

# Inhaltsverzeichnis

1	Klausur relevante Bereiche	3
2	Alphabete, Wörter, Sprachen	4
3	Algorithmen	4
3.1	Anforderungen an Algorithmen . . . . .	5
3.2	Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit . . . . .	5
3.2.1	Entscheidbare Sprachen . . . . .	5
4	Turingmaschinen	6
5	Registermaschinen	8
6	Registermaschinen und Registeroperatoren	9
7	Primitiv rekursive Funktionen	9
8	Rekursiv aufzählbare Mengen	9
9	!Härte und Vollständigkeit	9
10	Kontextfreie Sprachen	9
11	(Rechts-)Lineare Sprachen	9
12	Reguläre Sprachen	9
13	!Reduktionsmethode	9
14	Chomsky-Hierarchie	9
15	!Diagonalisierung	10
16	Komplexitätsklassen	10
17	Entscheidbarkeitsfragen	10
18	Nichtkontextfreie Sprachen	10
19	Endliche Automaten	10
20	Rechtslineare Sprachen / Endliche Automaten	10

<b>21 Kontextfreie vs. rechtslineare Sprachen</b>	<b>10</b>
<b>22 Partiell rekursive Funktionen</b>	<b>10</b>

# 1 Klausur relevante Bereiche

Von den betrachteten 10 Altklausuren ergab zu den folgenden Aufgaben die entsprechende Häufigkeit

Aufgabentyp	Häufigkeit
Turingmaschinen	10
Registeroperatoren	10
Primitiv rekursive Funktionen	11
Rekursiv aufzählbare Mengen	11
Härte und Vollständigkeit	1
Kontextfreie Sprachen	15
(Rechts-)Lineare Sprachen	2
Reguläre Sprachen	4
Reduktionsmethode	5
Chomsky-Hierarchie	1
Registermaschinen	2
Diagonalisierung	1
Komplexitätsklassen	1
Entscheidbarkeitsfragen	1
Nichtkontextfreie Sprachen	3
Endliche Automaten	1
Rechtslineare Sprachen / Endliche Automaten	4
Kontextfreie vs. rechtslineare Sprachen	1
Partiell rekursive Funktionen	1

## 2 Alphabete, Wörter, Sprachen

Wörter sind endliche Zusammensetzungen aus Elementen einer Menge, auch Buchstaben genannt. Da es eine abzählbar unendliche Menge an Zusammensetzungen gibt, kann man Wörter auch durch Zahlen ersetzen.

Ein Wort kann aufgelistet werden mit  $w = w(0) \dots w(n-1)$  wobei  $w(i)$  der  $(i+1)$ te Buchstabe ist

Die Länge eines Wortes wird mit  $|\cdot|$  bestimmt. Es gilt hierbei, dass das leere Wort  $\lambda$  die Länge 0 hat, also  $|\lambda| = 0$ . Jedes Wort  $w$  hat die Länge  $|w| = |w(0) \dots w(n-1)| = n$

Die Funktion  $\#_a(w)$  gibt die Häufigkeit des Buchstabens  $a$  im Wort  $w$  an.

Das Funktionszeichen  $\circ$  beschreibt die Konkatenation von Wörtern. ( $v \circ w = vw$ )

Das Funktionszeichen  $\upharpoonright x$  beschreibt die Beschränkung eines Wortes auf die ersten  $x$  Buchstaben

Es gibt verschiedene Arten die Zahlen zu Ordnen, eine davon ist die Lexikographische Ordnung, welche den Nachteil haben kann, dass manche Wörter unendlich viele Vorgänger haben können

(Beispiel mit dem binären Alphabet  $\Sigma_2$  mit

$$\lambda <_{lex} 0 <_{lex} 00 <_{lex} 000 <_{lex} \dots <_{lex} 1$$

)

Eine Zweite Ordnung ist die Längen-lexikographische Ordnung mit der Definition

$$v < w \Leftrightarrow |v| < |w| \text{ oder } [|v| = |w| \text{ und } v <_{lex} w]$$

LOADS OF RANDOM PROBABLY NOT NEEDED STUFFS

## 3 Algorithmen

Typen von Algorithmen:

- Entscheidungsverfahren
- Aufzählungsverfahren
- Berechnungsverfahren

Ein Entscheidungsverfahren ist ein Algorithmus zur Lösung eines Entscheidungsproblems  $L$ . Es erhält eine Eingabe und kann basierend auf bestimmten Parametern bestimmen, ob die Eingabe wahr ( $x \in L$  auch  $E(X)$ ) oder falsch ( $x \notin L$  auch  $\neg E(X)$ ) ist.

Sollte  $E(X)$  oder  $\neg E(X)$  gegeben sein oder einfach zu zeigen, gilt auch das Komplement ist entscheidbar.

