Formele Talen en Berekenbaarheid

Rudy van Vliet

Bachelor Informatica
Universiteit Leiden

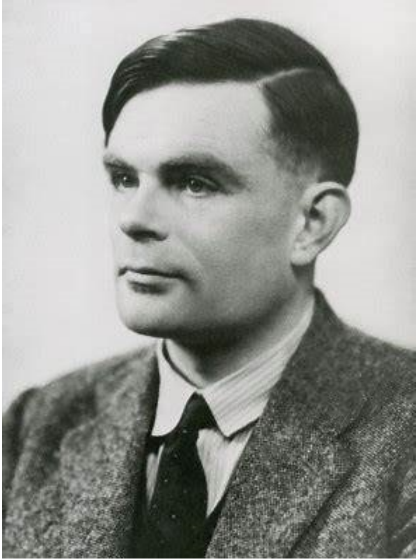
najaar 2026

Universiteit Leiden
Leiden Institute of Advanced Computer Science

This course and other courses...

- Automata Theory (6 EC), Computability (3 EC)
- FTB last year (6 EC)
- Automaten (3 EC)
- FTB this year (6 EC)
- DSAI: Languages and Computation
- students from earlier years. . .

Why this course



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

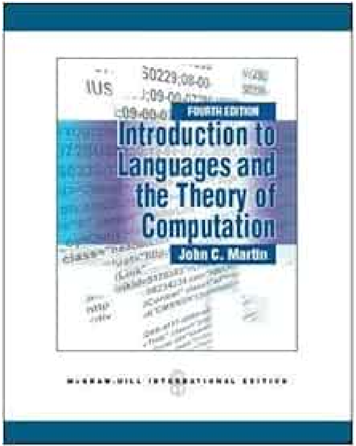
Submissions homework, grades and email via Brightspace

Lectures and exercise classes

Lectures: Wednesday 11:00–12:45 in BM.1.23 (of BM.1.33)
Lecturer: Rudy van Vliet

Tutorials: Thursday 11:00–12:45 in BW.0.05 / BW.0.06

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



- 4. context-free grammars
- 5. push-down automata
- 6. pumping lemma
- 7. Turing machines
- 8. unrestricted grammars
- 9. decidability

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

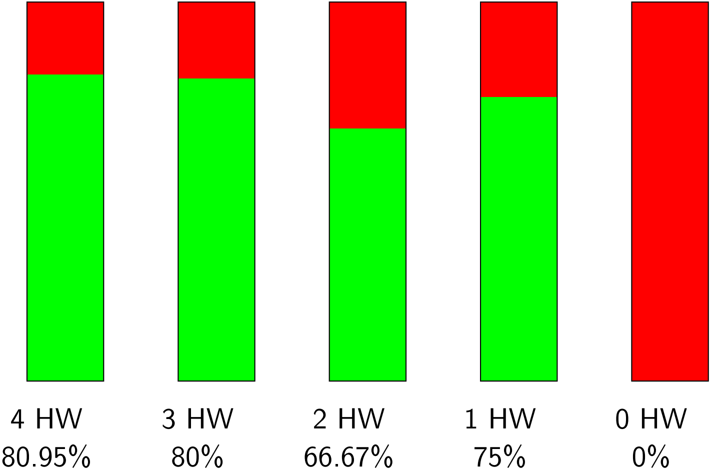
Let $F = 0.3H + 0.7W$.

if $W < 5.5$ then grade $= W$
else if $F \geq 5.5$ then grade $= F$
else grade $= 6.0$

Homework

- not mandatory, but ...
- individual

first exam, 9 January, 2026



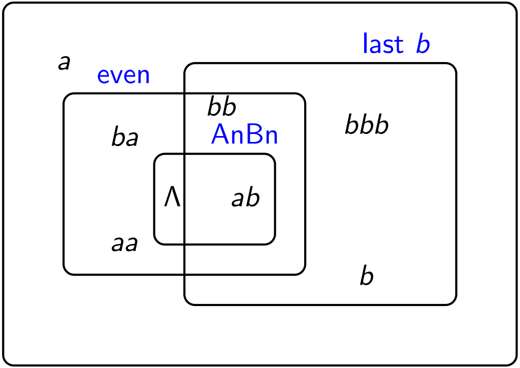
During lectures: sometimes complete proofs, sometimes only sketch

In exercises, validity of a statement requires *proof*

To prove non-validity, provide *counterexample*

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, $aaabbb$ $(\mathbb{N} = \{0, 1, 2, 3, \ldots\})$



