Lexical Analysis

Eduardo Souza, Guido Wachsmuth, Eelco Visser

Overview

today's lecture

Overview

today's lecture

Lexical analysis

Overview

today's lecture

Lexical analysis
Regular languages

- regular grammars
- regular expressions
- finite state automata

Overview

today's lecture

Lexical analysis

Regular languages

- regular grammars
- regular expressions
- finite state automata

Equivalence of formalisms

- constructive approach

Overview
today's lecture

Lexical analysis
Regular languages

- regular grammars
- regular expressions
- finite state automata

Equivalence of formalisms

- constructive approach

Tool generation

I

Regular Grammars

Recap: A Theory of Language

formal languages

Recap: A Theory of Language

formal languages

vocabulary Σ

- finite, nonempty set of elements (words, letters)
- alphabet

Recap: A Theory of Language

formal languages

vocabulary Σ

- finite, nonempty set of elements (words, letters)
- alphabet

string over Σ

- finite sequence of elements chosen from Σ
- word, sentence, utterance

Recap: A Theory of Language

formal languages

vocabulary Σ

- finite, nonempty set of elements (words, letters)
- alphabet

string over Σ

- finite sequence of elements chosen from Σ
- word, sentence, utterance

formal language λ

- set of strings over a vocabulary Σ
- $\lambda \subseteq \Sigma^*$

Recap: A Theory of Language

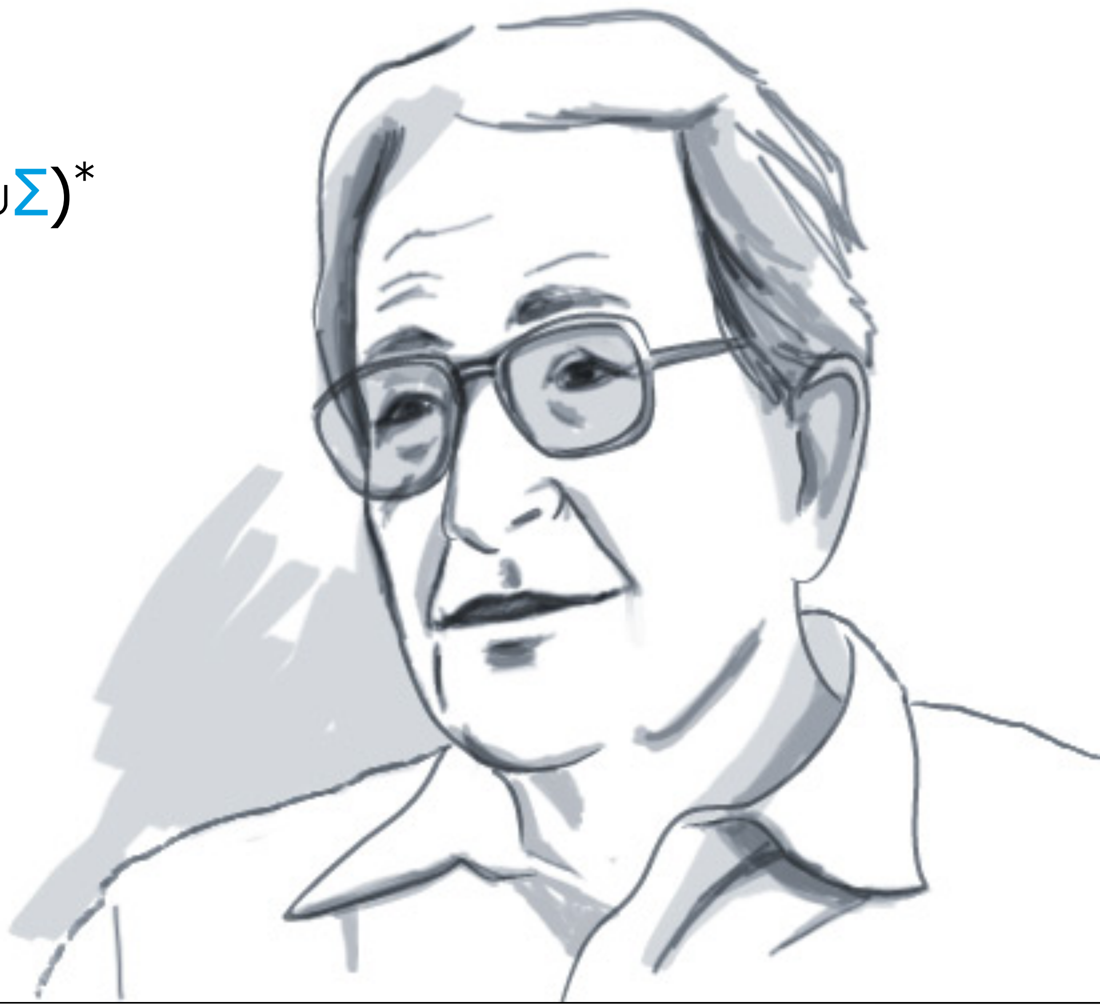
formal grammars

Recap: A Theory of Language

formal grammars

formal grammar $G = (N, \Sigma, P, S)$

- nonterminal symbols N
- terminal symbols Σ
- production rules $P \subseteq (N \cup \Sigma)^* \, N \, (N \cup \Sigma)^* \times (N \cup \Sigma)^*$
- start symbol $S \in N$



Recap: A Theory of Language

formal grammars

formal grammar $G = (N, \Sigma, P, S)$

nonterminal symbols N

nonterminal symbol

terminal symbols Σ

production rules $P \subseteq (N \cup \Sigma)^* \boxed{N} (N \cup \Sigma)^* \times (N \cup \Sigma)^*$

start symbol $S \in N$

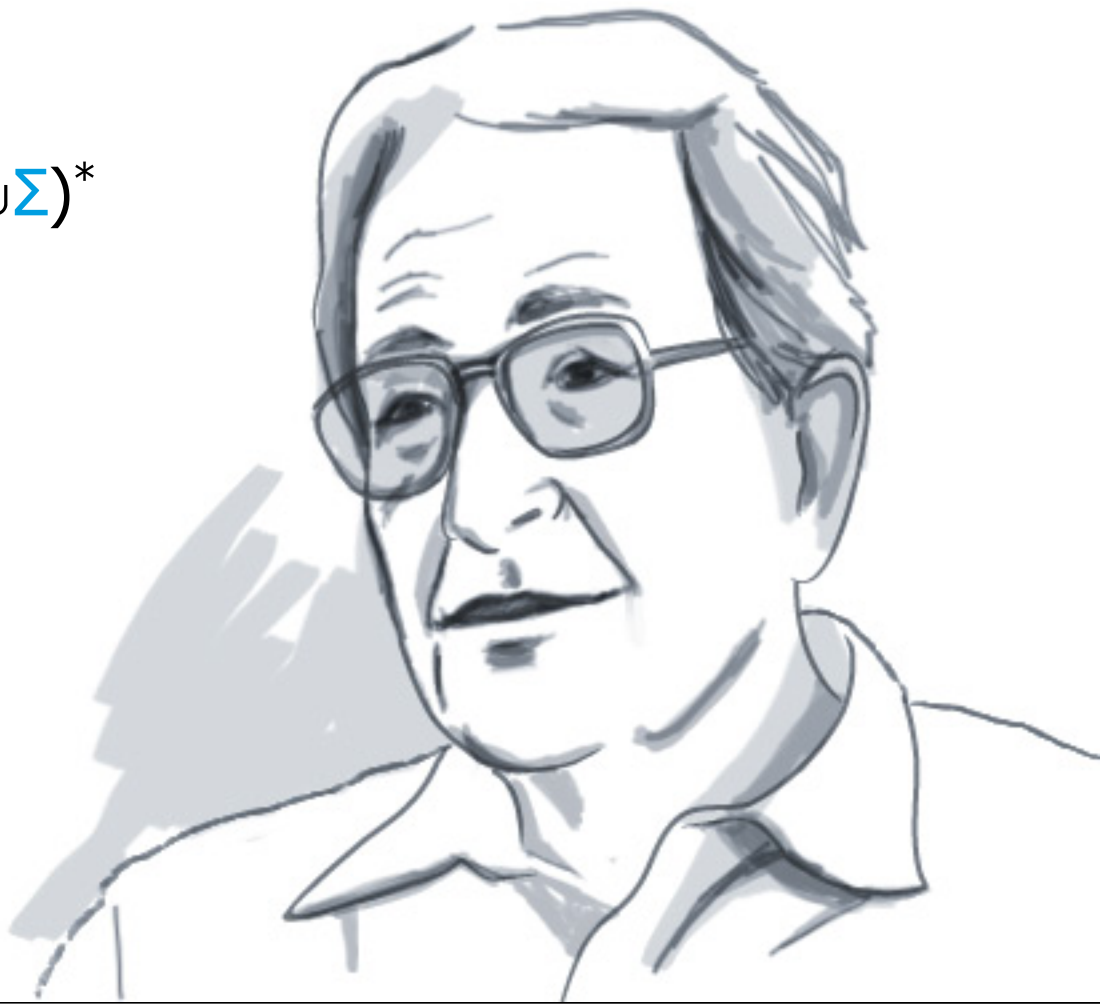
Recap: A Theory of Language

formal grammars

formal grammar $G = (N, \Sigma, P, S)$

- nonterminal symbols N
- terminal symbols Σ
- production rules $P \subseteq \boxed{(N \cup \Sigma)^*}\, N\, \boxed{(N \cup \Sigma)^*} \times (N \cup \Sigma)^*$
- start symbol $S \in N$

context



Recap: A Theory of Language
formal grammars

formal grammar $G = (N, \Sigma, P, S)$

- nonterminal symbols N
- terminal symbols Σ
- production rules $P \subseteq (N \cup \Sigma)^* \, N \, (N \cup \Sigma)^* \times \boxed{(N \cup \Sigma)^*}$
- start symbol $S \in N$

