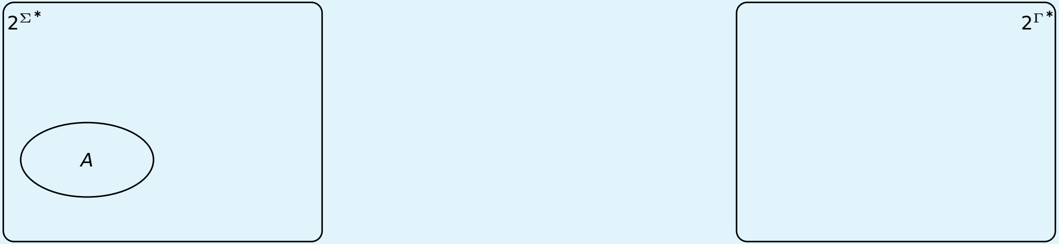
  so homomorphism is completely determined by its effect on Σ

- if $A \subseteq \Sigma^*$ then $\quad h(A) = \{h(x) \mid x \in A\} \subseteq \Gamma^*$ "image of A under h"
  if $B \subseteq \Gamma^*$ then $\quad h^{-1}(B) = \{x \mid h(x) \in B\} \subseteq \Sigma^*$ "preimage of B under h"

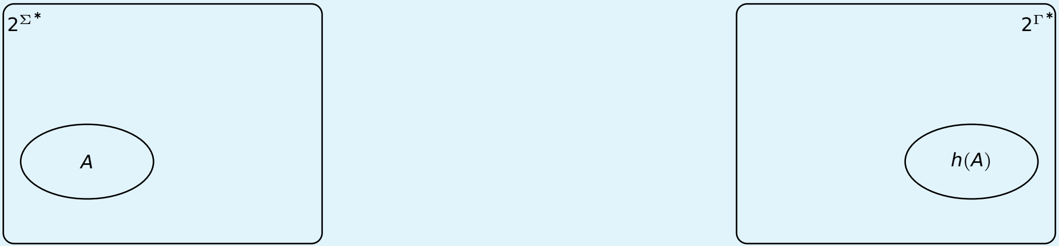
2^{Σ^*}

2^{Γ^*}

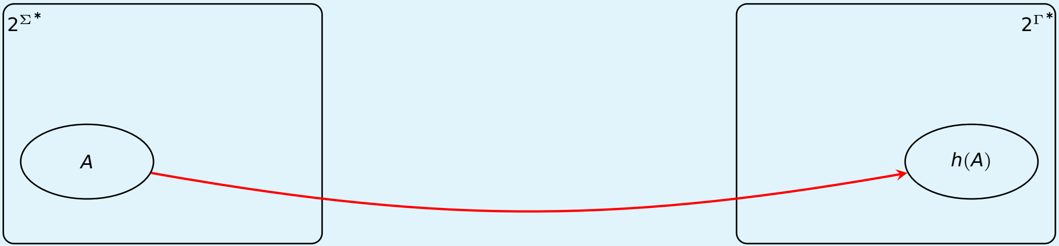
- homomorphism $h\colon \Sigma^* \to \Gamma^*$



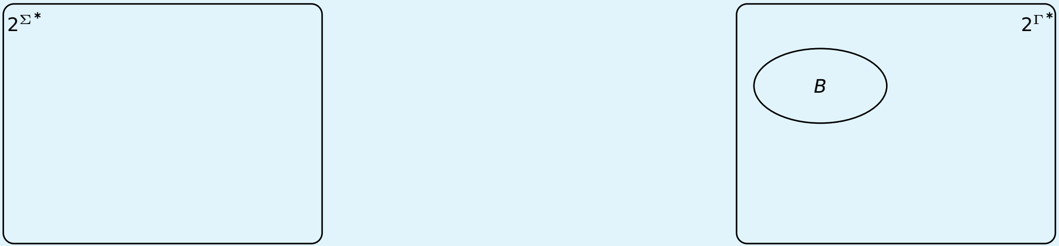
- homomorphism $h \colon \Sigma^* \to \Gamma^*$



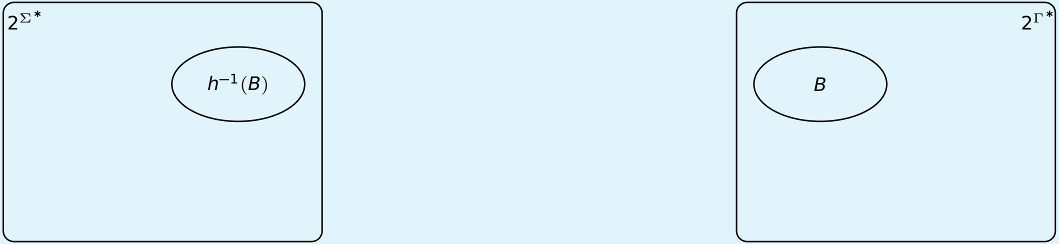
- homomorphism $h\colon \Sigma^* \to \Gamma^*$