Ein Aufzählungsverfahren listet alle Elemente mit einer gewissen Eigenschaft (die man sich dann immer aussuchen kann). Das Verfahren nimmt keine Eingabe, gibt dann alle zutreffenden Elemente in beliebiger Reihenfolge wieder.

Ein Berechnungsverfahren nimmt eine Eingabe und führt vorgegebene Operationen durch und gibt das Resultat zurück.

### 3.1 Anforderungen an Algorithmen

Ein Algorithmus ist eine Rechenvorschrift, welche rein mechanisch ausgeführt werden kann. Heißt es muss bestimmte Anforderungen erfüllen

- Finitheit - Beschreibung ist endlich
- Determiniertheit - Ausgabe hängt nur von Beschreibung und Eingabewerten ab (selbe Eingabe muss in selbe ausgabe resultieren)
- Effektivität - Das Verfahren ist durchführbar
- Sequentialität - Die Beschreibung muss aus einzelnen Schritten bestehen, welche nacheinander ausgeführt werden.
- Determinismus - Die Reihenfolge der Schritte ist eindeutig festgelegt, wenn auch durch die Eingabe veränderbar
- Häufig wird auch gefordert: Terminierung - Algorithmus stoppt nach endlich vielen schritten

Dadurch, dass man eine einzige Antwort oder eine bestimmte Antwort erwartet, müssen Entscheidungs- und Berechnungsverfahren diese Anforderungen erfüllen. Hingegen Partielle Berechnungsverfahren und Aufzählungsverfahren müssen nicht terminieren.

### 3.2 Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit

#### 3.2.1 Entscheidbare Sprachen

Wenn wir eine Menge  $A$  haben und kennen einige Eigenschaften, können wir andere Eigenschaften daraus schließen:

- $A$  endlich  $\Rightarrow A$  Entscheidbar
- $A$  entscheidbar  $\Rightarrow \overline{A}$  Entscheidbar
- $A$  entscheidbar  $\Leftrightarrow c_A$  berechenbar
- $A$  entscheidbar  $\Leftrightarrow A$  monoton auf zählbar
- $A$  entscheidbar  $\Leftrightarrow A$  und  $\overline{A}$  auf zählbar

### 3.2.2 Aufzählbare Sprachen

[Bla Bla Bla ...]

## 4 Turingmaschinen

Turingmaschinen operieren auf Wörtern (d.h. benutzen Zahldarstellungen). Grundoperationen sind elementare Zeichenmanipulationen (Lesen, Streichen, Hinzufügen, Ersetzen einzelner Buchstaben).

Die Speicherstruktur einer Turingmaschine hat folgende Komponenten:

- Das Band hat einen Cursor, von dem auf beiden Seiten das Band ins Unendliche verläuft. Das Band ist in dementsprechend unendlich viele Felder aufgeteilt
- Jedes Feld des Bandalphabeten  $\Gamma$  wird durch einen Buchstaben beschrieben, ein Leeres Feld wird mit einem 'b' beschrieben
- Der Cursor hat zugriff auf die Daten und kann diese Lesen und neu beschreiben. Der Cursor befindet sich zu jedem Zeitpunkt auf genau einem Feld, kann aber in jedem Zeitpunkt sich nach links oder rechts bewegen

Die Eingabe und Ausgabe haben beide ihre eigenen Alphabete, welche für jeweilige Programme festgelegt sind.

Das Eingabewort befindet sich anfangs immer rechts neben dem Cursor. Außer der Eingabe ist auf dem ganzen Band nur Blanks aufzufinden. Sollten mehrere Eingaben eingegeben worden sein, werden diese durch Blanks getrennt.

Auch das Ausgabewort befindet sich am ende rechts von dem Cursor

Die Turingmaschine befindet sich zu jedem Zeitpunkt in einem von endlich viel beschriebenen Zuständen.

Für jede Rechnung beginnt die Maschine in einem Startzustand. Von hier befolgt die Maschine ein endlichen Ablauf von Instruktionen, welche durch das Programm festgelegt wurden. Eine Instruktion gibt an, welchen Ablauf die Maschine befolgen sollte. Es wird der momentane Zustand betrachtet, von da auf was für einen Buchstaben sich der Cursor befindet. Basierend auf den zwei Parametern, wird der betrachtete Buchstabe überschrieben, der Kopf bewegt sich und der Zustand wird gewechselt.

Jede Kombination von Zustand und gelesenen Buchstaben hat maximal einen Folgezustand. Das Programm wird anhand partieller Funktionen beschrieben mit folgender Form

$$\delta : Z \times \Gamma \rightarrow \Gamma \times Bew \times Z$$

Hierbei beschreibt  $Z$  die Menge an Zuständen,  $\Gamma$  das Bandalphabet und  $Bew$  die Menge der Möglichen Bewegungen.

