



Technische
Universität
Braunschweig



Theoretische Informatik 2

Arne Schmidt

Das Ende!

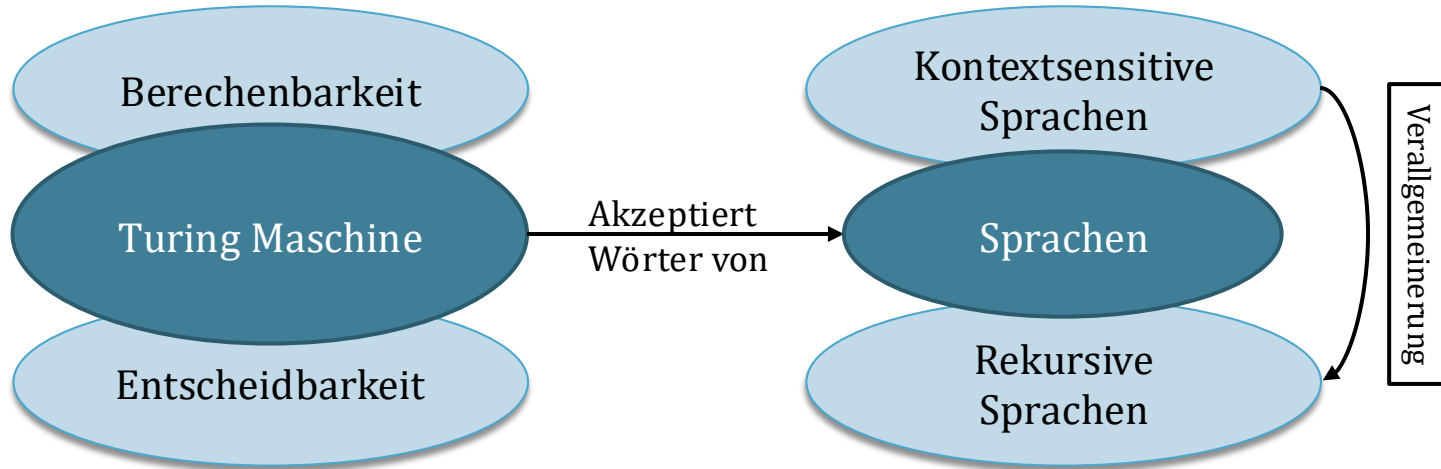
Vorlesungszeit
endet



Klausurenphase
beginnt



Theoretische Informatik 2 – Part 1



TMs, Sprachen und Entscheidbarkeit

Typische Aufgaben:

- Konstruiere eine TM, die eine Sprache (semi-)entscheidet
- Konstruiere eine TM, die eine (partielle) Funktion berechnet
- Analysiere eine gegebene TM
- Analysiere eine Grammatik
- Zeige (Über-)Abzählbarkeit
- Zeige (Un-)Berechenbarkeit / (Un-)Entscheidbarkeit
- Wende Satz von Rice an, betrachte nicht-monotone Eigenschaften

Kenne:

- Nicht-semi-entscheidbare Probleme
- Nicht-co-semi-entscheidbare Probleme
- Nicht-co- und nicht-semi-entscheidbare Probleme



(Nicht-)Entscheidbare Probleme

Semi-entscheidbar, aber nicht co-semi-entscheidbar:

- **ACCEPT** (Akzeptiert eine gegebene TM M das Wort x?)
- **HP** (Hält eine gegebene TM M auf dem Wort x?)