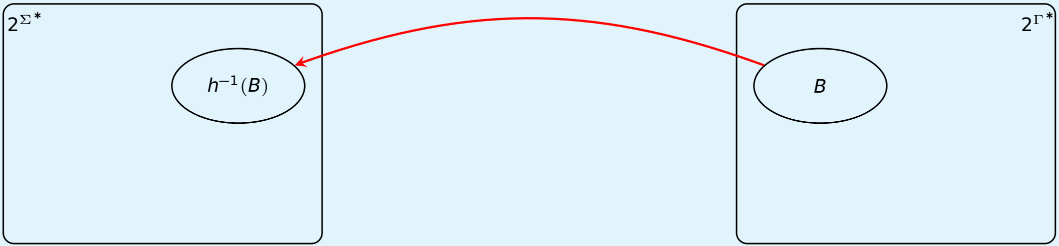
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$



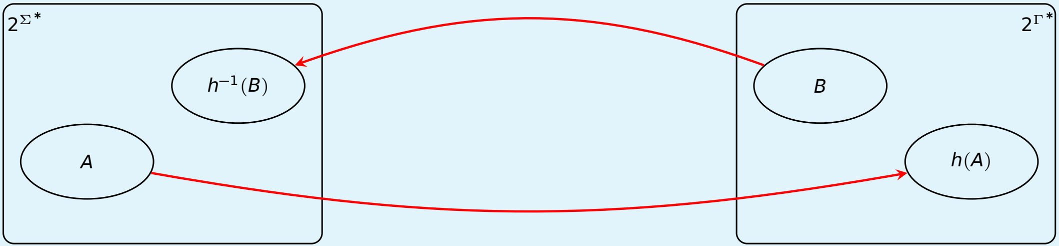
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$



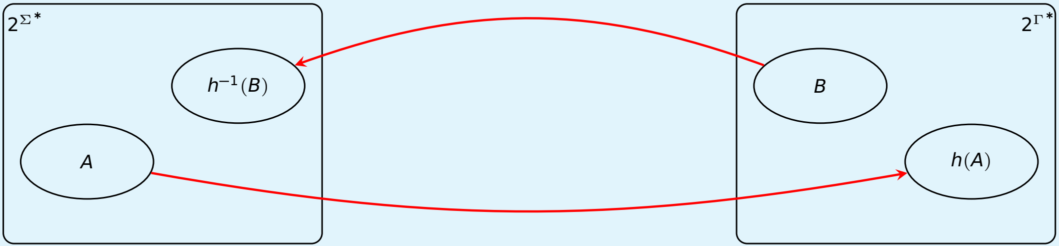
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$



- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$
- $h(h^{-1}(B)) \subseteq B$



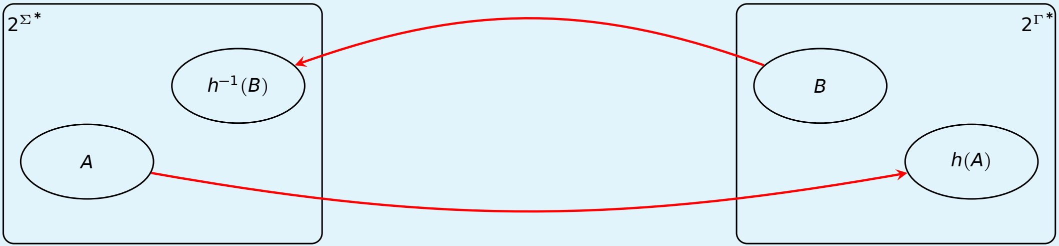
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$
- $h(h^{-1}(B)) \subseteq B$



- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$
- $h(h^{-1}(B)) \subseteq B$