replacement

Recap: A Theory of Language

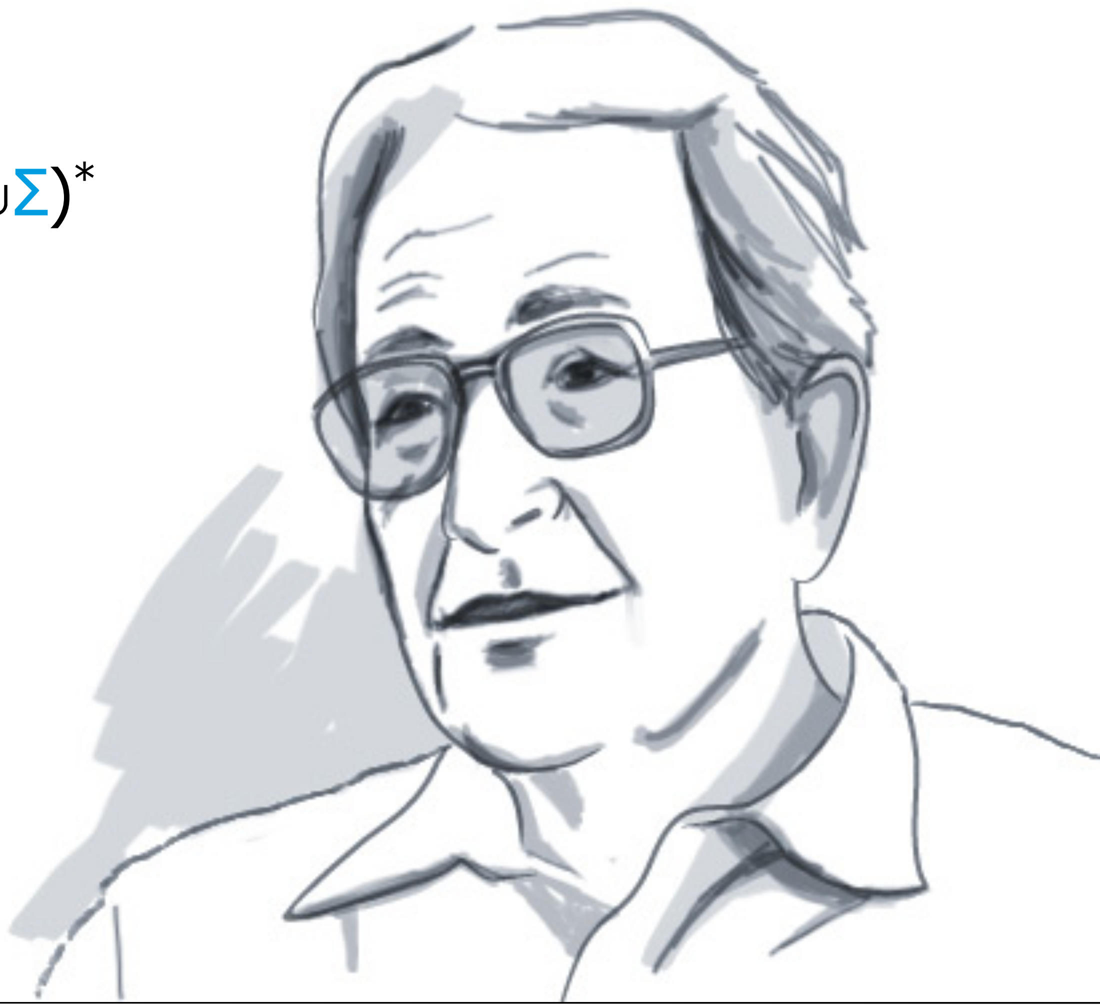
formal grammars

formal grammar $G = (N, \Sigma, P, S)$

- nonterminal symbols N
- terminal symbols Σ
- production rules $P \subseteq (N \cup \Sigma)^* \, N \, (N \cup \Sigma)^* \times (N \cup \Sigma)^*$
- start symbol $S \in N$

grammar classes

- type-0, unrestricted
- type-1, context-sensitive: (a A c, a b c)
- type-2, context-free: $P \subseteq N \times (N \cup \Sigma)^*$
- type-3, regular: (A, x) or (A, xB)



Decimal Numbers

right regular grammar

Num → "0" Num
Num → "1" Num
Num → "2" Num
Num → "3" Num
Num → "4" Num
Num → "5" Num
Num → "6" Num
Num → "7" Num
Num → "8" Num
Num → "9" Num

Num → "0"
Num → "1"
Num → "2"
Num → "3"
Num → "4"
Num → "5"
Num → "6"
Num → "7"
Num → "8"
Num → "9"

Identifiers

right regular grammar

Id → "a" R
…
Id → "z" R

R → "a" R
…
R → "z" R
R → "0" R
…
R → "9" R

Id → "a"
…
Id → "z"

R → "a"
…
R → "z"
R → "0"
…
R → "9"

Recap: A Theory of Language

formal languages

Recap: A Theory of Language

formal languages

formal grammar $G = (N, \Sigma, P, S)$

Recap: A Theory of Language

formal languages

formal grammar $G = (N, \Sigma, P, S)$

derivation relation $\Rightarrow_G \subseteq (N \cup \Sigma)^* \times (N \cup \Sigma)^*$

$w \Rightarrow_G w' \Leftrightarrow$

$\exists (p, q) \in P: \exists u,v \in (N \cup \Sigma)^*:$

$w = u\ p\ v \wedge w' = u\ q\ v$

Recap: A Theory of Language

formal languages

formal grammar $G = (N, \Sigma, P, S)$

derivation relation $\Rightarrow_G \subseteq (N \cup \Sigma)^* \times (N \cup \Sigma)^*$

$w \Rightarrow_G w' \Leftrightarrow$

$\exists (p, q) \in P: \exists u, v \in (N \cup \Sigma)^*:$

$w = u\ p\ v \wedge w' = u\ q\ v$

formal language $L(G) \subseteq \Sigma^*$

$L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$

Recap: A Theory of Language

formal languages

formal grammar $G = (N, \Sigma, P, S)$

derivation relation $\Rightarrow_G \subseteq (N \cup \Sigma)^* \times (N \cup \Sigma)^*$

$w \Rightarrow_G w' \Leftrightarrow$

$\exists (p, q) \in P: \exists u, v \in (N \cup \Sigma)^*:$

$w = u\ p\ v \wedge w' = u\ q\ v$

formal language $L(G) \subseteq \Sigma^*$

$L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$

classes of formal languages

Example

Is G regular?
L(G) = ?

Example

G:

Is G regular?
L(G) = ?

Example

G:

Is G regular?

L(G) = ?

Example

G:

$S \rightarrow a$

Is G regular?

$L(G) = ?$

Example

G:

$S \rightarrow a$
$S \rightarrow aA$

Is G regular?
L(G) = ?

Example

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$

Is G regular?
L(G) = ?

Example

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$
$B \rightarrow aC$

Is G regular?
L(G) = ?

Example

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$
$B \rightarrow aC$
$C \rightarrow aD$

Is G regular?
L(G) = ?

Example

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$
$B \rightarrow aC$
$C \rightarrow aD$
$D \rightarrow aS$

Is G regular?
L(G) = ?

Example

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$
$B \rightarrow aC$
$C \rightarrow aD$
$D \rightarrow aS$

Is G regular?
L(G) = ?

II

Regular Expressions

Recap: Regular Expressions

overview

basics

- symbol from an alphabet
- ε

combinators

- alternation: E1 | E2
- concatenation: E1 E2
- repetition: E*
- optional: E? = E | ε
- one or more: E+ = E E*

Decimal Numbers & Identifiers

regular expressions

Num: (0|1|2|3|4|5|6|7|8|9)+

Id: (a|…|z)(a|…|z|0|…|9)*

Regular Expressions
formal languages

basics

- L(a) = {"a"}
- L(ε) = {""}

combinators

- L(E1 | E2) = L(E1) ∪ L(E2)
- L(E1 E2) = L(E1) · L(E2)
- L(E*) = L(E)*

Decimal Numbers & Identifiers

regular expressions

Num: (0|1|2|3|4|5|6|7|8|9)+

A valid Num is any word that consists of a sequence of one or more digits.

Id: (a|…|z)(a|…|z|0|…|9)*

A valid Id is any word that starts with a lowercase letter followed by sequence of zero or more lowercase letters or digits.

Example

Example

G_r:

Example

G_r:

Example

G_r:

$S \rightarrow ('A'= \mid \dots \mid 'Z')^*(0 \mid \dots \mid 9)^*$

Example

G_r:

$S \rightarrow ('A'= \mid \ldots \mid 'Z')^*(0 \mid \ldots \mid 9)^*$

Example

G_r:

$S \rightarrow$ ('A'= | … | 'Z')*(0 | … | 9)*

$L(G_r) = ?$

Example

Example

G_r:

Example

G_r:

Example

G_r:

$S \rightarrow ('A'= | \dots | 'Z')^*(0 | \dots | 9)^*$

Example

G_r:

$S \rightarrow ('A'= | \dots | 'Z')^*(0 | \dots | 9)^*$

Example

G_r:

$S \rightarrow ('A'= | \ldots | 'Z')^*(0 | \ldots | 9)^*$

$L(G_r)$ = The set of words that start with zero or more capital letters, followed by zero or more decimal digits.

III

Finite Automata

Finite Automata

formal definition

Finite Automata

formal definition

finite automaton $M = (Q, \Sigma, T, q_0, F)$

states Q

input symbols Σ

transition function T

start state $q_0 \in Q$

final states $F \subseteq Q$

Finite Automata
formal definition

finite automaton $M = (Q, \Sigma, T, q_0, F)$

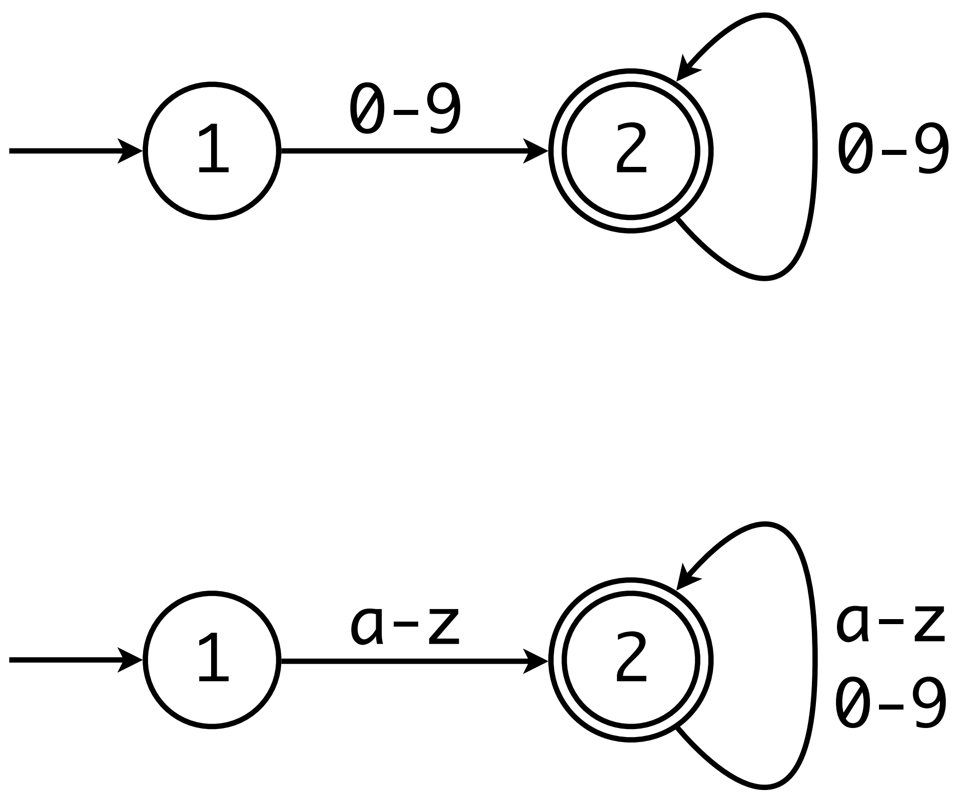
- states Q
- input symbols Σ
- transition function T
- start state $q_0 \in Q$
- final states $F \subseteq Q$

transition function

- nondeterministic FA $T : Q \times \Sigma \rightarrow P(Q)$
- NFA with ε-moves $T : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$
- deterministic FA $T : Q \times \Sigma \rightarrow Q$

