

## Begrüßung, Organisatorisches und Inhaltsübersicht

Prof. Dr. David Sabel

LFE Theoretische Informatik



Letzte Änderung der Folien: 14. Mai 2019

## Zielgruppe der Veranstaltung (Hörerkreis)

- Studierende der Informatik
- Studierende der Bioinformatik
- Studierende im Lehramt
- Studierende im Nebenfach Informatik

## Personen

- Prof. Dr. David Sabel  
Büro: Zimmer L 107 (Oettingenstr. 67)  
Email: david.sabel@ifi.lmu.de
- Wissenschaftliche Mitarbeiter:  
Stephan Barth  
Sebastian Sturm
- Tutoren und Korrektoren:  
Elisabeth Lempa  
David Tellenbach  
Xingyu Long

Email-Adresse, die alle obigen Personen erreicht:

**fsk-ss19@tcs.ifi.lmu.de**

## Struktur der Veranstaltung (Präsenzveranstaltungen)

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
8-10					
10-12		Übung (Stephan Barth)			
12-14			Vorlesung		
14-16					
16-18	Übung (Sebastian Sturm)	Übung (David Sabel)		Vorlesung	Zentralübung
18-20	Übung (Elisabeth Lempa)	Übung (David Tellenbach)			

Zentralübung: Vertiefung und ausführlichere Beispiele zur Vorlesung  
Gruppenübungen: Besprechung und Vorbereitung der Hausaufgaben

## Webseite

### Webseite zur Veranstaltung:

[www.tcs.ifl.lmu.de/lehre/ss-2019/fsk](http://www.tcs.ifl.lmu.de/lehre/ss-2019/fsk)

### Anmeldung zur Veranstaltung:

Anmeldung zur Veranstaltung mit Uni2Work:

[uni2work.ifl.lmu.de](http://uni2work.ifl.lmu.de)

- Nachfolger von UniWorX
- noch im Probebetrieb

Anmeldung ist **notwendig** für:

- Zugriff auf Material z.B. Hausaufgaben
- Abgabe & Korrektur der Hausaufgaben
- Anmeldung zur Klausur (wird noch implementiert)

Keine Anmeldung für die einzelnen Übungstermine.

Bitte verteilen Sie sich selbst.

## Korrektur und Bonuspunkte

- Ausgewählte Hausaufgaben werden bepunktet
- Für jede Lösung zu einer bepunkteten Aufgabe gibt es 0 oder 1 Punkt (1 Punkt bei vollständiger und sinnvoller Bearbeitung)
- Alle abgegebenen Aufgaben werden korrigiert.

### Bonusregelung: (gilt für Klausur und Nachklausur im SoSe 2019)

100% der erreichbaren Übungspunkte entsprechen 10% der Klausurpunkte

$$\text{Klausurbonus} = \frac{\text{erreichte Übungspunkte}}{\text{maximale Übungspunkte}} \cdot 0,1 \cdot \text{maximale Klausurpunkte}$$

#### wenn

- die Klausur bestanden ist (Bonuspunkte helfen **nicht zum Bestehen**)
- **eine** (Teil-)aufgabe in der Übung **vorgerechnet** wurde

Die Klausur ist auf jeden Fall bestanden, wenn 50% der Klausurpunkte erreicht wurden.

## Hausaufgaben

Montag			Abgabe Blatt $x$ bis Mo 9:00 Uhr
Dienstag			Übungen Mo&Di: Besprechung Blatt $x$
Mittwoch			
Donnerstag			
Freitag	Ausgabe Blatt $x$	Ausgabe Blatt $x + 1$	Ausgabe Blatt $x + 2$
Samstag			
Sonntag			
	Woche $x$	Woche $x + 1$	Woche $x + 2$

- Übungen nächste Woche: Besprechung Blatt 0 (ohne Abgabe)
- Ausgabe, Abgabe und Korrektur nur elektronisch über Uni2Work
- Klausurbonus für erfolgreiches Bearbeiten der Aufgaben

