Automata Theory

Mark van den Bergh / Rudy van Vliet

Bachelor Informatica
Data Science and Artificial Intelligence
Universiteit Leiden

Fall 2024

Universiteit
Leiden
Leiden Institute of
Advanced Computer Science

Practical information on website:
https://liacs.leidenuniv.nl/~vlietrvan1/automata/

Grades and announcements via Brightspace:
course code 4032AUTTHY_2425_S1

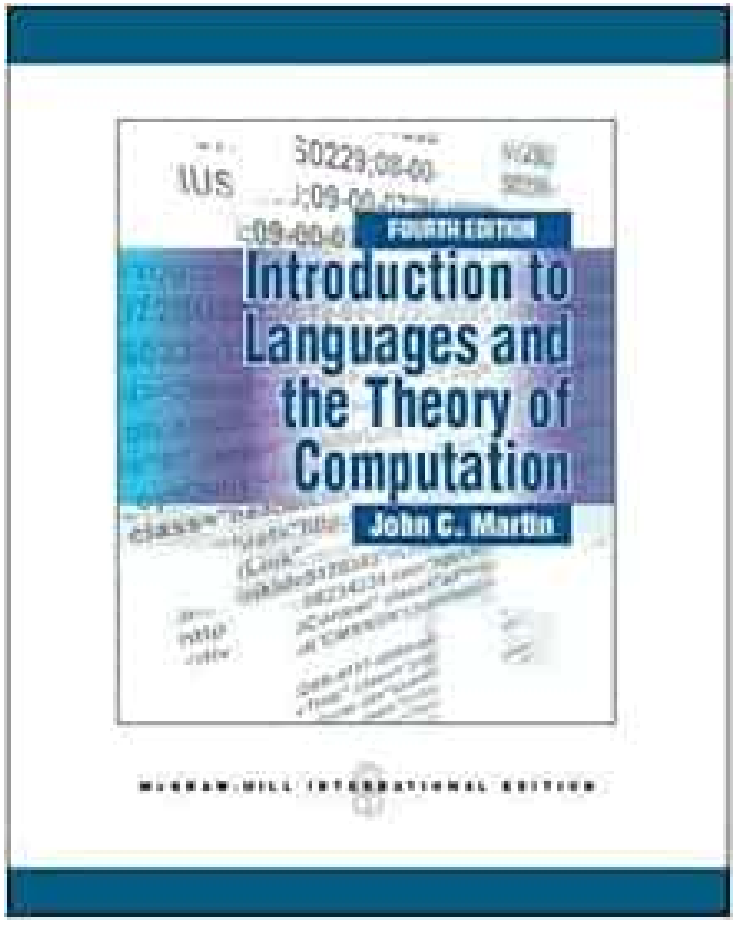
Lectures and exercise classes

Lectures: Monday 11:00–12:45 in Gorlaeus C3
Lecturers: Mark van den Bergh, Rudy van Vliet

Tutorials: Friday 11:00–12:45 in Gorlaeus BM.1.26 / BM.1.33 / BW.0.40

e-mail: automata(at)liacs(dot)leidenuniv(dot)nl

Introduction to Languages and the Theory of Computation, John C. Martin, 4th edition

