

# Themenüberblick

Dies ist eine Art Checkliste für die Klausurvorbereitung. Zu jedem Thema im Skript sind hier ein paar Leitfragen aufgelistet. Besonders nützlich sind die Tabellen und Abbildungen auf den letzten Seiten.

## 1 Formale Sprachen

- Wie zeigt man Mengeninklusionen  $A \subseteq B$ ?
- Wie sind die Operationen  $\bar{A}$ ,  $A^*$ ,  $A^+$  und  $AB$  definiert?

## 2 Phrasenstrukturgrammatiken

- Wann ist eine Grammatik monoton?
- Wann ist eine Grammatik regulär bzw. rechtslinear?
- Wann ist eine Sprache regulär bzw. rechtslinear?
- Wann ist eine Grammatik kontextfrei?
- Wann ist eine Sprache kontextfrei?
- Wann ist eine Grammatik kontextsensitiv?
- Wann ist eine Sprache kontextsensitiv?
- Wann ist eine Grammatik rechtslinear?
- Wann ist eine Sprache rechtslinear?

## 3 Deterministische endliche Automaten

- Wie sind deterministische endliche Automaten (DFAs) definiert?
- Was ist der Unterschied zwischen  $\delta$  und  $\hat{\delta}$ ?

## 4 Nichtdeterministische endliche Automaten

- Wie sind nichtdeterministische endliche Automaten (NFAs) definiert?
- Was ist der Unterschied zwischen  $\delta$  und  $\hat{\delta}$ ?
- Wie kann man ein NFA in ein DFA überführen? (Myhill- bzw. Potenzmengenverfahren)

## 5 NFAs mit $\epsilon$ -Übergängen

- Wie sind NFAs mit  $\epsilon$ -Übergängen ( $\epsilon$ -NFAs) definiert?
- Wie kann man ein  $\epsilon$ -NFA in ein NFA überführen? (Sättigungsverfahren)

## 6 Reguläre Ausdrücke

- Wie sind reguläre Ausdrücke (REs) definiert?
- Wann sind zwei REs äquivalent?
- Wie kann man ein RE in ein  $\epsilon$ -NFA überführen? (Thompson-Verfahren)

## 7 Minimale DFAs

- Wann sind zwei Zustände eines DFAs äquivalent?
- Wann sind zwei Wörter einer Sprache äquivalent?
- Wann ist ein DFA minimal?
- Wie findet man einen minimalen DFA mit dem Minimierungsalgorithmus (Myhill-Nerode)?

## 8 Pumping Lemma für reguläre Sprachen

- Was kann man mit dem Pumping Lemma zeigen und was nicht?
- Wie benutzt man das Pumping-Lemma?

## 9 Abschlusseigenschaften für RLs

(siehe Tabelle 1 auf Seite 6)

- Welche der Sprachen  $\bar{L}$ ,  $L^*$  und  $L^R$  sind für eine reguläre Sprache  $L$  auch regulär?
- Welche der Sprachen  $L_1 \cap L_2$ ,  $L_1 \cup L_2$  und  $L_1L_2$  sind für reguläre Sprachen  $L_1$  und  $L_2$  auch regulär?

## 10 Entscheidbarkeit bei RLs

(siehe Tabelle 2 auf Seite 6)

- Ist das Wortproblem für reguläre Sprachen entscheidbar?
- Ist das Leerheitsproblem für reguläre Sprachen entscheidbar?
- Ist das Schnittproblem für reguläre Sprachen entscheidbar?
- Ist das Äquivalenzproblem für reguläre Sprachen entscheidbar?

## 11 Kontextfreie Grammatiken

- Was ist eine Ableitung eines Wortes?
- Was ist ein Syntabaum eines Wortes?
- Wann ist eine kontextfreie Grammatik (CFG) eindeutig?
- Wie zeigt man induktiv, dass alle Wörter aus einer Grammatik eine bestimmte Eigenschaft besitzen?

## 12 CYK-Algorithmus

- Wie ist die Chomsky-Normalform (CNF) für CFGs definiert?
- Wie kann man eine CFG in CNF überbringen?
- Was kann man mit dem CYK-Algorithmus zeigen?
- Wie funktioniert der CYK-Algorithmus?