$\{a, b\}^*$

Dealing with languages / sets of strings

1. Automata: abstract machines to accept or to recognize languages, implement an algorithm

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

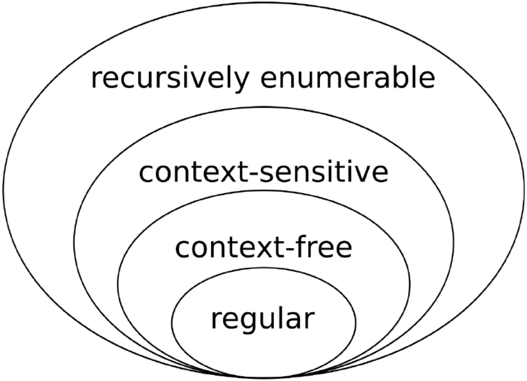
   count a and b

2. Grammars to generate languages

   today

3. Expressions to describe languages

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



Turing machines

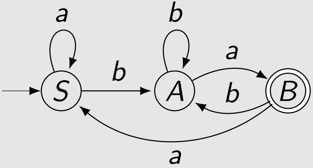
Linear-bounded automata

Pushdown automata

Finite automata

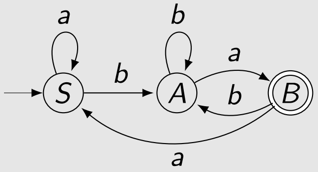
[M] Table 8.21

Example



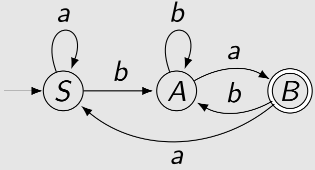
$L(M) = \dots$

Example

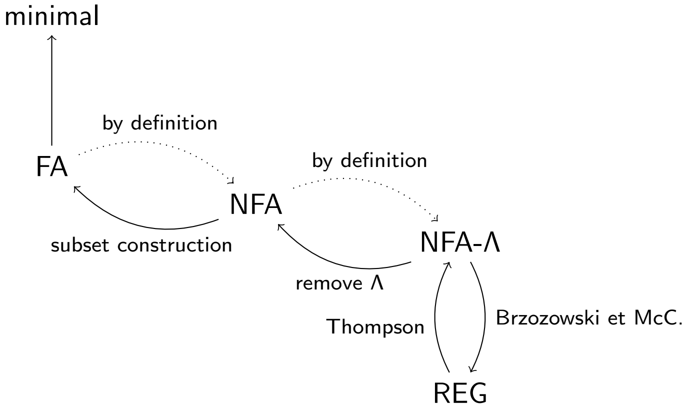


$L(M) = \{x \in \{a, b\}^* \mid \; x \text{ ends with } ba\}$
regular expression for $L(M)$...

Example



$L(M) = \{x \in \{a, b\}^* \mid \ x \text{ ends with } ba\}$
regular expression for $L(M)$: $(a + b)^* ba$


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

Section 4

Context-Free Languages

1. Context-Free Languages
   - Examples: recursion
   - Regular operations
   - Regular grammars

reg. languages	FA	reg. grammar	reg. expression
determ. cf. languages	DPDA		
cf. languages	PDA	cf. grammar	
cs. languages	LBA	cs. grammar	
re. languages	TM	unrestr. grammar	

$\langle\text{assignment}\rangle ::= \langle\text{variable}\rangle = \langle\text{expression}\rangle$

$\langle\text{statement}\rangle ::= \langle\text{assignment}\rangle \mid$
$\quad\langle\text{compound-statement}\rangle \mid$
$\quad\langle\text{if-statement}\rangle \mid$
$\quad\langle\text{while-statement}\rangle \mid \ldots$

$\langle\text{if-statement}\rangle ::=$
$\quad\text{if } \langle\text{test}\rangle \text{ then } \langle\text{statement}\rangle \mid$
$\quad\text{if } \langle\text{test}\rangle \text{ then } \langle\text{statement}\rangle \text{ else } \langle\text{statement}\rangle$

$\langle\text{while-statement}\rangle ::=$
$\quad\text{while } \langle\text{test}\rangle \text{ do } \langle\text{statement}\rangle$

Definition (well-formed formulas)

...by using the construction rules below, and only those, finitely many times:

– every propositional atom $p, q, r, \ldots$ is a wff

– if ϕ is a wff, then so is $(\neg\phi)$

– if ϕ and ψ are wff, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$,

BNF Backus Naur form

$\psi ::= p \mid (\neg\psi) \mid (\psi \wedge \psi) \mid (\psi \vee \psi) \mid (\psi \rightarrow \psi)$