Example

$\Sigma = \Gamma = \{0, 1\} \quad h(0) = 11 \quad h(1) = 1$

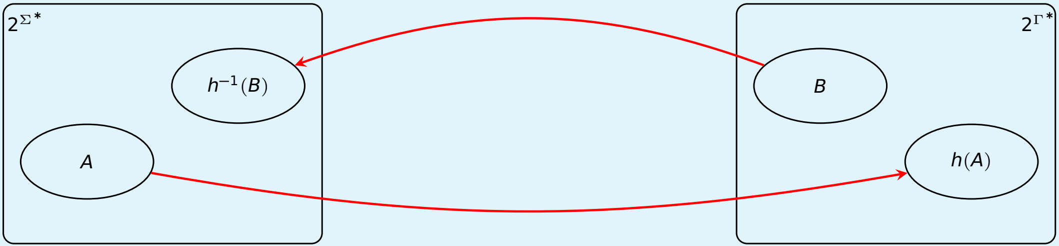


- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$
- $h(h^{-1}(B)) \subseteq B$

Example

$\Sigma = \Gamma = \{0, 1\}$ $\quad h(0) = 11$ $\quad h(1) = 1$ $\quad A = \{0\}$

- $h^{-1}(h(A)) = h^{-1}(\{11\}) = \{0, 11\} \supset A$



- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(h(A)) \supseteq A$
- $h(h^{-1}(B)) \subseteq B$

Example

$\Sigma = \Gamma = \{0, 1\} \quad h(0) = 11 \quad h(1) = 1 \quad A = B = \{0\}$

- $h^{-1}(h(A)) = h^{-1}(\{11\}) = \{0, 11\} \supset A$
- $h(h^{-1}(B)) = h(\varnothing) = \varnothing \subset B$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$
- define homomorphisms $h, i \colon \Gamma^* \to \Sigma^*$ by

$$h(0) = 0 \quad h(1) = h(2) = 1 \quad i(0) = 0 \quad i(1) = 1 \quad i(2) = \varepsilon$$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$
- define homomorphisms $h, i \colon \Gamma^* \to \Sigma^*$ by

$$h(0) = 0 \quad h(1) = h(2) = 1 \quad i(0) = 0 \quad i(1) = 1 \quad i(2) = \varepsilon$$

- $h^{-1}(A) = \{x \mid h(x) \in A\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$
- define homomorphisms $h, i \colon \Gamma^* \to \Sigma^*$ by

$$h(0) = 0 \quad h(1) = h(2) = 1 \quad i(0) = 0 \quad i(1) = 1 \quad i(2) = \epsilon$$

- $h^{-1}(A) = \{x \mid h(x) \in A\}$
- $h^{-1}(A) \cap L((0+1)^* 2 (0+1)^*) = \{x2y \mid x1y \in A\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$
- define homomorphisms $h, i \colon \Gamma^* \to \Sigma^*$ by

$$h(0) = 0 \quad h(1) = h(2) = 1 \quad i(0) = 0 \quad i(1) = 1 \quad i(2) = \varepsilon$$

- $h^{-1}(A) = \{x \mid h(x) \in A\}$
- $h^{-1}(A) \cap L((0+1)^*2(0+1)^*) = \{x2y \mid x1y \in A\}$
- $\{xy \mid x1y \in A\} = i(h^{-1}(A) \cap L((0+1)^*2(0+1)^*))$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\{xy \mid x1y \in A\}$ is regular

Proof.

- $\Sigma = \{0,1\}$ and $\Gamma = \{0,1,2\}$
- define homomorphisms $h, i \colon \Gamma^* \to \Sigma^*$ by

$$h(0) = 0 \quad h(1) = h(2) = 1 \quad i(0) = 0 \quad i(1) = 1 \quad i(2) = \epsilon$$

- $h^{-1}(A) = \{x \mid h(x) \in A\}$
- $h^{-1}(A) \cap L((0+1)^*2(0+1)^*) = \{x2y \mid x1y \in A\}$
- $\{xy \mid x1y \in A\} = i(h^{-1}(A) \cap L((0+1)^*2(0+1)^*))$ is regular

Theorem

regular sets are effectively closed under homomorphic image and preimage

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- DFA $M = (Q, \Gamma, \delta, s, F)$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- DFA $M = (Q, \Gamma, \delta, s, F)$
- homomorphism $h\colon \Sigma^* \to \Gamma^*$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- DFA $M = (Q, \Gamma, \delta, s, F)$
- homomorphism $h \colon \Sigma^* \to \Gamma^*$
- $h^{-1}(L(M)) = L(M')$ for DFA $M' = (Q, \Sigma, \delta', s, F)$ with $\delta'(q, a) := \widehat{\delta}(q, h(a))$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- DFA $M = (Q, \Gamma, \delta, s, F)$
- homomorphism $h \colon \Sigma^* \to \Gamma^*$
- $h^{-1}(L(M)) = L(M')$ for DFA $M' = (Q, \Sigma, \delta', s, F)$ with $\delta'(q, a) := \widehat{\delta}(q, h(a))$
- claim: $\widehat{\delta'}(q, x) = \widehat{\delta}(q, h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- DFA $M = (Q, \Gamma, \delta, s, F)$
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h^{-1}(L(M)) = L(M')$ for DFA $M' = (Q, \Sigma, \delta', s, F)$ with $\delta'(q,a) := \widehat{\delta}(q, h(a))$
- claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x))$ $\forall x \in \Sigma^*\ \forall q \in Q$
  proof of claim: induction on $|x|$ (see next slide)