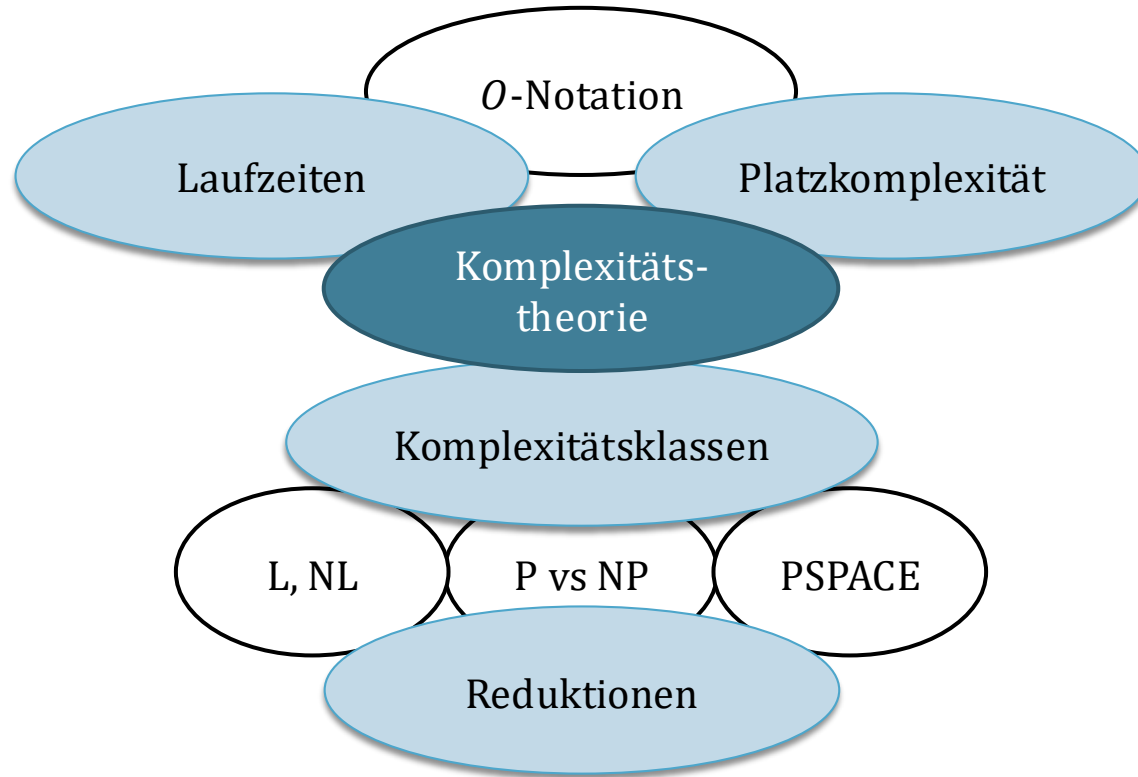
Co-Semi-Entscheidbar, aber nicht semi-entscheidbar

- **Emptiness** (Hält eine gegebene TM M nie?)

Weder co-semi-entscheidbar, noch semi-entscheidbar:

- **Totality** (Hält eine gegebene TM M immer?)

Theoretische Informatik 2 – Part 2



Algorithmen und Reduktionen

Typische Aufgaben:

- Zeige Problem ist in der Klasse K
- Zeige Problem ist K -schwer
- Zeige Problem ist K -vollständig

Unterscheide:

- Logspace-Reduktion
- Polyzeit-Reduktion

Besitze:

- Für jede Klasse ein paar Probleme
(vor allem für NL bis $PSPACE$)



Masterprobleme

Klasse	Masterproblem	Notiz
NL	Wegsuche in Graphen	Existiert in einem gegebenen Graphen ein Weg zwischen zwei Knoten?
P	Circuit Value Problem	Eingänge sind bekannt und müssen nur ausgewertet werden.
NP	Circuit-Sat / 3SAT	Eingänge sind variabel. Existiert eine erfüllende Belegung?
PSPACE	Quantified Sat	Variablen sind mit Existenz- und Allquantoren versehen.

Weitere Aufgabetypen

Typische Aufgaben:

- Zeige systematisch:
 - Ein Wort gehört nicht zu einer Grammatik.
 - Ob eine TM ein gegebenes Wort akzeptiert.
 - Ob eine 2SAT-Formel erfüllbar ist.
 - ...
- Neue Definitionen
 - Problemdefinition + Beispiele analysieren.
 - Modelldefinition + Mächtigkeit analysieren.
 - ...

Achtung: Diese Listen sind ggf. nicht vollständig, sodass auch andere Aufgabentypen vorkommen können

Das Ende?



Was macht Probleme
wirklich schwer?

Was ist der **Kern** einer
schweren Instanz?

Unter welchen
Betrachtungen lassen sich
Probleme effizient lösen?

Vorlesung im Winter:
Parametrisierte Algorithmen

There is more!



Wie kann man
Probleme mit mehreren
Prozessoren lösen?

Welcher Speed-Up ist
möglich?

Gibt es Probleme, die nicht
effizient parallelisierbar sind?

Neue Vorlesung im Winter:
Einführung in parallele und verteilte Algorithmen

Veranstaltungen im Winter

Veranstaltung	Empfehlung	Inhalte
Computational Geometry (Wahlpflicht Masterbereich)	AuD 1 bestanden	Konvexe Hüllen, Triangulationen, Standortprobleme, Tourprobleme, ...
Mathematische Methoden der Algorithmik (Wahlpflicht Masterbereich)	Lineare Algebra bestanden Interesse an lin Alg	(Ganzzahlige) lineare Optimierung Simplex-Algorithmus Dualität
Parametrisierte Algorithmen (Wahlpflicht Informatik)	AuD 1 + 2 bestanden (Theo Inf 2 bestanden)	Kernelisation FPT, W-Hierarchie Komplexitätstheorie "Was macht Probleme schwer zu lösen?"
Einführung in parallele und verteilte Algorithmen (Neu! Wahlpflicht Informatik)	AuD 1 + 2 bestanden (Theo Inf 2 bestanden)	Modelle für paralleles Rechnen Sortier- / Graphalgos. P-Schwere Probleme "Wie gut kann man Probleme parallelisieren?"