



Technische  
Universität  
Braunschweig



# Theoretische Informatik 2

Arne Schmidt

# Vorstellung

## Interessen

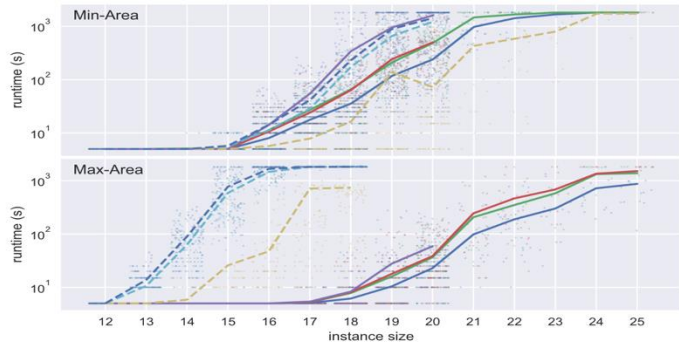
- Geometrische Optimierung,
- Programmierbare Materie,
- Komplexitätstheorie

## #GernePerDu

### Computing Area-Optimal Simple Polygonizations

SÁNDOR P. FEKETE, ANDREAS HAAS, PHILLIP KELDENICH, MICHAEL PERK, and ARNE SCHMIDT, Department of Computer Science, TU Braunschweig

We consider the problem of finding a simple polygonization of a set of points in the plane. This problem is NP-hard and has been the focus of recent research. We present a new algorithm for this problem and provide experimental results.



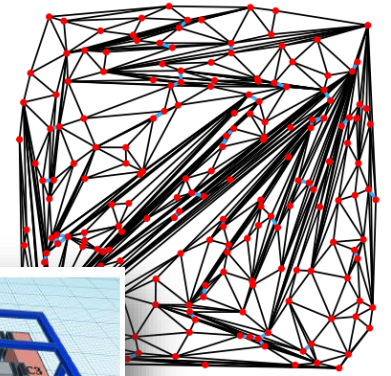
### Computing MaxMin Edge Length Triangulations

Sándor P. Fekete\* Winfried Hellmann\* Michael Hemmer\* Arne Schmidt\*  
Julian Troegel\*

#### Abstract

$\epsilon^2$ ) algorithm  
of a set of  
of finding  
as a natural  
problem by  
te. Moreover,  
polynomial-  
imate MELT  
edge

whil  
hull  
varie  
a ste  
com  
triar  
opti  
angl  
edge

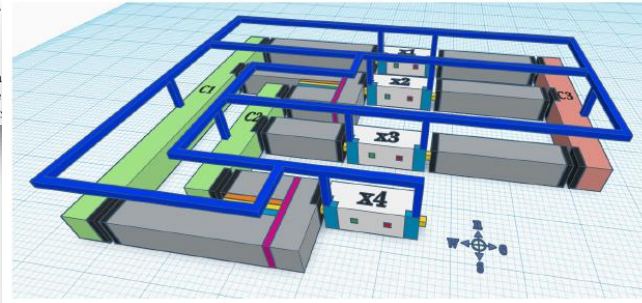


### Particle-Based Assembly Using Precise Global Control\*

Christian Rieck<sup>1</sup>[0000-0001-9988-953X], Christian Rieck<sup>1</sup>[0000-0003-0846-5163],  
Julian Scheffer<sup>2</sup>[0000-0002-3471-2706], and Arne Schmidt<sup>1</sup>[0000-0001-8950-3963]

<sup>1</sup> Department of Computer Science, TU Braunschweig, Germany.  
{jkeller, rieck, aschmidt}@ibr.cs.tu-bs.de  
<sup>2</sup> Department of Computer Science, University of Münster, Germany.  
christian.

micro- and  
external force like  
particles that



# Organisation

Vorlesung

- Grundlagen

+

Gr. Übung

- Vertiefungen



Arne Schmidt

Kl. Übung

- Besprechung Hausaufgaben

# Fragen



Inhaltlich oder  
allgemeiner Ablauf

Vorlesung / große Übung



Zu Übungsblättern  
oder Korrektur

Kleine Übungen / Tutoren



Individuelle Fragen

**Immer per Mail** an [aschmidt@ibr.cs.tu-bs.de](mailto:aschmidt@ibr.cs.tu-bs.de)  
(Nicht über StudIP)



Sprechstunde

Montags, 09:45 Uhr im Raum IZ 333  
(Am besten vorher per Mail ankündigen)

# Mailingliste (<https://lists.ibr.cs.tu-bs.de/postorius/lists/theo2.ibr.cs.tu-bs.de/>)



Postorius



Listen



Archiv

Anmelden

Registrieren

Theo2 theo2@ibr.cs.tu-bs.de

## Zusammenfassung

Benutzen Sie folgende Adresse, um die Listen-Besitzer zu kontaktieren: *theo2-owner@ibr.cs.tu-bs.de*

You have to sign in to visit the archives of this list.

## Mitglied werden/Mitgliedschaft beenden

To subscribe or unsubscribe from this list, please sign in first. If you have not previously signed in, you may need to set up an account with the appropriate email address.