Decimal Numbers & Identifiers

finite automata



Nondeterministic Finite Automata

formal languages

Nondeterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

Nondeterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

transition function $T : Q \times \Sigma \rightarrow P(Q)$

$T(\{q_1,\ldots, q_n\}, x) := T(q_1, x) \cup \ldots \cup T(q_n, x)$

$T^*(\{q_1,\ldots, q_n\}, \varepsilon) := \{q_1,\ldots, q_n\}$

$T^*(\{q_1,\ldots, q_n\}, xw) := T^*(T(\{q_1,\ldots, q_n\}, x), w)$

Nondeterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

transition function $T : Q \times \Sigma \rightarrow P(Q)$

$T(\{q_1, \ldots, q_n\}, x) := T(q_1, x) \cup \ldots \cup T(q_n, x)$

$T^*(\{q_1, \ldots, q_n\}, \varepsilon) := \{q_1, \ldots, q_n\}$

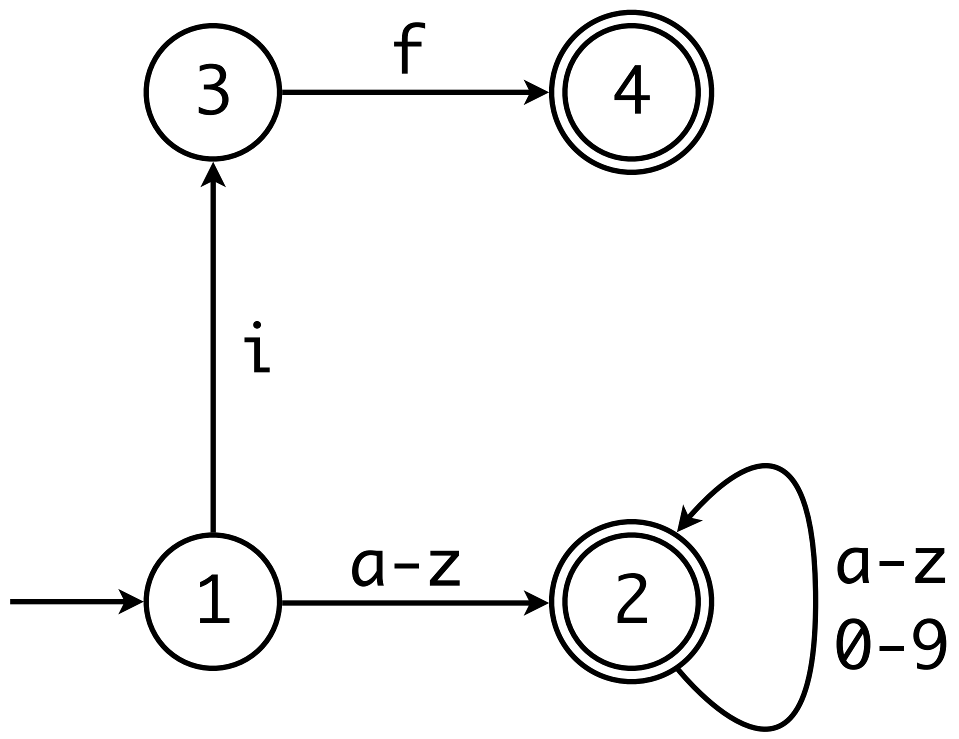
$T^*(\{q_1, \ldots, q_n\}, xw) := T^*(T(\{q_1, \ldots, q_n\}, x), w)$