M.Huet & M.Ryan, Logic in Computer Science

$AnBn = \{\ a^n b^n \mid n \geq 0\ \} \subseteq \{a, b\}^*$
Recursive definition:

Example

– $\Lambda \in AnBn$
– for every $x \in AnBn$, also $axb \in AnBn$

[M] E 1.18

$AnBn = \{ a^n b^n \mid n \geq 0 \} \subseteq \{a, b\}^*$
Recursive definition:

Example

– $\Lambda \in AnBn$ (basis)
– for every $x \in AnBn$, also $axb \in AnBn$ (induction)

$S \to \Lambda$
$S \to aSb$

$$S \;\Rightarrow\; aSb \;\Rightarrow\; aaSbb \;\Rightarrow\; aa\,bb$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaa\,bbb$$

if $S \Rightarrow^* x$ then also $S \Rightarrow^* axb$
[M] E 4.1

$AnBn = \{ a^n b^n \mid n \geq 0 \}$

variants

$\{ a^n b^{n+1} \mid n \geq 0 \}$

$S \to b$ (end with extra b)
$S \to aSb$

$\{ a^i b^j \mid i \leq j \}$

$S \to \Lambda$
$S \to aSb \mid Sb$ (free b's)

$\{ a^i b^j \mid i \neq j \}$

$S \to A \mid B$ (choice!)
$A \to aAb \mid aA \mid a$ $(i > j)$
$B \to aBb \mid Bb \mid b$ $(i < j)$

$Pal = \{x \in \{a, b\}^* \mid x^r = x\}$ (reverse)
In canonical (shortlex) order: ...
[M] E 1.18

$Pal = \{x \in \{a, b\}^* \mid x^r = x\}$ (reverse)
In canonical (shortlex) order:
$\Lambda, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, abba, \ldots$
Recursive definition:

Example

– $\Lambda, a, b \in Pal$
– for every $x \in Pal$, also $axa, bxb \in Pal$

[M] E 1.18

$Pal = \{x \in \{a, b\}^* \mid x^r = x\}$ (reverse)
In canonical (shortlex) order:
$\Lambda, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, abba, \ldots$
Recursive definition:

Example

- $\Lambda, a, b \in Pal$
- for every $x \in Pal$, also $axa, bxb \in Pal$

$S \to \Lambda \mid a \mid b$
$S \to aSa$
$S \to bSb$

$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aababaa$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa and bSb are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa, bSb, aSb and bSa are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb \mid aSb \mid bSa$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa, bSb, aSb and bSa are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb \mid aSb \mid bSa$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbSaba \Rightarrow abbbAaaba \Rightarrow abbbaaba$
[M] E 4.3

Balanced $\subseteq \{(,)\}^*$: $\Lambda, (), (()), ()(), ((())), (())(), \ldots$
Recursive definition:

Example

– $\Lambda \in$ *Balanced*
– for every $x, y \in$ *Balanced*, also $xy \in$ *Balanced*
– for every $x \in$ *Balanced*, also $(x) \in$ *Balanced*

[M] E 1.19

$Balanced \subseteq \{(,)\}^*$: $\Lambda, (), (()), ()(), ((())), (())(), \ldots$
Recursive definition:

Example

– $\Lambda \in Balanced$
– for every $x, y \in Balanced$, also $xy \in Balanced$
– for every $x \in Balanced$, also $(x) \in Balanced$

$S \to \Lambda \mid SS \mid (S)$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()((S)) \Rightarrow ()(())$

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

[M] E 1.19

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

$S \to a \mid S + S \mid S * S \mid (S)$
derivation(s) for $a + (a * a)$ and $a + a * a$...

[M] E 4.2

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

$$S \to a \mid S + S \mid S * S \mid (S)$$

$$S \Rightarrow S + S \Rightarrow S + S * S$$
$$S \Rightarrow S * S \Rightarrow S + S * S$$

ambiguity

[M] E 4.2

alphabet $\{ 1, 2, 5, = \}$

$\{ x{=}y \mid x \in \{1, 2\}^*, y \in \{5\}^*, n_1(x) + 2n_2(x) = 5n_5(y) \}$

$n_\sigma(x)$ number of σ occurrences in x

212=5 22222=55 121221221222=5555

The problem with most solutions is that when read from left to right the initial string over $\{1, 2\}$ cannot always be chopped into parts with exact value 5, without chopping the symbol 2.
The solution is like a finite automaton, which reads $1, 2$ and 'saves' the values until the value 5 is reached, then we write a 5 to the right.