Anmelden

You can also subscribe without creating an account. If you wish to do so, please use the form below.

Ihre E-Mail-Adresse

Ihr Name (optional)

Abonnieren

# Semesterplan (vorläufig)

Datum (Woche)	VL / Ü	VL / Ü Inhalt	Hausaufgabe (Ausgabe, Mo)	Hausaufgabe (Abgabe, Mo, 13 Uhr)	Hausaufgabe (Besprechung)
<b>06. April</b>	- + -				
<b>13. April</b>	0+1	Grammatiken, Turing Maschinen			
<b>20. April</b>	2+3	LBAs, Determinismus	HA1		
<b>27. April</b>	4+5	Immermann & Szelepcsény, Berechnbarkeit			
<b>04. Mai</b>	6+7	(Un-) Entscheidbarkeit	HA2	HA1	
<b>11. Mai</b>	8+9	Reduktionen			HA1
<b>18. Mai</b>	10+11	Satz von Rice	HA3	HA2	
<b>25. Mai</b>	Exkursionswoche				
<b>01. Juni</b>	12+13	Komplexitätsklassen			HA2
<b>08. Juni</b>	14+15	Komplemente von Komplexitätsklassen	HA4	HA3	
<b>15. Juni</b>	16+17	Reduktionen II, PATH, 2-SAT			HA3
<b>22. Juni</b>	18+19	P und NP	HA5	HA4	
<b>29. Juni</b>	20+21	Zertifikate			HA4
<b>06. Juli</b>	22+23	PSPACE und Hierarchiesätze		HA5	
<b>13. Juli</b>	24+25	Zusammenfassung, Fragestunde			HA5

# Hausaufgaben

20 Punkte pro Blatt

5 Blätter

50% aller Punkte (50 Punkte) für die Studienleistung

Abgaben per Zweierteams

Auf Papier

# Anmeldung kleine Übungen

Offen bis einschl. 14.04.

Gruppenaufteilung erfolgt am 15.04. per Mail.

<https://www.ibr.cs.tu-bs.de/courses/ss26/theo2/index.xml>

Nachname *	<input type="text"/>
Vorname *	<input type="text"/>
Matrikelnummer *	<input type="text"/>
Studiengang *	Informatik <input type="button" value="v"/>
Semester *	<input type="text"/>
y-Nummer *	<input type="text"/>
E-Mail *	<input type="text"/>
Mittwoch, 11:30	ja <input type="button" value="v"/>
Donnerstag, 09:45	ja <input type="button" value="v"/>
Donnerstag, 15:00	ja <input type="button" value="v"/>
Donnerstag, 16:45	ja <input type="button" value="v"/>
Freitag, 09:45	ja <input type="button" value="v"/>
Freitag, 15:00	ja <input type="button" value="v"/>
y-Nummer Wunschpartner	<input type="text"/>

Mit Absenden dieses Formular stimmen Sie der Verarbeitung der von Ihnen selbst angegebenen Daten nur zum Zwecke der hier erfolgten Erhebung zu. Außerdem nehmen Sie die [Datenschutzerklärung der TU Braunschweig](#) und die [Datenschutzerklärung des IBR](#) zur Kenntnis nach denen die Verarbeitung Ihrer Daten erfolgt.