formal language $L(M) \subseteq \Sigma^*$

$L(M) = \{w \in \Sigma^* \mid T^*(\{q_0\}, w) \cap F \neq \emptyset\}$

Nondeterministic Finite Automata

formal languages



Deterministic Finite Automata

formal languages

Deterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

Deterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

transition function $T : Q \times \Sigma \rightarrow Q$

$T^*(q, \varepsilon) := q$

$T^*(q, xw) := T^*(T(q, x), w)$

Deterministic Finite Automata

formal languages

finite automaton $M = (Q, \Sigma, T, q_0, F)$

transition function $T : Q \times \Sigma \rightarrow Q$

$T^*(q, \varepsilon) := q$

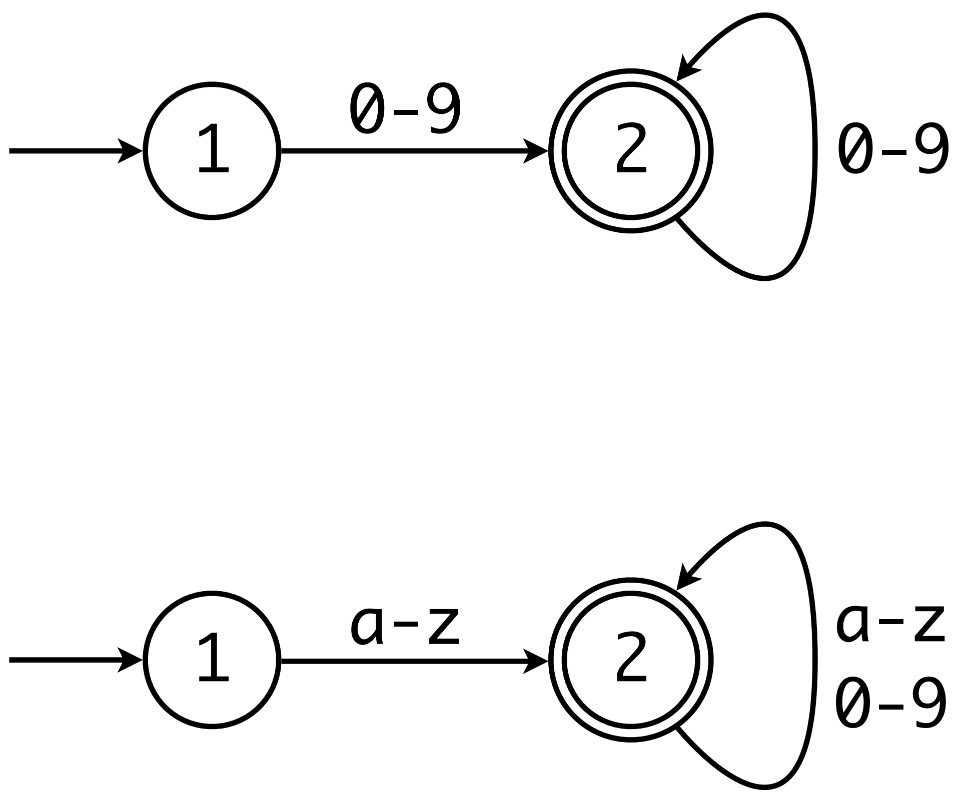
$T^*(q, xw) := T^*(T(q, x), w)$

formal language $L(M) \subseteq \Sigma^*$

$L(M) = \{w \in \Sigma^* \mid T^*(q_0, w) \in F\}$

Deterministic Finite Automata

formal languages



IV

Equivalence

Regular Languages

formalisms

left regular grammars

right regular grammars

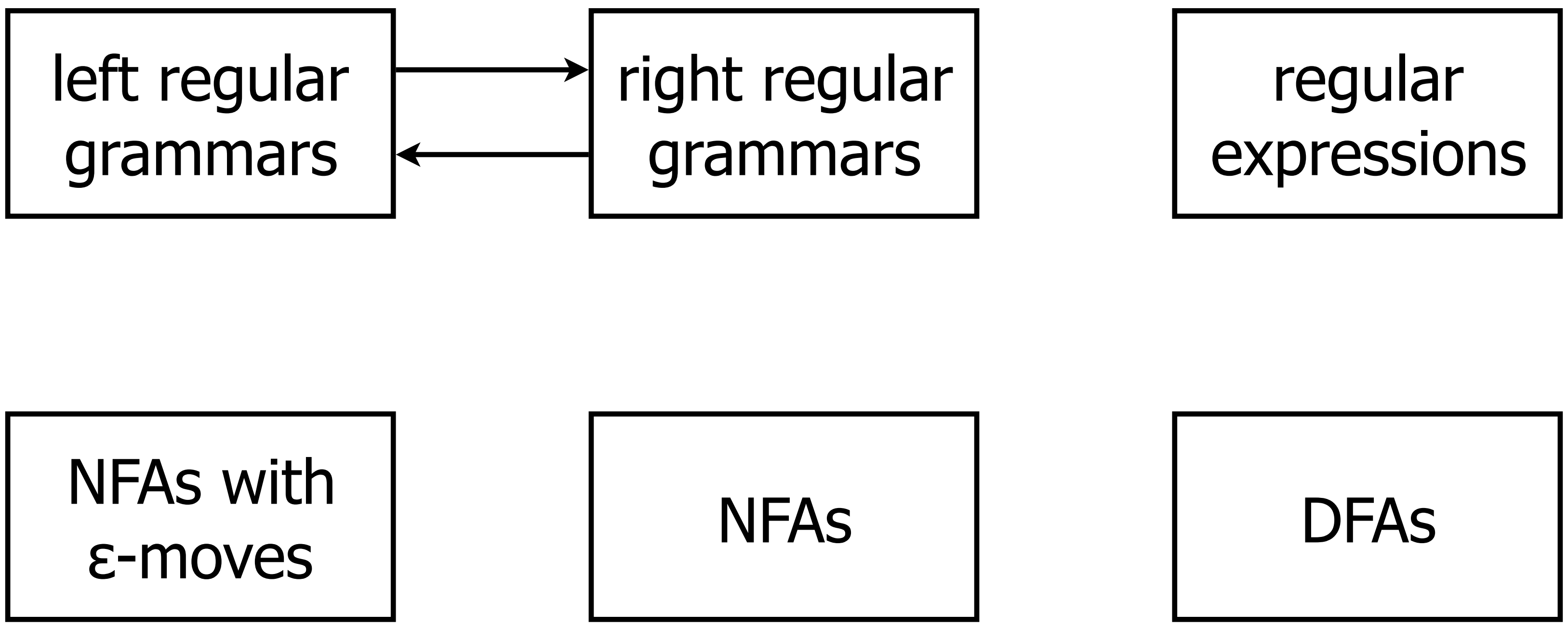
regular expressions

NFAs with ε-moves

NFAs

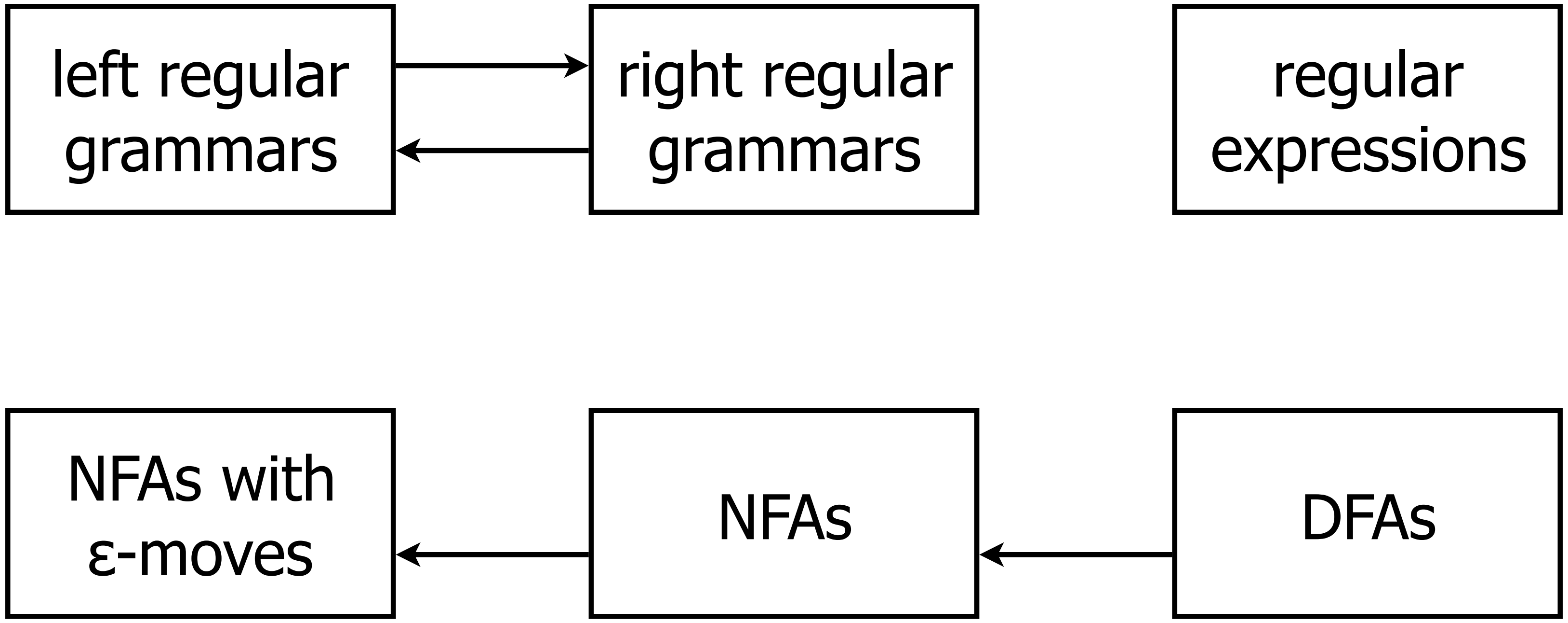
DFAs

Regular Languages
formalisms

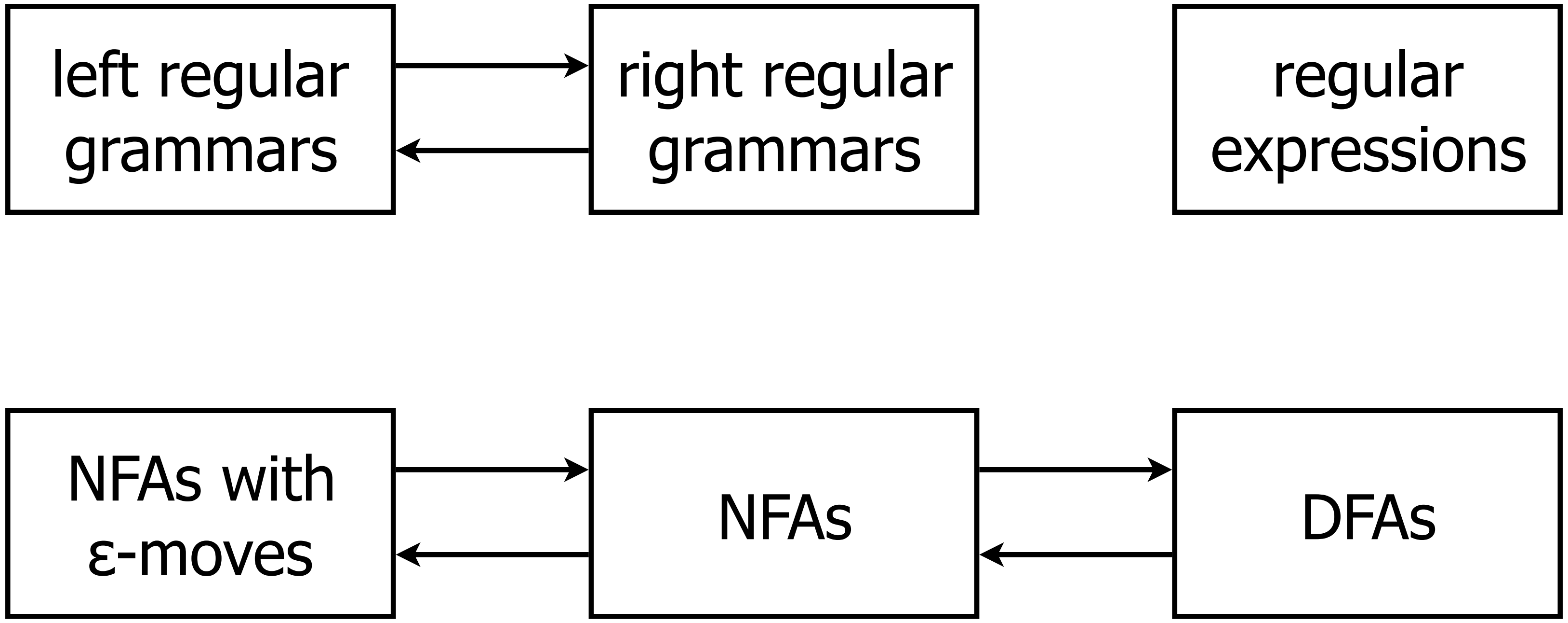


Regular Languages

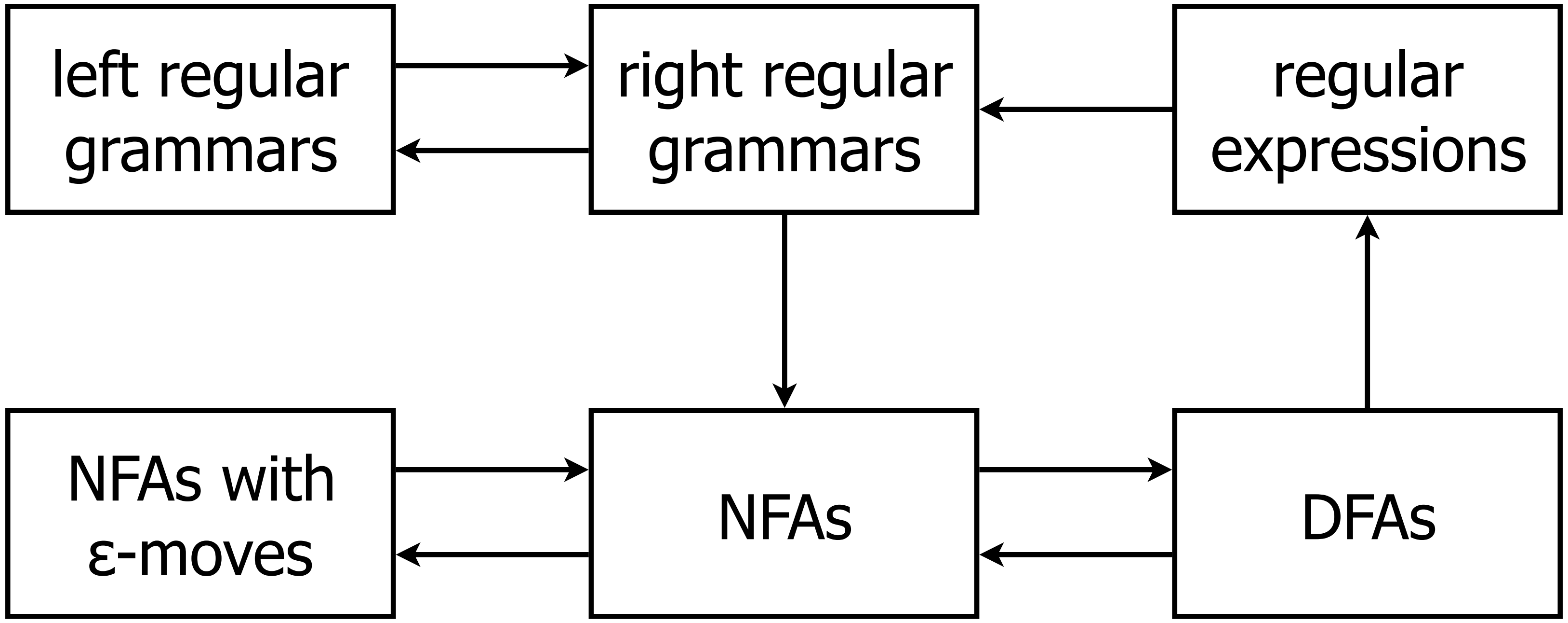
formalisms



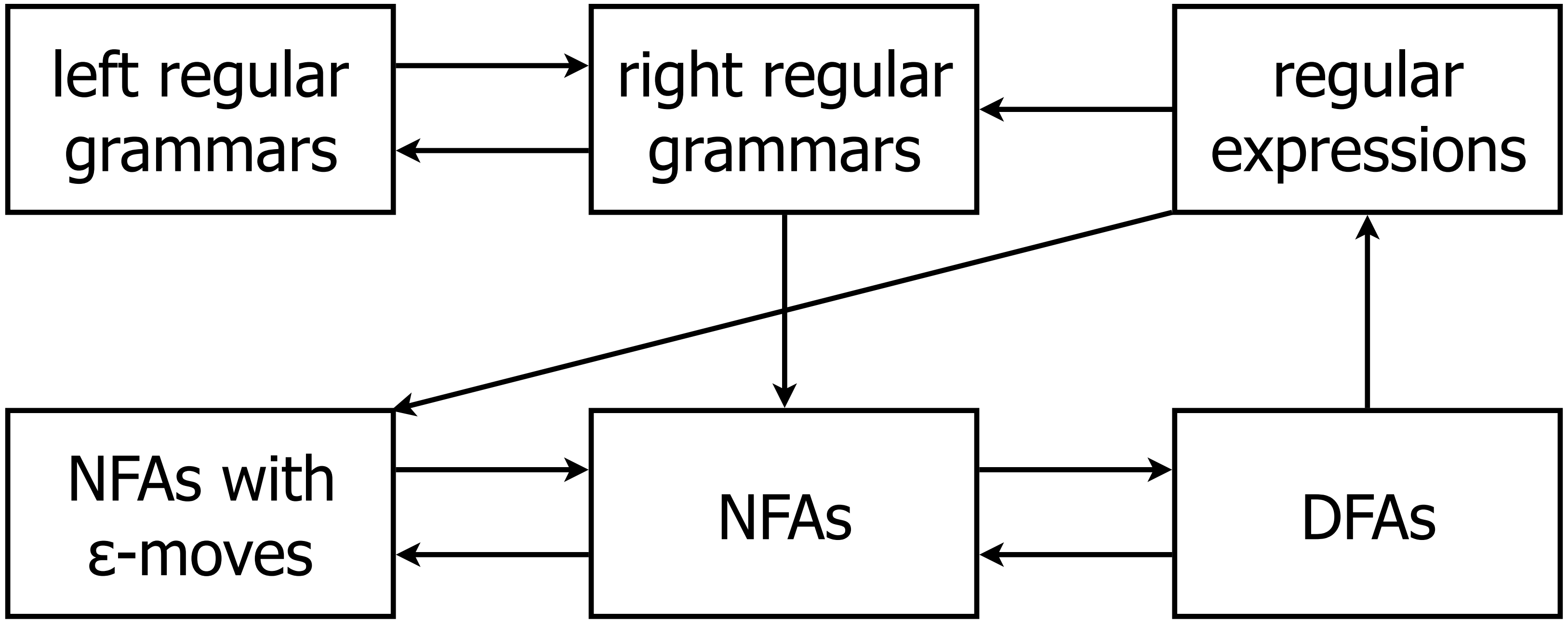
Regular Languages
formalisms



Regular Languages
formalisms

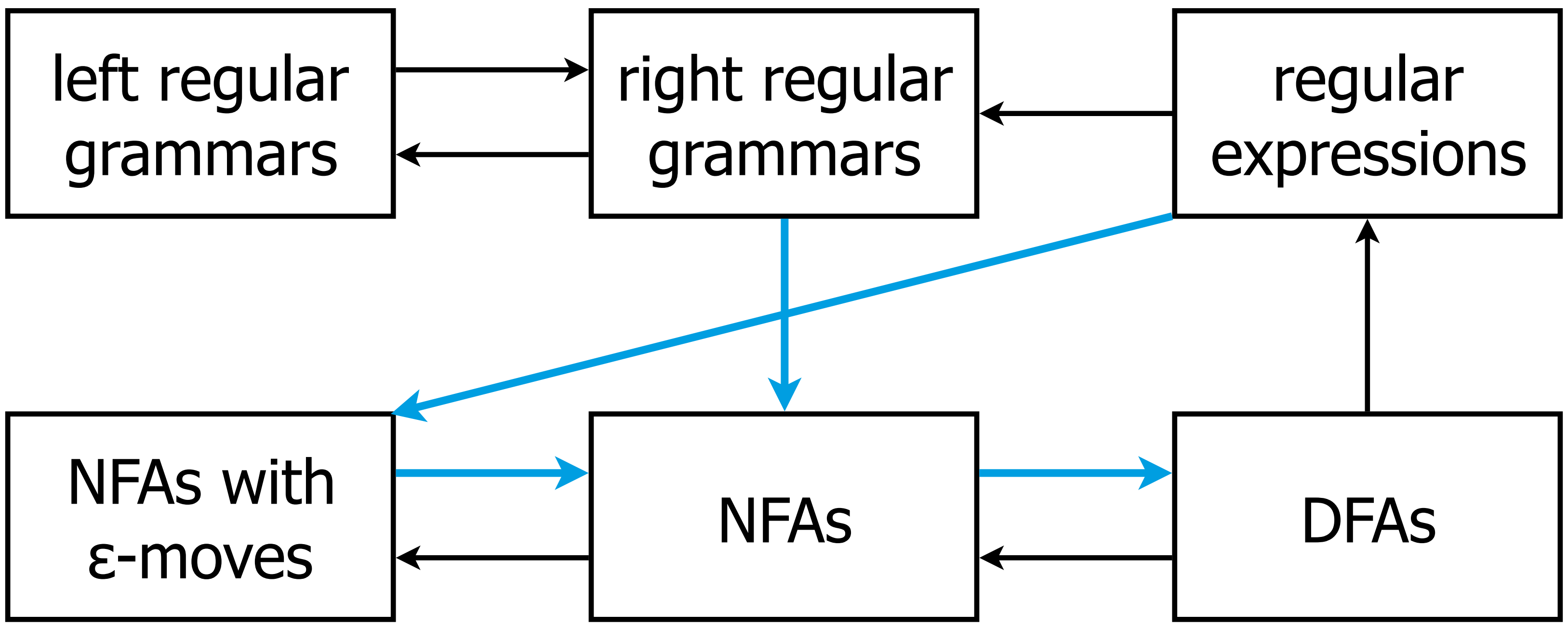