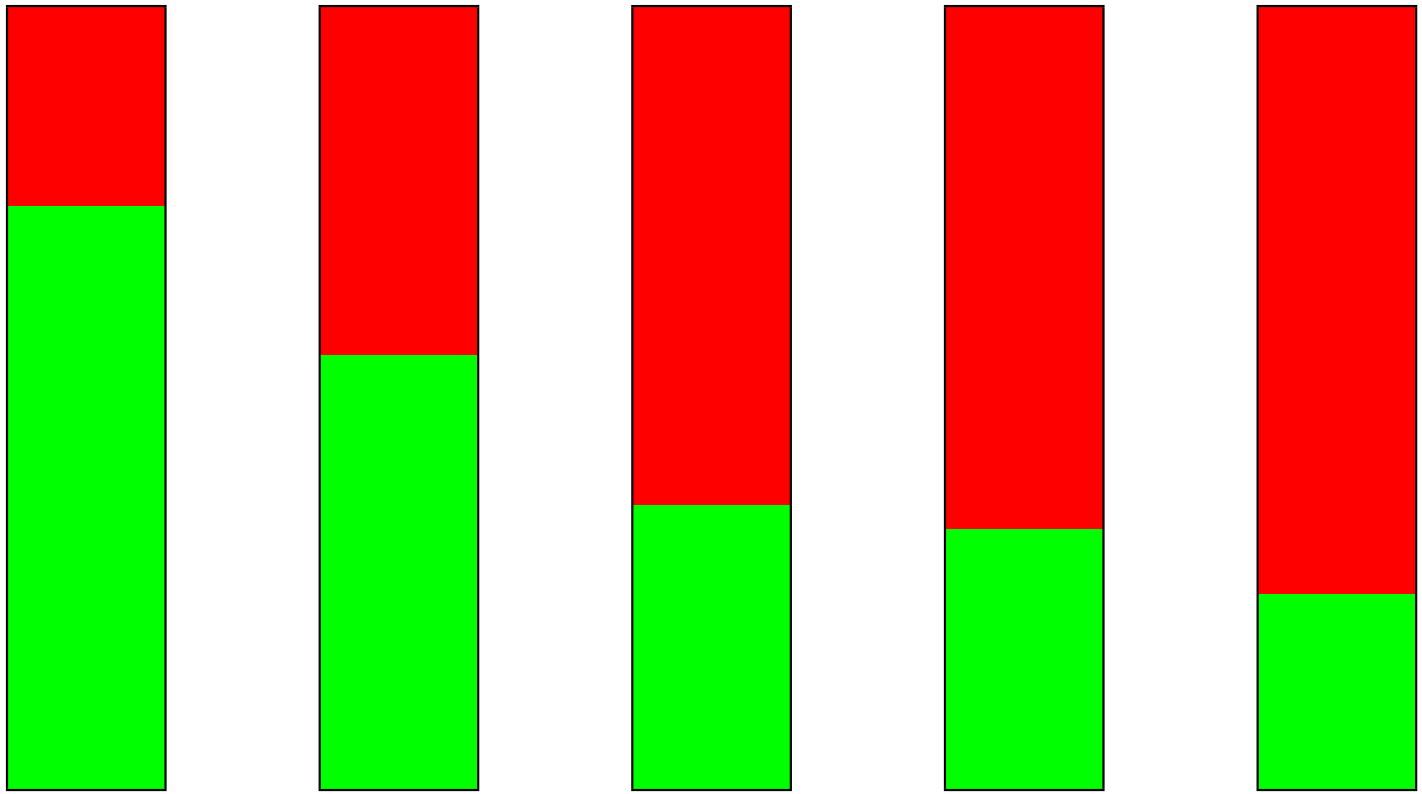


- Four homework assignments (30%), average grade H
- Written exam (70%), grade W

Let $A = 0.3H + 0.7W$.

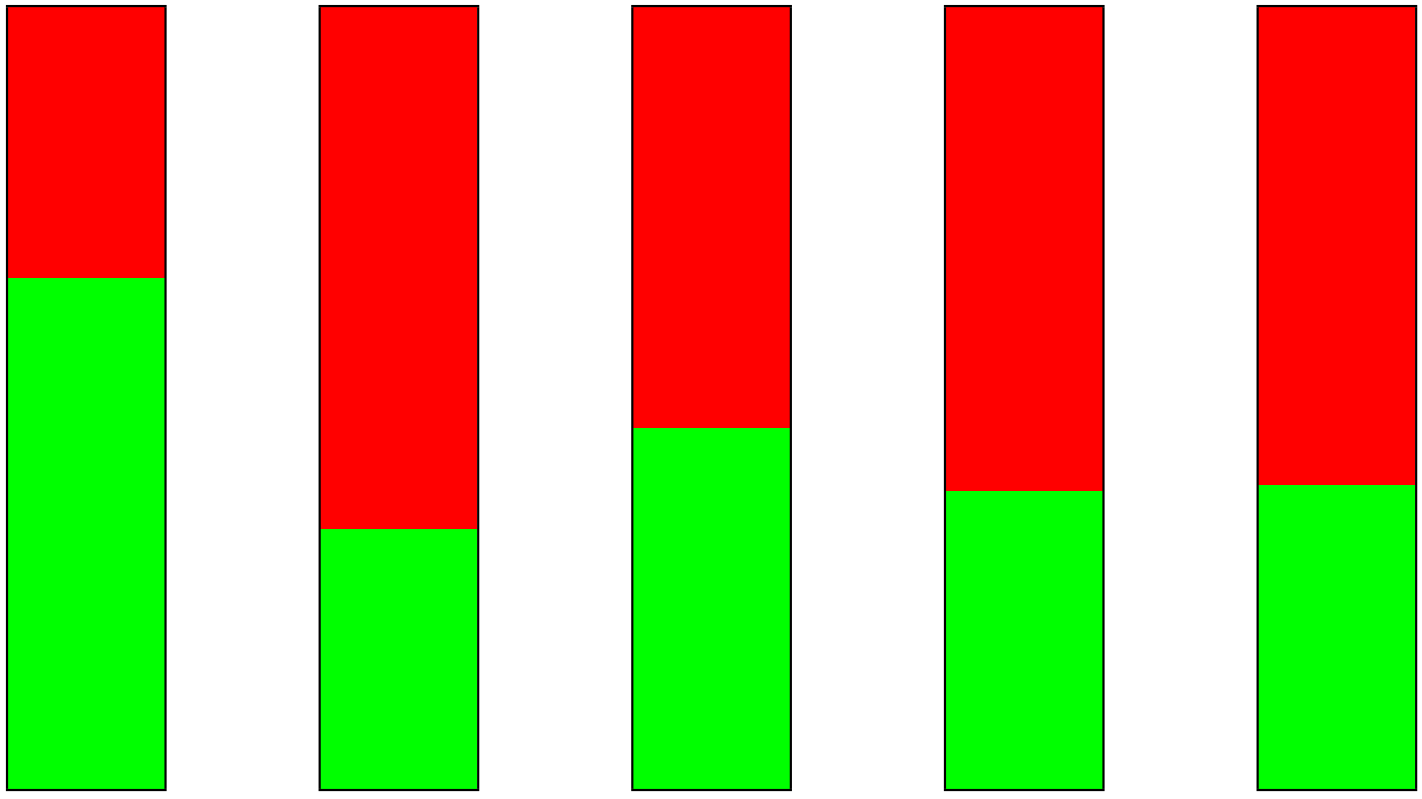
if $W < 5.5$ then grade $= W$
else if $A \geqslant 5.5$ then grade $= A$
else grade $= 6.0$