# Klausur – Vorläufig



Datum:  
25.08.26



Uhrzeit:  
10:30 Uhr bis  
12:30 Uhr



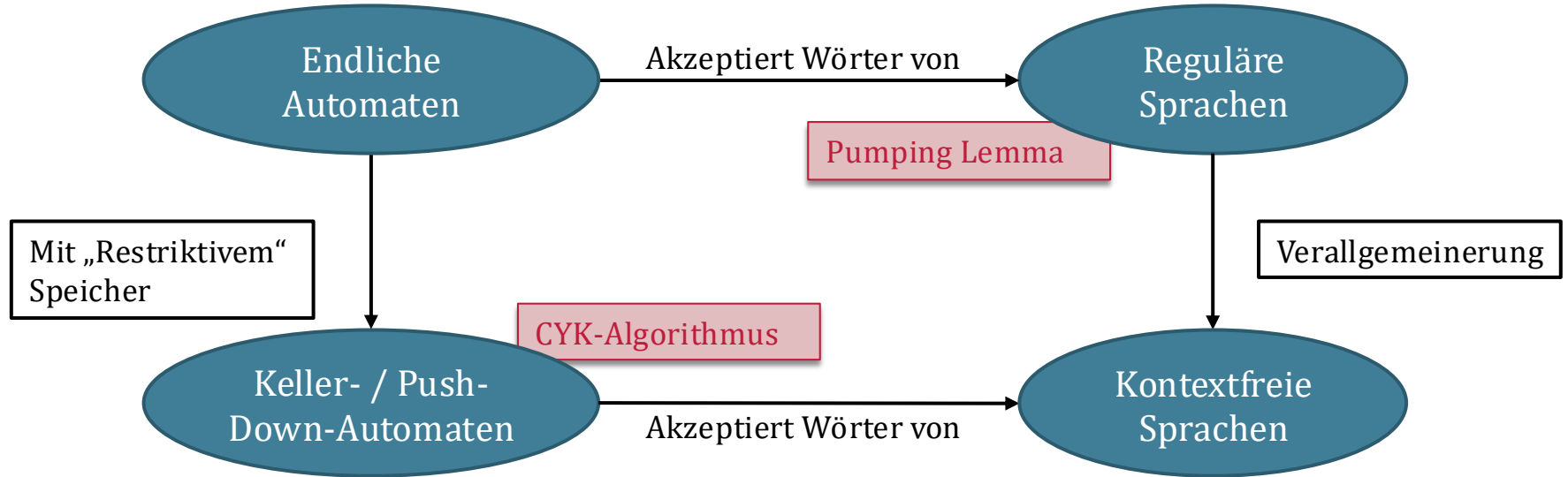
Ort:  
TBA



Inhalt:  
Hauptsächlich  
Vorlesung

# Überblick

# Theoretische Informatik 1

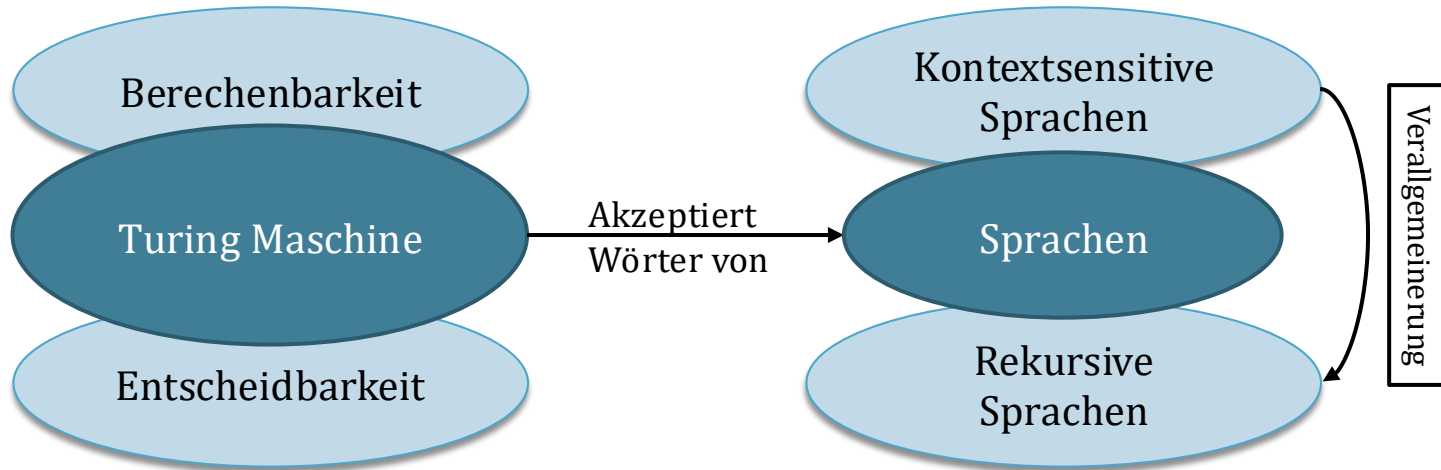


Generelle Fragen:

Was können die Automaten berechnen? (Berechnungsmächtigkeit)

Besitzen Automaten bestimmte Eigenschaften? (Verifikation)