Existiert kein Folgezustand für die gegebene Kombination, so Terminiert das Programm.

Formal ist eine Turingmaschine durch ein 7-Tupel gegeben mit der Definition  $M = (\Sigma, m, T, \Gamma, Z, z_0, \delta)$  wobei<sup>1</sup>

- $\Sigma$  das Eingabealphabet darstellt
- $T$  das Ausgabealphabet darstellt
- $m$  die Stelligkeit der zu berechnenden Funktion (Wie viele Eingabewörter existieren) darstellt
- $\Gamma$  das Bandalphabet, welches mindestens aus der Vereinigung der Ein- und Ausgabealphabete und dem  $b$  (Blank) besteht, darstellt
- $Z$  die endliche an Zuständen, welche in diesem Programm vorkommen, beschreibt
- $z_0$  ein spezielles Element aus  $Z$  welches den Anfang des Programms beschreibt
- $\delta$  eine partielle Funktion darstellt.

Eine  $k$ -Band-Turingmaschine  $M$  wird beschrieben durch

$$M = (k, \Sigma, m, T, \Gamma, Z, z_0, \delta)$$

---

<sup>1</sup>Disclaimer: Grammatik könnte hier keinen Sinn ergeben

Wobei  $k$  die Anzahl der Bänder darstellt und das  $\delta : Z \times \Gamma^k \rightarrow (\Gamma \times Bew)^k \times Z$ . Der Rest wie oben definiert ist.

Die Eingabe ist wieder rechts neben dem Cursor, nur hier wird noch spezifiziert, dass es das erste Band ist. Alle anderen Bänder sind komplett leer.

Die Ausgabe ist ebenfalls rechts vom Cursor, nur dass es hier auf dem letzten Band ist.

Man Bemerke, dass jedes Programm, welches auf einem  $k$ -Band möglich ist, auch auf einer normalen 1-Band-Turingmaschine möglich ist.

Blatt 3

Insert Beispielaufgaben here

## 5 Registermaschinen

Es wird von der Zahldarstellung abstrahiert. Grundoperationen sind elementare Zähloperationen (Inkrementieren, Dekrementieren, Nulltest).

Der Speicher besteht aus  $k$  Registern mit  $k \geq n + 1$

Wobei die Aufteilung der Register aussieht wie folgt

- Register 1 bis  $n$  sind die Eingaberegister
- Register  $n + 1$  ist das Ausgabe Register
- sofern man noch weitere Hilfsregister benötigt sind diese  $n + 2$  bis  $k$

Die Operationen werden abgekürzt mit

$a_i(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_k) = (x_1, \dots, x_{i-1}, x_i + 1, x_{i+1}, \dots, x_k)$   
 $s_i(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_k) = (x_1, \dots, x_{i-1}, x_i - 1, x_{i+1}, \dots, x_k)$  wobei beachtet werden muss, dass an jeder stelle die Rechnung  $0 - 1 = 0$

$$t_i(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_k) = \begin{cases} 1 & \text{falls } x_i = 0 \\ 0 & \text{falls } x_i > 0 \end{cases}$$

Die Beschreibung des Programmes ist wie folgt

- $(z, a_i, z')$  bzw  $(z, s_i, z')$  für die Inkrementierung bzw. Dekrementierung
- $z, t_i, z', z''$ ) um zu testen, ob das  $i$ -te Register leer ist, wenn dem so ist, ist der nächste Zustand  $z''$ , sonst  $z'$

Die formale Beschreibung von  $k$ -Registermaschinen lautet  $M = (k, n, Z, z_0, \delta)$

Blatt 5

VL 5



## **6 Registermaschinen und Registeroperatoren**

Blatt 5 VL 3

## **7 Primitiv rekursive Funktionen**

Blatt 6-9(?) VL 5

## **8 Rekursiv aufzählbare Mengen**

Blatt 8? VL 7

## **9 !Härte und Vollständigkeit**

## **10 Kontextfreie Sprachen**

VL 15?

## **11 (Rechts-)Lineare Sprachen**

VL 13+?

## **12 Reguläre Sprachen**

VL 16?

## **13 !Reduktionsmethode**

## **14 Chomsky-Hierarchie**

VL 13?

## **15 !Diagonalisierung**

## **16 Komplexitätsklassen**

VL 12?

## **17 Entscheidbarkeitsfragen**

Blatt 1, 2, 4 VL 2

## **18 Nichtkontextfreie Sprachen**

VL 15?

## **19 Endliche Automaten**

## **20 Rechtslineare Sprachen / Endliche Automaten**

VL 13+?

## **21 Kontextfreie vs. rechtslineare Sprachen**

VL 15?

## **22 Partiell rekursive Funktionen**

Blatt 10 VL 3?