first exam, 19 December, 2022



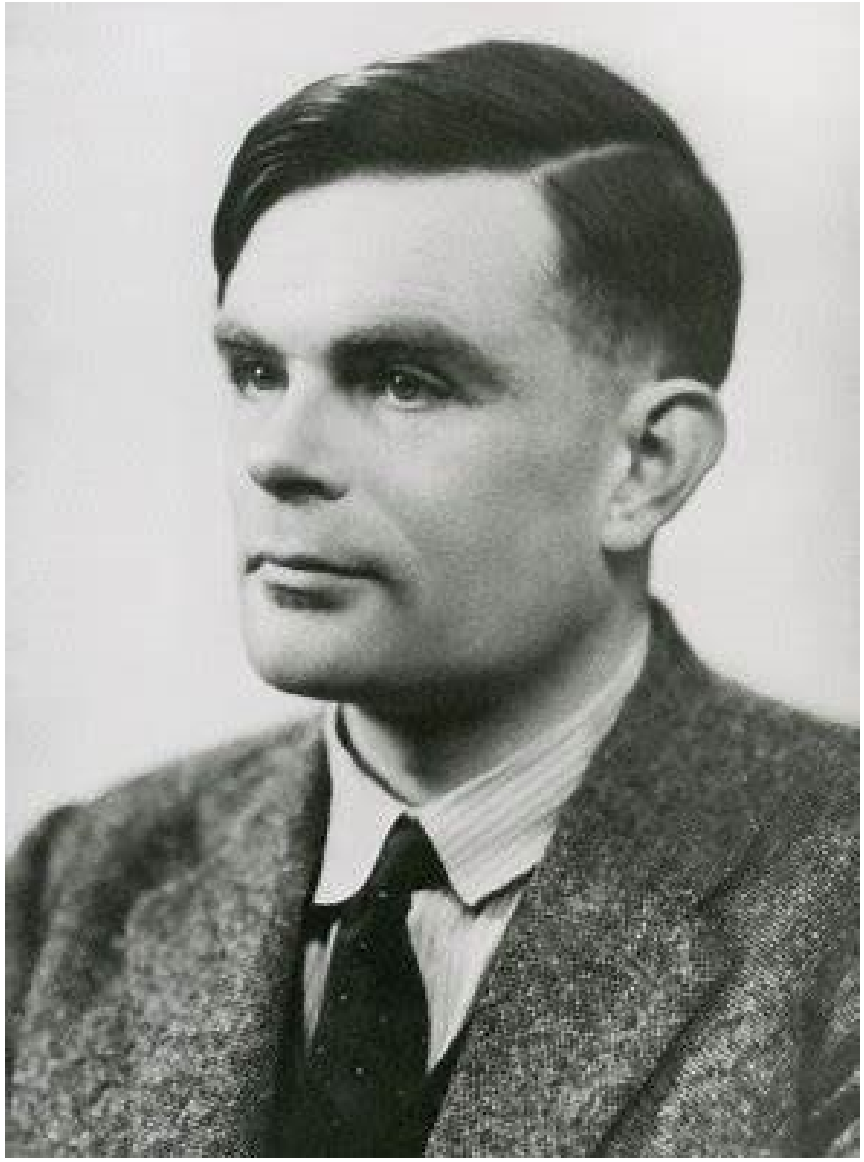
4 HW	3 HW	2 HW	1 HW	0 HW
74.67%	55.56%	36.36%	33.33%	25.00%

first exam, 21 December, 2023



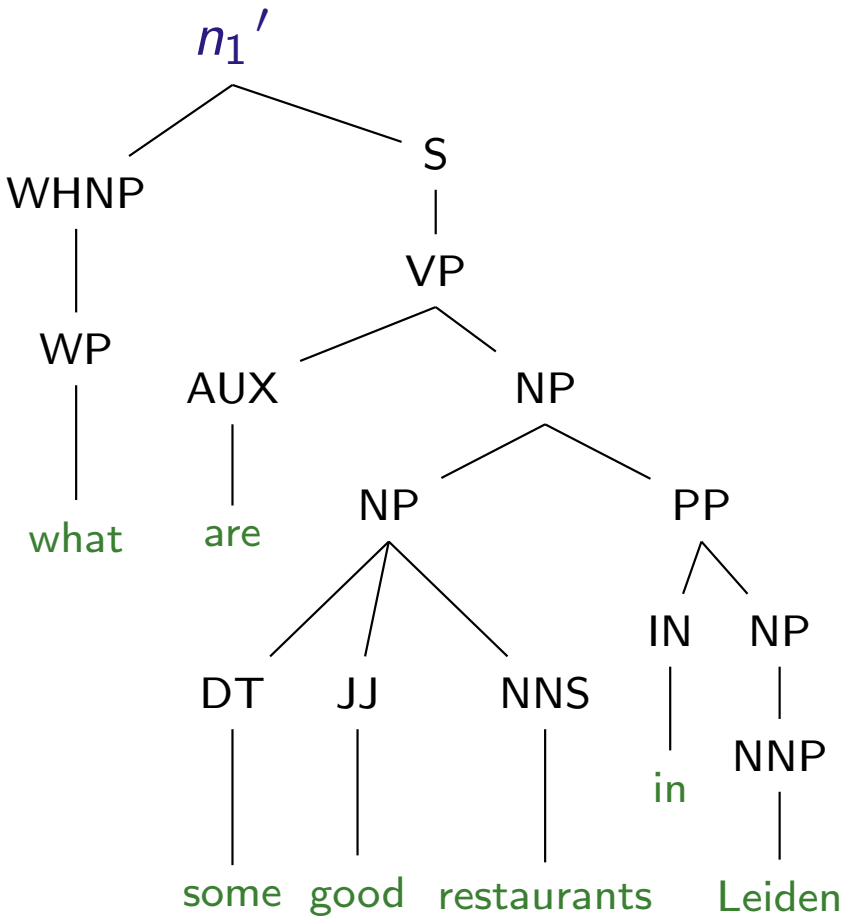
4 HW	3 HW	2 HW	1 HW	0 HW
65.42%	33.33%	46.15%	38.10%	38.89%

