



# Computational Complexity

Green board scans

Winter 2014/15

Christian Schindelhauer

Technical Faculty

Computer-Networks and Telematics

University of Freiburg

# 1. Introduction & Motivation

## Computational Complexity

– What is Computational Complexity about?

P, NP, problems, complexity analysis

What is computable?

# 1. Introduction & Motivation

Example Multiplication

◦  $3 \times 5 = \dots$

◦  $\pi \times e = 8.539734\dots$

◦  $\exists_{x, y > 1}^{x, y \in \mathbb{N}}: x \cdot y = 124959859$   
Yes

What is a problem?

# 1. Introduction & Motivation

1. Functions  $f: A \times B \times C \rightarrow D$

e.g.  $f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$g: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

$h: \Sigma^* \times \Sigma^* \rightarrow \{0,1\}$

$\Sigma$  is an alphabet, finite set of symbols

$\Sigma = \{0,1\}, \Sigma = \{a,b,c\}$

$\Sigma^* = \emptyset \cup \Sigma \cup \Sigma^2 \cup \Sigma^3 \dots$

2. Predicates

$f: A \times B \times C \rightarrow \{0,1\}$   
 $\{\text{false, true}\}$

3. Boolean Functions

$f: \{0,1\}^n \rightarrow \{0,1\}^m$

4. Boolean Predicates

$f: \{0,1\}^n \rightarrow \{0,1\}$

# 1. Introduction & Motivation

## 5. Languages

$$L \subseteq \Sigma^*$$

characteristic function

$$X_L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$$

enumerable language

$$f: \mathbb{N} \rightarrow \Sigma^* \quad \forall i \in \mathbb{N}. f(i) \in L$$

$$\forall w \in L \exists i: f(i) = w$$

# 1. Introduction & Motivation

Why TM?  
 - because it is simple

---

Finite Automata

$A = (\Sigma, Q, q_0, Q_{acc}, \Delta)$

alphabet    set of states    initial state    set of accepting states    transition function