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

proof of the claim

claim: $\widehat{\delta'}(q, x) = \widehat{\delta}(q, h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q, \varepsilon) = q = \widehat{\delta}(q, h(\varepsilon))$$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with $\mathrm{IH} : \widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with IH : $\widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

$$\widehat{\delta'}(q,ya) \quad = \quad \delta'(\widehat{\delta'}(q,y),a) \qquad \text{(by definition of } \widehat{\delta'}\text{)}$$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with IH $: \widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

$$\begin{aligned} \widehat{\delta'}(q,ya) &= \delta'(\widehat{\delta'}(q,y),a) && \text{(by definition of } \widehat{\delta'}\text{)} \\ &= \delta'(\widehat{\delta}(q,h(y)),a) && \text{(by induction hypothesis IH)} \end{aligned}$$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with IH : $\widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

$$\begin{aligned}
\widehat{\delta'}(q,ya) &= \delta'(\widehat{\delta'}(q,y),a) && \text{(by definition of } \widehat{\delta'}\text{)}\\
&= \delta'(\widehat{\delta}(q,h(y)),a) && \text{(by induction hypothesis IH)}\\
&= \widehat{\delta}(\widehat{\delta}(q,h(y)),h(a)) && \text{(by definition of } \delta'\text{)}
\end{aligned}$$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with IH : $\widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

$$\begin{aligned}
\widehat{\delta'}(q,ya) &= \delta'(\widehat{\delta'}(q,y),a) && \text{(by definition of } \widehat{\delta'}\text{)} \\
&= \delta'(\widehat{\delta}(q,h(y)),a) && \text{(by induction hypothesis IH)} \\
&= \widehat{\delta}(\widehat{\delta}(q,h(y)),h(a)) && \text{(by definition of } \delta'\text{)} \\
&= \widehat{\delta}(q,h(y)h(a)) && \text{(by distributivity of } \widehat{\delta} \text{ – w3.pdf, slide 10)}
\end{aligned}$$

proof of the claim

claim: $\widehat{\delta'}(q,x) = \widehat{\delta}(q,h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q,\varepsilon) = q = \widehat{\delta}(q,h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with IH : $\widehat{\delta'}(q,y) = \widehat{\delta}(q,h(y))$

$$\begin{aligned}
\widehat{\delta'}(q,ya) &= \delta'(\widehat{\delta'}(q,y),a) && \text{(by definition of } \widehat{\delta'}\text{)}\\
&= \delta'(\widehat{\delta}(q,h(y)),a) && \text{(by induction hypothesis IH)}\\
&= \widehat{\delta}(\widehat{\delta}(q,h(y)),h(a)) && \text{(by definition of } \delta'\text{)}\\
&= \widehat{\delta}(q,h(y)h(a)) && \text{(by distributivity of } \widehat{\delta} \text{ – w3.pdf, slide 10)}\\
&= \widehat{\delta}(q,h(ya)) && \text{(by definition of homomorphism)}
\end{aligned}$$

proof of the claim

claim: $\widehat{\delta'}(q, x) = \widehat{\delta}(q, h(x)) \quad \forall x \in \Sigma^* \ \forall q \in Q$

- base case: $|x| = 0$ thus $x = \varepsilon$

$$\widehat{\delta'}(q, \varepsilon) = q = \widehat{\delta}(q, h(\varepsilon))$$

- step case: $|x| > 0$ thus $x = ya$ s.t. $|y| = |x| - 1$ with $\mathrm{IH} : \widehat{\delta'}(q, y) = \widehat{\delta}(q, h(y))$

$$\begin{aligned}
\widehat{\delta'}(q, ya) &= \delta'(\widehat{\delta'}(q, y), a) && \text{(by definition of } \widehat{\delta'}\text{)} \\
&= \delta'(\widehat{\delta}(q, h(y)), a) && \text{(by induction hypothesis IH)} \\
&= \widehat{\delta}(\widehat{\delta}(q, h(y)), h(a)) && \text{(by definition of } \delta'\text{)} \\
&= \widehat{\delta}(q, h(y)h(a)) && \text{(by distributivity of } \widehat{\delta} \text{ – w3.pdf, slide 10)} \\
&= \widehat{\delta}(q, h(ya)) && \text{(by definition of homomorphism)} \\
&= \widehat{\delta}(q, h(x))
\end{aligned}$$