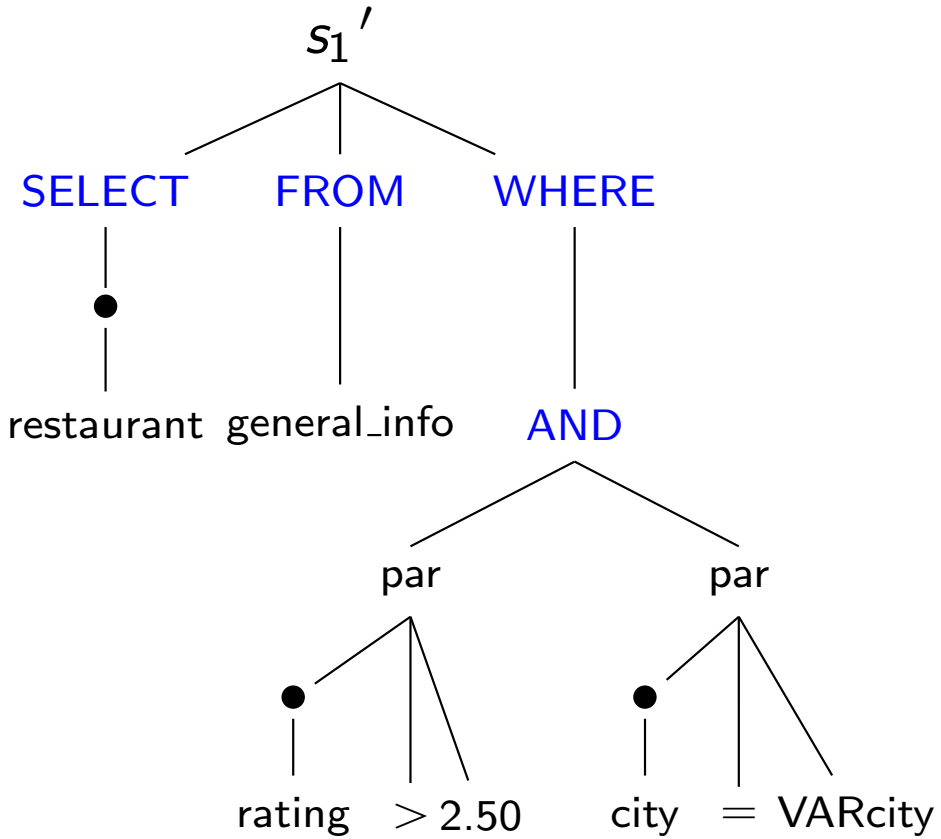
Alan Turing



Dealing with languages / sets of instances

1. Abstract machines to accept or to recognize languages
2. Grammars to generate languages
3. Expressions to describe languages



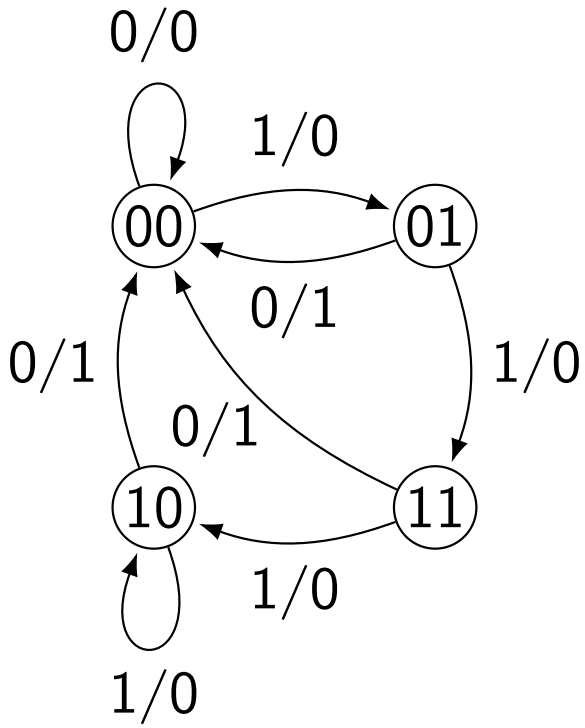


A. Giordani and A. Moschitti. Corpora for Automatically Learning to Map Natural Language Questions into SQL Queries (LREC 2010)

Formal Languages: Origins

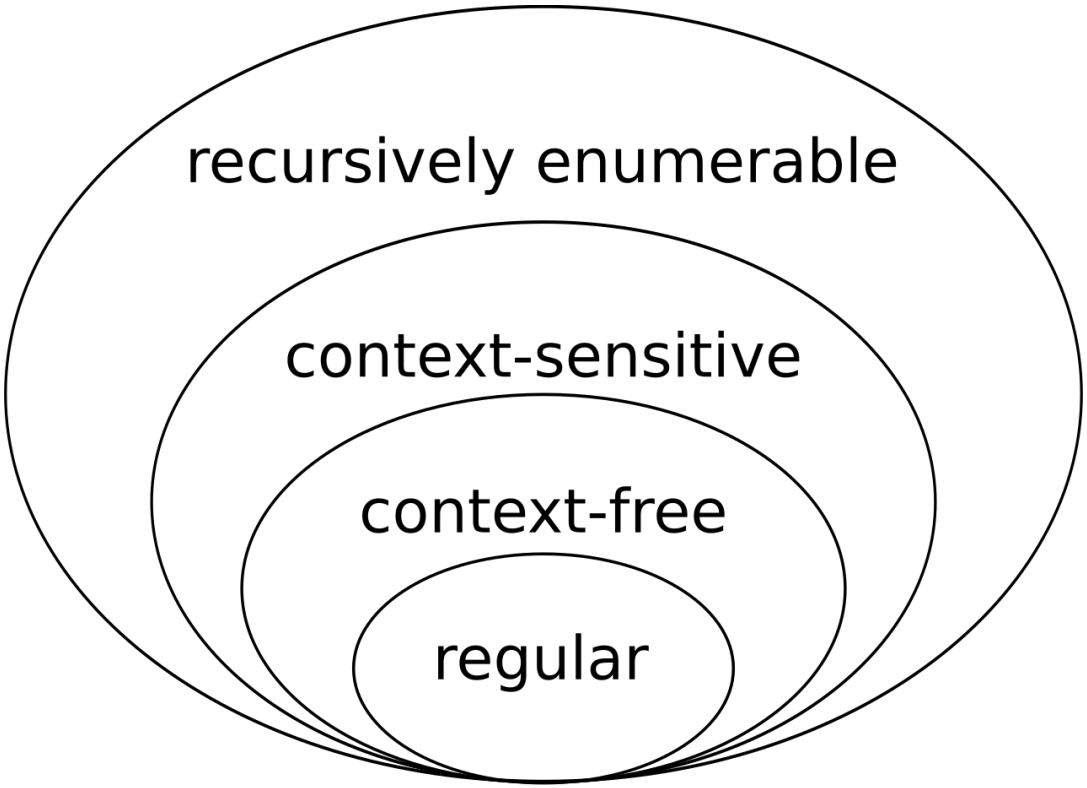
1. Logic and recursive-function theory — Introduction to Logic
2. Switching circuit theory and logical design — FDSD
3. Modeling of biological systems, particularly developmental systems and brain activity
4. Mathematical and computational linguistics
5. Computer programming and the design of ALGOL and other problem-oriented languages