## Klausuren

- Bearbeitungszeit der Klausur: 120 Minuten
- Klausur am Di 30.7.2019 um 16 Uhr
- Nachklausur am Di 24.9.2019 um 14 Uhr
- Anmeldung zur Klausur wird noch freigeschaltet (spätestens 4 Wochen vor der Klausur)
- Bonuspunkte gelten für Klausur und Nachklausur
- Teilnahme an der Nachklausur auch ohne Teilnahme an der Klausur möglich

- Vorlesungsfolien
- Skript zur Vorlesung (wird nach und nach bereit gestellt)
- Lehrbuch: Uwe Schöning, Theoretische Informatik – Kurz gefasst
- Hausaufgaben (Übungsblätter im Uni2Work)

## Inhalte der Veranstaltung

Drei große wesentliche Themen der Theoretischen Informatik:

- 1 Formale Sprachen und Automatentheorie  
*Wie stellt man Entscheidungsprobleme formal dar?*
- 2 Berechenbarkeitstheorie  
*Welche Probleme kann man algorithmisch (bzw. mit dem Computer) überhaupt lösen?*
- 3 Komplexitätstheorie  
*Welche Probleme kann man in annehmbarer Zeit lösen?*

Wesentliche Quellen:

- Vorlesungsskript
- Uwe Schöning: Theoretische Informatik - kurz gefasst, 5. Auflage, Spektrum Akademischer Verlag, 2008 (ältere Auflagen sind auch in Ordnung)

Weitere Literatur

- Alexander Asteroth und Christel Baier: Theoretische Informatik, Pearson Studium 2002.
- John E. Hopcroft, Rajeev Motwani und Jeffrey D. Ullman: Introduction to Automata Theory, Languages, and Computation, 3. Auflage, 2006
- Ingo Wegener: Theoretische Informatik - eine algorithmenorientierte Einführung, 3.Auflage, Teubner Verlag, 2005.
- ...

## Inhalte: Formale Sprachen und Automatentheorie

- Chomsky-Hierarchie und -Grammatiken zur Klassifizierung Darstellung von formalen Sprachen
- Das Wortproblem und weitere Entscheidungsprobleme
- Reguläre Sprachen: reguläre Grammatiken, deterministische endliche Automaten, nichtdeterministische endliche Automaten,  $\epsilon$ -Übergänge, reguläre Ausdrücke, Äquivalenz der Formalismen, Pumpinglemma, Satz von Myhill-Nerode, Minimalautomaten, Abschlusseigenschaften
- Kontextfreie Sprachen: kontextfreie Grammatiken, Chomsky-Normalform, Greibach-Normalform, Pumpinglemma, Ogden's Lemma, Cocke-Younger-Kasami-Algorithmus, Kellerautomaten, Abschlusseigenschaften
- Kontextsensitive Sprachen und Typ 0-Sprachen: kontextsensitive Grammatiken, Kuroda-Normalform, Turingmaschinen, Linear bounded automata (LBA), LBA-Probleme

- Intuitive Berechenbarkeit, Churchsche These
- Turing-Berechenbarkeit, Varianten von Turingmaschinen (z.B. Mehrbandmaschinen)
- LOOP-, WHILE-, GOTO-Berechenbarkeit: LOOP-Programme, WHILE-Programme, GOTO-Programme, Äquivalenz zu Turingmaschinen
- Primitiv-rekursive Funktionen, Ackermannfunktion,  $\mu$ -Rekursion
- Halteproblem, Unentscheidbarkeit
- Rekursiv aufzählbar
- Reduktionen
- Postsches Korrespondenzproblem

- Zeitkomplexität
- Klassen P und NP
- NP-Härte, NP Vollständigkeit
- polynomielle Reduktionen
- das SAT-Problem
- Satz von Cook
- weitere NP vollständige Probleme (z.B. 3-SAT, Clique, Vertex Cover, Subset Sum, Knapsack, Directed Hamiltonian Circuit, Hamiltonian Circuit, . . .)