$\square$

Proof. (closedness under complement homomorphic preimage)

statement: $L(M') = h^{-1}(L(M))$

Proof. (closedness under complement homomorphic preimage)

statement: $L(M') = h^{-1}(L(M))$

$$\forall x \in \Sigma^*,\, x \in L(M') \quad \Longleftrightarrow \quad \widehat{\delta'}(s, x) \in F \qquad \text{(by definition of acceptance)}$$

Proof. (closedness under complement homomorphic preimage)

statement: $L(M') = h^{-1}(L(M))$

$$
\begin{aligned}
\forall x \in \Sigma^*,\ x \in L(M') \quad &\iff \quad \widehat{\delta'}(s,x) \in F && \text{(by definition of acceptance)} \\
&\iff \quad \widehat{\delta}(s,h(x)) \in F && \text{(by claim proven in slide 25)}
\end{aligned}
$$

Proof. (closedness under complement homomorphic preimage)

statement: $L(M') = h^{-1}(L(M))$

$$
\begin{aligned}
\forall x \in \Sigma^*,\ x \in L(M') &\iff \widehat{\delta'}(s, x) \in F && \text{(by definition of acceptance)} \\
&\iff \widehat{\delta}(s, h(x)) \in F && \text{(by claim proven in slide 25)} \\
&\iff h(x) \in L(M) && \text{(by definition of acceptance)}
\end{aligned}
$$

Proof. (closedness under complement homomorphic preimage)

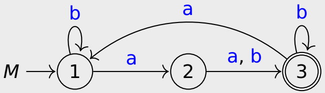
statement: $L(M') = h^{-1}(L(M))$

$$
\begin{aligned}
\forall x \in \Sigma^*,\ x \in L(M') &\iff \widehat{\delta'}(s, x) \in F && \text{(by definition of acceptance)}\\
&\iff \widehat{\delta}(s, h(x)) \in F && \text{(by claim proven in slide 25)}\\
&\iff h(x) \in L(M) && \text{(by definition of acceptance)}\\
&\iff x \in h^{-1}(L(M)) && \text{(by definition of homomorphic preimage)}
\end{aligned}
$$

$\square$

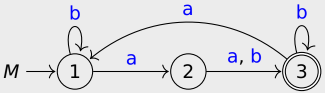
Example

- DFA M



Example

- DFA M

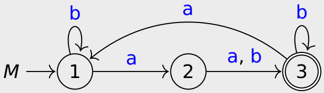


- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

Example

- DFA M



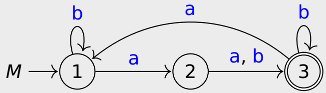
- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

- DFA M'

Example

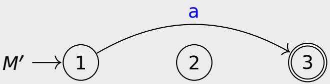
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \qquad h(b) = \varepsilon \qquad h(c) = bab$$

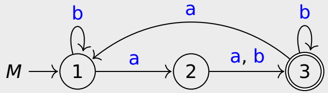
- DFA M'



$$\delta'(1, a) = \widehat{\delta}(1, aa) = 3$$

Example

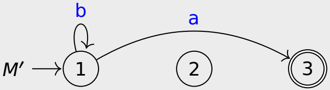
- DFA M



- homomorphism $h\colon \{a,b,c\}^* \to \{a,b\}^*$

$$h(a) = aa \qquad h(b) = \varepsilon \qquad h(c) = bab$$

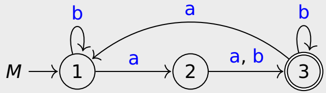
- DFA M'



$$\delta'(1,b) = \widehat{\delta}(1,\varepsilon) = 1$$

Example

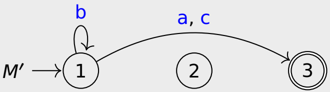
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \qquad h(b) = \varepsilon \qquad h(c) = bab$$

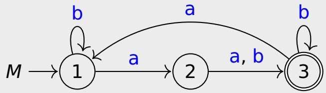
- DFA M'



$$\delta'(1, c) = \widehat{\delta}(1, bab) = 3$$

Example

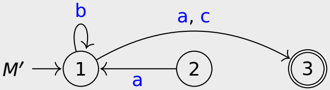
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

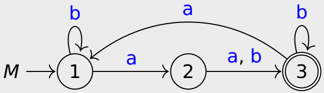
- DFA M'