S.A. Greibach. Formal Languages: Origins and Directions.
Annals of the History of Computing (1981) doi:10.1109/MAHC.1981.10006



Fundamentals of Digital Systems Design by Todor Stefanov, Leiden University

Chomsky hierarchy



Turing machines
Computability

Pushdown automata

Finite automata

[M] Table 8.21

Sets and boolean algebra $\cup, \cap, {}^c$

commutativity	$A \cup B = B \cup A$	...
associativity	$(A \cup B) \cup C = A \cup (B \cup C)$	
distributivity	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	
idempotency	$A \cup A = A$	$A \cap A = A$
De Morgan	$(A \cup B)^c = A^c \cap B^c$	
unit	$A \cup \varnothing = A$	$A \cap U = A$
	$A \cap \varnothing = \varnothing$	$A \cup U = U$
involution	$(A^c)^c = A$	
complement	$A \cap A^c = \varnothing$	
		duality

brackets

priority ^{c} before $\cup, \cap$

$K \cap L \cup M$??

[M] page 4 FDSD, FOCS

letter, symbol σ $0, 1$ a, b, c

alphabet Σ $\{a, b, c\}$

(finite, nonempty)

string, word w finite

$w = a_1 a_2 \ldots a_n$, $a_i \in \Sigma$ *abbabb*

empty string λ, Λ, ε

length $|x|$ $|\Lambda| = 0$ $|xy| = |x| + |y|$

concatenation $a_1 \ldots a_m \cdot b_1 \ldots b_n$ $ab \cdot babb$

$w\Lambda = \Lambda w = w$ $(xy)z = x(yz)$

string $w \in \Sigma^*$ $w \in \{a, b\}^*$

$\Sigma^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$ *canonical order*

infinite set of finite strings

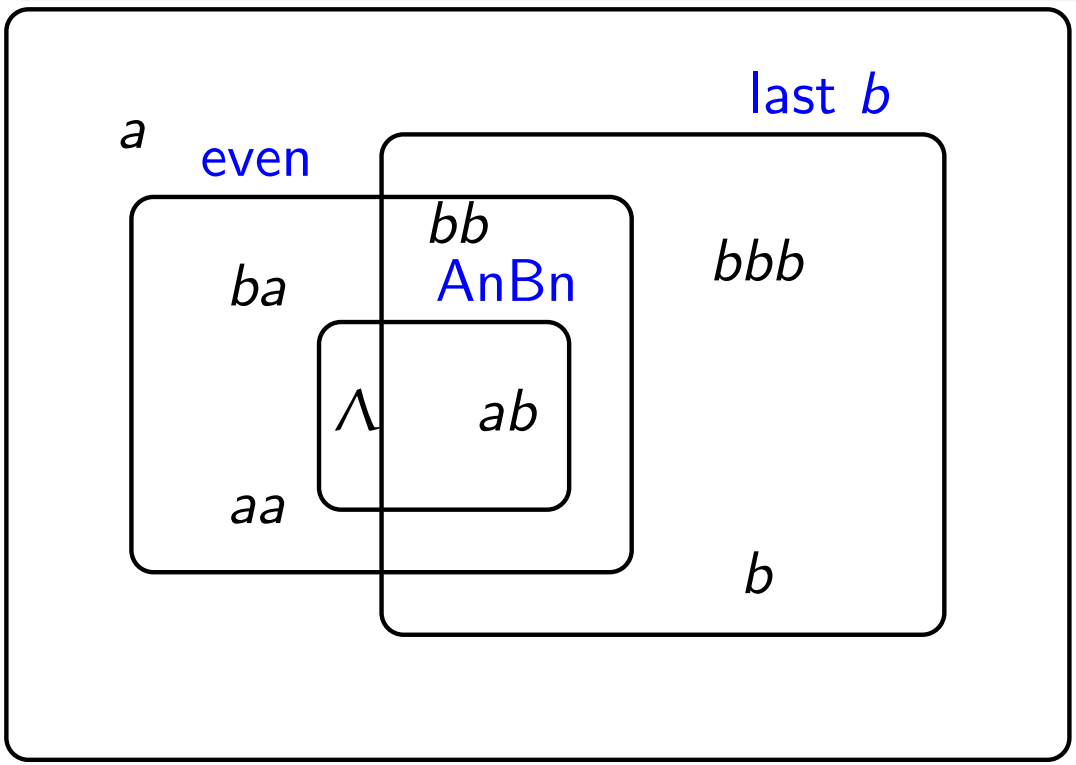
language $L \subseteq \Sigma^*$

Confusion

Λ vs. $\{\Lambda\}$ vs. $\varnothing$

Example

- $\{a, b\}^*$ all strings over $\{a, b\}$ Λ, baa, $aaaaa$
- all strings of even length Λ, $babbba$
- all strings with last letter b bbb, $aabb$
- $AnBn = \{a^n b^n \mid n \in \mathbb{N}\}$ Λ, $aaabbb$ $(\mathbb{N} = \{0, 1, 2, 3, \ldots\})$



$\{a, b\}^*$

Definition

$K \cdot L = KL = \{\, xy \mid x \in K, y \in L \,\}$

$\{a, ab\}\{a, ba\} = \{aa, aba, abba\}$

one	$\{\Lambda\}L = L\{\Lambda\} = L$
zero	$\varnothing L = L\varnothing = \varnothing$
associative	$(KL)M = K(LM)$

$L^0 = \{\Lambda\}$, even if $L = \varnothing$, c.f. 0^0
$L^1 = L, \quad L^2 = LL, \ldots$
$L^{n+1} = L^n L.$

Definition

$L^* = \bigcup_{n \geqslant 0} L^n$

$$L^n = \underbrace{L \cdot L \cdot \ldots \cdot L}_{n \text{ times}}$$

$L^n = \{\, w_1 w_2 \ldots w_n \mid w_1, w_2, \ldots, w_n \in L \,\}$ fixed n

$L^* = \{\, w_1 w_2 \ldots w_n \mid w_1, w_2, \ldots, w_n \in L, n \in \mathbb{N} \,\}$

c.f. Σ^*

Example

$\{a\}^* \cdot \{b\} = \{\Lambda, a, aa, aaa, \ldots\} \cdot \{b\} = \{b, ab, aab, aaab, \ldots\}$