Regular Languages
formalisms



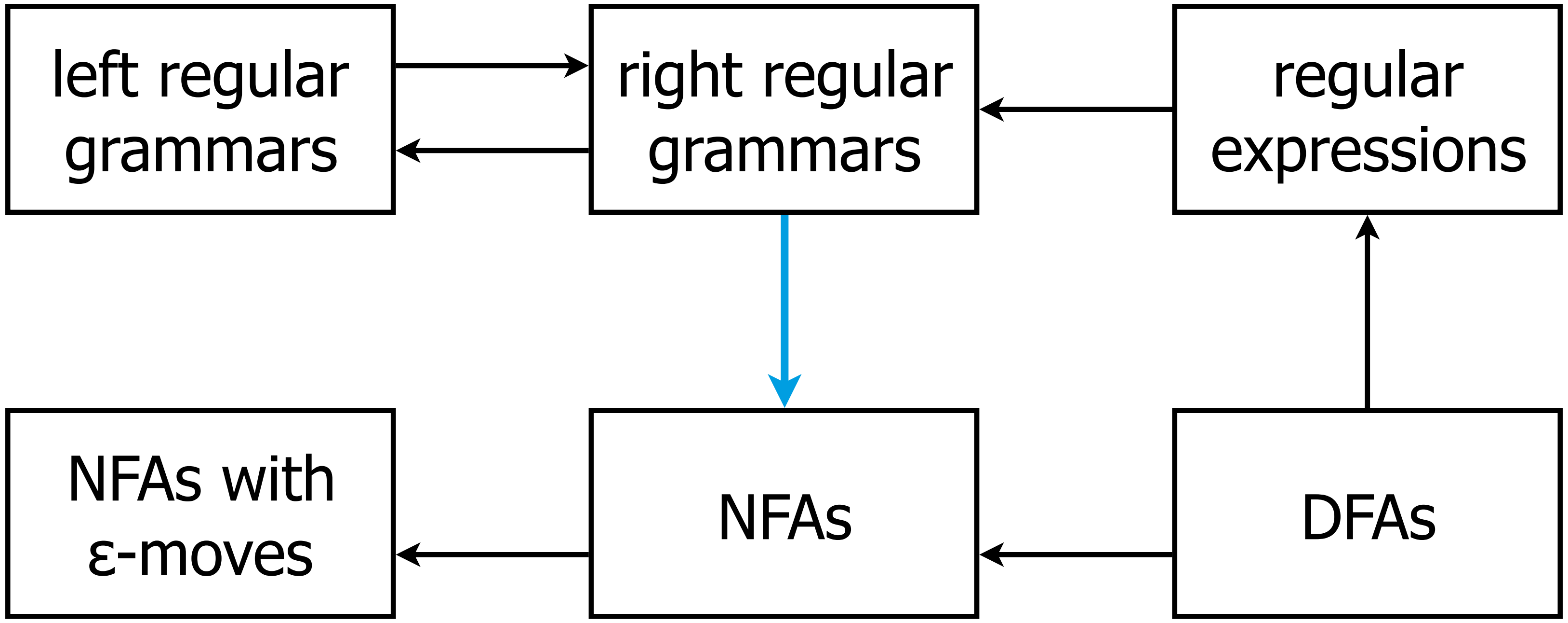
Regular Languages

formalisms



Regular Languages

formalisms



NFA construction

right regular grammar

NFA construction

right regular grammar

formal grammar $G = (N, \Sigma, P, S)$

NFA construction

right regular grammar

formal grammar $G = (N, \Sigma, P, S)$

finite automaton $M = (N \cup \{f\}, \Sigma, T, S, F)$

NFA construction

right regular grammar

formal grammar $G = (N, \Sigma, P, S)$

finite automaton $M = (N \cup \{f\}, \Sigma, T, S, F)$

transition function T

$(X, aY) \in P : (X, a, Y) \in T$

$(X, a) \in P : (X, a, f) \in T$

NFA construction

right regular grammar

formal grammar $G = (N, \Sigma, P, S)$

finite automaton $M = (N \cup \{f\}, \Sigma, T, S, F)$

transition function T

$(X, aY) \in P : (X, a, Y) \in T$

$(X, a) \in P : (X, a, f) \in T$

final states F

$(S, \varepsilon) \in P : F = \{S, f\}$

else: $F = \{f\}$

NFA construction

Example

Num → "0" Num	Num → "0"
Num → "1" Num	Num → "1"
Num → "2" Num	Num → "2"
Num → "3" Num	Num → "3"
Num → "4" Num	Num → "4"
Num → "5" Num	Num → "5"
Num → "6" Num	Num → "6"
Num → "7" Num	Num → "7"
Num → "8" Num	Num → "8"
Num → "9" Num	Num → "9"

NFA construction

Example

Num → "0" Num
Num → "1" Num
Num → "2" Num
Num → "3" Num
Num → "4" Num
Num → "5" Num
Num → "6" Num
Num → "7" Num
Num → "8" Num
Num → "9" Num