## 13 Pumping Lemma und Ogdens Lemma für CFLs

- Was kann man mit dem Pumping Lemma zeigen und was nicht?
- Wie benutzt man das Pumping-Lemma?
- Was ist der Unterschied zum Ogdens-Lemma?

## 14 Kellerautomaten

- Wie sind Kellerautomaten (PDAs) definiert?
- Wie kann man aus einer CFG einen PDA erzeugen, der dieselbe Sprache erkennt?
- Wann ist ein Kellerautomat deterministisch (DPDA)?
- Wann ist ein DPDA in Normalform?
- Was besagt die Fano-Bedingung?
- Welche Sprachklassen werden von PDAs und DPDAs beschrieben?

## 15 LR( $k$ )- und LL( $k$ )-Grammatiken

- Was sind LR( $k$ )- und LL( $k$ )-Grammatiken?
- Welche Eigenschaften besitzen sie?

## 16 Earley's Algorithmus

- Was kann man mit Earley's Algorithmus berechnen?
- Welche Laufzeit besitzt der Algorithmus?

## 17 Abschlusseigenschaften für CFLs

(siehe Tabelle 1 auf Seite 6)

- Welche der Sprachen  $\bar{L}$ ,  $L^*$  und  $L^R$  sind für eine kontextfreie Sprache  $L$  auch kontextfrei?
- Welche der Sprachen  $L_1 \cap L_2$ ,  $L_1 \cup L_2$  und  $L_1 L_2$  sind für kontextfreie Sprachen  $L_1$  und  $L_2$  auch kontextfrei?

## 18 Entscheidbarkeit bei CFLs

(siehe Tabelle 2 auf Seite 6)

- Ist das Wortproblem für kontextfreie Sprachen entscheidbar?
- Ist das Leerheitsproblem für kontextfreie Sprachen entscheidbar?
- Ist das Schnittproblem für kontextfreie Sprachen entscheidbar?
- Ist das Äquivalenzproblem für kontextfreie Sprachen entscheidbar?

## 19 Turingmaschinen

- Wie funktionieren Turingmaschinen (TMs)?
- Wann ist eine TM deterministisch (DTM)?
- Wann ist eine TM linear beschränkt (LBA)?
- Welche Sprachklassen werden durch TMs, DTMs und LBAs beschrieben?
- Unter welchen Operationen sind diese Klassen abgeschlossen? (siehe Tabelle 1 auf Seite 6)
- Welche Probleme sind für diese Klassen entscheidbar? (siehe Tabelle 2 auf Seite 6)

## 20 Berechenbarkeit

- Wann ist eine Funktion total?
- Wann ist eine Funktion (Turing-)berechenbar?

## 21 WHILE-Berechenbarkeit

- Wie sind WHILE-Programme definiert?

## 22 GOTO-Berechenbarkeit

- Wie sind GOTO-Programme definiert?

## 23 Primitiv-rekursive Funktionen

- Wie sind primitiv-rekursive (PR) Funktionen definiert?
- Wie beweist man, dass eine Funktion PR ist?
- Welche Erweiterungen gibt es für die Definition?
- Was machen die Cantorsche Paarungsfunktion  $p$  und ihre Umkehrfunktionen  $c_1, c_2$ ?

## 24 LOOP-Berechenbarkeit

- Wie sind LOOP-Programme definiert?
- Wie kann man ein LOOP-Programm durch ein WHILE-Programm simulieren?

## 25 $\mu$ -rekursive Funktionen

- Wie sind  $\mu$ -rekursive Funktionen definiert?
- Wie bestimmt man für eine Funktion  $f$  die Funktion  $f_\mu$ ?

## 26 Entscheidbarkeit

- Wann ist eine Sprache, eine Menge oder ein Prädikat entscheidbar (rekursiv) bzw. semi-entscheidbar (rekursiv aufzählbar)?
- Wie sind  $\chi_A$  und  $\chi'_A$  für eine Sprache  $A$  definiert?

## 27 Unentscheidbarkeit und Halteprobleme

- Wann ist eine Sprache  $A$  auf eine Sprache  $B$  effektiv reduzierbar ( $A \leq B$ )?
- Was kann man alles aus  $A \leq B$  schließen?
- Wie sind die Halteprobleme  $K$  ( $= H_s$ ),  $H$  und  $H_0$  definiert?
- Was besagt der Satz von Rice und wann ist dieser anwendbar?
- Wie beweist man durch Reduktion, dass eine Sprache unentscheidbar ist?