$(\ \{a\}^* \cdot \{b\}\ )^* = \{b, ab, aab, aaab, \ldots\}^* =$
$\{\Lambda, b, ab, bb, aab, abb, bab, bbb, aaab, \ldots\}$

$(\ \{a\}^* \cdot \{b\}\ )^* = \{a, b\}^* \{b\} \cup \{\Lambda\}$

family all languages that can be defined by
– type of automata
(deterministic) finite aut. FA, NFA, pushdown aut. PDA
– type of grammar
context-free grammar CFG, regular (aka right-linear)
– certain operations
regular REG

Boolean operations: $\cup$, $\cap$, ^{c}
Regular operations: $\cup$, $\cdot$, *

family F *closed under* operation ∇:
if $K, L \in F$, then $K \nabla L \in F$.

RECOGNIZING, algorithm

$L = \{ x \in \{a, b\}^* \mid n_a(x) > n_b(x) \}$

count a and b

deterministic [finite] automaton

GENERATING, description

regular expression

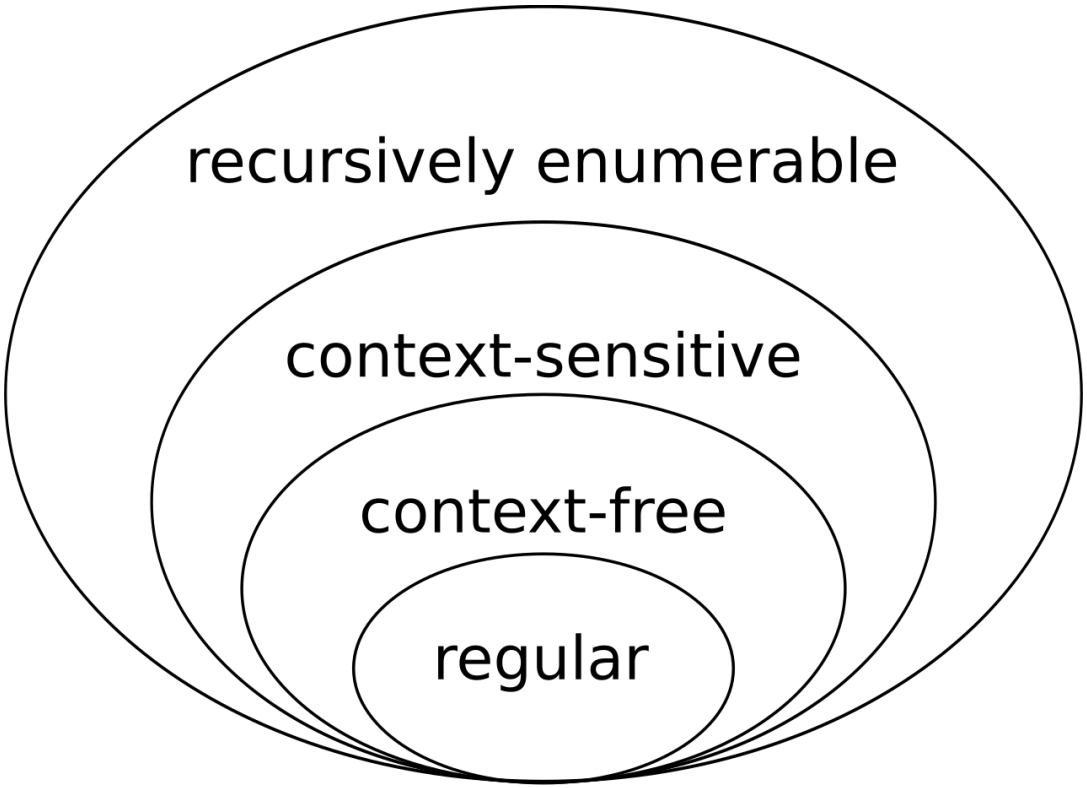
$K = (\{ab, bab\}^*\{b\})^*\{ab\} \cup \{b\}\{ba\}^*\{ab\}^*$

recursive definition

$\hookrightarrow$well-formed formulas

grammar

Chomsky hierarchy



Turing machines
Computability

Pushdown automata

Finite automata

[M] Table 8.21

During lectures: sometimes complete proofs, sometimes only sketch

In exercises, validity of a statement requires *proof*

To prove non-validity, provide *counterexample*

Quiz[1]

L_1, L_2, L_3 are languages over some alphabet Σ.