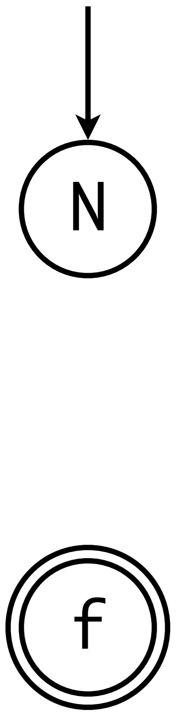
Num → "0"
Num → "1"
Num → "2"
Num → "3"
Num → "4"
Num → "5"
Num → "6"
Num → "7"
Num → "8"
Num → "9"

NFA construction

Example

Num → "0" Num
Num → "1" Num
Num → "2" Num
Num → "3" Num
Num → "4" Num
Num → "5" Num
Num → "6" Num
Num → "7" Num
Num → "8" Num
Num → "9" Num

Num → "0"
Num → "1"
Num → "2"
Num → "3"
Num → "4"
Num → "5"
Num → "6"
Num → "7"
Num → "8"
Num → "9"

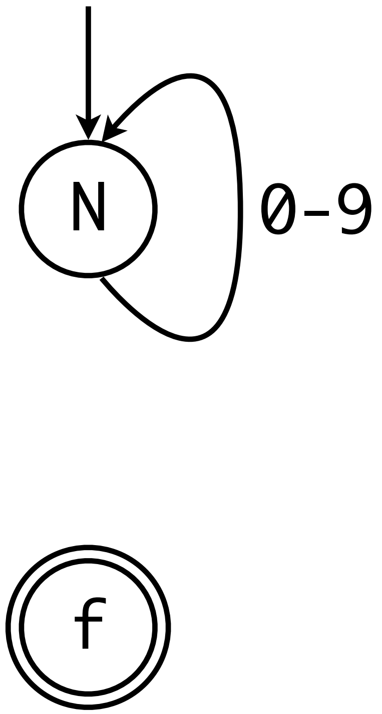


NFA construction

Example

Num → "0" Num
Num → "1" Num
Num → "2" Num
Num → "3" Num
Num → "4" Num
Num → "5" Num
Num → "6" Num
Num → "7" Num
Num → "8" Num
Num → "9" Num

Num → "0"
Num → "1"
Num → "2"
Num → "3"
Num → "4"
Num → "5"
Num → "6"
Num → "7"
Num → "8"
Num → "9"

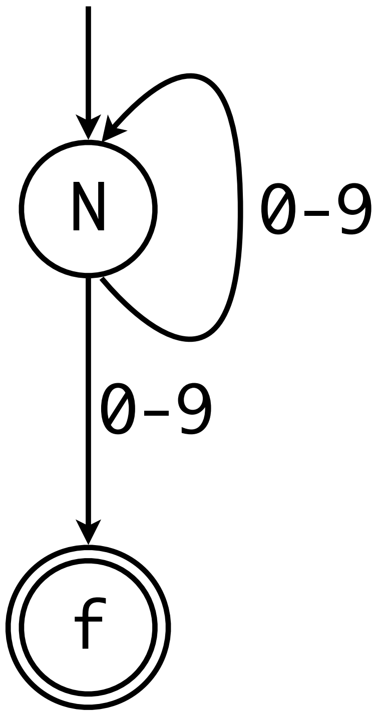


NFA construction

Example

Num → "0" Num
Num → "1" Num
Num → "2" Num
Num → "3" Num
Num → "4" Num
Num → "5" Num
Num → "6" Num
Num → "7" Num
Num → "8" Num
Num → "9" Num

Num → "0"
Num → "1"
Num → "2"
Num → "3"
Num → "4"
Num → "5"
Num → "6"
Num → "7"
Num → "8"
Num → "9"



NFA construction
Example

formal grammar $G = (N, \Sigma, P, S)$

finite automaton $M = (N \cup \{f\}, \Sigma, T, S, F)$

transition function T

$(X, aY) \in P : (X, a, Y) \in T$

$(X, a) \in P : (X, a, f) \in T$

final states F

$(S, \varepsilon) \in P : F = \{S, f\}$

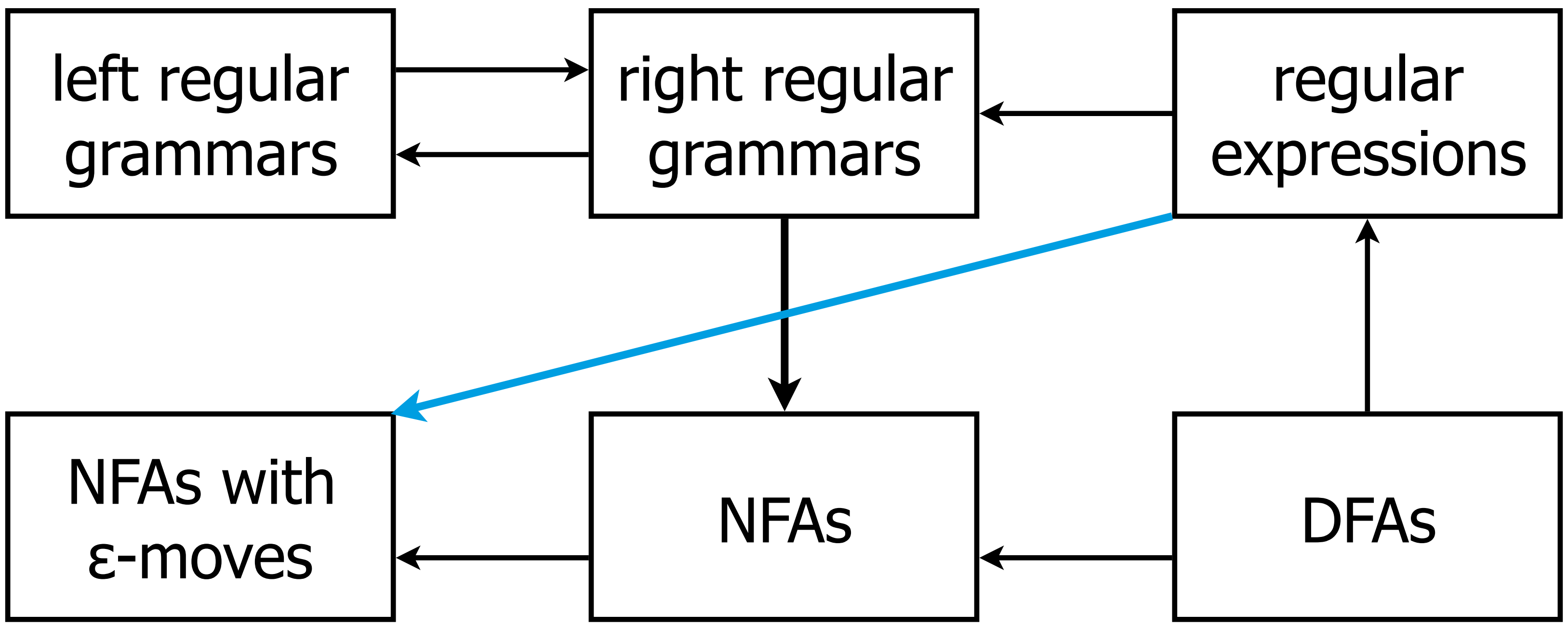
else: $F = \{f\}$

G:

$S \rightarrow a$
$S \rightarrow aA$
$A \rightarrow aB$
$B \rightarrow aC$
$C \rightarrow aD$
$D \rightarrow aS$