# Theoretische Informatik 2 – Part 1

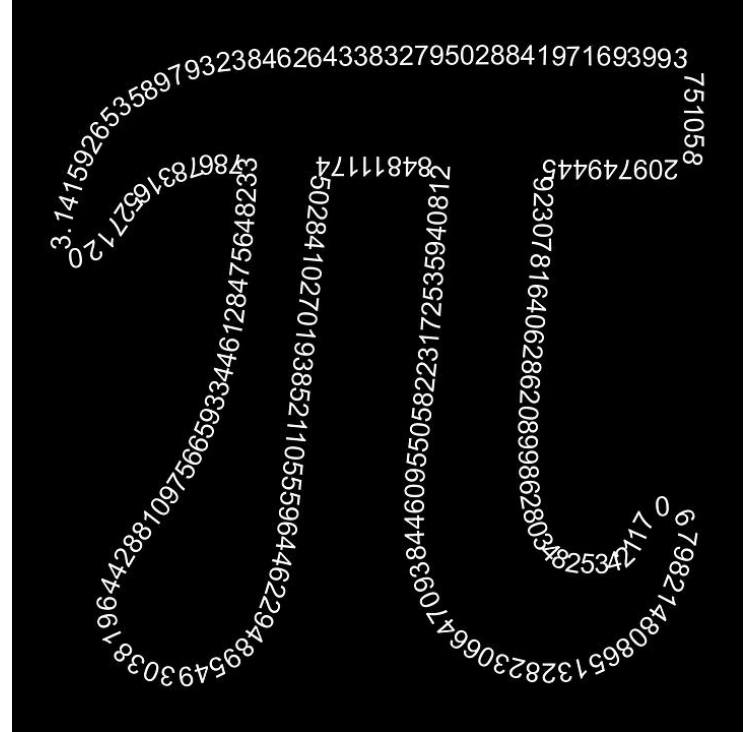


# Berechenbarkeit

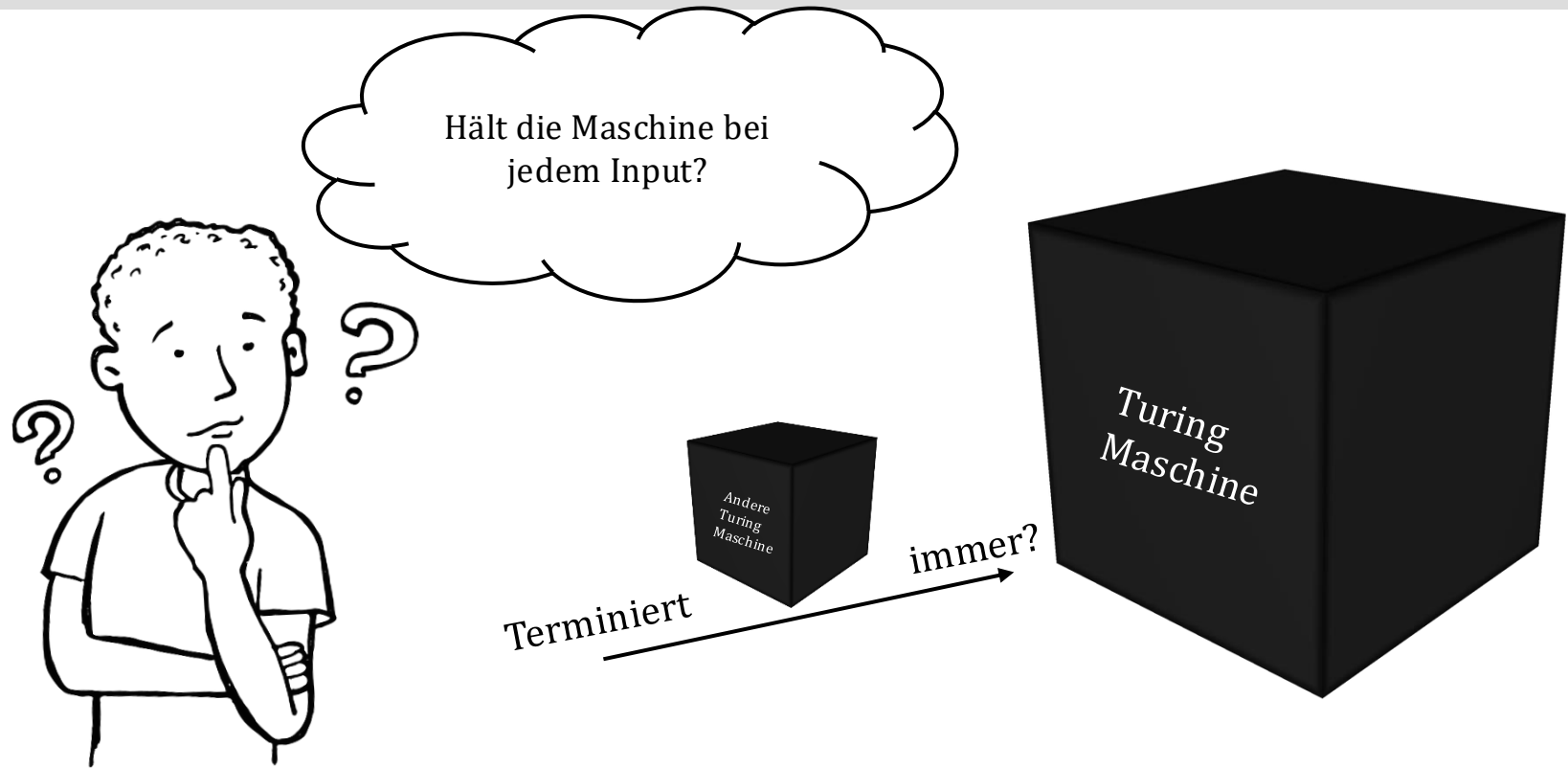


Enthält  $\pi$  die Zahlenfolge  
1597367594816754391  
0832418054193875461  
3958619854?

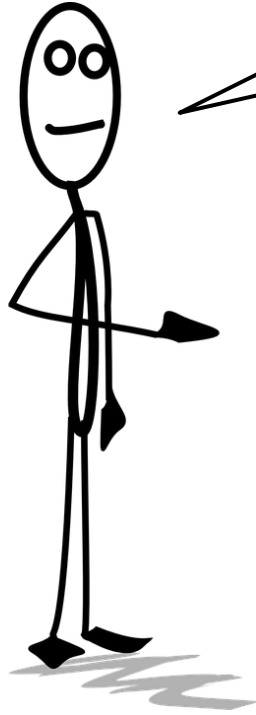
Man kann  $\pi$  beliebig approximieren.  
Aber bis zu welcher Stelle müssen  
wir das tun? Solange bis wir die  
Folge gefunden haben? Was  
passiert, wenn sie nicht existiert?!