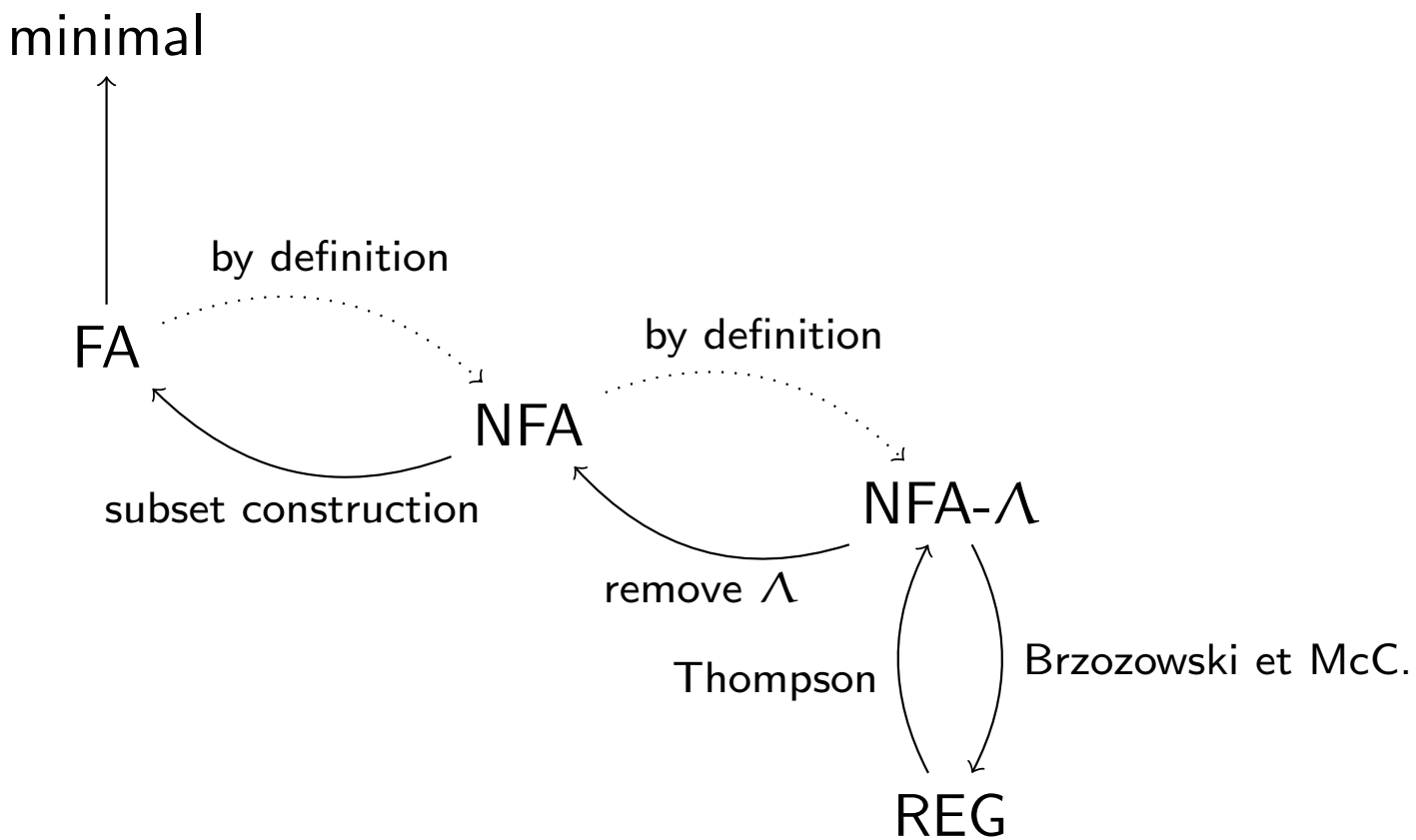
For each pair of languages below, what is their relationship?

Are they always equal? If not, is one always a subset of the other?

1. $L_1(L_2 \cap L_3)$ vs. $L_1L_2 \cap L_1L_3$
2. $L_1^* \cap L_2^*$ vs. $(L_1 \cap L_2)^*$
3. $L_1^*L_2^*$ vs. $(L_1L_2)^*$

[M] Exercise 1.37

[1] A quiz is a brief assessment used in education to measure growth in knowledge, abilities, and/or skills. Wikipedia

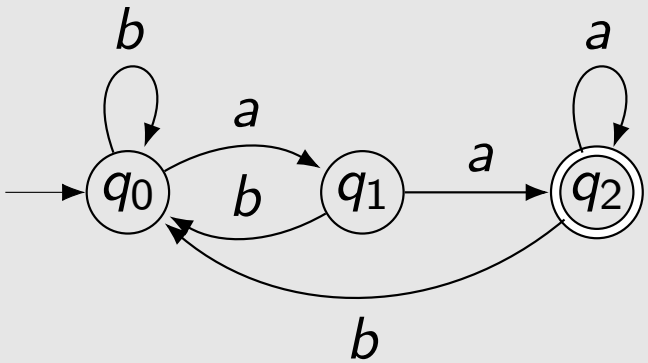

minimal
by definition
FA
by definition
NFA
subset construction
NFA-Λ
remove Λ
Thompson
Brzozowski et McC.
REG

Section 1

(Deterministic) Finite Automata

Example

$L_1 = \{ x \in \{a, b\}^* \mid x \text{ ends with } aa \}$

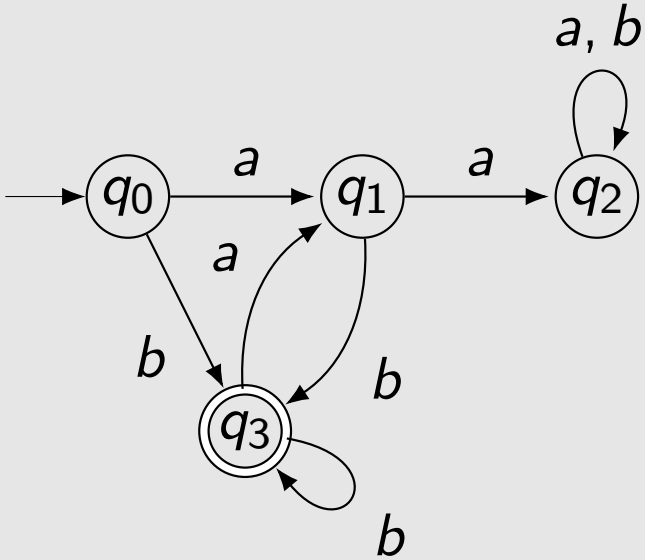


δ	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_0

[M] E. 2.1

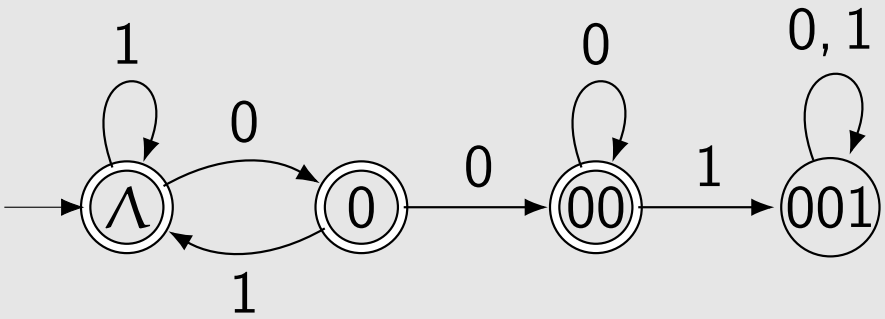
Example

$L_2 = \{\, x \in \{a, b\}^* \mid x \text{ ends with } b \text{ and does not contain } aa \,\}$