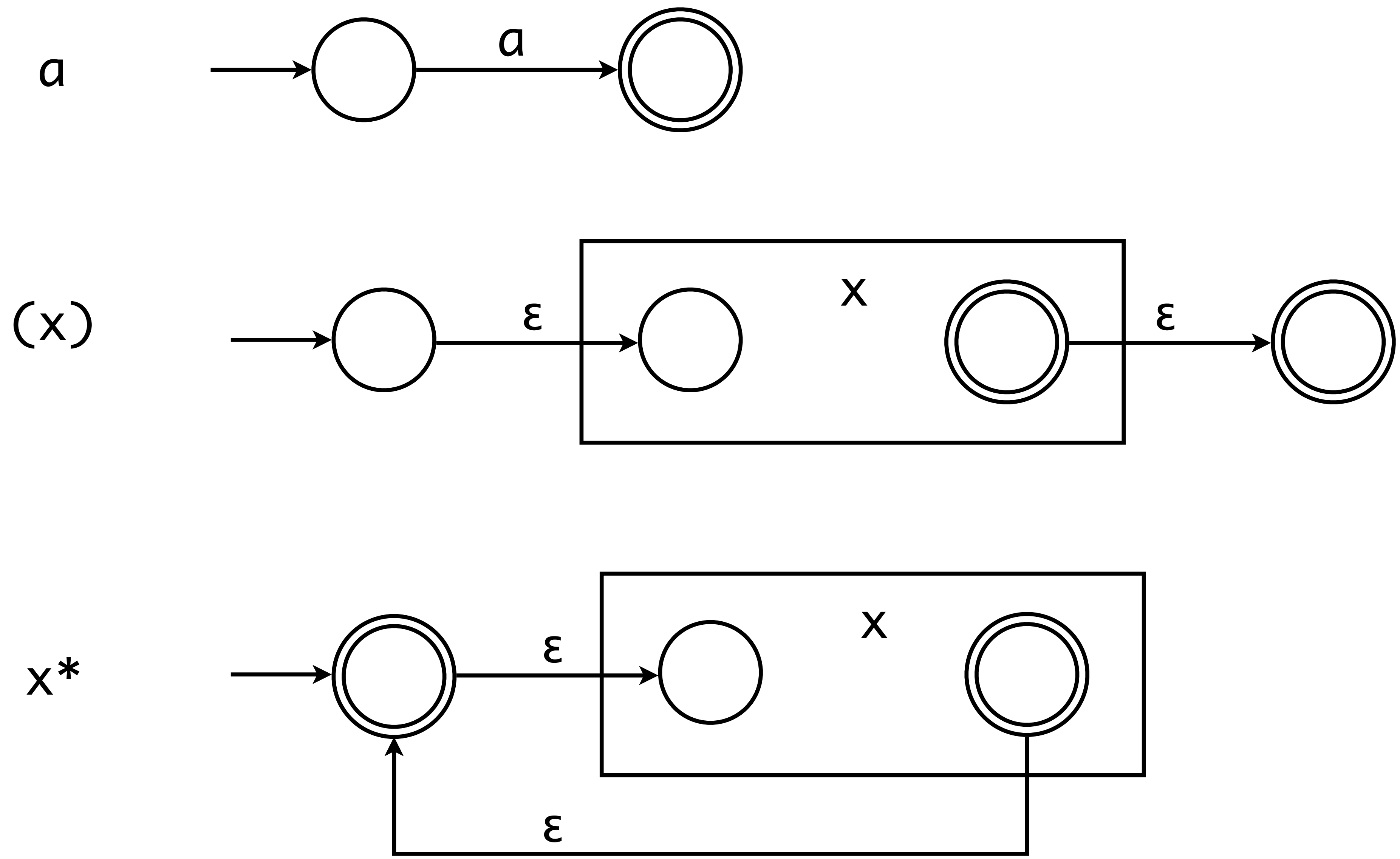
Regular Languages

formalisms



NFA construction

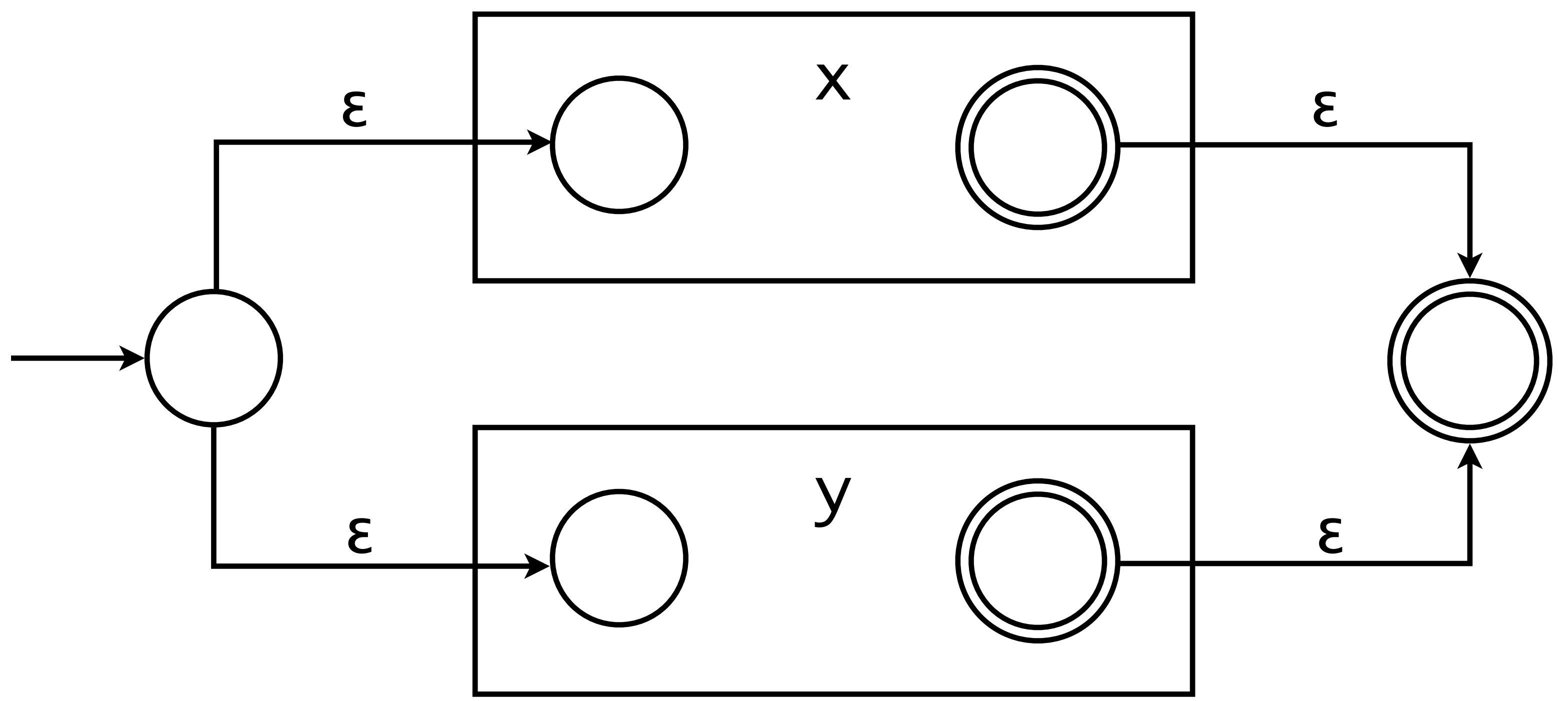
regular expressions

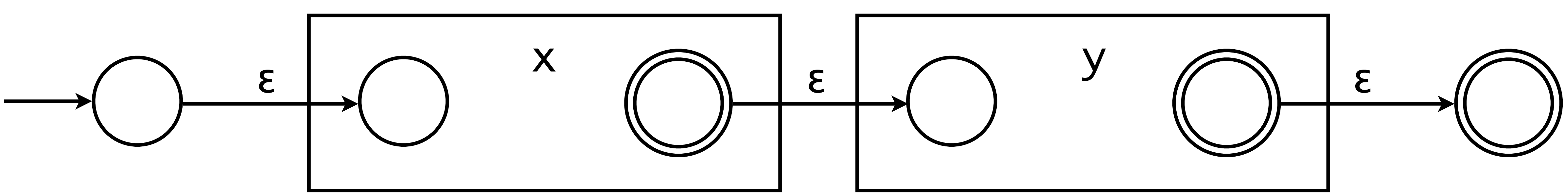


NFA construction

regular expressions

x|y





xy

NFA construction

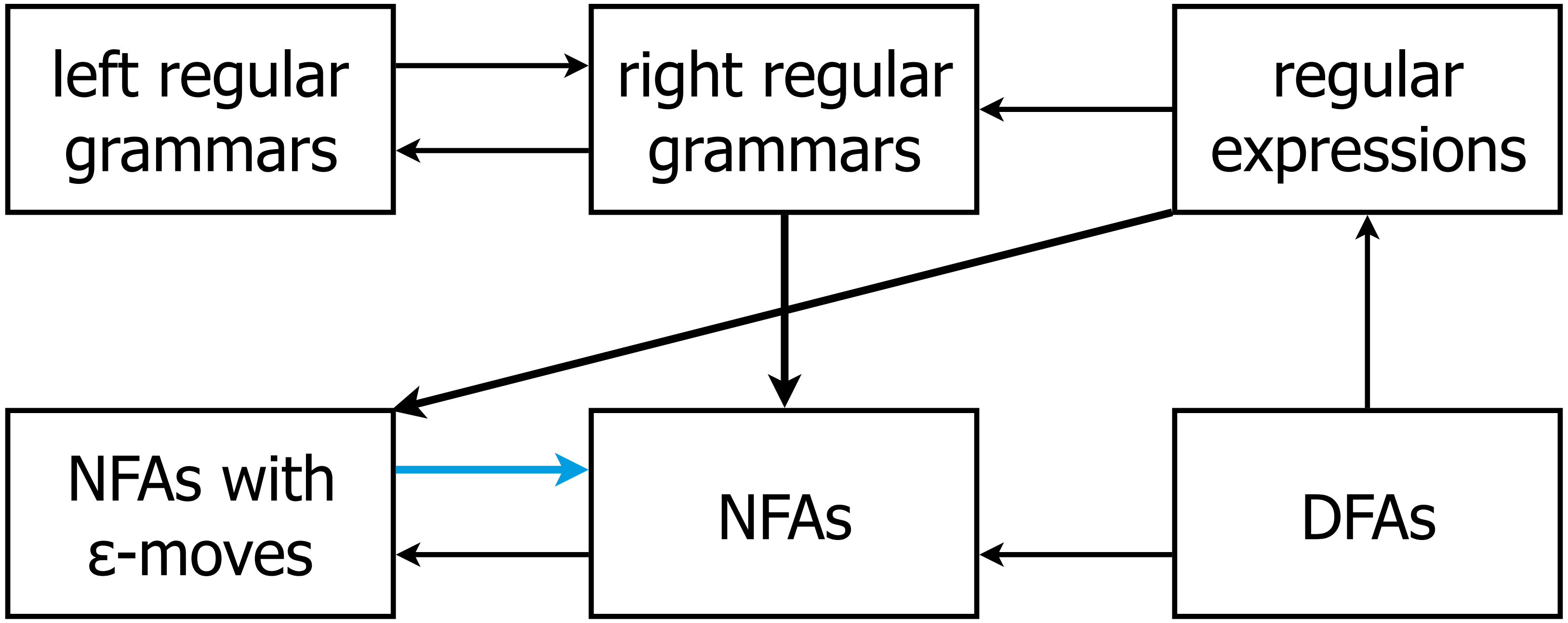
regular expressions

G_r:

S → ('A'= | … | 'Z')*(0 | … | 9)*

Regular Languages

formalisms



NFA construction

ε elimination

additional final states

- states with ε-moves into final states
- become final states themselves

additional transitions

- ε-move from source to target state
- transitions from target state
- add these transitions to the source state

NFA construction

ε elimination

G_r:

S → ('A'= | … | 'Z')*(0 | … | 9)*

additional final states

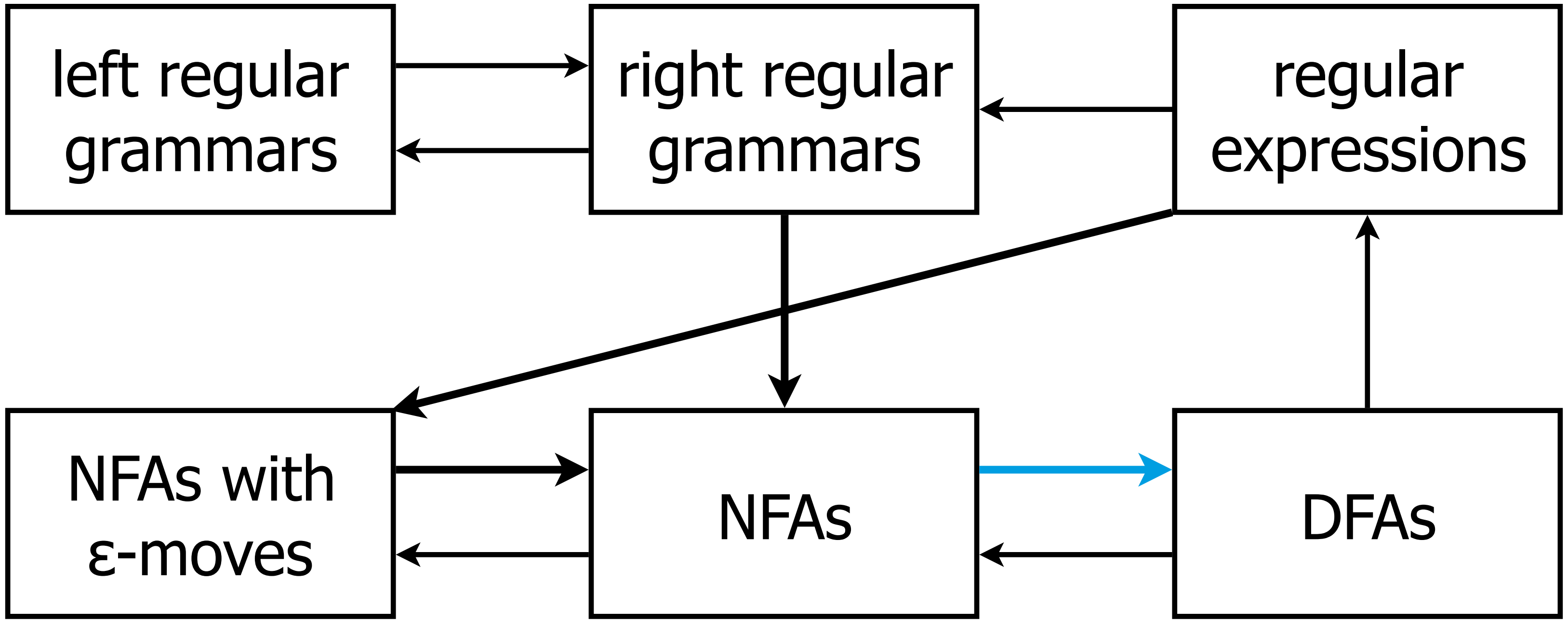
- states with ε-moves into final states
- become final states themselves

additional transitions

- ε-move from source to target state
- transitions from target state
- add these transitions to the source state