# Entscheidbarkeit



# Mächtigkeit



These: Alles, was sich berechnen lässt, kann auch mit einer Turing Maschine berechnet werden.

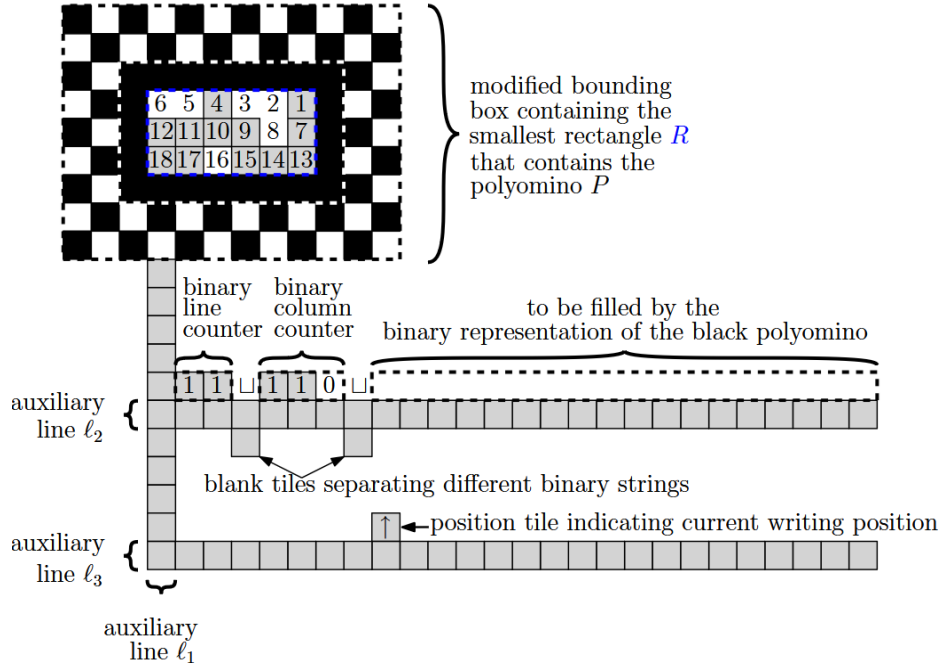
Viele andere Berechnungsmodelle sind genauso mächtig:

- Programmiersprachen: C, Java, Assembler, ...
- Real-RAM-Model
- Lambda-Kalkül
- LaTeX
- PowerPoint, Excel, ...
- Magic the Gathering, Minecraft, Opus Magnum, ...

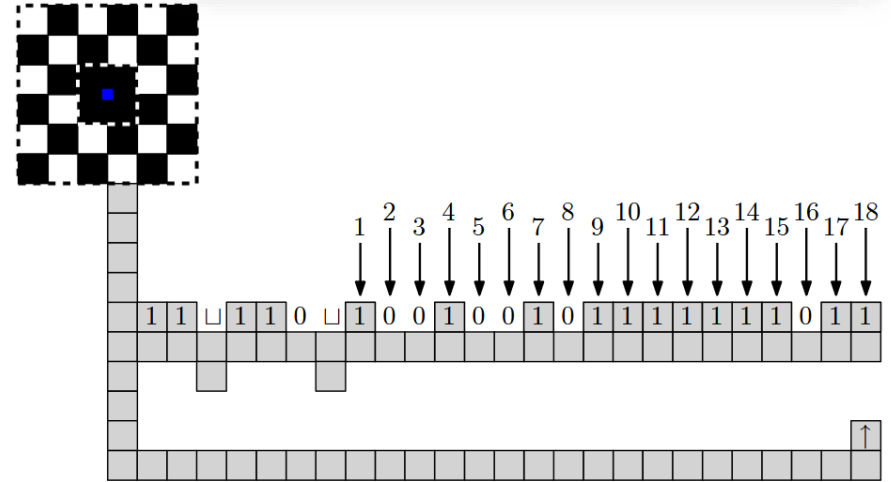
# Turing Maschinen in der Forschung

CADbots: Algorithmic Aspects of Manipulating Programmable Matter with Finite Automata

Sándor P. Fekete<sup>1</sup> · Robert Gmyr<sup>2</sup> · Sabrina Hugo<sup>1</sup> · Phillip Keldenich<sup>1</sup> · Christian Scheffer<sup>1</sup> · Arne Schmidt<sup>1</sup>