[M] E. 2.3

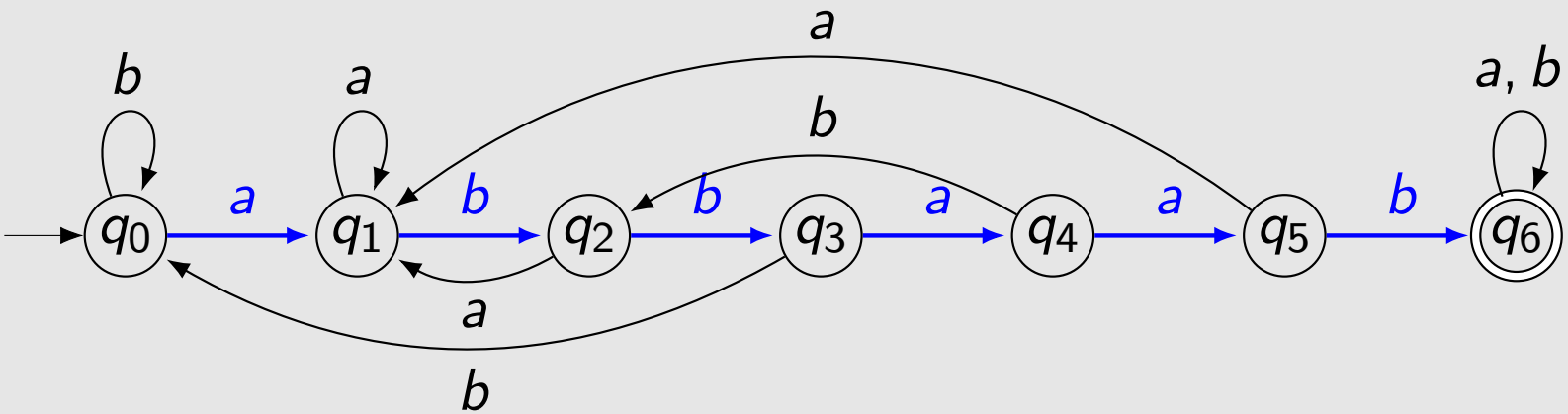
Example (Strings not containing 001)



[L] E 2.4

Example (Similar to Knuth-Morris-Pratt string search)

$L_3 = \{ x \in \{a, b\}^* \mid x \text{ contains the substring } abbaab \}$



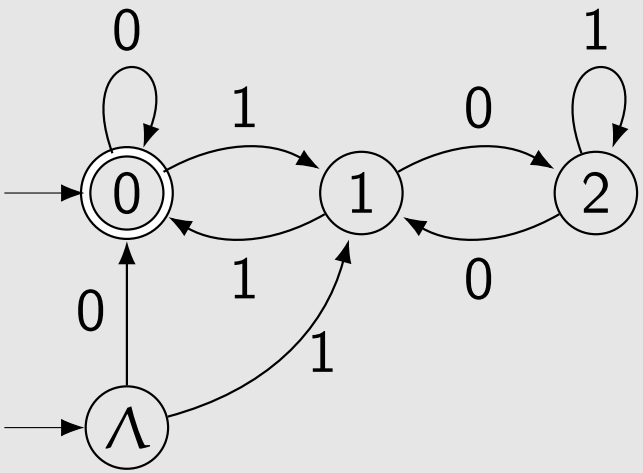
[M] E. 2.5

Note: x is divisible by $3 \Leftrightarrow x \bmod 3 = 0$.

Note: $x0 = 2 \cdot x$, and $x1 = 2 \cdot x + 1$.

Therefore:
if $x \bmod 3 = 0$, then $x0 \bmod 3 = 0$ and $x1 \bmod 3 = 1$;
if $x \bmod 3 = 1$, then $x0 \bmod 3 = 2$ and $x1 \bmod 3 = 0$;
if $x \bmod 3 = 2$, then $x0 \bmod 3 = 1$ and $x1 \bmod 3 = 2$.

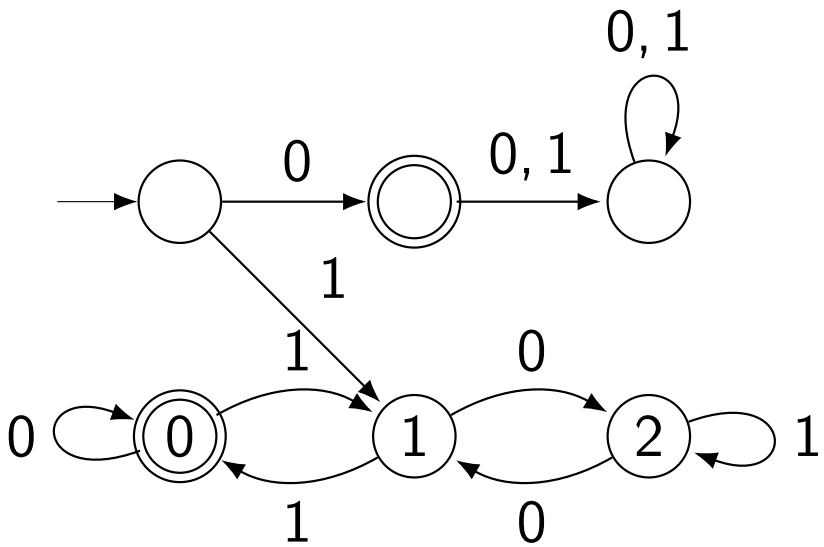
Example



δ	0	1
x	$2x$	$2x+1$
Λ	0	1
0	0	1
1	2	0
2	1	2

States represent x mod 3.

Disallows leading 0's in binary representations, e.g., 0001.



[M] E. 2.7