Regular Languages

formalisms



Powerset construction

eliminating nondeterminism

nondeterministic finite automaton $M = (Q, \Sigma, T, q_0, F)$

deterministic finite automaton $M' = (P(Q), \Sigma, T', \{q_0\}, F')$

transition function T'

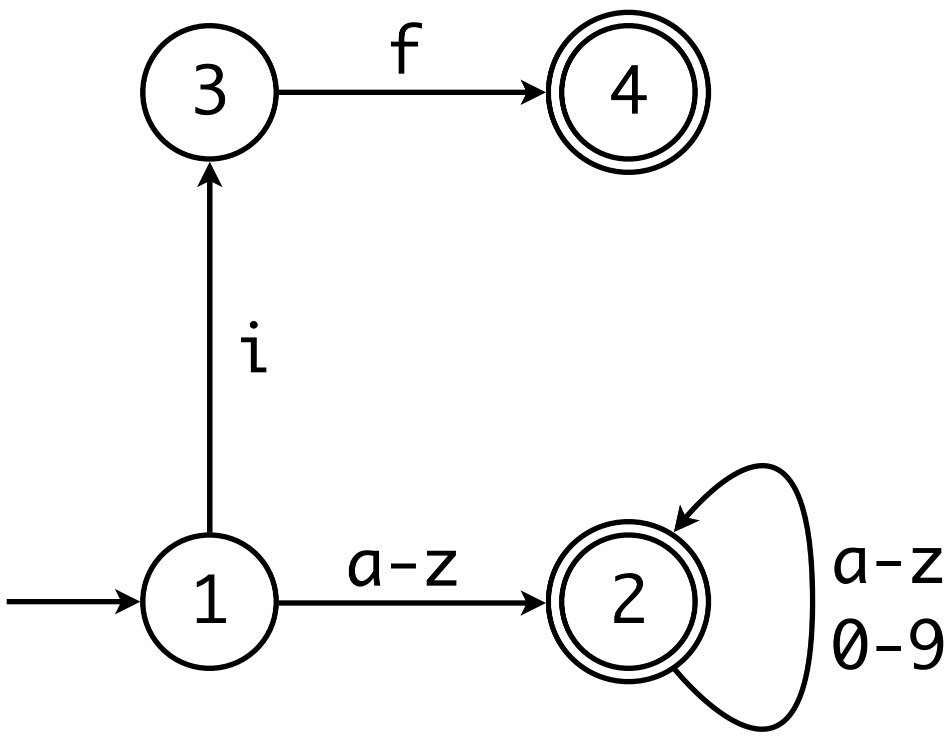
- $T'(\{q_1,\dots, q_n\}, x) = T(\{q_1,\dots, q_n\}, x) = T(q_1, x) \cup \dots \cup T(q_n, x)$

final states $F' = \{S \mid S \subseteq Q, S \cap F \neq \varnothing\}$

- all states that include a final state of the original NFA

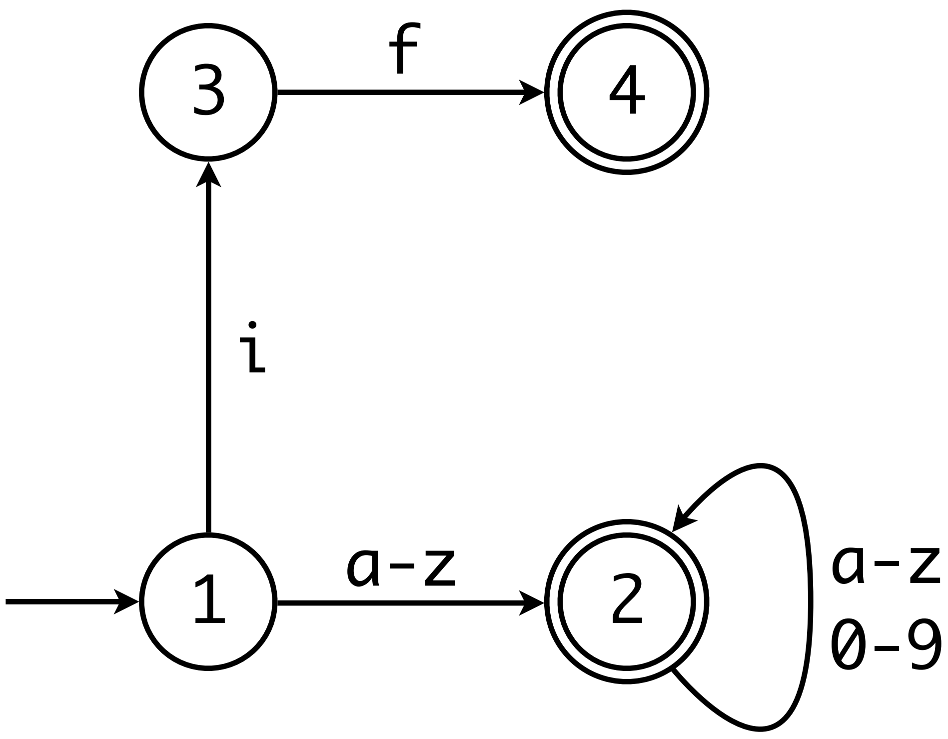
Powerset construction

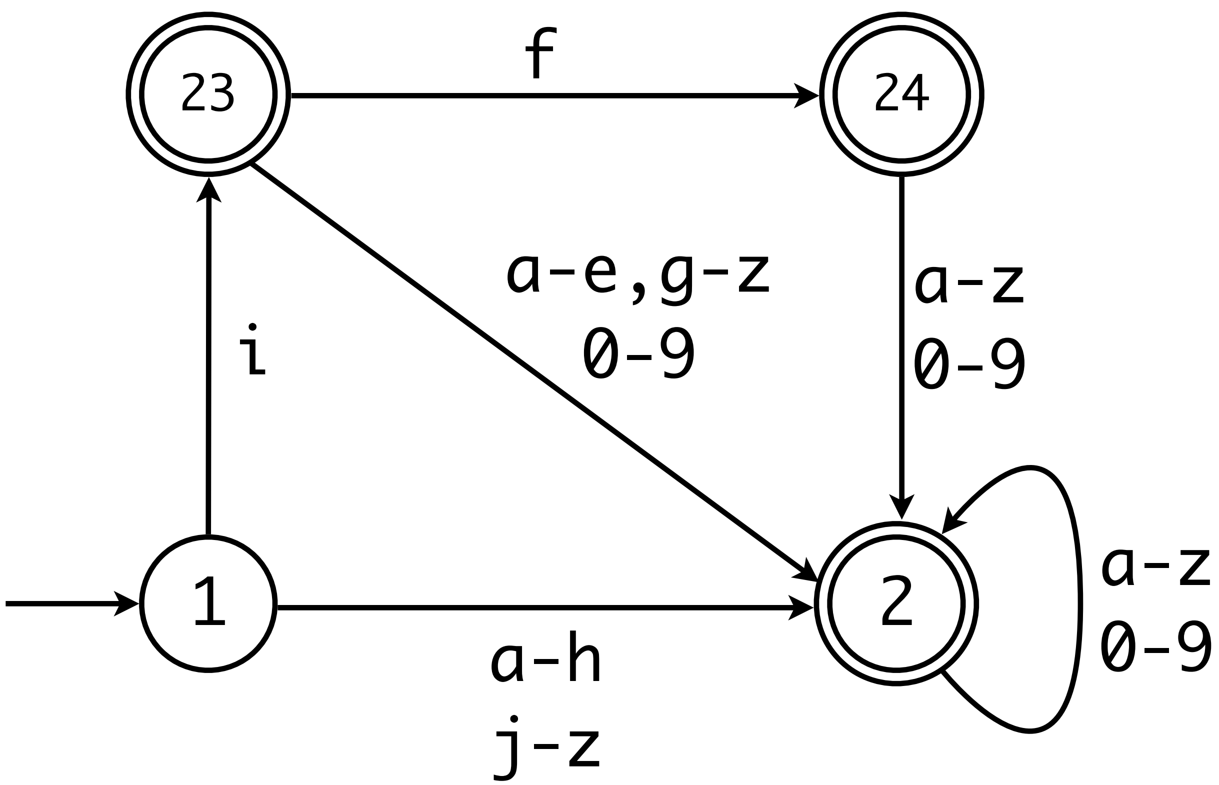
example



Powerset construction

example





V

Summary

Summary
lessons learned

Summary
lessons learned

What are the formalisms to describe regular languages?

- regular grammars
- regular expressions
- finite state automata

Summary
lessons learned

What are the formalisms to describe regular languages?

- regular grammars
- regular expressions
- finite state automata

Why are these formalisms equivalent?

- constructive proofs

Summary
lessons learned

What are the formalisms to describe regular languages?

- regular grammars
- regular expressions
- finite state automata

Why are these formalisms equivalent?

- constructive proofs

How can we generate compiler tools from that?

- implement DFAs

Literature

learn more

Literature

learn more

Formal languages

Noam Chomsky: Three models for the description of language. 1956

J. E. Hopcroft, R. Motwani, J. D. Ullman: Introduction to Automata Theory, Languages, and Computation. 2006

Literature
learn more

Formal languages

Noam Chomsky: Three models for the description of language. 1956

J. E. Hopcroft, R. Motwani, J. D. Ullman: Introduction to Automata Theory, Languages, and Computation. 2006

Lexical analysis

Andrew W. Appel, Jens Palsberg: Modern Compiler Implementation in Java, 2nd edition. 2002

Copyrights

Pictures
copyrights