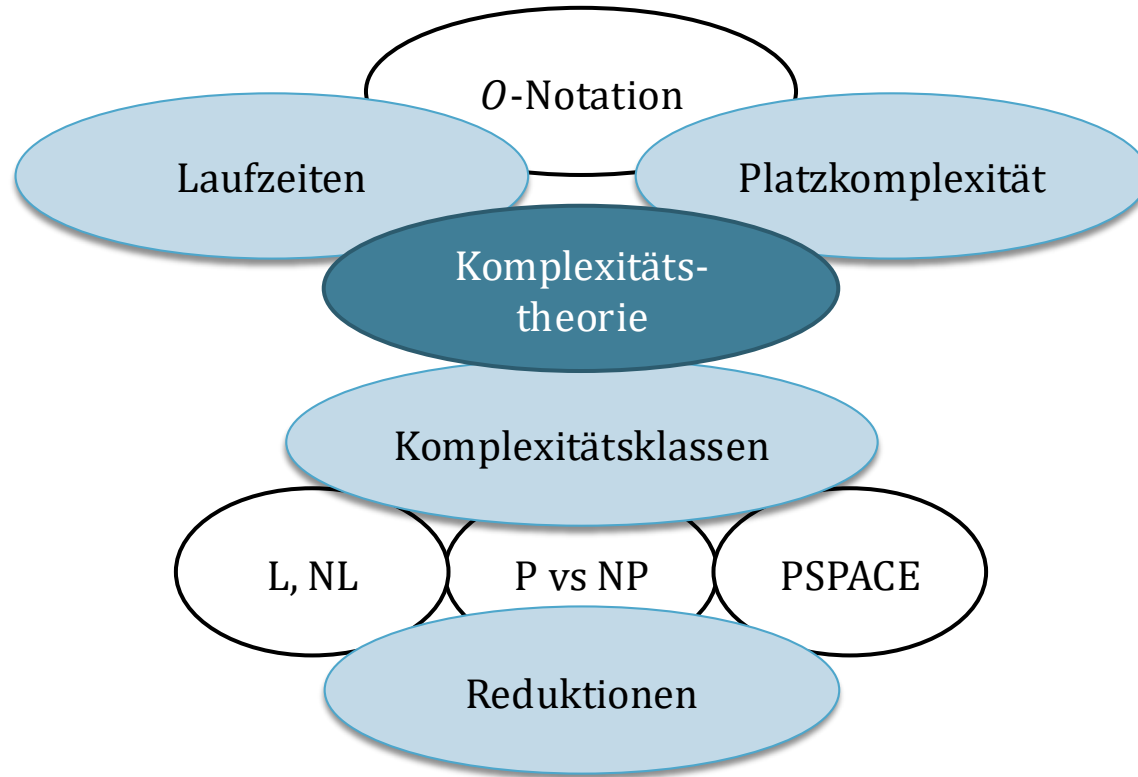


(a) State after counting numbers of lines and columns.

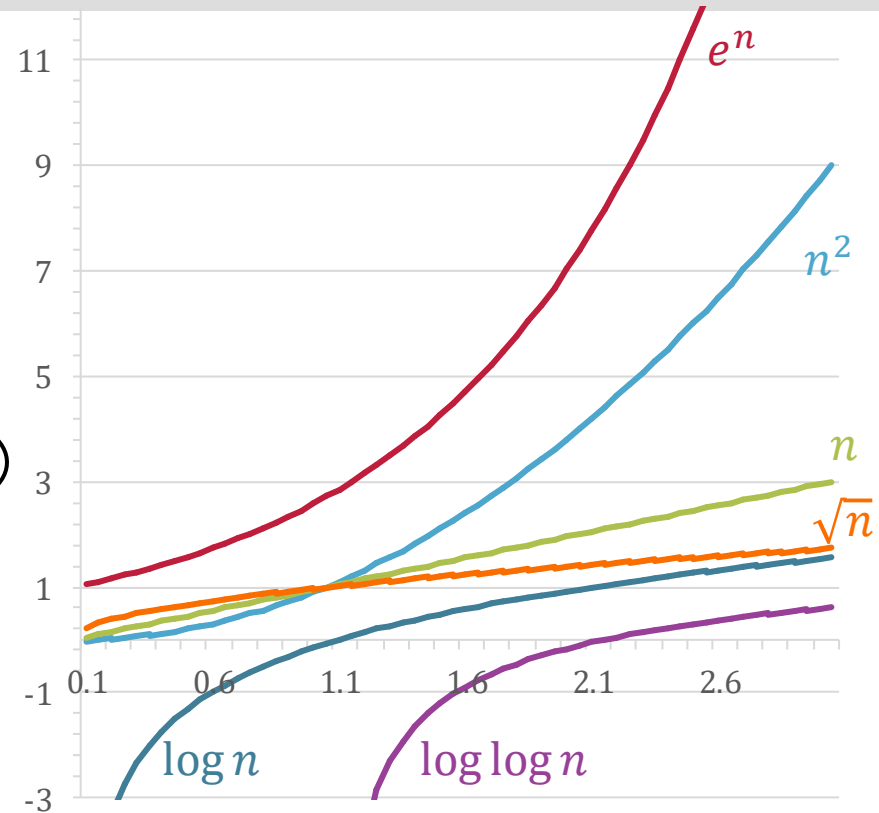


(c) Final state of Phase (2) after writing the binary representation of the input polyomino.