## 28 Komplexitätstheorie

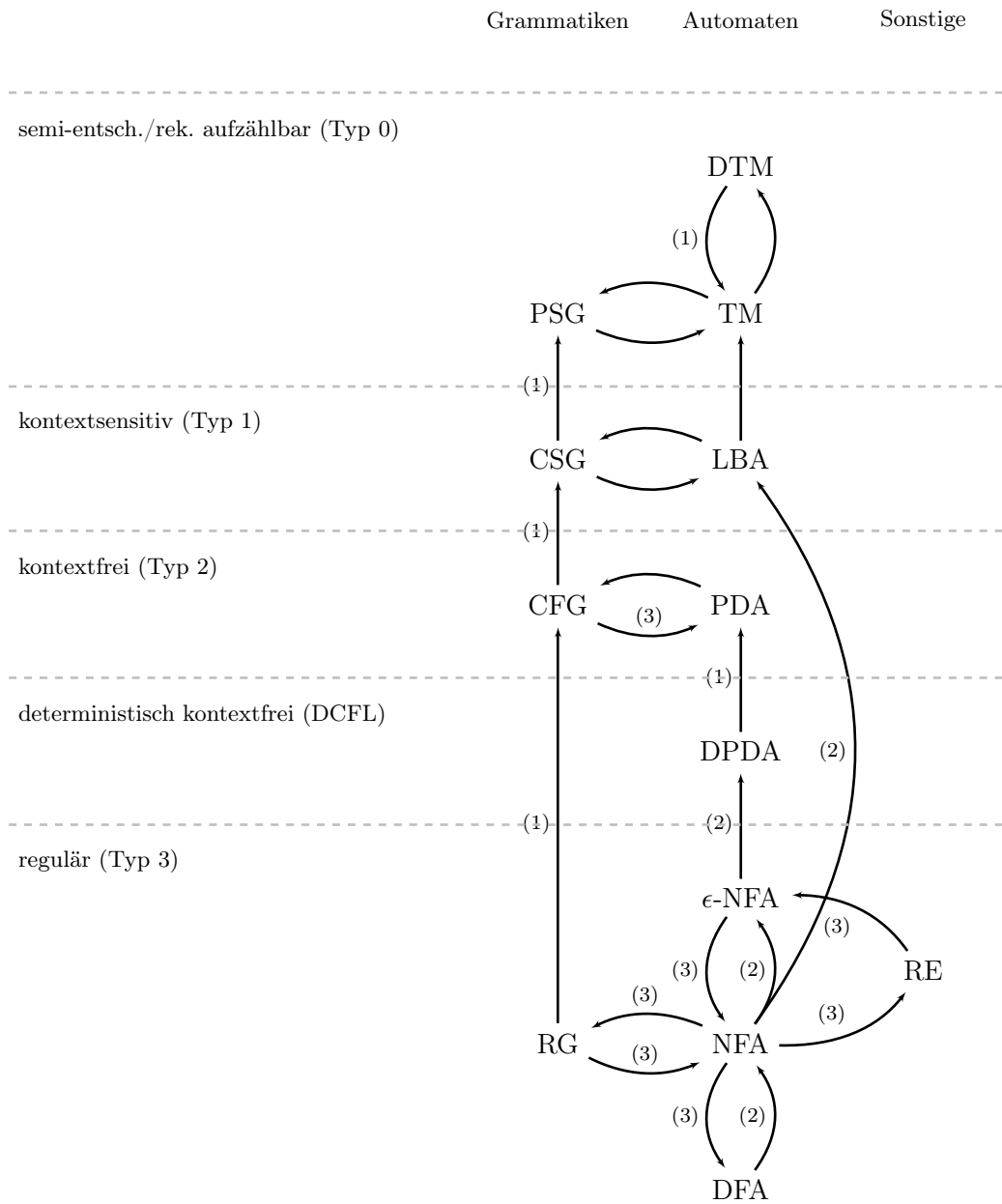
- Wie sind  $\text{TIME}_M$ ,  $\text{NTIME}_M$ ,  $\text{SPACE}_M$  und  $\text{NSPACE}_M$  definiert?
- Wie sind  $\text{DTIME}$ ,  $\text{NTIME}$ ,  $\text{DSPACE}$  und  $\text{NSPACE}$  definiert?
- Wie sind  $\mathcal{P}$ ,  $\mathcal{NP}$ ,  $\mathcal{L}$  und  $\mathcal{NL}$  definiert?