$\delta'(2, a) = \widehat{\delta}(2, aa) = 1$

Example

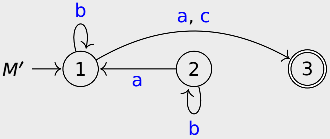
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

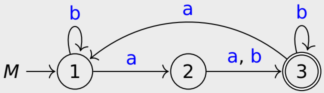
- DFA M'



$\delta'(2, b) = \widehat{\delta}(2, \varepsilon) = 2$

Example

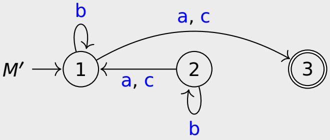
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

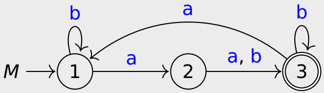
- DFA M'



$$\delta'(2, c) = \widehat{\delta}(2, bab) = 1$$

Example

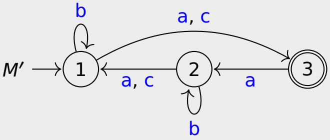
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

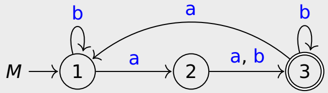
- DFA M'



$\delta'(3, a) = \widehat{\delta}(3, aa) = 2$

Example

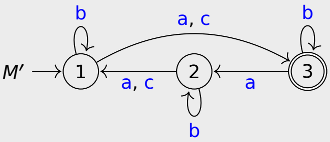
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

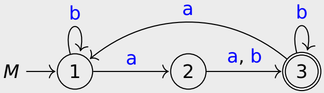
- DFA M'



$\delta'(3, b) = \widehat{\delta}(3, \varepsilon) = 3$

Example

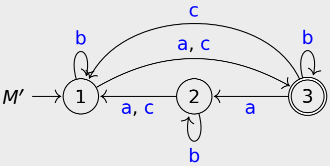
- DFA M



- homomorphism $h\colon \{a, b, c\}^* \to \{a, b\}^*$

$$h(a) = aa \quad h(b) = \varepsilon \quad h(c) = bab$$

- DFA M'



$\delta'(3, c) = \widehat{\delta}(3, bab) = 1$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$\mathbf{a'} \quad = \quad h(\mathbf{a}) \quad \text{for } \mathbf{a} \in \Sigma$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$\begin{aligned} \mathbf{a}' &= h(\mathbf{a}) && \text{for } \mathbf{a} \in \Sigma \\ \boldsymbol{\epsilon}' &= \boldsymbol{\epsilon} \end{aligned}$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$
\begin{aligned}
\mathbf{a}' &= h(\mathbf{a}) \quad \text{for } \mathbf{a} \in \Sigma \\
\boldsymbol{\varepsilon}' &= \boldsymbol{\varepsilon} \\
\boldsymbol{\varnothing}' &= \boldsymbol{\varnothing}
\end{aligned}
$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$\begin{aligned} \mathbf{a}' &= h(\mathbf{a}) \quad \text{for } \mathbf{a} \in \Sigma & \qquad (\beta+\gamma)' &= \beta' + \gamma' \\ \boldsymbol{\varepsilon}' &= \boldsymbol{\varepsilon} & & \\ \boldsymbol{\varnothing}' &= \boldsymbol{\varnothing} & & \end{aligned}$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$
\begin{aligned}
\mathbf{a}' &= h(\mathbf{a}) \quad \text{for } \mathbf{a} \in \Sigma & \qquad (\beta+\gamma)' &= \beta' + \gamma' \\
\boldsymbol{\epsilon}' &= \boldsymbol{\epsilon} & (\beta\gamma)' &= \beta'\gamma' \\
\boldsymbol{\varnothing}' &= \boldsymbol{\varnothing} & &
\end{aligned}
$$

Theorem

regular sets are effectively closed under homomorphic image and preimage

Proof.

- regular expression α over Σ
- homomorphism $h\colon \Sigma^* \to \Gamma^*$
- $h(L(\alpha)) = L(\alpha')$ for regular expression α' defined inductively:

$$\begin{aligned} \mathbf{a}' &= h(\mathbf{a}) \quad \text{for } \mathbf{a} \in \Sigma & \qquad (\beta+\gamma)' &= \beta' + \gamma' \\ \boldsymbol{\epsilon}' &= \boldsymbol{\epsilon} & (\beta\gamma)' &= \beta'\gamma' \\ \boldsymbol{\varnothing}' &= \boldsymbol{\varnothing} & (\beta^*)' &= (\beta')^* \end{aligned}$$