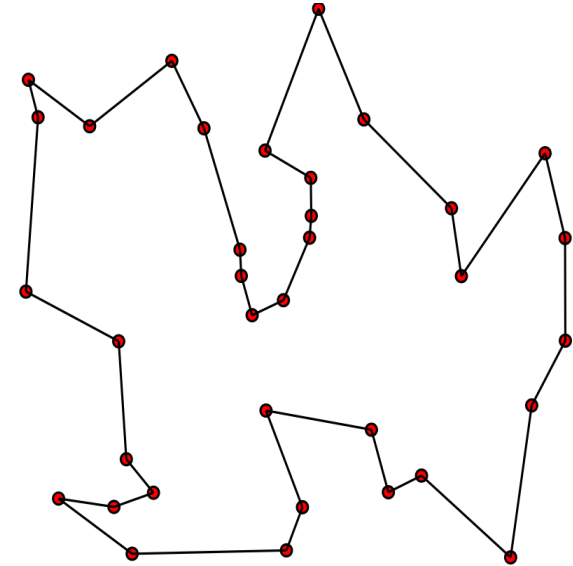
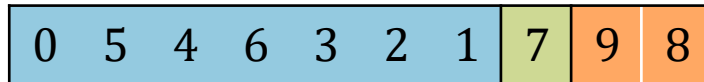
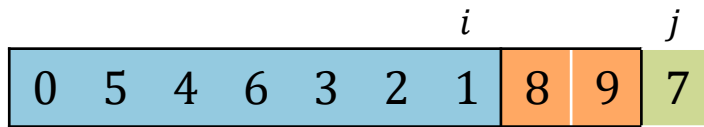
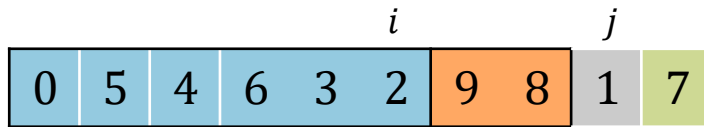
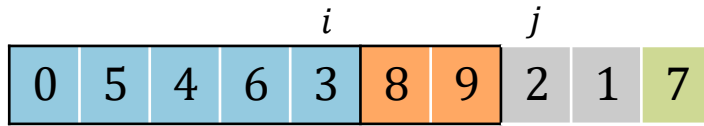
# Theoretische Informatik 2 – Part 2



# Laufzeiten



# Probleme



## Traveling Salesman Problem

Vermutlich existiert kein „effizienter“ Algorithmus

Sortieren

Best-Case

Average-Case

Worst-Case

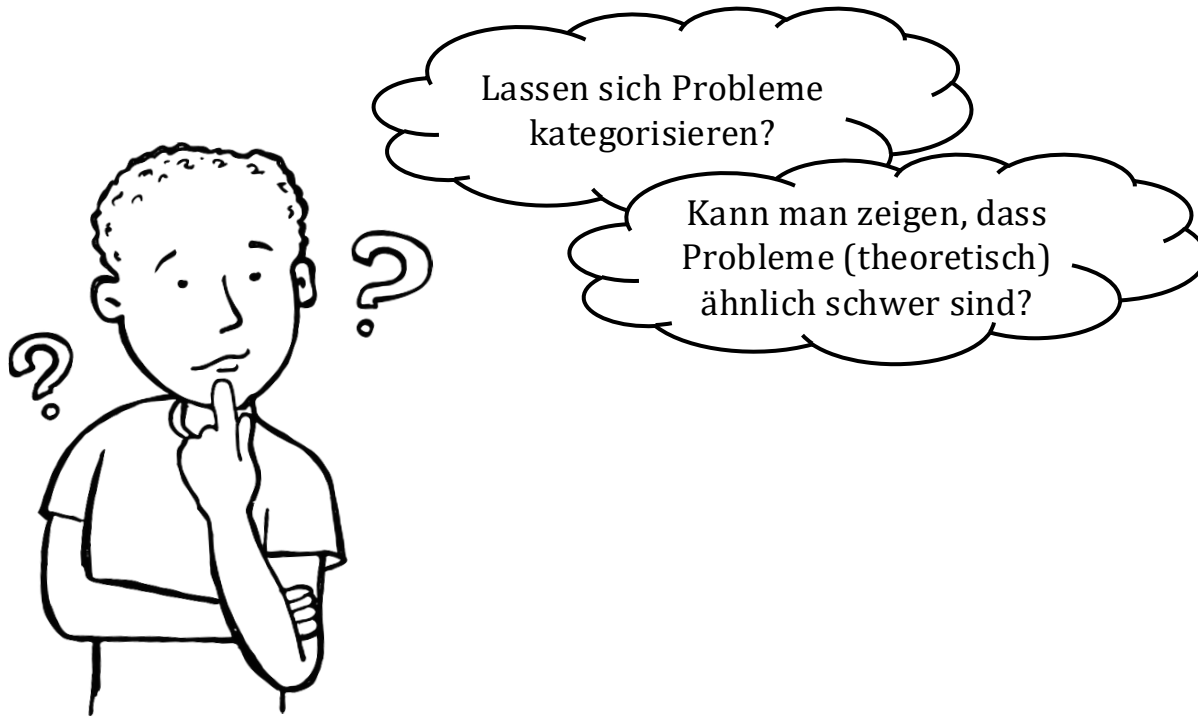
Quicksort

$\Theta(n \log n)$

$\Theta(n \log n)$

$\Theta(n^2)$

# Kategorien



# Part I – Berechenbarkeit & Entscheidbarkeit

# Kapitel 1 – Kontextsensitive Sprachen

# Beispiel 1.2

Betrachte folgende Sprache

$$\mathcal{L} = \{w \in \{a, b, c\}^* \mid \#_a(w) = \#_b(w) = \#_c(w)\}$$

Zeige:  $\mathcal{L}$  ist kontextsensitiv.