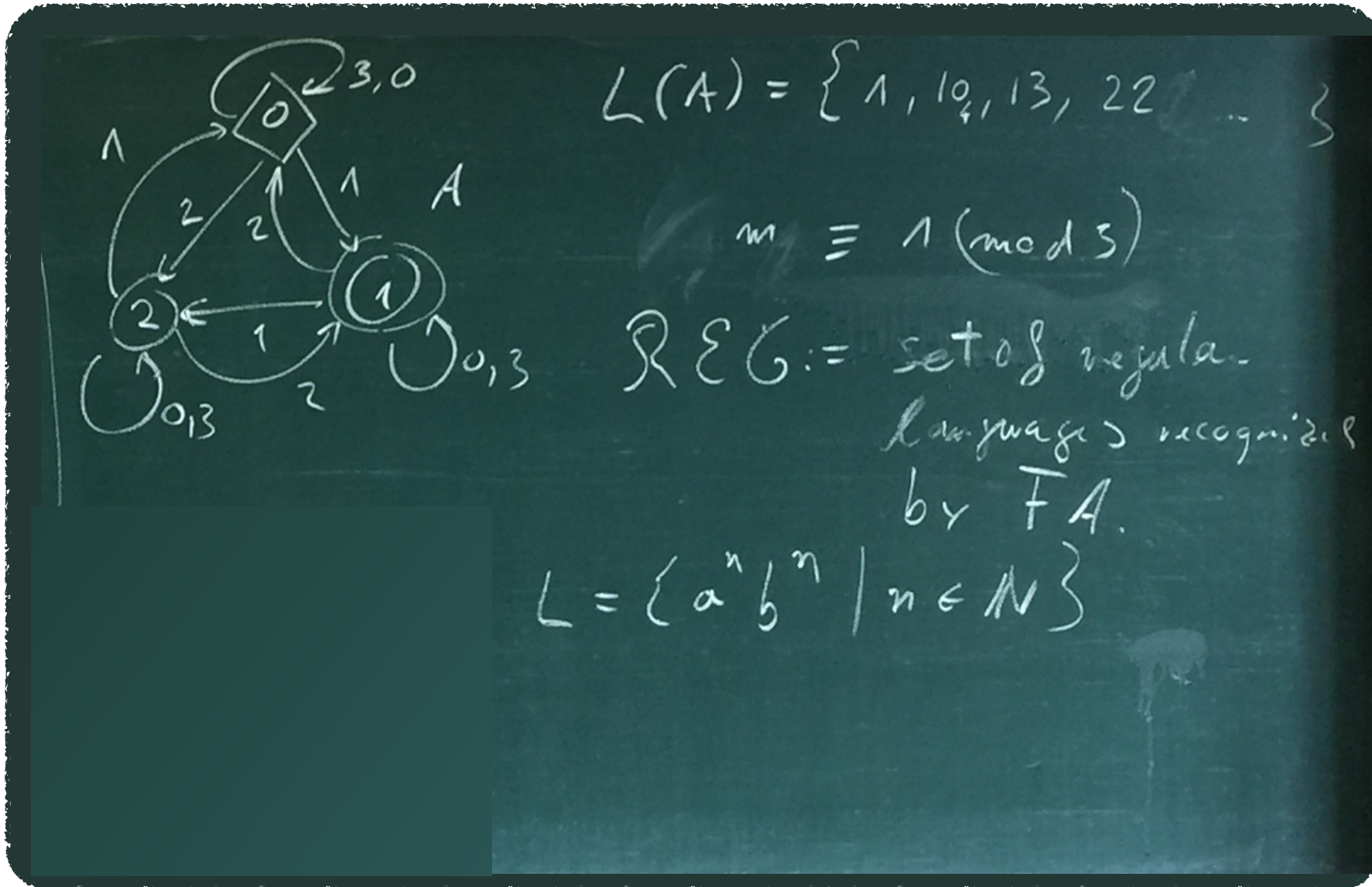
$\Delta: Q \times \Sigma \rightarrow Q$

configuration  $C \in Q$   
 $C_A(\epsilon) := q_0$

$C_A(aw) = \Delta(C_A(w), a)$

$L(A) := \{w \in \Sigma^* \mid C_A(w) \in Q_{acc}\}$

# 1. Introduction & Motivation



$L(A) = \{1, 10, 13, 22, \dots\}$

$m \equiv 1 \pmod{3}$

$\mathcal{REG} :=$  set of regular languages recognized by FA.

$L = \{a^n b^n \mid n \in \mathbb{N}\}$

The diagram shows a DFA with three states: 0 (top), 1 (bottom right), and 2 (bottom left). State 1 is the start state (indicated by an incoming arrow from the left) and an accepting state (indicated by a double circle). Transitions are: 0 to 0 on 3, 0 to 2 on 0; 2 to 0 on 1, 2 to 1 on 2; 1 to 2 on 1, 1 to 1 on 0, 3. Self-loops on 0, 1, and 2 are labeled 0, 3.

# 1. Introduction & Motivation

How to compute?

- Turing machine, PRAM, RAM, Petri Nets, human brain, pseudo-language, Finite Automata, Boolean circuits
- Java, C, Algol, FORTRAN, Haskell, Lisp, ...
- calculator,  $\lambda$ -calculus, LOOP, WHILE, BrainFuck, White Space, Cellular Automata
- Analog computers, Quantum computers
- String rewriting systems, Formal Grammars



# 1. Introduction & Motivation

Aachen       $A_{alen} \stackrel{lex}{\prec} A_{achen}?$   
 Aalen       $|w_1| < |w_2| \Rightarrow w_1 \stackrel{lex}{\prec} w_2$

Probability function       $p(w)$ : prob. that  $w \in \Sigma^*$  happens

-  $f: \Sigma^* \rightarrow [0, 1]$

- random variable  $X$  :  $P[X=w] = p(w)$

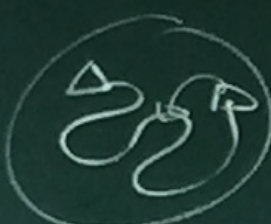
e.g. - prime number generator

$f(n) \rightarrow$  produces a prime number  $\in [2^{n-1}, 2^n]$   
 with equal probability

✓ POL-Sampleable

# 1. Introduction & Motivation

1. Fix an automata  $Q$  is finite



$a^i \rightarrow q_i = q_j !$   
 $a^j \rightarrow q_j$

$q_i = q_j$

$a^i$	$b^i$	$\in L$
$a^j$	$b^j$	$\in L$
$a^j$	$b^i$	$\notin L$

$\varepsilon \quad q_0$   
 $a \quad q_1$   
 $aa \quad q_2$   
 $aaa \quad q_i$   
 $aaaa \quad q_j$   
 $aaaaa \quad q_j$   
 $\vdots$

# 1. Introduction & Motivation

Myhill - Nerode-  
Equivalency-Class

$$\text{COPY} = \{ww \mid w \in \Sigma^*\}$$

$$\Sigma = \{a, b\}$$

$$= \{\varepsilon, aa, bb, aaaa, abab, baba, \dots\}$$

is not in CFL (context-free)