alphabet $\{\ 1, 2, 5, =\ \}$

$\{\ x{=}y \mid x \in \{1,2\}^*, y \in \{5\}^*, n_1(x) + 2n_2(x) = 5n_5(y)\ \}$

$n_\sigma(x)$ number of σ occurrences in x

212=5 22222=55 121221221222=5555

variables S_i, $0 \le i \le 4$

axiom S_0

productions

$S_0 \to 1S_1 \mid 2S_2$
$S_1 \to 1S_2 \mid 2S_3$
$S_2 \to 1S_3 \mid 2S_4$
$S_3 \to 1S_4 \mid 2S_05$
$S_4 \to 1S_05 \mid 2S_15$
$S_0 \to =$

Definition

context-free grammar (CFG)   4-tuple $G = (V, \Sigma, S, P)$

– V alphabet   *variables* / *nonterminals*

– Σ alphabet   *terminals*   disjoint $V \cap \Sigma = \varnothing$

– $S \in V$ *axiom*, *start symbol*

– P finite set   rules, *productions*

of the form $A \to \alpha$, $A \in V$, $\alpha \in (V \cup \Sigma)^*$

derivation step $\alpha = \alpha_1 A \alpha_2 \Rightarrow_G \alpha_1 \gamma \alpha_2 = \beta$ for $A \to \gamma \in P$

Definition

language generated by G

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

[M] Def 4.6 & 4.7

NonPal, its grammar components

$$S \to aAb \mid bAa \mid aSa \mid bSb$$
$$A \to \Lambda \mid aA \mid bA$$

variables $V = \{ S, A \}$
terminals $\Sigma = \{ a, b \}$
axiom S
productions
$P = \{A \to \Lambda, A \to aA, A \to bA, S \to aAb, S \to bAa, S \to aSa, S \to bSb\}$

$\Rightarrow_G^*$ is the *transitive and reflexive closure* of $\Rightarrow_G$

zero, one or more steps

general case $\quad \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = \beta$

$\alpha \Rightarrow_G^* \beta$ iff there are strings $\alpha_0, \alpha_1, \ldots, \alpha_n$ such that
- $\alpha_0 = \alpha$
- $\alpha_n = \beta$
- $\alpha_i \Rightarrow \alpha_{i+1} \quad$ for $0 \leq i < n$.

special case $\quad n = 0 \quad \alpha = \alpha_0 = \beta$

Variables can be rewritten regardless of context

Lemma

If $u_1 \Rightarrow^* v_1$ and $u_2 \Rightarrow^* v_2$, then $u_1 u_2 \Rightarrow^* v_1 v_2$.

Lemma

If $u \Rightarrow^* v_1 v v_2$ and $v \Rightarrow^* w$, then $u \Rightarrow^* v_1 w v_2$.

Lemma

If $u \Rightarrow^* v$ and $u = u_1 u_2$,
then $v = v_1 v_2$ such that $u_1 \Rightarrow^* v_1$ and $u_2 \Rightarrow^* v_2$.

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

$aaabbb, ababab, aababb, \ldots$

[M] E 4.8

From Automaten:

– Even number of both a and b

two letters together

aa and bb keep both numbers even [odd]

ab and ba switch between even and odd, for both numbers

$(\ aa + bb + (ab + ba)(aa + bb)^*(ab + ba) \ )^*$

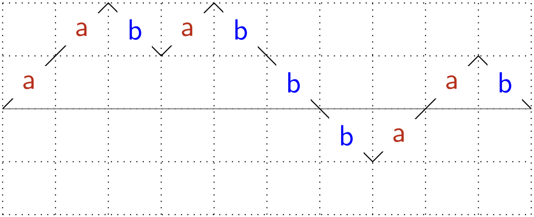
[M] E 3.4

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

$aaabbb, ababab, aababb, \ldots$

[M] E 4.8

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



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

$aaabbb, ababab, aababb, \ldots$

$$\begin{aligned} S &\to \Lambda \mid aB \mid bA \\ A &\to aS \mid bAA \\ B &\to bS \mid aBB \end{aligned}$$

A generates $n_a(x) = n_b(x) + 1$
B generates $n_a(x) + 1 = n_b(x)$

$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow \ldots$ (different options)

(1) $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$
(2) $\ldots$ (ambiguous, later)

[M] E 4.8