Definitions

- Hamming distance $H(x, y)$ is number of places where bit strings x and y differ (if $|x| \neq |y|$ then $H(x, y) = \infty$)
- $N_k(A) := \{x \in \{0, 1\}^* \mid H(x, y) \leq k \text{ for some } y \in A\}$

Definitions

- Hamming distance $H(x, y)$ is number of places where bit strings x and y differ (if $|x| \neq |y|$ then $H(x, y) = \infty$)
- $N_k(A) := \{x \in \{0,1\}^* \mid H(x, y) \leq k \text{ for some } y \in A\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\forall k \in \mathbb{N}$, $N_k(A)$ is regular

Definitions

- Hamming distance $H(x, y)$ is number of places where bit strings x and y differ (if $|x| \neq |y|$ then $H(x, y) = \infty$)
- $N_k(A) := \{x \in \{0,1\}^* \mid H(x, y) \leq k \text{ for some } y \in A\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\forall k \in \mathbb{N}, N_k(A)$ is regular

Proof.

$D_k = \{x \in (\{0,1\} \times \{0,1\})^* \mid x \text{ contains at most } k \text{ pairs } (0,1) \text{ or } (1,0)\}$ is regular

Definitions

- Hamming distance $H(x, y)$ is number of places where bit strings x and y differ (if $|x| \neq |y|$ then $H(x, y) = \infty$)
- $N_k(A) := \{x \in \{0, 1\}^* \mid H(x, y) \leq k$ for some $y \in A\}$

Lemma

$A \subseteq \{0, 1\}^*$ is regular $\implies$ $\forall k \in \mathbb{N}$, $N_k(A)$ is regular

Proof.

$$\begin{aligned} D_k &= \{x \in (\{0,1\} \times \{0,1\})^* \mid x \text{ contains at most } k \text{ pairs } (0,1) \text{ or } (1,0)\} \quad \text{is regular} \\ &= \{x \in (\{0,1\} \times \{0,1\})^* \mid H(fst(x), snd(x)) \leq k\} \end{aligned}$$

Definitions

- Hamming distance $H(x, y)$ is number of places where bit strings x and y differ (if $|x| \neq |y|$ then $H(x, y) = \infty$)
- $N_k(A) := \{x \in \{0,1\}^* \mid H(x, y) \leq k \text{ for some } y \in A\}$

Lemma

$A \subseteq \{0,1\}^*$ is regular $\implies$ $\forall k \in \mathbb{N}, N_k(A)$ is regular

Proof.

$$
\begin{aligned}
D_k &= \{x \in (\{0,1\} \times \{0,1\})^* \mid x \text{ contains at most } k \text{ pairs } (0,1) \text{ or } (1,0)\} \quad \text{is regular} \\
&= \{x \in (\{0,1\} \times \{0,1\})^* \mid H(fst(x), snd(x)) \leq k\} \\
N_k(A) &= fst(snd^{-1}(A) \cap D_k)
\end{aligned}
$$

Example

- $A = \{0011\} \quad k = 2$
-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of
-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

0	1	1	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

0	1	1	1

0	0	0	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

0	1	1	1

0	0	0	1

0	0	1	0

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0011	1011	0111	0001	0010	1111
1001					

-

Example

- $A = \{0011\}$ $k = 2$
- $N_k(A)$ consists of

$\boxed{0}\boxed{0}\boxed{1}\boxed{1}$ $\boxed{1}\boxed{0}\boxed{1}\boxed{1}$ $\boxed{0}\boxed{1}\boxed{1}\boxed{1}$ $\boxed{0}\boxed{0}\boxed{0}\boxed{1}$ $\boxed{0}\boxed{0}\boxed{1}\boxed{0}$ $\boxed{1}\boxed{1}\boxed{1}\boxed{1}$

$\boxed{1}\boxed{0}\boxed{0}\boxed{1}$ $\boxed{1}\boxed{0}\boxed{1}\boxed{0}$

-

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

0	1	1	1

0	0	0	1

0	0	1	0

1	1	1	1

1	0	0	1

1	0	1	0

0	1	0	1

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0	0	1	1

1	0	1	1

0	1	1	1

0	0	0	1

0	0	1	0

1	1	1	1

1	0	0	1

1	0	1	0

0	1	0	1

0	1	1	0

-

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

$\boxed{0}\boxed{0}\boxed{1}\boxed{1}$ $\boxed{1}\boxed{0}\boxed{1}\boxed{1}$ $\boxed{0}\boxed{1}\boxed{1}\boxed{1}$ $\boxed{0}\boxed{0}\boxed{0}\boxed{1}$ $\boxed{0}\boxed{0}\boxed{1}\boxed{0}$ $\boxed{1}\boxed{1}\boxed{1}\boxed{1}$