Wir erstellen eine Typ-1-Grammatik  $G = (\{S, R, A, B, C\}, \{a, b, c\}, P, S)$ , die  $\mathcal{L}$  erzeugt:

$$S \rightarrow \varepsilon \mid R$$

$$R \rightarrow ABC \mid ABCR$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$C \rightarrow c$$

Erstelle Wörter mit gleich  
vielen a's, b's und c's

$$AB \rightarrow BA$$

$$AC \rightarrow CA$$

$$BA \rightarrow AB$$

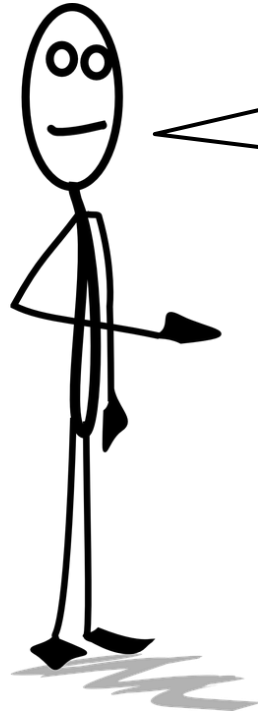
$$BC \rightarrow CB$$

$$CA \rightarrow AC$$

$$CB \rightarrow BC$$

Buchstaben dürfen in belie-  
biger Reihenfolge stehen.

# Unterschied zu kontextfrei



Kontextfreie Grammatiken erlauben  
in Produktionen nur genau ein  
Nicht-Terminal auf der linken Seite

# Wortproblem

## Theorem 1.3

Für jede kontextsensitive Grammatik  $G = (N, \Sigma, P, S)$  kann das Wortproblem für  $\mathcal{L}(G)$  („ist  $w \in \mathcal{L}(G)$ ?“) in exponentieller Zeit gelöst werden.

Da durch Ableiten die Satzform nur länger werden kann, können wir einfach alle Satzformen der Länge  $\leq |w|$  enumerieren und testen, ob  $w$  darunter ist.

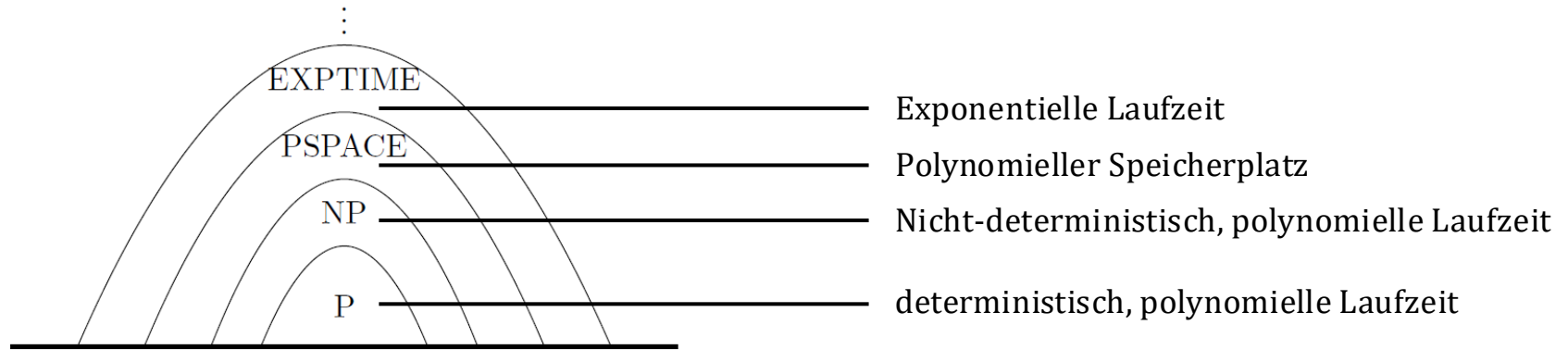
Das sind maximal  $(|N| + |\Sigma| + 1)^{|w|}$  viele Möglichkeiten.

Für eine fixe Grammatik, kann  $|N| + |\Sigma| + 1$  als konstant angesehen werden.

Damit wächst die Laufzeit in etwa wie  $2^{c \cdot |w|}$  mit  $c = \log(|N| + |\Sigma| + 1)$ .

# Ausblick und Bemerkung 1.4

Das Wortproblem lässt sich mit polynomiell viel Speicherplatz (in  $|w|$ ) lösen.



PSPACE ist eine wichtige Komplexitätsklasse für Verifikationsprobleme:

- Erreichbarkeit in Booleschen While-Programmen
- Erreichbarkeit in Multithreaded-Programmen
- Rekonfigurationsprobleme in der (Schwarm-)Robotik

# Nächstes Mal



Endliche und Keller-Automaten  
können nicht benutzt werden,  
um solche Wörter zu erkennen.

Wir benötigen also einen  
mächtigeren Automaten.