Typ	Name	$\bar{L}$	$L^*$	$L_1 \cap L_2$	$L_1 \cup L_2$	$L_1 L_2$
Typ 0	semi-entscheidbar	✗	✓	✓	✓	✓
Typ 1	kontextsensitiv	✓	✓	✓	✓	✓
Typ 2	kontextfrei	✗	✓	✗	✓	✓
	deterministisch kontextfrei	✓	✗	✗	✗	✗
Typ 3	regulär	✓	✓	✓	✓	✓

Tabelle 1: Abschlusseigenschaften.

Typ	Name	Wort	Leerheit	Äquivalenz	Schnitt
Typ 0	semi-entscheidbar	✗	✗	✗	✗
Typ 1	kontextsensitiv	✓	✗	✗	✗
Typ 2	kontextfrei	✓	✓	✗	✗
	deterministisch kontextfrei	✓	✓	✓	✗
Typ 3	regulär	✓	✓	✓	✓

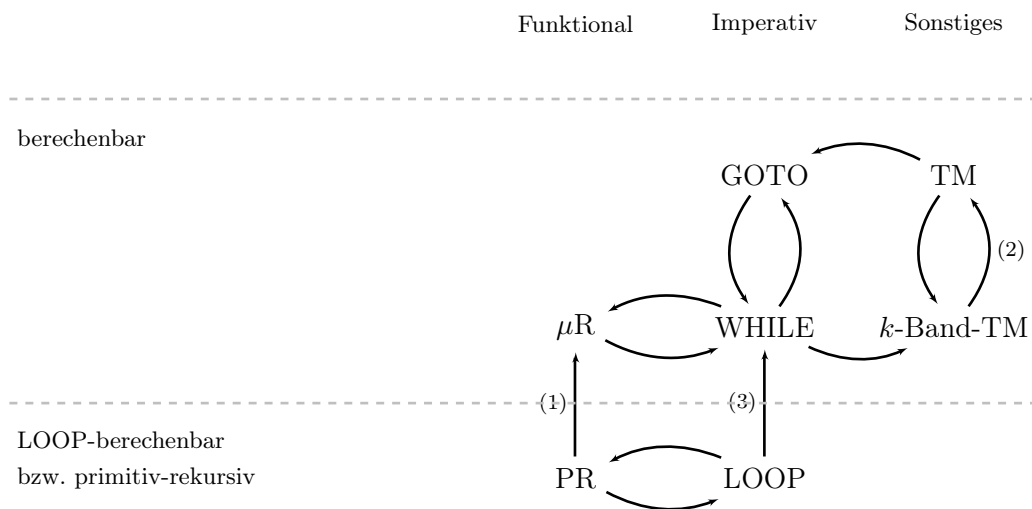
Tabelle 2: Entscheidbare Probleme.



$A \rightarrow B$  bedeutet, dass jede Sprache, für die es ein Modell vom Typ A gibt, auch von einem Modell vom Typ B beschrieben werden kann. Für uns sind folgende drei Fälle relevant.

- (1) A ist ein Spezialfall von B, d.h. keine Überführung ist nötig.
- (2) A verhält sich wie ein Spezialfall von B, müsste aber formal angepasst werden.
- (3) A kann mit einem der gelernten Verfahren in B überführt werden.

Abbildung 1: Maschinenmodelle für Sprachen.



$A \rightarrow B$  bedeutet, dass jede Funktion, für die es ein Modell vom Typ A gibt, auch von einem Modell vom Typ B beschrieben werden kann. Für uns sind folgende drei Fälle relevant.

- (1) A ist ein Spezialfall von B, d.h. keine Überführung ist nötig.
- (2) A verhält sich wie ein Spezialfall von B, müsste aber formal angepasst werden.
- (3) A kann mit einem der gelernten Verfahren in B überführt werden.

Abbildung 2: Maschinenmodelle für Funktionen.