$\boxed{1}\boxed{0}\boxed{0}\boxed{1}$ $\boxed{1}\boxed{0}\boxed{1}\boxed{0}$ $\boxed{0}\boxed{1}\boxed{0}\boxed{1}$ $\boxed{0}\boxed{1}\boxed{1}\boxed{0}$ $\boxed{0}\boxed{0}\boxed{0}\boxed{0}$

-

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0011	1011	0111	0001	0010	1111
1001	1010	0101	0110	0000	

- $snd^{-1}(A)$ consists of

0000	0001	0010	0011	0100	0101
0011	0011	0011	0011	0011	0011
0110	0111	1000	1001	1010	1011
0011	0011	0011	0011	0011	0011
1100	1101	1110	1111		
0011	0011	0011	0011		

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

$\boxed{0|0|1|1}$ $\boxed{1|0|1|1}$ $\boxed{0|1|1|1}$ $\boxed{0|0|0|1}$ $\boxed{0|0|1|0}$ $\boxed{1|1|1|1}$

$\boxed{1|0|0|1}$ $\boxed{1|0|1|0}$ $\boxed{0|1|0|1}$ $\boxed{0|1|1|0}$ $\boxed{0|0|0|0}$

- $snd^{-1}(A) \cap D_k$ consists of

$\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&0&0&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&0&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 0&1&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

0011	1011	0111	0001	0010	1111
1001	1010	0101	0110	0000	

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

0000	0001	0010	0011		0101
0011	0011	0011	0011		0011
0110	0111		1001	1010	1011
0011	0011		0011	0011	0011
			1111		
			0011		

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

0011	1011	0111	0001	0010	1111
1001	1010	0101	0110		

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

0001	0010	0011	0101
0011	0011	0011	0011

0110	0111	1001	1010	1011
0011	0011	0011	0011	0011

1111
0011

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

$\begin{array}{|c|c|c|c|}\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&0&1&0\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&0&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&0\\\hline\end{array}$

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

$\begin{array}{|c|c|c|c|}\hline 0&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&0&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 0&1&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

0011	1011	0111		1111
1001	1010	0101	0110	

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

		0011 / 0011		0101 / 0011
0110 / 0011	0111 / 0011	1001 / 0011	1010 / 0011	1011 / 0011
		1111 / 0011		

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

$\boxed{1}\boxed{0}\boxed{1}\boxed{1}$ $\boxed{0}\boxed{1}\boxed{1}\boxed{1}$ $\boxed{1}\boxed{1}\boxed{1}\boxed{1}$

$\boxed{1}\boxed{0}\boxed{0}\boxed{1}$ $\boxed{1}\boxed{0}\boxed{1}\boxed{0}$ $\boxed{0}\boxed{1}\boxed{0}\boxed{1}$ $\boxed{0}\boxed{1}\boxed{1}\boxed{0}$

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

$\begin{array}{|c|c|c|c|}\hline 0&1&0&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 0&1&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

	1011	0111		1111	
1001	1010		0110		

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

0110	0111	1001	1010	1011	1111
0011	0011	0011	0011	0011	0011

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

$\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline\end{array}$

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

$\begin{array}{|c|c|c|c|}\hline 0&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&0&1\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&0\\\hline 0&0&1&1\\\hline\end{array}$ $\begin{array}{|c|c|c|c|}\hline 1&0&1&1\\\hline 0&0&1&1\\\hline\end{array}$

$\begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline 0&0&1&1\\\hline\end{array}$

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

1 0 1 1

1 1 1 1

1 0 0 1

1 0 1 0

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

1	0	0	1
0	0	1	1

1	0	1	0
0	0	1	1

1	0	1	1
0	0	1	1

1	1	1	1
0	0	1	1

Example

- $A = \{0011\}$ $k = 2$
- $N_k(A)$ consists of

1	0	1	1

1	1	1	1

1	0	1	0

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

1	0	1	0
0	0	1	1

1	0	1	1
0	0	1	1

1	1	1	1
0	0	1	1

Example

- $A = \{0011\}$ $\quad k = 2$
- $N_k(A)$ consists of

1	0	1	1

1	1	1	1

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

1	0	1	1
0	0	1	1

1	1	1	1
0	0	1	1

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of

1	1	1	1

- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

1	1	1	1
0	0	1	1

Example

- $A = \{0011\} \quad k = 2$
- $N_k(A)$ consists of
- $fst(snd^{-1}(A) \cap D_k) = N_k(A)$

Thanks! & Questions?