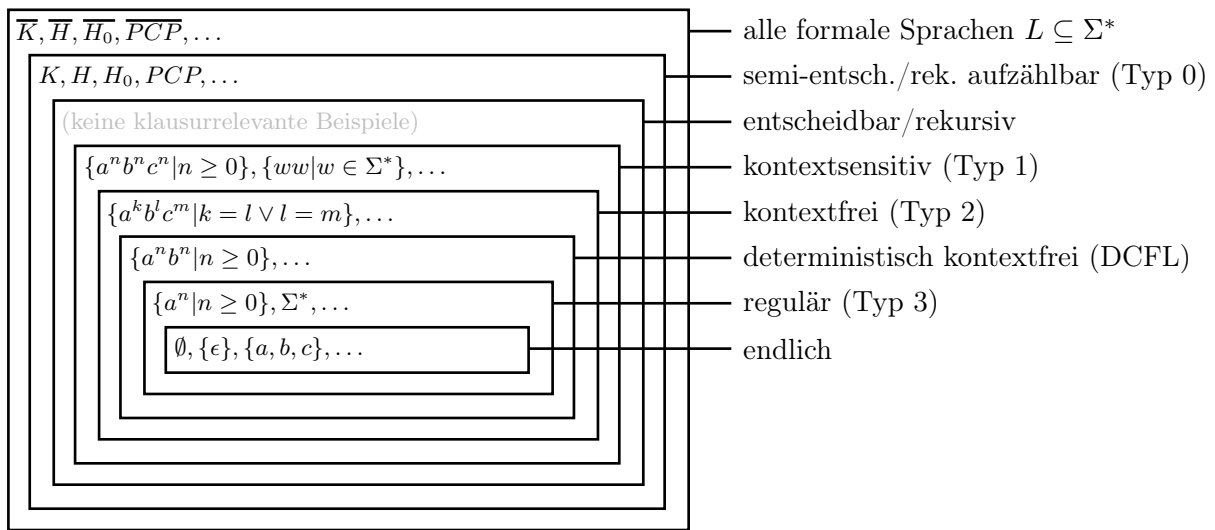


Abbildung 3: Klassen formaler Sprachen

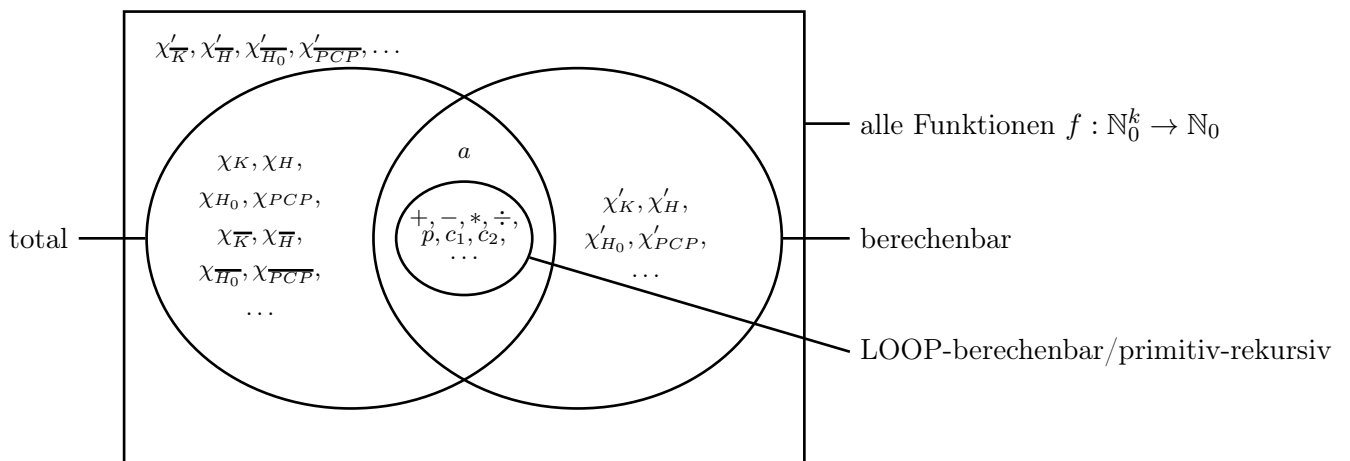


Abbildung 4: Klassen von Funktionen.